DISTRIBUTED DETECTION AND DECISION FUSION USING PARTICLE FILTER CONCEPTS

by

Umut Mamıkoğlu

B.S., Electrical and Electronics Engineering, Bilkent University, 2011

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Electrical and Electronics Engineering

Boğaziçi University

2014

DISTRIBUTED DETECTION AND DECISION FUSION USING PARTICLE FILTER CONCEPTS

APPROVED BY:

Prof. Ayşın Baytan Ertüzün (Thesis Supervisor)	.....
Prof. Mustafa Aktar	.....
Assoc. Prof. Murat Saraçlar	.....

DATE OF APPROVAL: 19.08.2014

ACKNOWLEDGEMENTS

First of all, I am very grateful to my thesis advisor Prof. Ayşın Ertüzün for providing me the chance of working on this exciting and challenging topic. Her invaluable guidance and support has been great inspiration and motivation for me.

I would like to thank to the juri members of my thesis comitee, Prof. Mustafa Aktar and Assoc. Prof. Murat Saraçlar for their invaluable advices and ideas about possible future work. I would like to express my gratitude to Prof. Mustafa Aktar and Birsen Can providing the earthquake data and their guidance throughout this thesis.

I would like to thank Görkem Ülkar for his discussions and friendship.

Last, but not least, I have my special appreciation to Sarp and my family for their endless patience, love and support.

This thesis is supported by Boğaziçi University Research Fund (BAP) under the project code 12A02P4.

ABSTRACT

DISTRIBUTED DETECTION AND DECISION FUSION USING PARTICLE FILTER CONCEPTS

In this thesis, a distributed detection and decision fusion system that operates under non Gaussian noise with unknown parameters is developed. The main objective is to find decision rules for the local detectors and the fusion center without making unrealistic assumptions about statistics of the observed data. The proposed scheme is based on concepts of using particles. In order to form a dynamical model of the problem, observed data is modeled as an AR process which is driven by Gaussian mixture noise. The proposed system consists of a particle filter, used for estimating the unknown noise parameters, followed by particle swarm optimization (PSO) which achieves distributed detection and decision fusion of local decisions. The fusion rule is designed, without assuming independence of the decisions of the local sensors, by using copula functions to relate the marginal densities of the sensor observations to the statistical dependency between the sensor decisions. The parameter of the copula function used is estimated using PSO. The probability of error values obtained by using the proposed method are compared with theoretical values and promising results are obtained.

ÖZET

PARÇACIK SÜZGEÇLEME TEMELLİ DAĞITIMLI SEZİM VE KARAR TÜMLEŞTİRME

Bu tezde, Gauss olmayan ve değişken değerleri bilinmeyen gürültü altında çalışabilen bir dağıtımlı sezim ve karar tümleştirme sistemi geliştirilmiştir. Öncelikli olarak amaçlanan, yerel sezimleyici ve tümleştirme merkezi karar verme kurallarının, gözlem verisi istatistiklerine dair gerçekçi olmayan varsayımlardan kaçınılarak belirlenmesidir. Önerilen yapı parçacık temelli kavramlara dayandırılmaktadır. Problemin dinamik bir modelinin kurulabilmesi için, gözlem verisi Gauss karışımı gürültü ile özbağlanımlı süreç olarak modellenmiştir. Önerilen sistem bilinmeyen gürültü değişkenlerinin kestirimi için kullanılan bir parçacık süzgeci ve sonrasında dağıtımlı sezim ve yerel kararları tümleştirmeyi gerçekleştiren parçacık sürü optimizasyonundan (PSO) oluşmaktadır. Tümleştirme kuralı, algılayıcıların kararları arasında istatistiksel olarak bağımsızlık varsayımında bulunmadan, copula fonksiyonu kullanılarak algılayıcı kararlarının marjinal olasılık yoğunluklarının ortak olasılık yoğunluklarına ilişkilendirilmesiyle tasarlanmıştır. Kullanılan copula fonksiyonuna ait değişken PSO kullanılarak kestirilmektedir. Önerilen yöntemlerle elde edilen hata olasılıkları teorik hata olasılıkları ile karşılaştırıldığında umut vaat eden sonuçlar elde edilmiştir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

a_i	i^{th} coefficient of autoregressive process
c_g	Gaussian copula
C	Copula
C_{ij}	Cost of chosing i^{th} hypothesis when j^{th} hypothesis is true
f	Proability density function
F	Function
H_i	i^{th} hypothesis
$\mathcal{IW}$	Inverse-Wishart density
J	Number of Gaussian mixtures
k	Order of Autoregressive process
K	Correlation coefficient
m	Mean of Gaussian density
n	Observation noise
$\mathcal{N}$	Gaussian (Normal) density
p^i	Position of the i^{th} particle
$p(x)$	Probability density function of random variable x
$p_{x,y}(x,y)$	Joint probability density function of random variables x and y
$p(y\|\theta)$	Probability density of random variable y given random variable θ
P	Probability density of observation given alternative hypothesis
P_D	Probability of detection
P_{FA}	Probability of false alarm
P_M	Probability of miss
P_i	Probability of occurance of the i^{th} hypothesis
$P_{y\|H_i}(y\|H_i)$	Conditional probability density of data given the i^{th} hypothesis
q	Random noise sample
Q	Probability density of observation given null hypothesis

$\mathbb{R}$	Bayes risk function
T	Time
u_0	Decision of the fusion center
u_i	Decision of the i^{th} sensor
$\mathcal{U}$	Uniform distribution
ν^i	Velocity of the i^{th} particle
w^i	Weight of the i^{th} particle
$\tilde{w}^i$	Unnormalized weight of the i^{th} particle
x^i	i^{th} particle
x	State
y_i	i^{th} observation
Z_i	Observation space corresponding to the i^{th} hypothesis
α	Probability of false alarm
β	Probability of miss
δ	Dirac delta function
η	Fusion threshold
θ	Parameter
Θ	Parameter set
λ	Detector threshold
λ_{max}	Maximum value that threshold may take
λ_{min}	Minimum value that threshold may take
λ'	Lagrange multiplier
μ	Mean of Gaussian distribution
ν	Process noise
π	Weight of Gaussian mixture component
σ^2	Variance of Gaussian distribution
τ	Transpose operator
ψ	Scale matrix of Inverse-Wishart distribution
ϕ	The univariate Gaussian distribution function
φ_1	Weight determining the effect of particle's previous best position

φ_2	Weight determining the effect of global previous best position
Λ	Degree of freedom of Inverse-Wishart distribution
Σ	Covariance matrix of Gaussian distribution
Σ_c	Covariance matrix of Gaussian copula

LIST OF ACRONYMS/ABBREVIATIONS

AR	Autoregressive
arg min	Argument of the minimum
arg max	Argument of the maximum
KOERI	Kandilli Observatory and Earthquake Research Institute
LRT	Likelihood Ratio Test
LSE	Least Squares Error
MAP	Maximum A Posteriori
MCMC	Markov Chain Monte Carlo
MMSE	Minimum Mean-Squared Error Estimation
MLE	Maximum Likelihood Estimation
MSE	Mean Square Error
NMSE	Normalized Mean Square Error
PSO	Particle Swarm Optimization
ROC	Receiver Operating Characteristics
SIS	Sequential Importance Sampling
UDIM	National Earthquake Monitoring Center

1. INTRODUCTION

1.1. Problem Statement

Current multiple sensor-based signal processing mostly relies on a centralized scheme where the individual sensors transmit the recorded data to a central fusion center, which then either estimates the parameters of interest or makes a decision on a postulated hypothesis. Centralized scheme has some disadvantages such as usage of wide bandwidth and high computational complexity at the fusion center. Recently, signal processing with distributed sensors has gained importance. In distributed detection systems, each sensor processes information and transmits its corresponding decision to a fusion center where a global decision is made. Using a distributed scheme significantly reduces bandwidth and computational complexity in fusion center; thus increases the robustness of the overall system as the influence of a single sensor to the overall decision is reduced.

The challenge in a sensor network is to design the decision rules at the individual sensors and at the fusion center to achieve minimization of global probability of error or a Bayesian risk parameter. In addition to minimizing the probability of error, for example, other criteria such as minimization of the energy consumption and/or maximization of the rate of transmission may be taken into account. This problem, which has gained much attention in recent years, is referred to as *multi-objective optimization* [1]. Recently an optimization method, namely Particle Swarm Optimization (PSO), which aims to converge optimum thresholds dynamically based on particles is suggested in [2] where observation noise is assumed to be known and Gaussian. Also, the effectiveness of PSO in finding the thresholds of local detectors in distributed detection scenario is shown through simulations; however performance of this method has not been investigated using real data.

Often, strong assumptions such as the choice of a parametric model and a probability distribution function of the observations (typically the normal distribution) as

well as assumptions on the optimal detector design such as linearity, stationarity, uncorrelatedness or independence are made in the design of decision rules. However, this optimality is obtained only if the assumptions hold exactly, which is often unrealistic in real life. Any deterioration from these strong assumptions may lead to performance degradation and hence harm the robustness of the overall system [3], [4], [5], [6]. In particular, measurement campaigns in numerous applications such as outdoor mobile communication channels, radar and sonar systems, biomedical, seismology indicate the presence of impulsive (heavy-tailed) noise, which can cause the detector to be suboptimal or even may cause them to fail.

Impulsive noise is generally modeled as alpha stable distribution, ϵ-contaminated mixture model or Gaussian mixture model [5], [7]. Gençağa et al obtained promising results in modeling seismological events as non-stationary autoregressive (AR) processes driven by alpha stable noise [8], [9], [10]. Using a particle filter-based approach, they successfully estimated the time-varying coefficients of the AR process and the noise statistics. However, modeling impulsive noise as alpha stable distribution may not be suitable in the context of distributed detection, since alpha stable distribution cannot be expressed in a closed form and it requires numerical calculations to obtain corresponding thresholds which may be an additional burden to the processors of the local detectors.

Moreover, fusion of local decisions is an important task in distributed detection. In [4], optimum fusion rule is described by weighing the individual decisions based on the probability of detection of local detectors under the assumption that sensor decisions are independent. When this assumption does not hold, overall performance of the distributed detection system may behave poorly compared to the cases where the dependency between decisions are embedded in the fusion decision rule. Local decisions may still be correlated especially when the sensor locations are close, even when the sensors make their decisions independently. However, taking dependency of the decisions into account is not an easy task, since it requires the joint probability density of the decisions. Bahadur-Lazarfeld expansion is proposed to compute the joint probabilities [4]. Still, this approach may not be efficient when the dependency between

the decisions are nonlinear.

Recently, copula theory has been introduced to the problem of fusion of correlated decisions which provides an approach that does not necessarily require prior information about the joint statistics of the sensor decisions [11]. Using a copula function, the marginal densities of the sensor observations or the marginal probability mass functions of the sensor decisions can be related to the joint probability density function of the sensor decisions. Iyengar et al. proposes a copula solution to the joint processing of heterogeneous data, specifically for a binary classification problem, where the observed signals may be statistically dependent [12]. For the cases when parameters of the copula function used is unknown, estimation of these parameters are attempted using maximum likelihood estimation [11], [12].

Alternatively, sequential Monte Carlo methods can be used in the estimation of the copula parameters [13], [14]. In this approach, a dynamic model is formed to express the dependency between data sets by writing them as factor models that are known priorly. However this approach may not be applied to the estimation of the copula parameters used in the description of the dependency between local decisions, when a dynamic model to describe this relation could not be formed using state space equations.

In this thesis, a distributed detection scheme shown in Figure 1.1 is considered. The main objective is to find decision rules for the local detectors and the fusion center while avoiding to make unrealistic assumptions. Observation noise is modeled as Gaussian mixture which may express multi-modalities or heavy-tails due to selection of the parameters and also it is a generalization of ϵ-contaminated mixture model. Therefore, we avoid Gaussianity assumption of observation noise and aim to infer its parameters from observed data in a sequential way using particle filtering. The reason for the operation of the system to be in a sequential manner is to provide adaptivity to the cases where statistical properties of data change over time. In order to form a dynamical model of the problem, observed data is modeled as an AR process which is driven by Gaussian mixture noise. The parameters of the observation noise as well as

the coefficients of the AR process are not presumably known and aimed to be estimated using particle filters. As the noise parameters are estimated, local decision rules are designed using the knowledge of noise parameters within PSO framework. Estimates of the AR coefficients are not used in the PSO algorithm, however this step enables us to investigate the most general version of our problem, i.e., when both the AR coefficients and the noise parameters are unknown. Moreover, inferring additional knowledge from the AR coefficients may be utilized in a future work of an early warning system for the detection of earthquakes.

The thresholds of the local detectors are found within PSO framework using estimated parameters of Gaussian mixture noise. For each observation, each local detector makes a decision according to its threshold value which is obtained as weighted sum of thresholds corresponding to the mixture parameters of the Gaussian mixture noise. Then, local decisions are transmitted to the fusion center. We design the fusion rule without assuming independence of the decisions of the local sensors. In order to form the statistical dependency between the sensor decisions, *copula theory* is used. Copula theory enables us to use copula functions to relate the marginal densities of the sensor observations to the statistical dependency between the sensor decisions. Also, we do not assume that the copula parameters are known; rather we aim to estimate the parameter of the used copula by using PSO. For simplification Gaussian copula is used as copula function, since it requires one parameter to be estimated.

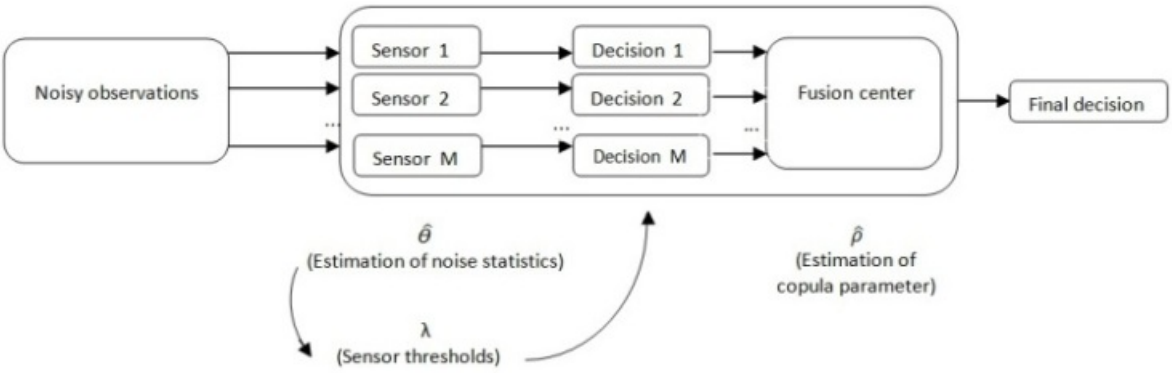

Figure 1.1. Block diagram of the proposed distributed detection and decision fusion system.

Performance of the estimation of the observation noise parameters using particle filtering are investigated in the experiments using both simulated and real seismological data. Using these estimations, thresholds of the local detectors, and then local decisions, are obtained. Local decisions are fused after estimating the copula parameter of the copula function which yields the joint probability distribution of the local decisions. Performance of the overall distributed detection system is measured by the probability of error of the system.

1.2. Contributions of the Thesis

In this work, particle based methods are applied to the problem of distributed detection and decision fusion. The problem of finding decision rules for the local detectors and the fusion center where the statistics of data are unknown or changing with time is attempted using particle based methods, namely as PSO and particle filtering, in a sequential manner. Therefore, typical assumptions made in design of decision rules of distributed detection systems such as linearity, Gaussianity, stationarity of the parametric model and uncorrelatedness of decisions are avoided. Our main contribution in this work is to propose modeling observations as AR process that is driven by Gaussian mixture noise which enables estimation of observation noise parameters sequentially using particle filtering. Another contribution of this work is to introduce decision rules for the fusion center in which the correlation among decisions are estimated using particles. Here, copula theory is utilized in forming joint distribution densities of the local decisions where the unknown parameter of the copula function is estimated using PSO. Hence, a modified version of fusion rule is suggested which is shown to increase performance of distributed detection system. Moreover, performance of the proposed distributed detection system is tested in real seismological data for detection of earthquakes which are obtained from different seismic stations in Turkey by Boğaziçi University Kandilli Observatory and Earthquake Research Institute.

1.3. Organization of the Thesis

The organization of this thesis is as follows:

In Chapter 2, classical and modern approaches to detection theory are reviewed and the distributed detection and decision problem is introduced. Application of the PSO in the context of distributed detection is reviewed.

In Chapter 3, estimation theory is reviewed. Sampling techniques are introduced and particle filtering is briefly explained.

Chapter 4 starts with a general description of the proposed distributed detection system. Then, the proposed distributed detection system is explained in details within three main parts: AR modeling of observations and estimation of noise parameters, decision rules of the local detectors and decision fusion. Implementation details and experimental results of the methods used are provided.

In Chapter 5, proposed distributed detection system is tested with real earthquake data and results are compared with the results of the simulated data.

Chapter 6 concludes the thesis and includes comments about the future work on the subject.

2. DETECTION THEORY

In this chapter, we introduce detection theory concepts. We start with the hypothesis testing, Neyman Pearson formulation of the detection problem and the Bayes' formulation of the detection problem. Then, we relate the PSO algorithm to the Bayesian approach of detection problem.

2.1. Hypothesis Testing

A statistical hypothesis is an assertion or a claim regarding the true value of an unknown population parameter. Hypothesis testing is the use of statistics to determine the probability of a given hypothesis to be true. Hypothesis testing is formulated in terms of two hypotheses: null and alternative. The *null hypothesis*, denoted by H_0, represents the case when the assertation does not hold and it is the assumed outcome of the experiments if the tested assertation is false. The *alternative hypothesis*, denoted by H_1, is the opposite, complementing H_0. Thus, the hypothesis testing procedure is about gathering evidence to find out if the null hypothesis H_0 should be rejected and the alternative hypothesis H_1 accepted, or not. Statistical evidence is gathered from the observation space that corresponds to a set of N observations: y_1,y_2,...,y_N. The conditional probability densities $p_{\mathbf{y}|H_0}(\mathbf{Y}|H_0)$ and $p_{\mathbf{y}|H_1}(\mathbf{Y}|H_1)$, that are assumed to be completely known, map hypotheses to points in observation space according to a *decision rule* [15]. There are four possible outcomes for each time a decision is made:

- H_0 is true and H_0 is chosen.
- H_0 is true and H_1 is chosen.
- H_1 is true and H_0 is chosen.
- H_1 is true and H_1 is chosen.

The choise of decision rule generally affects the values of two types of error in hypothesis testing:

- False alarm (α): Rejecting H_0 when it is true.
- Miss (β): Accepting H_0 when it is false.

where α and β denote the probabilities of *false alarm* and *miss*, which are denoted as P_{FA} and P_M, respectively. The probability of accepting H_1 when it is true is referred to as probability of *detection* and denoted as P_D.

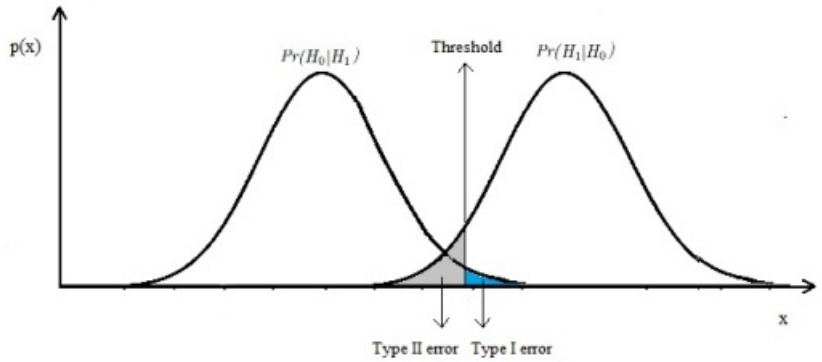

Figure 2.1. Possible hypothesis testing errors.

Two of the most common decision criteria, the Neyman-Pearson and the Bayes are presented next.

2.1.1. Bayesian Criterion

Bayes test is based on usage of prior knowledge upon the probability of occurrences of two hypotheses which are called as the *a priori* probabilities and denoted as P_0 and P_1 . For each outcome of a decision, costs are assigned as C_{00}, C_{10}, C_{11}, C_{01} where the first subscripts denote the hypothesis chosen and true hypothesis, respectively. Bayesian criterion is formed in order to minimize the *average cost* or the risk $\mathbb{R}$ which is defined as [15]:

$$\begin{aligned}\mathbb{R} = {} & C_{00}P_0Pr(choose\ H_0|H_0\ is\ true) + C_{10}P_0Pr(choose\ H_1|H_0\ is\ true) \\ & + C_{11}P_1Pr(choose\ H_1|H_1\ is\ true) + C_{01}P_1Pr(choose\ H_0|H_1\ is\ true). \quad (2.1)\end{aligned}$$

Since there are two hypotheses, the decision rule divides the observation space Z into two regions: Z_0 and Z_1. Then the risk $\mathbb{R}$ can be written in terms of the transition probabilities and decision regions:

$$\mathbb{R} = C_{00}P_0 \int_{Z_0} p_{\mathbf{y}|H_0}(\mathbf{Y}|H_0)d\mathbf{Y} + C_{10}P_0 \int_{Z_1} p_{\mathbf{y}|H_0}(\mathbf{Y}|H_0)d\mathbf{Y} + C_{11}P_1 \int_{Z_1} p_{\mathbf{y}|H_1}(\mathbf{Y}|H_1)d\mathbf{Y} + C_{01}P_1 \int_{Z_0} p_{\mathbf{y}|H_1}(\mathbf{Y}|H_1)d\mathbf{Y}. \quad (2.2)$$

Regions Z_0 and Z_1 are chosen such that $\mathbb{R}$ is minimized. Using the fact that $Z{=}Z_0{+}Z_1$ and ignoring the fixed costs, decision regions are generally defined as:

$$\frac{p_{\mathbf{y}|H_1}(\mathbf{Y}|H_1)}{p_{\mathbf{y}|H_0}(\mathbf{Y}|H_0)} \underset{H_0}{\overset{H_1}{\gtrless}} \frac{P_0(C_{10}-C_{00})}{P_1(C_{01}-C_{11})}, \quad (2.3)$$

where the quantity on the left is named as the *likelihood ratio* and denoted by $\Lambda(\mathbf{Y})$. The quantity on the right of (2.3) is the *threshold* of the test and denoted by λ. Thus, Bayesian criterion leads to *likelihood ratio test* (LRT) as:

$$\Lambda(\mathbf{Y}) \underset{H_0}{\overset{H_1}{\gtrless}} \lambda. \quad (2.4)$$

2.1.2. Neyman-Pearson Criterion

Neyman-Pearson approach to hypothesis testing uses error probabilities as conflicting objectives where probability of false alarm (P_{FA}) is constrained and probability of detection (P_D) is maximized [15].

Neyman-Pearson Criterion: Constrain $P_{FA} = \alpha' \leq \alpha$ and design a test to maximize P_D under this constraint [15].

This constraint on P_{FA} can be satisfied by chosing λ such that:

$$P_{FA} = \int_{\lambda}^{\infty} p_{\boldsymbol{y}|H_0}(\mathbf{Y}|H_0)dY = \alpha', \tag{2.5}$$

where the density of $\boldsymbol{y}$ when H_0 is true is denoted as $p_{\boldsymbol{y}|H_0}(\mathbf{Y}|H_0)$. Similarly, the value of P_D for the given threshold can be found as:

$$P_D = \int_{\lambda}^{\infty} p_{\boldsymbol{y}|H_1}(\mathbf{Y}|H_1)dY, \tag{2.6}$$

where the density of $\boldsymbol{y}$ when H_1 is true is denoted as $p_{\boldsymbol{y}|H_1}(\mathbf{Y}|H_1)$.

A popular approach to Neyman Pearson criterion makes use of Lagrange multipliers by constructing the function F [16]:

$$F = P_M + \lambda'[P_{FA} - \alpha], \tag{2.7}$$

where P_M is probability of miss which is equal to $(1 - P_D)$ and λ' is the Lagrange multiplier. Substituting P_M and P_{FA} in the function F results in:

$$F = \lambda'(1 - \alpha) + \int_{-\infty}^{\lambda} \left[p(\boldsymbol{y}|H_1) - \lambda' p(\boldsymbol{y}|H_0)\right] dY. \tag{2.8}$$

The first term in Eq. (2.8) is fixed, therefore minimizing the second term yields LRT to be expressed as follows :

$$\Lambda(\mathbf{Y}) = \frac{p(\boldsymbol{y}|H_1)}{p(\boldsymbol{y}|H_0)} \underset{H_0}{\overset{H_1}{\gtrless}} \lambda'. \tag{2.9}$$

The threshold λ of the test is the Lagrange multiplier λ' which satisfies the constraint in (2.5).

2.1.3. Partical Swarm Optimization (PSO)

The PSO algorithm is a computational method that optimizes a given function iteratively. It was originally represented by Kennedy and Eberhart in 1995 as a simulation of social behaviors in a community [17]. The PSO algorithm has been applied to multi-dimensional optimization problems where each *particle* is a possible solution to the multi-dimensional optimization problem [18]. Particles continuously seek through a multi-dimensional search space to find the optimum solution. Each solution is evaluated upon a multi-objective performance function related to the optimization problem being solved. As shown in Figure 2.2, the trajectories of the particles are affected by the best solution achieved so far, called *pbest*, and the best solution achieved amongst all particles, called *gbest* [19]. At each iteration, *pbest* and *gbest* are updated for each particle. The algorithm continues until particles converge to the optimum solution of the problem or the maximum number of iterations are reached.

Each particle in M-dimensional space is represented as $\boldsymbol{x}^i = (x^i_1, x^i_{2,}..., x^i_M)$ where $i = \{1, 2, ...N\}$ is the particle number. The previous best positions (*pbest*) for each particle are represented as $\boldsymbol{p}^i = (p^i_1, p^i_{2,}..., p^i_M)$ and the velocity of the particles are as $\boldsymbol{\nu}^i = (\nu^i_1, \nu^i_{2,}..., \nu^i_M)$. At time instant t, the velocity of each particle is updated as:

$$\nu^i_{t,m} = w.\nu^i_{t-1,m} + \mathcal{U}(0,1).\varphi_1.(pbest^i_{t-1,m} - x^i_{t-1,m}) + \mathcal{U}(0,1).\varphi_2.(gbest_{t-1,m} - x^i_{t-1,m}), \tag{2.10}$$

where $\mathcal{U}(0,1)$ is a sample of uniform random number generator, φ_1 is a weight determining the effect of *pbest,* φ_2 is a weight determining the effect of *gbest* and w is a weight determining the effect of particle's previous velocity on particle's updated velocity. The value of the particle is updated as:

$$x^i_{t,m} = x^i_{t-1,m} + \nu^i_{t,m}. \tag{2.11}$$

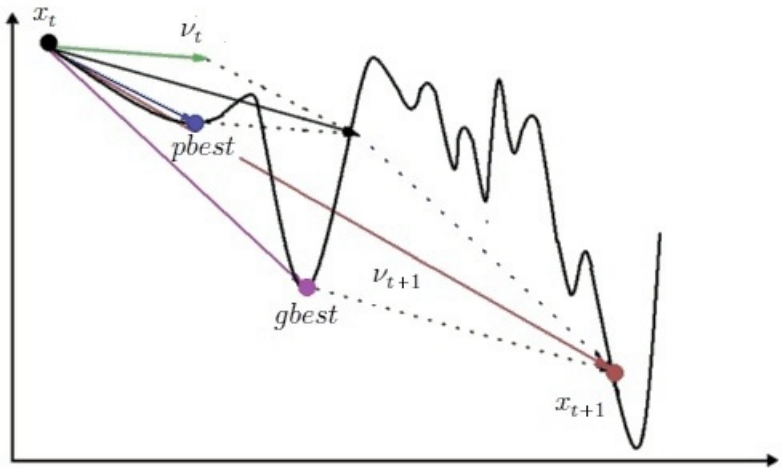

Figure 2.2. Update of a particle in PSO algorithm.

2.1.3.1. Particle Swarm Optimization in Distributed Detection Scheme. In the distributed detection problem, determination of local sensor thresholds are crucial. Sensor thresholds can be determined by optimization of a multi-objective function that is composed of sensors' P_{FA} and P_D values. In this scheme, sensor thresholds are represented by particles of M-dimension and the multi-objective optimization function P_e can be written as a function of sensor thresholds $(\lambda_1, \lambda_2, ..., \lambda_M)$ as [2]:

$$P_e(\boldsymbol{\lambda}) = Pr(H_0)Pr(H_1|H_0) + Pr(H_1)Pr(H_0|H_1), \tag{2.12}$$

where $Pr(H_1)$ and $Pr(H_0)$ denote the prior probability on H_1 and H_0, respectively. $Pr(H_1|H_0)$ is the probability of false alarm and $Pr(H_0|H_1)$ is the probability of miss which are described in (2.2). Dividing (2.7) by $Pr(H_1)$ and using the fact that $Pr(H_0|H_1) = 1 - Pr(H_1|H_1)$ results in that minimizing probability of error $P_e(\boldsymbol{\lambda})$ is equivalent to maximizing $\widetilde{P}_e(\boldsymbol{\lambda})$ as:

$$\widetilde{P}_e(\boldsymbol{\lambda}) = Pr(H_1|H_1) - \frac{Pr(H_0)}{Pr(H_1)}Pr(H_1|H_0). \tag{2.13}$$

Initially, each of the particles are sampled independently and uniformly over a region $[\lambda_{min}, \lambda_{max}]$, which includes all the values that each particle can take. Positions

of particles are propagated using the previous particle positions ($\boldsymbol{\lambda}_{t-1}^{i}$), the previous best particle ($\boldsymbol{\lambda best}_{t-1}$) and random noise samples $\boldsymbol{q}$ as:

$$\boldsymbol{\lambda}_{t}^{i} = \frac{\boldsymbol{\lambda}_{t-1}^{i} + \boldsymbol{\lambda best}_{t-1}}{2} + \boldsymbol{q}. \tag{2.14}$$

Then, the particle weights are updated as:

$$w_{t}^{i} = w_{t-1}^{i} \widetilde{P}_{e}(\boldsymbol{\lambda}_{t}^{i}). \tag{2.15}$$

Sequentially, the algorithm converges to the particles which maximize $\widetilde{P}_{e}(\boldsymbol{\lambda})$, hence the set of thresholds $(\lambda_1, \lambda_2, ..., \lambda_M)$ which minimizes the probability of error can be obtained.

2.2. Distributed Detection

This section introduces the distributed detection problem which has been comprehensively studied by Varshney et. al in his work [4]. Decision making can be extended to cases where multiple decision makers are used. In classical multisensor detection, all the sensors send their data to a central processing unit where decisions are made. If sensors have processing capability, then certain amount of computation can be performed locally and sensors also make decisions. In decentralized processing, local sensors process data and their local decisions are sent to the central processing unit which is generally referred to as the *fusion center*.

Distributed detection can be done in various configurations such as parallel, serial or tree [3]. In paralel topology, each sensor makes hard decisions and send them to a fusion center. However, sending less information to the fusion center may result in performance degradation compared to classical detection methods. There are also variations of parallel topology where additional information is monitored by the fusion center. In some applications where a global decision is not necessary, sensors can make local decisions in order to optimize a systemwide objective function. In this work

parallel topology is selected due to its ease of implementation and wide usage.

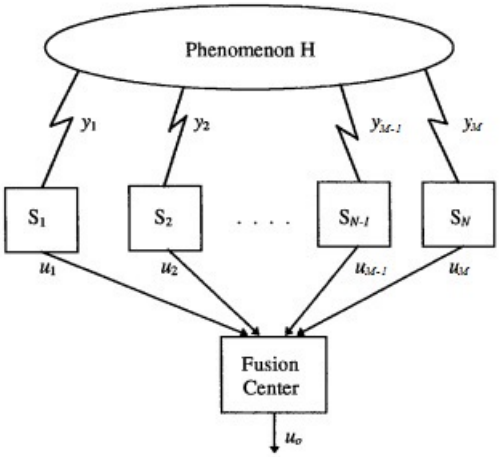

Figure 2.3. Distributed detection with fusion in parallel configuration.

In serial (or tandem) detection topology, all of the sensors are connected in series and receive observations from the common phenomenon directly. The first sensor makes a decision upon its observations and send its decision to the second sensor. The second sensor makes a decision upon its observations and the decision of the first sensor. Each sensor after the first one makes its decision upon its observations and the decision of the previous sensor (see Figure 2.4). This procedure continues until the last sensor in the network makes its decision [4].

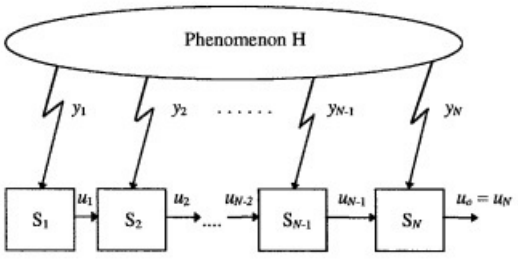

Figure 2.4. Distributed detection in serial configuration.

Another topology used in distributed detection systems is a tree network. In a tree network, the configuration is represented by a directed tree (see Figure 2.5). The fusion center becomes the root of the tree and local sensors send their information to the fusion center using a certain path which is defined by a communication matrix [3].

Finding optimum thresholds for tree networks are generally difficult to derive, since it requires solving set of nonlinear equations. More information about tree networks can be found in [20].

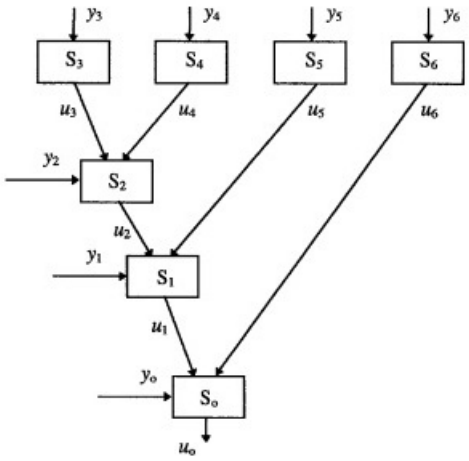

Figure 2.5. Distributed detection in tree configuration.

2.2.1. Distributed Detection in Parallel Configuration without Fusion

We consider a distributed detection system with M non-identical nodes as shown in Figure 2.6. Each of the sensors observes a common phenomenon and makes decisions locally. In this topology, it is assumed that sensors do not communicate with each other and a fusion center. However due to the fact that systemwide optimization is performed, sensors' operations are assumed to be coupled [4]. The observation of the j^{th} sensor is denoted as y_j and its decision is denoted as u_j. Prior probabilities of the null hypothesis H_0 and the alternative hypothesis H_1 are denoted as P_0 and P_1, respectively.

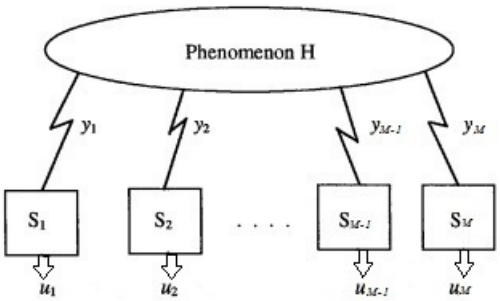

Figure 2.6. M-sensor distributed detection without a fusion center.

Bayesian approach of distributed detection aims to minimize the average cost of decision making. For the two-sensor case, assuming independence of the sensors, the Bayes risk function $\mathbb{R}$ can be written as follows [4]:

$$\mathbb{R} = \sum_{i,j,k} \int_{y_1,y_2} P_k C_{ijk} p(u_1|y_1) p(y_1, y_2|H_k), \tag{2.16}$$

where C_{ijk} represents the cost of detector 1 deciding H_i and detector 2 deciding H_j when H_k is present. Likelihood ratio at detector 1 is given by the following expression [4]:

$$\Lambda(y_1) = \frac{p(y_1|H_1)}{p(y_1|H_0)} \underset{u_1=0}{\overset{u_1=1}{\gtrless}} \frac{P_0 \sum_j \int_{y_2} p(u_2|y_2) p(y_2|y_1, H_0) \left[C_{1j0} - C_{0j0}\right]}{P_1 \sum_j \int_{y_2} p(u_2|y_2) p(y_2|y_1, H_1) \left[C_{1j1} - C_{1j1}\right]} \tag{2.17}$$

and threshold at the detector 1 is given by [3]:

$$\lambda_1 = \frac{P_0 \int_{y_2} p(y_2|H_0) \left\{\left[C_{110} - C_{010}\right] + p(u_2 = 0|y_2) \left[C_{100} - C_{000} - C_{010} - C_{110}\right]\right\}}{P_1 \int_{y_2} p(y_2|H_0) \left\{\left[C_{011} - C_{111}\right] + p(u_2 = 0|y_2) C_{001} - C_{101} - C_{111} - C_{011}\right\}}. \tag{2.18}$$

Likelihood ratio and threshold of detector 2 can be written in a similar manner:

$$\Lambda(y_2) = \frac{p(y_2|H_1)}{p(y_2|H_0)} \underset{u_2=0}{\overset{u_2=1}{\gtrless}} \frac{P_0 \sum_j \int_{y_1} p(u_1|y_1) p(y_1|y_2, H_0) \left[C_{j10} - C_{j00}\right]}{P_1 \sum_j \int_{y_1} p(u_1|y_1) p(y_1|y_2, H_1) \left[C_{j11} - C_{j11}\right]} \tag{2.19}$$

and

$$\lambda_2 = \frac{P_0 \int_{y_1} p(y_1|H_0) \left\{\left[C_{110} - C_{010}\right] + p(u_1 = 0|y_1) \left[C_{010} - C_{000} + C_{100} - C_{110}\right]\right\}}{P_1 \int_{y_1} p(y_1|H_0) \left\{\left[C_{011} - C_{111}\right] + p(u_1 = 0|y_1) C_{001} - C_{011} + C_{111} - C_{101}\right\}}. \tag{2.20}$$

As can be seen from the above that the threshold of detector 1 is a function of the threshold at detector 2 and, similarly, the threshold of detector 2 is a function of

the threshold of detector 1. Finding thresholds that satisfy these coupled equations results in finding locally optimum solutions. When there are multiple local minima, these solutions must be examined to find the globally optimum solution.

2.2.2. Decision Fusion

In a distributed detection system, fusion center makes a global decision u_0 that is:

$$u_0 = \begin{cases} 0, & if\ H_0\ is\ decided \\ 1, & otherwise \end{cases}. \tag{2.21}$$

As shown in Figure 2.3, local decisions are transmitted to a fusion center where the local decisions are processed to obtain a global decision. The objective is to determine the fusion rule to combine local decisions according to an optimization criterion. In a distributed detection system where there are M sensors, there are 2^{2^M} fusion rules. Some commonly used logical functions as decision rules are AND and OR functions which are represented in Table 2.1 by f_2 and f_8, respectively.

Table 2.1. Fusion decisions for two sensors.

u_1	u_2	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Selecting the fusion rule among all possible choices may not provide a successful combination of the local decisions. Some of the fusion rules such as all zero function f_1 and all one function f_{16} would not be preferred in most situations, since these fusion rules decide regardless of the local decisions. Design of a fusion rule can be carried out using a Bayesian approach by minimizing an average cost.

The probabilities of false alarm, detection and miss of i^{th} sensor are, respectively, given as follows:

$$P_{FAi} = p(u_i = 1|H_0), \tag{2.22}$$

$$P_{Di} = p(u_i = 1|H_1) \tag{2.23}$$

and

$$P_{Mi} = p(u_i = 0|H_1). \tag{2.24}$$

Similarly, the probabilities of false alarm, detection and miss of the fusion center are denoted as P_{FA}, P_D and P_M, respectively.

Determination of a fusion rule can be regarded as a detection problem of two hypotheses with two sensors where sensors' decisions are used as observations. The cost of global decision being H_i when H_j is present is denoted as C_{ij}. Then, the optimum fusion rule is given as follows [4]:

$$\frac{p(u_1, u_2, ..., u_M|H_1)}{p(u_1, u_2, ..., u_M|H_0)} \underset{u_0 = 0}{\overset{u_0 = 1}{\gtrless}} \frac{P_0(C_{10} - C_{00})}{P_1(C_{01} - C_{11})} \triangleq \eta, \tag{2.25}$$

where η represents threshold of the fusion center. Assuming independence of the sensors, left hand side of (2.25) can be written as follows:

$$\begin{aligned}\frac{p(u_1,u_2,...,u_M|H_1)}{p(u_1,u_2,...,u_M|H_0)} &= \prod_{i=1}^{M}\frac{p(u_i|H_1)}{p(u_i|H_0)} \\ &= \prod_{S_1}\frac{p(u_i=1|H_1)}{p(u_i=1|H_0)}\prod_{S_0}\frac{p(u_i=0|H_1)}{p(u_i=0|H_0)} \\ &= \prod_{S_1}\frac{1-P_{Mi}}{P_{FAi}}\prod_{S_0}\frac{P_{Mi}}{1-P_{FAi}},\end{aligned} \tag{2.26}$$

where S_1 and S_0 represent the sets of all local decisions that are equal to 1 and 0, respectively. Taking logarithm of (2.25) and substituting logarithm of the (2.26), (2.25) can be written as follows:

$$\sum_{S_1}\log(\frac{1-P_{Mi}}{P_{FAi}})+\sum_{S_0}\log(\frac{P_{Mi}}{1-P_{FAi}}) \underset{u_0=0}{\overset{u_0=1}{\gtrless}} \log\eta, \tag{2.27}$$

which can also be expressed as:

$$\sum_{i=1}^{M}\left[\log\frac{(1-P_{Mi})(1-P_{FAi})}{P_{Mi}P_{FAi}}\right]u_i+\sum_{i=1}^{M}\log(\frac{P_{Mi}}{1-P_{FAi}}) \underset{u_0=0}{\overset{u_0=1}{\gtrless}} \log\eta, \tag{2.28}$$

which implies that the optimum fusion rule is a comparison between a weighted sum of the local decisions and a threshold where the weights are determined according to P_M and P_{FA} values of the sensors while the threshold depends on the prior probabilities of hypotheses and the costs as well as P_M and P_{FA} values of the sensors.

2.2.2.1. Fusion of Correlated Decisions. When the local decisions are statistically independent, the optimum fusion rule is described by (2.28). In cases where knowledge of the joint distribution describing the dependecy between local decisions is not available, fusion rule is designed under the assumption that decisions are conditionally independent which may result in performance degradation [11]. When the sensor decisions

collected at the fusion center are correlated, left hand side of (2.25) includes correlation of the local decisions which leads to a more complex formulation of the problem. In order to solve this problem, the Bahadur-Lazarfeld series expansion of probability density functions is used in [21]. Then, optimum fusion rule in (2.28) is expressed as follows:

$$\sum_{i=1}^{M}\left[\log\frac{(1-P_{Mi})(1-P_{FAi})}{P_{Mi}P_{FAi}}\right]u_i+\sum_{i=1}^{M}\log(\frac{P_{Mi}}{1-P_{FAi}})+$$

$$\log\frac{1+\sum_{i<j}K^1_{ij}z^1_iz^1_j+\sum_{i<j<k}K^1_{ijk}z^1_iz^1_jz^1_k+...+K^1_{12..n}z^1_1z^1_2...z^1_n}{1++\sum_{i<j}K^0_{ij}z^0_iz^0_j+\sum_{i<j<k}K^0_{ijk}z^0_iz^0_jz^0_k+...+K^0_{12..n}z^0_1z^0_2...z^0_n}\begin{array}{c}u_0=1\\ \gtrless\\ u_0=0\end{array}\log\eta, \quad (2.29)$$

where

$$z^h_i=\frac{u_i-p(u_i=1|H_h)}{\sqrt{p(u_i=1|H_h)\left[1-p(u_i=1|H_h)\right]}},\ h=0,1 \quad (2.30)$$

and

$$\begin{aligned}K^h_{ij}&=\sum_u z^h_iz^h_jp(\boldsymbol{u}|H_h),\\ K^h_{ijk}&=\sum_u z^h_iz^h_jz^h_kp(\boldsymbol{u}|H_h),\\ &...\\ K^h_{12...n}&=\sum_u z^h_1z^h_2...z^h_np(\boldsymbol{u}|H_h),\end{aligned} \quad (2.31)$$

where correlation coefficients between the sensors $\{1,2,...,n\}$ under hypothesis h are denoted as $K^h_{12...n}$. When conditional correlation coefficients are all zero, (2.29) reduces to (2.28).

This approach requires complete knowledge of the form of the joint distribution of sensor observations. Moreover, when dependency between sensor decisions are non-linear, this approach may not be efficient. In [11], *copula theory* is introduced to the

problem of fusion of correlated decisions which provides an appoach that does not necessarily require prior information about the joint statistics of the sensor decisions. The joint probability density function of the sensor decisions are determined using only the knowledge of the marginal densities of the sensor observations or the marginal probability mass functions of the sensor decisions using copulas. The usage of copulas while relating marginal distribution functions to a multivariate probability distribution function is stated by Sklar's theorem [22], [23].

Sklar's Theorem: *Consider an M-dimensional distribution function F with marginal distribution functions $F_1, F_2, ..., F_M$. Then there exists a copula C, such that for all $x_1, x_2, ..., x_M$ in $[-\infty, \infty]$,*

$$F(x_1, x_2, ..., x_M) = C(F_1(x_1), F_2(x_2), ..., F_M(x_M)). \tag{2.32}$$

Using Sklar's theorem, the joint probability distributions in (2.25) are expressed in terms of copulas. For the two-sensor case, the joint densities of the decisions given the hypotheses at time instant t become as follows:

$$p(u_1, u_2|H_1) = P_{00}^{(1-u_{1t})(1-u_{2t})} P_{01}^{(1-u_{1t})u_{2t}} P_{10}^{u_{1t}(1-u_{2t})} P_{11}^{u_{1t}u_{2t}} \tag{2.33}$$

and

$$p(u_1, u_2|H_0) = Q_{00}^{(1-u_{1t})(1-u_{2t})} Q_{01}^{(1-u_{1t})u_{2t}} Q_{10}^{u_{1t}(1-u_{2t})} Q_{11}^{u_{1t}u_{2t}}, \tag{2.34}$$

where P_{ij} and Q_{ij} denote the joint probabiliry mass functions of the sensor decisions $\{i, j = 0, 1\}$ which can be expressed using copula function C as follows:

$$\begin{aligned} P_{00} &= C(1 - p_1, 1 - p_2), \\ P_{01} &= 1 - p_1 - C(1 - p_1, 1 - p_2), \\ P_{10} &= 1 - p_2 - C(1 - p_1, 1 - p_2), \\ P_{11} &= p_1 + p_2 + C(1 - p_1, 1 - p_2) - 1 \end{aligned} \tag{2.35}$$

and

$$\begin{aligned} Q_{00} &= C(1-q_1, 1-q_2), \\ Q_{01} &= 1 - q_1 - C(1-q_1, 1-q_2), \\ Q_{10} &= 1 - q_2 - C(1-q_1, 1-q_2), \\ Q_{11} &= q_1 + q_2 + C(1-q_1, 1-q_2) - 1, \end{aligned} \tag{2.36}$$

where the probabilities p_i and q_i $\{i = 1, 2\}$ are defined, respectively, as follows:

$$p_i = \int_{-\infty}^{\lambda_i} f(y_i|H_1) \tag{2.37}$$

and

$$q_i = \int_{-\infty}^{\lambda_i} f(y_i|H_0). \tag{2.38}$$

For the two-sensor case, after obtaining the joint probability density of sensor decisions and taking logarithm of both sides, (2.26) becomes:

$$\begin{aligned} \log\left[\frac{p(u_1, u_2|H_1)}{p(u_1, u_2|H_0)}\right] = \log\left(\frac{P_{10}Q_{00}}{P_{00}Q_{10}}\right) \sum_{t=1}^{T} u_{1,t} + \log\left(\frac{P_{01}Q_{00}}{P_{00}Q_{01}}\right) \sum_{t=1}^{T} u_{2,t} \\ + \log\left(\frac{P_{00}P_{11}Q_{01}Q_{10}}{P_{01}P_{10}Q_{00}Q_{11}}\right) \sum_{t=1}^{T} u_{1,t}u_{2,t}, \end{aligned} \tag{2.39}$$

where $t = 1, 2, .., T$ represent time value.

3. ESTIMATION THEORY

In this chapter, we introduce estimation theory concepts. We start with briefly explaining three of the most common classical estimation methods and sampling techniques. Then, the sampling techniques are related to the sequential estimation methods such as sequential importance sampling.

3.1. Classical Estimation

Estimation theory attempts to solve the problem of extracting values of parameters from continuous-time signals or data sets. Suppose there is a T-point data set $\boldsymbol{y} = \{y_1, y_2, ..., y_T\}$ that depends on an unknown parameter θ. Based on the data set, an *estimator* is defined as:

$$\hat{\theta} = g(\boldsymbol{y}), \tag{3.1}$$

where g is some function. Generally, the set of data $\boldsymbol{y}$ is assumed to be random and have a probability distribution $p(\boldsymbol{y}|\theta)$ that is *parametrized* by the unknown parameter θ.

The three types of most common estimation procedures are presented next.

3.1.1. Minimum Mean-Square Error Estimation (MMSE)

One common measure of accuracy of the estimator is the *mean square error* that is defined as [24]:

$$MSE(x) \triangleq E[|\hat{x} - x|^2]. \tag{3.2}$$

Suppose X and Y are two random variables that are independently and identically distributed according to a joint probability density function $p_{x,y}(x, y)$. The observed

data set is Y and the estimator gives the predicted value of X by using a transformation g as:

$$\hat{X} = g(X, \theta). \tag{3.3}$$

The mean square error of the estimation can be written as:

$$E[|\hat{X} - X|^2] = E[|g(Y, \theta) - X|^2]. \tag{3.4}$$

The process of finding the parameter $\hat{\theta}_{MMSE}$ that minimizes the mean square error is known as the *minimum mean square error estimator* and defined as:

$$\hat{\theta}_{MMSE} = \text{argmin}_\theta(E[|g(Y, \theta) - X|^2]). \tag{3.5}$$

In cases where the joint probability of density function is unknown, the parameter θ can be estimated using the data set of samples (x_t, y_t) where $t = 1, 2, ..., T$ using the *least squares error* (LSE) criterion:

$$\hat{\theta}_{LSE} = \text{argmin}_\theta \sum_{i=1}^{T} (|g(y_i, \theta) - x_i|^2). \tag{3.6}$$

3.1.2. Maximum Likelihood Estimation (MLE)

Assume that a set of random samples (observations) are independently drawn from a probability density function $p(y|\theta)$ and denoted as $\boldsymbol{y} = y_1, y_2, ..., y_N$. Since the samples are independently drawn, their joint likelihood can be written as [15]:

$$p(\boldsymbol{y}|\theta) = \prod_{i=1}^{T} p(y_i|\theta). \tag{3.7}$$

The maximum likelihood estimate is the parameter $\hat{\theta}_{MLE}$ that maximizes the joint likelihood as:

$$\hat{\theta}_{MLE} = \text{argmax}_{\theta} p(\boldsymbol{y}|\theta) = \text{argmax}_{\theta} \prod_{i=1}^{T} p(y_i|\theta). \tag{3.8}$$

Since the logarithm is a monotonically decreasing function, MLE is generally written in log-likelihood form as:

$$log(\hat{\theta}_{MLE}) = \text{argmax}_{\theta} \log(p(\boldsymbol{y}|\theta)) = \text{argmax}_{\theta} \sum_{i=1}^{T} \log(p(y_i|\theta)). \tag{3.9}$$

3.1.3. Maximum A Posteriori (MAP) Estimation

MMSE, LSE and MLE methods assume that the parameter θ is unknown but deterministic. However, MAP estimation assumes that the parameter θ is also a random variable and has a prior probability distribution $p(\theta)$. Given the observed data $\boldsymbol{y}$, the posterior probability distribution of the parameter θ can be found using the Bayes' rule as:

$$p(\theta|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\theta)p(\theta)}{p(\boldsymbol{y})} \propto p(\boldsymbol{y}|\theta)p(\theta), \tag{3.10}$$

where $p(\boldsymbol{y})$ denotes probability density function of observations. The MAP estimate of the parameter θ can be found as:

$$\hat{\theta}_{MAP} = \text{argmax}_{\theta} p(\theta|\boldsymbol{y}). \tag{3.11}$$

The MAP estimator enables us to use prior information that may be available upon the parameter θ. If the prior probability distribution $p(\theta)$ is set to a constant value, MAP estimation becomes equivalent to MLE [25] .

3.2. Sampling Techniques

In cases where the probability densities and/or their integrals used in classical estimation methods cannot be expressed analytically, numerical sampling methods can be used to approximate the probability density functions and hence obtain the estimate of desired parameters. Sampling a random variable from a desired probability density is denoted as:

$$x \sim p(x). \tag{3.12}$$

Having N samples, the target probability density function $p(x)$ can be approximated as [26]:

$$p(x) \approx \sum_{i=1}^{N} \left[\frac{p(x^i)}{\sum_{j=1}^{N} p(x^j)} \right] \delta(x - x^i), \tag{3.13}$$

where δ is the Dirac delta function and x^i denotes i^{th} sample $\{i = 1, 2, ...N\}$.

By definition, conditional expectation of function $f(x)$ of a random variable x is calculated using following integration:

$$I(f) = \int f(x)p(x|\boldsymbol{y})dx, \tag{3.14}$$

where $\boldsymbol{y}$ denotes the observations. If the integral in (3.14) is intractable, then by using (3.13), (3.14) can be approximated as:

$$E_{p(x|y)}[f(x)] = I(f) \approx \sum_{i=1}^{N} f(x^i) \left[\frac{p(x^i|\boldsymbol{y})}{\sum_{j=1}^{N} p(x^j|\boldsymbol{y})} \right]. \tag{3.15}$$

In order to obtain an approximation to the desired probability density function, the following requirements must be satisfied:

- The target probability density function must be suitable for random variables to

be sampled from.

- These random variables must be able to be evaluated in the functional form of the target probability density function.

Since probability density function of a random variable completely describes its probabilistic features, sampling methods have the advantage of estimating of probability density function of the desired parameter over classical estimation methods. Therefore, once the probability density function is estimated, any estimation method can be used to obtain an estimate of the desired parameter.

There are various types of sampling techniques, named as: uniform sampling, rejection sampling, importance sampling, sequential importance sampling, Markov Chain Monte Carlo (MCMC) method, Metropolis-Hastings sampling, Gibbs sampling and slice sampling. In the next sections importance sampling and sequential importance sampling methods (particle filters) are introduced, since this thesis focuses on particle filtering scheme. Details of other sampling methods can be found in [27].

3.2.1. Importance Sampling

Importance Sampling is a method of estimating a probability distribution where a different distribution other than target distribution is used to draw samples from, which is referred to as *importance density* and denoted as $q(x)$. This sampling technique is useful when it is not easy to draw samples form the target density and/or it is not possible to express the target density analitically since it enables to draw samples form the importance density which is easier to be sampled from. Using the importance density, the expectation (3.15) can be expressed as follows [27]:

$$I(f) \approx \frac{\sum_{i=1}^{N} f(x^i) \left[\frac{p(x^i|\boldsymbol{y})}{q(x^i|\boldsymbol{y})} q(x^i|\boldsymbol{y}) \right]}{\sum_{i=1}^{N} \left[\frac{p(x^i|\boldsymbol{y})}{q(x^i|\boldsymbol{y})} q(x^i|\boldsymbol{y}) \right]} = \frac{E_{q(x)} \left[f(x) \frac{p(x)}{q(x)} \right]}{E_{q(x)} \left[\frac{p(x)}{q(x)} \right]}. \tag{3.16}$$

The right hand side of (3.16) can be written as:

$$I(f) \approx \frac{E_{q(x)}\left[f(x)\frac{p(x)}{q(x)}\right]}{E_{q(x)}\left[\frac{p(x)}{q(x)}\right]} = \frac{\sum_{i=1}^{N} f(x^i)w^i}{\sum_{j=1}^{N} w^j}, \tag{3.17}$$

where the samples x^i are drawn from the importance density for $i = 1, 2, ...N$ and the importance weights are defined as:

$$w(x) \triangleq \frac{p(x)}{q(x)}. \tag{3.18}$$

3.2.2. Sequential Importance Sampling

In some applications where data arrives sequentially or its statistical properties change over time, processing data as a batch may result in obtaining incorrect statistics or missing out important changes in the observed phenomenon. Such a dynamic system can be given by the following state space equations [28]:

$$\begin{aligned} x_t &= f_t(x_{t-1}, v_t), \\ y_t &= h_t(x_t, n_t), \end{aligned} \tag{3.19}$$

where x_t and y_t represent the hidden state and observation at time instant t, respectively. Process and observation functions are denoted by f_t and h_t, respectively. Process noise is represented by v_t and observation noise is represented by n_t.

Here, the aim is to estimate the posterior density of the states $p(x|y)$ sequentially upon receiving observations. Sequential estimation of the posterior distribution of the hidden variable is performed in two stages, namely *prediction* and *update*. In the prediction stage, the current value of the hidden variable is predicted using previous

observations and shown by the following equation referred to as Chapman-Kolmogorov equation [26]:

$$p(x_t|y_{1:t-1}) = \int p(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})dx_{t-1}, \tag{3.20}$$

where $y_{1:t-1}$ denotes previous observations, $p(x_t|x_{t-1})$ denotes state transition density and $p(x_{t-1}|y_{1:t-1})$ denotes posterior density of the hidden variable at time $t-1$. In (3.20), $p(x_t|x_{t-1})$ is used instead of $p(x_t|x_{t-1}, y_{1:t-1})$ as (3.19) describes a Markov process of order one. When new data y_t is received at time t, the prior is updated by the measurement using Bayes' rule as:

$$\begin{aligned} p(x_t|y_{1:t}) &= p(x_t|, y_t, y_{1:t-1}) \\ &= \frac{p(y_t|x_t, y_{1:t-1})p(x_t|y_{1:t-1})}{p(y_t|y_{1:t-1})} \\ &= \frac{p(y_t|x_t)p(x_t|y_{1:t-1})}{p(y_t|y_{1:t-1})}, \end{aligned} \tag{3.21}$$

where the normalizing constant is given by the following equation:

$$p(y_t|y_{1:t-1}) = \int p(y_t|x_t)p(x_t|y_{1:t-1})dx_t. \tag{3.22}$$

In the update stage, the measurement y_t is used to modify the prior density to obtain the posterior density of the hidden state. This recursive propagation of the posterior density cannot be determined analytically in general, hence can be approximated using importance sampling methods.

In order to compute equations (3.20) and (3.21) in a sequential manner, importance sampling methods are modified to a new method which is known as *sequential importance sampling* (SIS). In the SIS, the posteriori distribution of the hidden states are approximated by N particles and corresponding normalized *importance weights* as

follows:

$$p(x_{0:t}|y_{1:t}) \approx \sum_{i=1}^{N} \tilde{w}_t^i \delta(x_{0:t} - x_{0:t}^i), \tag{3.23}$$

where $x_{0:t}^i$ represent particles $\{i = 1, 2, ..., N\}$and $\tilde{w}_t^i$ represent importance weights which are normalized as follows:

$$\tilde{w}_t^i = \frac{w(x_{0:t}^i)}{\sum_{j=1}^{N} w(x_{0:t}^j)}, \tag{3.24}$$

where $w(x_{0:t}^i)$ denotes the unnormalized importance weight corresponding to the i^{th} particle, which is defined in importance sampling as follows:

$$w(x_t^i) = \frac{p(x_{0:t}^i|y_{1:t})}{q(x_{0:t}^i|y_{1:t})}. \tag{3.25}$$

In order to update the importance weights sequentially, importance density is needed to be expressed in a sequential form. Suppose that at time t, approximation of $p(x_{0:t-1}|y_{1:t-1})$ is available and $p(x_{0:t}|y_{1:t})$ is aimed to be approximated by using a new set of particles. Then, the importance density is chosen to be in the following form:

$$q(x_{0:t}|y_{1:t}) = q(x_t|x_{0:t}, y_{1:t}) q(x_{0:t-1}|y_{1:t-1}), \tag{3.26}$$

then the new samples $x_{0:t}^i$ can be drawn from $q(x_{0:t}|y_{1:t})$ by *augmenting* the existing samples and the new state. The importance weights are updated sequentially as shown below [28]:

$$w_t^i = w_{t-1}^i \frac{p(y_t|x_t^i) p(x_t^i|x_{t-1}^i)}{q(x_t^i|x_{0:t-1}^i, y_{1:t})}. \tag{3.27}$$

In general, all of the previous states can be stored, however if only an estimate

of $p(x_t|y_{1:t})$ is needed at each time state, then (3.27) can be written as

$$w_t^i = w_{t-1}^i \frac{p(y_t|x_t^i)p(x_t^i|x_{t-1}^i)}{q(x_t^i|x_{t-1}^i, y_{1:t})} \tag{3.28}$$

and the posteriori density of the states can be approximated as (3.23).

The SIS algorithm enables us to estimate posterior densities of hidden variables in a dynamic system without making restictive assumptions such as linearity, Gaussianity or stationarity. However, one common problem, named as *degeneracy phenomenon*, of the SIS algorithm is that after few iterations importance weights of a large number of particles converge to zero. In [26], this problem is explained to have arisen from the selection of the importance function as in (3.26) which causes the variance of the importance weights increase in time. Degeneracy phenomenon results in having unnecessary computational burden while updating particles which have weights nearly zero and hence negligible contribution to the approximation of $p(x_t|y_{1:t})$. In [29], the following choice of importance function is shown to minimize the variance of the importance weights and limit degeneracy:

$$q(x_t|x_{0:t}^{(i)}, y_{1:t}) = p(x_t|x_{0:t}^{(i)}, y_{1:t}), \tag{3.29}$$

which is named as *optimal importance function*. However, in many scenarios sampling from the optimal importance function is not possible, but it is aimed to be approximated [30].

Degeneracy is measured using *effective sample size* N_{eff} as shown below [28]:

$$N_{eff} = \frac{N}{1 + Var(\tilde{w}_t^i)}, \tag{3.30}$$

where $\tilde{w_t^i}$ represents the unnormalized importance weight and N is the sample size. Since (3.30) cannot be calculated analytically, an approximate to N_{eff} is formed as

shown below [26]:

$$\hat{N}_{eff} \approx \frac{1}{\sum_{i=1}^{N}(w_t^i)^2}, \tag{3.31}$$

where w_t^i represents the normalized importance weight.

In order to reduce the effect of the degeneracy problem, *resampling* methods are introduced to eliminate the particles that have small importance weights and produce new particles from the existing particles which have higher importance weights . When the value of $\hat{N}_{eff}$ becomes smaller than a threshold value, resampling is performed. The main idea of resampling is to generate a new set of particles from the existing particles by using corresponding importance weights. For $i = \{1, 2, ...N\}$, a new index value, j, for each i is found by sampling from the discrete distribution that is the cumulative mass function of the importance weights. Then the particles corresponding to new index value j are given place in the new set of particles. In this manner, the particles with high importance weights survive in the resampling step and the particles with small importance weights tend to be given no index and eliminated. At the end of resampling step, all of the importance weights take equal value, i.e. $1/N$. This resampling scheme is generally referred to as *systematic resampling*.

There are other resampling techniques which can be found in [31], however systematic resampling is preferred in our work since it is simple to implement and has low computational complexity.

4. PARTICLE FILTER METHODS FOR DISTRIBUTED DETECTION AND DECISION FUSION

In this chapter, we will investigate particle-based methods in distributed detection problem. The common approach in distributed detection system design includes strong assumptions such as Gaussianity of the observation noise, having knowledge about the observation noise parameters, linearity and stationarity of the observed phenomenon or uncorrelatedness of the local decisions. However, when these assumptions do not hold exactly, the detectors become sub-optimal or they can even break down. In order to avoid unrealistic assumptions, we aim to model the observed phenomenon dynamically, infer statistical information using this dynamic model and using this information, determine the decision rules for the distributed detection system.

In [2], a distributed detection scheme which operates dynamically based on particles is suggested, in which observation noise is assumed to be known and Gaussian. However, it is shown that in some applications observed data may include heavy-tailed (impulsive) noise [10], which harms Gaussianity assumption of observation noise. Motivated by Gençağa et. al, we model observed data as an AR process, in order to form a dynamic model of the observed data which may include impulsive noise. Impulsive noise is generally modeled as alpha stable distribution, ϵ-contaminated mixture model or Gaussian mixture model [5], [7]. Gençağa et. al models impulsive noise as alpha-stable and estimates the parameters representing the noise statistics using a Bayesian approach [32]. However, alpha stable distribution cannot be expressed in a closed form and it requires numerical calculations which may be additional burden to the processors of the local detectors. In this thesis, observation noise is modeled as Gaussian mixture which may express multi-modalities or heavy-tails depending on the choice of the parameters and also it is a generalization of ϵ-contaminated mixture model. Therefore, we avoid Gaussianity assumption of observation noise and aim to infer its parameters from observed data in a sequential way using particle filtering.

In this study, the parameters of the observation noise as well as the coefficients of the AR process are not known and aimed to be estimated using particle filters. Therefore, the unknown parameter set is a vector which consists of the observation noise parameters and the AR coefficients. As the noise parameters are estimated, local decision rules are designed using a particle based framework, namely PSO, with the knowledge of noise parameters. Estimates of the AR coefficients are not used in the remaining part of distributed detection system, however this step is necessary to introduce a method that is applicable to the most general version of our problem, i.e., when both the AR coefficients and the noise parameters are unknown. Moreover, AR coefficients and their changes may be utilized in a future work of an early warning system for the detection of earthquakes.

Using estimated parameters of Gaussian mixture noise, thresholds of the local detectors are found within PSO framework where particles represent sensor thresholds. As observations are received, local detectors make decisions according to the threshold values and transmit them to the fusion center. Then, we consider the design of the fusion rule without assuming independence of the decisions of the local sensors. Here, the statistical dependency between the sensor decisions is formed by relating the marginal densities of the sensor observations using copulas. Also, we do not assume that the copula parameters are known; rather we aim to estimate the parameter of the used copula by using PSO.

4.1. AR Modeling of Observations and Estimation of Noise Parameters

In order to estimate observation noise parameters sequentially, a dynamic model for the observed data is required. Once a dynamic model is formed, particle filtering enables us to estimate the desired parameters. Due to the abscence of a general physically motivated model, AR process in Gaussian mixture noise is used to model the observed data. Then, our model is aimed to be applicable to the most general cases of observation noise.

Sensor observation at time t is modeled as a k^{th} order AR process as follows:

$$y_t = \boldsymbol{y}_{t-1}^{\tau} \boldsymbol{a}_t + n_t, \tag{4.1}$$

where the observation vector is $\boldsymbol{y}_{t-1} = [y_{t-1}, y_{t-2}, ..., y_{t-k}]$ and AR coefficients are represented by $\boldsymbol{a}_t = [a_t, a_{t-1}, ..., a_{t-k}]$. Observation noise n_t is described as a Gaussian mixture with J components as shown below:

$$n_t = \sum_{j=1}^{J} \pi_j \mathcal{N}(\mu_j, \sigma_j^2), \tag{4.2}$$

where π_j, μ_j and σ_j^2 represent weight, mean and variance of the j^{th} component of Gaussian mixture, respectively. Since n_t is a probability density function, summation of π_j's must be equal to 1, $j = \{1, 2, ..., J\}$. When there are two components in the Gaussian mixture, it is possible to estimate one of the π_j's and find the other one as $(1 - \pi_j)$. The noise parameters are placed in vector $\boldsymbol{\theta}_t = [\boldsymbol{\pi}_t, \boldsymbol{\mu}_t, \boldsymbol{\sigma}_t^2]$. Here, the aim is to estimate the noise parameters and the AR coefficients jointly. Hence, the desired parameters are placed in vector $\boldsymbol{x}_t = [\boldsymbol{a}_t, \boldsymbol{\theta}_t]$ which represents the state variables of the process equation as in (3.19). Since we have no prior information regarding the transition of the state variables, a random walk model is used to form a process equation as follows:

$$\boldsymbol{x}_t = \boldsymbol{x}_{t-1} + \boldsymbol{\nu}_{t-1}, \tag{4.3}$$

where $\boldsymbol{\nu}_{t-1}$ denotes the process noise. Process noise is modeled as Gaussian, i.e. $\boldsymbol{\nu}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_\nu)$ where the covariance matrix is diagonal as follows:

$$\boldsymbol{\Sigma}_\nu = diag(\sigma_{a(1)}^2, \sigma_{a(2)}^2, ..., \sigma_{\theta(1)}^2, \sigma_{\theta(2)}^2, ..., \sigma_{\theta(J)}^2). \tag{4.4}$$

Here, particle filtering enables us to sequentially estimate the elements of the hidden state vector $\boldsymbol{x}_t$ [28]. Expressing the transition process as (4.3) and the observation

process as (4.1), state space equations that form particle filtering scheme as in (3.19) are defined. Importance function is chosen as the transition distribution $p(x_t|x_{t-1})$, since *a priori* information about the states is not sufficient to construct a good approximation of the optimal importance function. Hence, particles are sampled from the importance density as follows:

$$x_t^{(i)} \sim \mathcal{N}(x_{t-1}^{(i)}, \mathbf{\Sigma}_\nu), \tag{4.5}$$

where $x_t^{(i)}$ denotes the drawn samples at time t. Substituting the transition density as importance density in (3.28), importance weights are updated according to the following:

$$w_t^i = w_{t-1}^i p(y_t|x_t^i), \tag{4.6}$$

where $p(y_t|x_t^i)$ is *likelihood* density which is expressed as follows when n_t is mixture of Gaussians with J components:

$$p(y_t|x_t^i) = \sum_{j=1}^{J} \pi_{t,j}^i \mathcal{N}(y_t - \boldsymbol{y}_{t-1}^\tau \boldsymbol{a}_t^i - \mu_{t,j}^i, (\sigma_{t,j}^2)^i). \tag{4.7}$$

After updating the importance weights, these weights are normalized according to (3.29). Then systematic resampling is performed. A pseudo-code of this method is given in Table 4.1.

Table 4.1. Pseudo-code for the AR modeling of observations and estimation of the unknown parameters and coefficients.

Input: Sensor observations, initial distributions of the states (desired parameters), time steps.

Output: Estimated states.

Initiation:

1. Draw particles $\boldsymbol{x}_t^i = [\boldsymbol{a}_t, \boldsymbol{\pi}_t, \boldsymbol{\mu}_t, \boldsymbol{\sigma}_t^2]$ $\{i = 1, 2, ..N\}$ from the initial distributions of the states:

$$\boldsymbol{a}_0^i \sim \mathcal{U}(-1, 1),\ \boldsymbol{\pi}_0^i \sim \mathcal{U}(0, 1)$$
$$\boldsymbol{\mu}_0^i \sim \mathcal{N}(\boldsymbol{m}_\mu, \boldsymbol{\Sigma}_j),\ (\boldsymbol{\sigma}_0^2)^i \sim \mathcal{IW}(\boldsymbol{\psi}_\sigma, \boldsymbol{\Lambda}_\sigma)$$

where m and Σ represent mean and covariance matrices of Gaussian distributions; ψ and Λ represent scale matrix and degrees of freedom of Inverse-Wishart distributions.

2. Set initial importance weights to equal value, i.e. $1/N$.

For t=1:T

3. Draw new particles $\boldsymbol{x}_t^i$ $\{i = 1...N\}$ from transition density using process equation in (4.3):

$$\boldsymbol{x}_t^{(i)} \sim \mathcal{N}(\boldsymbol{x}_{t-1}^{(i)}, \boldsymbol{\Sigma}_\nu)$$

4. Calculate the importance weights by substituting (4.7) in (4.6):

$$w_t^i = w_{t-1}^i \sum_{j=1}^{J} \pi_{t,j}^i \mathcal{N}(y_t - \boldsymbol{y}_{t-1}^\tau \boldsymbol{a}_t^i - \mu_{t,j}^i, (\sigma_{t,j}^2)^i)$$

5. Normalize the importance weights:

$$\tilde{w}_t^i = \frac{w_t^i}{\sum_{j=1}^{N} w_t^j}$$

6. Resample the particles and go to step 3.

end

4.1.1. Experiments with First Order AR Simulated Data

A 1^{st} order AR process in additive Gaussian mixture noise, with two components, which can be expressed in the following form:

$$y_t = y_{t-1}a_t + \sum_{j=1}^{2} \pi_j \mathcal{N}(\mu_j, \sigma_j^2) \tag{4.8}$$

is simulated. For parameter estimates, average of 20 Monte Carlo simulations are used. The parameter estimation results are evaluated using the Normalized Mean Square Errors (NMSE), which can be given as follows:

$$NMSE_t = \frac{\sum_{l=1}^{20}(\hat{\theta}_{t,l} - \theta)^2}{\sum_{l=1}^{20}\sum_{t=1}^{T}\theta_t^2}, \tag{4.9}$$

where T and θ denote data length and the desired parameter, respectively.

4.1.1.1. Constant AR and noise parameters. Parameters of the Gaussian mixture are $\boldsymbol{\pi}$=[0.2, 0.8], $\boldsymbol{\mu}$=[2, 0.5], $\boldsymbol{\sigma}^2$ = [1, 2]. AR coefficient is taken to be 0.8. In the experiment, 500 particles are used and data lenght is 1000. Process noise is constant, and chosen as $\boldsymbol{\Sigma}_\nu = diag(10^{-6}, 10^{-6}, 10^{-6}, 10^{-6}, 10^{-6}, 10^{-6})$. Initially, the values of the particles of means and variances are drawn from $\mu_0 \sim \mathcal{N}(1, 1)$ and $\sigma_0^2 \sim \mathcal{IW}(5, 10)$, respectively.

In Figure 4.1, estimates of the noise parameters and AR coefficient are shown. Performance of the method is examined by calculating NMSE of the estimations which is given by Figure 4.2. It can be observed from Figure 4.1 and 4.2 that the estimated values of the parameters converge to their true values, yet there are also fluctuations in the estimations due to the process noise. The reason for that is the choice of the elements of the covariance matrix in (4.4). For this selection of the variance of the process noise the parameter value converges to its true value after an acceptable number of samples, however it may also result in observable jitters.

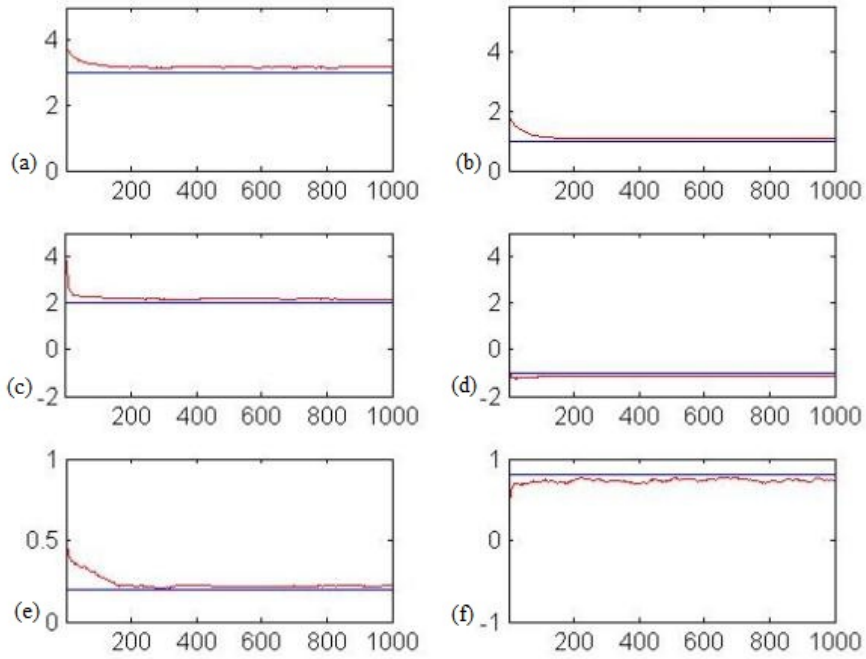

Figure 4.1. Estimation of constant parameters a) $\sigma^2(1) = 3$, b) $\sigma^2(2) = 1$, c) $\mu(1) = 2$, d) $\mu(2) = -1$, e) $\pi(1) = 0.2$, f) $a = 0.8$.

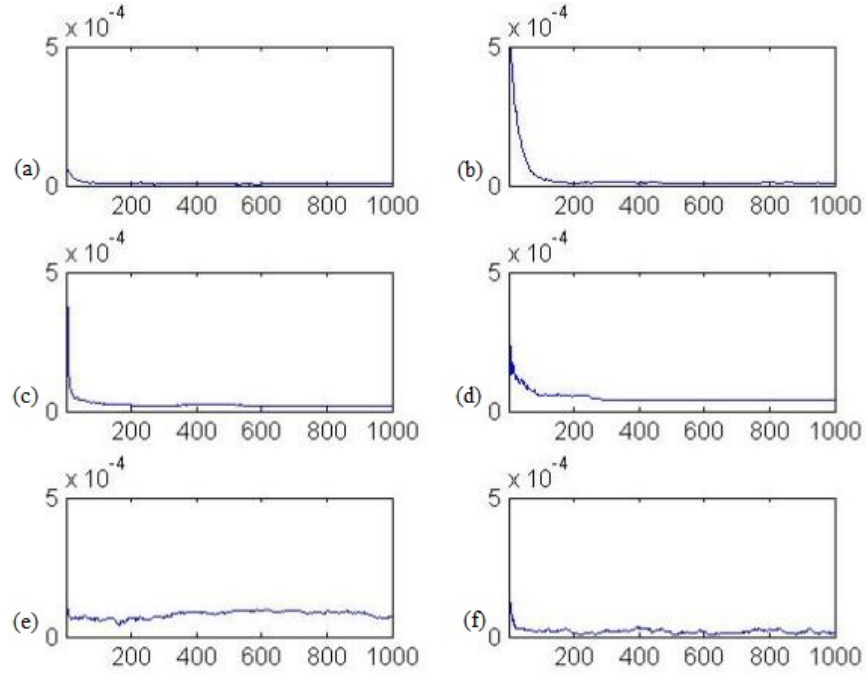

Figure 4.2. NMSE of estimation of constant parameters a) $\sigma^2(1)$, b) $\sigma^2(2)$, c) $\mu(1)$, d) $\mu(2)$, e) $\pi(1)$, f) a.

4.1.1.2. Time-varying AR and constant noise parameters. Parameters of the Gaussian mixture are $\boldsymbol{\pi}$=[0.2, 0.8], $\boldsymbol{\mu}$=[2, 0.5], $\boldsymbol{\sigma}^2$ = [1, 2]. AR coefficient is taken to be 0.8 until the 1000 th sample, where it changes abruptly to 0.3. In the experiment, 500 particles are used and data lenght is 2000. Process noise is constant, and chosen as $\boldsymbol{\Sigma}_\nu = diag(10^{-6}, 10^{-6}, 10^{-6}, 10^{-6}, 10^{-6}, 10^{-6})$. Initially, the values of the particles of means and variances are drawn from $\mu_0 \sim \mathcal{N}(1, 1)$ and $\sigma_0^2 \sim \mathcal{IW}(5, 10)$, respectively.

In Figure 4.3, estimates of the noise parameters and AR coefficient are shown. Performance of the method is examined by calculating NMSE of the estimations which is given by Figure 4.4.

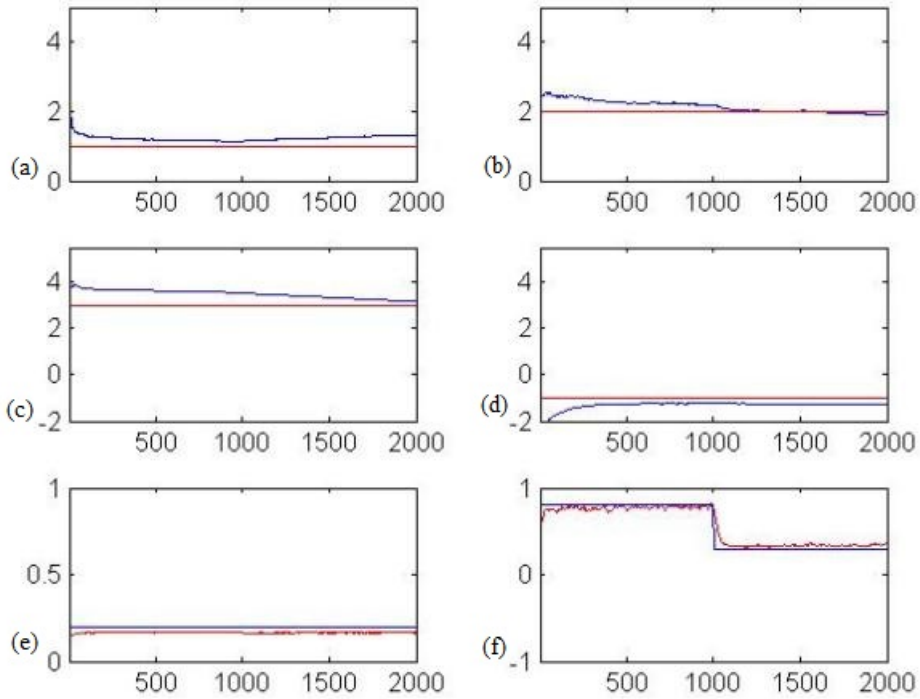

Figure 4.3. Estimation of parameters a)$\sigma^2(1) = 3$, b) $\sigma^2(2) = 1$, c) $\mu(1) = 2$, d) $\mu(2) = -1$, e) $\pi(1) = 0.2$, f) $a = 0.8$ until 1000^{th} sample, then $a = 0.3$.

As Figure 4.4 reveals, an abrupt change in the AR coefficient may result in observable deviations in the estimation of the noise parameters. As expected NMSE of the AR coefficient abruptly inceases at $t = 1000$, then it falls below its previous

values. NMSE of the variances also indicates a change in the model. The reason for that is placing all of the unknown parameters in the same vector which has only one weight. Comparing Figure 4.1 and 4.3, it can be said that the Gaussian mixture weight π converges quicker in the latter which may have occured due to having a closer initial starting point to true value of the coefficient.

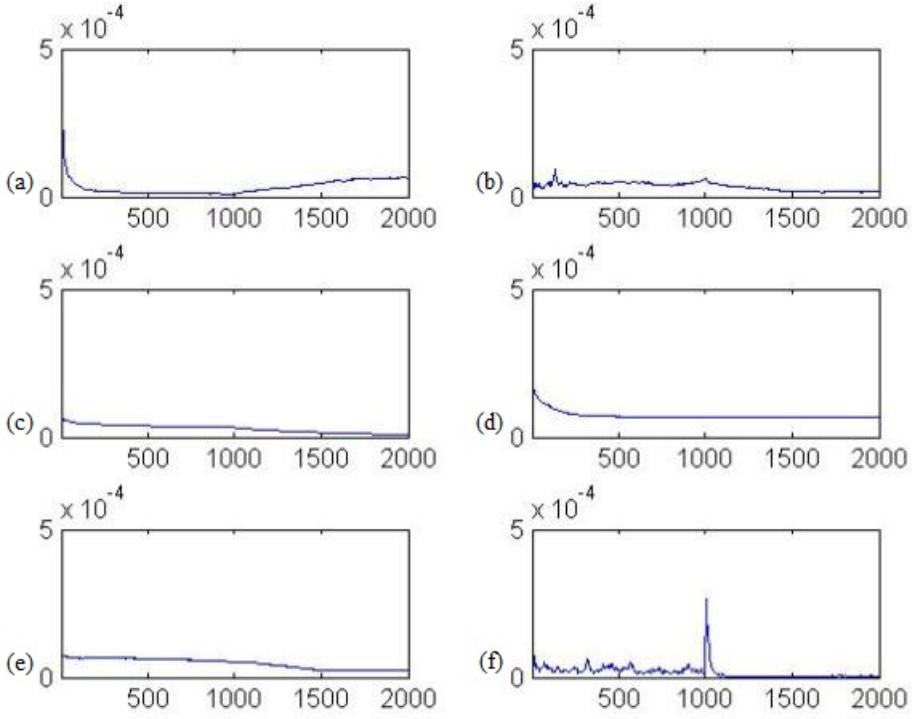

Figure 4.4. NMSE of estimation of parameters a) $\sigma^2(1)$, b) $\sigma^2(2)$, c) $\mu(1)$, d) $\mu(2)$, e) $\pi(1)$, f) a.

In a more realistic approach, changes in the AR coefficient may also be smoother. In this experiment, AR coefficient has a sinusoidal change. The values of the noise parameters are $\boldsymbol{\pi}$=[0.8, 0.2], $\boldsymbol{\mu}$=[2, 0.5], $\boldsymbol{\sigma}^2$ = [1, 2] and the AR coefficient is a sinusoidal given as in Figure 4.5. In order to AR process to be stable, maximum value of AR coefficient becomes 0.998. In this experiment, 500 particles are used and data length is 2000. Process noise is constant, and chosen as $\boldsymbol{\Sigma}_\nu = diag(10^{-6}, 10^{-6}, 10^{-6}, 10^{-6}, 10^{-6}, 10^{-6})$. Initially, the values of the particles of means and variances are drawn from $\mu_0 \sim \mathcal{N}(1, 1)$ and $\sigma_0^2 \sim \mathcal{IW}(5, 10)$, respectively.

In Figure 4.5, estimates of the noise parameters and AR coefficient are shown.

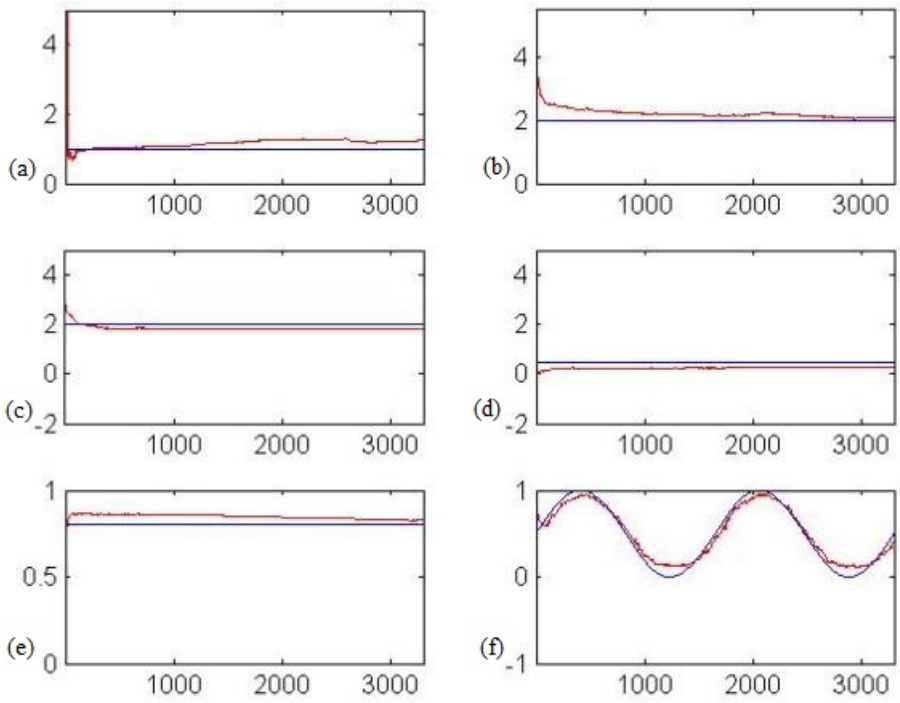

Figure 4.5. Estimation of parameters a) $\sigma^2(1) = 3$, b) $\sigma^2(2) = 1$, c) $\mu(1) = 2$, d) $\mu(2) = -1$, e) $\pi(1) = 0.8$, f) $a = 0.499(1 + \sin(t))$.

Performance of the method is examined by calculating NMSE of the estimations which is given by Figure 4.6. It can be observed from Figure 4.5 and 4.6 that even all of the parameters are treated as one particle, a smooth change of an element in the particle vector does not significantly degrade performance of the estimation of the parameters. Moreover, changing the AR coefficient as sinusoidal which reaches to zero enables us to conclude about the performance of the estimator in the abscence of a signal, i.e. when hypothesis H_0 is true. As expected, the AR coefficient becomes very small and the estimation performances of the noise parameters are generally preserved.

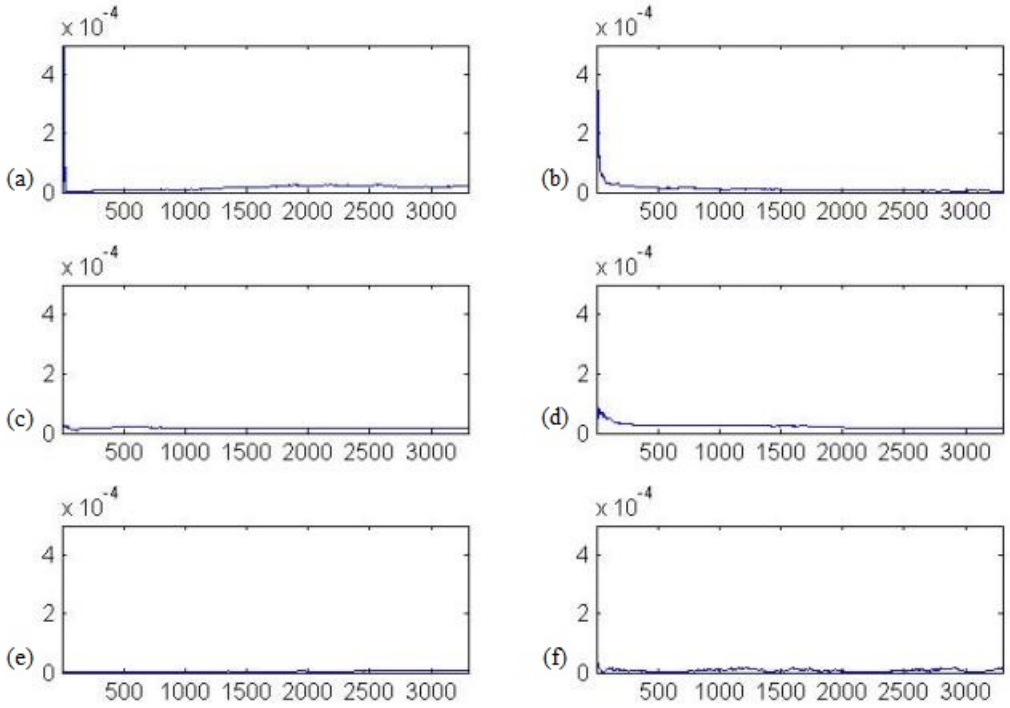

Figure 4.6. NMSE of estimation of parameters a) $\sigma^2(1)$, b) $\sigma^2(2)$, c) $\mu(1)$, d) $\mu(2)$, e) $\pi(1)$, f) a.

4.1.1.3. Constant AR and time varying noise parameters. The weights and the means of the Gaussian mixture are $\boldsymbol{\pi}$=[0.2, 0.8], $\boldsymbol{\mu}$=[2, 0.5] where the variances of the Gaussian mixture are taken to be $\boldsymbol{\sigma}^2 = [1, 2]$ until the 1000 th sample, where they change abruptly to $\boldsymbol{\sigma}^2 = [1, 10]$.

In the experiment, 500 particles are used and data length is 2000. Process noise is constant, and chosen as $\boldsymbol{\Sigma}_\nu = diag(10^{-6}, 10^{-6}, 10^{-6}, 10^{-6}, 10^{-6}, 10^{-6})$. Initially, the values of the particles of means and variances are drawn from $\mu_0 \sim \mathcal{N}(1, 1)$ and $\sigma_0^2 \sim \mathcal{IW}(5, 10)$, respectively.

In Figure 4.7, estimates of the noise parameters and AR coefficient are shown. Then, performance of the method is examined by calculating NMSE of the estimations which is given by Figure 4.8.

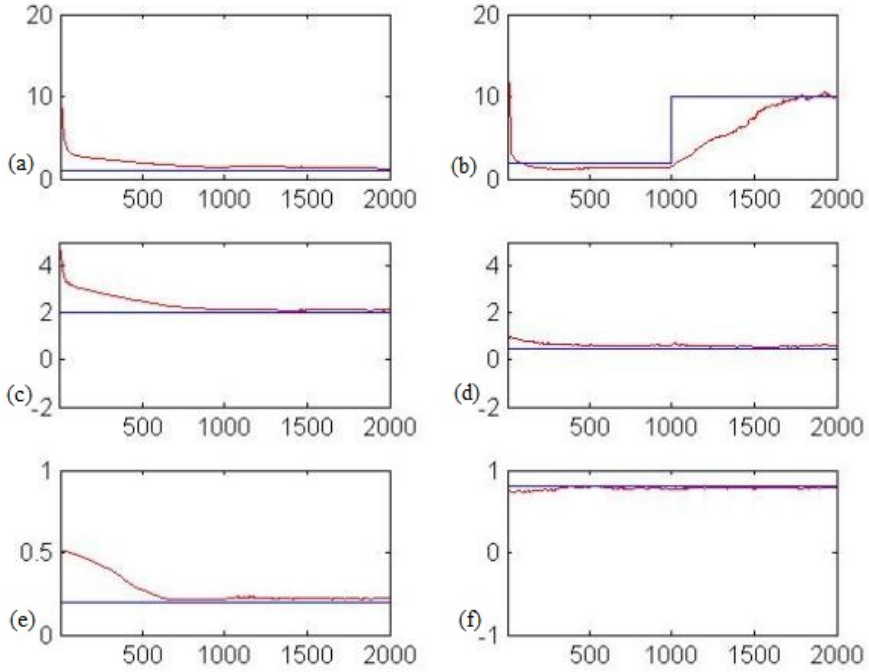

Figure 4.7. Estimation of parameters a) $\sigma^2(1) = 1$, b) $\sigma^2(2) = 2$ until 1000^{th} sample, then $\sigma^2(2) = 10$, c) $\mu(1) = 2$, d) $\mu(2) = -1$, e) $\pi(1) = 0.2$, f) $a = 0.8$.

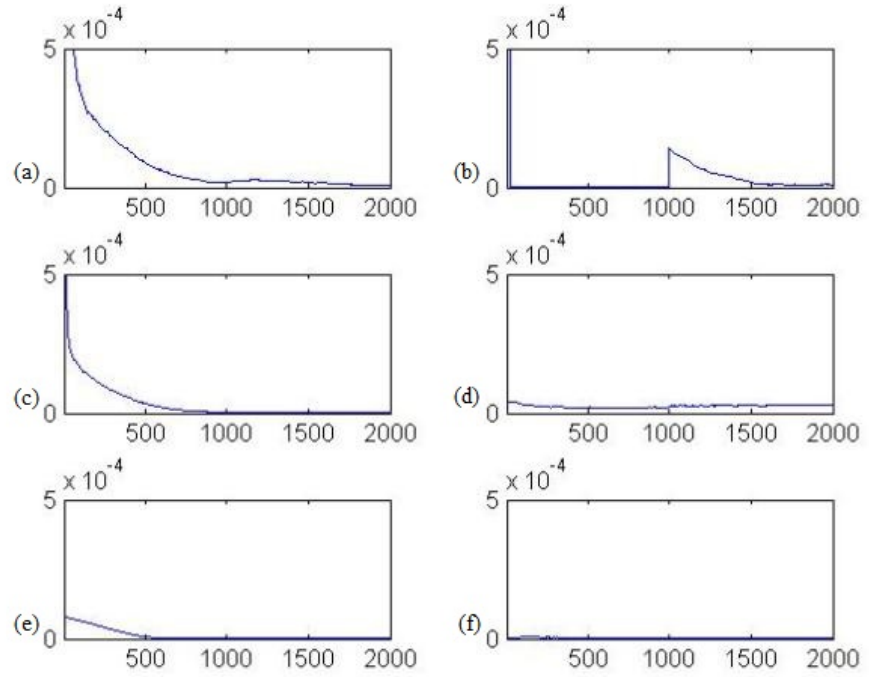

Figure 4.8. NMSE of estimation of parameters a) $\sigma^2(1)$, b) $\sigma^2(2)$, c) $\mu(1)$, d) $\mu(2)$, e) $\pi(1)$, f) a.

Figure 4.7 and Figure 4.8 demonstrate that changing one variance term of the Gaussian mixture does not significantly affect estimation performance of the other parameters in the model. Similar to as in Figure 4.1, the Gaussian mixture weight π starts from a further value than its true value which results in a slower convergence.

4.1.2. Experiments with Second Order AR Simulated Data

In some cases, higher order of AR processes may be necessary while modeling data. In order to test the algorithm with, a 2^{nd} order AR process in additive Gaussian mixture noise, with two components, which can be expressed in the following form:

$$y_t = y_{t-1}a_t(1) + y_{t-2}a_t(2) + \sum_{j=1}^{2} \pi_j \mathcal{N}(\mu_j, \sigma_j^2)$$

is simulated. For parameter estimates, average of 20 Monte Carlo simulations are used. The parameter estimation results are evaluated using NMSE.

4.1.2.1. Constant AR and noise parameters. Parameters of the Gaussian mixture are $\boldsymbol{\pi}$=[0.2, 0.8], $\boldsymbol{\mu}$=[1, 1], $\boldsymbol{\sigma}^2$ = [25, 1]. AR coefficients are 0.5 and -0.75. In the experiment, 500 particles are used and data lenght is 1000. Process noise is constant, and chosen as $\boldsymbol{\Sigma}_\nu = diag(10^{-6}, 10^{-6}, 10^{-6}, 10^{-6}, 10^{-6}, 10^{-6}, 10^{-6})$. Initially, the values of the particles of means and variances are drawn from $\mu_0 \sim \mathcal{N}(1, 1)$ and $\sigma_0^2 \sim \mathcal{IW}(5, 10)$, respectively.

In Figure 4.9, estimates of the noise parameters and AR coefficients are shown. Then, performance of the method is examined by calculating NMSE of the estimations which is given by Figure 4.10.

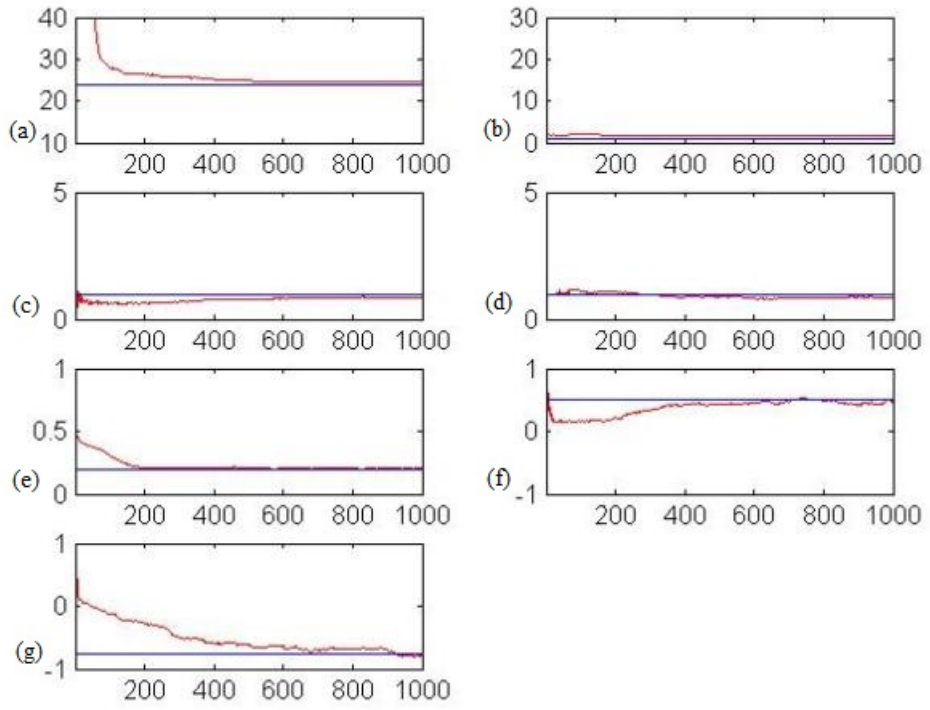

Figure 4.9. Estimation of parameters a) $\sigma^2(1) = 25$, b) $\sigma^2(2) = 1$, c) $\mu(1) = 1$, d) $\mu(2) = 1$, e) $\pi(1) = 0.2$, f) $a(1) = 0.5$, g) $a(2) = -0.75$.

Figure 4.9 and 4.10 show that second order modeling of data does not degrade the estimation performance of the algorithm. In this experiment, initial values of the AR coefficients $a(1)$ and $a(2)$ and the Gaussian mixture weight π are relatively further from their true values, therefore it takes more samples for their estimates to converge their true values, as expected. In addition to this, estimation of $\mu(2)$ shows that starting from the exact value parameter may still result in non zero NMSE values due to jittering from process noise.

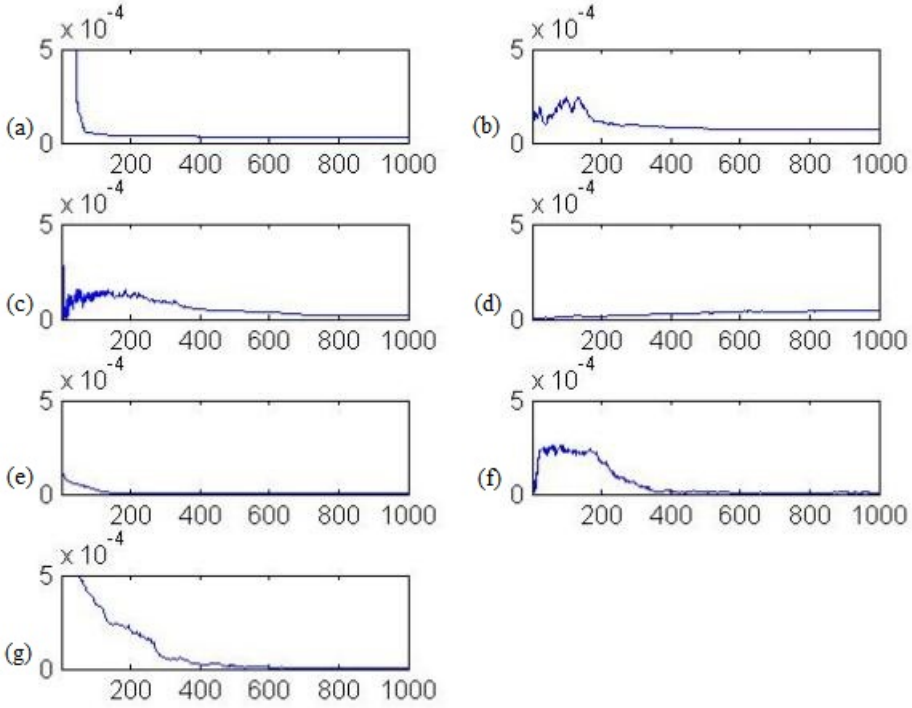

Figure 4.10. NMSE of estimation of parameters a) $\sigma^2(1)$, b) $\sigma^2(2)$, c) $\mu(1)$, d) $\mu(2)$, e) $\pi(1)$, f) $a(1)$, g) $a(2)$.

4.1.2.2. Time varying first AR coefficient and other parameters are constant. Parameters of the Gaussian mixture are $\boldsymbol{\pi}$=[0.2, 0.8], $\boldsymbol{\mu}$=[1, 1], $\boldsymbol{\sigma}^2$ = [25, 1]. First AR coefficient is 0.5 until 1100^{th} sample and it changes to 0.3. Second AR coefficient is -0.75. In the experiment, 500 particles are used and data lenght is 3300. Process noise is constant, and chosen as $\boldsymbol{\Sigma}_\nu = diag(10^{-6}, 10^{-6}, 10^{-6}, 10^{-6}, 10^{-6}, 10^{-6}, 10^{-6})$. Initially, the values of the particles of means and variances are drawn from $\mu_0 \sim \mathcal{N}(1, 1)$ and $\sigma_0^2 \sim \mathcal{IW}(5, 10)$, respectively.

In Figure 4.11, estimates of the noise parameters and AR coefficients are shown. Then, performance of the method is examined by calculating NMSE of the estimations which is given by Figure 4.12.

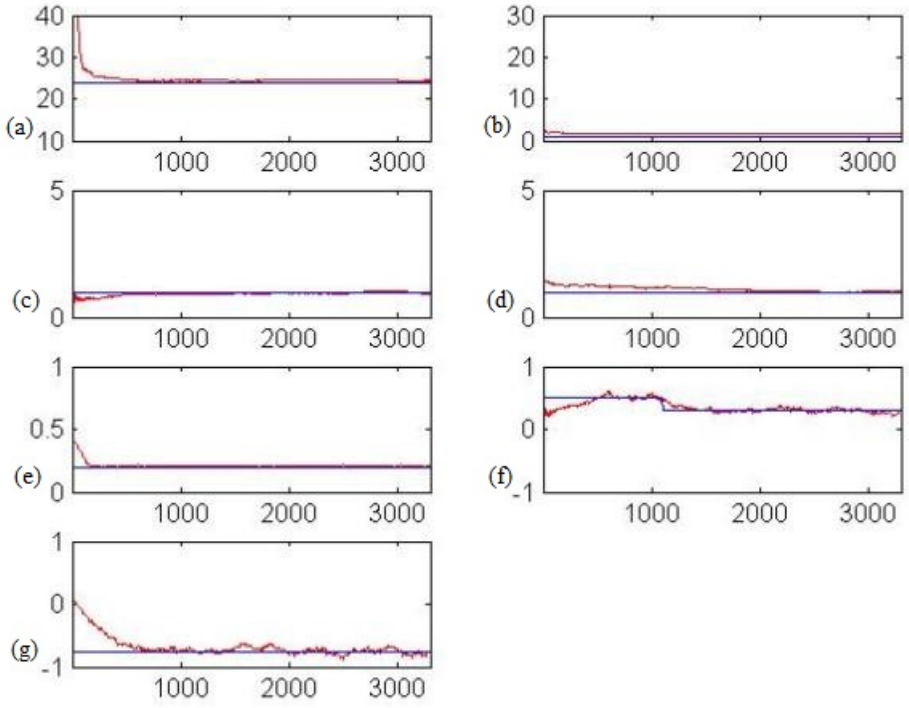

Figure 4.11. Estimation of parameters a) $\sigma^2(1) = 25$, b) $\sigma^2(2) = 1$, c) $\mu(1) = 1$, d) $\mu(2) = 1$, e) $\pi(1) = 0.2$, f) $a(1) = 0.5$ until 1100^{th} sample and it changes to 0.3, g) $a(2) = -0.75$.

Figure 4.11 and Figure 4.12 demonstrate that changing one AR coefficient does not significantly affect estimation performance of the other parameters in the model. Since the abrupt change in first AR coefficient is relatively small compared to its value, NMSE has smaller increase than it is in Figure 4.4. As in the previous experiment, second AR coefficient starts from a further value than its true value which results in a slower convergence.

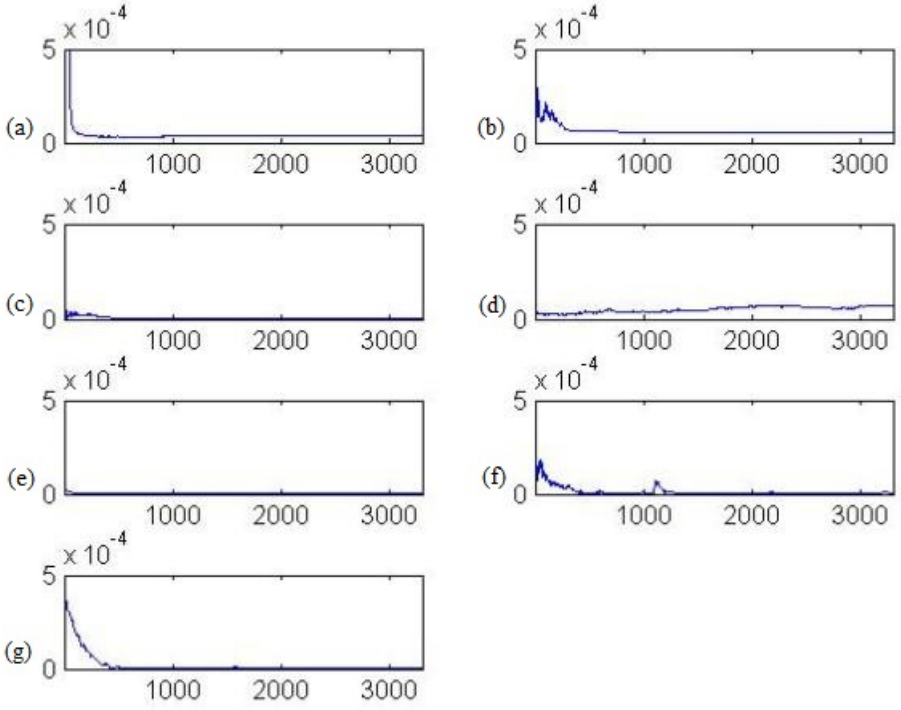

Figure 4.12. NMSE of estimation of parameters a) $\sigma^2(1)$, b) $\sigma^2(2)$, c) $\mu(1)$, d) $\mu(2)$, e) $\pi(1)$, f) $a(1)$, g) $a(2)$.

4.2. Local Detector Thresholds

In this section, local decision rules of the distributed detection system are designed. Parallel configuration with M sensors as in Figure 2.3 is considered to be the distributed detection scheme due to its ease of ease of implementation and wide usage [4]. In this scheme, the sensors do not communicate with each other and the fusion center. Here, we assume that binary hypothesis testing problem is examined by the local detectors. Observed data at each sensor can be expressed as follows:

$$\begin{cases} y_i = n_i & when\ H_0\ is\ true \\ y_i = s + n_i & otherwise \end{cases}, \tag{4.10}$$

where s, y_i and n_i represent the common phenomenon (signal), observed data and observation noise at i^{th} sensor, respectively. Instead of making assumptions about

the observation noise, decision rules of the detectors are designed using the statistical knowledge gathered in previous section. Since the observation noise is modeled as mixture of Gaussians; estimating weights, means and variances of the mixtures enables us to obtain sensor threshold as a weighted sum of thresholds corresponding to Gaussian mixtures.

Let y_m denote received data by the m^{th} sensor, then the m^{th} sensor employs a decision rule to make its decision, u_m. This procedure requires determining a threshold value λ_m for each of the sensors. Motivated by [2], sensor thresholds are determined by minimizing a probability of error which can be defined using P_M and P_{FA} of each sensor as a function of the sensor thresholds $\boldsymbol{\lambda} = [\lambda, \lambda_2, ..., \lambda_M]$ as follows:

$$Pe(\boldsymbol{\lambda}) = Pr(H_0)P_{FA} + Pr(H_1)P_M, \tag{4.11}$$

where $Pr(H_1)$ and $Pr(H_0)$ denote the prior probability on H_1 and H_0, respectively. Dividing (4.9) by $Pr(H_1)$ and using the fact that $P_{FA} = 1 - P_D$ results in that minimizing the probability of error $P_e(\boldsymbol{\lambda})$ is equivalent to maximizing $\widetilde{P}_e(\boldsymbol{\lambda})$ as:

$$\widetilde{P}_e(\boldsymbol{\lambda}) = P_D - \frac{Pr(H_0)}{Pr(H_1)}P_{FA}. \tag{4.12}$$

Probability of detection and probability of false alarm can be described for the m^{th} sensor as functions of the sensor threshold λ_m, respectively, as follows:

$$P_{D,m} = \int_{\lambda_m}^{\infty} p(y_m|H_1)dy_m \tag{4.13}$$

and

$$P_{FA,m} = \int_{\lambda_m}^{\infty} p(y_m|H_0)dy_m. \tag{4.14}$$

Finding $\boldsymbol{\lambda} = [\lambda, \lambda_2, ..., \lambda_M]$ values which yield to the maximum value $\widetilde{P}_e(\boldsymbol{\lambda})$ is a subject of multi-objective optimization. Since a global optimization is performed while de-

termining the threshold set $\boldsymbol{\lambda}$, it is assumed that the sensors' operations are coupled. Then, P_D and P_{FA} in (4.12) can be written as follows:

$$P_D = \sum_{u \in H_1} \prod_{m=1}^{M} \left[u_m P_{D,m} + (1 - u_m)(1 - P_{D,m}) \right] \tag{4.15}$$

and

$$P_{FA} = \sum_{u \in H_1} \prod_{m=1}^{M} \left[u_m P_{FA,m} + (1 - u_m)(1 - P_{FA,m}) \right]. \tag{4.16}$$

It can be seen from above that substituting (4.15) and (4.16) in (4.12) requires finding thresholds by considering a large search space, especially as the number of sensors increases. In order to obtain the threshold set that maximizes $\widetilde{P}_e(\boldsymbol{\lambda})$, a continuous space of all possible threshold values is searched using PSO algorithm.

Initially, each of the particles $\lambda^i_{t=0}$, $\{i = 1, 2, ..., N\}$ are sampled independently and uniformly from a region $[\lambda_{min}, \lambda_{max}]$, which includes all the values that each particle can take. Inital weights of the particles are determined as follows:

$$w^i_{t=0} = \widetilde{P}_e(\boldsymbol{\lambda}^i_{t=0}). \tag{4.17}$$

Then, these weights are normalized and particles are resampled according to their weights. Position of the particle which corresponds to the heighest weight is stored at each iteration. In the following steps of the algorithm, positions of the particles are found using previous particle positions ($\boldsymbol{\lambda}^i_{t-1}$) and previous best particle ($\boldsymbol{\lambda best}_{t-1}$) as follows:

$$\boldsymbol{\lambda}^i_t = \frac{\boldsymbol{\lambda}^i_{t-1} + \boldsymbol{\lambda best}_{t-1}}{2} + \boldsymbol{q}, \tag{4.18}$$

where $\boldsymbol{q}$ represents uniformly sampled noise for jittering particle trajectories. Particle

weights are updated as follows:

$$w_t^i = w_{t-1}^i \widetilde{P}_e(\boldsymbol{\lambda}_t^i). \tag{4.19}$$

After normalization of these weights, resampling of the particles are performed. Sequentially, the algorithm converges to the particles and corresponding weights which maximize $\widetilde{P}_e(\boldsymbol{\lambda})$, hence the set of thresholds $(\lambda_1, \lambda_2, ..., \lambda_M)$ which minimizes the probability of error $Pe(\boldsymbol{\lambda})$ can be obtained. A pseudo-code of this method is given in Table 4.2.

Table 4.2. Pseudo-code for using PSO in the determination of sensor thresholds.

Input: Estimated noise parameters for all of the sensors, time steps, maximum and minimum values of all possible values that thresholds may take.

Output: Local decisions of the sensors.

Initiation:

1. Draw particles $\boldsymbol{\lambda}_0^i = [\lambda_{1,0}^i, \lambda_{2,0}^i, ..., \lambda_{M,0}^i]$ $\{i = 1, 2, ..N\}$ uniformly over region $[\lambda_{min}, \lambda_{max}]$.
2. Calculate $\widetilde{P}_e(\boldsymbol{\lambda}_0^i)$ and set initial importance weights as follows:

$$w_0^i = \widetilde{P}_e(\boldsymbol{\lambda}_0^i)$$

3. Normalize the weights.
4. Resample the particles.

For t=1:T

5. Propogate particles $\boldsymbol{\lambda}_t^i$ $\{i = 1...N\}$ by adding uniformly drawn noise samples $\boldsymbol{q}$:

$$\boldsymbol{\lambda}_t^i = \frac{\boldsymbol{\lambda}_{t-1}^i + \boldsymbol{\lambda best}_{t-1}}{2} + \boldsymbol{q}$$

6. Update the particle weights:

$$w_t^i = w_{t-1}^i \widetilde{P}_e(\boldsymbol{\lambda}_t^i)$$

7. Normalize the weights.
8. Keep the best performing particle $\lambda best_t$.
9. Make local decisions $\boldsymbol{u}_t = u_{t,1}, u_{t,2}, ..., u_{t,m}, ..., u_{t,M}$ as follows:

$$u_{t,m} = \begin{cases} 0 & if\ \lambda best_{t,m} > y_{t,m} \\ 1 & otherwise \end{cases}$$

10. Resample the particles and go to step 5.

end

4.3. Decision Fusion

In this section, design of the fusion rule of the distributed detection system depicted in Figure 2.3 is considered. The local detectors transmit their decisions $\boldsymbol{u} = [u_1, u_2, ..., u_M]$ to the fusion center which combines local decisions according to an optimization criterion to make a global decision, u_0. The optimum fusion rule is given by (2.28) under the assumption that sensor decisions are independent. When the assumption of independence of the decisions does not hold, overall performance of the distributed detection system may degrade. Even for the cases where sensors make their decisions independently, their decisions may be still correlated especially when the sensor locations are close. In order to utilize the correlation of the decisions in the fusion center, the joint probability density of the decisions are required. An alternative method utilizes the Bahadur-Lazarfeld expansion to compute the joint probabilities [4]. However, when dependency between sensor decisions are nonlinear, this approach may not be efficient. Moreover, this approach requires prior information about the joint statistics of sensor observations or decisions.

Here, we use copula theory which is introduced to the problem of fusion of correlated decisions in [11]. Copula theory provides an appoach that does not necessarily require prior information about the joint statistics of the sensor decisions. This approach enables us to determine the joint probability density function of the sensor decisions using only the knowledge of the marginal densities of the sensor observations or the marginal probability mass functions of the sensor decisions. When copula parameters are known, (3.33)-(3.36) can be used to obtain the joint probability density function of the sensor decisions. In cases when copula parameters are unknown, batch estimation methods are used to estimate the parameters under each hypothesis, then the unknown copula parameters are replaced by their respective estimate. However estimating copula parameters for each hypothesis from batch of decisions may perform poorly in cases where the copula parameters change over time as the true hypothesis change. Therefore we suggest a sequential estimation of copula parameters by using a particle-based method.

A sequential Monte Carlo method, namely Gibbs sampling, is introduced to the estimation of the copula parameters [13]. In this approach, dependency between data sets is modeled by introducing time-variation into the densities by writing them as factor models. Since copula relates the marginal distribution functions of local decisions to the joint probability distribution function, a dynamic model to describe this relation could not be formed using state space equations. Thus, particle filter can not be used for this purpose and PSO method is utilized to obtain the copula parameters which yields the maximum value of a cost function. Motivated by MLE, the parametric likelihood function of the unknown copula parameters is chosen as the cost function to be maximized. In this work, we specifically deal with Gaussian copula, since it is a widely used copula type in signal processing and it requires a single parameter to be estimated [23].

Suppose that ρ_{kl} specify the correlation between the random variables u_k and u_l $\{k, l = 1, 2, ..., M\}$. The Gaussian copula c_g embeds correlation using the correlation matrix Σ as follows:

$$c_g(u_k, u_l) = \frac{1}{|\Sigma|^{1/2}} \exp\left[\frac{[\phi^{-1}(u_k)\ \phi^{-1}(u_l)]^{\tau}(\Sigma^{-1} - I)[\phi^{-1}(u_k)\ \phi^{-1}(u_l)]}{2}\right], \tag{4.20}$$

where I is the identity matrix and Σ is defined as:

$$\Sigma = \begin{cases} 1 & if\ k = l \\ \rho_{kl} & otherwise \end{cases} \tag{4.21}$$

and ϕ denotes the univariate Gaussian distribution function [11]. For simplicity, we assume that the number of sensors is two and the correlation parameter under H_0 is either known or negligible. Then, the parametric likelihood function of ρ under H_1 can be expressed using (2.33) as follows:

$$p(u_1, u_2|\rho) = P_{00}(\rho)^{(1-u_1)(1-u_2)} P_{01}(\rho)^{(1-u_1)u_2} P_{10}(\rho)^{u_1(1-u_2)} P_{11}(\rho)^{u_1 u_2}, \tag{4.22}$$

where $P_{00}(\rho)$, $P_{01}(\rho)$, $P_{10}(\rho)$ and $P_{11}(\rho)$ are defined by (2.35) and can be expressed using Gaussian copula in (4.20) for two-sensor case as follows:

$$\begin{aligned} P_{00} =& C(1-p_1, 1-p_2), \\ P_{01} =& 1-p_1-C(1-p_1, 1-p_2), \\ P_{10} =& 1-p_2-C(1-p_1, 1-p_2), \\ P_{11} =& p_1+p_2+C(1-p_1, 1-p_2)-1, \end{aligned} \tag{4.23}$$

where $C(1-p_1, 1-p_2)$ and Σ are expressed as follows:

$$C(1-p_1, 1-p_2) = \frac{1}{|\Sigma|^{1/2}} \exp\left[\frac{[\phi^{-1}(u_1)\ \phi^{-1}(u_2)]^\tau(\Sigma^{-1}-I)[\phi^{-1}(u_1)\ \phi^{-1}(u_2)]}{2}\right] \tag{4.24}$$

and

$$\Sigma = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}.$$

PSO algorithm described in Table 4.2 is utilized to find ρ which maximizes the cost function given by (4.22).

4.4. Performance of the overall system

In this experiment, a distributed detection system with two sensors and a fusion center is simulated. The signal is simulated using a multi-variate Gaussian probability density with means equal to μ and variances equal to σ^2 as follows:

$$S = [s_1\, s_2]^\tau = \mathcal{N}([\mu\, \mu], \mathbf{\Sigma}), \tag{4.25}$$

where the covariance matrix Σ is as follows:

$$\Sigma = \sigma^2 \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$$

and ρ denotes the correlation coefficient. Here, the parameters of the multi-variate Gaussian density is taken as μ=40, $\sigma^2 = 10$ and $\rho = 0.8$.

Observation noise is simulated as Gaussian mixtures for both of the sensors. Then, observed data at each sensor can be expressed as follows:

$$\begin{cases} y_i = n_i & when\ H_0\ is\ true \\ y_i = s_i + n_i & otherwise \end{cases}, \tag{4.26}$$

where n_i is as follows:

$$n_i = \sum_{j=1}^{2} \pi_j \mathcal{N}(\mu_j, \sigma_j^2).$$

At sensor 1, observation noise parameters are $\boldsymbol{\pi}_1$=[0.2, 0.8], $\boldsymbol{\mu}_1$=[5, 5], $\boldsymbol{\sigma}_1^2$ = [5, 1]. At sensor 2, observation noise parameters are $\boldsymbol{\pi}_2$=[0.2, 0.8], $\boldsymbol{\mu}_2$=[1, 1], $\boldsymbol{\sigma}_2^2$ = [15, 1]. Using a switching function, the signal is present within the observed data (H_1) with probability 0.2. For both of the sensors, H_0 is true until 1600^{th} sample where it changes to H_1.

The number of particles used in PSO is 500. Random noise q in (4.18) is sampled from uniform distribution $\mathcal{U}(-0.001, 0.001)$. First noise parameters are estimated from the observed data given by (4.23) using the algorithm in Table 4.1. Initially, the values of the particles of means and variances are drawn from $\mu_0 \sim \mathcal{N}(1, 1)$ and $\sigma_0^2 \sim \mathcal{IW}(5, 10)$, respectively.

After estimating observation noise statistics, sensor thresholds are calculated using the algorithm in Table 4.2. Using each component of the Gaussian mixture, a threshold is estimated; then the sensor threshold is found as a weighted sum where the weights are estimated values of $\boldsymbol{\pi}_i$ for i^{th} sensor. Sensor thresholds and decisions are presented in Figure 4.13. Real threshold values for sensor 1 and sensor 2 are calculated as 42.5 and 34.5, respectively. Optimum results refer to the decision rules given in [4].

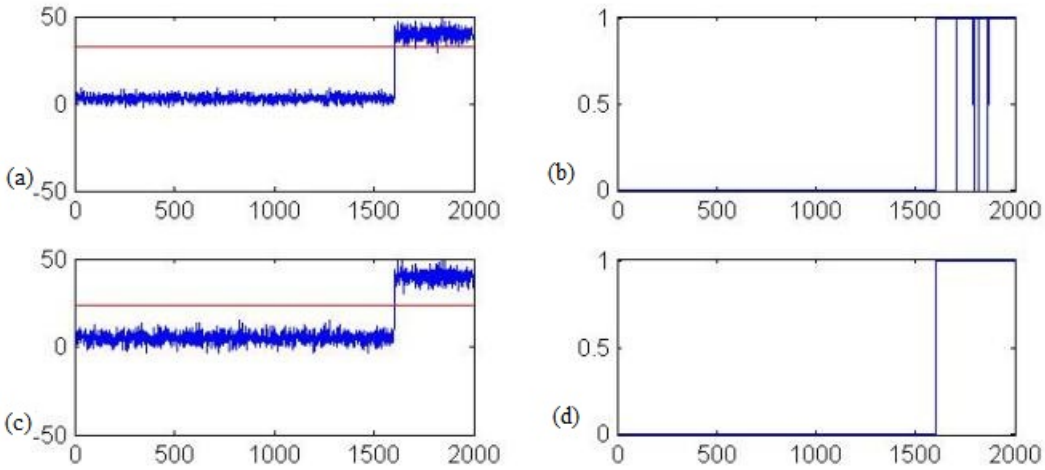

Figure 4.13. Sensor observations, thresholds and corresponding local decisions a) Sensor 1 observations and its threshold, b) Sensor 1 decisions, c) Sensor 2 observations and its threshold, d) Sensor 2 decisions.

Figure 4.13 demonstrates that the thresholds converges to their theoretical values and hypothesis H_1 after 1600^{th} sample is detected. However there are few such as the misses observed in sensor 1 at samples between $1600 - 2000$.

Estimated value of the correlation coefficient using PSO and its true value (0.8) are given in Figure 4.14. MLE estimate of the copula parameter results in to 0.78 which is closer than the result of PSO. Using PSO the parameter estimation performance is not as good as it is with MLE which is a batch estimation method, however PSO is a method which performs sequentially.

Also for different ρ values, estimation results using PSO algorithm are compared in Figure 4.15. Final values of the estimations of ρ values for 0.8, 0.6, 0.2 and 0.2 are 0.76, 0.58, 0.32 and 0.01, respectively. Performance of the method is examined by calculating NMSE of the estimations which is given by Figure 4.16. It can be observed that for higher values of ρ, estimation performance is better compared to cases of where values of ρ decreases to zero which could be a result of using sensor decisions in the estimation of correlation instead of observed data.

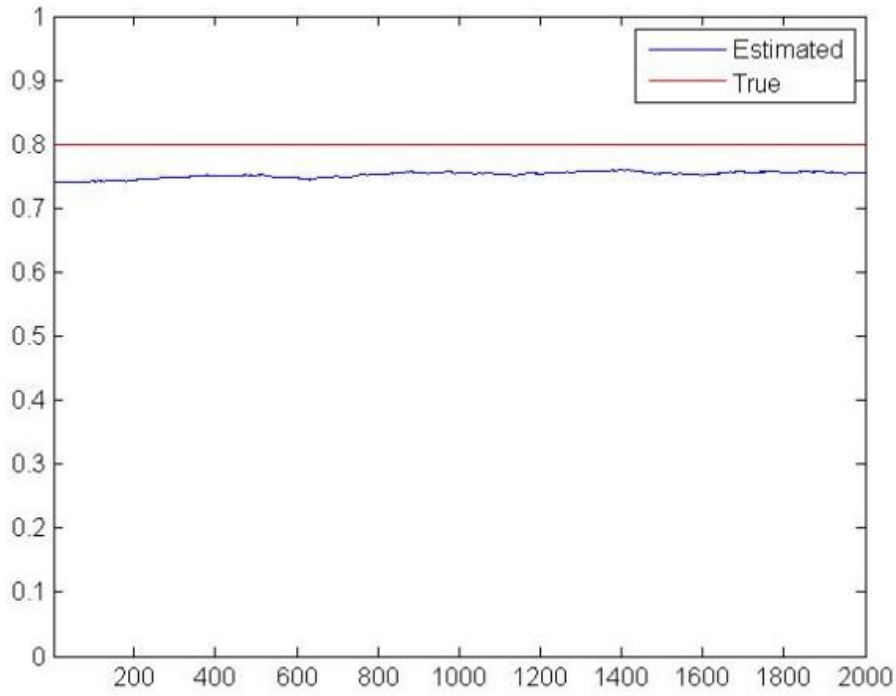

Figure 4.14. Estimated value of the correlation coefficient and its true value.

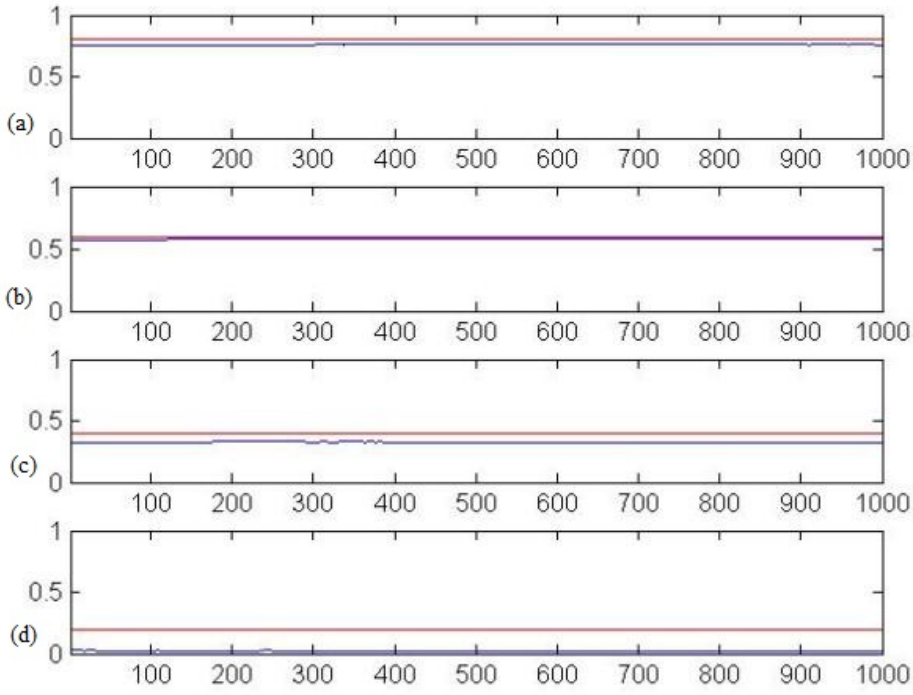

Figure 4.15. Estimated and true values of ρ (in blue and red, respectively) a)$\rho = 0.8$, b) $\rho = 0.6$, c) $\rho = 0.4$, d) $\rho = 0.2$.

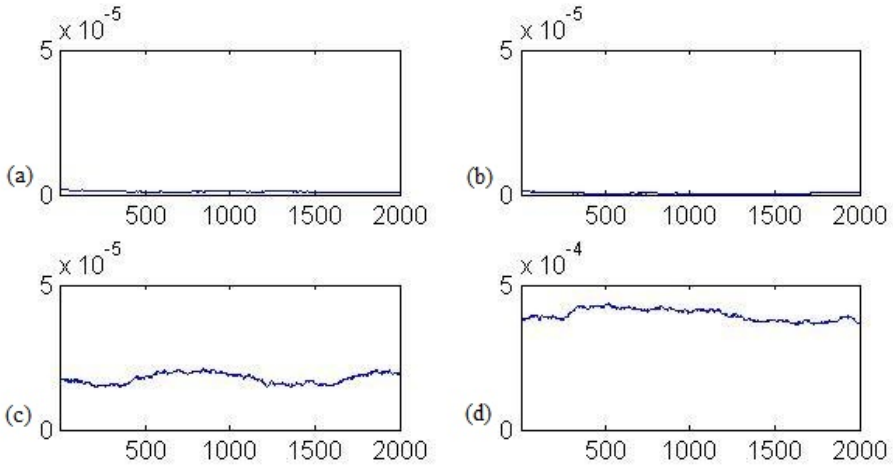

Figure 4.16. NMSE of estimations of ρ a)$\rho = 0.8$, b) $\rho = 0.6$, c) $\rho = 0.4$, d) $\rho = 0.2$.

The signal and the global decision given by the fusion center are given in Figure 4.17.

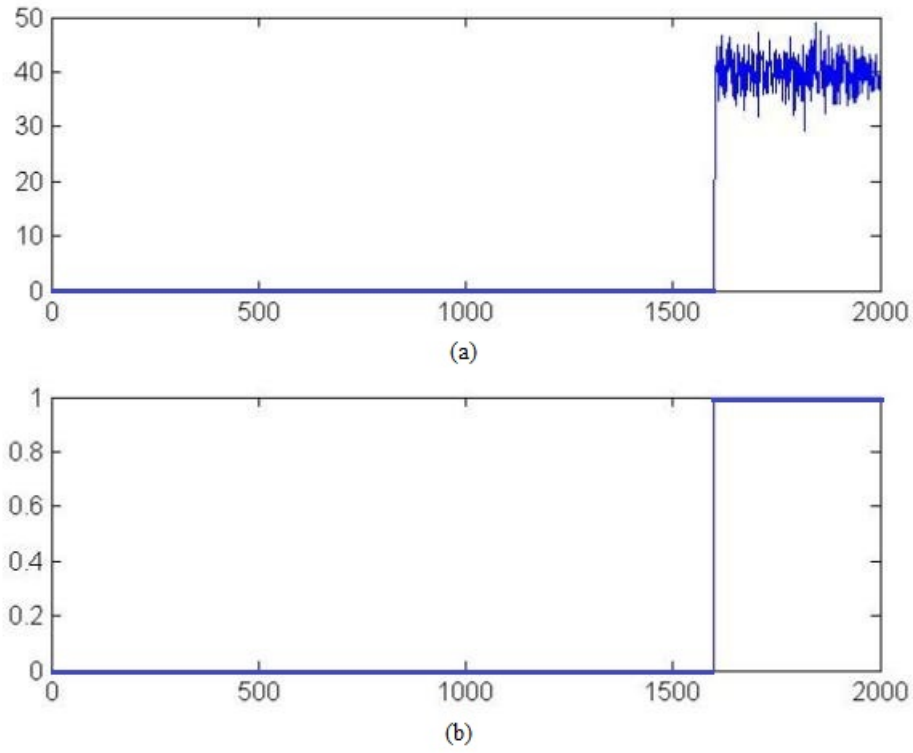

Figure 4.17. The signal and the decision of the fusion center a) the signal, b) global decision.

Figure 4.17 shows that the fusion decisions describe the true hypothesis of the signal. Probability of error values are obtained for two cases, where uncorrelatedness of the local decisions is assumed and correlation is taken into account using the estimated copula parameter. In Figure 4.18, the probability of error when it is assumed that local decisions are uncorrelated is shown. Figure 4.19 shows the probability of error when the correlation among local decisions are utilized in the fusion rule using Gaussian copula whose parameter is estimated using PSO.

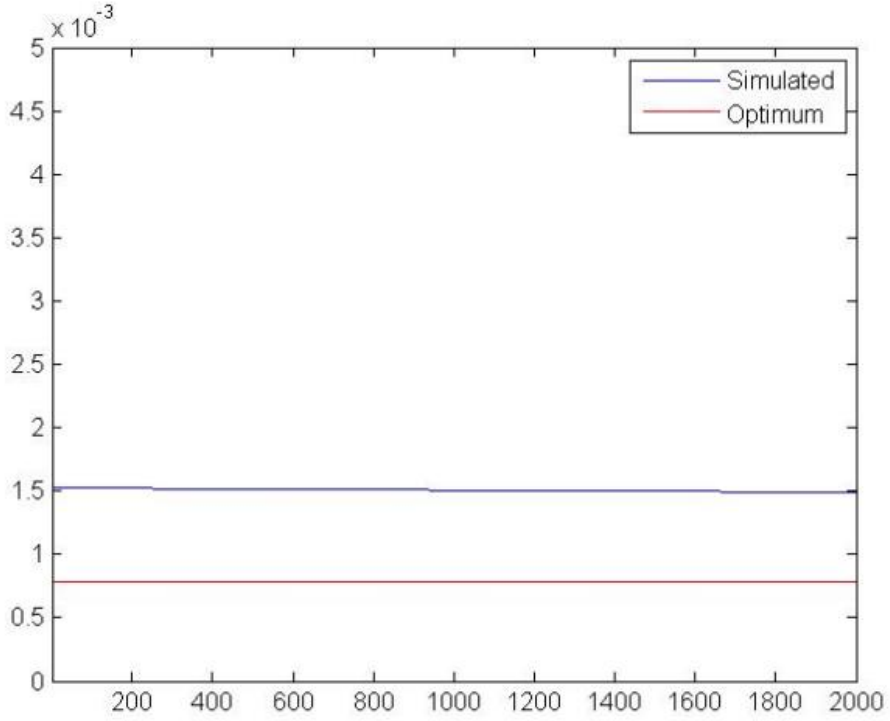

Figure 4.18. Probability of error when it is assumed that local decisions are uncorrelated and optimum value.

The probability of error when the local decisions are assumed to be uncorrelated is higher than optimum value obtained by (2.29)-(2.31). However, when copula function is used probability of error that is achieved approaches to its optimum value. Hence, it can be concluded that considering the correlation among the sensor decisions results in improvement in detection performance than not considering correlation.

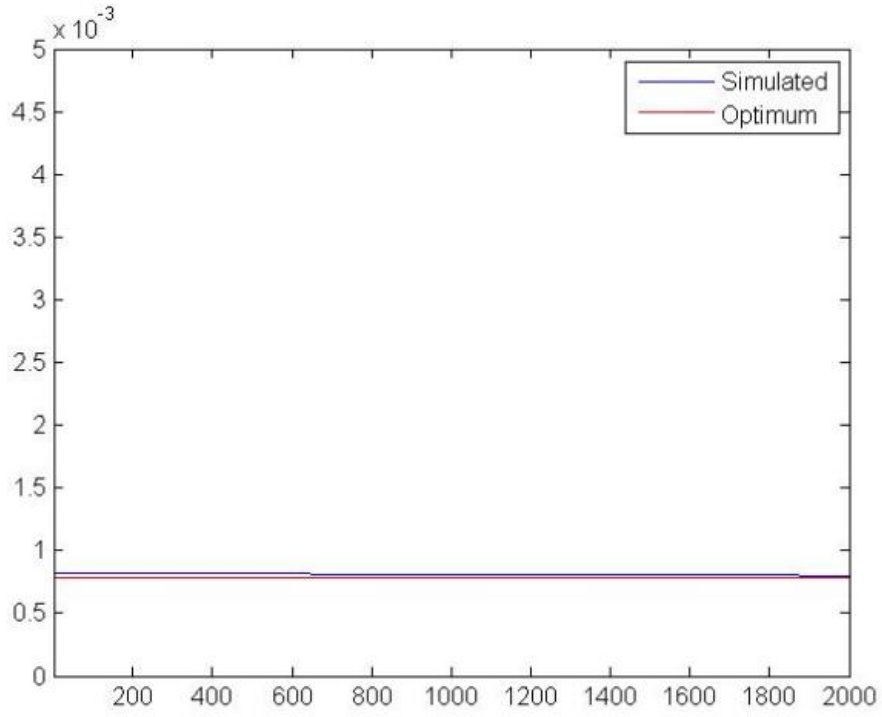

Figure 4.19. Probability of error when Gaussian copula with estimated parameter is used and optimum value.

In addition to the probability of error, receiver operating characteristics (ROC) curve is a graph for illustrating performance of the detection. ROC curve obtained by the proposed decision rules and the optimum ROC curve are given in Figure 4.20.

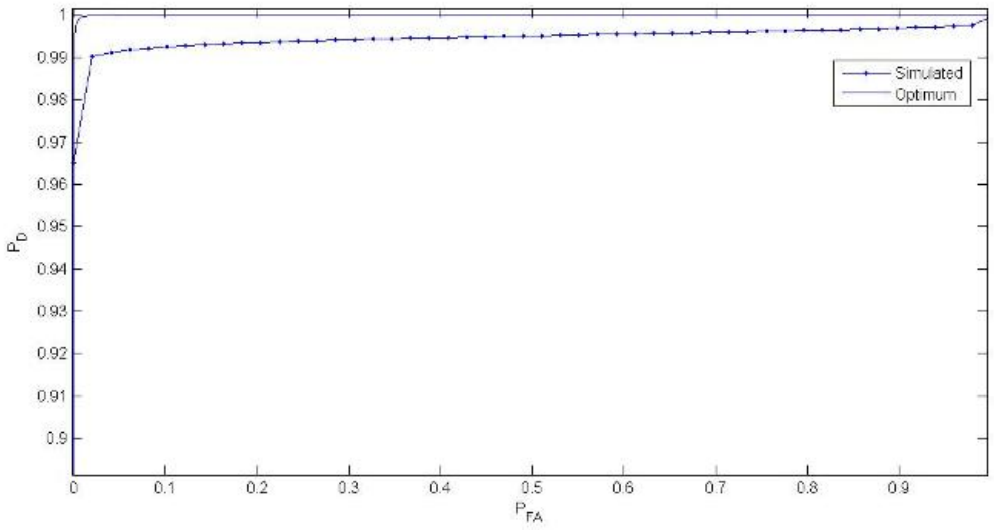

Figure 4.20. ROC curves of the simulation and optimum (theoretical).

It can be observed from Figure 4.20 that in the simulations ROC curve of optimum decision rules result better. Our distributed detection system become closer to optimum as P_{FA} increases.

5. EXPERIMENTS WITH EARTHQUAKE DATA

The performance of the proposed method is tested on real earthquake data provided from the National Earthquake Monitoring Center (UDIM) of Boğaziçi University Kandilli Observatory and Earthquake Research Institute (KOERI). In the experiment, earthquake data from two different sensors are used. Observations of the first and second sensors are given in Figure 5.1. It is given that the earthquake occurs at time $t = 24140$.

Earthquake signal is modeled as 2^{nd} order AR process as in (4.1), observation noises on the sensors are modeled as mixture of two Gaussian components as in (4.2). Noise parameters and AR coefficient are estimated for each sensor. Estimated noise parameters are fed to PSO method; the local and global (by fusion) decisions are made.

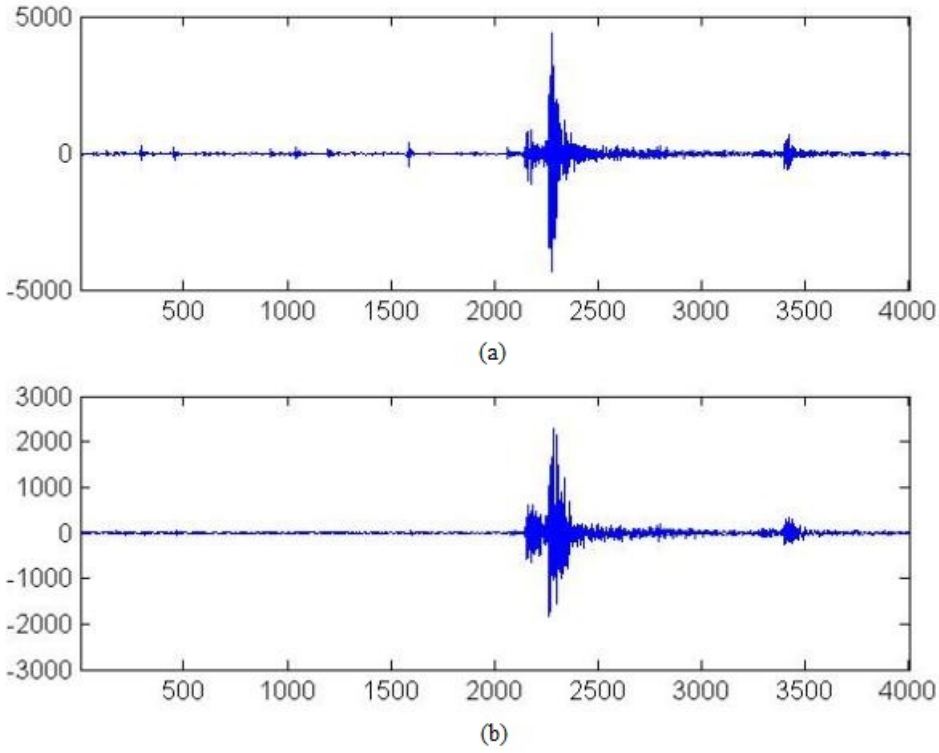

Figure 5.1. Sensor observations a) Sensor 1 b) Sensor 2.

It can be observed from Figure 5.1 that data between $2100 - 2700^{th}$ samples indicate significant seismic activities. Also, between $3200 - 3500^{th}$ samples there are observable activities in seismic data which represent after-shocks.

In Figure 5.2 and 5.3, estimated parameters of the sensors 1 and 2 are given, respectively. In this experiment, true values of the estimated parameters are not available since real data is used. Hence, NMSE could not be calculated in this experiment.

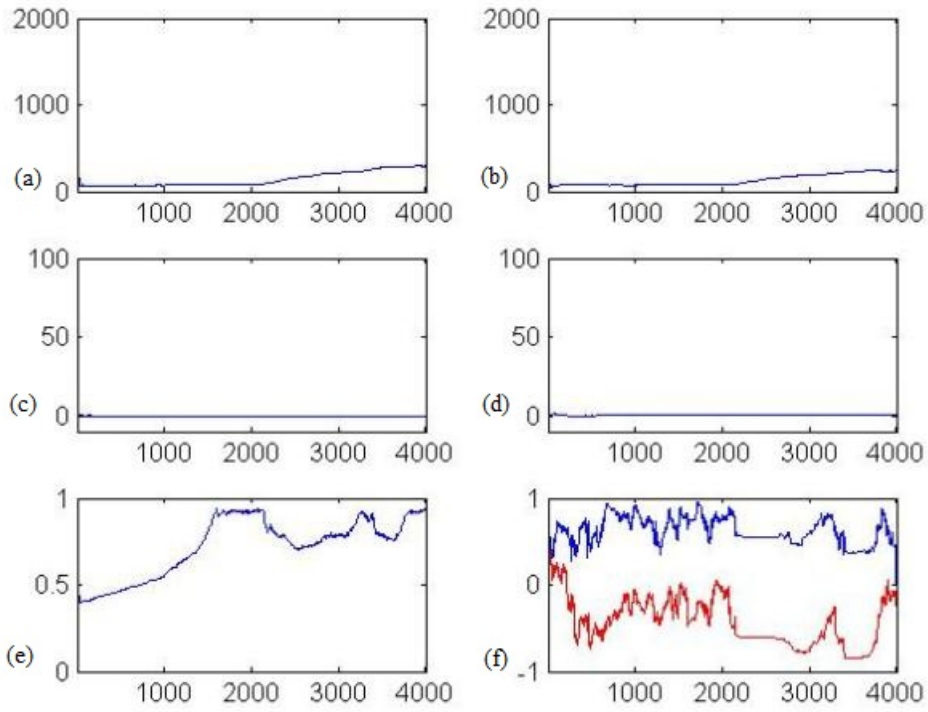

Figure 5.2. Estimated parameters of the Sensor 1 a) $\sigma^2(1)$, b) $\sigma^2(2)$, c) $\mu(1)$, d) $\mu(2)$, e) $\pi(1)$, f) $a(1)$ and $a(2)$ (red).

It can be observed from Figure 5.2 and Figure 5.3 that means of the Gaussian mixtures are the same; however variances of the Gaussian mixtures differ in both of the sensors, especially after $t = 2140$. For both of the sensors, AR coefficients also show the effects of the earthquake after $t = 2140$, as well as $t = 3400$ where after-shocks occurred. This experiment demonstrates that having change in the type of seismic activity results in having changes in the AR coefficients, as expected.

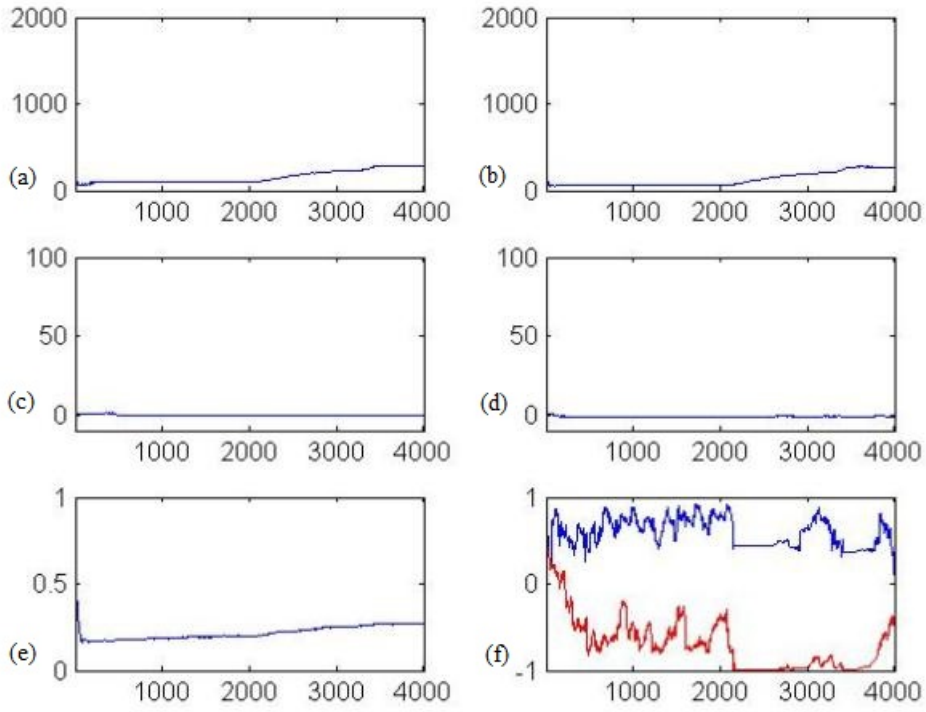

Figure 5.3. Estimated parameters of the Sensor 2 a) $\sigma^2(1)$, b) $\sigma^2(2)$, c) $\mu(1)$, d) $\mu(2)$, e) $\pi(1)$, f) $a(1)$ and $a(2)$ (red).

Thresholds of the sensors upon observed data are presented in Figure 5.4. It can be observed that, the sensor thresholds change when the estimated noise parameters, specifically, variances change after $t = 2140$.

Decision of the fusion center is given in Figure 5.5 which shows that the earthquake at $t = 2141$ is detected, however there are false alarms also. The fusion decision at 3400 may have occurred due to after-shocks. It can be concluded that these findings are consistent with the ground truth the experts specified.

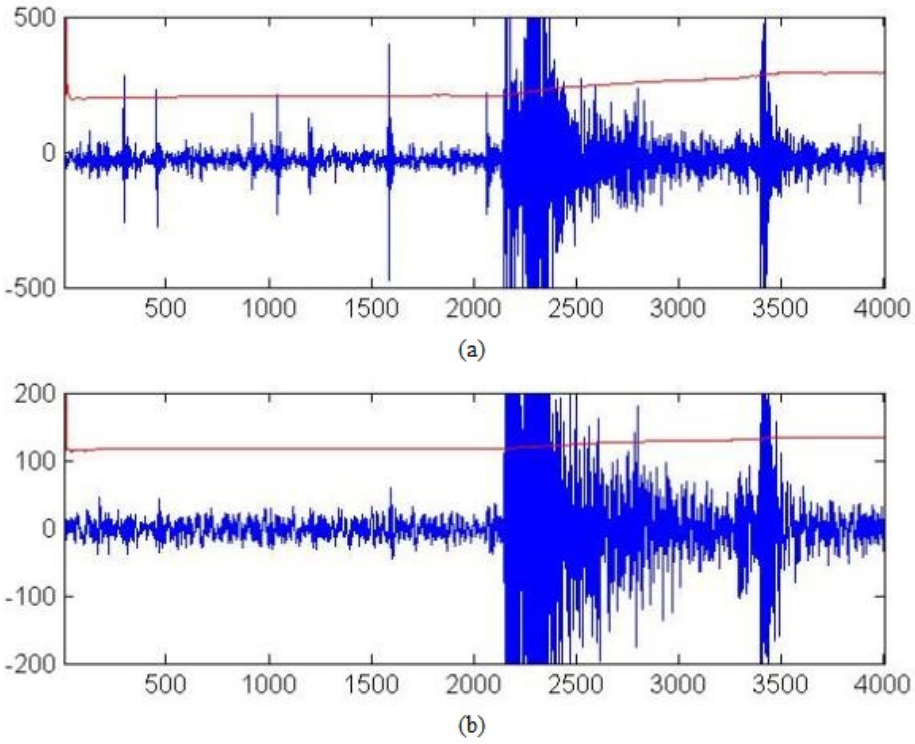

Figure 5.4. Observed data and thresholds of the sensors a) Sensor 1 b) Sensor 2.

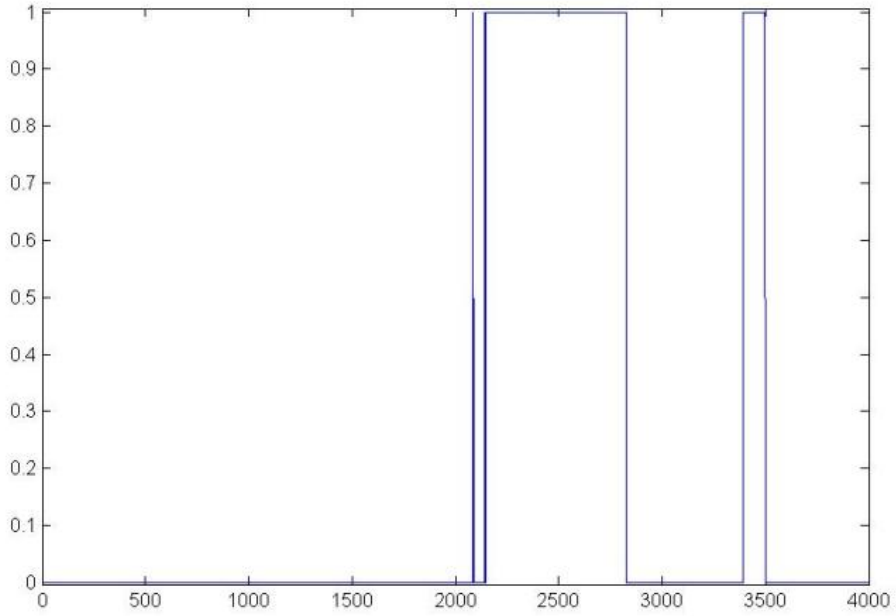

Figure 5.5. Fusion decision.

6. CONCLUSION

In this thesis, usage of particle filtering concepts in distributed detection is investigated. Decision rules for the local detectors and fusion center are developed. Strong assumptions such as the choice of a parametric model and a probability distribution function of the observations as well as assumptions on the optimal detector design such as linearity, stationarity, uncorrelatedness or independence are avoided in the design of decision rules.

First, observed data is modeled as an AR process driven by Gaussian mixture noise. The parameters of the observation noise as well as the coefficients of the AR process are estimated using particle filters that enables adaptivity to time variations both in signal and noise statistics. With the estimation of noise parameters, sensor decisions are estimated using PSO algorithm and a final decision is given. After the noise parameters are estimated, local decision rules are designed using the knowledge of noise parameters. PSO framework is utilized in obtaining the threshold values of the local detectors. Then, local decisions are transmitted to the fusion center. The fusion rule is designed without assuming independence of the decisions of the local sensors. In order to form the statistical dependency between the sensor decisions, Gaussian copula function is used and its parameter is estimated using PSO. Performance of the estimation of the observation noise parameters using particle filtering are investigated in the experiments using simulated and earthquake data. Performance of the overall distributed detection system is measured by the probability of error of the system and ROC curve. Estimation performance of the observation noise parameters are illustrated using NMSE.

The experiments demonstrate that the selection of the variance of the process noise of the parameter affects its convergence time to its true value within an acceptable time interval, however it may also results in observable jitters. Choice of a smaller variance of the process noise could reduce these jitters, but for this case the convergence time would be greater. Moreover, it is observed that when initial values of the param-

eters start from further values than their true values, then their convergences become slower than it is for the case where they start with initial values closer to their true values. Therefore, it could be concluded as the choice of initial values and variance of process noise are important issues within the proposed system. The distributed detection system obtained a higher probability of error than optimum probability of error obtained by (2.29)-(2.31), when the local decisions are assumed to be uncorrelated. When the correlation between the decisions are utilized using Gaussian copula, probability of error value obtained become smaller. In the experiments with real earthquake data, the proposed system yields promising results as the obtained thresholds detect seismic activities that are observed. The experiments demonstrated that sequentially estimating unknown parameters and decision rules provide adaptivity to changes.

It can be concluded that the proposed system can operate under non Gaussian noise without the necessity of making strong assumptions or requiring prior knowledge about statistics of observations. For some experiments the performance may not reach to those with optimum (theoretical), however it provides operating sequentially which is an advantage in real life applications, especially in dynamic environments. As future work, this scheme would be improved to work using additional knowledge from the AR coefficients for an early warning system for the detection of earthquakes. Furthermore, higher order of AR modeling may be utilized for different applications.

REFERENCES

1. Masazade, E., R. Rajagopalan, P. Varshney, G. Sendur and M. Keskinoz, "Evaluation of Local Decision Thresholds for Distributed Detection in Wireless Sensor Networks using Multiobjective Optimization", *Proceedings of the 2008 IEEE Asilomar Conference on Signals, Systems and Computers*, pp. 1958–1962, 2008.

2. Kyriakides, I. and D. Cochran, "Threshold Optimization for Distributed Detection using Particle Filtering Methods", *Fourth IEEE Workshop on Sensor Array and Multichannel Processing, 2006.*, pp. 481–485, 2006.

3. Viswanathan, R. and P. K. Varshney, "Distributed Detection with Multiple Sensors I. Fundamentals", *Proceedings of the IEEE*, Vol. 85, No. 1, pp. 54–63, 1997.

4. Varshney, P. K. and C. S. Burrus, *Distributed Detection and Data Fusion*, Signal Processing and Data Fusion, Springer, New York, 1997.

5. Warren, D. and J. B. Thomas, "Asymptotically Robust Detection and Estimation for Very Heavy-Tailed Noise", *IEEE Transactions on Information Theory*, Vol. 37, No. 3, pp. 475–481, 1991.

6. Chen, B. and P. Varshney, "A Bayesian Sampling Approach to Decision Fusion using Hierarchical Models", *IEEE Transactions on Signal Processing*, Vol. 50, No. 8, pp. 1809–1818, 2002.

7. Nikias, C. L. and M. Shao, *Signal Processing with Alpha-Stable Distributions and Applications*, John Wiley and Sons, New York, 1995.

8. Gençağa, D., A. Ertüzün and A. Kuruoğlu, "Modeling of Non-Stationary Autoregressive Alpha Stable Processes by Particle Filters", *Digital Signal Processing*, Vol. 18, 2008.

9. Gençaga, D., E. E. Kuruoglu, A. Ertüzün and S. Yildirim, "Estimation of Time-Varying AR Symmetric Alpha Stable Processes using Gibbs Sampling.", *Signal Processing*, Vol. 88, No. 10, pp. 2564–2572, 2008.

10. Gençağa, D., A. Ertüzün and E. E. Kuruoğlu, "Modeling of Non-stationary Autoregressive Alpha-stable Processes by Particle Filters", *Digit. Signal Process.*, Vol. 18, No. 3, pp. 465–478, 2008.

11. Sundaresan, A., P. Varshney and N. S. V. Rao, "Copula-Based Fusion of Correlated Decisions", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 47, No. 1, pp. 454–471, 2011.

12. Iyengar, S. G., *Decision-Making with Heterogeneous Sensors - A Copula Based Approach*, Ph.D. Thesis, Syracuse University, 2011.

13. Creal, D. D. and R. S. Tsay, "High Dimensional Dynamic Stochastic Copula Models", *Chicago Booth School of Business Research Paper Series*, 2013.

14. Hafner, C. M. and H. Manner, "Dynamic Stochastic Copula Models: Estimation, Inference and Applications", *Journal of Applied Econometrics*, Vol. 27, No. 2, pp. 269–295, 2012.

15. Van Trees, H., *Detection, Estimation and Modulation Theory*, Wiley, 2004.

16. Kay, S., *Fundamentals of Statistical Signal Processing: Detection Theory*, Prentice Hall Signal Processing Series, Prentice-Hall PTR, 1998.

17. Kennedy, J. and R. Eberhart, "Particle Swarm Optimization", *IEEE International Conference on Neural Networks, 1995. Proceedings.*, Vol. 4, pp. 1942–1948 vol.4, 1995.

18. Poli, R., "Analysis of the Publications on the Applications of Particle Swarm Optimization", *Journal of Artificial Evolution and Applications*, Vol. 2008, pp. 1–10,

2008.

19. Ma, Y., *Crystal Structure Prediction via Particle Swarm Optimization*, 2014, `http://mym.calypso.cn/YanmingMa.html`, July 2014.

20. Tay, W. P., J. Tsitsiklis and M. Win, "Data Fusion Trees for Detection: Does Architecture Matter?", *IEEE Transactions on Information Theory*, Vol. 54, No. 9, pp. 4155–4168, 2008.

21. Kam, M., Q. Zhu and W. Gray, "Optimal Data Fusion of Correlated Local Decisions in Multiple Sensor Detection Systems", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 28, No. 3, pp. 916–920, 1992.

22. Nelsen, R. B., *An Introduction to Copulas (Lecture Notes in Statistics)*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

23. Schmidt, T., *Coping with Copulas*, London: Risk Books, UK, 2007.

24. Kay, S., *Fundamentals of Statistical Signal Processing: Estimation Theory*, Prentice Hall Signal Processing Series, Prentice-Hall PTR, 1998.

25. Gutierrez-Osuna, R., *Probability, Statistics, and Estimation Theory*, 2013, `http://research.cs.tamu.edu/prism/lectures/sp/l1.pdf`, May 2014.

26. Arulampalam, M. S., S. Maskell and N. Gordon, "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking", *IEEE Transactions On Signal Processing*, Vol. 50, pp. 174–188, 2002.

27. Candy, J. V., *Bayesian Signal Processing: Classical, Modern and Particle Filtering Methods*, Wiley-Interscience, New York, NY, USA, 2009.

28. Arulampalam, B., *Beyond the Kalman Filter: Particle Filters for Tracking Applications*, Artech House, London, UK, 2004.

29. Doucet, A., S. Godsill and C. Andrieu, "On Sequential Monte Carlo Sampling Methods for Bayesian Filtering", *Statistics and Computing*, Vol. 10, No. 3, pp. 197–208, 2000.

30. Doucet, A. and A. M. Johansen, "A Tutorial on Particle Filtering and Smoothing: Fifteen Years Later", *Oxford Handbook of Nonlinear Filtering*, pp. 656–704, 2011.

31. Liu, J. S. and R. Chen, "Sequential Monte Carlo Methods for Dynamic Systems", *Journal of the American Statistical Association*, Vol. 93, pp. 1032–1044, 1998.

32. Gençağa, D., *Sequential Bayesian Modeling Of Non-Stationary Non-Gaussian Processes*, Ph.D. Thesis, Boğaziçi University, 2007.