DATA DETECTION AND FUSION
IN
DECENTRALIZED SENSOR NETWORKS

by

NITHYA GNANAPANDITHAN

B.E., Madras University, India

2003

A THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Electrical and Computer Engineering
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2005

Approved by:

Major Professor
Balasubramaniam Natarajan

ABSTRACT

Decentralized sensor networks are collections of individual local sensors that observe a common phenomenon, quantize their observations, and send this quantized information to a central processor (*fusion center*) which then makes a global decision about the phenomenon. Most of the existing literature in this field consider only the data fusion aspect of this problem, i.e., the statistical hypothesis testing and optimal combining of the information obtained by the local sensors. In this thesis, we look at both the data detection and the data fusion aspects of the decentralized sensor networks. By data detection, we refer to the communication problem of transmitting quantized information from the local sensors to the fusion center through a multiple access channel.

This work first analyzes the data fusion problem in decentralized sensor network when the sensor observations are corrupted by additive white gaussian noise. We optimize both local decision rules and fusion rule for this case. After that, we consider same problem when the observations are corrupted by correlated gaussian noise. We propose a novel parallel genetic algorithm which simultaneously optimizes both the local decision and fusion rules and show that our algorithm matches the results from prior work with considerably less computational cost. We also demonstrate that, irrespective of the fusion rule, the system can provide equivalent performance with an appropriate choice of local decision rules.

The second part of this work analyzes the data detection problem in distributed sensor networks. We characterize this problem as a multiple input multiple output (MIMO) system problem, where the local sensors represent the multiple input nodes and the fusion center(s) represent the output nodes. This set up, where the number of input nodes (sensors) is greater than the number of output nodes (fusion center(s)), is known as an overloaded array in MIMO terminology. We use a genetic algorithm to solve this overloaded array problem.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

In this thesis, we study the performance of decentralized sensor networks in the presence of additive white gaussian noise (AWGN) and additive correlated gaussian noise. Although decentralized networks have been studied extensively in recent times, the focus of most of the literature in this field has been on the data fusion aspect of this problem, i.e., the statistical hypothesis testing and optimal combination of the information from all the sensors. The focus of this work is to study both the data fusion and the data detection aspects of decentralized sensor networks. We use the term data detection to refer to the communication problem of transmitting the quantized information from the local sensors to the fusion center through a multiple-access channel. In this chapter, we introduce the reader to sensor networks in general and decentralized sensor networks in particular. We also explain the motivation of this thesis and our key contributions.

This chapter is organized as follows. In section 1.1, we introduce wireless sensor networks and their applications and in section 1.2 we describe decentralized sensor networks. In section 1.3, we present the motivation behind this thesis and finally, in section 1.4, we present the key contributions of this work.

1.1 Introduction to Sensor Networks

Sensor networks are collections of individual sensors that observe a common phenomenon and collectively produce some globally meaningful information. In other words, sensor networks are collections of sensing devices (*local sensors*) that observe some common phenomenon, gather some information and pass it on to a central processor (*fusion center*) that uses this information to either classify or estimate the observation. This is illustrated in figure 1.1. Here, the circles 1 to N represent the sensing devices and $y_i, i = 1 \cdots N$ are their respective observations, which they transmit to the fusion center. Depending on the function of the fusion center, the problem of combining the information can be either a *detection* problem, i.e., deciding among a finite number of states or, an *estimation* problem, i.e., where the fusion center estimates the value of some quantity related to the observation [1]. Due to the low cost of sensors and the inherent redundancy in such systems, sensor networks have been attracting a lot of attention in recent years. Although they were originally used mainly in military

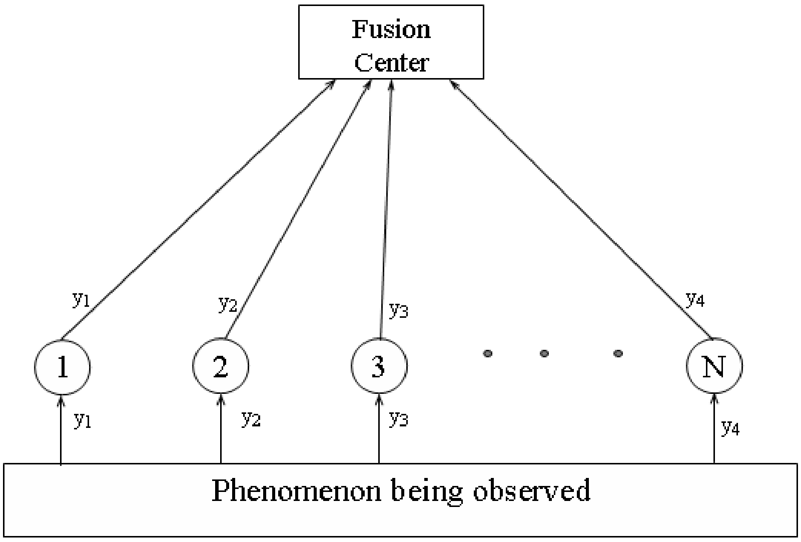

Figure 1.1: Illustration of a sensor network

tracking and control applications, they now have wide array of applications including scientific, industrial, health-care, agriculture, and domestic applications. Owing to the commercial availability of low-cost sensors with networking capabilities, wireless sensor networks have moved over from the research domain into the real world [2]. Some common applications of wireless sensor networks are listed below [3]:

- Industrial control and monitoring [4]
- Home automation and consumer electronics [5]
- Security and military sensing [6]
- Asset tracking and Supply chain management [7]
- Intelligent agriculture and Environmental sensing [8]
- Health monitoring [7]

Traditional multi-sensor systems, where the local sensors do not perform any preliminary processing of data and the central processor performs the detection operation, are known as centralized multi-sensor networks. A major hurdle faced while designing such centralized sensor networks is the communication bandwidth constraint while transmitting information from the local sensors to the fusion center. One way of overcoming this hurdle is by performing some preliminary processing of the data at each local sensor and then sending the condensed information to the fusion center [9]. Such networks are said to have intelligence at each node [10] and are called distributed or decentralized sensor networks. In this thesis we study two aspects of these decentralized sensor networks: data fusion and data detection.

1.2 Decentralized sensor networks

As mentioned in the previous section, decentralized sensor networks have intelligent local sensors that perform some preliminary processing of data before sending it to the fusion center. This is illustrated in figure 1.2. Here the squares represent the local sensors, with the colored part being the sensing component of the sensors and the white part being the processing component. The observations are denoted by $y_i, i = 1 \cdots N$ and the processed information sent to the fusion center is denoted by $u_i, i = 1 \cdots N$. The processing performed by the local sensors is usually some kind of lossy compression such as quantization. Thus, the fusion center has only partial information about the phenomenon observed. This results in a loss of performance in decentralized networks as compared to centralized networks [11]. However, this performance loss can be minimized by optimal processing of the information at the sensors [12]. Moreover, decentralized networks have the advantages of reduced communication bandwidth requirement, increased reliability, and reduced cost which make them very attractive. These advantages have lead to significant research activity in this area [13] -[21]. Although, the common model of distributed sensor networks consists of local sensors and a fusion center, there are sensor networks which operate without a fusion center. One example of such a network would be a sensor network

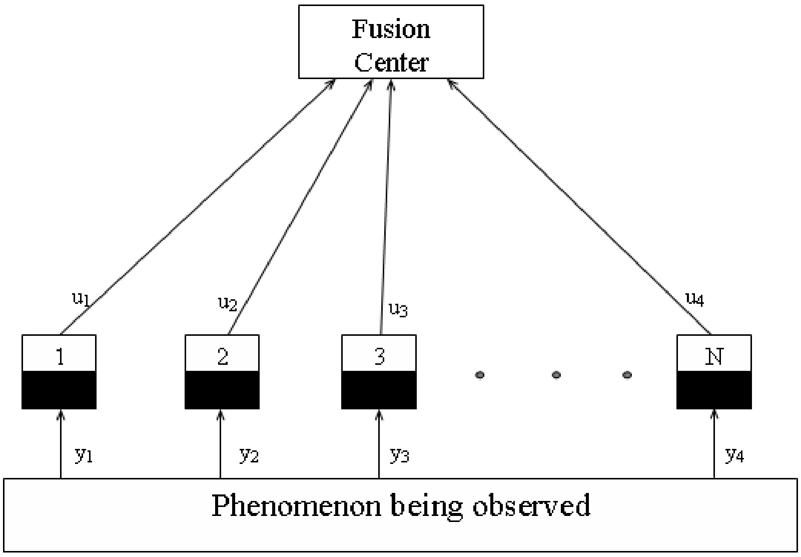

Figure 1.2: Illustration of a distributed sensor network

with serial topology [9], which is described in chapter 3. Another model for a sensor network without a fusion center, is one in which the sensors use parleying to collectively reach a decision [22]. However, in this work, we only consider parallel distributed sensor networks with a fusion center.

1.3 Motivation for thesis

As mentioned in the previous section, distributed sensor networks has been an area of active research within the wireless research community. However, most of the work in this field tend to focus on the data fusion aspect of the problem, which consists of statistical hypothesis testing and combining of the information from all the local sensors. For instance, in [15], Irving and Tsitsiklis demonstrated that there is no loss in optimality if the same decision rule is used in both sensors of a two-sensor distributed network. In [16], the authors analyzed the AND and OR fusion rules for distributed sensor systems and showed that the choice between the two rules depends on the desired false alarm rate as well as the parameters of the probability distributions under both hypotheses. In [14] and [17], the authors have analyzed the constant false alarm rate (CFAR) models of distributed sensor networks. Chamberland and Veeravalli demonstrated that having a set of identical binary sensors is asymptotically optimal as the number of observations goes to infinity [19]. The authors in [21] presented an adaptive fusion model for distributed sensor networks, which estimates the probabilities of detection and false alarm by a simple counting rule. While in [20], the authors investigated the impact of various system parameters on the detection performance of decentralized sensor networks. In this work, we consider both the *data fusion* and *data detection* aspects of decentralized sensor networks.

1.3.1 Data Fusion

The data fusion aspect of decentralized sensor network consists of statistical hypothesis testing and optimal combining of the information from all the sensors. In most decentralized sensor networks, the function of the fusion rule is to decide between one of two hypotheses. This is known as binary hypothesis testing (see chapter 2). The main goal of the designer of a such distributed sensor network is to find the optimal local decision rules and optimum fusion rule for the given network. This problem can be considered for two cases: (i) when the observations of the local sensors are conditionally independent when conditioned on the true hypothesis and; (ii) when the observations are correlated.

The assumption of conditional independence of the sensor observations simplifies the problem and makes it more tractable. This is because in this case, the optimal local classifiers are likelihood ratio tests characterized by a finite number of thresholds [23]. However, many works still resort to asymptotic assumptions and information-theoretic performance measures to further simplify the analysis and design of sensor networks [18]-[20]. On the other hand, even the few studies, such as [9] and [24], that do avoid the use of asymptotic assumptions tend to be limited to simple networks and fail to provide any insight into the structure of the optimal fusion rules. In [25], Alsodari and Moura have adopted a non-asymptotic approach to optimize both the local and global decision rules with respect to the probability of error. In their work, they used a gradient-based approach for optimization of the thresholds of the local detectors and a genetic algorithm (GA) for optimizing the fusion rule. The drawback of their method is that it requires the repeated computation of the gradient of the probability of error along the direction of each of the $N(L-1)$ variables until the algorithm converges. Here N is the number of sensors and L is the number of local decision classes. Such repeated computation of the gradient greatly increases the computational cost. Thus, there is a need for some computationally efficient algorithm that optimizes both the fusion rule and the local decision rules of distributed sensor networks. In chapter 3, we propose such a computationally efficient alternative to the algorithm in [25], that optimizes both the local and fusion rules with greatly reduced computational cost. The author presents these results in part in [26].

As mentioned in the previous paragraph, the conditional independence assumption on the sensor observations simplifies the distributed sensor network problem. However, this assumption is rarely valid in practical situations where the proximity of the sensors to one another will result in correlated observations. Thus, there is a need to study decentralized sensor networks in the presence of correlated observations. This problem has been studied in [27]- [33]. Lauer and Sandell studied the problem of distributed detection in presence of correlated Gaussian noise and derived suboptimum decision rules based on likelihood ratio tests in [27]. In [29], Aalo and Vishwanathan considered a similar problem and evaluated the probability of detection in the Neyman-Pearson (N-P) sense when the fusion rule was fixed to be one of the standard rules such as AND, OR or Majority Voting rule. In [30], the authors once again used the N-P rule to derive the optimum fusion rule for a given set of local decision rules with known correlation among the observations. Aalo and Viswanathan also studied the asymptotic performances of distributed and centralized detection systems in the presence of correlated Gaussian noise in [31]. In [32], the optimal fusion rule is developed for correlated local binary decisions by using the Bahadur-Lazarsfeld expansion of probability density functions. Blum and Kassam considered extending the classical locally optimum detection results to the case of distributed detection with dependent sensor observations in [33]. Thus, most of the literature has been devoted to deriving the optimal fusion rule for a given set of local decision rules or vice versa. To the best of this author's knowledge, no attempt has been made so far to jointly optimize both the local and fusion rules of a decentralized sensor network with correlated observations.

In chapter 4, we demonstrate the simultaneous optimization of both the local and fusion rules in the sense of minimizing the probability of error at the fusion center. We do this by using the algorithm that we propose in chapter 3. In [9] it has been shown that the optimal fusion rule for binary local detectors case with conditionally independent observations, is a majority voting rule. In [25], the authors showed that the optimal fusion rule takes the form of a majority-like voting rule even for the more general case of non-binary detectors. In this work, we show that the optimal fusion rule converges to the majority-like voting rule even for the case of correlated observations, provided that the optimal local decision rules can be assumed to be likelihood ratio tests. The author presents these results in part in [34].

Most of the literature in the field of decentralized sensor networks, including the those cited in the previous paragraphs, tend to focus on the fusion center. This is partly justified since the fusion center plays a very important role in the performance of the distributed sensor network. However, the design of the local decision rules is equally important. In chapter 4, we study the effect of the local decision rules on the performance of the distributed sensor network by fixing the fusion rule and optimizing only the local decision rules. These results were presented in part in [35] by this author.

1.3.2 Data Detection

The data detection aspect of decentralized sensor network refers to the communication problem of transmitting the processed information from the local sensors to the fusion center. This is usually done through a multiple-access channel (MAC). In chapters 3 and 4, we assume that the channel is an error-free MAC. This will not be the case in real life situations where, there will be external noise in the channel that corrupts the information transmitted by the sensors. Thus, it is important to study the communication issues involved in transmitting information from the multiple sensors to the fusion center.

In [36], Tong et. al. proposed an architecture for sensor networks called sensor networks with mobile agents (SENMA). In their architecture, the network consists of two node: the sensors and mobile agents. The mobile agents could be aerial or ground vehicles that moved around the network, collecting data from the sensors, and then transmitting this to the fusion center. Their work focused on the energy efficiency of this SENMA architecture. On the other hand, Cui, Goldsmith and Bahai investigated the energy efficiency of cooperative MIMO sensor networks in [37]. They allowed for cooperation among the sensors when transmitting information to the fusion center and treated the equivalent system as a MIMO system. In [38], the authors proposed a new energy efficient sensor network architecture called MIMO-Sensor networks with mobile agents (M-SENMA), which combined the advantages of both the SENMA and the cooperative MIMO from [37]. Their work also focused on the energy efficiency aspect of the sensor network.

In chapter 5, we analyze the communication aspects of transmitting information from the sensors through a known channel (i.e., the channel matrix is known), to the fusion center in the presence of additive white gaussian noise. We consider that the fusion center receives the transmitted information using N_r receive antennas where, the subscript t is used to denote the receiving side. Thus, we can model this problem as a virtual multiple input multiple output (MIMO) with the N_t local sensors (where the subscript t is used to denote the transmitting side) acting as the multiple input nodes and the N_r receive antennas as the multiple output nodes. MIMO system is the name given to systems that contain multiple input and multiple output nodes. Such systems have the potential to dramatically increase the channel capacity of fading channels as compared to single antenna systems [39] and due to this, they have been studied extensively in literature [40]- [47]. Since, the number of sensors will be greater than the number of receive antennas at the fusion center we have to deal with what is known as an "overloaded" array problem in MIMO terminology. This is the term given to MIMO systems which have greater number of transmitted signals as compared to the number of elements in the receiving antenna array. In [48], the authors proposed the use of a genetic algorithm to solve this overloaded array problem in MIMO systems. In this work, we adopt their approach and apply it our virtual MIMO decentralized sensor network.

1.4 Key Contributions

The key contributions of this thesis are summarized in this section.

In chapter 3, we propose a novel parallel genetic algorithm (PGA) to optimize a decentralized sensor network when the sensor observations are assumed to be conditionally independent when conditioned on the true hypothesis. This yielded the following important findings:

- Our PGA approach converges to the same majority-like fusion rule and minimum probability of error as the gradient-based approach of [25] with greatly reduced computational cost.
- The algorithm converges to the same optimal solution for both homogeneous (identical) and heterogeneous (non-identical) sensors.
- The convergence of the GA depends on the quality of the initial population. For proper convergence, we find that the local thresholds have to initialized close to the region of overlap of the probability distributions of the two hypotheses. The fusion rule has to be initialized such that the probability of deciding in favor of hypothesis H_1 increases as the number of sensors deciding in favor of H_1 increases.

Next, we analyze decentralized sensor networks in the presence of correlated observations. Specifically, we optimize both the local decision rules and fusion rule of the decentralized network in the presence of additive correlated gaussian noise by using the PGA from chapter 3. The results of this optimization are provided in chapter 4. In chapter 4, we also study the impact of the local decision rules on the performance of the decentralized sensor network. We do this by fixing the fusion rule to be one of the three common binary decision rules (AND, OR and majority voting rules) and optimizing only the local decision rules. The key findings from this research include:

- The optimal fusion rule for both correlated and uncorrelated observations is a majority-like fusion rule, irrespective of the degree of correlation.

- The local decision rule plays a key role in optimizing the sensor network when the observations are correlated.

- If the local decisions are assumed to be likelihood ratio tests (LRTs) and are defined completely by quantization thresholds, we find that these thresholds drift closer together as the degree of correlation increases.

- Both homogeneous and non-homogeneous sensors provide similar probability of error performance even with correlated observations

- Systems having different fusion rules can all give equivalent performance provided the local decision rules are chosen appropriately.

- Contrary to the observation made in [29], the OR and AND rules actually have comparable performance for the more general case when the false alarm rate is not fixed and the overall probability of error is minimized.

- Our approach is valid not only for the equicorrelated observations case but also for any arbitrary positive definite covariance matrix.

In chapters 3 and 4, we assumed that the channel is error-free and thus, we did not consider the communication aspects of transmitting the quantized information from the sensors to the fusion center. We consider this data detection problem in chapter 5, where we model our decentralized sensor network as a virtual MIMO system. Our key finding from this include:

- The convergence of the GA to the optimal BER value for a fixed SNR depends on the population size and the size of the mating pool in the GA.
- For fixed population and mating pool size, the BER value decreases with increase in SNR.
- After a certain SNR value, depending on the number of generations, an error floor is reached beyond which the BER does not reduce further with increase in SNR. This can be attributed to the fact that in an overloaded array, the number of variables is more than the number of equations.

In summary, this thesis studies both the data detection and data fusion aspects of decentralized sensor networks in the presence of both additive white gaussian noise and additive correlated gaussian noise.

Chapter 2

Hypothesis Testing

In this chapter, we introduce the reader to the basics of hypothesis testing and detection theory. We provide the measures by which optimality is defined in hypothesis testing problems. We also introduce the three most common tests/formulations - Bayes, minimax, and Neyman-Pearson tests. In addition, we also provide an example of the Gaussian Location Testing problem.

This chapter is organized as follows. Section 2.1 provides an introduction to hypothesis testing. Sections 2.2 and 2.3 introduce the reader to the concepts of Null and Alternative hypotheses, and the different types of errors in hypothesis testing, respectively. The three common tests - Bayes, minimax, and Neyman-Pearson tests - are detailed in section 2.4. In section 2.5, we provide the gaussian location testing example. Finally, in section 2.6, we provide a brief summary of this chapter.

2.1 Introduction

Hypothesis testing, or statistical decision theory, is one of the two broad areas in statistical inferencing, the other being estimation theory. Hypothesis testing can be defined as the problem of choosing one among a finite number of states given an observation y [1]. This observation can be a scalar, a vector or a function. A simple example of an application of hypothesis testing is in radar, where an electromagnetic pulse is sent out and a decision on whether or not the target is present is made based on the echo of the signal returned from the target. Here, the echo is the observation and the states to be decided are “Target present” and “Target absent”. Thus, this problem can be cast as a binary hypothesis testing problem, where the two states correspond to the two hypotheses. Due to the different kinds of noises present in this system, including receiver noise, atmospheric noise, spurious reflections from other objects, etc., the observation will not be completely reliable. Thus, the task of choosing the one of the two states is not very straight-forward and there is a need for some technique to make this inference based on some objective function. This technique, or at least the means for choosing a good technique, is provided by detection theory, which is the name given to hypothesis testing in the context of communication theory [9]. Another example of a situation where hypothesis testing is used is in digital communication system, where one of a set of different waveforms is transmitted over a channel corresponding to a particular symbol. At the receiver, an observation of this signal plus the noise is made, and a decision had to be made about which one of the possible symbols was transmitted. This problem can be modeled as a M-ary hypothesis testing problem, if there are M possible symbols at the transmitter. In this thesis, we consider the binary ($M = 2$) hypothesis problem, where the two hypotheses are presence and absence of the signal, respectively.

2.2 Null and Alternative Hypotheses

In binary hypothesis testing problem, the two states of nature or hypotheses are denoted by H_0 and H_1. The hypothesis H_0 is known as the null hypothesis. This is because it usually represents the absence of the signal or target. The hypothesis H_1 is known as the alternative hypothesis and it usually represents the presence of the signal or target. If P_1 is the probability distribution of the signal plus the noise and P_0 is the probability distribution of just the noise, then the binary hypothesis testing problem can be expressed as follows

$$\begin{aligned} H_0 &: Y \sim P_0 \\ H_1 &: Y \sim P_1 \end{aligned} \tag{2.1}$$

where the notation $Y \sim P_i$ denotes that Y has the distribution P_i. The observation space is represented by Γ. The purpose of detection theory is to provide a hypothesis test or decision rule δ which will partition the observation space into Γ_0 and Γ_1 such that, we will choose the hypothesis H_i when $y \in \Gamma_i$. Thus the decision rule δ can be written as

$$\delta(y) = \begin{cases} 1 & \text{if} \quad y \in \Gamma_1 \\ 0 & \text{if} \quad y \in \Gamma_0 \end{cases} \tag{2.2}$$

In statistics, the binary hypothesis testing problem is defined in terms of the null hypothesis. The hypothesis test results in either accepting the null hypothesis or rejecting it. Hence, the region Γ_0 is called the acceptance region and the region Γ_1 is known as the rejection or critical region.

The a priori or prior probabilities of the two hypotheses H_0 and H_1 are given by π_0 and π_1, respectively. That is, π_j is the probability that hypothesis H_j is true without any prior knowledge of the observation. Since, the two hypotheses are mutually exclusive, $\pi_1 = 1 - \pi_0$. Thus, only π_0 is needed to define the prior or true state of the two hypothesis.

As mentioned earlier, the objective of hypothesis testing is to decide between the two hypotheses, i.e., choose Γ_1 and Γ_0 in some optimum manner. In order to do so, optimality has to be defined in some way. Therefore, we assign costs to our decisions.

The cost incurred in choosing the hypothesis H_i when the true hypothesis is H_j is given by a non-negative number $C_{ij}, i = 0, 1; j = 0, 1$. We use the notation $P_j(\Gamma_i)$ to represent the conditional probability of choosing H_j when the true hypothesis is H_i. Now, we can define the conditional risk of a decision rule $\delta(y)$ under hypothesis H_j as the average cost incurred by the decision rule $\delta(y)$ when the hypothesis H_j is true, i.e.,

$$R_j(\delta) = C_{1j}P_j(\Gamma_1) + C_{0j}P_j(\Gamma_0) \tag{2.3}$$

2.3 Errors - Type I and Type II

In binary hypothesis testing two types of errors can occur. One error occurs when H_0 is true and we decide H_1. This error is known as error of the first kind or Type I error. It is also known as the "False Alarm" error since in radars, this error would correspond to deciding that a target is present when, in fact, there is no target. The probability corresponding to this type of error is thus known as the false alarm probability or false alarm rate. It is given as

$$P_F(\delta) = P_0(\Gamma_1) \tag{2.4}$$

The second kind of error in binary hypothesis testing occurs if we decide in favor of H_0 when H_1 is true. This error is known as error of the second kind or Type II error. Again, in accordance with radar terminology it is also known as the "Miss" probability since, it would correspond to deciding that there is no target when, in fact, the target is present. The probability corresponding to Type II error is thus termed the probability of miss and is given by

$$P_M(\delta) = P_1(\Gamma_0) \tag{2.5}$$

However, while discussing this quantity, it is common practice to use the probability of detection P_D instead which is defined as the probability of deciding in favor of H_1 when H_1 is true. This is given as

$$P_D(\delta) = P_1(\Gamma_1) = 1 - P_M(\delta) \tag{2.6}$$

As we can see, the design of a hypothesis test involves a trade-off between the probability of false alarm and the probability of miss. This is because either one can be made arbitrarily small at the expense of the other. Depending on the type of application, the two errors will have different degrees of importance. A simple example is when the test is used to determine the presence or the absence of a terminal disease. In such a situation, it is better to err on the right side and have more false alarm rather than miss the signs and diagnose that there is no disease when the disease is present. Similarly, there will be situations where it is more important to curb the false alarm rate at the expense of the probability of detection.

Based on the above definitions, we can now define the average probability of error involved in a decision as

$$P_e(\delta) = \pi_0 P_F(\delta) + \pi_1 P_M(\delta) \tag{2.7}$$

The goal of detection theory is to choose an appropriate test or decision rule that is optimum in some sense: either it should minimize the probability of error, or it should maximize the probability of detection, or it should minimize the conditional risk, etc. Therefore, depending on the goal, there are many different ways of choosing this decision rule. Three of the most common decision rules are - Bayes, minimax, and Neyman-Pearson tests. We discuss these three rules in detail in the next section.

2.4 Types of tests

As mentioned in the previous sections, the three most commonly used hypothesis tests are - Bayes test, Minimax test and the Neyman-Pearson test. Each of these tests is optimal in some sense. In this section, we will take a closer look at these three tests.

2.4.1 Bayes Test

In section 2.2, we defined the conditional risk $R_j(\delta)$ (see eqn(2.3)). This conditional risk is simply the cost of choosing H_1 when H_j is true times the probability of choosing thus, plus the cost of choosing H_0 when H_j is true times the probability of this happening. If we have the conditional risks and the prior probabilities of the two hypotheses, we can define an average or *Bayes risk* as the overall average cost incurred by using the decision rule $\delta(y)$. This Bayes risk is given by

$$\begin{aligned} r(\delta) &= \pi_0 R_0(\delta) + \pi_1 R_1(\delta) \\ &= \sum_{j=0}^{1} \pi_j R_j(\delta) \end{aligned} \tag{2.8}$$

where π_0 and π_1 are the a priori probabilities of the two hypotheses H_0 and H_1 and, $R_0(\delta)$ and $R_1(\delta)$ are the respective conditional risks.

The *Bayes rule* is the decision rule $\delta(y)$ that minimizes this $r(y)$. Thus, optimality in this case is defined as minimizing the Bayes risk over all decision rules. This can be done as follows.

$$\begin{aligned}
r(\delta) &= \sum_{j=0}^{1} \pi_j R_j(\delta) \\
&= \sum_{j=0}^{1} \pi_j (C_{1j} P_j(\Gamma_1) + C_{0j} P_j(\Gamma_0)) \\
&= \sum_{j=0}^{1} \pi_j C_{0j} + \sum_{j=0}^{1} \pi_j (C_{1j} - C_{0j}) P_j(\Gamma_1) \qquad \because P_j(\Gamma_0) = 1 - P_j(\Gamma_1) \\
&= \sum_{j=0}^{1} \pi_j C_{0j} + \int_{\Gamma_1} [\sum_{j=0}^{1} \pi_j (C_{1j} - C_{0j}) p_j(y)] \mu(dy) \qquad (2.9)
\end{aligned}$$

where $p_j(y)$ is the probability density function corresponding to P_j. In order to minimize the total risk, the second term in this equation must be ≤ 0. Thus,

$$\begin{aligned}
\Gamma_1 &= \{y \in \Gamma | \sum_{j} \pi_j (C_{1j} - C_{0j}) p_j(y) \leq 0\} \\
&= \{y \in \Gamma | \pi_1 (C11 - C_{01}) p_1(y) \leq -\pi_0 (C10 - C_{00}) p_0(y)\} \\
&= \{y \in \Gamma | p_1(y) \geq \tau p_0(y)\} \\
&= \{y \in \Gamma | \frac{p_1(y)}{p_0(y)} \geq \tau\} \quad \text{where } \tau = \frac{\pi_0 (C10 - C00)}{\pi_1 (C01 - C11)} \qquad (2.10)
\end{aligned}$$

We then define the *likelihood ratio* $L(y)$ as the ratio on the right hand side of the above inequality, i.e. $L(y)$ is the ratio of the probability densities of the two hypotheses.

$$L(y) = \frac{p_1(y)}{p_0(y)} \qquad (2.11)$$

Thus, the test reduces to

$$\Gamma_1 = \{y \in \Gamma | L(y) \geq \tau\} \qquad (2.12)$$

We can now define the Bayes rule $\delta_B(y)$ as follows

$$\delta_B(y) = \begin{cases} 1 & \text{if } L(y) \geq \tau \\ 0 & \text{if } L(y) < \tau \end{cases} \qquad (2.13)$$

As we can see, the value of the threshold τ depends on the prior probabilities and the cost values used. One common cost assignment used in practice is the *uniform* cost assignment. Here, a cost of 1 is assigned for an error, i.e. $C_{01} = C_{10} = 1$ and no cost is assigned for correct decision, i.e. $C_{00} = C_{11} = 0$. For such uniform cost assignment, τ simply becomes the ratio of the two prior probabilities, i.e. $\tau = \pi_0/\pi_1$ and the Bayes risk reduces to the average probability of error given in eqn(3.9). Thus, the Bayes rule for uniform costs is the rule that minimizes the average probability of error.

The *a posteriori* probability $\pi_j(y)$ is defined as the conditional probability that H_j is true given that $Y = y$. This can be written in terms of the prior probabilities $\pi_j, j = 0, 1$ and the conditional probabilities $P_j(y), j = 0, 1$ as

$$\pi_j(y) = \frac{P_j(y)\pi_j}{\pi_0 P_0(y) + \pi_1 P_1(y)} \tag{2.14}$$

We can also define the posterior cost as the average cost incurred by choosing H_i given $Y = y$, i.e. $C_{i0}\pi_0(y) + C_{i1}\pi_1(y)$. We can the rewrite the Bayes rule in terms of the *a posteriori* probabilities as

$$\Gamma_1 = \{y \in \Gamma | C_{10}\pi_0(y) + C_{11}\pi_1(y) \leq C_{00}\pi_0(y) + C_{01}\pi_1(y)\} \tag{2.15}$$

For the case of uniform costs, this becomes

$$\Gamma_1 = \{y \in \Gamma | \pi_0(y) \leq \pi_1(y)\} \tag{2.16}$$

Thus, the Bayes rule reduces to

$$\delta_B(y) = \begin{cases} 1 & \text{if } \pi_1(y) \geq \pi_0(y) \\ 0 & \text{if } \pi_1(y) < \pi_0(y) \end{cases} \tag{2.17}$$

In other words, the Bayes test chooses the hypothesis that has the maximum a posteriori probability of having occurred given that $Y = y$. Therefore, the Bayes rule is also known as the Maximum A Posteriori (MAP) decision rule.

2.4.2 Minimax Test

The Bayes test, as we have seen in the previous section, requires the knowledge of the prior probabilities π_0 and π_1 in order to determine the Bayes risk. This knowledge may frequently be unavailable to the designer of the rule in practical situations. In such cases, the minimax rule provides an good alternative. In Bayes rule, we optimize the decision rule for a single distribution. This decision rule will not necessarily be optimum for every possible prior distribution. In the case of minimax formulation, we try to minimize the maximum conditional risk over all possible decision rules i.e., $\min \max\{R_0(\delta), R_1(\delta)\}$. Hence, the name minimax rule. The minimax criterion is the average risk expressed as a function of the prior probability π_0,

$$\begin{aligned} r(\pi_0, \delta) &= \pi_0 R_0(\delta) + \pi_1 R_1(\delta) \\ &= \{R_0(\delta) - R_1(\delta)\}\pi_0 + R_1(\delta) \qquad (2.18) \end{aligned}$$

Thus, the minimax rule can now be re-written in terms of this risk function as

$$\min_{\delta} \max_{\pi_0} \ r(\pi_0, \delta)$$

For each prior probability π_0, we have Bayes test δ_{π_0} that minimizes the Bayes risk given in eqn(2.8). Let $V(\pi_0) = r(\pi_0), \delta_{pio})$ denote this minimum possible Bayes risk for the π_0. It can easily be shown that $V(\pi_0)$ is a continuous concave function of π_0. Figure 2.1 shows the plot of the average risk $r(\pi_0, \delta)$ and the minimum Bayes risk $V(\pi_0)$ versus the prior probability π_0 for different rules when $V(\pi_0)$ has an interior maximum. As we can see from this figure, if we fix the Bayes rule for a particular prior, say π_0', and then plot the average risk for this Bayes decision rule as a function of the prior probability π_0, we obtain a tangent to the $V(\pi_0)$ curve at that particular prior π_0'. From this, we notice that the Bayes rule which minimizes the Bayes risk for a given prior probability, will not necessarily minimize the risk at a different prior probability. In fact, the risk will not even remain constant for different values of π_0. But, if we find the Bayes rule for the prior probability of π_L and plot the Bayes risk for this rule, we find that it is horizontal . In this case, the value of the risk remains constant regardless of the value of π_L. In other words, for this rule, we are guaranteed that the risk will at least not exceed this value for any value of π_0. This prior π_L is called the least favorable prior as it corresponds to the maximum value of the minimum Bayes risk $V(\pi_0)$ and the Bayes decision rule for this prior corresponds to the minimax rule.

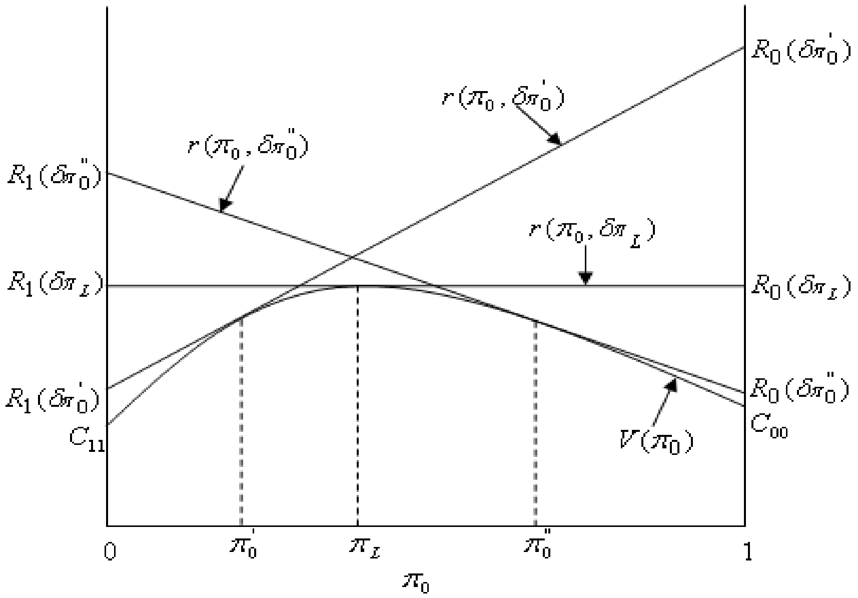

Figure 2.1: Illustration of minimax rule when V has an interior maximum

Thus, the minimax rule is the decision rule that minimizes the risk for the worst case and guarantees that the risk at any other prior will be equal to the risk at that least favorable prior.

2.4.3 Neyman Pearson Test

In Bayes formulation, it is necessary to have knowledge of the prior probabilities and also to be able to make suitable cost assignments. The minimax formulation requires no knowledge of the prior probabilities, but still we have to be able to assign costs to the different decision possibilities. In many situations of practical interest, not only is the knowledge of prior probabilities not available to the designer, even the cost assignments may be difficult to make. In some cases, it may even be undesirable to impose a specific cost structure to the problem. In such situations, Neyman-Pearson criterion provides a feasible alternative.

As discussed in section 2.3, there are two types of errors that can be made in binary hypothesis testing - *False alarm* and *Miss*. As explained in that section, the design of a hypothesis test involves a trade-off between the probabilities of these two errors i.e., probability of false alarm (P_F) and probability of miss (P_M), since either one can be made arbitrarily small at the expense of the other. Neyman-Pearson criterion places a bound on the false-alarm probability and then minimizes the probability of miss. In other words, the Neyman-Pearson rule maximizes the probability of detection ($P_D = 1 - P_M$) for a fixed false-alarm probability, i.e.,

$$\max_{\delta} P_D(\delta) \text{ subject to } P_F(\delta) \leq \alpha$$

where α is the bound on P_F. This value α is known as the significance or level of the Neyman-Pearson test. The probability of detection P_D is known as the power of the test. Thus, the goal of the Neyman-Pearson test is to find the most powerful α-level test of H_0 versus H_1.

The Neyman-Pearson lemma which specifies the general solution to problem of finding the most powerful α-level test is summarized as follows [1]

1. Optimality: Let $\tilde{\delta}$ be any decision rule that satisfies $P_F(\tilde{\delta}) \leq \alpha$, and let $\tilde{\delta}'$ be

any decision rule of the form

$$\tilde{\delta}'(y) = \begin{cases} 1 & \text{if } p_1(y) > \eta p_0(y) \\ \gamma(y) & \text{if } p_1(y) = \eta p_o(y) \\ 0 & \text{if } p_1(y) < \eta p_0(y) \end{cases} \tag{2.19}$$

where $\eta \geq 0$ and $0 \leq \gamma(y) \leq 1$ are such that $P_F(\tilde{\delta}) = \alpha$. Then, $P_F(\tilde{\delta}') \geq P_F(\tilde{\delta})$. In other words, any decision rule of the form of eqn(2.19) and size α is a Neyman-Pearson rule.

2. Existence: For every $\alpha \in (0, 1)$ there is a decision rule $\tilde{\delta_{NP}}$, of the form of eqn(2.19) with $\gamma(y) = \gamma_0$ (a constant) for which $P_F(\tilde{\delta_{NP}}) = \alpha$.

3. Uniqueness: If $\tilde{\delta}''$ is any α-level Neyman-Pearson decision rule, then $\tilde{\delta}''$ must be of the form of eqn(2.19) except possibly on a subset of Γ having zero probability under H_0 and H_1.

2.5 Example of Gaussian Location Testing

We will now provide an example of the gaussian location testing problem to better illustrate the different tests described in the previous section. Since, we have used the Additive Gaussian noise model throughout this thesis, it is important for the reader to understand how the different hypothesis testing schemes work for this model.

Consider the following two hypotheses about the real-valued observation Y:

$$\begin{aligned} H_0 &: y = \epsilon + \mu_0 \\ H_1 &: y = \epsilon + \mu_1 \end{aligned} \tag{2.20}$$

where ϵ is a Gaussian random variable with zero mean and variance σ^2, and μ_0 and μ_1 are fixed numbers with $\mu_1 > \mu_0$. The addition of μ_0 or μ_1 to the Gaussian random variable results in shifting the mean of the random variable to one of those two values. Thus, this hypothesis testing problem is essentially one where we are trying to determine the mean or "location" about which the observation is distributed. Hence the name, Gaussian location testing. The problem can now be re-written as follows

$$\begin{aligned} H_0 &: Y \sim N(\mu_0, \sigma^2) \\ H_1 &: Y \sim N(\mu_1, \sigma^2) \end{aligned} \tag{2.21}$$

where $N(\mu, \sigma^2)$ denotes normal distribution with mean μ and variance σ^2.

We will now derive the optimum decision rules for this hypothesis testing problem using the three criteria that we have described in the previous section.

2.5.1 Bayes rule

The Bayes rule involves the testing of the likelihood ratio $L(y)$ against the threshold τ as given in eqn(2.13). The likelihood ratio for this problem can be evaluated as follows:

$$\begin{aligned} L(y) &= \frac{p_1(y)}{p_0(y)} = \frac{\frac{1}{\sigma\sqrt{2\pi}}\exp(-\frac{(y-\mu_1)^2}{2\sigma^2})}{\frac{1}{\sigma\sqrt{2\pi}}\exp(-\frac{(y-\mu_0)^2}{2\sigma^2})} \\ &= \exp\{\frac{\mu_1 - \mu_0}{\sigma^2}(y - \frac{\mu_0 + \mu_1}{2})\} \end{aligned} \tag{2.22}$$

If we assume uniform costs, the threshold τ is just the ratio of the two prior probabilities i.e.,

$$\tau = \frac{\pi_0}{\pi_1} \tag{2.23}$$

Thus, the Bayes rule for the Gaussian location testing problem is

$$\delta_B(y) = \begin{cases} 1 & \text{if } \exp\{\frac{\mu_1-\mu_0}{\sigma^2}(y - \frac{\mu_0+\mu_1}{2})\} \geq \tau \\ 0 & \text{otherwise} \end{cases} \tag{2.24}$$

Instead of calculating the exponent term in the previous equation and then comparing it to τ, we can evaluate the threshold with which to compare y as follows:

$$\begin{aligned} \exp\{\frac{\mu_1-\mu_0}{\sigma^2}(y-\frac{\mu_0+\mu_1}{2})\} &\geq \tau \\ y \geq \frac{\sigma^2}{\mu_1-\mu_0}\log\tau + \frac{\mu_0+\mu_1}{2} &= \tau' \end{aligned} \tag{2.25}$$

The Bayes rule can now be written as a comparison of the observation y to this new threshold τ'

$$\delta_B(y) = \begin{cases} 1 & \text{if } y \geq \tau' \\ 0 & \text{if } y < \tau' \end{cases} \tag{2.26}$$

If we assume that the two hypotheses have equal a priori probabilities, the original threshold $\tau = 1$ and therefore, the new threshold simply becomes the average of the two means:

$$\tau' = \frac{\mu_0+\mu_1}{2} \tag{2.27}$$

The Bayes risk for this decision rule can be calculated as follows

$$\begin{aligned} r(\delta_B) &= \frac{1}{2}R_0(\delta_B) + \frac{1}{2}R_1(\delta_B) \\ &= \frac{1}{2}P_0(\Gamma_1) + \frac{1}{2}P_1(\Gamma_0) \\ &= \frac{1}{2}P_F(\delta) + \frac{1}{2}P_M(\delta) \\ &= \frac{1}{2}P_F(\delta) + \frac{1}{2}(1-P_D(\delta)) \\ &= \frac{1}{2} + \frac{1}{2}(P_F(\delta) - P_D(\delta)) \end{aligned} \tag{2.28}$$

To find this, we need to evaluate the value of $P_j(\Gamma_1), j = 0, 1$ as follows

$$\begin{aligned} P_j(\Gamma_1) &= \int_{\tau'}^{\infty} P_j(y)dy = 1 - \Phi(\frac{\tau'-\mu_j}{\sigma}) \\ &= \begin{cases} 1-\Phi(\frac{\log\tau}{d}+\frac{d}{2}) & \text{if } j=0 \\ 1-\Phi(\frac{\log\tau}{d}-\frac{d}{2}) & \text{if } j=1 \end{cases} \end{aligned} \tag{2.29}$$

where $\Phi(x)$ is the cumulative distributive function (CDF) of $N(0,1)$ and,

$$d = \frac{\mu_1 - \mu_0}{\sigma} \tag{2.30}$$

is a simple version of the signal to noise ratio (SNR). Thus, the Bayes risk in eqn(2.28) can now be simplified as

$$\begin{aligned} r(\delta_B) &= \frac{1}{2} + \frac{1}{2}(P_0(\Gamma_1) - P_1(\Gamma_1)) \\ &= 1 - \Phi(\frac{d}{2}) \end{aligned} \tag{2.31}$$

2.5.2 Minimax rule

To derive the minimax decision rule, we once again assume uniform costs. The minimum Bayes risk $V(\pi_0)$ can therefore be written as

$$V(\pi_0) = \pi_0 R_0(\delta) + (1 - \pi_0) R_1(\delta) \tag{2.32}$$

where $R_0(\delta)$ and $R_1(\delta)$ are the conditional risks due to hypotheses H_0 and H_1, respectively. These can be evaluated as follows

$$R_0(\delta) = C_{10} P_0(\Gamma_1) + C_{00} P_0(\Gamma_0) = P_0(\Gamma_1) = 1 - \Phi(\frac{\tau' - \mu_0}{\sigma}) \tag{2.33}$$

and

$$\begin{aligned} R_1(\delta) &= P_1(\Gamma_0) = 1 - P_1(\Gamma_1) \\ &= 1 - [1 - \Phi(\frac{\tau' - \mu_1}{\sigma})] \\ &= \Phi(\frac{\tau' - \mu_1}{\sigma}) \end{aligned} \tag{2.34}$$

where, we have used eqn(2.29) to get the expression for $P_j(\Gamma_1)$. Thus, $V(\pi_0)$ can be written as

$$V(\pi_0) = \pi_0(1 - \Phi(\frac{\tau' - \mu_0}{\sigma})) + (1 - \pi_0)\Phi(\frac{\tau' - \mu_1}{\sigma}) \tag{2.35}$$

For all the above equations, τ' is given by

$$\tau' = \frac{\sigma^2}{\mu_1 - \mu_0}\log(\frac{\pi_0}{1-\pi_0}) + \frac{\mu_0 + \mu_1}{2} \tag{2.36}$$

As explained in section 2.4.2, the objective of the minimax rule is to minimize the maximum risk over all possible priors. Or, alternatively, we can also find the prior π_0 which maximizes the minimum Bayes risk $V(\pi_0)$. We can do this by finding the slope of $V(\pi_0)$, equating it to zero and solving for τ', i.e.,

$$\text{Set } \frac{\partial V(\pi_0)}{\partial \pi_0} = 0 \text{ and solve for } \tau'$$

which results in the equation

$$1 - \Phi(\frac{\tau' - \mu_0}{\sigma}) = \Phi(\frac{\tau' - \mu_1}{\sigma}) \tag{2.37}$$

Solving this equation for τ' we get the following unique solution

$$\tau'_L = \frac{\mu_0 + \mu_1}{2} \tag{2.38}$$

This is shown diagrammatically in figure 2.2. Thus, the minimax decision rule for the gaussian location testing problem is given by

$$\delta_B(y) = \begin{cases} 1 & \text{if } y \geq \tau'_L \\ 0 & \text{if } y < \tau'_L \end{cases} \tag{2.39}$$

Substituting for τ' from eqn(2.38) into eqn(2.36) and solving for π_0, we get the least favorable prior as $\pi_L = 1/2$. The minimum Bayes risk corresponding to this least favorable prior, i.e. the minimax risk, is given by

$$V(1/2) = 1 - \Phi(\frac{\mu_1 - \mu_0}{2\sigma}) \tag{2.40}$$

2.5.3 Neyman-Pearson rule

The first step in the process of determining the Neyman-Pearson rule for any hypothesis testing problem is to find the threshold η which will result in a false alarm

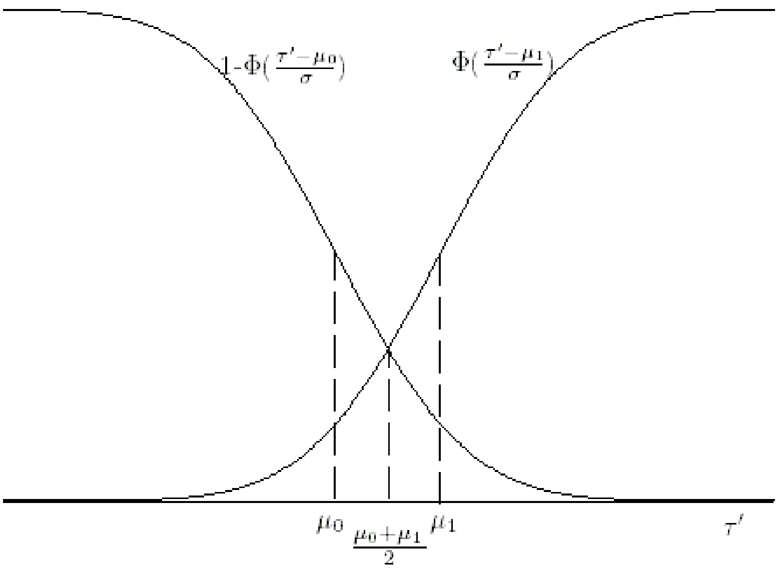

Figure 2.2: Plot of the conditional risks for gaussian location testing problem

rate of α. We do this for the gaussian location testing problem as follows. Let the Neyman-Pearson decision rule be of the form

$$\delta_{NP}(y) = \begin{cases} 1 & \text{if } L(y) \geq \eta \\ 0 & \text{if } L(y) < \eta \end{cases} \tag{2.41}$$

Then, the probability of false alarm can be obtained as

$$\begin{aligned} P_F(\delta) &= P_0(\Gamma_1) = P_0(L(y) > \eta) \\ &= P_0(y > L^{-1}(\eta)) \\ &= P_0(y > \eta') \\ &= 1 - \Phi(\frac{\eta' - \mu_0}{\sigma}) \end{aligned} \tag{2.42}$$

where, the likelihood ration $L(y)$ is as given by eqn.(2.22) and we have used eqn(2.29) to obtain the expression for $P_0(\Gamma_1)$. The expression for η' can be found by taking the inverse of the likelihood ratio, which results in

$$\eta' = \frac{\sigma^2}{\mu_1 - \mu_0} \log \eta + \frac{\mu_0 + \mu_1}{2} \tag{2.43}$$

We equate the expression for false alarm rate given in eqn(2.42) to the constant α and solve for η' as follows.

$$\begin{aligned} 1 - \Phi(\frac{\eta_0' - \mu_0}{\sigma}) &= \alpha \\ \eta_0' &= \sigma\Phi^{-1}(1-\alpha) + \mu_0 \end{aligned} \qquad (2.44)$$

Thus, the Neyman-Pearson decision rule is now given as a threshold comparison of the observation

$$\delta_{NP}(y) = \begin{cases} 1 & \text{if } y \geq \eta_0' \\ 0 & \text{if } y < \eta_0' \end{cases} \qquad (2.45)$$

where, η_0' is as given in eqn(2.44)

The detection probability of the δ_{NP} is given by

$$\begin{aligned} P_D(\delta_N P) &= P_1(\Gamma_1) = P_1(y \geq \eta_0') \\ &= 1 - \Phi(\frac{\eta_0' - \mu_1}{\sigma}) \\ &= 1 - \Phi(\Phi^{-1}(1-\alpha) - d) \end{aligned} \qquad (2.46)$$

where, d is the SNR ratio defined in eqn(2.30). For a fixed α, eqn(2.46) gives the probability of detection as a function of d for the Neyman-Pearson test. This relationship, which is known as the *power function* of the test, is shown in figure 2.3.

For a fixed value of d, eqn(2.46) gives the detection probability as a function of the false alarm rate. The plot of this relationship, which is known as the *receiver operating characteristics* (ROCs), is shown in figure 2.4.

2.6 Summary

In this chapter, we have seen a brief introduction to hypothesis testing. We introduce the reader to the different measures by which optimality is defined in hypothesis testing, such as the conditional and average risks. We define the two types of errors that

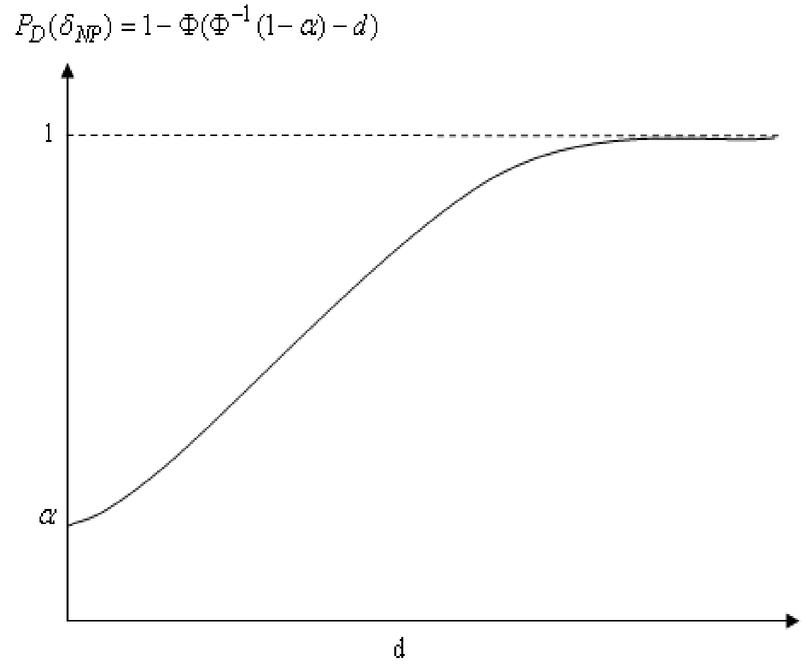

Figure 2.3: Power function plot for the Neyman-Pearson rule for Gaussian location testing

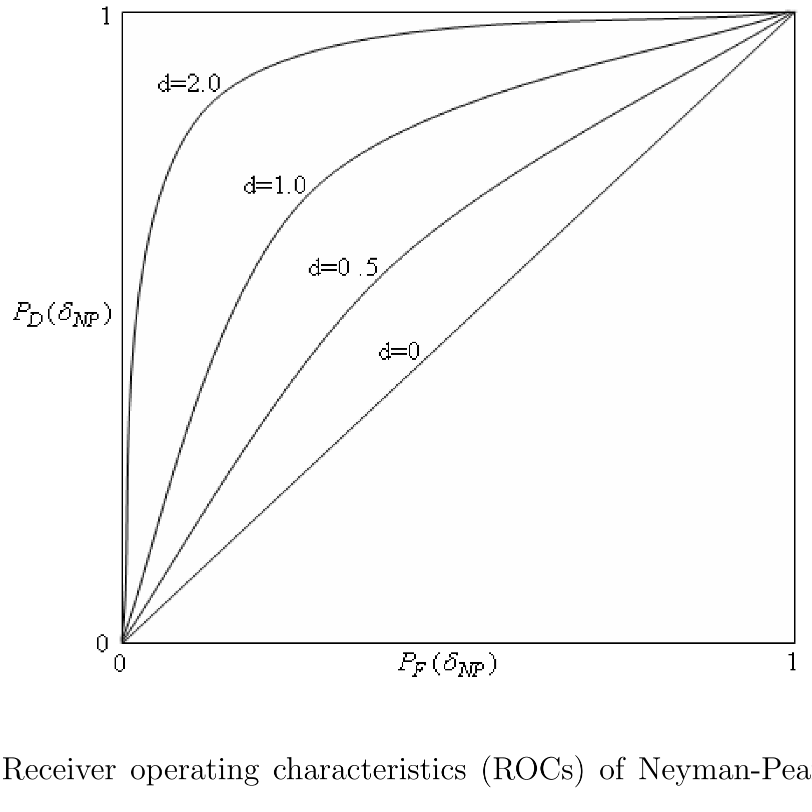

Figure 2.4: Receiver operating characteristics (ROCs) of Neyman-Pearson test for gaussian location testing

occur in hypothesis testing. We also describe the three common binary hypothesis testing schemes: Bayes rule, minimax rule and Neyman-Pearson rule. Finally, we illustrate each of these rules by using the Gaussian location testing example. In the chapters to follow, we will describe how hypothesis testing is applicable to sensor networks, specifically distributed sensor networks. We will detail an alternative method, other than the three rules described in this chapter, for choosing the optimum decision rules for such sensors.

Chapter 3

Decentralized detection - Uncorrelated Case

In this chapter, we introduce the reader to distributed or decentralized detection networks. Specifically, we look at detection and data fusion in distributed sensor networks. We explain the reasons for the popularity of such systems and outline some common areas for their application. We also introduce some of the popular topologies for distributed detection networks, including the parallel topology which is the scheme used throughout this thesis. We then describe the system model and optimization problem that we will be focusing on in this work. We also detail the algorithms used for the optimization. Finally, we present our results and conclusions.

The organization of this chapter is as follows. In section 3.1, we provide an introduction to distributed detection networks and decentralized sensor networks. Section 3.2 contains the description of the system model for the parallel decentralized sensor network analyzed in this thesis and details the optimization problem considered herein. The two kinds of optimization algorithms used in this work, i.e., the Genetic Algorithm-Stochastic Gradient-based Algorithm (GA-SGA) and the Parallel Genetic Algorithm (PGA) are detailed in section 3.3. Section 3.4 contains the analysis of the results obtained using the two algorithms and, Section 3.5 contains the conclusions and summary of this chapter.

3.1 Introduction

Distributed detection networks are detection schemes where group-decision making is employed. In other words, a number of entities are collectively used in the decision making process. The obvious advantage of such a scheme would be the increased reliability and the redundancy inherent in it. One area where distributed detection is widely used is in sensor networks.

Sensor networks are collections of individual or local sensors that observe a common phenomenon and collectively produce some globally meaningful information. Sensor networks have wide array of applications including military, scientific, industrial, health-care, agriculture, and domestic applications. Traditionally, multi-sensor systems consisted of a number of local sensors which sense the common observation and communicate all their data to a central processor, which then performs optimal decision making using some conventional technique. Such a system is known as a centralized multi-sensor network. One of the challenges faced in the design of such centralized sensor networks is the limited power available in the sensors and the communication bandwidth constraints. One way of reducing the bandwidth requirement is to perform some preliminary processing of the data at each local sensor and then send the condensed information to the central processor (*fusion center*). Decentralized sensor networks is the name given to such networks which are becoming increasingly popular. The reasons for this popularity are the relatively low cost of sensors, the redundancy inherent in multiple sensors, the availability of high speed communication networks, and increased computational capability [9]. These advantages have lead to significant research activity in this area[13].

The three major topologies used for distributed signal processing are the *parallel*, *serial* and *tree* topologies [9]. Figure 3.1 shows the general structure of a distributed sensor network with parallel topology. Here, H is the phenomenon that is being observed by the sensors S_1 through S_N. The observation of sensor s_i is denoted by y_i and u_i is the local decision that it makes based on the information in y_i. The local decisions of all the N sensors are transmitted to the fusion center which then makes the global decision u_0 based on the information from all the sensors.

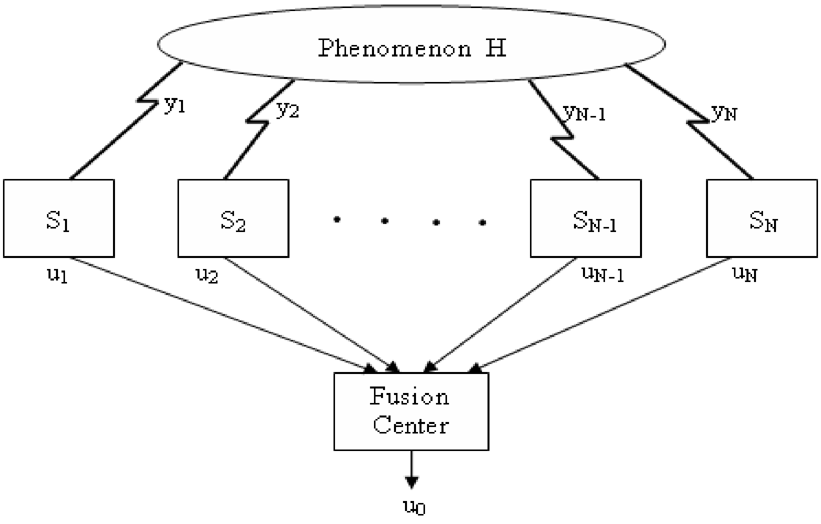

Figure 3.1: Distributed sensor network - Parallel topology

Throughout this thesis, we consider a distributed sensor network having this parallel topology.

Figure 3.2 shows the general form of a distributed sensor network having serial topology. Here, we have used the same notation as in the case of the parallel topology. In serial topology, there is no fusion center as in the case of the parallel network scheme. Here, each sensor generates its decision or quantized information based on its own observation and the quantized information received from the previous sensor, i.e., the ith sensor uses its observation y_i and the information u_{i-1} from the

$i-1$th sensor to generate its quantized information u_i. The first sensor in the network, S_1, uses only its observation to generate its quantized information. The decision of the last sensor S_N is taken as the global decision about which of the two hypotheses is true.

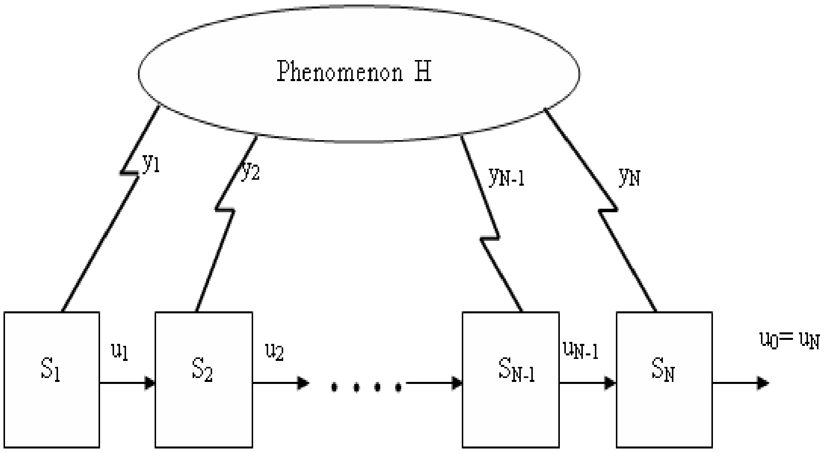

Figure 3.2: Distributed sensor network - Serial topology

Figure 3.3 shows an example of a distributed sensor network with tree topology. As we can see, the tree topology resembles a directed acyclic graph with the fusion center as the root of the tree. The information from all the sensors flows through a unique path to the fusion center.

As mentioned earlier, unlike the fusion center in centralized sensor networks, the fusion center in decentralized networks has only partial information about the observations. This results in a loss of performance in decentralized networks as compared to centralized networks [11]. Thus, one of the major challenges in the design of decentralized systems is to make this performance loss as small as possible by optimally processing the information at the sensors. This involves developing computationally efficient algorithms for processing the information at the sensors and for combining

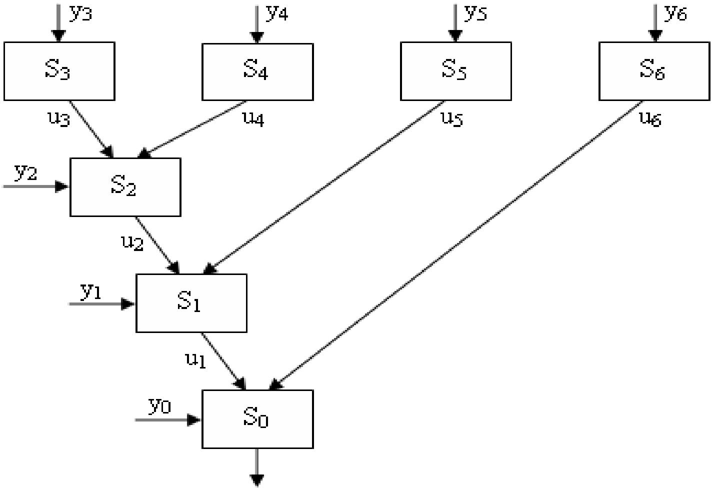

Figure 3.3: Distributed sensor network - Tree topology

the information at the fusion center. Thus, one of the main objectives in the design of decentralized sensor networks is to find the optimal local and global decision rules.

Most of the literature in the field of distributed sensor networks turn to asymptotic assumptions and information-theoretic performance measures to simplify the analysis and design of sensor networks [19]-[18]. This results in the abstraction of important details of the problem such as the structure of the fusion rule. Although there are a few studies that avoid the use of asymptotic assumptions (e.g., [9, 24]), these are mostly limited to simple networks and fail to provide an insight into the structure of the optimal fusion rules. In [25], Alsodari and Moura have adopted a non-asymptotic approach to optimize both the sensing and fusion side with respect to probability of error. Their work uses a gradient-based approach for optimizing the thresholds of the local detectors and a genetic algorithm (GA) for optimizing the fusion rule. This method requires the repeated computation of the gradient along the direction of each of the $N(L-1)$ variables (N being the number of sensors and L being the number of local decision classes), until the algorithm converges. This leads to high computational cost.

In this chapter, we propose a computationally efficient alternative to the method proposed in [25] using a *Parallel Genetic Algorithm*(PGA). In our algorithm, both the local thresholds as well as the fusion rule are simultaneously optimized within a single GA. We consider a parallel topology for the decentralized sensor network where there is no communication among the local sensors, and the local detectors feed their quantized decisions to a single fusion center (figure 3.4). In this chapter, we concentrate on the design of the fusion center, i.e., the fusion rule that will be optimal in a probability of detection sense. We compare our results to those obtained using the gradient-based approach outlined in [25]. Unlike [25], where only heterogeneous sensors are considered, we optimize the fusion rule for the case of both heterogeneous and homogeneous sensors. We show that both the cases of homogeneous and heterogeneous sensors converge to the same fusion rule and the same minimum probability of error. We also analyze the effect of the quality of initial solution on the convergence of the GA. Our results show that our PGA approach converges to the same majority-like fusion rule as the gradient-based approach of [25]. The advantage of our approach is a great reduction in the computational complexity. In the next section, we describe our decentralized sensor network model in detail and define the optimization problem that we consider in this chapter.

3.2 Network Model and Optimization Problem

We consider a parallel fusion network shown in figure 3.4, which has N local sensors and a single fusion center. The local sensors gather the measurements y_n, make a local decision u_n per sensor, and transmit these decisions to the fusion center γ_0 through an *error-free* multiple access channel (MAC). The fusion center makes a global decision $\tilde{H}$ about the true state H based on the set of the local decisions obtained from all sensors. Here, we assume that the fusion center itself does not sense the measurements directly.

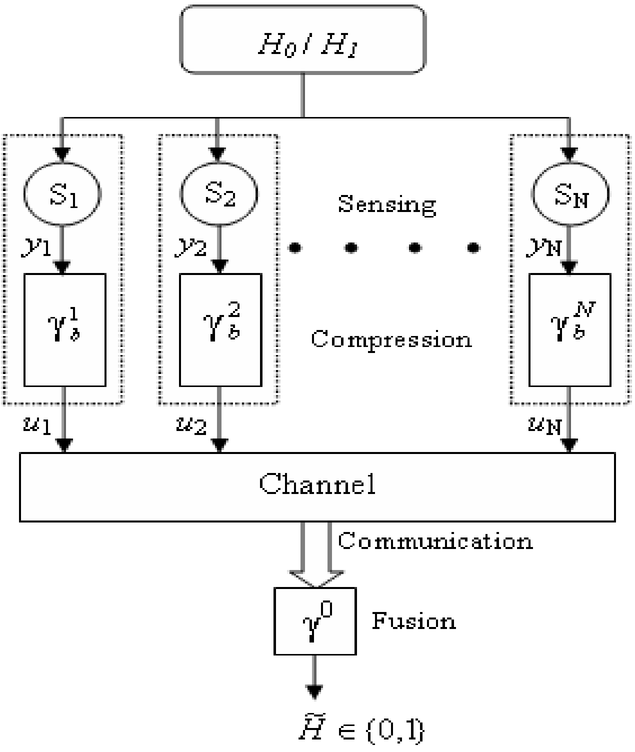

Figure 3.4: Parallel fusion network

We consider the binary detection problem in this decentralized sensor network with hypotheses H_0 and H_1, with known prior probabilities π_0 and π_1, respectively. In this chapter, we assume that the observations $y_n : n = 1, 2, ..., N$ are independent and identically distributed when conditioned on $H_i, i \in \{0, 1\}$. The case when the observations are not conditionally independent is considered in the next chapter.

The final output of the fusion center is binary, i.e., either H_0 or H_1. However, the local sensors are not restricted to binary outputs. Each local sensor classifies its observation y_n into one of $L = 2^b$ classes, where b is the number of transmitted bits per sensor. Thus, each sensor maps the observation space into a classification space that contains L classes, and, the fusion center maps the N local decisions into one of two classes, corresponding to the two hypotheses.

Each possible combination of local decisions are represented by a vector of N integers as follows

$$\mathbf{u} = (\ u_1 \quad u_2 \quad \cdots \quad u_N\ ),\ u_n \in \{0, 1, \cdots, L-1\}. \tag{3.1}$$

Assuming $L = 2^b$, $\mathbf{u}$ can be represented as a string of bN bits as follows

$$\mathbf{u} = (\ u_1^1 u_1^2 \cdots u_1^b\ u_2^1 u_2^2 \cdots u_2^b\ \cdots\ u_{\mathrm{N}}^1 u_{\mathrm{N}}^2 \cdots u_{\mathrm{N}}^b\ ),\ u_{\mathrm{n}}^j \in \{0, 1\} \tag{3.2}$$

Thus, the space of all possible local decisions is spanned by a single bN-bit integer q, whose value ranges from 0 to $2^{bN} - 1$. For a particular combination of the local decisions represented by q, the individual values of the local decisions $u_n, n = 1, 2, \cdots, N$, can be extracted by using a reverse mapping function $\Psi_n(q)$, which is defined as

$$\Psi_n(q) = \frac{q}{2^{b(N-n)}} \bmod L, \tag{3.3}$$

where mod is the modulo operation and all operations are carried out in integer mode.

We adopt the binary representation described in [25] to represent the fusion rules. This representation accounts for the output of the fusion rule under every possible combination of the local decisions. Since there are N sensors and each sensor classifies its measurement into L classes, each fusion rule should account for L^N local decision possibilities and, is therefore represented as a string of L^N bits as follows:

$$\begin{aligned} h = (\ h_0 \quad h_1 \quad \cdots \quad h_{L^N-2} \quad h_{L^N-1}\ ), \\ h_q \in \{0, 1\},\ q = 0, 1, \cdots, L^N - 1 \end{aligned} \tag{3.4}$$

In order to optimize this decentralized sensor network, the optimization has to be performed over all possible local classification rules and all possible fusion rules.

Problem Definition: Determine the optimum fusion rule h and the optimum local decision rules, where optimality is defined in the sense of minimizing the probability of error (P_e) at the fusion center.

The conditional independence assumption on the observations, simplifies the problem since, in this case, the optimal local classifiers are likelihood ratio tests characterized by a finite number of thresholds [49],[23]. Further, if the likelihood ratio $f_1(y)/f_0(y)$ is monotonic in y [50], we can quantize the measurements themselves directly rather than their likelihood ratios. For the case where the observations are Gaussian, at most $L(L-1)/2$ quantization thresholds per local sensor are required to preserve the global optimality of the sensor network [15]. According to [25], numerical results conducted for $b = 2$ on the asymptotic regime show that optimizing a network with $L(L-1)/2$ thresholds per local sensor always converges to a simpler one having only $L-1$ thresholds per local sensor. Hence, for the sake of simplicity, we assume that the local quantizers are characterized by $L-1$ thresholds as follows

$$u_n = \begin{cases} 0 & \text{if } y_n \leq \lambda_{n,1} \\ 1 & \text{if } \lambda_{n,1} \leq y_n \leq \lambda_{n,2} \\ \vdots & \vdots \\ L-1 & \text{if } y_n > \lambda_{n,L-1} \end{cases} \tag{3.5}$$

where, y_n is the local measurement at the nth sensor, u_n is the corresponding local decision, and $\lambda_{n,1}, \lambda_{n,2}, \cdots, \lambda_{n,L-1}$, are the $L-1$ quantization thresholds of that sensor.

Thus, the second part of the problem, namely, finding the optimum local decision rules, boils down to finding the optimum set of $L-1$ thresholds.

3.3 Optimization Algorithm

The problem of optimizing the decentralized sensor network over all possible fusion rules and local decision rules is an NP-complete optimization problem when a discrete observation space is assumed, i.e., the solution cannot be determined in polynomial time [25]. The problem cannot be any easier if we consider a continuous observation space [13]. Thus, this problem has a computational complexity that increases exponentially with the number of users and, hence, it is impractical to implement an exhaustive search. Evolutionary algorithms, such as genetic algorithms (GAs) are one among the many techniques that have been investigated to overcome this limitation. These GAs have been effective in finding approximate solutions for many NP-complete problems. A GA uses evolution and survival-of-the-fittest mechanisms to guide the search toward the fittest candidates [51].

In the following two sub-sections, we present two different algorithms that can be used to solve the optimization problem at hand. The first method outlined here is the approach used in [25]. After that, we detail the algorithm that we are proposing as a computationally efficient alternative to the former.

3.3.1 GA-Stochastic Gradient (GA-SG) Approach

The GA-SG approach uses a GA to search for the optimal fusion rule and a gradient-based algorithm for optimizing the local thresholds. Each chromosome in the population of the GA represents a candidate fusion rule h, which is represented as a string of L^N bits. A random initial population of such chromosomes is generated. The fitness of every chromosome is then calculated by optimizing the local thresholds for that particular fusion rule, and then evaluating the objective function $P_e(\lambda, h)$. A mating pool of parents are selected to undergo cross-over and obtain the offspring population. Once an offspring population is assembled, the fitness of each offspring is evaluated as before and the process is repeated till the search converges to the optimal solution. The optimization of the local thresholds for a particular fusion rule h is implemented by using a gradient-based approach [25]. For each fusion rule, there is a set of $N(L-1)$ thresholds to be optimized with respect to the probability of error, which is a function of both the local thresholds λ, and the fusion rule h. This is a $[N(L-1)]$-dimensional nonlinear constrained optimization problem. But, instead of moving in the direction of the $N(L-1)$-dimensional gradient, each optimization step involves moving along the direction of the one-dimensional gradient with respect to one of the variables as long as the constraints are satisfied. The optimization is then carried out cyclically over all the variables.

Although the GA-SG performs well and converges to the optimal solution (see [25]), it is computationally expensive. This is due to the repeated evaluation of the gradient with respect to the $N(L-1)$ variables for each candidate fusion rule in each generation of the GA. The gradient with respect to one variable $\lambda_{\nu,\tau}$ is evaluated using the following expressions [25]

$$\frac{\partial}{\partial \lambda_{\nu,\tau}} P_e(\lambda, h) = \sum_{k=0}^{1} \pi_k \frac{\partial}{\partial \lambda_{\nu,\tau}} P^0_{\mathrm{k}}(\bar{k}, \lambda, h) \tag{3.6}$$

$$\frac{\partial}{\partial \lambda_{\nu,\tau}} P^0_{\mathrm{k}}(\bar{k}, \lambda, h) = \sum_{\substack{q=0 \\ h_q = k}}^{L^N-1} \frac{\partial P^{\nu}_k(\Psi_\nu(q), \lambda)}{\partial \lambda_{\nu,\tau}} \prod_{\substack{n=1 \\ n \neq \nu}}^{N} P^n_k(\Psi_n(q), \lambda) \tag{3.7}$$

$$\frac{\partial P^n_k(m, \lambda)}{\partial \lambda_{n,\tau}} = \begin{cases} -f_k(\lambda_{n,\tau}) & if\ m = \tau \\ f_k(\lambda_{n,\tau}) & if\ m = \tau - 1 \\ 0 & otherwise \end{cases} \tag{3.8}$$

where π_k is the prior probability of hypothesis H_k, $P^0_k(\bar{k}, \lambda, h) = Pr(u_0 = \bar{k}|H_k)$, $P^n_k(m, \lambda)$ is the probability that the nth sensor decided m when H_k is present and

$f_k(y)$ is the probability density function of y conditioned on H_k.

As can be seen, the evaluation of the gradient is computationally expensive due to the need to iteratively calculate the summation and product terms. In the next subsection, we propose a Parallel GA approach that does not involve any such gradient evaluation.

3.3.2 Parallel GA Approach

The parallel GA that we propose is essentially one in which we optimize both the fusion rule and the local thresholds simultaneously. Each chromosome in the GA is divided into two parts:

- the fusion rule, and,
- a set of local thresholds.

A random population is generated consisting of a group of such chromosomes. The fitness of each chromosome of the population is calculated as the probability of error $P_e(\lambda, h)$, which is evaluated using the following expressions [25]:

$$P_e(\lambda, h) = \sum_{k=0}^{1} \pi_k \sum_{\substack{q=0 \\ h_q = \bar{k}}}^{L^N-1} \prod_{n=1}^{N} P_k^n(\Psi_n(q), \lambda) \tag{3.9}$$

$$P_k^n(m, \lambda) = \int_{\lambda_{n,m}}^{\lambda_{n,m+1}} f_k(y)dy = F_k(\lambda_{n,m+1}) - F_k(\lambda_{n,m}) \tag{3.10}$$

where $F_k(y)$ is the cumulative density function of y conditioned on H_k, and π_k, $P_k^n(m, \lambda)$ and $f_k(y)$ are as described previously. After evaluating the fitness, both the fusion rule part and the thresholds part of the chromosomes, undergo cross-over and mutation individually to produce an offspring population. Elitism is also introduced to ensure that the best solutions in each generation are carried over without any change into the next generation. Once an offspring population is assembled, the fitness is again evaluated and the process continues till a desired termination criterion is reached. Roulette wheel selection scheme is used for selecting parents for cross-over and cross-over of the non-binary thresholds part is performed as follows:

$$\lambda_{offspring} = x\lambda_{parent1} + (1 - x)\lambda_{parent2} \tag{3.11}$$

where x is a uniformly distributed random number between 0 and 1.

The advantage of this algorithm is that it greatly reduces the computational complexity, as the gradient calculations have been eliminated. Although, this means that it takes our parallel GA more number of generations to converge to the optimal solution, it must be kept in mind that each generation only involves the GA processes of cross-over and mutation unlike the GA-SG algorithm where, each generation also involves a complex gradient-based sub-process to optimize the local thresholds, in addition to the GA processes. Therefore, *per-generation computation of the parallel GA is much lower than that in the GA-SG algorithm.*

Another key point to be noted is that, for both the GA with gradient-based threshold optimization and the parallel GA, the initialization of the local thresholds plays a crucial part in the convergence of the algorithms. The local thresholds have to be initialized close to the region of overlap between $f_0(y)$ and $f_1(y)$, which is intuitively reasonable since this is the region where it is hardest to discriminate between the two hypotheses. The initialization of the fusion rule is also equally important. Both the algorithms are found to converge to the optimal solution sooner when the fusion rule is initialized such that the probability of getting a 1 in the rule (corresponding to deciding in favor of H_1) increases as we move from left to right along the fusion rule, i.e., the probability of the first bit (MSB) of the fusion rule being a 1 is 0 while the probability of the last bit (LSB) being a 1 is 1. This is because the first bit of the fusion rule corresponds to the case when all the sensors classify the observation as belonging to level L (highly in favor of H_1). Similarly, the last bit of the fusion rule corresponds to the case when all the sensors classify the observation as belonging to level 0 (highly in favor of H_0).

3.4 Results

In this section, we present the results obtained by using both the GA-SG algorithm and our PGA approach. We consider a parallel decentralized sensor network with N sensors, each making a single measurement, quantizing it into b bits per measurement and transmitting these b bits to a central fusion center via an *error-free* multiple access channel. The fusion center then makes the global decision. We use an additive noise model $y = m_i + n$, where m_i is the signal mean under H_i and n is a zero-mean, unit variance Gaussian noise. The signal means m_0 and m_1 are assumed to be 0 and 1 under H_0 and H_1, respectively. We consider 2 cases: one where the sensors use identical quantizers (homogeneous sensors) and one where the sensors use non-identical quantizers (non-homogeneous sensors). The local observations are assumed to be conditionally independent and identically distributed. We consider sensor networks with 4 sensors, each transmitting 2 bits per measurement. Thus, each sensor quantizes its measurement into one of 4 classes using 3 thresholds. The 3 local thresholds for each sensor are initialized close to -0.5, 0.5 and 1.5, respectively. This corresponds to the region of overlap of the distributions under the 2 hypotheses. The initial population of fusion rules is initialized such that the probability of getting a 1 in the fusion rule increases from 0 to 1 as we move from the least significant bit (LSB) to the most significant bit (MSB). The prior probability π_0 is assumed to be 0.6 for all the cases.

Figure 3.5 shows the evolution of the global probability of error of the sensor network over 100 generations using the GA-SG algorithm. The sensors for this case

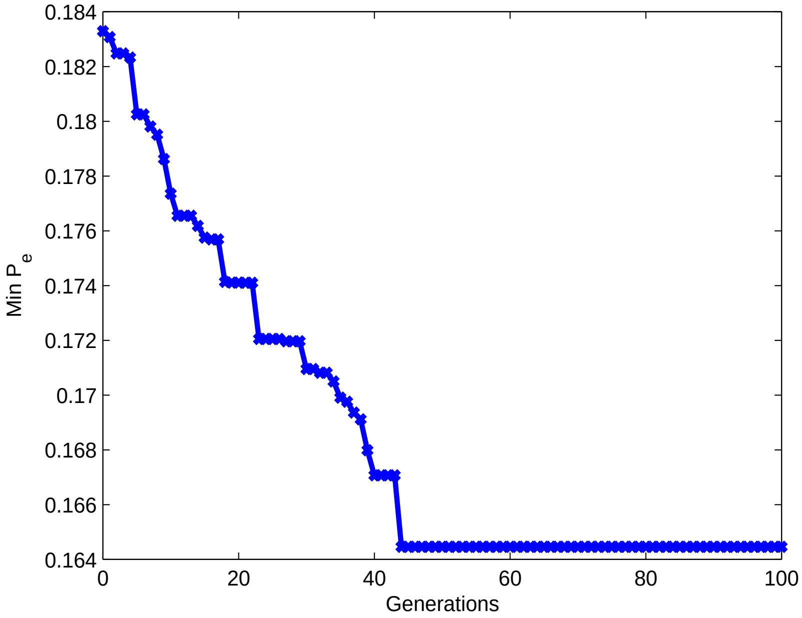

Figure 3.5: Evolution of probability of error using GA-SG Algorithm for non-homogeneous sensors $(N = 4, L = 4, \pi_0 = 0.6)$

are assumed to be heterogeneous, i.e., they have non-identical thresholds. As in [25], the population size is set at 1000 chromosomes while the crossover and mutation rates are 0.45 and 0.01, respectively. The algorithm for this case is found to converge after 45 generations.

Figure 3.6 shows the evolution of the probability of error for the same heterogeneous sensor case using the PGA approach over 4000 generations. The algorithm converges to a minimum after 2500 generations. *Although this is a much large number of generations as compared to that in the GA-SG algorithm, we must recall that the per generation computational complexity of the PGA algorithm is lower than that of the GA-SG by a large degree.* As explained in section 3.3, this is due to the absence of the complex gradient-based sub-process in the PGA approach. The crossover and mutation rates for the PGA are set at 0.45 and 0.03 for the fusion center binary GA. For the local thresholds which are non-binary, the mutation rate is 0.05.

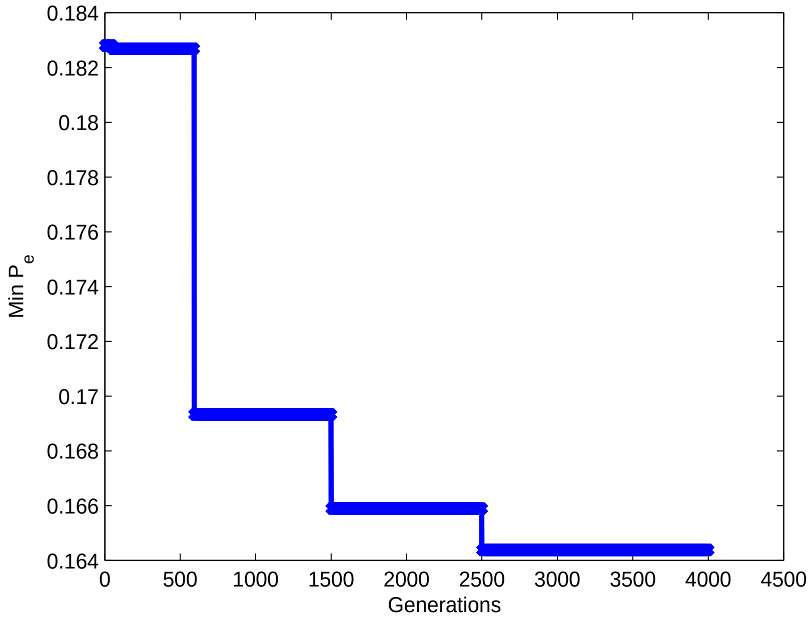

Figure 3.6: Evolution of probability of error using PGA for non-homogeneous sensors $(N = 4, L = 4, \pi_0 = 0.6)$

It should be noted that both the GA-SG algorithm and the PGA converge to the majority-like fusion rule described in [25], where the integer sum of all the local decisions is compared to a threshold given by

$$\lambda^0 \simeq \frac{1}{2}N(L-1) \tag{3.12}$$

where N is the number of sensors and L is the number of quantization levels per sensor. The fusion center decides in favor of hypothesis H_1 if the sum is greater than this threshold and in favor of H_0 otherwise.

Figure 3.7 shows the convergence of the probability of error for the homogeneous sensor case using the GA-SG algorithm over 500 generations. The algorithm is found to converge to the minimum after about 360 generations as opposed to the 50 generations in the case of heterogeneous sensors. Thus, it may seem that it is better to use heterogeneous sensors as the convergence is much faster in that case.

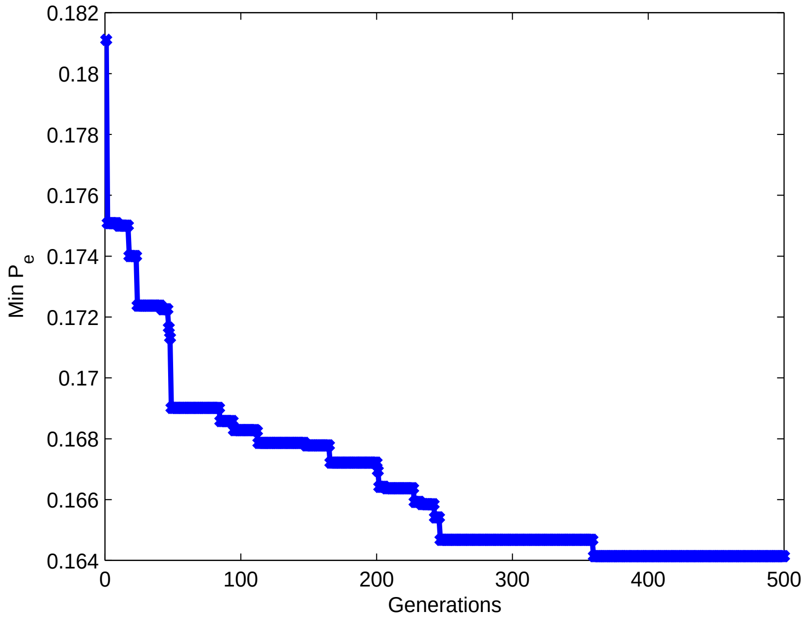

Figure 3.7: Evolution of probability of error using GA-SG Algorithm for homogeneous sensors $(N = 4, L = 4, \pi_0 = 0.6)$

But, we must bear in mind that the per generation computational complexity is much lower in the homogeneous sensor case as there are only L thresholds to optimize using the gradient-based sub-algorithm as opposed to the $N(L-1)$ thresholds in the heterogeneous sensor case.

Figure 3.8 shows the convergence plot of the probability of error for the same homogeneous case using the PGA approach over 8000 generations. Similar to the heterogeneous sensors case, both the PGA and GA-SG algorithms converge to the same majority-like fusion rule for the homogeneous case also. The PGA for this case is found to converge to a minimum after 6000 generations as compared to the 2500 generations in the case of optimizing heterogeneous sensors using the PGA. Even though the algorithm converges slower for homogeneous sensors as in the case of the GA-SG algorithm, the difference is not as pronounced in this case as in the GA-SG case. This is because, the per generation computational complexity is not affected

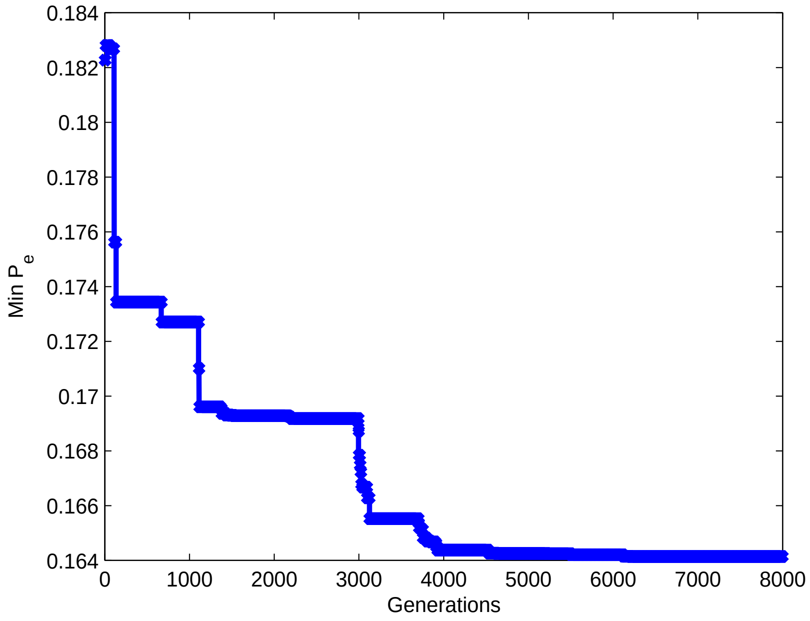

Figure 3.8: Evolution of probability of error using PGA for homogeneous sensors $(N = 4, L = 4, \pi_0 = 0.6)$

much by the number of local thresholds in the case of the PGA as the thresholds are also optimized using the GA and not by a separate sub-process.

Figures 3.9 and 3.10, show the evolution of the probability of error plots for the GA-SG and PGA algorithms, respectively, when the fusion rule is not initialized properly, as explained earlier. For both cases, we considered homogeneous sensors and all other parameters are the same as before. The only difference from the previous simulations is that we used a completely random population for the fusion rule. As we can see, both the algorithms converge to an error probability of around 0.2 and do not reach the optimal minimum probability of error of 0.16, that was obtained in the previous simulations (with proper initial populations). Thus, choosing a good initial population for the GA plays an important role in the proper convergence of the GA.

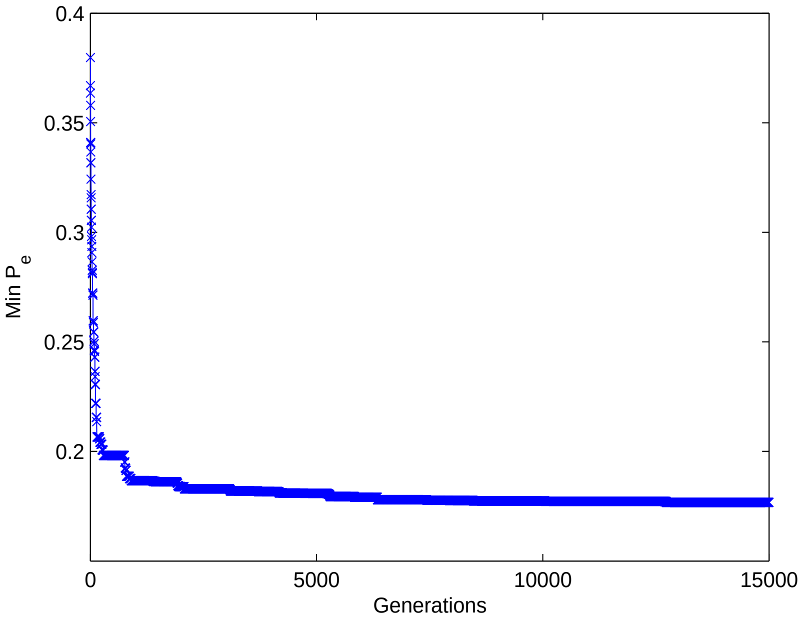

Figure 3.9: Evolution of probability of error using GA-SG Algorithm for homogeneous sensors without proper initialization of fusion rule

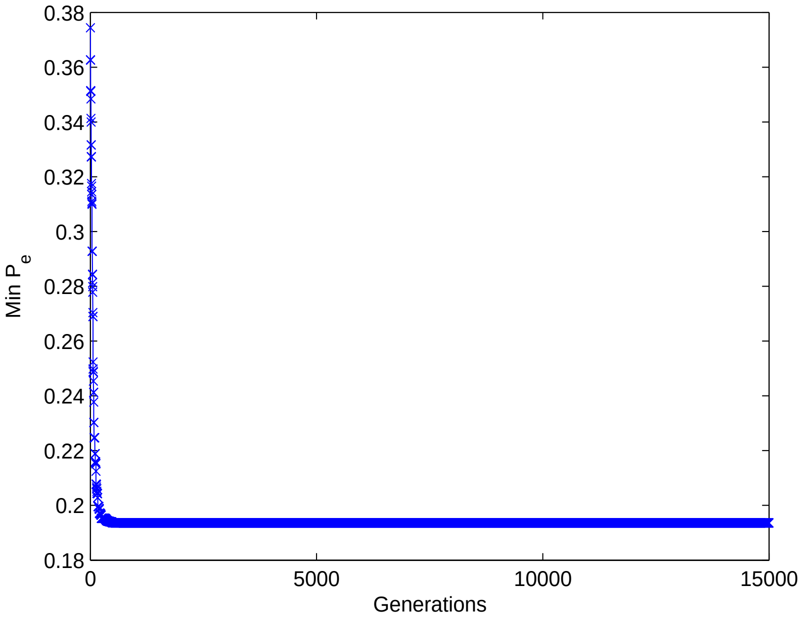

Figure 3.10: Evolution of probability of error using PGA for homogeneous sensors without proper initialization of fusion rule

3.5 Summary

In this chapter, we propose a Parallel Genetic Algorithm approach for optimizing both the fusion rule and local decision rules simultaneously in a probability of global detection error sense. We compare our results to those obtained using the gradient-based approach outlined in [25]. Our results show that our PGA approach converges to the same majority-like fusion rule and minimum probability of error as the gradient-based approach of [25] with greatly reduced computational cost. We optimize the fusion rule for the case of both heterogeneous and homogeneous sensors and show that our algorithm converges to the same optimal solution for both cases. We also analyze the effect of the quality of initial solution on the convergence of the GA. We conclude that the algorithm converges to the optimal solution if the initial population of the GA is selected appropriately. The local thresholds have to be initialized close to the region of overlap of the two hypotheses and the fusion rule has to be initialized such that the probability of deciding in favor of hypothesis H_1 increases as the number of sensors deciding in favor of H_1 increases.

Chapter 4

Decentralized detection - Correlated Case

In this chapter, we study the performance of a decentralized sensor network in the presence of correlated additive Gaussian noise. We first use the Parallel Genetic Algorithm (PGA) approach proposed in the previous chapter to simultaneously optimize both the fusion rule and the local decision rules in the sense of minimizing the probability of error. We show, with the support of our results, that the algorithm converges to a majority-like fusion rule irrespective of the degree of correlation and that the local decision rules play a key role in determining the performance of the overall system in the case of correlated observations. We also show that the performance of the system degrades with increase in the correlation between the observations. We consider both homogeneous (identical) and heterogeneous (non-identical) sensors for this purpose. Next, we analyze the impact of the local decision rules on the performance of a distributed sensor network. We do this by fixing the fusion rule to be one of the three common binary decision rules: majority voting, AND and OR rules, and optimizing the local decision rules with respect to the probability of error at the fusion center. For this analysis, we consider the general case of heterogeneous sensors. We show that systems having different fusion rules can all provide similar performance if the local decision rules are chosen appropriately.

The organization of this chapter is as follows: Section 4.1 introduces the reader to distributed detection with correlated sensor observations and lists some of the prior work done in this area. We provide the model of the system that we are considering in this work in section 4.2 . In section 4.3 we describe how the average probability of error at the fusion center is evaluated for the correlated observation case and describe our optimization algorithms. In section 4.4, we present our results, and in section 4.5 we give our conclusions.

4.1 Introduction

As explained in the previous chapter, decentralized processing, wherein the local sensors perform some preliminary processing of data and then send the compressed information to a central processor (*fusion center*), has the advantages of reduced communication bandwidth requirement, reduced cost and increased reliability. The fundamental problem in decentralized processing is to optimize the performance of the system with respect to the probability of detection at the fusion center by determining the optimal local and global decision rules. This problem has been studied extensively based on the assumption that the observations of the local sensors are conditionally independent when conditioned on the hypothesis [19]-[25]. This assumption simplifies the problem and makes it more tractable since, in this case, the optimal local classifiers are likelihood ratio tests characterized by a finite number of thresholds [23]. However, this assumption of conditional independence is not always valid in practice [27]. This is intuitively true in cases where the physical proximities of the local sensors to each other results in the noise on each sensor being dependent. Hence, there is a need to investigate the problem of distributed detection with correlated sensor observations.

This problem, although less tractable, has also been studied [27]-[30]. However, the focus of most of the literature in this field has been on the design of the fusion center [16]- [23]. While the fusion center does play a very important role in the performance of the distributed sensor network, the design of the local decision rules is equally important.

The analysis of different detector structures in the presence of dependent noise has been carried out for centralized detection scenario [52]-[54]. In [27], Lauer and Sandell analyzed the problem of distributed detection in presence of correlated Gaussian noise and derived suboptimum decision rules based on likelihood ratio tests. Aalo and Vishwanathan considered a similar problem in [29] assuming that the local sensors make binary decisions with all of them operating at the same threshold. Their study involved the evaluation of the probability of detection in the Neyman-Pearson (N-P) sense when the fusion rule was fixed to be one of the standard rules such as AND, OR or Majority Voting rule. In [30], Drakopolous and Lee derived the optimum decision rule in the N-P sense when the local decision rules and the correlations between the local observations are given. Thus, most of the literature has been devoted to deriving the optimal fusion rule for a given set of local decision rules or vice versa. The problem of simultaneously optimizing both the fusion and local decision rules has not been tackled for the dependent noise case. We use the Parallel Genetic Algorithm (PGA) approach outlined in the previous chapter for optimizing both the local and global decision rules simultaneously. We analyze the performance of distributed sensor networks in the presence of correlated Gaussian noise for the case when the local classifiers are assumed to be likelihood ratio tests characterized by a finite number of thresholds. Unlike most of the prior work in this field, we consider non-binary local decision rules. As in [29], we consider both positively and negatively correlated symmetric multidimensional noise distributions which can be completely characterized by a single correlation coefficient ρ. However, our approach can be extended for any arbitrary positive definite noise covariance matrix structure as shown in section 4.4. Furthermore, we present the results for the case when the sensor thresholds are non-homogeneous (non-identical) as well as for the homogeneous (identical) case. Our results show that the performance of the decentralized network degrades as the correlation between the sensors increases. This is intuitively correct as we would expect the distributed system to become equivalent to the single sensor system when the correlation coefficient is equal to 1. In addition to presenting all these results for the two sensor case, we also show the convergence of our algorithm for the three sensor case. Our results show that the algorithm converges to a majority-like fusion rule for all the cases irrespective of the degree of correlation. The local decision rules, on the other hand, are different for the different cases. Thus, it is the local decision rules that play a major role in determining the performance of the decentralized sensor network when the sensor observations are correlated. The PGA approach proposed in this thesis is effective in determining the optimal local decision rules for the correlated observations case.

In [9] and [25], it has been shown that the optimal fusion rule for the case when the sensor observations are assumed to be conditionally independent, when conditioned on the hypothesis, is either a majority voting rule (in the case of binary local detectors) or a majority-like voting rule (for the more general case of non-binary detectors) respectively. In this work, we show that the optimal fusion rule converges to the majority-like voting rule even for the case of correlated observations, if the optimal local decision rules are assumed to be likelihood ratio tests. Other common fusion rules found in literature for the binary local detector case include the AND and the OR rules. In order to investigate the importance of local decision rules, we study the effect of these local decision rules on the performance of the distributed sensor network when the fusion rule is fixed to be one of the 3 rules (majority voting, AND, and OR rules) mentioned above. We assume that the sensors are all binary sensors for this analysis in order to implement the binary fusion rules. In [29], Aalo and Viswanathan have provided numerical examples illustrating the relative performance of the AND, OR and majority rules for a fixed false alarm rate with Neyman-Pearson test. The authors in [29] consider equicorrelated observations with homogeneous (identical) sensors. In this work, we consider a more general case of heterogeneous (non-identical) sensors. We use a genetic algorithm (GA) to optimize the local decision rules with respect to the global probability of detection for a fixed fusion rule without fixing the false alarm or miss probabilities. Also, unlike [29], we show that our approach is valid not only for the equicorrelated observations case but also for any arbitrary positive definite covariance matrix (see section 4.4).

In [29], Aalo and Viswanathan show that the majority voting rule is the best fusion rule only when the false alarm probability is low and, also that the OR rule is inferior to the other two at high false alarm probabilities. This is intuitively true as we would expect the OR rule, which chooses hypothesis 1 under all conditions except when all the sensors decide in favor of hypothesis 0, to have the worst probability of false alarm performance. We show that the OR and AND rules actually have comparable performance for the more general case when the false alarm rate is not fixed and the overall probability of error at the fusion center is minimized. We also show that systems having different fusion rules can all give practically the same performance provided the local decision rules are optimized properly. Thus, we showcase the impact that the local decision rules have on the overall system performance of the distributed sensor network.

In the sections to follow, we provide the system model that we have used for our analysis and explain how we have adapted the PGA from chapter 3 for the correlated observations case. We also provide the derivation of the average probability of error for at the fusion center for the case when the observations are correlated.

4.2 System Model

The system model that we have used for the analysis of distributed sensor networks with correlated observations is essentially the same model that we have used for the uncorrelated case in chapter 3. Here, we have N local sensors gathering observations y_n, making a local decision u_n per sensor, and transmitting these decisions to a single fusion rule γ_0 through an *error-free* multiple access channel (see Figure 3.4). Again, we consider the binary detection problem in such a system, (i.e.) we are testing the two hypotheses H_1 (signal present) and H_0 (no signal). The two hypotheses have prior probabilities π_1 and π_0, respectively. The observation, y_i, at each sensor is given by,

$$y_i = \begin{cases} s_i + n_i & \text{under } H_1 \\ n_i & \text{under } H_0 \end{cases} \quad i = 1, 2, \ldots, N \tag{4.1}$$

As before, the local detectors map these observations into one of $L = 2^b$ classes, where b is the number of bits transmitted to the fusion center by each sensor. The fusion center then makes a global decision $\tilde{H}$ about the true state H based on the set of local decisions from all N sensors.

We assume that the noise on the sensors is additive Gaussian dependent noise. As in [29], we first consider symmetric noise densities which can be completely described by a single correlation coefficient. Thus, for a three sensor system, the covariance matrix for a zero mean Gaussian noise with unit variance has the following form,

$$\Lambda = \begin{pmatrix} 1 & \rho & \rho \\ \rho & 1 & \rho \\ \rho & \rho & 1 \end{pmatrix} \tag{4.2}$$

However, we our approach and results are equally valid for any arbitrary positive definite covariance matrix as we show in section 4.4.

The optimum local classifiers are *assumed* to be likelihood ratio tests. While likelihood ratio tests have been shown to be optimal only for uncorrelated sensor observations ([49],[23]), it is still widely used as the local decision rule for the correlated observation case(see [27]-[32]). As in chapter 3, for the additive Gaussian noise case, the local sensors are assumed to be quantizers with L levels, i.e., $L-1$ thresholds [25]. Thus, the local decision rule corresponds to

$$u_n = \begin{cases} 0 & \text{if } y_n \leq \lambda_{n,1} \\ 1 & \text{if } \lambda_{n,1} \leq y_n \leq \lambda_{n,2} \\ \vdots & \vdots \\ L-1 & \text{if } y_n > \lambda_{n,L-1} \end{cases} \tag{4.3}$$

where, y_n is the local measurement at the nth sensor, u_n is the corresponding local decision, and $\lambda_{n,1}, \lambda_{n,2}, \cdots, \lambda_{n,L-1}$, are the $L-1$ quantization thresholds of that sensor.

We use the same notations as in chapter 3 for the local decisions vector and the fusion rule (See eqns. (3.1),(3.2), (3.3) & (3.4))

4.3 Optimization algorithm

We divide our analysis of the correlated observations case into two parts. First, we simultaneously optimize both the local thresholds and the fusion rule with respect to the probability of error. For this purpose, we use the PGA approach detailed in chapter 3. The algorithm remains essentially the same, with the only change being in the evaluation of the fitness function, i.e., the probability of error. For the second part of our analysis, we fix the fusion rule to be one of the three common binary rules (majority-voting rule, AND rule, and OR rule), and optimize only the local thresholds. Once again, the optimization is carried out in the sense of minimizing the average probability of error at the fusion center. In order to do this, we assume that the local sensors make binary decisions and we employ a single objective genetic algorithm (GA) for the optimization.

For both the problems, the optimization of the decentralized sensor network has to be performed over all possible local thresholds and, in the first case, it has to be performed over all possible fusion rules also. The resulting optimization problem in either case is NP-complete, i.e., the solution cannot be determined in polynomial time. The complexity of the problem increases exponentially with the number of sensors. Thus, an exhaustive search becomes impractical. As explained in the previous chapter, one approach commonly used with this kind of a problem is the use of evolutionary algorithms such as genetic algorithms (GAs).

As mentioned earlier, for the first part of the problem, we use the PGA that we detailed in section 3.3. This is essentially an algorithm which optimizes both the fusion rule and the local thresholds in parallel. Each chromosome in the GA consists of two parts:

- the fusion rule, and,
- a set of local thresholds.

A random initial population is generated which consists of a fixed number of such chromosomes. The fitness of each chromosome is evaluated as the average probability of error at the fusion center for that particular combination of fusion rule and local thresholds. After evaluating the fitness, the chromosomes undergo selection, crossover and mutation. These processes are carried out for both parts of the chromosome in parallel. As before, elitism is also used to ensure that the best solutions from each generation are carried over to the subsequent generation without any mutation. Once an offspring population is assembled with the required number of candidate solutions, the fitness is again evaluated and the whole process continues till a desired termination criterion is reached.

Thus, the PGA is essentially the same as for the uncorrelated case. It is in the evaluation of the probability of error that the correlated case differs from the uncorrelated case. We can no longer simply multiply the probability density functions of the observations of the different sensor as when they were considered to be conditionally independent. Therefore, we need to derive the average probability of error for the correlated observations case. This is done as follows.

The average probability of error at the fusion center is given by the weighted sum of type-I and type-II errors,

$$P_e(\lambda, h) = \sum_{k=0}^{1} \pi_k P_k^0(\bar{k}, \lambda, h) \tag{4.4}$$

where π_k is the prior probability of hypothesis H_k, $P_k^0(\bar{k}, \lambda, h) = Pr(u_0 = \bar{k}|H_k)$ is the probability of false alarm if $k = 0$ or the probability of miss if $k = 1$, and $\bar{k}$ is the binary NOT operation. Out of the L^N mutually exclusive possible local decision combinations, we sum over those that results in $u_0 = \bar{k}$ decision at the fusion center as follows

$$P_e(\lambda, h) = \sum_{k=0}^{1} \pi_k \sum_{\substack{q=0 \\ h_q = \bar{k}}}^{L^N - 1} Pr(u_1 = \Psi_1(q), \cdots, u_N = \Psi_n(q)) \tag{4.5}$$

where $Pr(u_1 = \Psi_1(q), \cdots, u_N = \Psi_n(q))$ is the joint probability of sensor 1 deciding $\Psi_1(q)$, sensor 2 deciding $\Psi_2(q)$, and so on. Since, the local sensors act as quantizers, this joint probability can be evaluated as the following set of multiple integrals

$$Pr(u_1 = \Psi_1(q), \cdots, u_N = \Psi_n(q)) = \int_{\lambda_{\Psi_1}}^{\lambda_{\Psi_1 + 1}} \cdots \int_{\lambda_{\Psi_N}}^{\lambda_{\Psi_N + 1}} f_k(y_1, \cdots, y_N) dy_1 \cdots dy_N \tag{4.6}$$

where $f_k(y_1, \cdots, y_N)$ is the joint probability density of the observations $y_1, \cdots, y_N$. Thus, we use a multiple iterated integral of the joint probability density function for the evaluation of the average probability of error for the correlated observations case.

For the second part of the analysis, we consider only binary sensors, i.e., sensors with $L = 2$ which map their observation into one of two classes. The fusion rule is fixed to be one of the three common binary decision rules: (i) Majority voting rule, (ii) AND rule, or (iii) OR rule. As the name implies majority voting rule decides in favor of the majority of the sensors. AND fusion rule chooses hypothesis H_1 only if all the sensors decide in favor of H_1, and OR rule chooses hypothesis H_0 only if all the sensors decide in favor of H_0. We attempt to determine the optimal local thresholds for each fixed fusion rule by using a GA. Each chromosome of the GA consists of a set of local thresholds. We use the probability of error evaluation, with the multiple integrals of the joint probability density function, which was derived above. The usual GA processes of selection, cross-over and mutation are performed on these chromosomes till the algorithm converges to the optimum set of thresholds. In the selection process, the fitness of each chromosome is evaluated as the average probability of error at the fusion center for the combination of the fixed fusion rule and the set of local thresholds associated with the chromosome.

The initialization of the local thresholds, once again, plays a crucial part in the convergence of both the PGA and the GA. The local thresholds have to be initialized close to the region of overlap between $f_0(y)$ and $f_1(y)$ for proper convergence of the algorithms. This makes intuitive sense since this is the region where it would be hardest to discriminate between the two hypotheses.

4.4 Results

In this section, we present the simulation results obtained by using our PGA for the simultaneous optimization problem and, the GA for the fixed fusion rule problem. We consider a parallel decentralized sensor network, where the global decision is made solely by one fusion center to which all the local sensors transmit their individual decisions through a *error-free* multiple access channel. The local observations are assumed to follow the additive noise model $y = m_i + n$, where m_i is the signal mean under $H_i, i = 0, 1$ and n is the correlated Gaussian noise with zero-mean and the symmetric covariance matrix given in eqn (4.2). The signal means m_0 and m_1 are assumed to be 0 and 1 under H_0 and H_1, respectively.

4.4.1 PGA results

For this problem, we consider both the homogenous and heterogeneous sensor cases, i.e., the cases where the sensors have identical and non-identical thresholds, respectively. We only consider sensor networks consisting of 2 and 3 local sensors since, the order of integration increases with the number of sensors and increases the computational complexity. But, our approach is equally valid for larger number of sensors as well. The number of bits per sensor is assumed to be 2. Thus, each sensor classifies its observation into one of 4 classes. In other words, we consider non-binary local decision rules. The prior probability π_0 is 0.5 for all the cases. The simulations were run for different values of ρ in the covariance matrix. The 3 local thresholds for each sensor are initialized as Gaussian distributed random values with means -0.5, 0.5 and 1.5, respectively. These correspond to the region of overlap of $f_0(y)$ and $f_1(y)$. The variance of all the 3 local thresholds is set to be 0.0025.

Figure 4.1 shows the convergence of the probability of error over 90 generations for different positive values of the correlation coefficient ρ ($\rho = 0, 0.2, 0.5, 0.9$). The local sensors for this case are assumed to be heterogeneous. $\rho = 0$ represents the uncorrelated case, where all the sensor observations are conditionally independent when conditioned on the hypothesis H_i. For this case, the minimum global probability of error goes down to 0.248 after 70 generations. For the case where $\rho = 0.2$, the minimum probability of error is 0.263 after 60 generations. Similarly, for

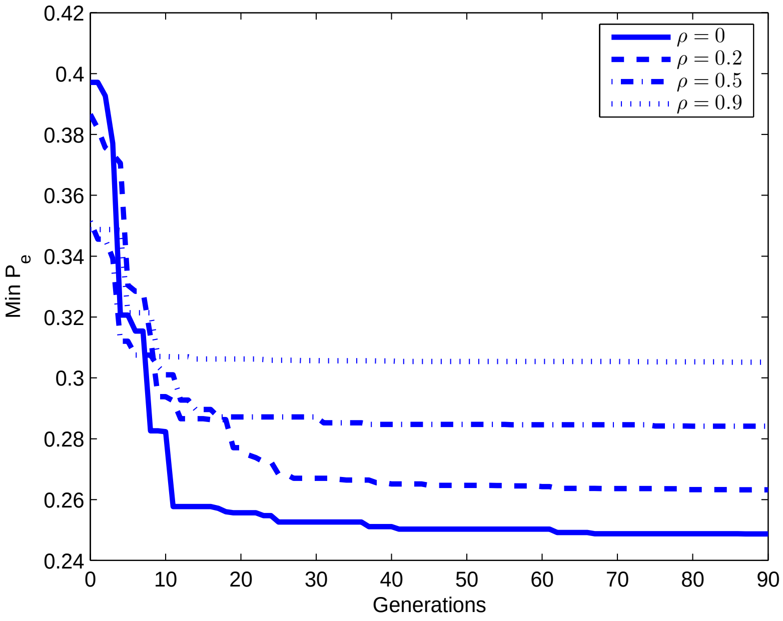

Figure 4.1: Evolution of probability of error - Positive correlation - Non-Homogenous sensors($N = 2, L = 4, \pi_0 = 0.5$)

$\rho = 0.5$ and $\rho = 0.9$, the minimum probability of error converges to 0.284 and 0.305 after 30 and 15 generations, respectively. From this, we can see that the probability of error decreases with decrease in the correlation between the local sensors. This is expected as increasing ρ increases the correlation among the sensor observations, thereby eventually reducing the distributed sensor network to a single sensor system as the correlation coefficient becomes 1. Also, we note that the cases with lower value of the correlation coefficient ρ take longer to converge to the optimum solution as compared to the cases with higher ρ values. This is once again because the higher the correlation, the closer the network becomes to a single sensor system. The optimal local thresholds for heterogenous sensors scenario are listed in Table. 4.1

Figure 4.2 shows a similar plot for different negative correlation coefficient values over 100 generations. Once again, the sensors are heterogeneous quantizers. Here again, we can see the same trend with more negative values of ρ resulting in

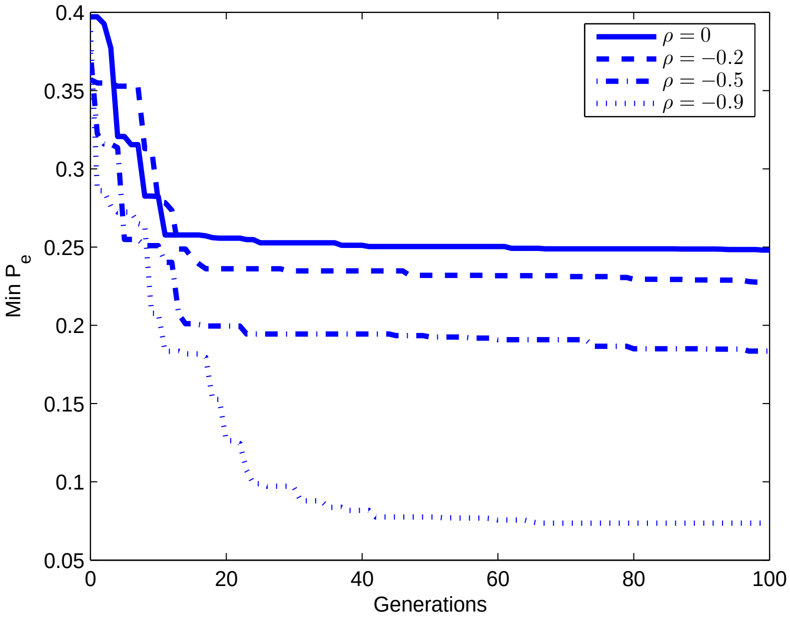

Figure 4.2: Evolution of probability of error - Negative correlation - Non-Homogenous sensors($N = 2, L = 4, \pi_0 = 0.5$)

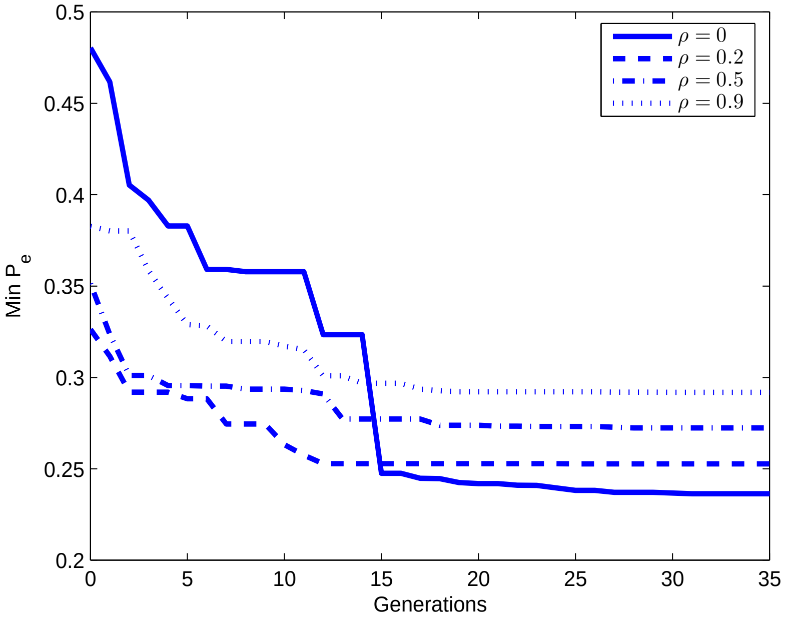

Figure 4.3: Evolution of probability of error - Positive correlation - Homogenous sensors($N = 2, L = 4, \pi_0 = 0.5$)

lower probability of error. The minimum probability of error for the 3 correlated observation cases where $\rho = -0.2, -0.5$ and -0.9 goes down to $0.23, 0.19$ and 0.075, respectively. Thus, we find that there is a drastic improvement in the performance of the sensor network as the negative correlation between the observations increases. An increase in the negative correlation essentially means that if the noise on one sensor is pushes the observation towards the wrong hypothesis, then the noise on the other sensor will push it toward the correct hypothesis. Thus, if one sensor makes an erroneous decision, the chances for the other sensor making the right decision are more. The worst case would be when the magnitude of noise on both the sensors is small. In this case, the performance would be equivalent to the uncorrelated case. Thus, negative correlation on the observations would cause the network to always perform better than or at least equal to the uncorrelated case. Table. 4.2 lists the optimal local thresholds for this case of heterogeneous sensor with negatively correlated observations.

Figure 4.3 shows the convergence plot of the probability of error for the case where the sensors are all homogeneous, i.e., having the same thresholds. We find that the results are similar to the heterogeneous case. But, the algorithm is found to converge much faster in this case. This due to the low complexity of problem since, for the homogeneous case, we need to optimize only one set of common thresholds for the whole network instead of a set of thresholds for each sensor. The GA converges in 35 generations here as opposed to the 90 generations in the heterogeneous case. Table. 4.3 lists the local thresholds for different cases of this scenario.

Figure 4.4 shows the evolution of the probability of error for the 3 sensor heterogeneous case with $\rho = 0.5$. The GA for this case converges after 150 generations and the minimum global probability of error at the end of 150 generations is 0.273, which is less than the minimum probability of error for the 2 sensor case with the same value of ρ. Due to the computational complexity of the 3 sensor case which

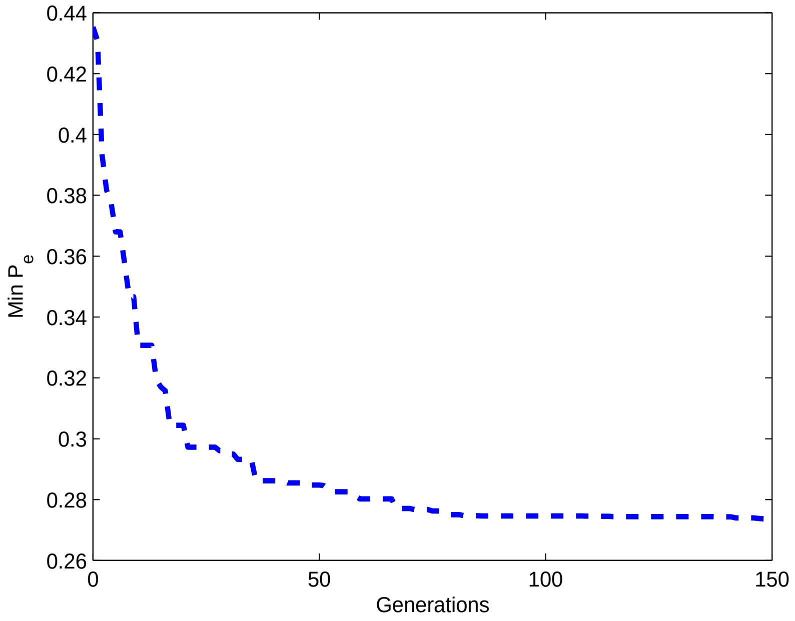

Figure 4.4: Evolution of probability of error - Non-Homogenous sensors($N = 3, L = 4, \pi_0 = 0.5$)

involves a triple integral, we only provide the results for one value of ρ.

The optimal fusion rule in all the above cases was found to converge to a majority-like fusion rule, where the integer sum of all the local decisions is compared to a threshold given by

$$\lambda^0 \simeq \frac{1}{2}N(L-1) \tag{4.7}$$

where N is the number of sensors and L is the number of quantization levels per sensor. Although, all the cases converge to the same fusion rule, there is a degradation in the performance with increasing correlation. The only difference between the cases with varying degrees of correlation is in the local thresholds. Thus, we find that the local thresholds play a major role in determining the performance of the decentralized sensor network when the observations are correlated. We can see from Tables 4.1, 4.2 and 4.3 that as the correlation coefficient increases, the 3 local thresholds of each

	Sensor 1			Sensor 2		
ρ	$\lambda_{1,1}$	$\lambda_{1,2}$	$\lambda_{1,3}$	$\lambda_{2,1}$	$\lambda_{2,2}$	$\lambda_{2,3}$
0	-0.0329	0.2177	1.1636	-0.7661	0.4617	1.6683
0.2	-0.0942	0.7001	1.6162	-0.1284	0.6458	1.6397
0.5	-0.1513	0.6462	1.7067	-0.1436	0.7152	1.6469
0.9	0.1989	0.7523	1.6695	-0.6167	0.5248	1.6571

Table 4.1: Table of optimum local thresholds for different values of ρ for non-homogeneous sensors case with positively correlated observation

	Sensor 1			Sensor 2		
ρ	$\lambda_{1,1}$	$\lambda_{1,2}$	$\lambda_{1,3}$	$\lambda_{2,1}$	$\lambda_{2,2}$	$\lambda_{2,3}$
0	-0.0329	0.2177	1.1636	-0.7661	0.4617	1.6683
-0.2	-0.1658	0.7673	1.8580	-0.2652	0.7204	1.8023
-0.5	-0.2989	0.6892	1.9470	-0.2724	0.7702	1.9656
-0.9	-1.0056	0.3437	1.6285	-0.5272	0.0991	1.3754

Table 4.2: Table of optimum local thresholds for different values of ρ for non-homogeneous sensors case with negatively correlated observations

sensor drift closer to each other.

4.4.2 Fixed rule results

In this section, we present the results obtained by optimizing the local thresholds of the parallel distributed sensor network, shown in Figure 3.4, when the fusion rule is kept fixed. The sensors are assumed to be binary ($L = 2$), heterogeneous (non-identical thresholds) sensors. As mentioned in Section 4.3, we fix the fusion rule to be one of the three common binary fusion rules: (i) Majority voting rule, (ii) AND rule, or (iii) OR rule. We only consider networks consisting of 3 sensors ($N = 3$) since, the order of integration increases with the number of sensors and increases the

ρ	λ_1	λ_2	λ_3
0	-0.5232	0.4906	1.3661
0.2	-0.3809	0.5445	1.3514
0.5	-0.0867	0.6615	1.2830
0.9	-0.8301	0.7861	1.2114

Table 4.3: Table of optimum local thresholds for different values of ρ for homogeneous sensors case

computational complexity. However, the proposed approach is valid for any number of sensors. We use a GA to perform the optimization of the local thresholds. Each chromosome in the GA consists of 3 real threshold values, one corresponding to each sensor. The initial population of the GA is chosen such that all the thresholds are close to the region of overlap between $f_0(y)$ and $f_1(y)$. In our case, this corresponds to a value of 0.5 where the pdf curves of H_0 and H_1 overlap. Roulette wheel selection is used to select parents for cross-over and the cross-over is performed in accordance with eqn (3.11). The mutation rate for the GA is fixed at 0.05.

Figure 4.5, shows the convergence of the probability of error at the fusion center over 200 generations for the three different fusion rules, when the prior probability $\pi_0 = 0.5$ and the correlation coefficient $\rho = 0.5$. We can see from this figure, that the minimum probability of error converges to almost the same value (0.28) for all 3 fusion rules. The majority voting rule, being the best rule, starts out at its minimum and shows no further convergence, while the AND and OR rules converge more slowly after about 150 generations.

Figure 4.6, shows a similar convergence plot of the probability of error at the fusion center over 200 generations for the three different fixed fusion rules, when the prior probability $\pi_0 = 0.6$ and correlation coefficient $\rho = 0.5$. Once again, all three cases converge to almost the same minimum probability of error. However, the

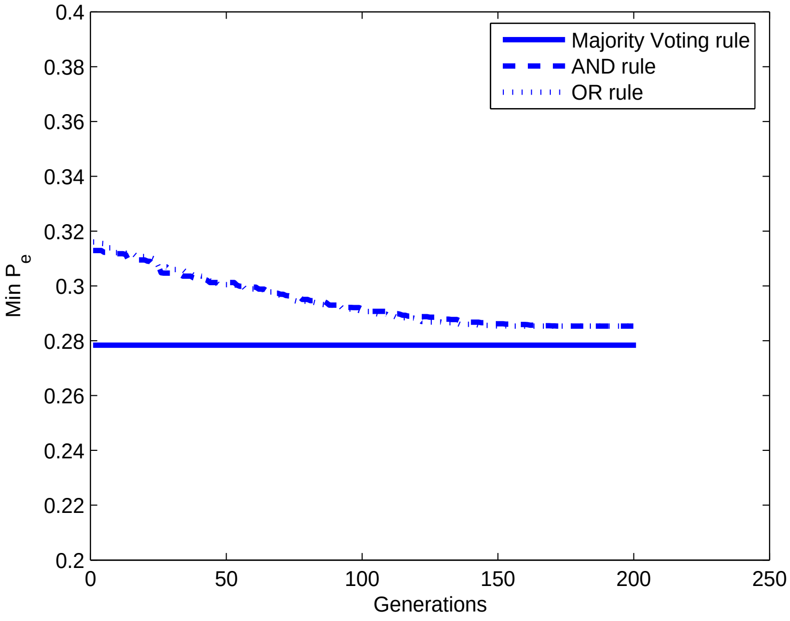

Figure 4.5: Evolution of probability of error with fixed fusion rule($N = 3, L = 2, \rho = 0.5, \pi_0 = 0.5$)

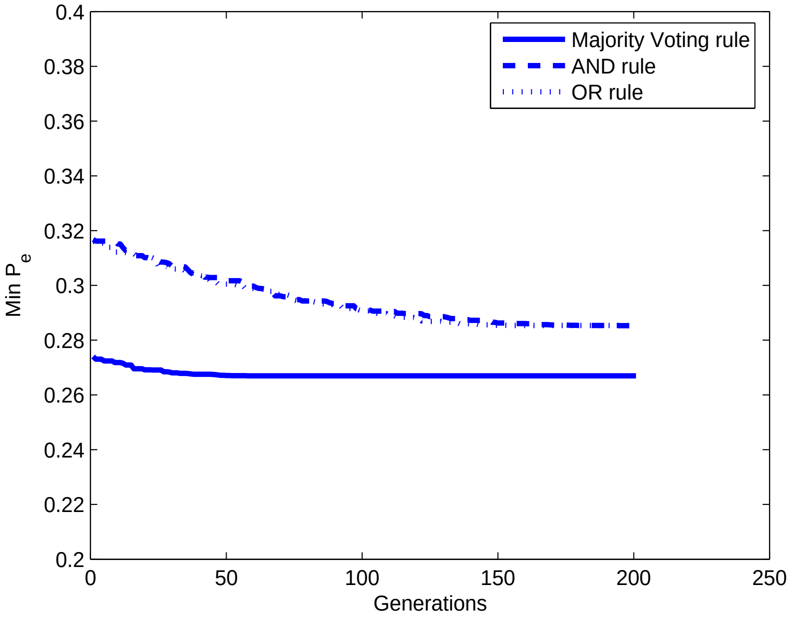

Figure 4.6: Evolution of probability of error with fixed fusion rule($N = 3, L = 2, \rho = 0.5, \pi_0 = 0.6$)

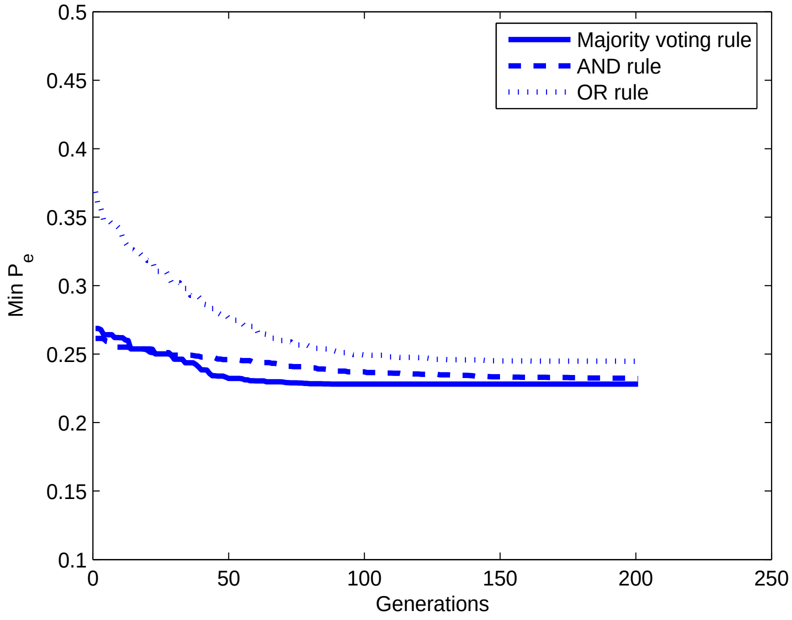

Figure 4.7: Evolution of probability of error with fixed fusion rule($N = 3, L = 2, \pi_0 = 0.6$)

difference between the optimal majority voting rule and, the AND and OR rules is slightly more pronounced in this case. Also, we observe that in this case, the majority voting rule does converge to a lower minimum P_e value than its initial starting point unlike the previous case. This is because the majority voting fusion rule with all the thresholds close to 0.5 is the best combination of fusion rule and local thresholds for $\pi_0 = 0.5$ case, if the optimal local decision rules are assumed to be LRTs, due to the symmetry of the problem. On the other hand, for $\pi_0 = 0.6$, the thresholds have to be moved a little away from the initial 0.5 to minimize the P_e.

Figure 4.7 shows the convergence plot of the probability of error over 200 generations for same prior probability of $\pi_0 = 0.6$, but unlike the equicorrelated noise structure considered in the previous two cases, we have the following covariance matrix

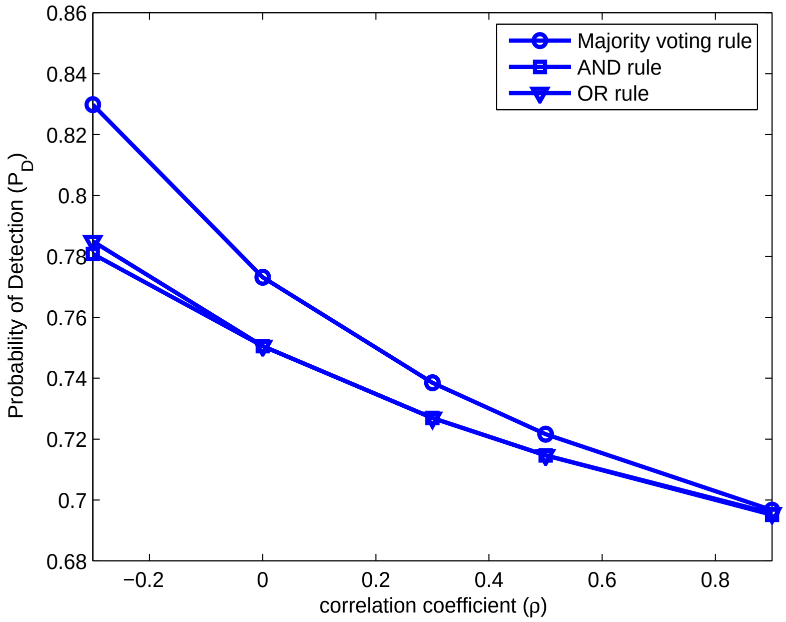

Figure 4.8: Plot of the maximum probability of detection Vs. the correlation coefficient $\rho(N = 3, L = 2, \pi_0 = 0.5)$

$$\Lambda = \begin{pmatrix} 1 & 0.7 & 0.2 \\ 0.7 & 1 & -0.4 \\ 0.2 & -0.4 & 1 \end{pmatrix} \tag{4.8}$$

Once again we observe similar trends with all 3 fusion rules converging to the nearly the same minimum probability of error value. Also, the majority voting rule shows more convergence in this case than in the previous cases.

All the above results serve to illustrate that the local decision rules play a very important role in the overall performance of the distributed sensor network with respect to minimizing the probability of error.

Figure 4.8 shows the plot of the maximum probability of detection, corresponding to the minimum probability of error, with respect to the correlation coefficient ρ for values of ρ ranging from -0.3 to 0.9. The covariance matrix in this case is of the form given in eqn(4.2). From this figure, we can see that the majority voting rule gives the best probability of detection performance, i.e., the least probability of error if the local decision rules are assumed to be LRTs regardless of the value of ρ. But, we also observe that the AND and the OR rules get progressively closer to the majority voting rule as the correlation between the sensor observations increases, with all three rules providing same performance at $\rho = 0.9$. This is to be expected as increasing ρ increases the correlation among the sensor observations, eventually reducing the distributed sensor network to a single sensor system as the correlation coefficient becomes 1. Thus, all the three rules provide similar performance at higher correlation since the system then starts to approximate a single sensor network. Another important observation is that the AND and the OR rules give equivalent performances for all values of ρ. This is contrary to the conjecture made in [29] that the OR rule is inferior to both AND and majority voting rules and is relatively insensitive to changes in the correlation coefficient. Although this may be true if the false alarm probability is fixed at large values, in the more general case, we find that the OR rule is in fact comparable to the AND rule. Our analysis thus demonstrates that, by appropriately choosing the local decision rules, we can obtain equivalent performance from any fusion rule.

4.5 Summary

In this chapter, we analyze the performance of a decentralized sensor network with parallel fusion architecture in the presence of correlated noise. We study this problem in two parts: (i) we simultaneously optimize both the local threshold and the fusion rule using the Parallel Genetic Algorithm approach proposed in chapter 3, (ii) we fix the fusion rule to be one of the three common binary rules (majority voting, AND, and OR rules) and optimize only the local thresholds using a GA. In both cases, the optimization is performed with respect to the average probability of error at the fusion center.

Our results show that the optimal fusion rule for both correlated and uncorrelated observations is a majority-like fusion rule, irrespective of the degree of correlation. We also illustrate that the local decision rule plays a key role in optimizing the sensor network when the observations are correlated. If the local decisions are assumed to be likelihood ratio tests (LRTs) and are defined completely by quantization thresholds, we show that these thresholds drift closer together as the degree of correlation increases. We also demonstrate that both homogeneous and non-homogeneous sensors provide similar probability of error performance. Using the results from the fixed rule analysis, we show that systems having different fusion rules can all give practically the same performance provided the local decision rules are optimized properly. We also show that, contrary to the observation made in [29], the OR and AND rules actually have comparable performance for the more general case when the false alarm rate is not fixed and the overall probability of error is minimized. Also, unlike [29], we show that our approach is valid not only for the equicorrelated observations case but also for any arbitrary positive definite covariance matrix.

Chapter 5

Virtual MIMO Sensor Network

So far in this thesis, we have only considered the data fusion aspects of the decentralized sensor networks. In this chapter, we will analyze the data detection problem in distributed sensor networks, i.e, the problem of communicating the quantized information from the local sensors to the fusion center. We characterize this problem as a virtual multiple input multiple output (MIMO) problem. Specifically, the problem turns out to be an overloaded array problem. We use a genetic algorithm to solve this overloaded array problem.

The organization of this chapter is as follows. In section 5.1, we introduce the reader to MIMO systems and their applications. Section 5.2 details how the distributed sensor network can be characterized as a virtual MIMO system. We present the system model that we use for simulating the virtual MIMO system in section 5.3. In section 5.4 we present our simulation results. Finally, in section 5.5, we present the summary of this chapter .

5.1 Introduction to MIMO systems

Multiple input multiple output (MIMO) systems, as the name suggests, are communication systems with multiple input nodes and multiple output nodes. As shown in figure 5.1, the input and output nodes are generally antennas and thus, these systems are also known as multi-antenna systems. MIMO systems have been shown to support higher data rates under the same power budget and bit error rate constraints. Specifically, they have the potential to dramatically increase the channel capacity of fading channels [39]. Due to this, there has been a lot of interest in these systems and they have been studied extensively in literature [40]- [47]. MIMO systems also offer very high reliability because they exploit spatial diversity.

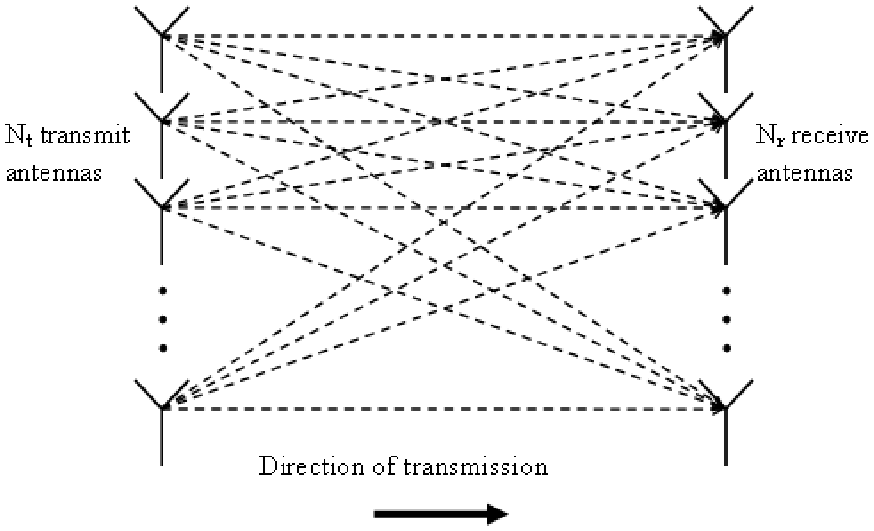

Figure 5.1: Multiple-Antenna MIMO system

Traditionally, spatial diversity has been implemented in wireless communication systems by using antenna arrays on the receiving side and only one transmitter [55]. Such systems are known as single input multiple output or SIMO systems. These systems perform relatively better as compared to single input single output (SISO) systems. However, more recently, MIMO systems where multiple input as well as multiple output nodes are present have been studied extensively. These MIMO systems provide even higher gains and are, therefore, becoming increasingly more popular. MIMO system techniques have also been incorporated into wireless multiple access networks and sensor networks [36]-[38]. In the following section, we will describe how sensor networks can be regarded as virtual MIMO systems.

5.2 Sensor Network as Virtual MIMO

In [36], Tong, et.al. proposed the SENMA (SEnsor Networks with Mobile Agents) architecture, which was a network architecture for low power and large scale sensor networks. These SENMA had two types of nodes: sensors and mobile agents. The mobile agents in SENMA are aerial or ground vehicles that collect data from the sensor nodes and then transmit this data to the fusion center, which could be located miles away from the sensors.

On the other hand, Cui, Goldsmith and Bahai, considered cooperative MIMO techniques in sensor networks in [37]. In such a scheme, the sensors cooperate amongst themselves while transmitting their information to the fusion center or a relay node. This is essentially a MISO (multiple input single output) system. In their work, they analyzed the energy efficiency of such cooperative sensor networks for the MISO, SIMO and the general case of multiple input and multiple output nodes (MIMO).

In [38], Xiao and Xiao proposed a new kind of energy efficient sensor network called the MIMO-Sensor Networks with Mobile Agents (M-SENMA). Like the SENMA system, this system also contains two types of nodes: the local sensors, which are the input nodes and the mobile agents, which are the output nodes. The mobile agents are assumed to be helicopters with N_r receive antennas The sensor nodes are divided into virtual cells and the antennas from N_t sensor nodes in each cell are used to cooperatively transmit the same amount of information to the N_r antennas of the mobile agents. Thus, the transmission for each cell can be viewed as an $N_t \times N_r$ MIMO system. The analysis in [38] was also based on energy efficiency.

Other ways of modeling sensor networks as virtual MIMO systems include considering that the N_t local sensors of the distributed sensor network are the input nodes and, there are N_r intermediate nodes which act as the output nodes. These intermediate nodes could be either relay nodes or mobile agents which then convey the information from the sensors to the fusion center. If there are no intermediate nodes, the system reduces to a MISO system. Alternatively, we could have multiple fusion centers and no intermediate nodes, which again results in a MIMO system.

Thus, there are many ways of modeling sensor networks as virtual MIMO systems. In this work, we consider a distributed sensor network with N_t local sensors which transmit their quantized information to a fusion center, which has N_r receive antennas. This is a virtual MIMO with N_t input nodes and N_r output nodes. We will take a closer look at this model and the algorithm that we use at the receiver (fusion center) to detect the information from the N_t transmitters (sensors) in the next section.

5.3 System Model and Algorithm Description

As mentioned in the previous section, we model our distributed sensor network as a virtual MIMO system with the N_t local sensors acting as the input nodes and, the N_r receive antennas at the fusion center acting as the output nodes (see figure 5.2). We analyze the bit error rate (BER) performance of this virtual MIMO system. We assume that there is perfect cooperation among the sensors while transmitting their quantized information to the intermediate nodes allowing us to treat them as multiple antennas to the destination nodes. We do not concern ourselves with the power or delay considerations in our system. We can thus ignore the loss in power and delay that may result due to such cooperation.

The local sensors transmit their decisions after appropriate coding to the fusion center. If the local decisions are binary, binary phase shift keying (BPSK)

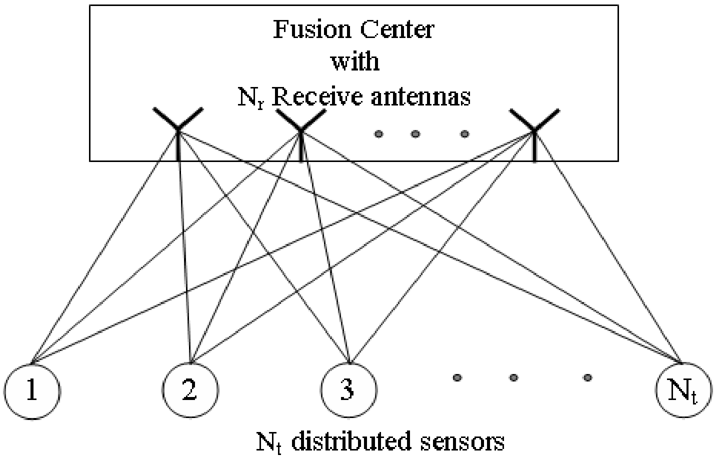

Figure 5.2: Virtual MIMO sensor network

scheme can be used for coding. If the local sensors classify their decisions into 4 classes, then quadrature phase shift keying (QPSK) could be used, and so on. The signals from the N_t local sensors are received by the N_r receive antennas at the fusion center with no additional interference except for the interference among the transmitted signals. The vector of transmitted signals is denoted by $\mathbf{s}(k)$ and the narrow-band complex channel between the sensors and the fusion center is given by the known $N_r \times N_t$ channel matrix $\mathbf{H}(k)$. The vector of the received signal at the fusion center, sampled at intervals kT, can then be written as

$$\mathbf{x}(k) = \mathbf{H}(k)\mathbf{s}(k) + \mathbf{n}(k) \tag{5.1}$$

where $\mathbf{n}(k)$ represents the sampled complex noise vector. We assume that the entries of the channel matrix are independent identically distributed complex gaussian random variables.

The spatial filter that maximizes the signal to noise ration (SNR) at it output is given by

$$\hat{\mathbf{s}}(k) = \mathbf{W}^H(k)\mathbf{x}(k) \tag{5.2}$$

where $\mathbf{W}(k)$ is the pseudo-inverse of $\mathbf{H}(k)$ and the superscript ^{H} represents the Hermitian transpose. In our model, the number of input nodes (sensors) N_t is greater than the number of intermediate nodes N_r. In such a case, spatial filtering will not remove all the interfering signals and results in very high error floors. This is because the number of variables is greater than the number of equations. In MIMO terminology, such a condition is known as an overloaded array problem [48]. The optimal receiver for such an overloaded system would be a maximum likelihood joint detector (MLJD) receiver. Unfortunately, the MLJD receiver has complexity on the order of $\mathcal{O}(2^{N_t}\log_2 M)$, where M represents the constellation size. Thus, the complexity of the MLJD receiver increases manifold with an increase in the number of sensors N_t.

In [48], Colman and Willink proposed the use of genetic algorithms (GAs) to solve the problem of overloaded arrays. In this work, we adopt their genetic algorithm to our virtual MIMO sensor network in order to find a good estimate of the transmitted signal vector, $\tilde{\mathbf{s}}(k)$ at the fusion center. This is done as follows. First, we generate a random initial population consisting of a fixed number (P) of candidate solutions (chromosomes), $\hat{\mathbf{s}}(i)$, where i is the index of each solution. Then we evaluate the fitness of each chromosome using the fitness function

$$f(k) = \|\mathbf{x}(k) - \mathbf{H}(k)\tilde{\mathbf{s}}(k)\| \tag{5.3}$$

where $\mathbf{x}(k)$ is the received signal vector and $\|$ represents norm of the vector. The good solutions are the ones with low value of this $f(k)$, since it represents the distance between the candidate solution and the actual solution. After evaluating the fitness of all the chromosomes, $T\%$ of the chromosomes having the least fitness values are selected for crossover. From this 'mating pool', we pick two parents at a time using roulette wheel selection and perform crossover. Crossover is performed by generating a random crossover mask and exchanging bits between the two parents in the places that correspond to 1 in the mask. Thus, each crossover generates two offsprings. These offspring form part of the next generation. Also, in each generation, we mutate each bit in each solution with probability P_m. Mutation is defined here as flipping the sign of the symbol in the case of QPSK and flipping between 0 and 1 in the case of BPSK. Elitism is also introduced to ensure that the best solutions from each generation survive to the next generation. We do this as follows. The best $T_e\%$ of chromosomes from each generation are passed on to the next generation without any mutation. Once, the population for the next generation is assembled, the same process continues. The GA is terminated after a fixed number of generations. We present the results obtained from the GA in the next section.

5.4 Results

We consider a virtual MIMO system consisting of N_t input nodes and N_r output nodes (figure 5.2). We consider that the transmitted vector $\mathbf{s}(k)$ consists of QPSK symbols. We assume perfect knowledge of the channel matrix $\mathbf{H}(k)$. We also assume that the noise is i.i.d complex gaussian. We use a genetic algorithm to get the best estimate of the transmitted signal vector $\mathbf{s}(k)$ at the receiver. The mutation rate was chosen to be 0.1 for all the simulations. The N_t and N_r values were set at 8 and 5 respectively.

For the first set of simulations, we plot the bit error rate (BER) versus the number of fitness evaluations. In order to do this, we evaluate the BER after every generation and consider the number of fitness evaluations in each generation to be equal to the population size. Thus, we place a marker of each BER value in the position corresponding to population size on the X axis.

Figure 5.3 shows the plot of BER versus the number of fitness evaluations

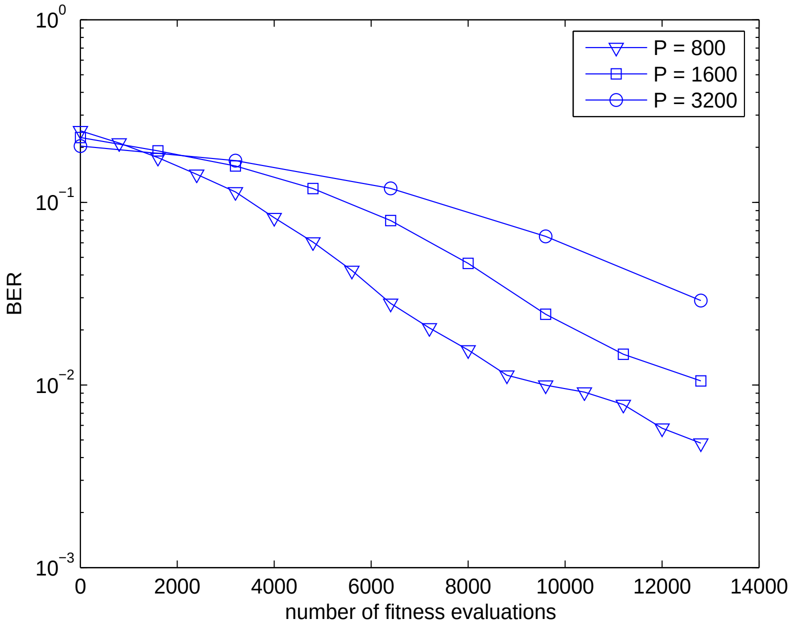

Figure 5.3: BER versus number of fitness evaluations for GA with $T = 50\%$

for the GA with mating pool containing $T = 50\%$ of the population, for population sizes of $P = 800$,1600 and 3200 at an SNR of 33.8 dB. The elite size was set at $T_e = 5\%$ of the population size. Thus, best 5% of the total population in each generation survives to the next generation. For these particular parameters, the best performance is obtained for $P = 800$ populations size and the minimum BER value is around 4×10^{-3}.

Figure 5.4 shows the BER versus number of fitness evaluations plot for GA with mating pool containing $T = 25\%$ of the population, once again for population sizes of $P = 800$,1600 and 3200. All other parameters are the same as in the previous simulation. Now, we observe that the population size of $P = 1600$ gives the best performance with a BER value of 1×10^{-3}. Also, we notice that the performance with $P = 3200$ has also improved compared to the previous simulation.

Figure 5.5 shows the same BER vs. number of fitness evaluations plot for mating pool size of $T = 15\%$ of the population size. As before, the rest of the

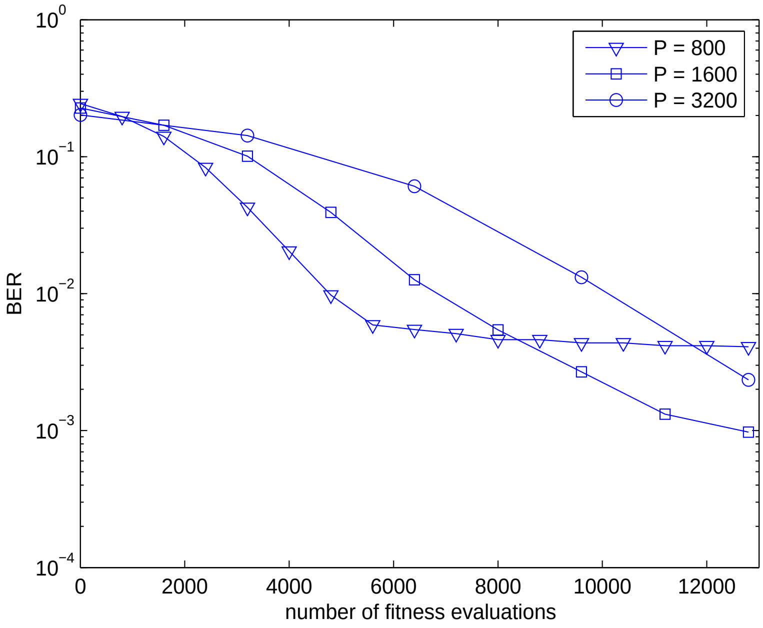

Figure 5.4: BER versus number of fitness evaluations for GA with $T = 25\%$

parameters are kept fixed at the same values as in the first simulation. For this case, we find that both the $P = 1600$ and $P = 3200$ population sizes result in nearly the same performance with the BER going down to 9×10^{-4}. The population size of 800 chromosomes, on the other hand, stays the same as in the previous case. Thus, decreasing the size of the mating pool affects the performance of the GA. But, the population size is also a factor in determining the effect on the performance.

Finally, in last simulation in this set, we set the mating pool size to be $T = 5\%$ of the population size. For this case, we chose only one elite chromosome from each generation to survive on to the next generation. All other parameters are the same as before. The results of this simulation are shown in figure 5.6. In this case, we observe that the population size of $P = 3200$ performs best and the GA for this case converges to 9×10^{-4} as in the previous case. We find that there is a pattern among the four simulations that we run. As the mating pool size is decreased, the GAs with bigger population sizes perform better. But, we find that $T = 15\%$ and

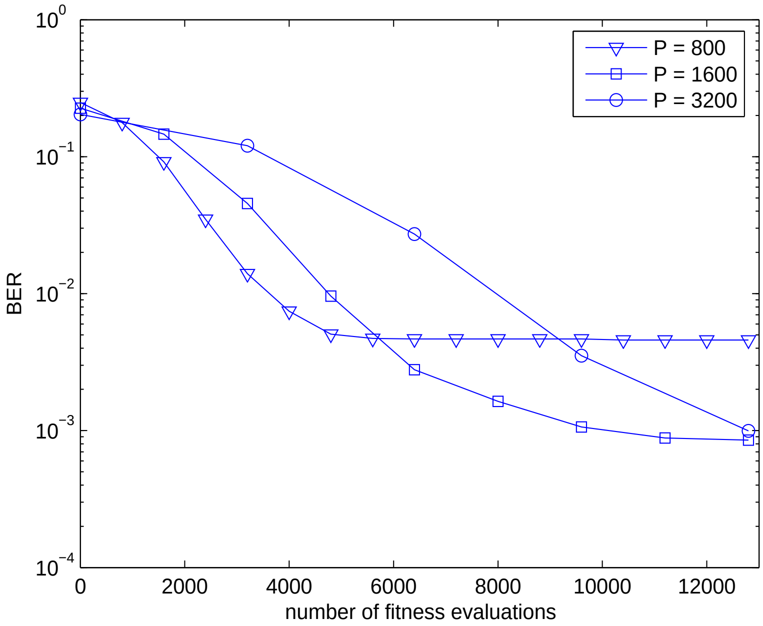

Figure 5.5: BER versus number of fitness evaluations for GA with $T = 15\%$

$T = 5\%$ mating pool sizes give the best performance with the algorithm in both cases converging to the same BER value of 9×10^{-4}.

For the next set of simulations, we fix the mating pool size to be $T = 15\%$ since, we found this to be optimal from the previous set of simulations. We now plot the BER versus the signal to noise ratio (SNR) curves for different number of generations. The population size is fixed at $P = 1600$ and the elite size is set to be $T_e = 5\%$ of the population size. Figure 5.7 show this BER vs. SNR plot for two different generations: $G = 4$ and $G = 8$. From this, we can see that as expected the BER performance of the system improves with increasing SNR. However, the BER curve hits an error floor after a certain value of SNR and the BER performance of the system does not improve significantly after this value os SNR. For the $G = 4$ case, the error floor is reached very early around SNR of 12 dB and BER of the order of 10^{-2}. On the other hand, the $G = 8$ case performs better as expected and the error floor for this case is reached after 18 dB SNR and BER of the order of 10^{-3}.

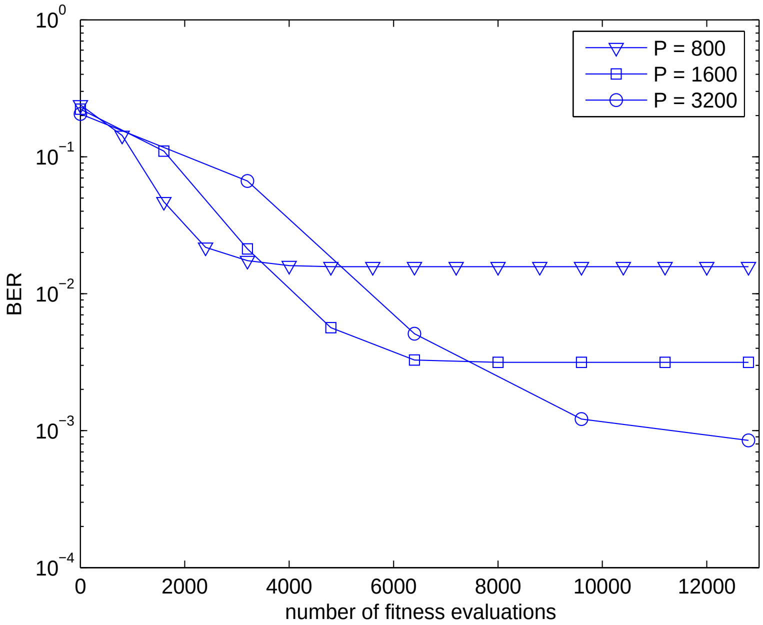

Figure 5.6: BER versus number of fitness evaluations for GA with $T = 5\%$ and only one elite

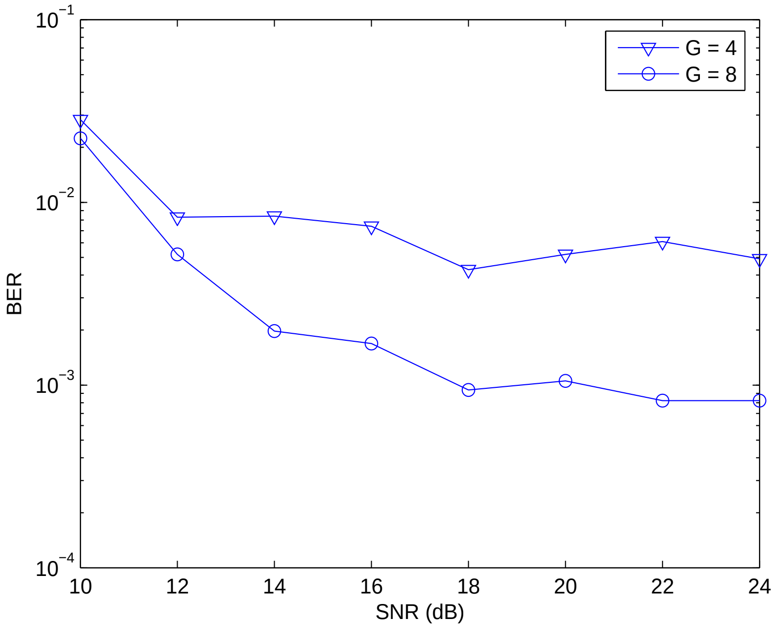

Figure 5.7: BER versus SNR for GA with $T = 15\%$ and $P = 1600$

5.5 Summary

In this chapter, we analyze the data detection aspect of decentralized sensor networks, which is the communication problem of transmitting the quantized information from the local sensors to the fusion center. We model our decentralized sensor network as a virtual MIMO system with N_t local sensors on the transmitting side and a N_r receive antenna array at the fusion center. Due to the greater number of sensors as compared to receive antennas, our virtual MIMO system turns out to be an overloaded MIMO system. We use a genetic algorithm to solve this overloaded array problem. We show that the convergence of the GA to the optimal BER value for a fixed SNR depends on the size of the mating pool and the population size of our GA. For fixed population and mating pool size, we show that the BER value reduces with increase in SNR. But, after a certain SNR value (18dB for GA with 8 generations and 12dB for GA with 4 generations), the BER does not decrease further and we hit an error floor. This is due to the overloaded nature of the MIMO problem. Thus, we demonstrate that a decentralized sensor network can be modeled as a virtual MIMO system and that a genetic algorithm can be used to estimate the quantized information transmitted by the local sensors at the fusion center.

Chapter 6

Conclusion

This chapter concludes this thesis by summarizing and highlighting the key contributions of this research. We also present areas of potential future work.

6.1 Summary of Key Contributions

In this thesis, we analyzed decentralized sensor networks in the presence of additive white gaussian noise and additive correlated gaussian noise. Traditionally, most of the research on decentralized sensor networks tend to focus only on the statistical hypothesis testing and optimal combining of the information from all the sensors. In this work, we study both this data fusion aspect and also the communication or data detection aspect of the problem.

This work first investigated decentralized sensor networks when the sensor observations are assumed to be conditionally independent when conditioned on the true hypothesis. A novel parallel genetic algorithm (PGA) was proposed which optimizes both the local decision rules and the fusion rule simultaneously. The results obtained using this approach were compared to those obtained using the GA-SG approach in [25]. We demonstrated that our PGA approach converges to the same optimal majority-like fusion rule and minimum probability of error as the GA-SG algorithm with greatly reduced computational cost. Our results also show that our algorithm converges to the same optimal solution for networks with homogeneous as well as those with heterogeneous sensors. We also demonstrated the importance of the initial population of the GA in the convergence of the algorithm. Our results show that the local thresholds have to be initialized close to the region of overlap of the distributions of the two hypotheses for proper convergence of the algorithm. Similarly, the initial population of fusion rules must be chosen such that the probability of getting a 1 in the rule increases as we move from the least significant bit to the most significant bit.

Next, we considered decentralized sensor networks in the presence of correlated observations. We first optimized both the local decision rules and fusion rule of the decentralized network in the presence of additive correlated gaussian noise by using the PGA approach that we proposed earlier. Our results show that the optimal fusion rule for the decentralized sensor network in the presence of additive gaussian noise is a majority-like fusion rule irrespective of the degree of correlation of the noise. Another important finding is that the local decision rules play an important role in the performance of the sensor network. When the local decisions were assumed to be likelihood ratio tests (LRTs) and were defined completely by quantization thresholds, we found that these thresholds drift closer together as the degree of correlation increases. Also, we observed that the performance of both homogeneous and heterogeneous sensors is identical even for the case of correlated observations. We also studied the impact of the local decision rules on the performance of the decentralized sensor network. In order to do this, we fixed the fusion rule to be one of the three common binary decision rules (AND, OR and majority voting rules) and optimized only the local decision rules using a GA. This provided us with the interesting result that systems having different fusion rules can all give equivalent performance provided the local decision rules are chosen appropriately. Also, we demonstrate that, contrary to the observation made in [29], the OR and AND rules provide comparable performance for the more general case when the false alarm rate is not fixed and the overall probability of error is minimized. We also show that our approach is valid for any arbitrary positive definite covariance matrix and not just for the equicorrelated observations case.

Finally, we considered the data detection problem in decentralized sensor network, i.e., the communication aspect of transmitting the quantized information from the sensors to the fusion center. We modeled our decentralized sensor network as a virtual MIMO system where the sensors represent the input nodes and the receive antenna array at the fusion center represent the output nodes. This modeling resulted in an overloaded array problem which we solved using a genetic algorithm. We demonstrated that the convergence of the GA to the optimal BER value for a fixed SNR depends on the population size and the mating pool size in the GA. We also illustrated that although the BER decreases with increase in SNR, for a fixed population and mating pool size, this BER vs. SNR curve hits an error floor after a certain value of SNR. The BER performance of the system does not improve any further with increase in SNR. This is due to the overloading of the MIMO system.

6.2 Future Work

Some possible future work based on the work in this thesis is provided in this section.

- In this work, we have considered both the data detection and data fusion aspects of decentralized sensor networks. However, our analysis of these two problems has been carried out independent of each other. In future, these two aspects can be integrated to form one model which incorporates the communication aspect into the data fusion problem.

- In this thesis, we only considered the gaussian location testing problem for the data fusion analysis. It would be an interesting exercise to consider other common detection problems (for instance, problems with exponentially distributed observations) and determine the optimal fusion and local rules for such problems.

- For the data detection problem, we have used a Rayleigh channel model with no multi-path. It would be instructive to assume a multi-path fading channel and check if the GA still provides good results.

- In this work, we have assumed synchronous transmission from the sensors. It is a natural extension to use an asynchronous model and evaluate the system performance under such conditions.

- Another interesting study would be to compare the performance results obtained with the PGA and the theoretical results obtained under asymptotic assumptions (i.e., when the number of sensors $N \to \infty$).

References

[1] H. V. Poor, *An Introduction to Signal Detection and Estimation*. Springer-Verlag, 1988.

[2] C. S. Raghavendra(Editor), K. M. Sivalingam(Editor), and T. Znati(Editor), *Wireless Sensor Networks*. Kluwer Academic Publishers, 2004.

[3] J. E. H. Callaway, *Wireless Sensor Networks: Architectures and Protocols*. Auerbach Publications, 2003.

[4] Intel Technologies, "Intel research: Vibration monitoring." www.intel.com/research/exploratory/wireless_sensors.htm.

[5] R. Allan, "Wireless sensors land anywhere and everywhere." http://www.elecdesign.com/Articles/ArticleID/10710/10710.html.

[6] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, p. 393422, 2002.

[7] M. Gaynor, S. L. Moulton, M. Welsh, E. LaCombe, A. Rowan, and J. Wynne, "Integrating wireless sensor networks with the grid," *IEEE Internet Computing*, vol. 8, no. 4, pp. 32 – 39, 2004.

[8] Intel, "Vineyard smart agriculture:intel researchers use wireless sensor motes to help farmers make better wine." http://www.intel.com/research/vert_agri.htm.

[9] P. K. Varshney, *Distributed Detection and Data Fusion.* New York: Springer-Verlag, 1996.

[10] S. S. Iyengar, R. L. Kashyap, and R. N. Madan, “Distributed sensor networks – introduction to the special section,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. 21, pp. 1027 – 1031, Sept. 1991.

[11] R. Viswanathan and P. K. Varshney, “Distributed detection with multiple sensors: Part i - fundamentals,” in *Proceedings of the IEEE*, vol. 85, pp. 54 – 63, Jan. 1997.

[12] J. N. Tsitsiklis, “Decentralized detetcion,” in *Advances in Statistical Signal Processing, Signal Detection* (H. V. Poor and J. B. Thomas, eds.), vol. 2, Greenwich, CT: JAI, 1993.

[13] J. N. Tsitsiklis and M. Athans, “On the complexity of decentralized decision making and detection problems,” *IEEE Transactions on Automatic Control*, vol. 30, pp. 440–446, 1985.

[14] M. Barkat and P. K. Varshney, “Decentralized cfar signal detection,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 25, pp. 141 –149, Mar. 1989.

[15] W. W. Irving and J. N. Tsitsiklis, “Some properties of optimal thresholds in decentralized detection,” *IEEE Transactions on Automatic Control*, vol. 39, pp. 835–838, 1994.

[16] A. M. Aziz, M. Tummala, and R. Cristi, “Optimal data fusion strategies using multiple-sensor detection systems,” in *Conference Record of the Thirty-First Asilomar Conference on Signals, Systems & Computers*, vol. 1, pp. 941 – 945, 2-5 Nov. 1997.

[17] W. Liu, Y. Lu, and J. Fu, "Data fusion of multiradar system by using genetic algorithm," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, pp. 601 – 612, April 2002.

[18] J. N. Tsitsiklis, "Decentralized detection by a large number of sensors," *Mathematics of Control, Signals, and Systems*, vol. 1, no. 2, pp. 167 – 182, 1988.

[19] J. F. Chamberland and V. V. Veeravalli, "Decentralized detection in sensor networks," *IEEE Transactions on Signal Processing*, vol. 51, pp. 407 – 416, Feb. 2003.

[20] S. E. Aldosari and J. M. F. Moura, "Detection in decentralized sensor networks," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 277–80, May 2004.

[21] N. Mansouri and M. Fathi, "Simple counting rule for optimal data fusion," in *Proceedings of IEEE Conference on Control Applications*, vol. 1, pp. 1186 – 1191, June 2003.

[22] P. F. Swaszek and P. Willett, "Parley as an approach to distributed detection," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 31, pp. 447 – 457, Jan. 1995.

[23] S. C. A. Thomopoulos, R. Viswanathan, and D. K. Bougoulias, "Optimal distributed decision fusion," *IEEE Transactions on Aerospace Electrical Systems*, vol. 25, pp. 761–765, Sept. 1989.

[24] Z. B. Tang, K. R. Pattipati, and D. L. Kleinman, "An algorithm for determining the decision thresholds in a distributed detection problem," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 21, pp. 231 – 237, Jan/Feb 1991.

[25] S. A. Aldosari and J. M. F. Moura, "Fusion in sensor networks with commu-

nication constraints," in *Proceedings IEEE/ACM Symposium on Information Processing in Sensor Networks (IPSN'04)*, pp. 108–115, April 2004.

[26] N. Gnanapandithan and B. Natarajan, "Parallel genetic algorithm based optimal fusion in sensor networks," in *Proceedings of IEEE Consumer Communications and Networking Conference Special session*, Accepted for Publication, Jan. 2006.

[27] G. S. Lauer and N. R. Sandell, "Distributed detection of known signal in correlated noise." ALPHATECH, Burlington, MA, Rep., Mar. 1982.

[28] E. Geraniotis and Y. A. Chau, "Distributed detection of weak signals from multiple sensors with correlated observations," in *Proceedings of the IEEE Conference on Decision and Control*, vol. 3, pp. 2501 – 2506, Dec. 1988.

[29] V. Aalo and R. Viswanathan, "On distributed detection with correlated sensors: two examples," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 25, pp. 414–421, May 1989.

[30] E. Drakopolous and C. C. Lee, "Optimum multisensor fusion of correlated local decisions," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 27, pp. 593–606, July 1991.

[31] V. Aalo and R. Viswanathan, "Asymptotic performance of a distributed detection system in correlated gaussian noise," *IEEE Transactions on Signal Processing*, vol. 40, pp. 211–213, Jan. 1992.

[32] M. Kam, Q. Zhu, and W. S. Gray, "Optimal data fusion of correlated local decisions in multiple sensor detection systems," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 28, pp. 916–920, July 1992.

[33] R. S. Blum and S. A. Kassam, "Optimum distributed detection of weak signals in dependent sensors," *IEEE Transactions on Information Theory*, vol. 38, pp. 1066 – 1079, May 1992.

[34] N. Gnanapandithan and B. Natarajan, “Analysis of the performance of decentralized sensor network with correlated observations,” in *Proceedings of IEEE International Conference on Communications*, Submitted for Publication, June 2006.

[35] N. Gnanapandithan and B. Natarajan, “Impact of local decision rules in distributed sensor networks,” in *Proceedings of IEEE Vehicular Technology Conference*, Submitted for Publication, May 2006.

[36] L. Tong, Q. Zhao, and S. Adireddy, “Sensor networks with mobile agents,” *Proceedings of IEEE Military Communications Conference*, vol. 1, pp. 688 – 693, Oct. 2003.

[37] S. Cui, A. J. Goldsmith, and A. Bahai, “Energy-efficiency of mimo and cooperative mimo in sensor networks,” *IEEE Journal on Selected Areas in Communications*, vol. 22, pp. 1089 – 1098, Aug. 2004.

[38] L. Xiao and M. Xiao, “A new energy-efficient mimo-sensor network architecture m-senma,” *IEEE Vehicular Technology Conference*, vol. 4, pp. 2941 – 2945, Sept. 2004.

[39] A. Paulraj, R. Nabar, and D.Gore, *Introduction to Space-Time Wireless Communications.* Cambridge, U.K.: Cambridge Univ. Press, 2003.

[40] A. J. Goldsmith and P. P. Varaiya, “Capacity of fading channels with channel side information,” *IEEE Transactions on Information Theory*, vol. 43, pp. 1986 – 1992, Nov. 1997.

[41] G. J. Foschini and M. J. Gans, “On limits of wireless communications in a fading environment when using multiple antennas,” *Wireless Personal Communications*, vol. 6, p. 311335, 1998.

[42] D. Ikonomou and L. Vandendorpe, “A fractionally spaced df equalisation scheme performing joint detection for mc-cdma transmissions,” *IEEE Vehicular Technology Conference*, vol. 2, pp. 1150 – 1154, 1999.

[43] I. E. Telatar, “Capacity of multi-antenna gaussian channels,” *European Transactions on Telecommunication*, vol. 10, p. 585595, 1999.

[44] H. Wang, Z. Li, and J. Lilleberg, “Equalized parallel interference cancellation for mimo mc-cdma downlink transmissions,” in *14th IEEE Proceedings on Personal, Indoor and Mobile Radio Communications*, vol. 2, pp. 1250 – 1254, 2003.

[45] Y. L. C. de Jong and T. J. Willink, “Iterative trellis search detection for asynchronous mimo systems,” *IEEE Vehicular Technology Conference*, vol. 1, pp. 503 – 507, Oct. 2003.

[46] A. van Zelst and T. C. W. Schenk, “Implementation of a mimo ofdm-based wireless lan system,” *IEEE Transactions on Signal Processing*, vol. 52, pp. 483 – 494, Feb. 2004.

[47] L. S. Pillutla and S. K. Jayaweera, “Mimo capacity of an ofdm-based system under ricean fading,” *IEEE Vehicular Technology Conference*, vol. 2, pp. 618 – 622, May 2004.

[48] G. W. K. Colman and T. J. Willink, “Genetic algorithm assisted array processing in overloaded systems,” *IEEE Vehicular Technology Conference*, vol. 1, pp. 261 – 265, Sept. 2004.

[49] A. R. Reibman, *Performance and fault-tolerance of distributed detection networks.* Ph.d. dissertation, Department of Electrical Engineering, Duke University, Durham, NC, 1987.

[50] L. L. Scharf, *Statistical Signal Processing: Detection, Estimation, and Time Series Analysis.* Addison-Wesley, Reading, MA, 1991.

[51] J. H. Holland, *Adaptation in Natural and Artificial Systems.* Ann arbor University of Michigan Press, Ann Arbor, MI, 1975.

[52] H. V. Poor and J. B. Thomas, "Memoryless discrete-time detection of a constant signal in m-dependent noise," *IEEE Transactions on Information Theory*, vol. IT-25, pp. 54–61, Jan. 1979.

[53] D. R. Halverson and G. L. Wise, "A detection scheme for dependent noise sources," *Journal of the Franklin Institute*, vol. 309, pp. 287–300, 287-300May 1980.

[54] H. V. Poor, "Signal detection in the presence of weakly dependent noise-part i: optimum detection," *IEEE Transactions on Information Theory*, vol. IT-28, pp. 735–744, Sept. 1982.

[55] M. S. Alouini and A. J. Goldsmith, "Capacity of rayleigh fading channels under different adaptive transmission and diversity-combining techniques," *IEEE Transactions on Vehicular Technology*, vol. 48, p. 1165 1181, July 1999.