NEURAL NETWORK BASED DIRECTION OF ARRIVAL ESTIMATION

A DISSERTATION

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

MASTER OF TECHNOLOGY

in

ELECTRONICS AND COMMUNICATION ENGINEERING

(With Specialization in RF & Microwave Engineering)

By

KUSHMANDA SAURAV

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

JUNE, 2011

CANDIDATE'S DECLARATION

I hereby declare that the work, which is being presented in the dissertation entitled, ***"Neural Network Based Direction of Arrival Estimation "***, which is submitted in the partial fulfillment of the requirements for the award of degree of ***Master of Technology*** in ***RF & Microwave Engineering***, submitted in the Department of Electronics and Computer Engineering, ***Indian Institute of Technology***, Roorkee (INDIA), is an authentic record of my own work carried out under the guidance of **Dr. A. Patnaik**, Assistant Professor, Department of Electronics and Computer Engineering, Indian Institute of Technology, Roorkee.

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

Kushmanda Saurav

Date: 30 June, 2011 **(Kushmanda Saurav)**

Place: Roorkee **Enrol. No: 09533006**

SUPERVISOR'S CERTIFICATE

This is to certify that this dissertation entitled, ***"Neural Network Based Direction of Arrival Estimation,"***, which has been submitted by Kushmanda Saurav is record of his own work carried out by him under my supervision. I also certify that the above statement made by the candidates is correct to the best of my knowledge and belief.

Date: 30 June, 2011
Place: Roorkee

(Dr. A. PATNAIK)
Assistant Professor
E&CE Department
IIT – Roorkee
Uttaranchal – 247667

ACKNOWLEDGMENT

It has been a great privilege to be an M.Tech. student in the Electronics and Computer Engineering Department at IIT, Roorkee and work closely with my advisor **Dr. A. PATNAIK,** Assistant Professor, Department of Electronics and Computer Engineering, IIT Roorkee. His keen insight into soft computing techniques is the key factor in the completion of the project. I express my deepest gratitude to him for his valuable guidance, support and motivation in my work. I have deep sense of admiration for his boundless interest and readiness to help me. The valuable discussions and suggestions with him have helped me a lot in supplementing my thoughts in the right direction for attaining the desired objective.

I thank research scholar **Mr. Om Prakash Acharya,** for his valuable support and time to time guidance in technical issues, which was instrumental in completing this dissertation work..

Kushmanda Saurav

Enroll. No: 09533006

ABSTRACT

In order to reduce the time of computation and to mitigate the problem of array failure, the direction of arrival problem have been approached here using neural network technique, instead of analytical techniques. In the first phase of work a NN-based DoA estimator was designed for switched beam antenna system which can be applied in DS-CDMA. In this we found the SoI AoA in the presence of a varying number of mobile users and a varying SoI to MAI ratio.

In the second phase of the work the problem of multiple source tracking was accomplished with a NN-based smart antenna which is robust to sensor failure condition also. A 16 element uniform array was taken as the test antenna to validate the developed technique under different failure condition. The performance of the developed model was tested for multiple source directions.

ACRONYMS

AoA	Angle Of Arrival
DoA	Direction Of Arrival
SoI	Signal Of Interest
MAI	Multiple Acess Interference
SeIR	Signal to Interference Ratio
NN	Neural Network
SBS	Swtched Beam System
MUSIC	Multiple Signal Classification
MVDR	Minimum Variance Distortionless Response
LP	Linear Prediction
N-MUST	Neural Multiple Source Tracking
DS-CDMA	Direct Sequence Code Division Multiple Access
SLL	Side lobe level
CL	Crossover Level
HPBW	Half Power Beam Width
MLP	Multi layer Perceptron
RMSE	Root Mean Squae Error

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. Introduction

1.1 Motivation and Scope

Direction of arrival (DoA) denotes the direction from which usually a propagating wave arrives at a point, where a set of sensors are located. The set of sensors forms what is called a sensor array. DoA estimation can comprise of estimation of various parameters such Angle of Arrival (AoA), Time Difference of Arrival and Frequency Difference of Arrival. Frequently, In the literature the terms DoA and AoA are used interchangely. We have also adopted the same strategy here. The present discussion is devoted to AoA estimation. DoA estimation finds application in many fields of engineering such as acoustic signal processing, track and scan RADAR, radio astronomy and radio telescopes, and mostly in cellular systems like W-CDMA and UMTS.

Often the associated technique of DoA estimation is beamforming. Beamforming is the method used to create the radiation pattern of the antenna array by adding constructively the phases of the signals in the direction of the targets desired, and nulling the pattern of the targets that are undesired or interfering targets. Thus DoA estimation of signals impinging on an antenna array is a very important issue in the areas where beamforming is used. Beamforming has a greater role in wireless communication. Compared with the spatial diversity techniques, beamforming is preferred in terms of complexity. On the other hand beamforming in general has much lower data rates. In multiple access channels (CDMA, FDMA, TDMA), beamforming is necessary and sufficient. In order to accomplish tracking of desired users, direction finding algorithms are used to locate the position of the mobile users as they move within or between cells. DoA estimation algorithms can be used to find the location of a underwater sound source relative to the antenna array. Direction of different sound sources near us can also be located by use of these algorithms. Radio telescope use these techniques to look at certain locations in the sky.

In practice, the estimation is difficult by the fact that there are usually an unknown number of signals impinging on the array simultaneously, each from unknown

directions and with unknown amplitudes. Also, the received signals are always corrupted by noise. Nevertheless, there are several methods to estimate the number of signals and their directions. Several algorithms and techniques have been proposed concerning direction of arrival estimation. The most popular techniques are super resolution algorithms, Multiple Signal Classification (MUSIC), Estimation Of Signal Parameters via Rotational Invariance Techniques (ESPRIT) and their variants (e.g. Root MUSIC) [1]-[4], where many snapshots of uncorrelated signals are needed. Some faster algorithms are also proposed which use small number of signal snapshots [5],[6]. In these methods, DoA is estimated by computing the spatial spectrum $P(\theta)$, the mean power received by an array as a function of θ, and then determining the local maximas of this computed spectrum.

In practical situations, in a large array, it is quite possible that some of the antenna elements in the array not function properly. In that case our analytical techniques of DoA estimation may not work properly, as array correlation matrix changes under sensor failure condition. They involve extensive signal processing, like eigenvalue decomposition of the incoming signals and autocorrelation matrix calculations. As a result they are difficult to implement in real time. Their performance degrades in case of correlated sources. Signals have to be uncorrelated and there is a need for many signal snapshots. So, there is a need for DoA estimation techniques which are immune to few sensor failure condition and which can provide satisfactory results in case of correlated sources. In the present work, instead of analytical techniques, neural network (NN) technique have been used for direction finding of a single and multiple sources to mitigate this requirement.

A neural network is a massively parallel processor made up of simple processing units, which has a natural propensity for storing experiential knowledge and making it available for use [7]. NNs are trained to model complex relationships between certain inputs that produce certain outputs, determined by the weight connections of the neurons. They are computationally efficient and are more immune to noise. A drawback of neural scheme is the selection of network size which is done by trial and error.

In DS-CDMA all users communicate simultaneously at the same band [8], consequently each signal's DoA has to be estimated in the presence of many other interfering signals that constitute the Multiple Access Interference (MAI). A NN based DoA estimation in DS-CDMA can estimate the direction of desired user accurately even if the desired signal's strength is weaker in comparison to the interferers [9]-[11].

In the report a neural networks-based direction finding algorithm for single and multiple source direction finding is presented. NN techniques basically apply the mapping of the signal autocorrelation matrix with the signals' angles of arrival (AoA), requiring no extra eigenvalue decomposition processes [12]–[16].We need to train the network according to our requirement. A NN can be trained to handle sensor failure. So, a properly trained NN can be used to detect single and multiple users even in the case in which few sensors of array have suffered failures. The method also doesn't require the knowledge of the element in which failure has occurred [17]. The number of sensor failure it can handle depends on the number of sensor failures for which it is trained.

1.2 Problem Statement

The present research deals with DoA estimation using NN technique. The following problems are addressed in this report :

1) Neural Network based DoA estimation in DS-CDMA for a Switched Beam Antenna System

2) Neural Network based DoA estimation in a failed antenna array system.

1.3 Organization of the Dissertation

The remaining part of dissertation is organized into the following chapters-

Chapter 2 presents a brief mathematical background associated with DoA Estimation. Analytical methods of DoA estimation and their limitations are also presented in this chapter. Chapter 3 presents a literature survey on Neural Network. Chapter 4 discusses the application of NN based DoA estimation in DS-CDMA for a Switched Beam Antenna System. Chapter 5 presents a NN based DoA estimation using an array with faults. Conclusion and future scope of work follows Chapter-5.

2. DoA Estimation

2.1 Introduction

A linear array composed of M elements is considered to be located in the far field of K directional sources. The directional signal incident on the array can be considered as a plane wave front. Thus K $(K<M)$ number of narrowband plane waves, centered at frequency ω_0 impinge on the array from directions $\{\theta_1\ \theta_2\ \ldots\ldots\ \theta_K\}$ [1],[13].

Using complex signal representation, the received signal at the ith array element can be written as

$$x_i(t)=\sum_{m=1}^{K} s_m(t)e^{-j(i-1)k_m}+n_i(t);\ i=1,2,\ldots..,M. \tag{2.1}$$

where $s_m(t)$ is the signal of the mth wave, $n_i(t)$ is the noise signal received at the ith sensor and

$$k_m=\frac{\omega_0 d}{c}\sin(\theta_m) \tag{2.2}$$

where d is the spacing between the elements of the array, and c is the speed of light in free space. Using vector notation we can write the array output in a matrix form

$$X(t) = AS(t) + N(t) \tag{2.3}$$

where $X(t)$,$N(t)$,and $S(t)$ are given by

$$X(t)=[x_1(t)\,x_2(t)\cdots x_M(t)]^T \tag{2.4}$$

$$N(t)=[n_1(t)\,n_2(t)\cdots n_M(t)]^T \tag{2.5}$$

$$S(t)=[s_1(t)\,s_2(t)\cdots s_K(t)]^T \tag{2.6}$$

A is the $M\times K$ steering matrix of the array toward the direction of incoming signals defined as

$$A=[a(\theta_1)\ \ldots\ a(\theta_m)\ \ldots\ a(\theta_K)] \tag{2.7}$$

where $a(\theta_m)$ corresponds to

$$a(\theta_m)=\left[1\ e^{-jk_m}\ e^{-j2k_m}\ \cdots e^{-j(M-1)k_m}\right] \tag{2.8}$$

If it is assumed that the noisy signals $\{n_i(t), i=1:M\}$ received at the different sensors are statistically independent white noise signals of zero mean and variance σ^2 and are also independent of $S(t)$, then received spatial correlation matrix R of the received noisy signals can be expressed as

$$R = E[X(t)X^H(t)] = AE[S(t)S^H(t)]A^H + E[N(t)N^H(t)] \quad (2.9)$$

The antenna array can be thought of as performing a mapping G: $R^K \rightarrow C^M$ from the space of the DoA's $\{\theta = [\theta_1, \theta_2, \cdots, \theta_K]^T\}$ to the space of sensor output $\{X(t) = [x_1(t)\ x_2(t)\ \ldots\ x_M(t)]^T\}$. If the reverse mapping function F(x) is known, the DoA from the received signal $X(t)$ can be determined.

$$\theta = F(x) \quad (2.10)$$

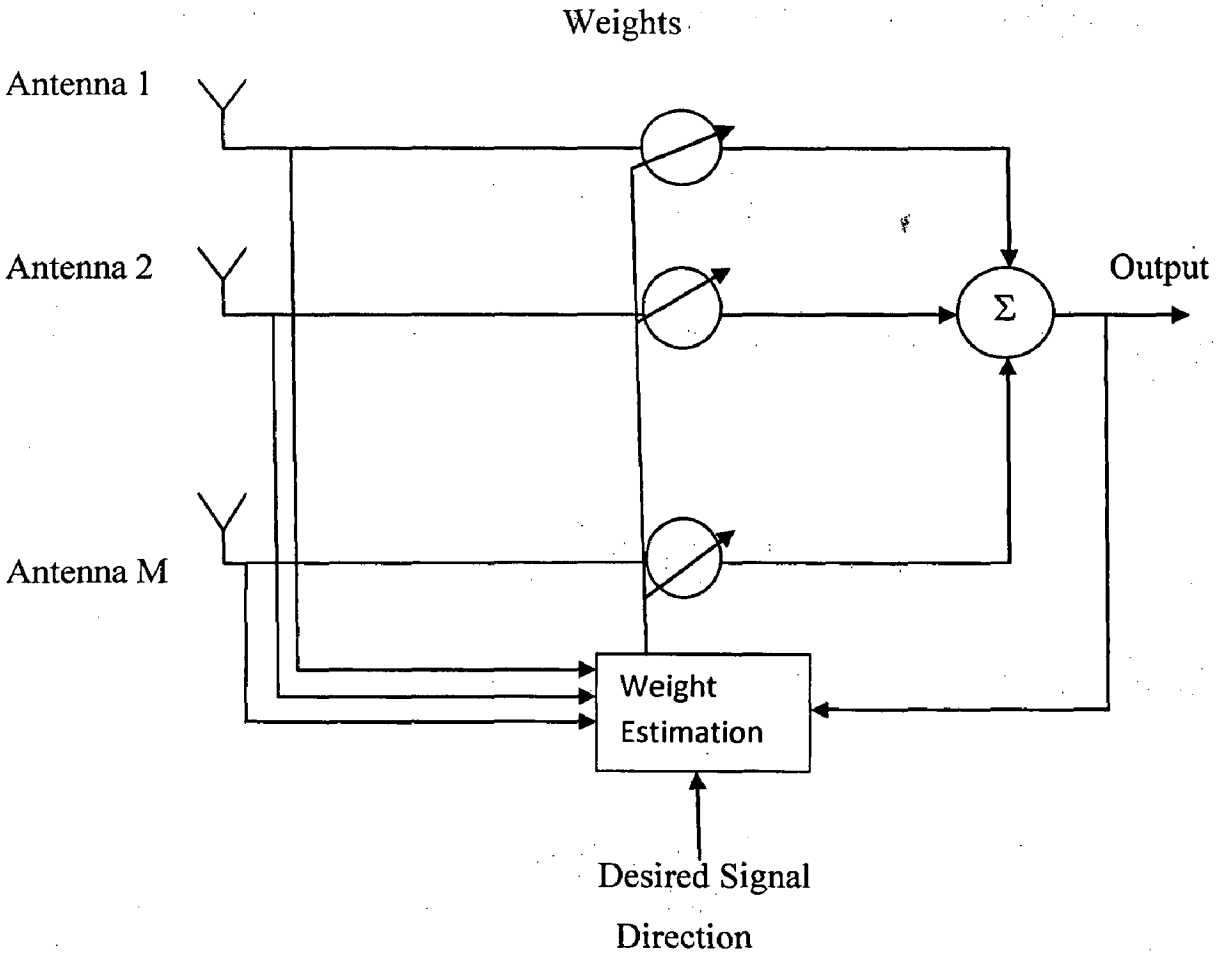

Fig 2.1 Block Diagram of front end of communication system using an antenna array [1]

The response of a processor with weight vector w connected to the antenna array is given by

$$Y(t)=w^H X(t) \tag{2.11}$$

where $w=[w_1, w_2, ..., w_M]^T$ (2.12)

The output power of the array at any time t is given by the magnitude square of the array output.

$$\begin{aligned} P(t) &= |Y(t)|^2 \\ &= w^H X(t) X^H(t) w \end{aligned} \tag{2.13}$$

The mean output power of the array system is obtained by taking conditional expectation over x(t):

$$\begin{aligned} P(w) &= E\left[w^H X(t) X^H(t) w\right] \\ &= w^H E\left[X(t) X^H(t)\right] w \\ &= w^H R w \end{aligned} \tag{2.14}$$

2.2 ANALYTICAL METHODS OF DoA ESTIMATION

These methods include spectral estimation, minimum variance distortion less response estimator (MVDR), linear prediction, maximum entropy, maximum likelihood and various eigenstructure methods [1],[2]. MUSIC is a simple and efficient eigenstructure variant of DoA estimation methods. There is a Fourier relationship between the beam pattern and the excitation at the array. This allows the DoA estimation problem to be treated as equivalent to spectral estimation. In these methods DoA is estimated by computing the spatial spectrum $P(\theta)$, the mean power received by an array as a function of θ, and then determining the local maximas of this computed spectrum.

2.2.1 MVDR Estimator

It finds the maximum likelihood estimate of power arriving from a point source in direction θ assuming that all other sources are interference. The goal is to maximize the signal to interference ratio while passing the signal of interest undistorted in phase and amplitude. The expression for power spectrum is

$$P_{MV}(\theta) = \frac{1}{a_\theta^H R^{-1} a_\theta} \tag{2.15}$$

where a_θ denotes the steering vector associated with direction θ and R is the array correlation matrix.

2.2.2 Linear Prediction (*LP*) Method

The linear prediction (*LP*) method estimates the output of one sensor using linear combinations of the remaining sensor outputs and minimizes the mean square prediction error, that is, the error between the estimate and the actual output. It obtains the array weights by minimizing the mean output power of the array subject to the constraint that the weight on the selected sensor is unity. Expressions for the array weights $\hat{w}$ and the power spectrum $P_{LP}(\theta)$, respectively, are

$$\hat{w} = \frac{R^{-1}u_1}{u_1^H R^{-1} u_1} \tag{2.16}$$

And

$$P_{LP}(\theta) = \frac{u_1^H R^{-1} u_1}{\left| u_1^H R^{-1} a_\theta \right|^2} \tag{2.17}$$

u_1 is a column vector such that one of its elements is unity and the other elements are zero.

2.2.3 Maximum Entropy Method

The maximum entropy (*ME*) method finds a power spectrum such that its Fourier transform equals the measured correlation subjected to the constraint that its entropy is maximized. The pseudospectrum is given by

$$P_{ME_j}(\theta) = \frac{1}{a_\theta^H c_j c_j^H a_\theta} \tag{2.18}$$

where c_j is the jth column of the inverse array correlation matrix (R_{xx}^{-1}).

2.2.4 MUSIC Algorithm

MUSIC is a simple and efficient eigenstructure variant of DoA estimation methods. These methods rely on the following properties of the array correlation matrix : (1) The space spanned by its eigenvectors may be partitioned in two subspaces, namely the signal subspace and the noise subspace; and (2) The steering vectors corresponding to the directional sources are orthogonal to the noise subspace. As the noise subspace is orthogonal to the signal subspace, these steering vectors are contained in the signal

subspace. It should be noted that the noise subspace is spanned by the eigenvectors associated with the smaller eigenvalues of the correlation matrix, and the signal subspace is spanned by the eigenvectors associated with its large eigenvalues.

Once the noise subspace has been estimated, a search for M directions is made by looking for steering vectors that are as orthogonal to the noise subspace as possible. This accomplished by searching for peaks in the MUSIC spectrum given by

$$P_{MU}(\theta)=\frac{1}{\left|a_\theta^H U_N\right|^2} \tag{2.19}$$

where U_N denotes an M by M-K dimensional matrix, with M-K columns being the eigenvectors corresponding to the M-K smallest eigenvalues of the array correlation matrix and a_θ denoting the steering vector that corresponds to direction θ. It should be noted that instead of using the noise subspace and searching for directions with steering vectors orthogonal to this subspace, signal subspace could also be used and search can be made for directions with steering vectors contained in this space. This amounts to searching for peaks in

$$P_{MU}(\theta)=\left|U_S^H a_\theta\right|^2, \tag{2.20}$$

where U_S denotes an $M\times K$ dimensional matrix with its K columns being the eigenvectors corresponding to the K largest eigenvalues of the array correlation matrix.

2.2.5 Simulated Results

Number of signals or desired user = 2

Number of antenna elements in array =6

Signals are assumed to be of binary Walsh type with amplitude 1.

Noise is assumed to be Gaussian distributed with variance of 0.1.

100 time samples of both signals and noise is considered.

Pseudospectrum of the DoA estimation techniques MVDR, Linear Prediction, Maximum Entropy, MUSIC are shown in fig 2.2, 2.3, 2.4, 2.5 for directional sources present at -5^0 and 5^0. Peaks should be present in the pseudospectrum at the position of the directional source. There is certain error as the peaks are not positioned exactly at -5^0 and 5^0 in Fig

(2.3), (2.4) and (2.5). But for MUSIC pseudospectrum Fig (2.5) peaks are obtained at the exact location of directional sources.

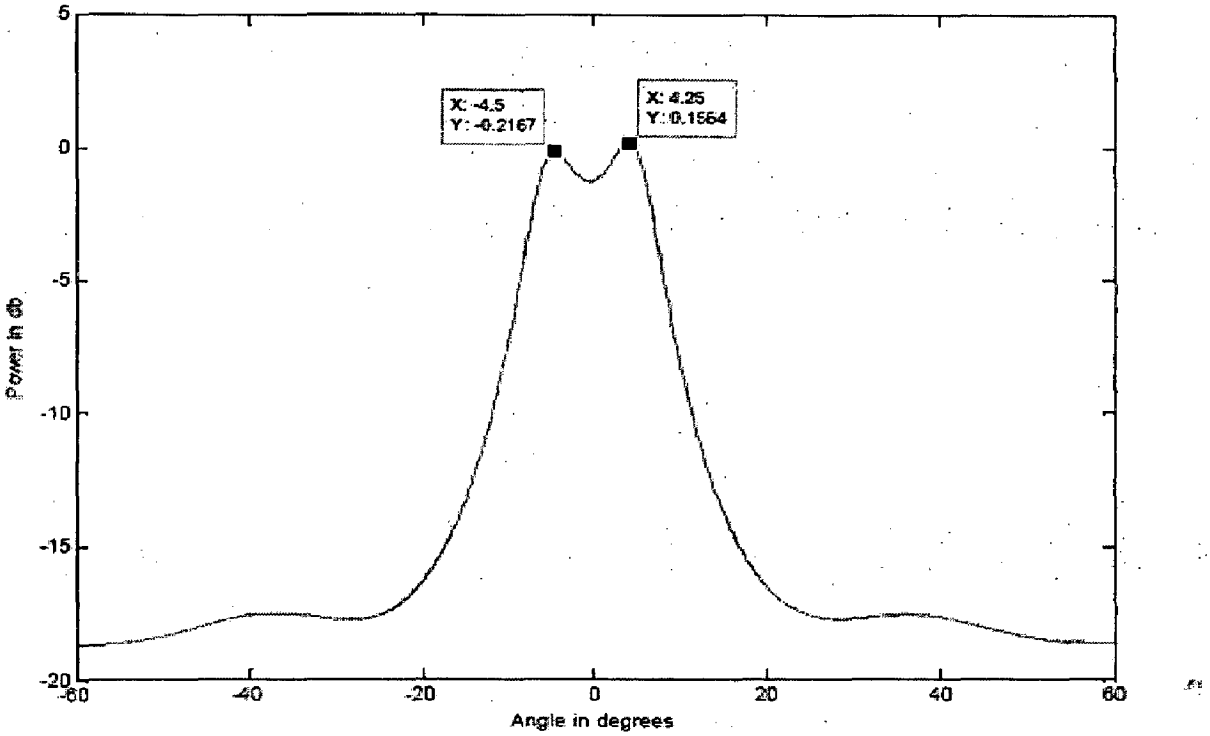

Fig 2.2 MVDR Pseudospectrum for θ_1=-5^0 and θ_2 =5^0

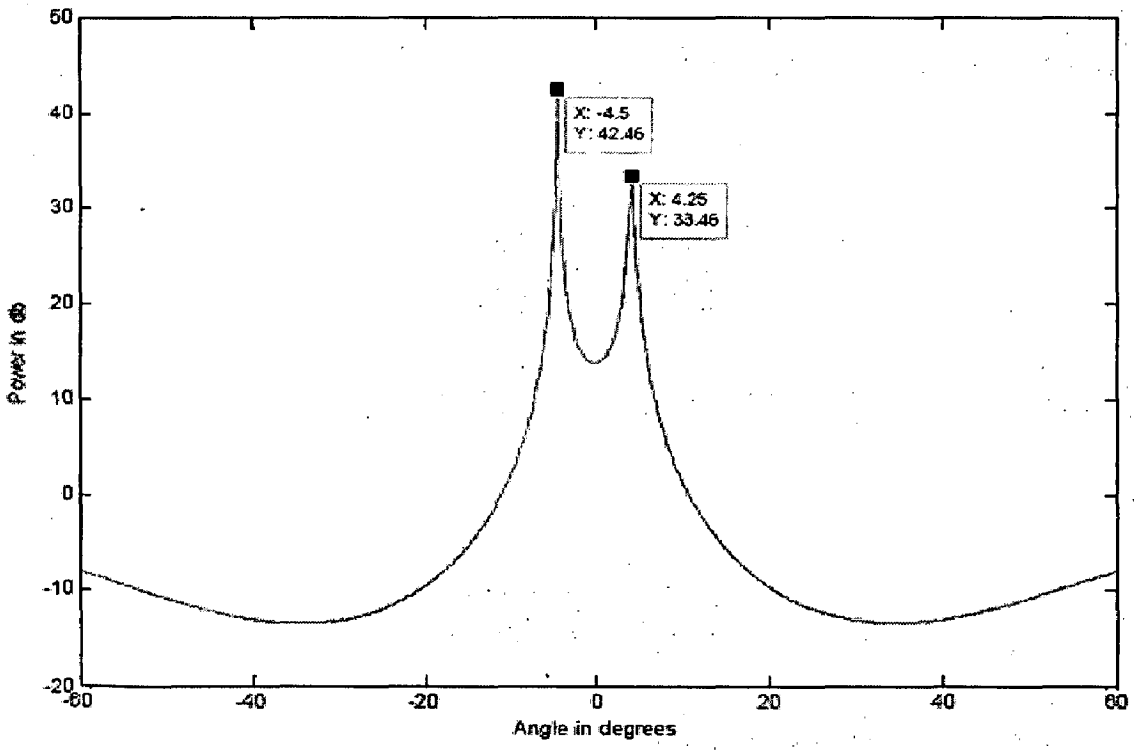

Fig 2.3 Linear Prediction Pseudospectrum for θ_1=-5^0 and θ_2 =5^0

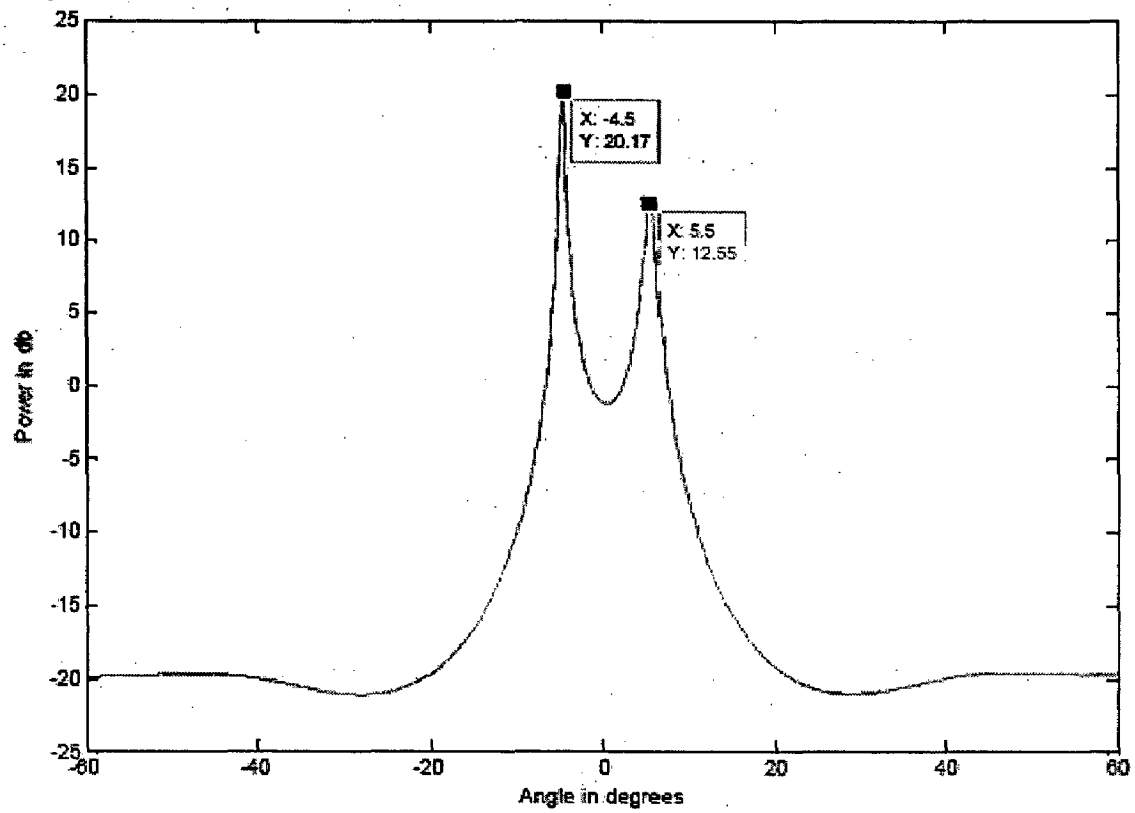

Fig 2.4 Maximum Entropy Pseudospectrum for θ_1=-5^0 and θ_2 =5^0

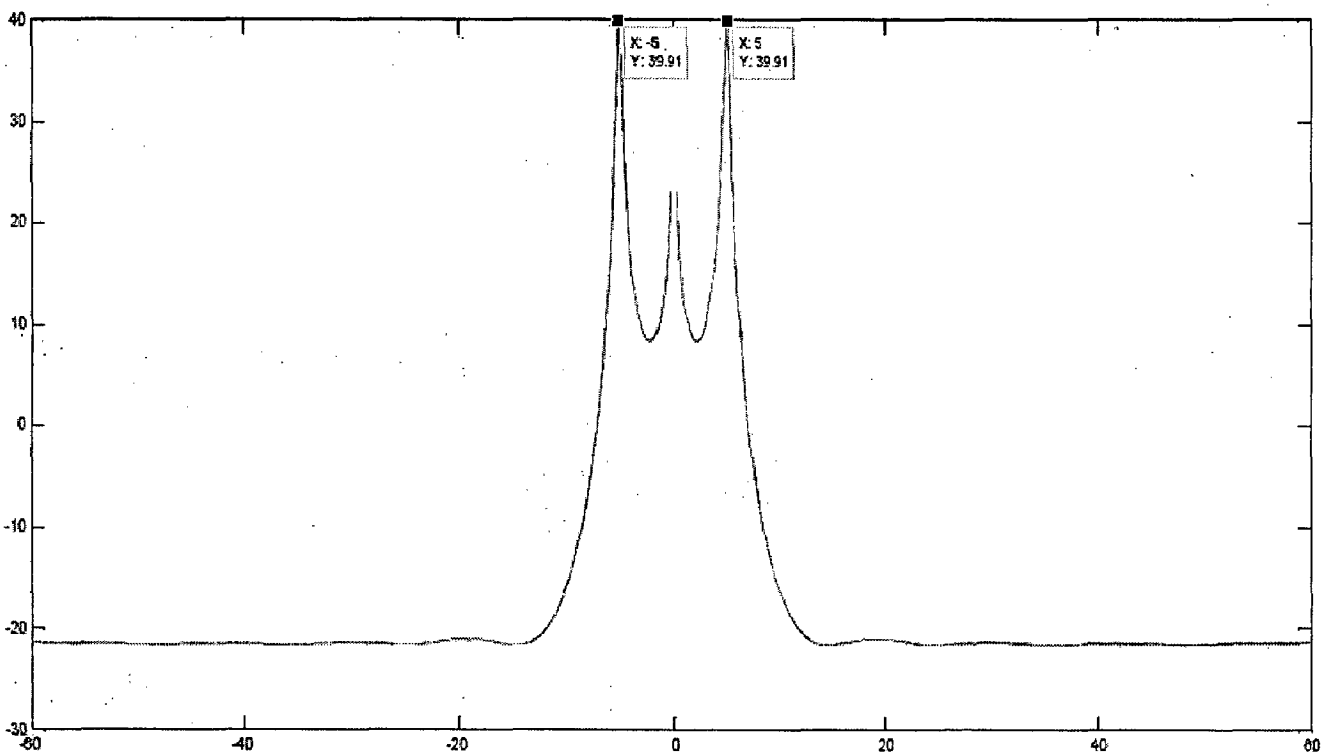

Fig 2.5 MUSIC Pseudospectrum for θ_1=-5^0 and θ_2 =5^0

2.3 Shortcomings

The MUSIC algorithm can estimate the DoAs of signals with adjacent incidence angles. However, many array input data snapshots are required to obtain a good performance. In addition, the DoAs of the signals must not be changed while the input data is being received to acquire the covariance matrix, plus the incidence signals must be uncorrelated. Thus, estimating this matrix is a significant computation load. When signals are correlated or the DoAs of signals change rapidly, the performance of these algorithms is severely degraded. It is also impossible to estimate the DoAs of signals quickly, because the MUSIC algorithm cannot perform a DoA estimation until it has received all the signal snapshots for estimating covariance matrix. Similar argument hold for other algorithms like MVDR, Linear prediction as they too are dependent on an estimate of covariance matrix. They involve extensive signal processing, like eigenvalue decomposition of the incoming signals and autocorrelation matrix calculations. As a result they are difficult to implement in real time.

In practical situation if we are using a large array then it is quite possible that some of the antenna elements in the array is not functioning properly. In that case our analytical techniques of DoA estimation may not work properly as array correlation matrix changes under sensor failure condition. For handling sensor failure many modifications have been proposed [18]-[20]. But for that also knowledge of element and time instant of failure is required.

For further savings in terms of computation load in DoA estimation we go for Neural Network based direction finding for single and multiple sources. These techniques are popular because of their robustness and efficiency in problems where an optimal solution is desired.

3. Neural Network

A neural network is a massively parallel processor made up of simple processing units, which has a natural propensity for storing experiential knowledge and making it available for use [7]. It resembles the brain in two respects:

1. Knowledge is acquired by the network from its environment through a learning process.
2. Interneuron connection strength, known as synaptic weights, are used to store the acquired knowledge.

A neuron is an information processing unit that is fundamental to the operation of neural network.

In mathematical terms a neuron k is written as

$$u_k = \sum_{j=1}^{m} w_{kj} x_j \tag{3.1}$$

$$y_k = f(u_k + b_k) \tag{3.2}$$

where

$x_1, x_2, \ldots, x_m$ are input signals. b_k is the bias which can be positive or negative. f is the activation function.

Types of Activation Function:

1. Threshold function
2. Piecewise linear function
3. Sigmoid function

Network Architectures:-

1. Single layer feed forward networks
2. Multilayer feed forward networks
3. Recurrent network

Neural Network can be utilized to solve a mapping problem. In the DoA Estimation problem NN can be used to resolve the mapping between received signal at the sensor array and the AoA of directional sources. A MLP NN can be used to solve some difficult diverse problems by training them in a supervised manner with error back propagation algorithm. This algorithm is based on error correction learning rule [7],[21].

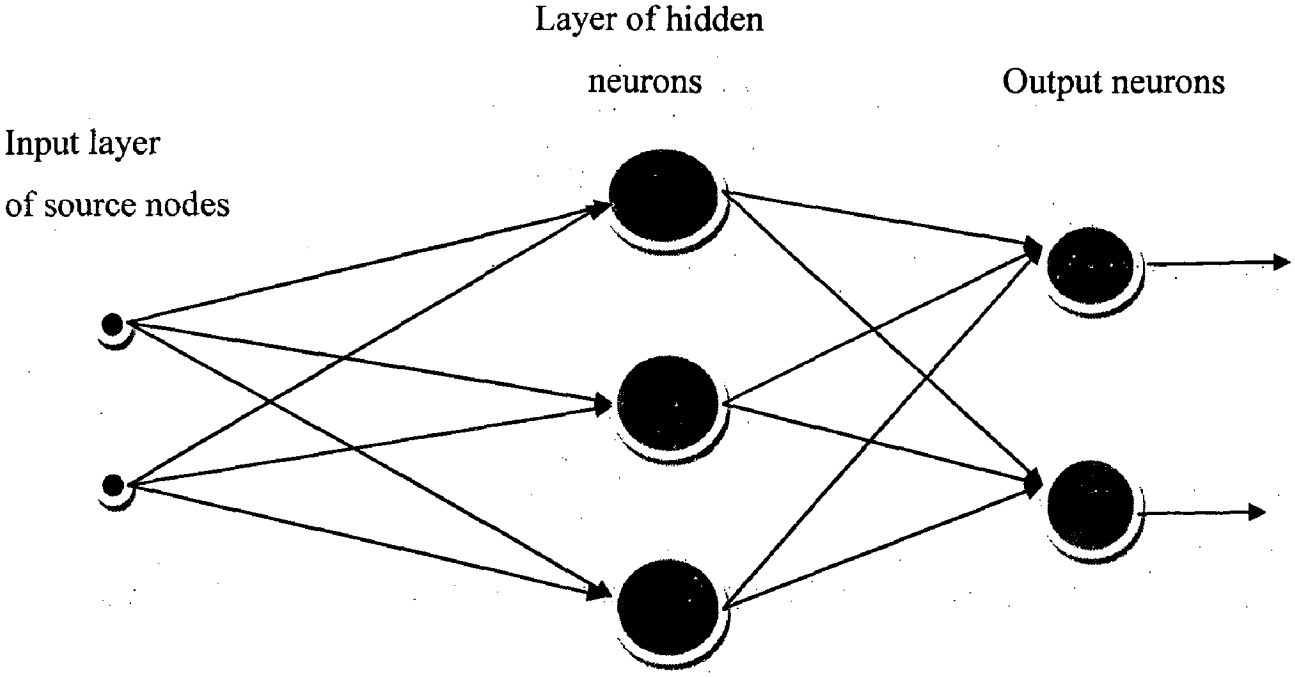

Fig 3.1 Multilayer feedforward Network [7]

Multilayer feedforward Network comprise of input layer composed of source nodes where input is fed to the network, hidden layer composed of hidden neurons and output layer composed of output nodes. There can be one or more hidden layers. The activation function of output nodes is linear, while the activation function of hidden layer is a sigmoid function. The synaptic weights undergo changes in accordance with the Levenberg-Marquardt (LM) algorithm [22]. To solve more complicated problem multiple hidden layers are used with increased number of hidden neurons.

4. Neural Network based DoA estimation for Switched-Beam Antenna System in DS-CDMA

4.1 The Switched Beam Antenna System

A smart antenna that relies on a fixed beamforming network, instead of a series of adaptive array processors, is called Switched beam system. In a SBS, a switch is used to select the best beam for receiving a particular signal, from a number of fixed beams [10]. Such systems are less expensive and less complicated in comparison to fully adaptive systems. The radiating part of the SBS used in the present work is a linear array of eight $\lambda/2$ spaced, microstrip patches structured on a single dielectric layer with substrate of $\varepsilon_r = 3.2$ (λ is the carrier wavelength). The array is fed by an 8×8 Butler matrix [23],[24] which is a fixed beamforming network of 8 input and 8 output ports. The entire system operates at 2.4 Ghz. The input ports of BM are connected to a switching network that performs beam switching using SPDT (single pole double throw) switches. Butler matrix consists of hybrid couplers, phase shifters, and crossovers. Corresponding to the excitation of each input port equal amplitude signals with a fixed phase shifts get generated through the output port. If these signals are used for feeding microstrip path antenna arrays then a set of orthogonal beams will be generated that can be used for scanning a particular sector of area for the purpose of tracking sources (mobile users).

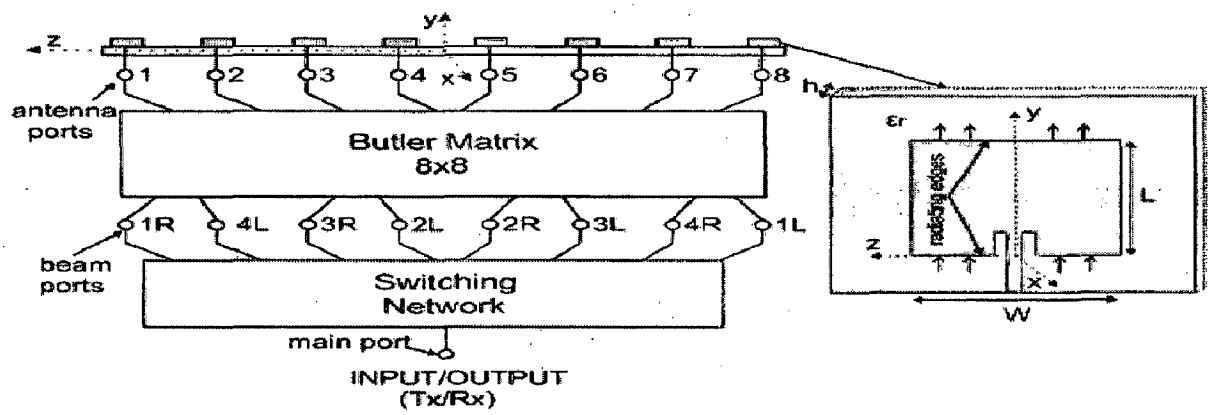

Fig 4.1 The proposed Switched Beam Antenna System [10]

A microstrip patch antenna is designed and simulated using CST Microwave studio 2009. The design frequency is 2.4 GHz and the other design parameters are ε_r=3.2 and h=1.524. The transmission line model [25] gives the length and width of patch, which are L=34.47 mm, W= 43.13 mm. In Fig 4.2 power pattern of patch has been plotted in the xz plane ($\Phi=0^0, 0^0 \leq \theta \leq 180^0$), which is the H-plane of microstrip antenna. The patch power pattern is denoted as $g(\theta)=\left|\overline{E(\theta)}\right|^2$, where $E(\theta)$ is Electric field vector.

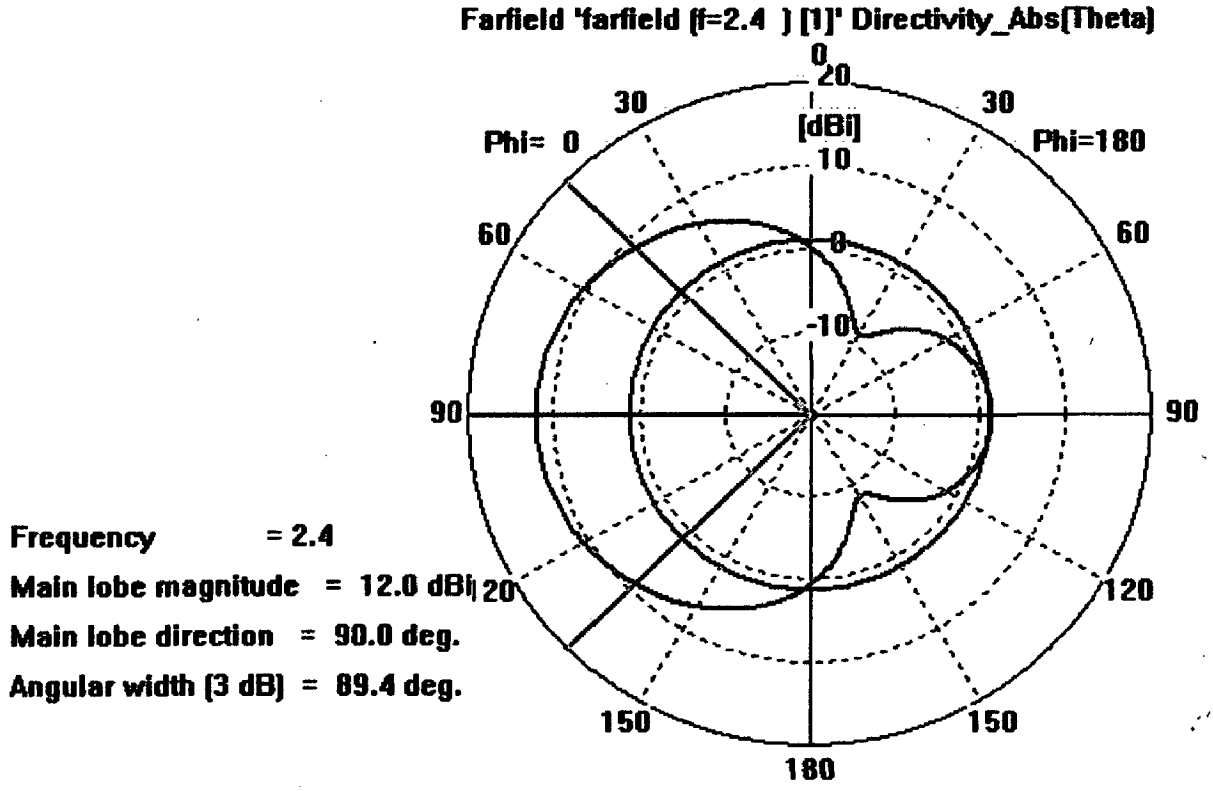

Fig 4.2 The power pattern of the rectangular microstrip patch in the x-z plane ($\Phi=0^0, 0^0 \leq \theta \leq 180^0$)

4.1.1 Uniform Illumination of Antenna Elements

The SBS is considered to be working in the transmission mode. The excitation of the ith input port correspond to an array weight vector

$$w_i^{uni} = \left[A, Ae^{j\beta_i^{uni}}, \cdots, Ae^{j(N_e-1)\beta_i^{uni}}\right]^T, i = 1,2,\cdots,8 \qquad (4.1)$$

that gives the uniform illumination of the antenna elements and a fixed phase difference β_i^{uni} between them. Each element of the above vector is the weight excited by the ith port at the n_eth antenna element ($n_e=1,2,\ldots\ldots,N_e$),and A is the uniform amplitude that depends

on input power. The above weight vectors generate a set of eight orthogonal beams that cover an angular sector of 120°. The total field pattern for the ith beam is given by

$$\vec{F}_i(\theta)=(w_i^{uni})^H a(\theta)g(\theta) \quad (4.2)$$

where $a(\theta)=[1\ e^{j\pi\sin\theta} \ldots\ldots e^{j(N_e-1)\pi\sin\theta}]^T$ (4.3)

is the steering vector in the direction θ.

g(θ) is the power pattern of a single microstrip patch antenna. The power pattern

$$P_i(\theta)=\left|\vec{F}_i(\theta)\right|^2 \quad (4.4)$$

is calculated and its plot is shown in Fig 4.3.

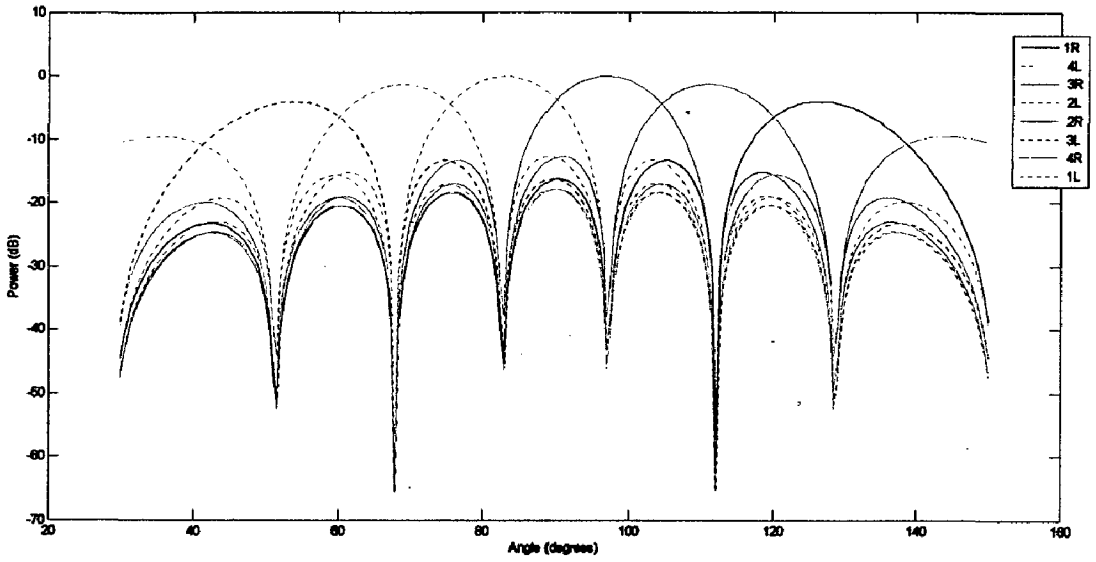

Fig 4.3 Eight orthogonal beams produced by the uniform illumination of eight microstrip patches fed by an 8 x 8 Butler Matrix.

4.1.2 Cosine Illumination Of Antenna Elements

In order to achieve a better SLL and CL performance, the equal combining of two input ports that correspond to two adjacent orthogonal beams should take place. Such a combination superimposes the excitation of the two beams producing the cosine illumination of the antenna elements. Seven pairs of inputs can be combined and the ith pair correspond to an array weight vector

$$w_i^{\cos} = \left[A_1, A_2 e^{j\beta_i^{\cos}}, \cdots, A_{N_e} e^{j(N_e-1)\beta_i^{\cos}}\right]^T, i = 1,2,\cdots,7 \quad (4.5)$$

that gives a phase difference $\beta_i^{\cos}$ and an amplitude A_{ne} at the n_eth antenna element [26]. The power pattern plot is shown in fig 4.4.

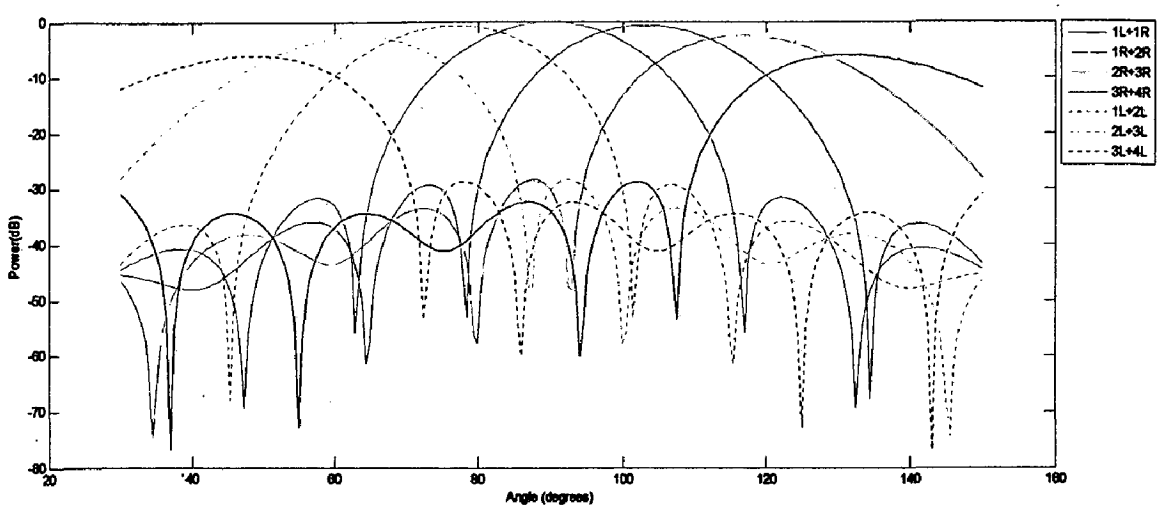

Fig 4.4 Seven beams produced by the cosine illumination of eight microstrip patches fed by an 8 x 8 Butler matrix.

Radiation pattern characteristics of each beam produced by uniform and cosine illumination is shown in table 4.1 and 4.2

Table 4.1

Radiation Pattern Characteristics For Each Orthogonal Beam Produced By The Uniform Illumination

Beam	Phase diff (deg)	Hpbw (deg)	Sll (dB)	Beam max. (deg)	Cl (dB)
4L	-157.5	19	-6.1	35.5	3L-4L : -6.56
3L	-112.5	15	-9.28	54	2L-3L : -5.2
2L	-67.5	13.4	-11.41	69	1L-2L : -4.41
1L	-22.5	12.5	-13.24	83	1R-1L : -3.72
1R	22.5	12.5	-13.24	97	1R-2R : -4.41
2R	67.5	13.4	-11.41	111	2R-3R : -5.2
3R	112.5	15	-9.28	126	3R-4R : -6.56
4R	157.5	19	-6.1	144.5	

Table 4.2

Radiation Pattern Characteristics For Each Beam Produced By The Cosine Illumination

Beam	Phase diff (deg)	Hpbw (deg)	Sll (dB)	Beam max. (deg)	Cl (dB)
3L+4L	-135	23	-22.725	48	3L4L-2L3L : -3.64
2L+3L	-90	20.5	-25.83	63	2L3L-1L2L : -2.38
1L+2L	-45	19.3	-28.56	77	1L2L-1L1R : -1.63
1L+1R	0	18.8	-31.50	90	1L1R-1R2R : -1.63
1R+2R	45	19.3	-28.56	103	1R2R-2R3R : -2.38
2R+3R	90	20.5	-25.83	117	2R3R-3R4R : -3.64
3R+4R	135	23	-22.725	132	

4.2 THE NN-SBS DOA ESTIMATION METHODOLOGY

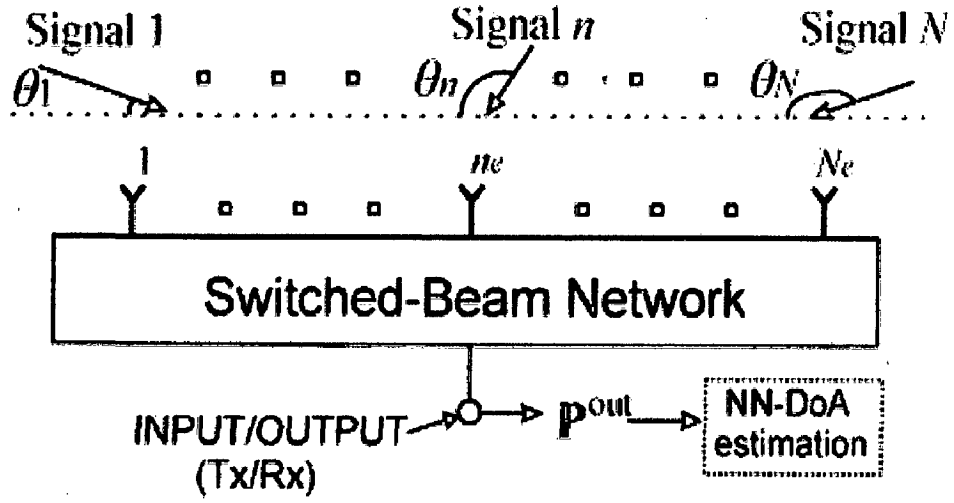

Fig 4.5 DoA Estimation for Switched Beam System [10]

N uncorrelated signals with amplitude h_i, $i=1.....N$ impinging at the N_e element antenna array of a SBS is considered. The angles of arrival θ_i, $i=1.....N$ compose a vector $\theta=(\theta_1,\theta_2,.....,\theta_N)$. M beams are used to cover the desired sector. If beam switching

takes place, the mth beam P_m gives at the main output of the system total received power $P_m^{out}(\theta)$, $m=1,2,\ldots\ldots,M$

$$\begin{aligned} P_1^{out}(\theta) &= h_1^2 P_1(\theta_1) + h_2^2 P_1(\theta_2) + \ldots\ldots + h_N^2 P_1(\theta_N) \\ P_2^{out}(\theta) &= h_1^2 P_2(\theta_1) + h_2^2 P_2(\theta_2) + \ldots\ldots + h_N^2 P_2(\theta_N) \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ P_M^{out}(\theta) &= h_1^2 P_M(\theta_1) + h_2^2 P_M(\theta_2) + \ldots\ldots + h_N^2 P_M(\theta_N) \end{aligned} \tag{4.6}$$

The system of equations shows the contribution of each signal to the total received power depends on its angle of incidence and the power pattern of the receiving beam. A power vector mapped to the corresponding angle vector θ can be constructed.

$$P^{out}(\theta) = \left\{P_1^{out}(\theta), P_2^{out}(\theta), \ldots\ldots, P_M^{out}(\theta)\right\} \tag{4.7}$$

Our aim is to utilize this mapping in order to accomplish DoA estimation. Mapping is resolved through a neural network. NNs should be trained that accept as input the measured/calculated power for each beam, and give as output the signals' DoA. For the NN to work the number of the output nodes must be equal to the number of the angles of arrival to be estimated

A set of K vectors θ_k is created, each one composed by randomly selected N angles of arrival θ_{ki}. The index k denotes the kth vector and the index i its ith element. The random angle values are equal to integral multiples of an angle step $\Delta\theta$. For every vector θ_k, the corresponding vector P_k^{out} is calculated. Thus a collection of randomly created pairs (P_k^{out}, θ_k) is generated, composing the NN training set. The general practice concerning the determination of training pairs number K, is its gradual increase until performance starts deteriorating due to overtraining.

4.3 NN-SBS DoA Estimation In DS-CDMA

4.3.1 DS-CDMA

DS-CDMA is a widely used access schemes in modern wireless communications, so the development of DoA estimation methods for smart antenna is of great importance and applicability [9]. In DS-CDMA all users communicate at the same time and band. Consequently, the SoI (Signal of Interest) angle of arrival has to be estimated in the

presence of a big number of other simultaneous signals that constitute the MAI. The DoA of the SoI is estimated by exploiting the suppression of the MAI by the de-spreading function of the code-matched filters. Thus, the problem is reduced to that of finding the angle of arrival of a single signal in the presence of a noise background.

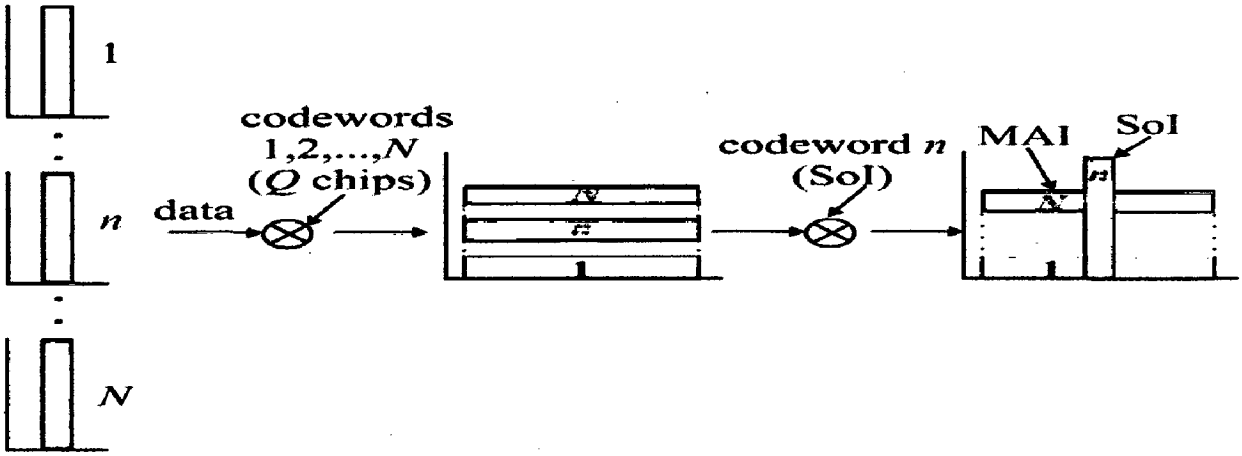

Fig 4.6 Principle of DS-CDMA operation [10]

4.3.2 DoA Estimation

Considering the transmitted signals from N mobile users impinge on the Switched Beam Antenna System which operates in DS-CDMA. The signals come from random directions $\theta_1,\ldots,\theta_n,\ldots\theta_N$ within an angle sector of 120^0.

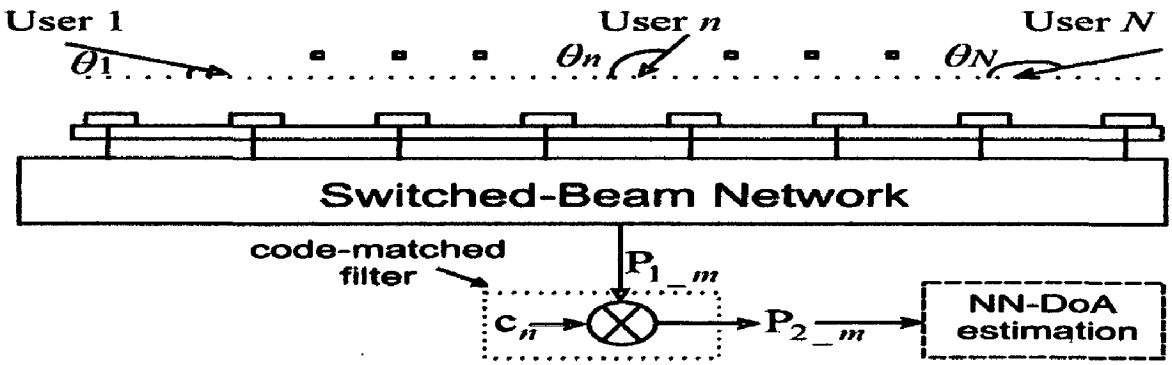

Fig 4.7 Switched-beam system used for DoA estimation in DS-CDMA [10]

Let the SoI come from the nth angle θ_n, while the other angles correspond to the MAI composing a vector $\theta_I = (\theta_1, \theta_2, \cdots \theta_N), \theta_n \neq \theta_I$. The stastical properties of the code sequences and data bits [9] have been used to give the total received power for the mth beam of a SBS before (P_{1_m}) and after (P_{2_m}) the code-matched filter

$$P_{1-m} = \underbrace{Qh_n^2 P_m(\theta_n)}_{SoI} + \underbrace{Q\sum_{l=1,l\neq n}^{N} h_l^2 P_m(\theta_l)}_{MAI} \tag{4.8}$$

$$P_{2-m} = \underbrace{Q^2 h_n^2 P_m(\theta_n)}_{SoI} + \underbrace{Q\sum_{l=1,l\neq n}^{N} h_l^2 P_m(\theta_l)}_{MAI} \tag{4.9}$$

where $P_m(\theta)$ is the power pattern of the mth beam, h_n is the SoI amplitude, h_l is the amplitude of the lth incoming signal (interferer) and Q is the number of chips per bit.

The first term of the above equations corresponds to the SoI contribution to the total received power, whereas the second to the MAI. Comparing $P_{1\text{-}m}$ and $P_{2\text{-}m}$ it is observed that the SoI term is despread and boosted by a factor of Q, called the processing gain of the DS-CDMA system. The interfering signals remain spread over a large bandwidth (Fig. 4.6). The proposed method exploits the MAI suppression so as to extract the SoI DoA. Thus, there is no problem if an interfering signal comes from a close or even the same direction as the SoI. Total received power after code matched filter, for each one of the M beams

$$\begin{aligned}
P_{2-1} &= Q^2 h_n^2 P_1(\theta_n) + Q\sum_{l=1,l\neq n}^{N} h_l^2 P_1(\theta_l) \\
P_{2-2} &= Q^2 h_n^2 P_2(\theta_n) + Q\sum_{l=1,l\neq n}^{N} h_l^2 P_2(\theta_l) \\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
P_{2-M} &= Q^2 h_n^2 P_M(\theta_n) + Q\sum_{l=1,l\neq n}^{N} h_l^2 P_M(\theta_l)
\end{aligned} \tag{4.10}$$

Power control is an inherent feature of DS-CDMA, thus it is assumed that all signals hit the base station antenna array at the same mean power level $\left(h_l^2 = 1, l = 1,2,\cdots\cdots,N\right)$. Therefore, for a given processing gain Q, the total power measured for the Mth beam depends on the AoA of the incoming signals and the radiation pattern of the beam. The mapping between the power vector, the corresponding SoI AoA and the MAI AoA vector is utilized in order to achieve DoA estimation based on neurocomputational techniques. For training of the NNs K sets of N AoA are randomly generated. N-1 angles of each set

compose the MAI AoA vector θ_{I-k} and an angle θ_{n-k} is the SoI AoA. The index k denotes the kth angle vector and the kth SoI angle. The random angle values are equal to integer multiples of $\Delta\theta$ within the sector angular range. $\Delta\theta$ is the sampling angle step of each beam's radiation pattern. Equation (4.9) gives the corresponding power vectors P_{2-k}.

4.3.2.1 Simulation Reults

A collection of 5000 randomly created pairs (P_{2-k}, θ_{n-k}) comprise the training set for the NN. Input and target pairs are preprocessed using the mapminmax function before applying as training pairs to the network. Neural Network employed is a Multilayer Perceptron with two hidden layers, the first hidden layer is having 20 nodes while the second one comprises of 30 nodes. Input layer will be having 8 nodes in case of uniform illumination and 7 nodes in case of cosine illumination. Since in DS-CDMA only one DoA (the SoI) is estimated, the NN output layer should have one node. Thus, the performance function is reduced to

$$E(x) = \frac{1}{2}\sum_{k=1}^{K}\left(\theta_{n-k} - \theta_{n-k}^{tr}\right)^2 \tag{4.11}$$

Training is achieved through error back propagation algorithm with the weights and biases updated using Levenberg-Marquardt algorithm. Once the training is achieved the network can be used to obtain outputs for input combinations other than for which it is trained. The actual output is obtained by postprocessing the output of network means doing the reverse of preprocessing. Mapminmax function used with a 'reverse' argument and same settings as in the preprocessing stage is applied over the output obtained from the network to get the actual output (SoI AoA). $\Delta\theta$=0.5, so, there will be a total of 241 possible values of the AoA in the sector of 30^0-150^0(30,30.5,.....,149.5,150).

Fig 4.8 and 4.10 shows the comparison of DoA Estimated from trained network against the desired DoA for 2000 testing samples for uniform and cosine illumination of antenna elements. The distribution of errors in both the case is depicted in the histogram plots of Fig 4.9 and Fig 4.11. A root mean square error of 0.4080 and 0.2168 is obtained for both the cases.

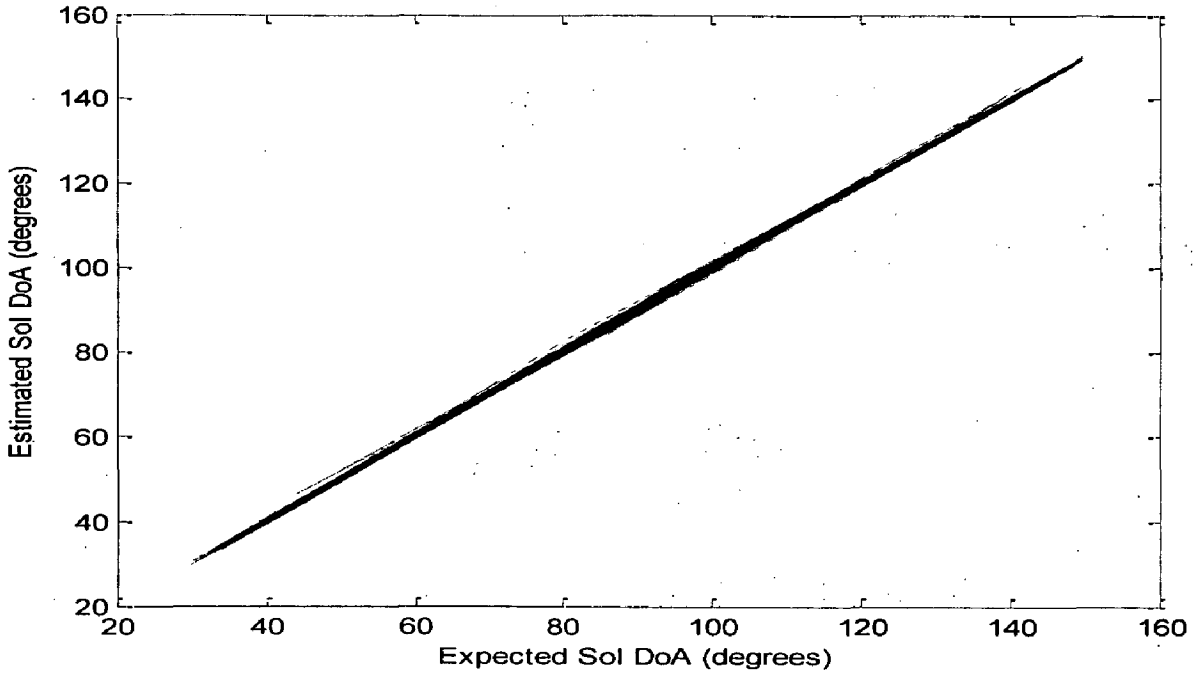

Fig 4.8 Estimated versus expected SoI DoA for the uniform illumination of the antenna elements and angle gap 0.5^0. RMSE=0.4080.

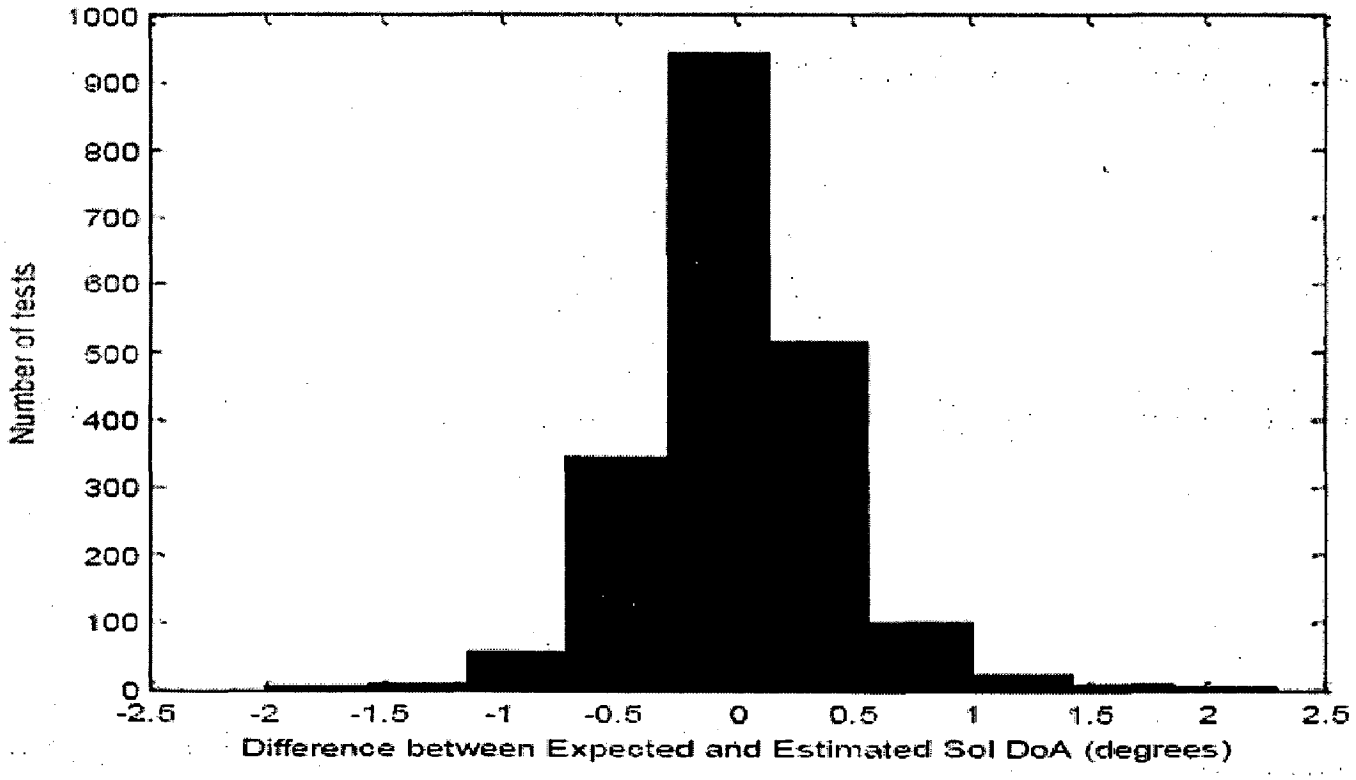

Fig 4.9 Distribution of difference between expected and estimated SoI DoA for the uniform illumination of the antenna elements and angle step 0.5^0.

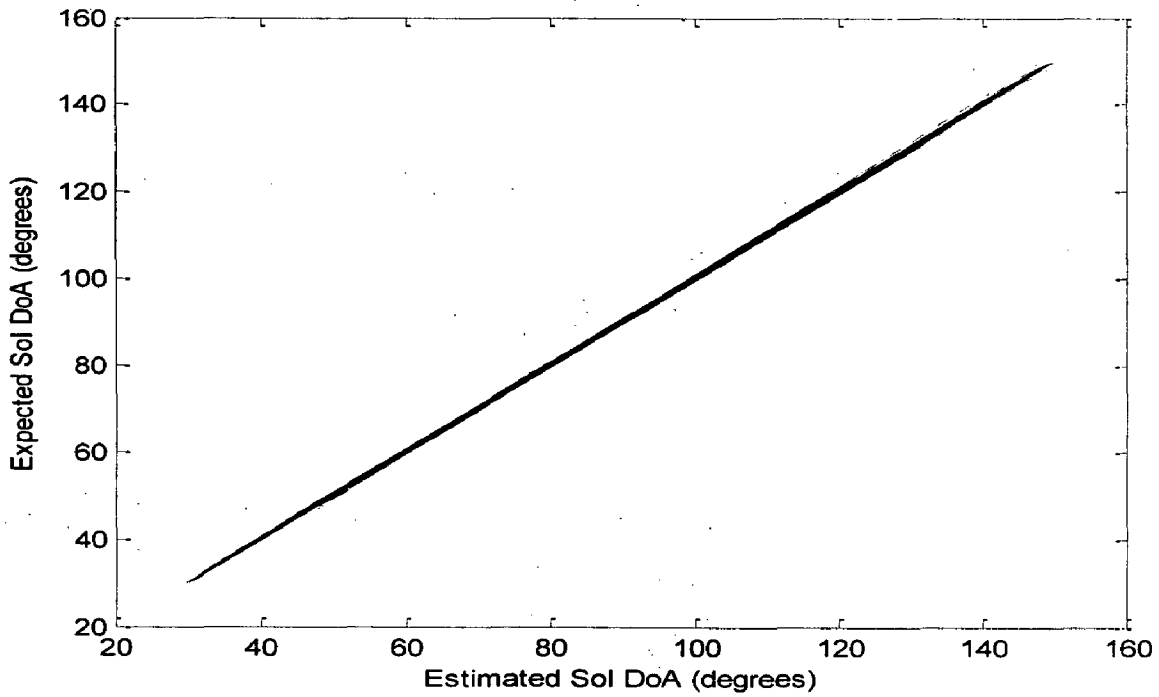

Fig 4.10 Estimated versus expected SoI DoA for the cosine illumination of the antenna elements and angle gap 0.5^0. RMSE=0.2168.

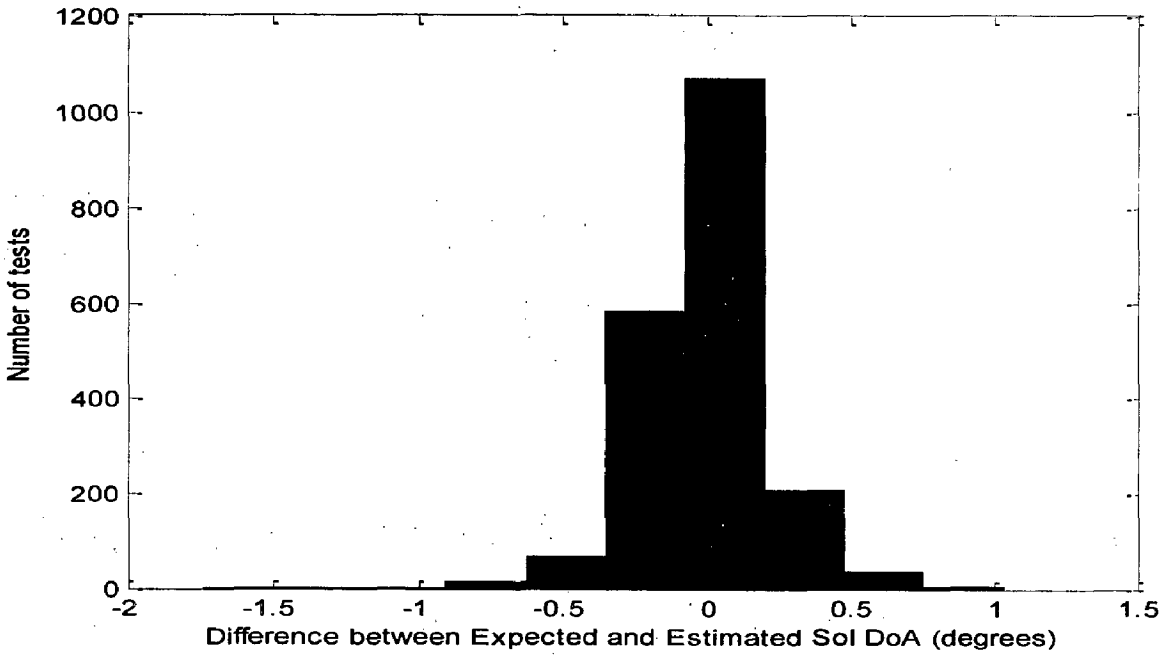

Fig 4.11 Distribution of difference between expected and estimated SoI DoA for the cosine illumination of the antenna elements and angle step 0.5^0.

4.3.3 IMPROVEMENT ON THE NN-SBS DOA ESTIMATION METHODOLOGY

If the desired signal impinges on the antenna array with different amplitude then the method discussed will not work. NNs have been trained with amplitude fluctuations, but the outcome was not desirable. Therefore, there is need for some additions or modifications so as to broaden the applicability of the technique by enabling DoA estimation regardless of the power level of the incoming signals.

4.3.3.1 Cosine Illumination Plus Sector Beam

In UMTS, besides the fixed beams, the base station needs to offer a sector beam for the common channels that are not allowed to use beamforming [27]. Usually a single element of the antenna array is used to serve the entire sector. In Fig.4.12 the seven beams produced by the cosine illumination are presented, plus the sector beam created by a single microstrip patch. The sector beam could be exploited to enhance the efficiency and applicability of the method.

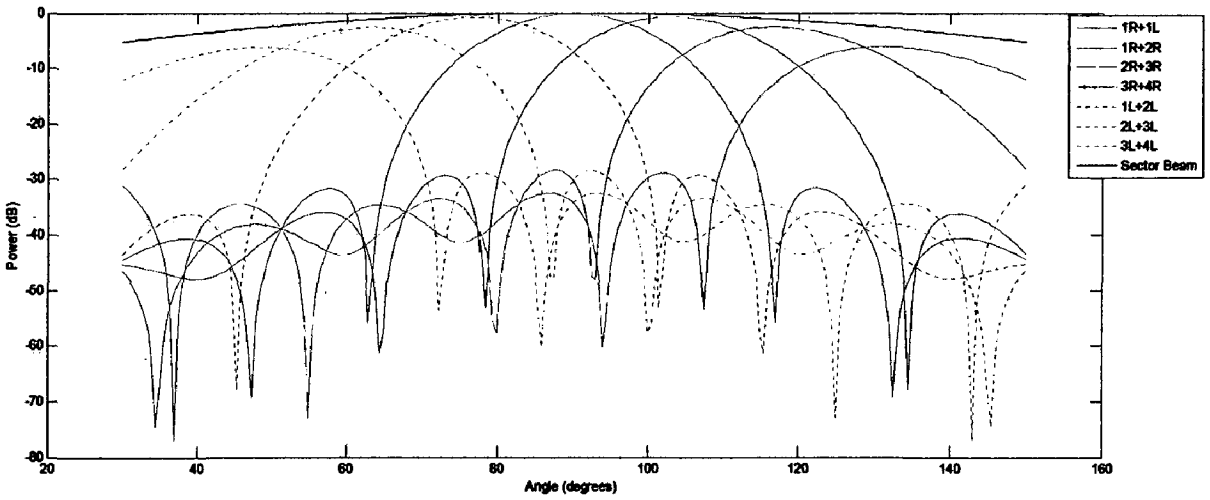

Fig 4.12 Seven fixed beams produced by the cosine illumination of eight microstrip patches and the sector beam produced by a single microstrip patch.

Like (4.9) the received power for the patch beam after the code-matched filter is written as

$$P_{2-p} = Q^2 h_n^2 P_p(\theta_n) + Q \sum_{l=1, l \neq n}^{N} h_l^2 P_p(\theta_l) \qquad (4.12)$$

The quotient of (4.9) and (4.12) gives

$$\frac{P_{2-m}}{P_{2-p}} = \frac{QP_m(\theta_n) + \sum_{l=1,l\neq n}^{N}\left(\frac{h_l}{h_n}\right)^2 P_m(\theta_l)}{QP_p(\theta_n) + \sum_{l=1,l\neq n}^{N}\left(\frac{h_l}{h_n}\right)^2 P_p(\theta_l)} \tag{4.13}$$

It is observed that the result of the above quotient does not depend on the absolute value of the signals' power. It depends on the ratio of the amplitude of each MAI signal to the SoI amplitude. If a NN is trained getting as inputs the ratio of each fixed beam's received power to the received power of the sector beam, the method works independently of the incoming signals' power level. Namely, there is no need for the amplitudes to be equal or close to any particular predefined value (e.g., unity).

4.3.3.2 Background Noise Addition

Besides the MAI caused by the undesired mobile users, background noise has to be added to our model. The additive background noise has been assumed to be spatially and temporally white complex Gaussian with zero mean and variance σ^2. If the variance σ^2 corresponds to each antenna element, the resultant mean noise power at the output of the beamforming network, before the code-matched filter, is

$$P_{gn-1} = w_m^H \sigma^2 I w_m \tag{4.14}$$

where w_m is the weight vector producing the mth beam pattern and I is the 8x8 identity matrix. Using the array weight vector from (4.5)

$$w_m^H I w_m = A_1^2 + A_2^2 + \cdots + A_8^2 = 1 \Rightarrow P_{gn-1} = \sigma^2 \tag{4.15}$$

The noise after the code-matched filter retains its properties having zero mean and variance $Q\sigma^2$ hence the noise power becomes $P_{gn-2} = Q\sigma^2$. The mth beam and the patch beam after the despreading give

$$\begin{aligned} P_{2-m} &= Q^2 h_n^2 P_m(\theta_n) + Q\sum_{l=1,l\neq n}^{N} h_l^2 P_m(\theta_l) + Q\sigma^2 \\ P_{2-p} &= Q^2 h_n^2 P_p(\theta_n) + Q\sum_{l=1,l\neq n}^{N} h_l^2 P_p(\theta_l) + Q\sigma^2 \end{aligned} \tag{4.16}$$

$$\frac{P_{2-m}}{P_{2-p}} = \frac{QP_m(\theta_n) + \sum_{l=1,l\neq n}^{N}\left(\frac{h_l}{h_n}\right)^2 P_m(\theta_l) + \left(\frac{\sigma}{h_n}\right)^2}{QP_p(\theta_n) + \sum_{l=1,l\neq n}^{N}\left(\frac{h_l}{h_n}\right)^2 P_p(\theta_l) + \left(\frac{\sigma}{h_n}\right)^2} \tag{4.17}$$

It is observed that the value of the above quotient depends on the angles of arrival, the signal to noise ratio (SNR) and the signal to interference ratio (SeIR).

4.3.3.3 Simulation Results

The MLPNN architecture and other parameters are same as in the equal amplitude signal level case. A processing gain of Q=256 is used. Background noise variance $\sigma^2 = 1$ is considered. Inputs to the network comprise the ratio of each fixed beam's received power to the received power of the sector beam while the outputs comprise the desired SoI AoA. 25000 training pairs are generated with the kth pair having a signal to each interferer ratio (SeIR) given by

$$\text{SeIR}_k = 10 * \log_{10}\left(\frac{h_{nk}}{h_l}\right)^2, l = 1,\ldots\ldots, N_k \neq n \tag{4.18}$$

N_k takes random values from 2 to 40.

In the first set of simulations the mean power of each interfering signal has been set to unity $(h_l^2 = 1)$, whereas the k^{th} SoI mean power takes random values from 0 to 0.1. Thus, each training pair has a random desired to undesired signal power ratio that ranges from 0 dB to -10 dB. The trained NN is tested with a total of 2000 testing samples for SeIR values of 0dB, -3dB, and -6dB . Fig (4.13) and Fig (4.14) shows the comparison of estimated SoI DoA against expected SoI DoA and error distribution histogram for SeIR=0dB. Similarly Fig (4.15), (4.16) and Fig (4.17), (4.18) show the corresponding plots for SeIR=-3dB and SeIR=-6dB. In the second set of simulations NN is trained with SoI mean power set to unity while interfering signal mean power taking values randomly between -5dB and 5dB. Fig (4.19) and (4.20) shows the comparison between Expected SoI DoA against the Estimated SoI DoA and error distribution histogram for 2000 samples tested with the trained NN.

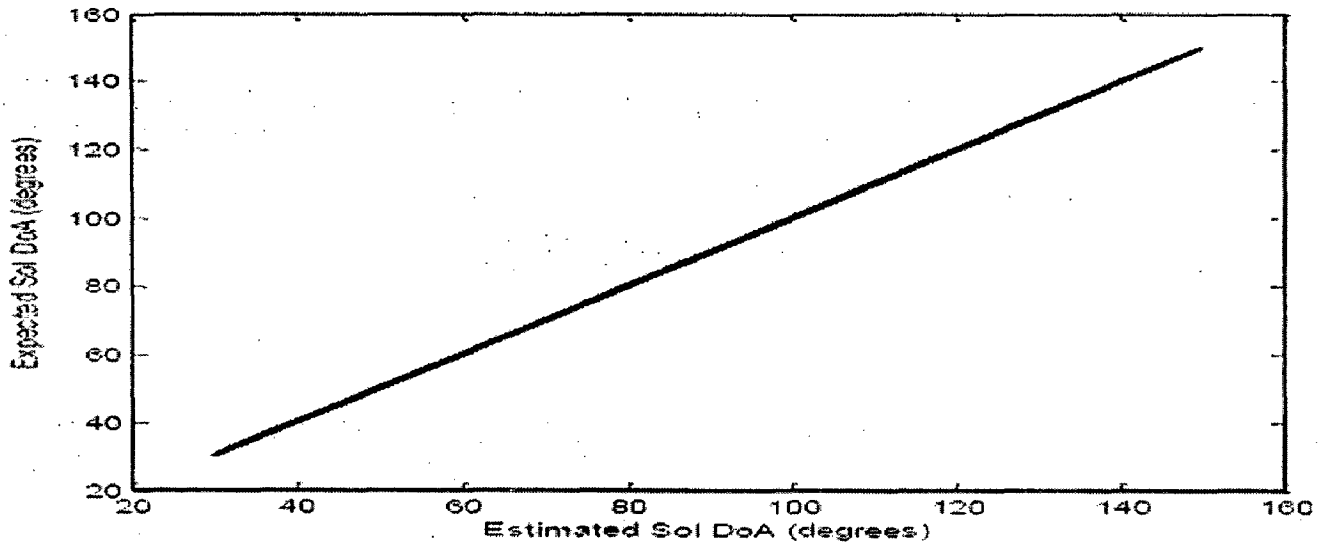

Fig 4.13 Estimated versus Expected SoI DoA for the cosine illumination of the antenna elements plus the sector beam and the angle step 0.5^0. 40 signals and SoI to each MAI signal power ratio 0 dB.RMSE=0.1405

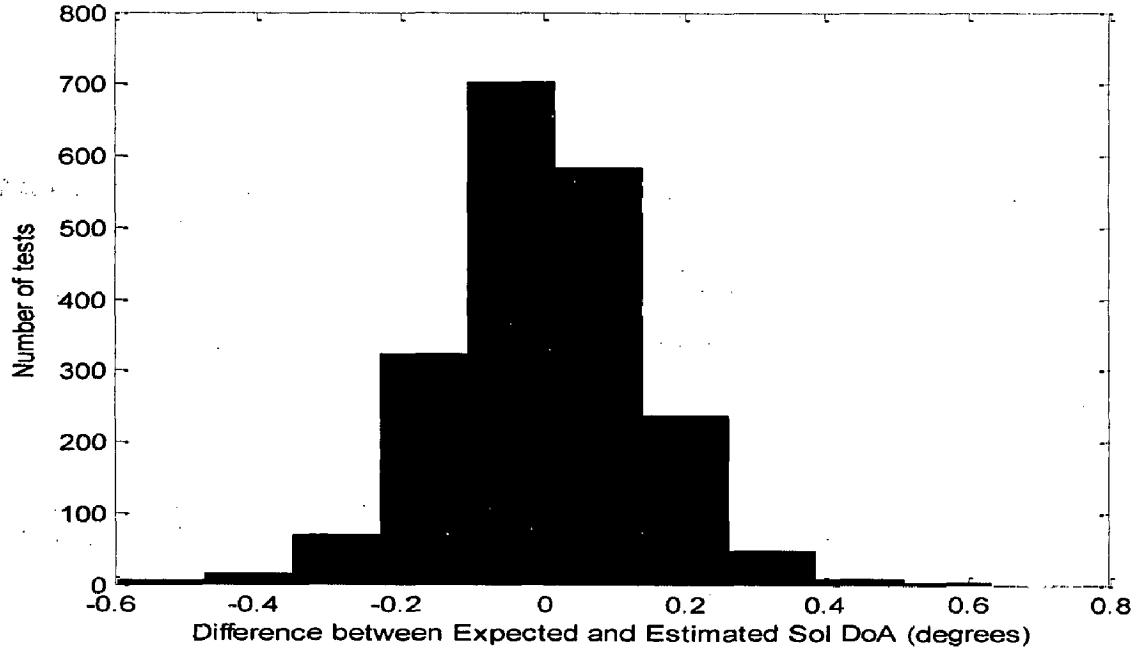

Fig 4.14 Distribution of difference between expected and estimated SoI DoA for the cosine illumination of the antenna elements plus sector beam and angle step 0.5^0, SeIR=0dB.

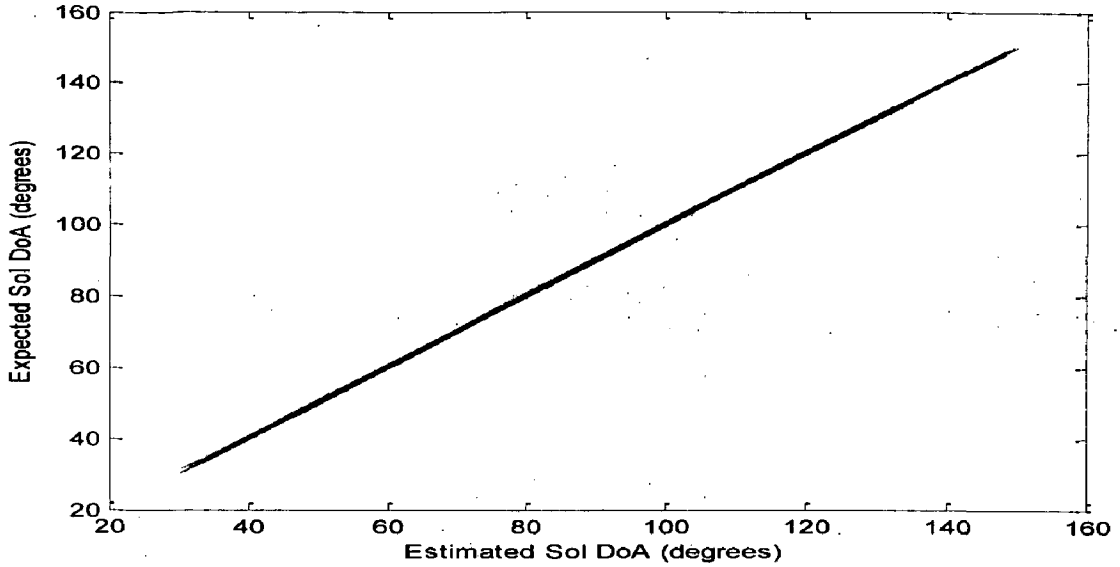

Fig 4.15 Estimated versus Expected SoI DoA for the cosine illumination of the antenna elements plus the sector beam and the angle step 0.5^0. 40 signals and SoI to each MAI signal power ratio -3dB.RMSE=0.2498

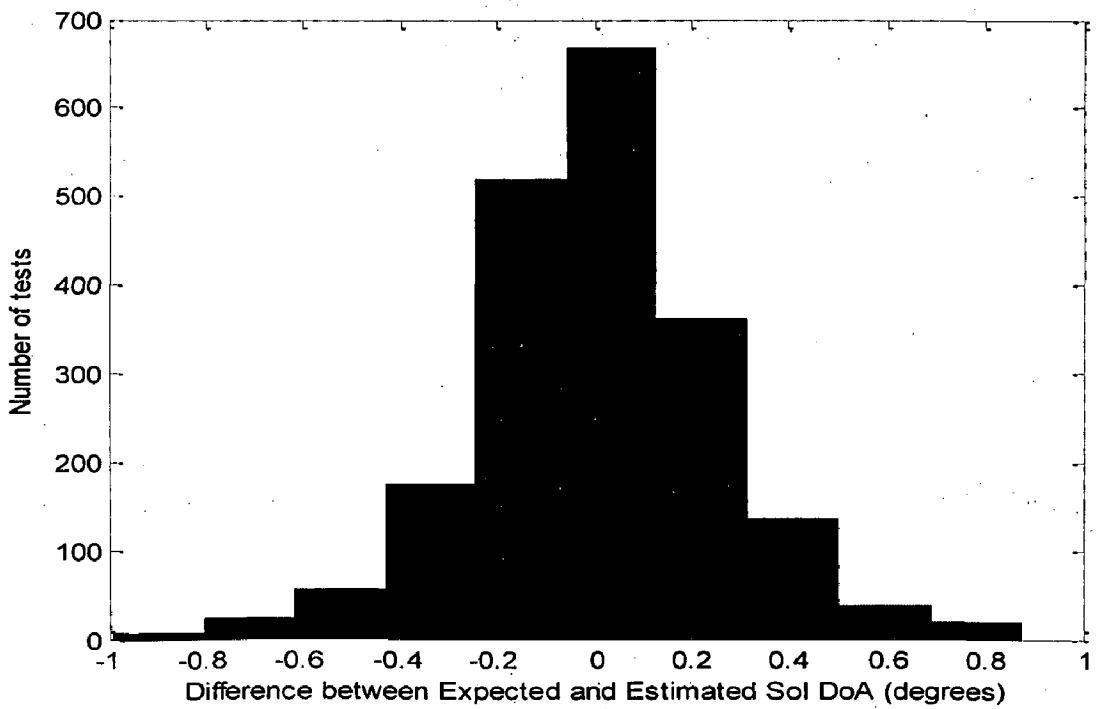

Fig 4.16 Distribution of difference between expected and estimated SoI DoA for the cosine illumination of the antenna elements plus sector beam and angle step 0.5^0, SeIR=-3dB.

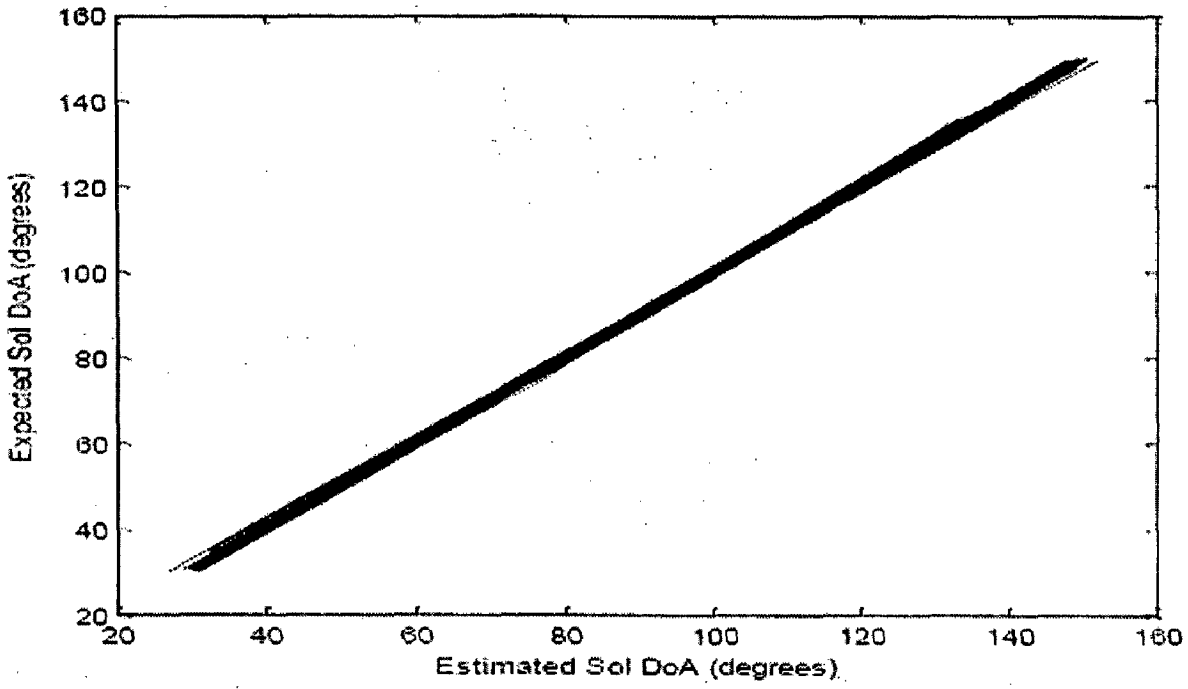

Fig 4.17 Estimated versus Expected SoI DoA for the cosine illumination of the antenna elements plus the sector beam and the angle step 0.5^0. 40 signals and SoI to each MAI signal power ratio -6dB.RMSE=0.5033

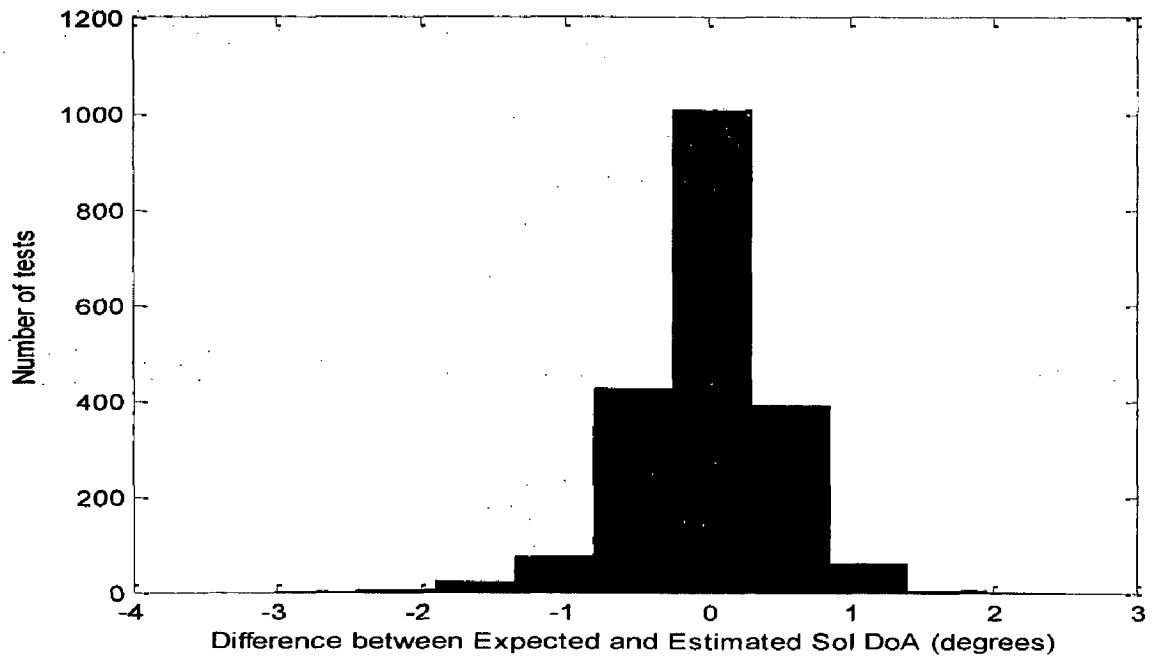

Fig 4.18 Distribution of difference between expected and estimated SoI DoA for the cosine illumination of the antenna elements plus sector beam and angle step 0.5^0, SeIR=-6dB

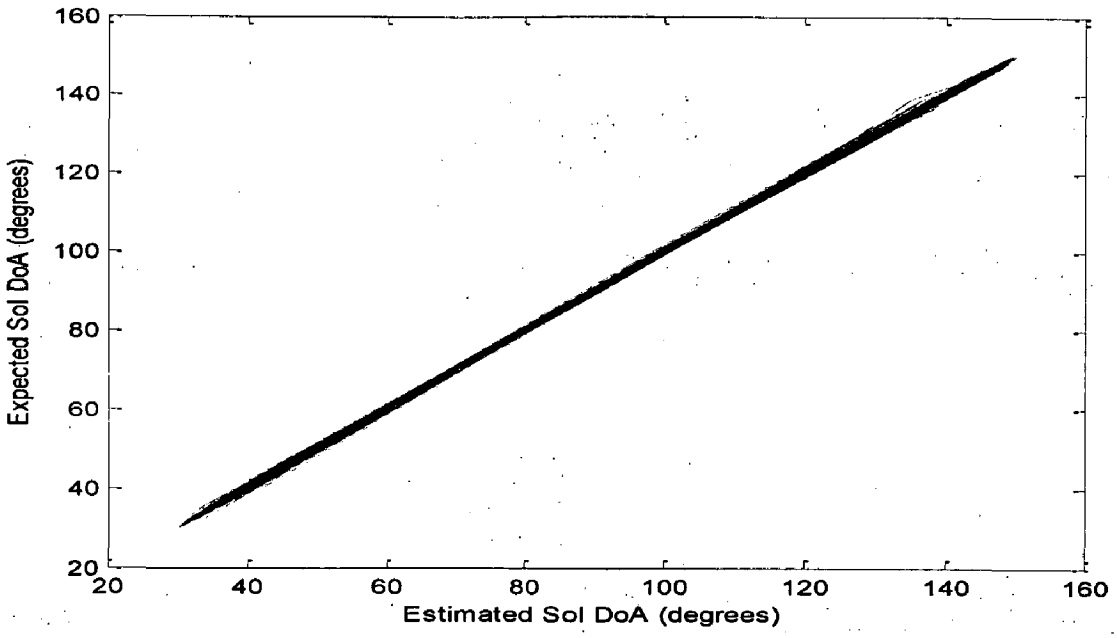

Fig 4.19 Estimated versus expected SoI DoA. The processing gain Q=256. SNR of the SoI is 0 dB and the SNRs of 39 desired users are randomly distributed in the range[-5 dB, 5dB]. RMSE=0.3815

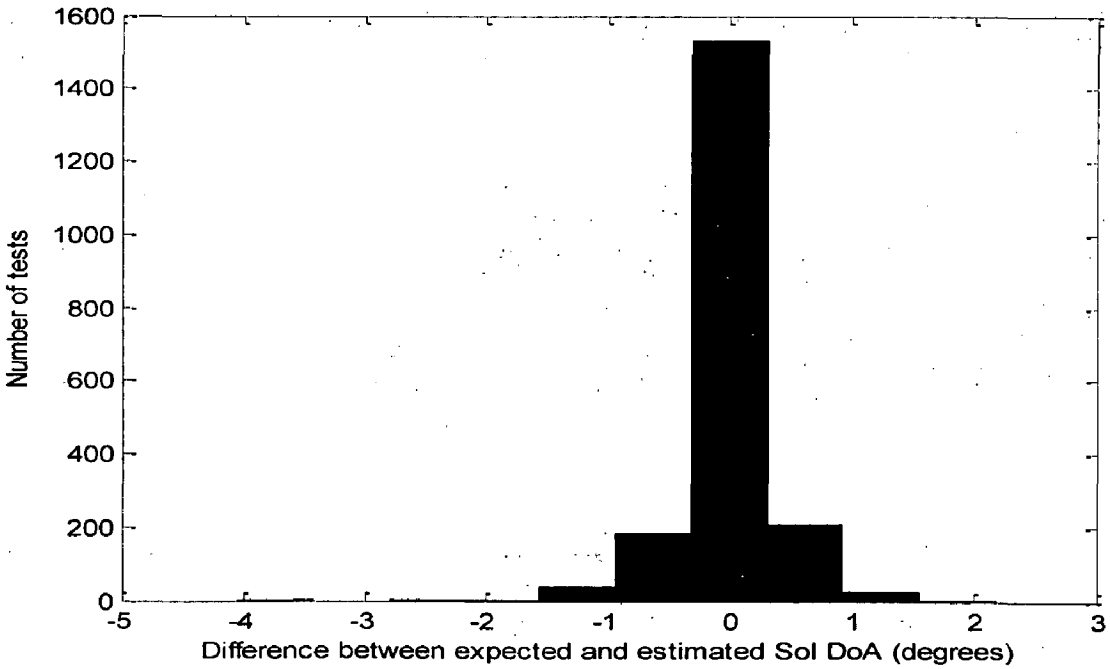

Fig 4.20 Distribution of difference between expected and estimated SoI DoA for the cosine illumination of the antenna elements plus sector beam and angle step 0.5^0, SeIR varying from -5dB to 5dB.

5. DoA Estimation In a Failed Antenna Array

5.1 NN – Based Direction Finding

If a M element antenna array is receiving signals from K directional sources then the antenna array can be thought of as performing a mapping G: $R^K \rightarrow C^M$ from the space of the DoA's $\{\theta = [\theta_1, \theta_2, \cdots, \theta_K]^T\}$ to the space of sensor output $\{X(t) = [x_1(t)\ x_2(t)\ \ldots\ x_M(t)]^T\}$. A neural network is used to perform the inverse mapping F: $C^M \rightarrow R^K$. If the reverse mapping function F(x) is known, the DoA from the received signal X(t) can be determined [13].

$$\theta = F(x). \tag{5.1}$$

Array processing algorithm utilize the correlation matrix R for direction of arrival estimation purposes instead of received signal X(t).

Array correlation matrix for a 3-element array can be written as

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{5.2}$$

By exploiting the symmetry in the correlation matrix R one need only consider either the upper or lower triangular part of the matrix. If we are using upper triangular part of the matrix then $M \times M$ spatial correlation matrix R can be organized in an $M(M+1)$-dimensional vector of real and imaginary parts denoted by b.

$$b = [r_{11}\ r_{12}\ r_{13}\ r_{22}\ r_{23}\ r_{33}] \tag{5.3}$$

The input vector b is normalized by its norm prior to being applied at the input layer of the neural network.

$$Z = \frac{b}{\|b\|} \tag{5.4}$$

5.2 Neural Multiple Source Tracking (N-MUST) Algorithm

The training a single neural network to detect the angle of arrival of multiple sources is not an easy task. Training become exhaustive for three or four sources since the number of possible training data combinations become enormous. To circumvent this problem, multiple, but smaller, neural networks are employed. Each neural network then tracks a smaller number of sources within a smaller angular sector.

The N-MUST algorithm is composed of two stages-the detection stage and the estimation stage. In the first stage, a number of NN's are trained to perform the detection phase, while in second stage another set of networks is trained for the direction of arrival estimation phase. The NN's used are multilayer feedforward network.

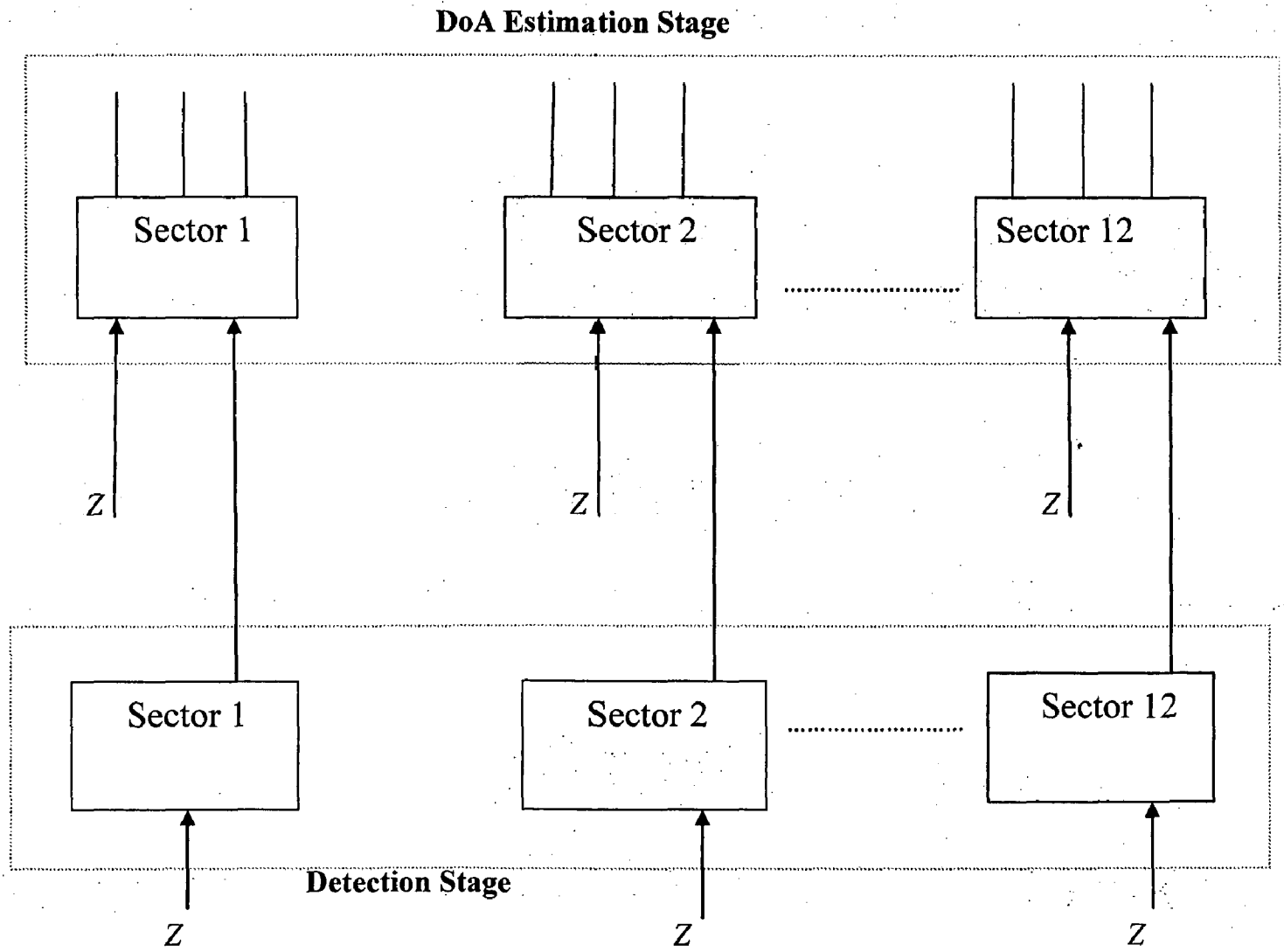

Fig. 5.1 The Neural Multiple Source Tracking Architecture [13]

5.2.1 Detection Stage

In this stage an arbitrary number of mobile users (sources) can be tracked and no prior knowledge of the number of mobile users is required. It consists of P NN's, each of width θ_w. Each one of the P NN has $M(M+1)$ input nodes representing the correlation matrix R and one output node. The number of hidden nodes in the second layer is chosen to be greater than $M(M+1)$.The entire angular spectrum is divided in P sectors. The pth NN is trained to determine if one or more signals exist within the $[(p-1)\theta_w, p\theta_w]$ sector. If there are any signals present in the corresponding sector, the neural network will give the value one for an answer. Otherwise, the network will register a zero as its output value. This information is passed to the second stage, the DoA stage, which estimates the angles of these signals.

5.2.2 DoA Estimation Stage

The second stage of NN is trained to perform the actual DoA estimation stage. The P networks of the DoA estimation stage are assigned to the same spatial sectors as in the detection stage. When the output of one or more networks from the first stage is 1, the corresponding second stage networks are activated. The input information to each second stage network is the correlation matrix R, while the output is the actual DoA of the sources. The number of hidden nodes is chosen to be greater than the number of input nodes given by $M(M+1)$.

5.3 Simulated Results

Simulations is performed based on following data,

Number of signals or desired user = 6

Number of antenna elements in array =16

Signals are assumed to be of binary Walsh type with amplitude 1.

Noise is assumed to be Gaussian distributed with variance of 0.1.

100 time samples of both signals and noise is considered.

Received signals for 100 samples are calculated from equation 2.3. Mean autocorrelation matrix is calculated from equation 2.9. Since the Autocorrelation matrix is symmetrical we will be using only the upper triangular half part of the matrix. The upper triangular part of the matrix can be organized in a 272 (16x17) dimensional vector of real and

imaginary parts denoted by b. The input vector b is normalized by its norm to obtain the normalized matrix Z.

5.3.1 Detection Stage

Entire area in the angular range of -60^0 to 60^0 in reference to the antenna array is divided into 12 small sectors each of width 10^0. Separate NN's, are employed for each sector. The NN for a particular sector will detect if any source is present in that particular sector or not. If any source is present in that sector then the NN is trained to produce a output of 1 otherwise 0. 50,000 combinations of the 6 AoA is generated by randomly taking six combinations of AoA of signal vectors from total of 121 (-60,-59,.....,-1,0,1,.....,59,60) possibilities. Corresponding to these combinations of AoA and randomly creating faults in any two or one positions of a 16 element array normalized correlation matrix Z is calculated. Normalized matrix Z is applied to the preprocessing stage of the neural network.

Preprocessing

1) REMOVECONSTANTROWS processes matrices by removing rows with constant values.
2) MAPSTD processes matrices by tranforming the mean and standard deviation of each row to YMEAN=0 and YSTD=1.
3) PROCESSPCA processes matrices using principal component analysis so that each row is uncorrelated, the rows are in the order of the amount they contribute to total variation, and rows whose contribution to total variation are less than MAXFRAC=0.02 are removed.
4) MAPMINMAX processes matrices by normalizing the minimum and maximum values of each row to [YMIN, YMAX]. YMIN=-1, YMAX=1.

The matrix Y obtained after the preprocessing stage of ANN serve as input vector for the NN.

For the 50,000 combinations of 6 AoA response vector of all the 12 NNs of Detection stage will be computed which will be 1 or 0 corresponding to the sources present in that particular sector or not. The response vector after undergoing MAPMINMAX processing function will be serving as the target vector of the NN. Training is achieved through error back propagation algorithm with the weights and biases updated using Levenberg-Marquardt algorithm. Two hidden layers are employed in the network. 20 nodes are present in the first hidden layer and 30 nodes are present in the second hidden layer. The actual output is obtained by postprocessing the output

obtained from the NN. NNs are tested with six sources separated by 1^0 and randomly creating faults at any two or one position of 16 element array. For a particular sector if any of the six source is situated in that sector the output of the NN corresponding to that sector comes close to 1 otherwise nearly 0. In the Fig (5.2) to (5.4) we are comparing the output of the network in detection stage to the desired output for the sectors (0^0-10^0), (-60^0- -50^0) and (40^0-50^0) for different combination of faults in the 16 element array.

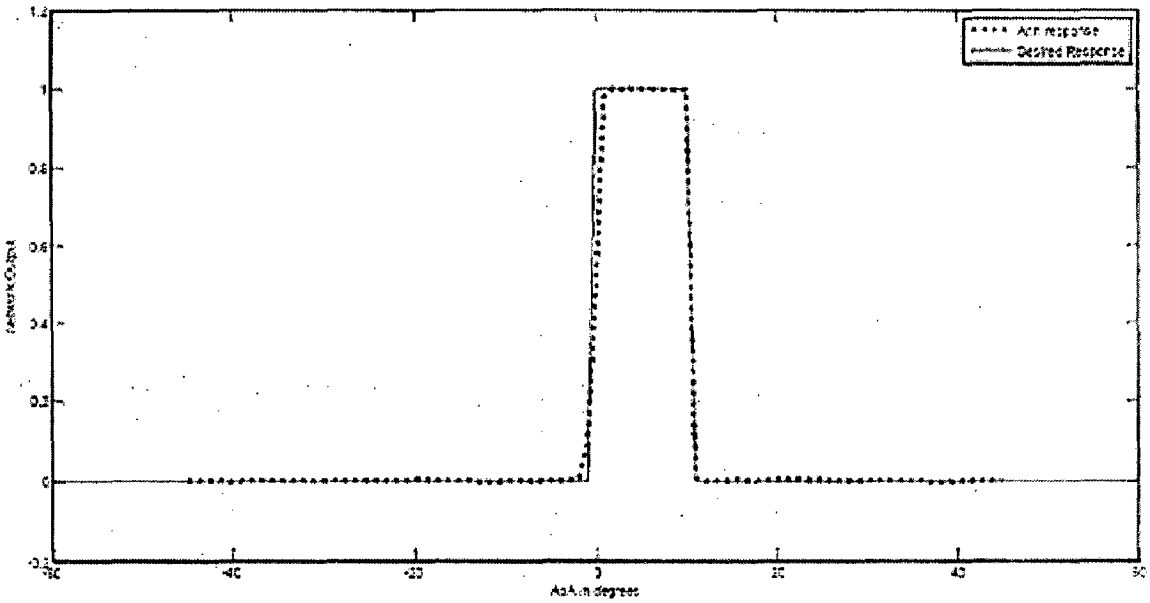

Fig 5.2 Comparison between Network Output and desired response for sources lying in (0^0-10^0) estimated with array having fault at 7^{th} and 8^{th} position

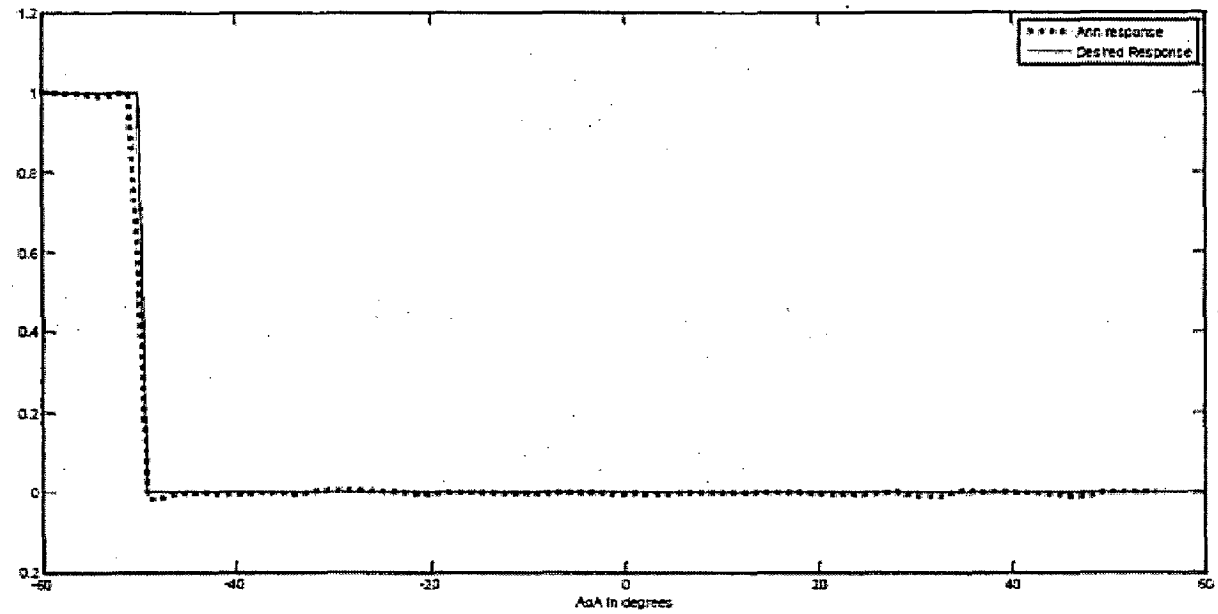

Fig 5.3 Comparison between Network Output and desired response for sources lying in (-60^0- -50^0) estimated with array having fault at 8^{th} position

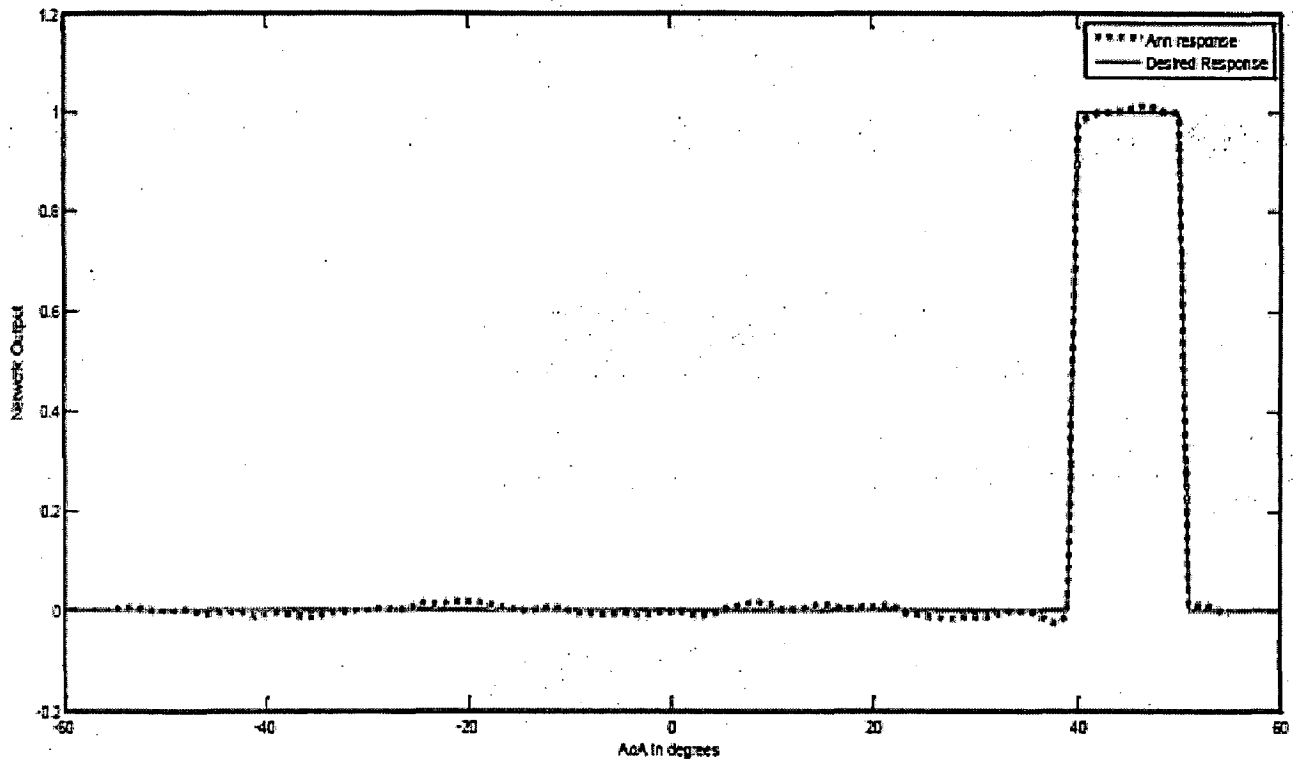

Fig 5.4 Comparison between Network Output and desired response for sources lying in (40^0- 50^0) estimated with array having fault at 4^{th} and 13^{th} position.

5.3.2 Estimation Stage

The second stage networks are trained to perform the actual DoA estimation. The 12 networks of the DoA estimation stage are assigned to the same spatial sectors as in the detection stage. When the output of one or more networks from the first stage is 1, the corresponding second stage networks are activated. The input information to each second stage network is same as the input of the detection stage network which is normalized correlation matrix *Z*, while the output is the actual DoA of the sources. The number of output nodes in Estimation Stage NN is 3. So each NN in estimation stage can estimate DoA of a maximum of 3 sources present in that sector. For the purpose of training a NN corresponding to a particular sector 23000 combinations of 6 AoA are generated. 4000 combinations correspond to 1 AoA from the desired sector and rest 5 from the other sectors. 7000 combinations correspond to 2 AoA from the desired sector and rest 4 from the other sectors. 12000 combinations correspond to 3 AoA from the desired sector and rest 3 from the other sectors. Faults are created randomly in any two and one positions of

a 16 element array. Corresponding to 23000 combinations of AoA and randomly creating 2 and 1 element faults in a 16 element array, correlation matrix R is calculated. Inputs to the NN are generated by applying the four preprocessing functions which are outlined in the detection stage to the normalized correlation matrix. Target pairs of the NN for last 12000 pairs of AoA is the three AoA which are present in that sector. For the other pairs of AoA, output vector is AoA lying in that sector and 10^0 more than the maximum limit of the sector. For a 10^0-20^0 sector with one source lying in that sector output vector will comprise of AoA of source lying in that sector and 30^0,30^0 as the rest of elements. The target pairs are preprocessed by using the mapminmax function before presenting them to the NN. Training is achieved through error back propagation algorithm with the weights and biases updated using Levenberg-Marquardt algorithm [22]. Two hidden layers are employed in the network. 20 nodes are present in the first hidden layer and 30 nodes are present in the second hidden layer. The actual output of the NN is obtained after post processing the NN output using mapminmax function with same settings.

In the linear plots between Figures (5.2) to (5.13) a comparison is drawn between output obtained from the network in DoA Estimation stage and the desired output for three cases- single source present in the desired sector, two sources present in the desired sector and three sources present in the desired sector. 110, 100, 90 tests are performed for single source, two sources and three sources present in the desired sector. For multiple sources in the sector testing is done with 1^0 angular separations among the sources present in the sector. Root mean square error is calculated for the testing data set and is shown beneath the Fig.

The histogram plots from (5.14) to (5.25) shows the distribution of errors between the expected and estimated SoI DoA on carrying 2000 tests with the trained network for different sectors for a particular number of sources in a desired sector which can be one, two or three. For multiple sources in the sector testing is done with different angular separations among the sources present in the sector ranging from 1^0,2^0 upto 8^0.

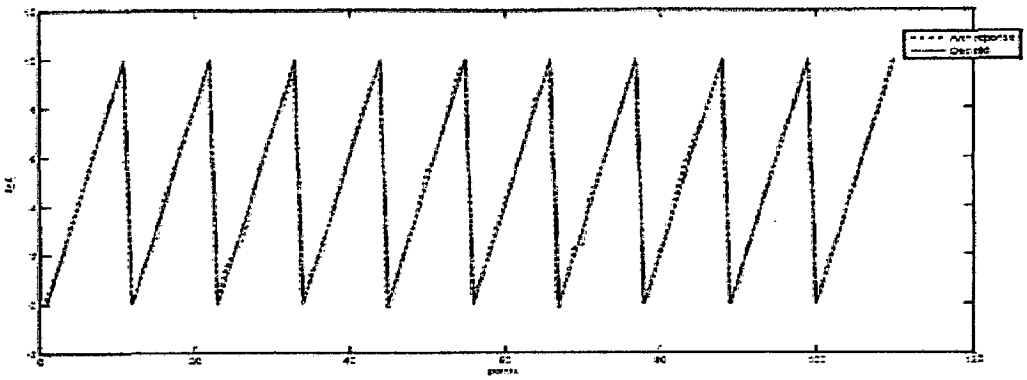

Fig 5.5 Comparison between Network Output and Desired response for one Source lying in (0^0- 10^0) estimated with array having fault at 3rd and 13th position. A RMSE of 0.1945 is obtained.

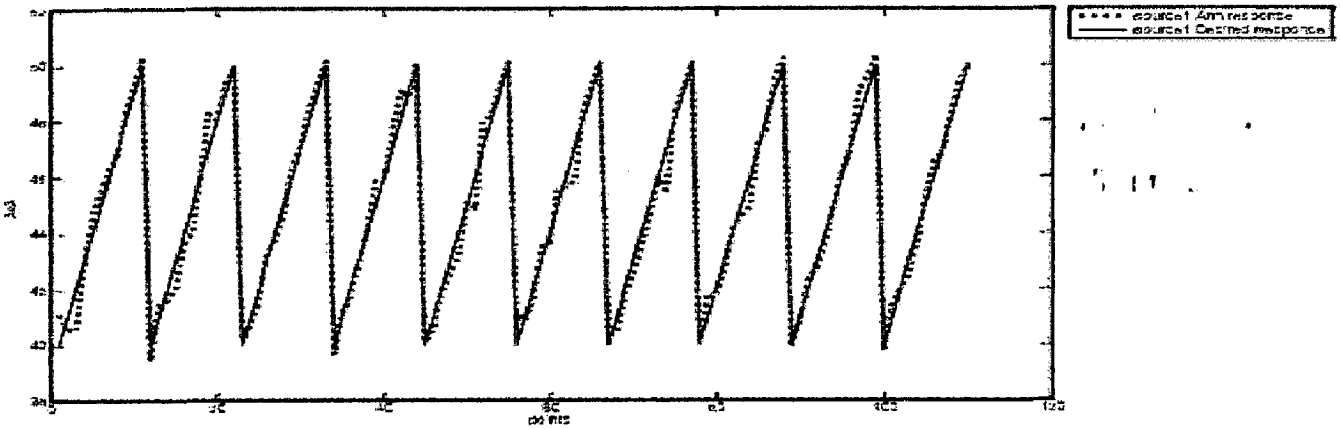

Fig 5.6 Comparison between Network Output and Desired response for one source lying in (40^0-50^0) estimated with array having fault at 8th and 9th position. A RMSE of 0.5794 is obtained in the estimation of the source.

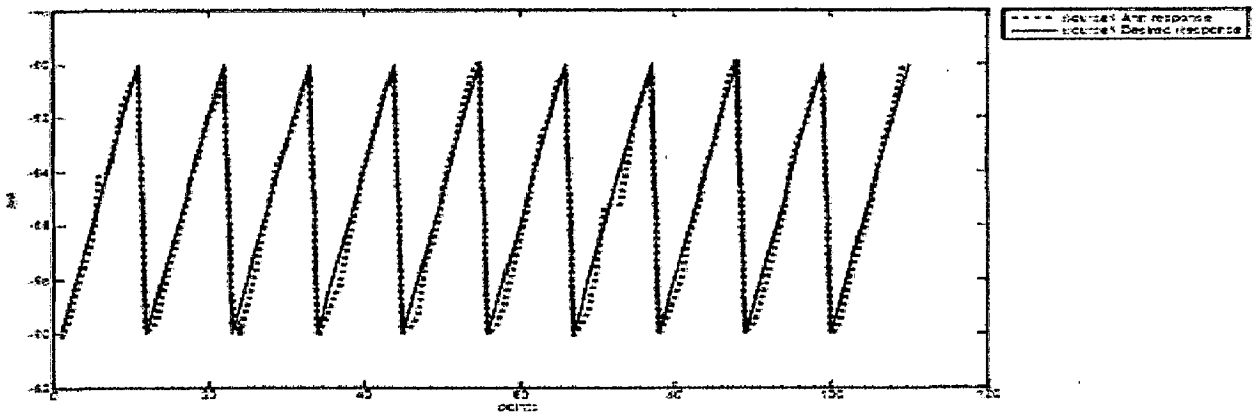

Fig 5.7 Comparison between Network Output and Desired response for one source lying in (-60^0- -50^0) estimated with array having fault at 7th and 9th position. A RMSE of 0.5712 is obtained in the estimation of source.

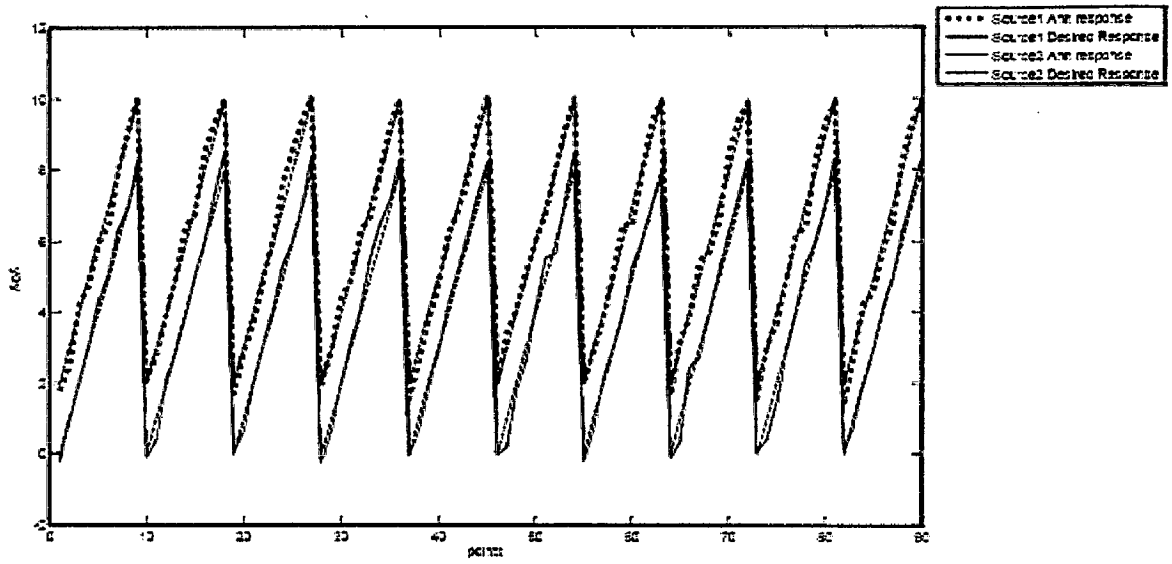

Fig 5.8 Comparison between Network Output and Desired response for two sources lying in (0^0- 10^0) estimated with array having fault at 7^{th} and 8^{th} position. A RMSE of 0.3011 and 0.275 is obtained in the estimation of first and second sources.

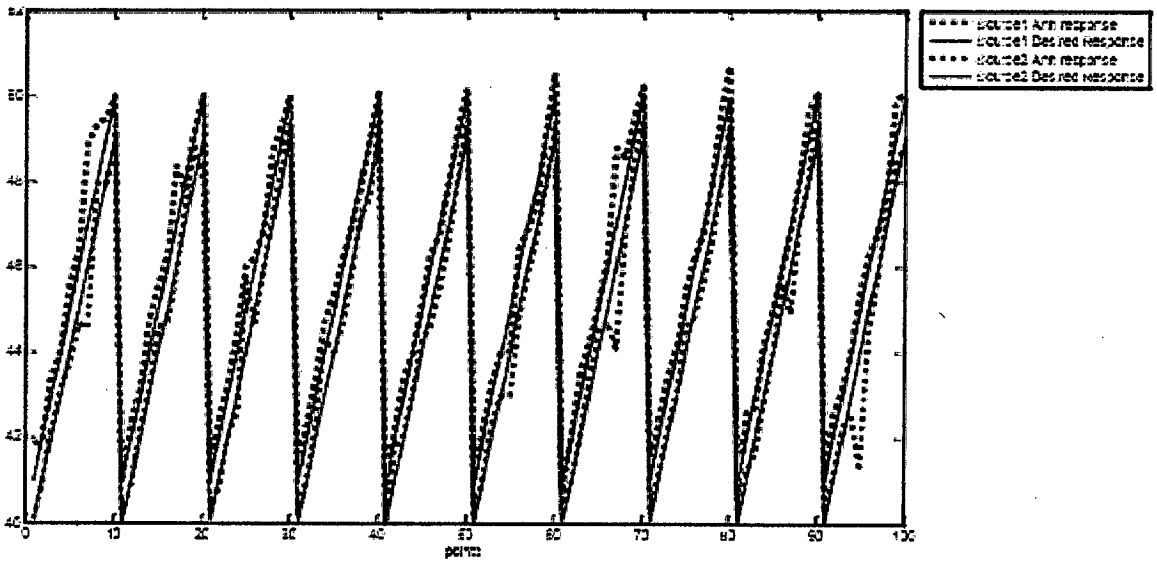

Fig 5.9 Comparison between Network Output and Desired response for two sources lying in (40^0-50^0) estimated with array having fault at 5^{th} and 13^{th} position. A RMSE of 0.5283, 0.6031 is obtained in the estimation of first and second sources.

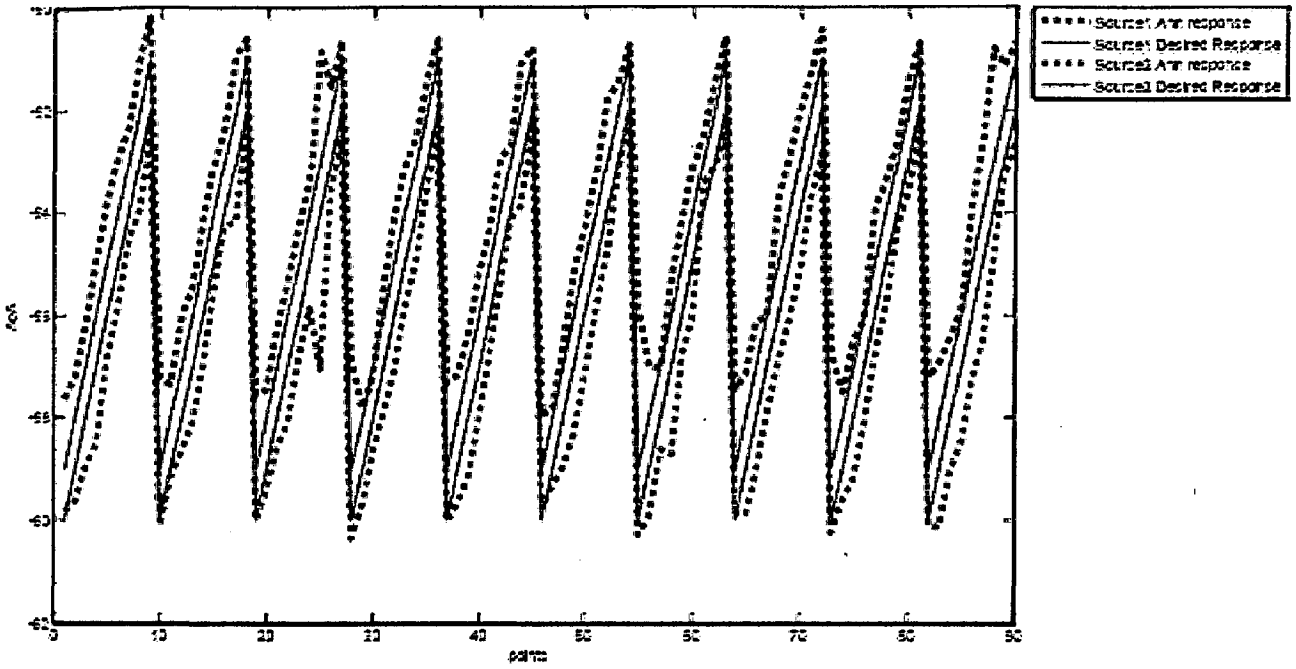

Fig 5.10 Comparison between Network Output and Desired response for two source lying in (-60^0- -50^0) estimated with array having fault at 3^{rd} and 13^{th} position. A Rmse of 1, 0.85 is obtained in the estimation of sources.

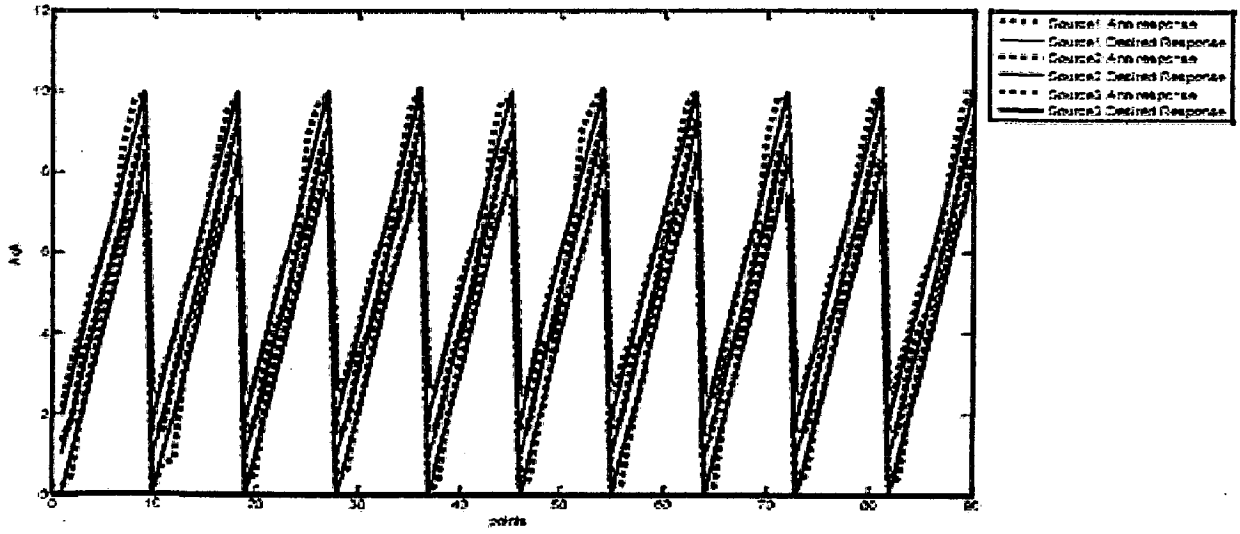

Fig 5.11 Comparison between Network Output and Desired response for three sources lying in (0^0- 10^0) estimated with array having fault at 8^{th} position. A Rmse of 0.3848, 0.2695, 0.3478 is obtained in the estimation of first, second and third sources.

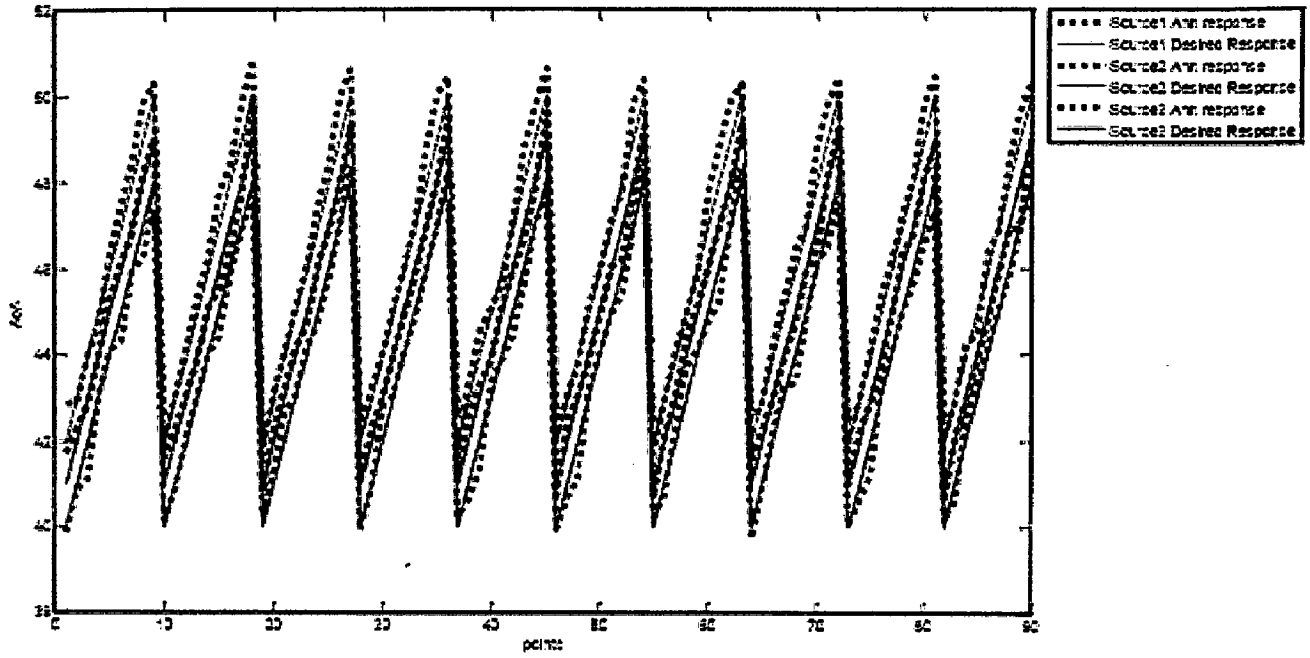

Fig 5.12 Comparison between Network Output and Desired response for three sources lying in (40^0- 50^0) estimated with array having fault at 4^{th} and 13^{th} position. A Rmse of 0.467, 0.3195, 0.3878 is obtained in the estimation of first, second and third sources.

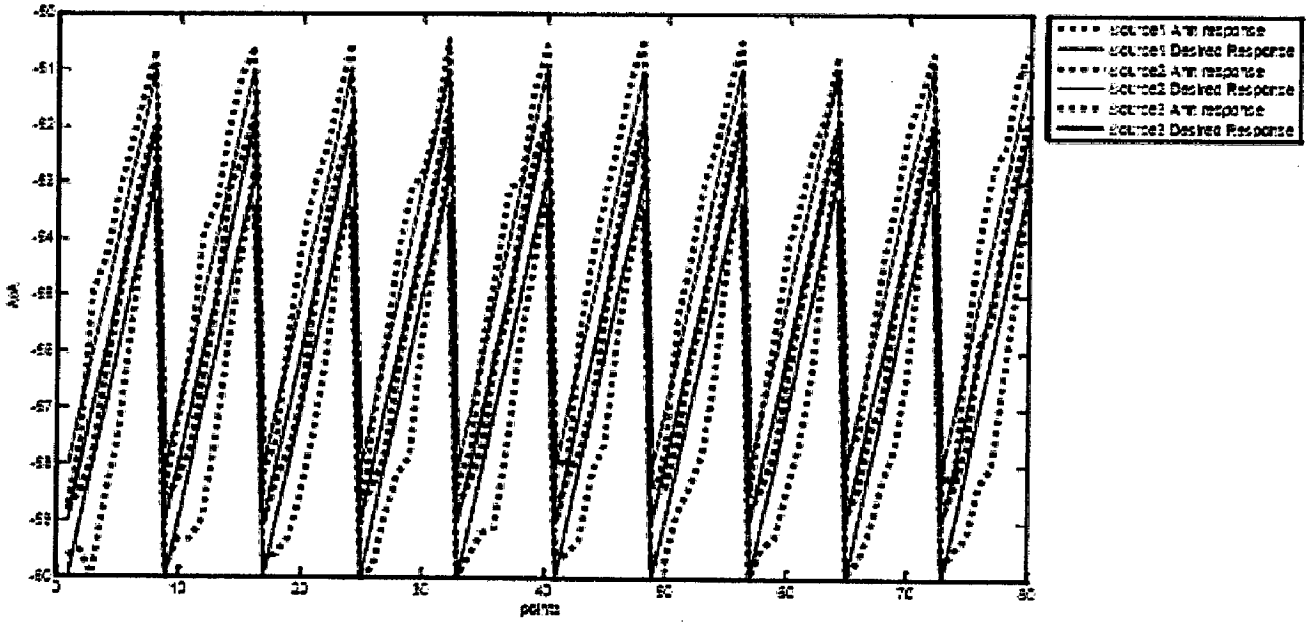

Fig 5.13 Comparison between Network Output and Desired response for three source lying in (-60^0- -50^0) estimated with array having fault at 7^{th} and 8^{th} position. A Rmse of 0.65, 0.39, 1.03 is obtained in the estimation of first, second and third sources.

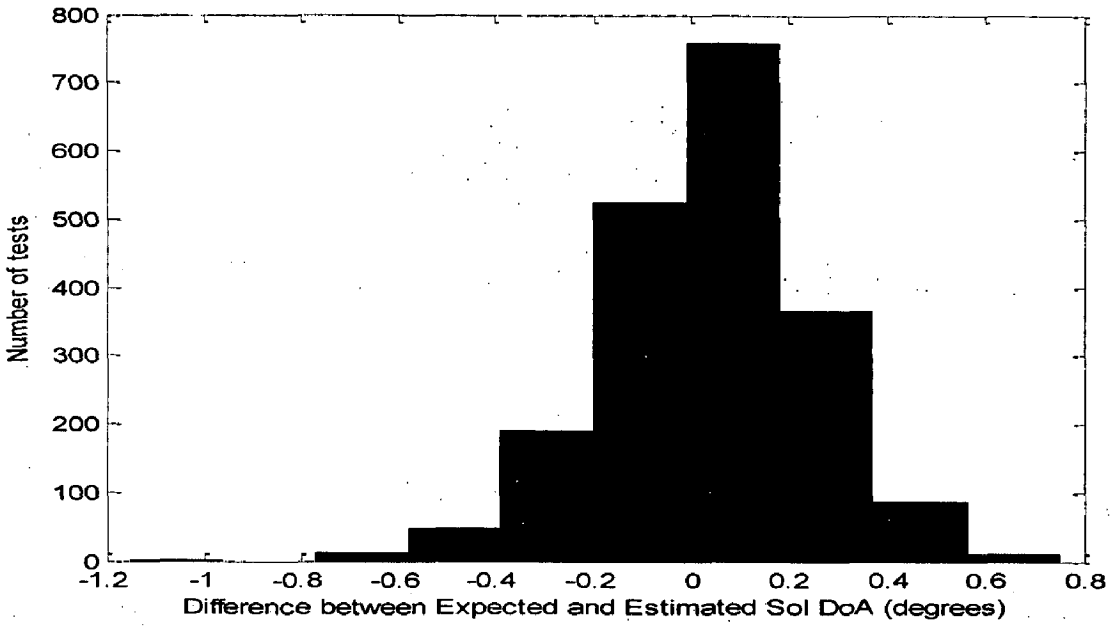

Fig 5.14 Distribution of difference between Expected and Estimated SoI DoA (degrees) for a single source lying in sector (0-10) estimated with array having fault at 4^{th} and 11^{th} position. RMSE=0.2165

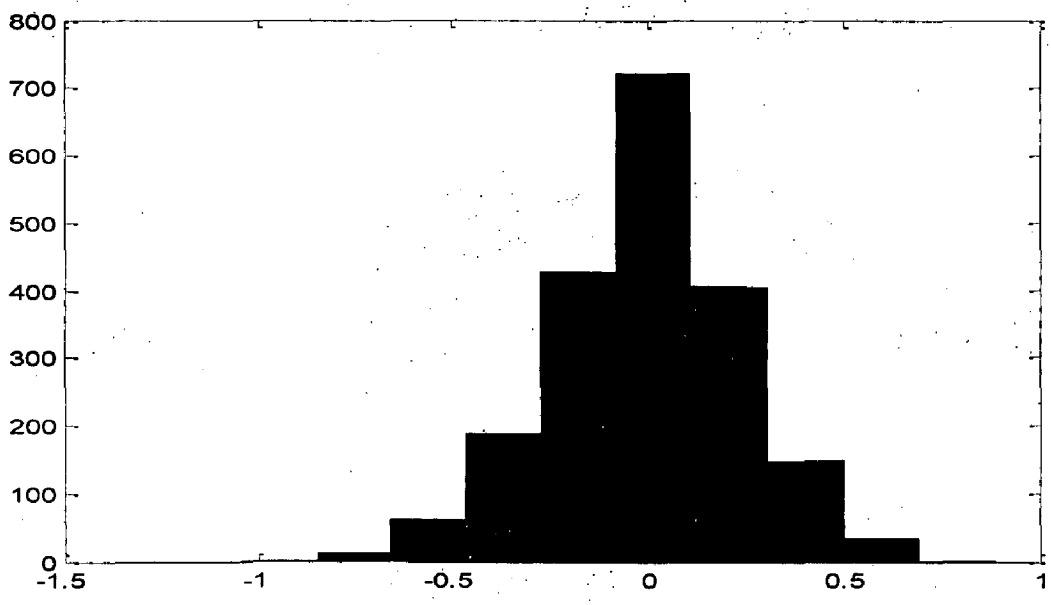

Fig 5.15 Distribution of difference between Expected and Estimated SoI DoA (degrees) for a single source lying in sector (0-10) estimated with array having fault at 6^{th} and 9^{th} position. RMSE=0.237

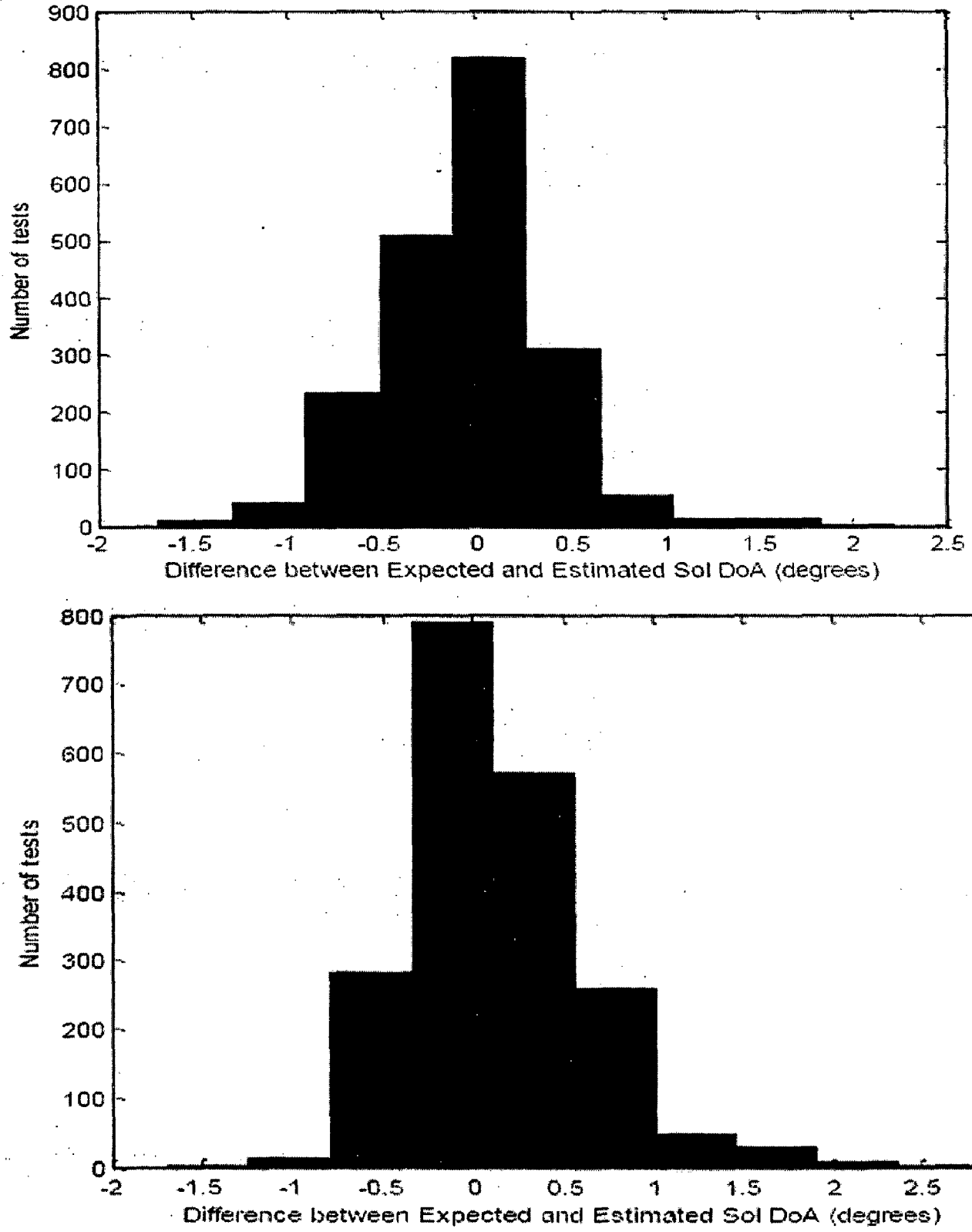

Fig 5.16 Distribution of difference between Expected and Estimated SoI DoA (degrees) for two sources lying in sector (0-10) estimated with array having fault at 7th and 8th position. RMSE1=0.4330, RMSE2=0.4987

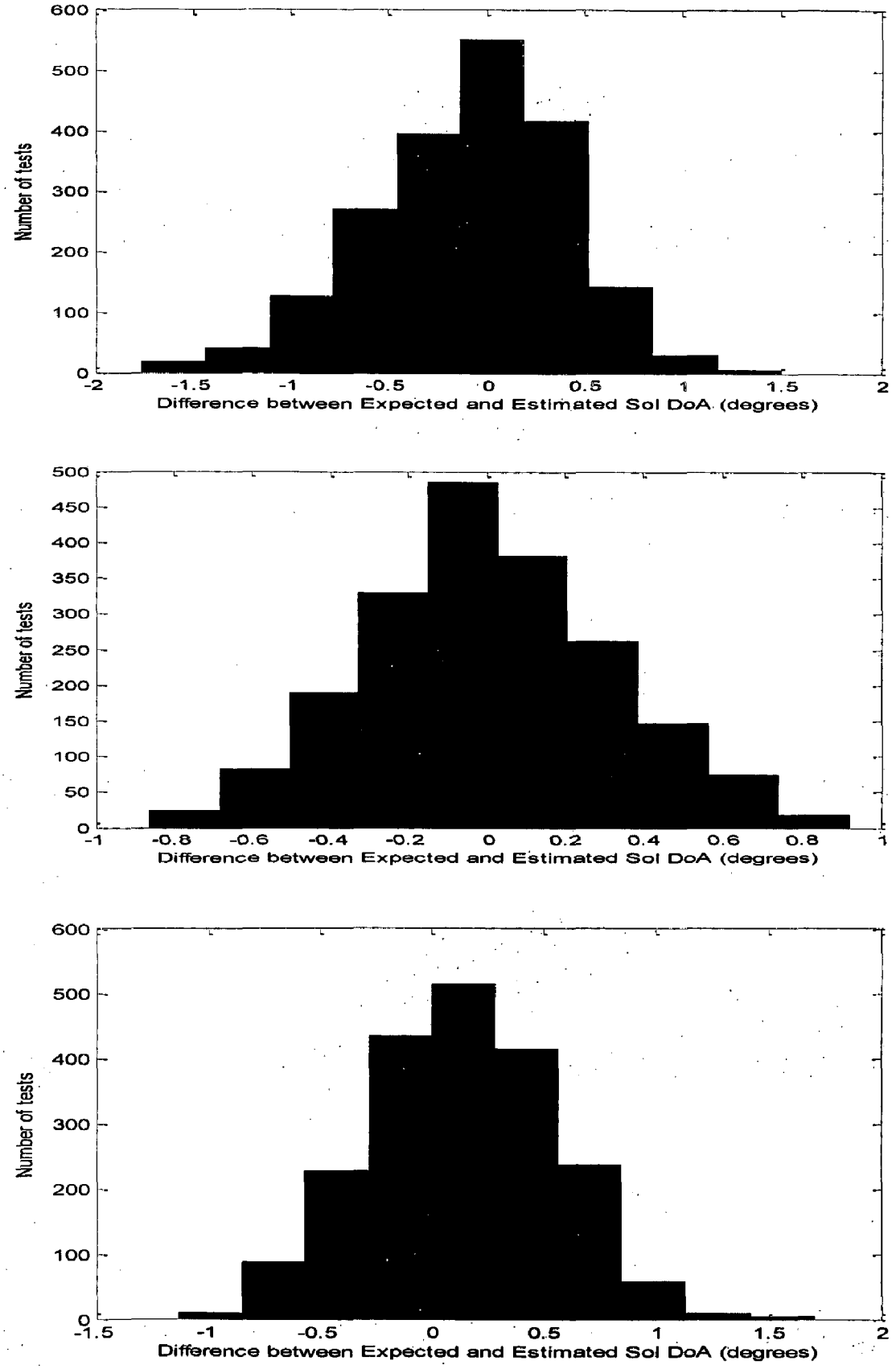

Fig 5.17 Distribution of difference between Expected and Estimated SoI DoA (degrees) for three sources lying in sector (0-10) estimated with array having fault at 3rd and 8th position.RMSE1=0.5038, RMSE2=0.3148,RMSE3=0.4368

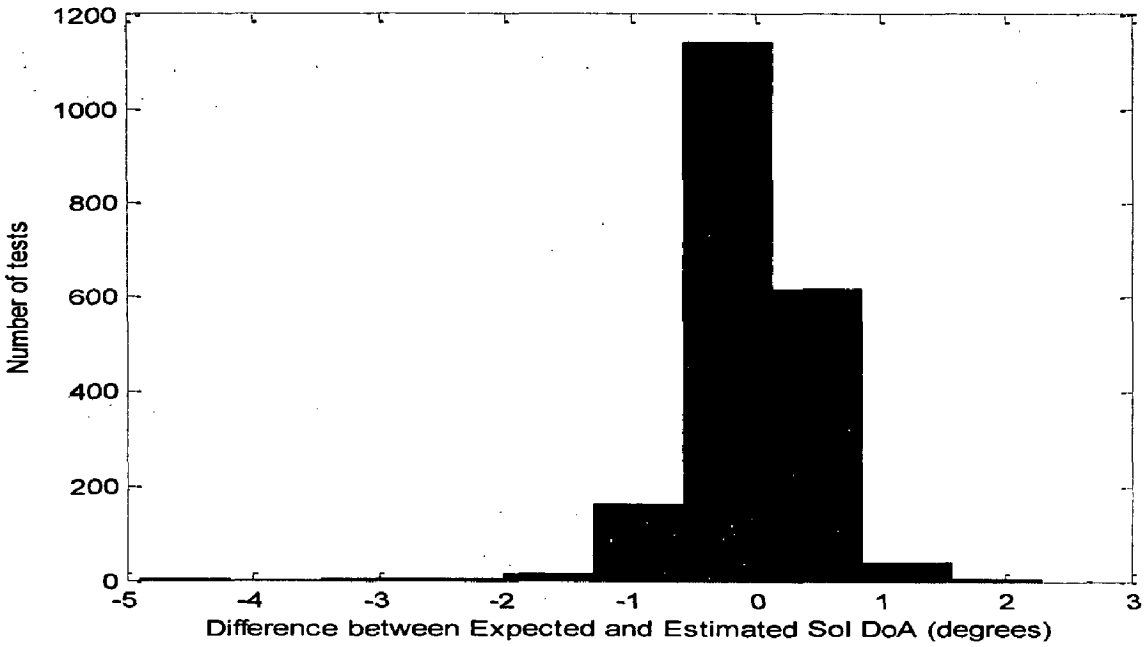

Fig 5.18 Distribution of difference between Expected and Estimated SoI DoA (degrees) for a single source lying in sector (40-50) estimated with array having fault at 7^{th} and 10^{th} position. RMSE=0.5289

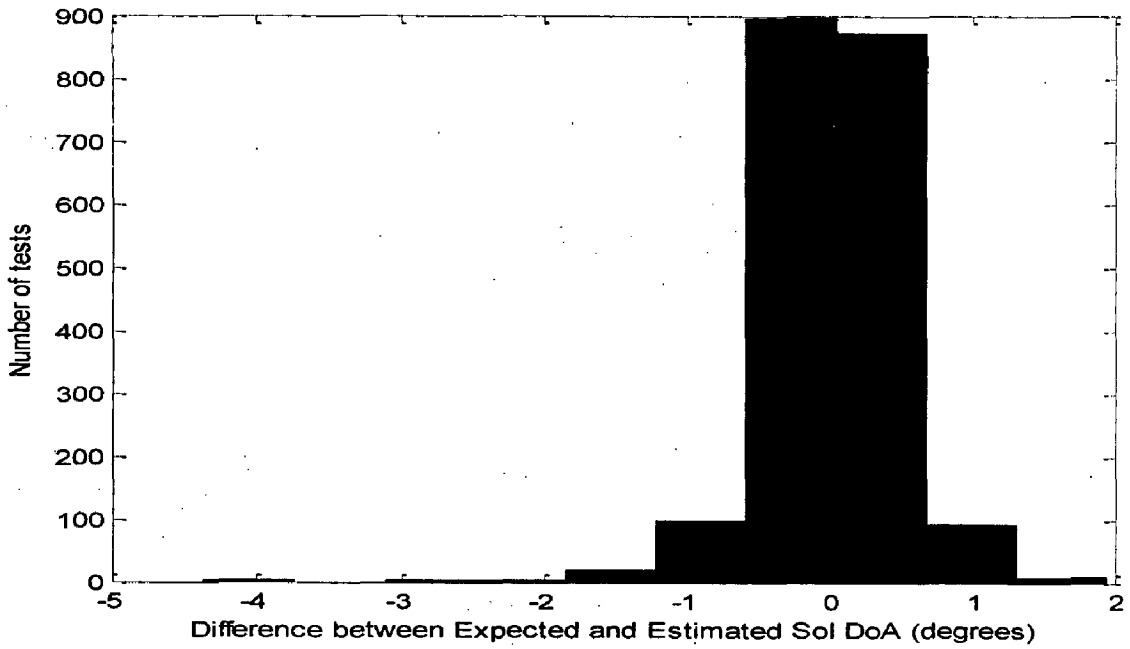

Fig 5.19 Distribution of difference between Expected and Estimated SoI DoA (degrees) for a single source lying in sector (40-50) estimated with array having fault at 5^{th} and 12^{th} position. RMSE=0.456

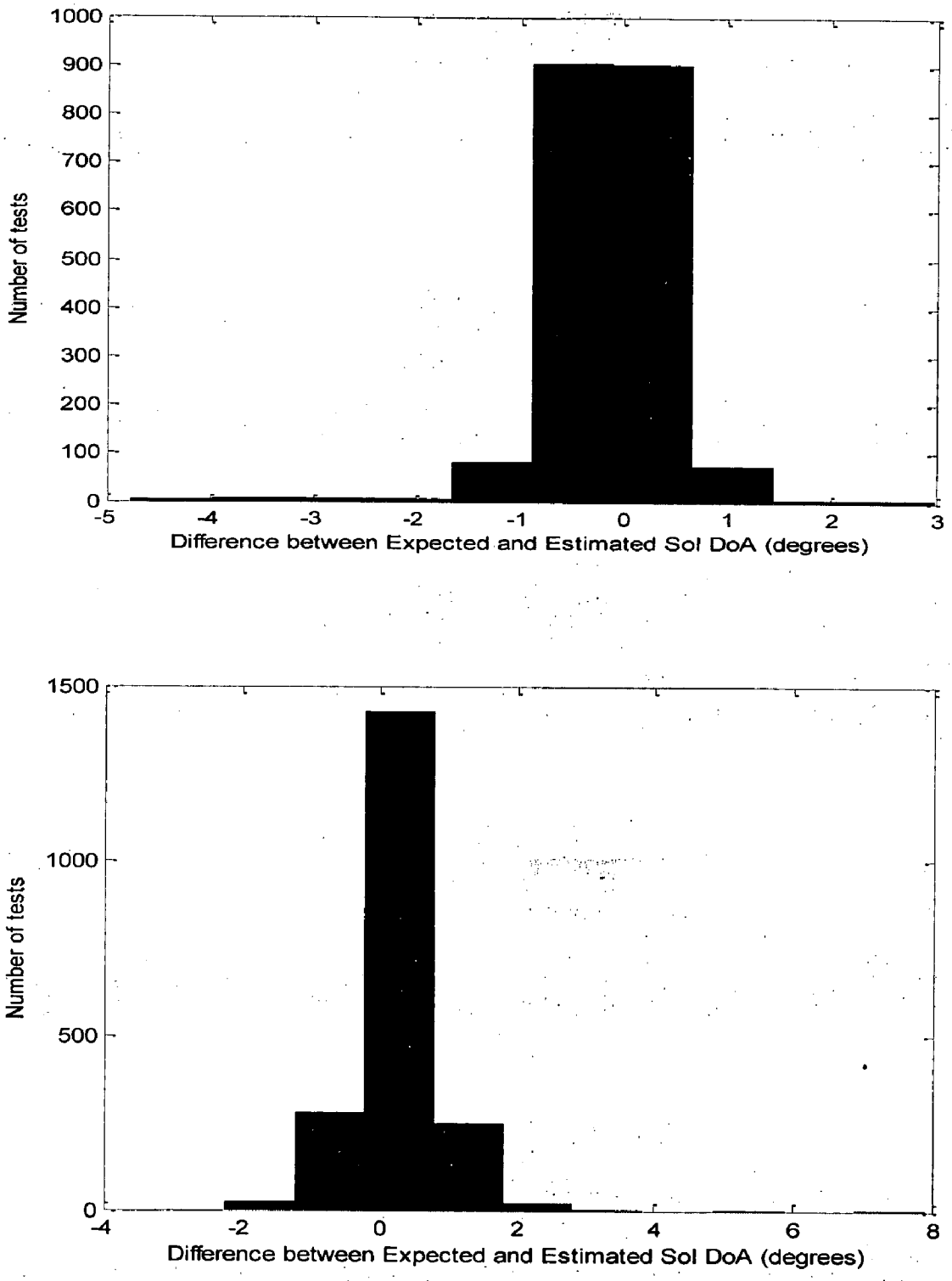

Fig 5.20 Distribution of difference between Expected and Estimated SoI DoA (degrees) for two sources lying in sector (40-50) estimated with array having fault at 7^{th} and 9^{th} position. RMSE1=0.5951, RMSE2=0.6682

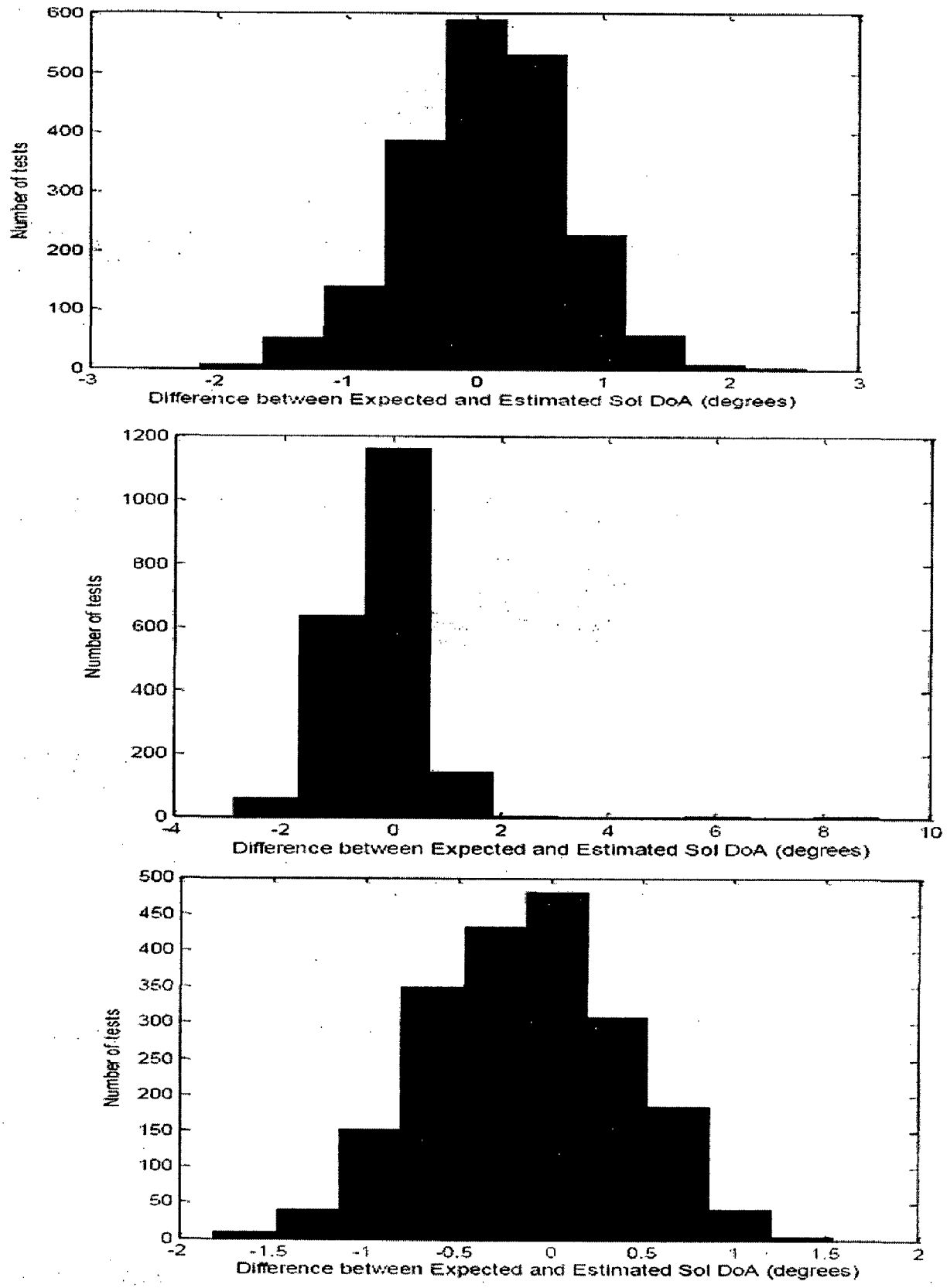

Fig 5.21 Distribution of difference between Expected and Estimated SoI DoA (degrees) for three sources lying in sector (40-50) estimated with array having fault at 3rd and 14th position RMSE1=0.6238, RMSE2=0.5435,RMSE3=0.8091

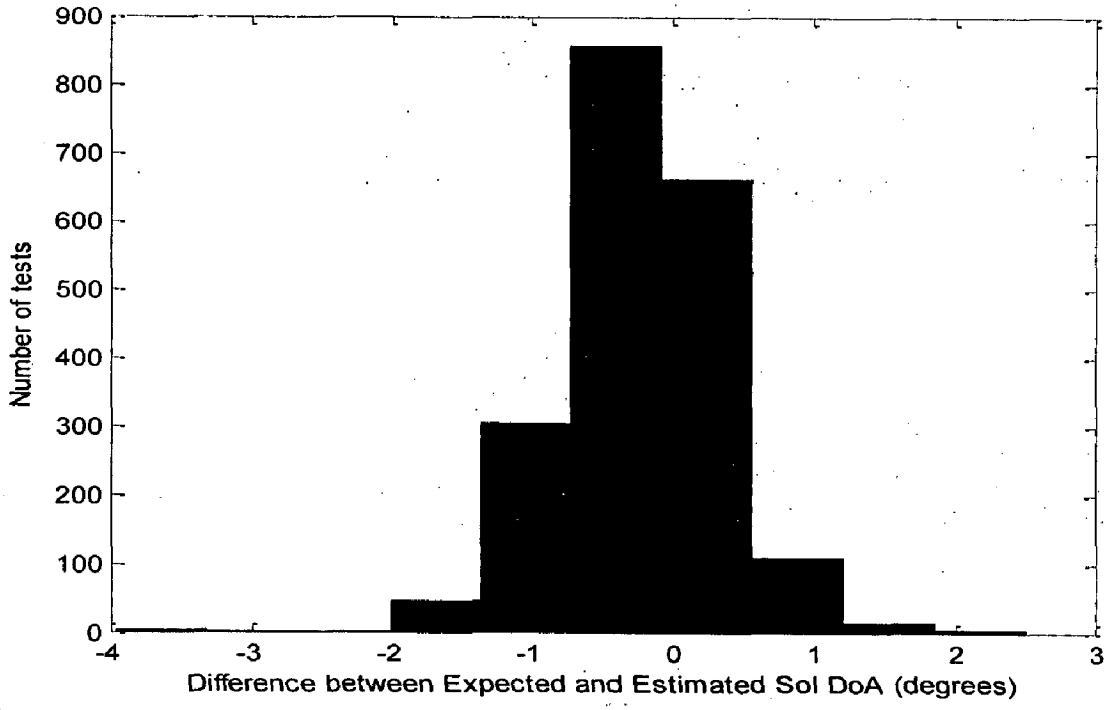

Fig 5.22 Distribution of difference between Expected and Estimated SoI DoA (degrees) for a single source lying in sector (-60 - -50) estimated with array having fault at 6^{th} and 10^{th} position. RMSE=0.6192.

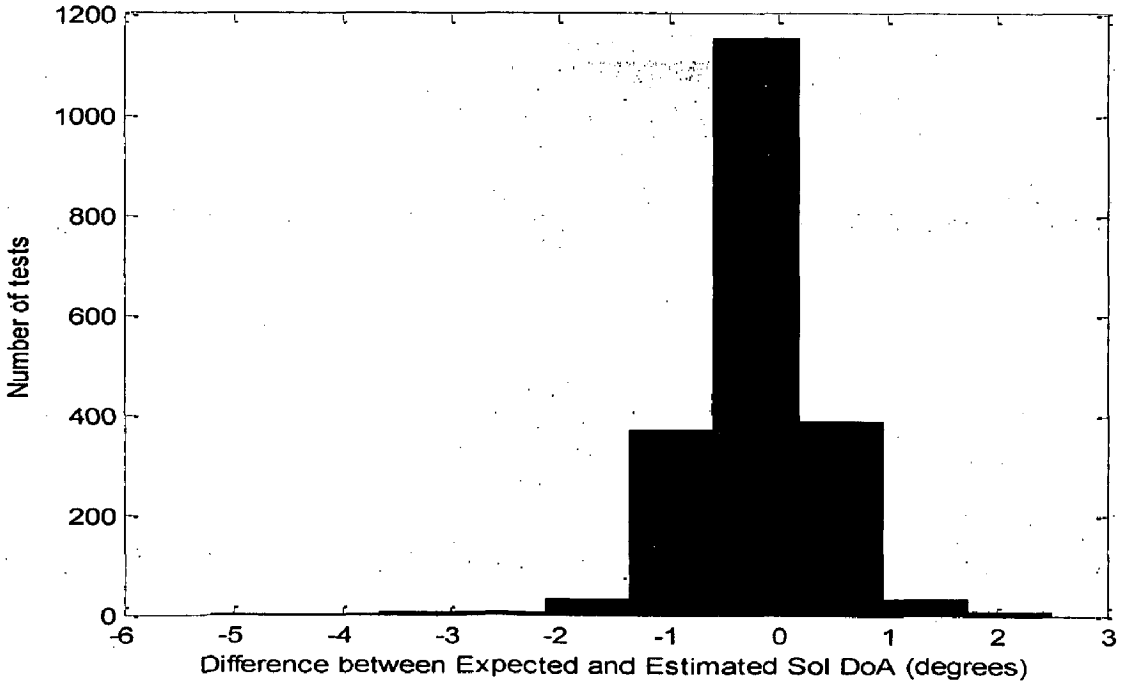

Fig 5.23 Distribution of difference between Expected and Estimated SoI DoA (degrees) for a single source lying in sector (-60 - -50) estimated with array having fault at 5^{th} and 9^{th} position. RMSE=0.6309.

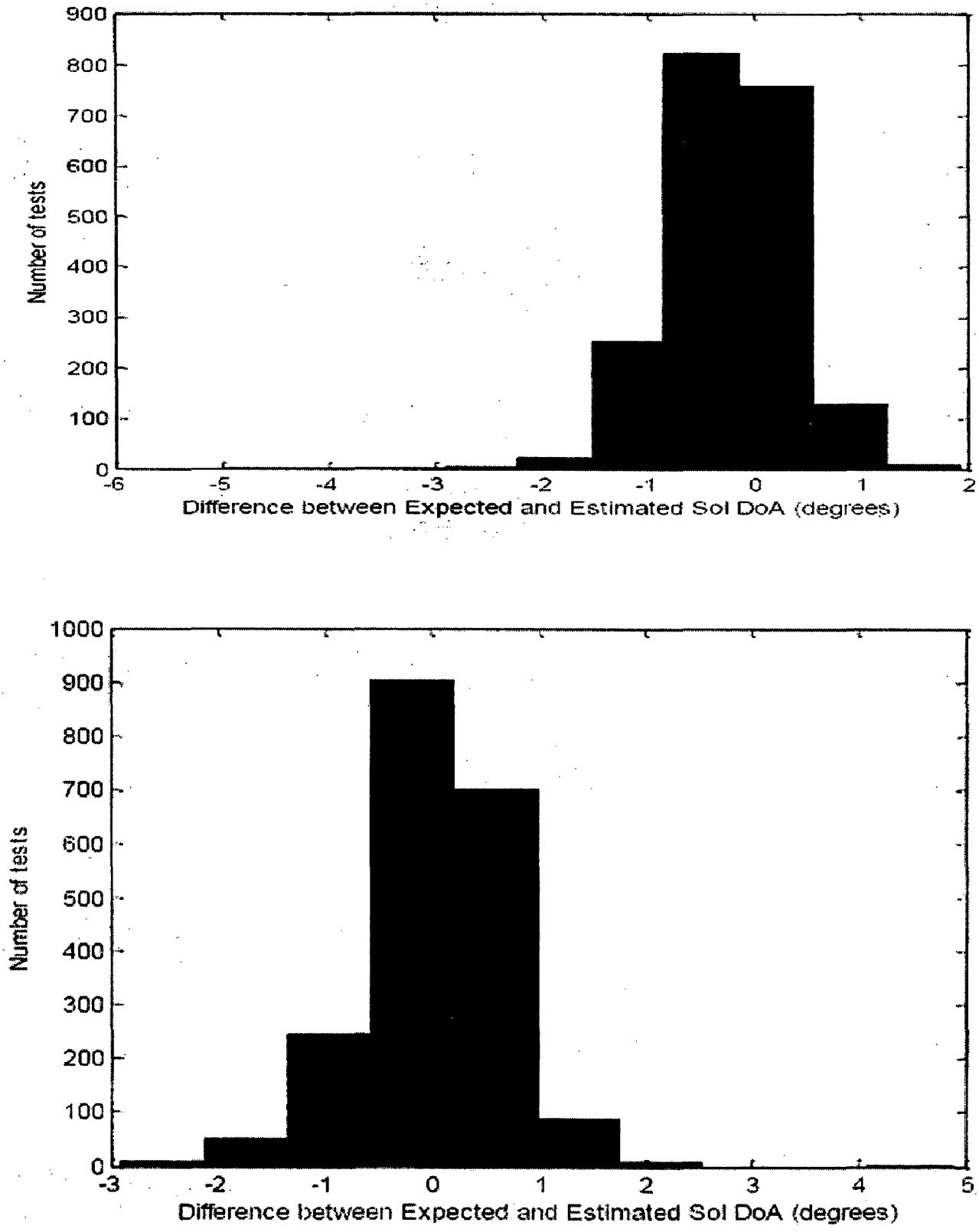

Fig 5.24 Distribution of difference between Expected and Estimated SoI DoA (degrees) for two sources lying in sector (-60 - -50) estimated with array having fault at 6th and 11th position. RMSE1=0.6229, RMSE2=0.6437.

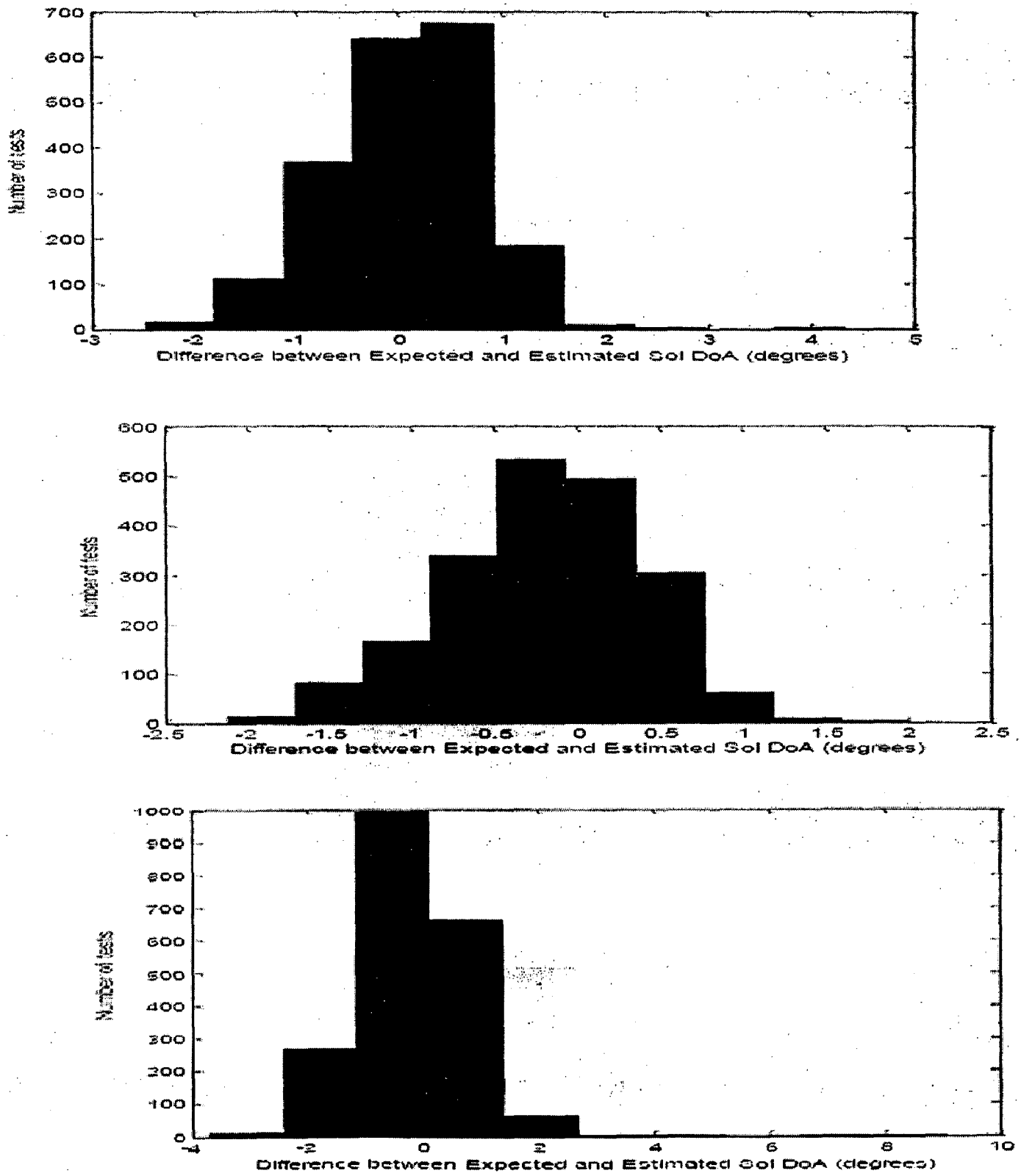

Fig 5.25 Distribution of difference between Expected and Estimated SoI DoA (degrees) for three sources lying in sector (-60 - -50) estimated with array having fault at 7th and 8th position. RMSE1=0.7221, RMSE2=0.6249, RMSE3=0.923

6. Conclusions

The popular analytical methods of DoA estimation require intensive signal processing at every antenna element and also their performance degrades in case of correlated sources. So, their penetration to commercial smart antenna applications seems to be complicated. The method proposed in chapter 4 namely NN based DoA estimation in DS-CDMA, needs only received power measurements for each beam and their mapping, through NNs, with the signals' angles of arrival. It has been concluded that the set of seven beams, produced by the cosine illumination of the antenna elements, provides significantly better results compared to the set of eight orthogonal beams produced by the uniform illumination. In both cases the feeding network has been a 8x8 Butler matrix. But, still in some cases, the uniform illumination is preferred due to the orthogonal property of beams. The sector beam can be used to serve the common communication channels and can thus enhance the method's efficiency. With the use of sector beam the desired signal's DoA can be extracted with a less than one degree RMSE, even if its power level is 6 dB down from each one of 39 interfering signals. This configuration is very robust as the same NN can work for any signal, independently of the signals' amplitudes and the total number of mobile users.

As the SeIR gets lower, the DoA estimation of the desired signal at the edges of the sector fail as the beam coverage is weaker. The method needs further improvement in order to cope better with this, point. To sum up, the proposed methodology can be easily implemented into existing base stations, determining the smarter operation of a SBS based on the DoA information. The method's extension to DS-CDMA gives comparable results with the MUSIC algorithm at less complexity and greater speed.

The method proposed in chapter 5 namely DoA Estimation in a failed antenna array, is based on N-MUST algorithm. DoA is estimated in two stages-Detection Phase and DoA Estimation Phase. Regions at angular locations from -60^0 to 60^0 with respect to antenna array is considered for the purpose of DoA Estimation. This area of 120^0 is divided into 12 regions each of width 10^0. In the first stage, corresponding to each small

area we had employed two separate NNs – One for detecting, whether any sources are present in that particular sector or not and the other one for estimating the exact location of the source, if the source is present in that sector. So, in total 12 NN were employed in detection phase and another 12 were used in DoA Estimation phase. This algorithm was used to estimate the direction of arrival of six radiating sources. The antenna array used comprised of 16 elements. Neural Networks employed in detection and estimation phases were trained considering all possible cases in which a maximum of two elements in the antenna array can suffer faults. The proposed scheme does not need the knowledge of the position of failed array elements and work well for both correlated and uncorrelated sources.

The output of DoA estimation stage was different for different sectors. Root mean square error between the actual DoA and that estimated from the network varies for different sectors. Error is least for (0^0-10^0) and (-10^0-0^0) sector and is maximum for (50^0-60^0) and (-60^0- -50^0) sector. But for all sectors a RMSE within 1.5^0 was observed. Errors is more at the edges of sector as compared to the interior points within the sector.

7. Future Scope of Work

All the implementations of the NN were done at software level. Because, efforts have already made in hardware implementation of NNs [27], so if the developed trained NNs could be implemented in hardware, that can be used in cell sites/base stations for both DoA estimation and beamforming simultaneously.

For real-time operation, efforts should be made to reduce the processing time further.

[1] L.C. Godara, Smart Antennas. Boca Raton: CRC Press, 2004.

[2] Frank Gross, Smart Antennas For Wireless Communications With Matlab : McGraw-Hill,2005.

[3] R. O. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Trans Antennas Propag.*, vol. 34, no. 3, pp. 276–280, Mar.1986.

[4] R. Ray and T. Kailath, "ESPRIT-Estimation of Signal Parameter via Rotational Invariance Techniques," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP- 37, pp. 984–995, 1989.

[5] J.-T. Kim, S.-H. Moon, D. S. Han, and M.-J. Cho, "Fast DOA estimation algorithm using pseudocovariance matrix," *IEEE Trans. Antennas and Propag.*, vol. 53, no. 4, pp. 1346– 1351, Apr. 2005.

[6] J. Koh and T. K. Sarkar, "High resolution DOA estimation using matrix pencil," in *Proc. IEEE Antennas and Propag. Soc. Int. Symp.*, Jun. 2004, vol. 1, pp. 423–426.

[7] Simon Haykin,Neural Networks : A Comprehensive Foundation Second Edition : Practise Hall International, Inc.1999.

[8] Theodore Rappaport, Wireless Communications Second Edition : PHI Learning Private Limited,2008.

[9] K. Gotsis, T. Kaifas, K. Siakavara, and J. Sahalos, "Direction of Arrival (DoA) Estimation for a switched-beam DS-CDMA system using neural networks," in *Proc. 19th Int. Conf. Appl. Electromagn. Commun.*, Dubrovnik, Croatia, Sep. 24–26, 2007, pp. 47–50.

[10] Konstantinos A. Gotsis, Katherine Siakavara, and John N. Sahalos, "On the DoA Estimation for a Switched-Beam Antenna System Using Neural Networks", *IEEE Transactions On Antennas And Propagation,* vol.57,NO. 5,May 2009.

[11] K.A. Gotsis, E.G. Vaitsopoulos, K. Siakavara, and J.N. Sahalos, "Multiple Signal DoA Estimation for a Switched-Beam System Using Neural Networks, " *PIERS Proceeding,* August 27-30, Prague, Czech Republic, 2007.

[12] A. H. El Zooghby, C. G. Christodoulou, and M. Georgiopoulos, "Performance of radial basis function networks for direction of arrival estimation with Antenna

Arrays," *IEEE Trans. Antennas Propagat.*, vol. 45, pp. 1611–1617, Nov. 1997.

[13] Ahmed H. EI Zooghby, "A Neural Network-Based Smart Antenna For multiple Source Tracking", *IEEE Trans. On Antennas And Propagation,* vol.48 , No.5, May 2000.

[14] K.-L. Du, A. K. Y. Lai, K. K. M. Cheng, and M. N. S. Swamy, "Neural methods for antenna array signal processing:A review," *Signal Proces.*,vol. 82, pp. 547–561, 2002.

[15] Y.Kuwahara and T. Matsumoto, "Experiments on Direction Finder Using RBF neural network with post-processing," *ELECTRONICS LETTERS* 12th May 2005 vol.41 NO.10

[16] S.H. Zainud-Deen, H.A. Malhat, K.H. Awadalla and E.S EL-Hadad," Direction of Arrival And State Of Polarization Estimation Using RBFNN," *Progress In Electromagnetics Research B,* Vol. 2, 137-150,2008

[17] S. Vigneshwaran, Narasimhan Sundararajan, and P. Saratchandran, "Direction of Arrival (DoA) Estimation Under Array Sensor Failures Using a Minimal Resource Allocation Neural Net ", *IEEE Transactions On Antennas And Propagation*, vol. 55, no. 2, February 2007.

[18] E. G. Larsson and P. Stoica, "High-resolution direction finding: The missing data case," *IEEE Trans. Signal Processing*, vol. 49, no. 5, pp.950–958, May 2001.

[19] T. W. Pirinen, J. Yli-Hietanen, P. Pertila, and A.Visa, "Detection and compensation of sensor malfunction in time delay based direction of arrival estimation," in *Proc. Int. Symp. on Circuits and Systems, ISCAS'04*, May 2004, vol. 4, no. 23–26, pp. IV-872–875.

[20] B.-K. Yeo and Y. Lu, "Array failure correction with a genetic algorithm," *IEEE Trans. Antennas Propag.*, vol. 47, no. 5, pp. 823–828, May 1999.

[21] H. Demuth and M. Beale, *Neural Network Toolbox for Use With MATLAB, User's Guide (Seventh Edition)*. Natick, MA: The MathWorks, Inc., 2010.

[22] Martin T. Hagan and Mohammad B. Menhaj, "Training Feedforward Networks with the Marquardt Algorithm", *IEEE Transactions On Neural Networks*, vol. *5,* no. *6, N*ovember 1994.

[23] T. N. Kaifas and J. N. Sahalos, "On the design of a single-layer wideband Butler Matrix for switched-beam UMTS system applications," *IEEE Antennas Propag. Mag.*, vol. 48, no. 6, pp. 193–204, Dec. 2006.

[24] Mourad Nedil, Tayeb A. Denidni, and Larbi Talbi, "Novel Butler Matrix Using CPW Multilayer Technology," *IEEE Transactions On Microwave Theory And Techniques*, vol. 54, no. 1, January 2006.

[25] C. A. Balanis, Antenna Theory: Analysis and Design. New York : Wiley, 2007.

[26] E. Siachalou, E. Vafiadis, S. Goudos, T. Samaras, C. S. Koukourlis, and S. Panas, "On the design of switched-beam wideband base stations,"*IEEE Antennas Propag. Mag.*, vol. 46, no. 1, pp. 158–167, Feb. 2004.

[27] K. I. Pedersen, P. E. Mogensen, and J. R. Moreno, "Application and performance of downlink beamforming techniques in UMTS," *IEEE Commun. Mag.*, pp.134–143, Oct. 2003.

[28] R.S. Zebulum, M.A.C. Pacheco, M.M.B.R. Vellasco, Evolutionary Electronics, CRC Press, USA, 2002.

Matlab Code :-

1. Code for MVDR AoA Estimator for detecting sources at -5^0 and 5^0

```
clear all;
clc;
m=1;
j=0:5;
c=exp(pi*j*sin(pi/36)*i);
c=c.';
d=exp(pi*j*sin(-pi/36)*i);
d=d.';
A=[c,d];
k=100;
s=sign(randn(2,k));
n=sqrt(0.1)*randn(6,k);
Rss=s*s'/k;
Rns=n*s'/k;
Rsn=s*n'/k;
Rnn=n*n'/k;
Rxx=A*Rss*A'+A*Rsn+Rns*A'+Rnn;
for theta=-pi/3:pi/720:pi/3
    j=0:5;
    z=exp(pi*j*sin(theta)*i);
    z=z.';
    Pmu(m)=1/(z'*inv(Rxx)*z);
    m=m+1;
end
r=abs(Pmu);
r=10*log10(r);
theta=-60:0.25:60;
plot(theta,r);
xlabel('Angle in degrees');
ylabel('Power in db');
title('MVDR pseudospectrum for theta1=-5, theta2=5');
```

2. Code for Maximum Entropy AoA Estimator for detecting sources at -5^0 and 5^0

```
clear all;
clc;
m=1;
j=0:5;
c=exp(pi*j*sin(-pi/36)*i);
c=c.';
d=exp(pi*j*sin(pi/36)*i);
d=d.';
A=[c,d];
k=100;
s=sign(randn(2,k));
n=sqrt(0.1)*randn(6,k);
Rss=s*s'/k;
Rns=n*s'/k;
```

```
Rsn=s*n'/k;
Rnn=n*n'/k;
Rxx=A*Rss*A'+A*Rsn+Rns*A'+Rnn;
% Rss=[1 0;0 1];
% Rnn=0.1*eye(12);
% Rxx=A*Rss*A'+Rnn;
Rxx1=inv(Rxx);
for theta=-pi/3:pi/720:pi/3
    j=0:5;
    z=exp(pi*j*sin(theta)*i);
    z=z.';
    c=Rxx1(:,3);
    Pmu(m)=1/(z'*c*c'*z);
    m=m+1;
end
r=abs(Pmu);
r=10*log10(r)
theta=-pi/3:pi/720:pi/3;
theta=theta*(180/pi);
plot(theta,r);
xlabel('Angle in degrees');
ylabel('Power in db');
title('Maximum entropy pseudospectrum for theta1=-5, theta2=5');
```

3. Code for Linear Prediction AoA Estimator for detecting sources at -5^0 and 5^0

```
clear all;
clc;
m=1;
j=0:5;
c=exp(pi*j*sin(pi/36)*i);
c=c.';
d=exp(pi*j*sin(-pi/36)*i);
d=d.';
A=[c,d];
k=100;
s=sign(randn(2,k));
n=sqrt(0.1)*randn(6,k);
Rss=s*s'/k;
Rns=n*s'/k;
Rsn=s*n'/k;
Rnn=n*n'/k;
Rxx=A*Rss*A'+A*Rsn+Rns*A'+Rnn;
% Rss=[1 0;0 1];
% Rnn=0.1*eye(6);
% Rxx=A*Rss*A'+Rnn;
for theta=-pi/3:pi/720:pi/3
    j=0:5;
    z=exp(pi*j*sin(theta)*i);
    z=z.';
    u=[0 0 1 0 0 0] ;
    u=u.';
    Pmu(m)=(u.'*inv(Rxx)*u)/power((det(u.'*inv(Rxx)*z)),2);
    m=m+1;
end
```

```
r=abs(Pmu);
r=10*log10(r)
theta=-60:0.25:60;
plot(theta,r);
xlabel('Angle in degrees');
ylabel('Power in db');
title('Linear predictive pseudospectrum for theta1=-5,theta2=5');
```

4. Code for MUSIC Algorithm for detecting sources at -5^0 and 5^0

```
clear all;
clc;
j=0:5;
c=exp(pi*sin(pi/36)*j*i);
c=c.';
d=exp(pi*sin(-pi/36)*j*i);
d=d.';
A=[c,d];
k=100;
s=sign(randn(2,k));
s0=[1;1];
n0=0.1*eye(6);
n=sqrt(0.1)*randn(6,k);
Rss=s*s'/k;
Rss0=s0*s0'/k;
Rns=n*s'/k;
Rsn=s*n'/k;
Rnn=n*n'/k;
Rnn0=n0*n0'/k;
Rxx=A*Rss*A'+A*Rsn+Rns*A'+Rnn;
Rxx0=A*Rss0*A'+Rnn0;
[V,Dia]=eig(Rxx0);
[Y,Index]=sort(diag(Dia));
EN=V(:,Index(1:4));
m=1;
for theta=-pi/3:pi/720:pi/3
    k=0:5;
    z=exp(pi*k*sin(theta)*i);
    z=z.';
    w=exp(pi*k*sin(-theta)*i);
    w=w.';
    v=[z,w];
    Pmu(m)=1/(det(v'*EN*EN'*v));
    m=m+1;
end
r=abs(Pmu);
r=10*log10(r);
theta0=-60:0.25:60;
plot(theta0,r);
title('MUSIC Pseudospectrum for theta1=-5 and theta2=5');
xlabel('Angle in Degrees');
ylabel('Power in dB');
```

5. Code for generation of orthogonal beams produced by uniform illumination of eight microstrip patches fed by an 8x8 Butler Matrix

```
clc;
close all;
clear all;
E=importdata('C:\kushrel\kush12.mat');
theta0=30:0.5:150;
theta=theta0*(pi/180);
 beta=[22.5 -157.5 112.5 -67.5 67.5 -112.5 157.5 -22.5];
 beta=beta*(pi/180);
for j=1:241
 a(j,1)=theta(1,j);
end
C=a;
for q=1:8
    for k=1:241
   p=1

g(q,k)=E(k,p)*exp(3.5*(pi*cos(C(k,p))+beta(1,q))*i)*(sin(4*(pi*cos(C(k,
p))+beta(1,q)))/(sin(pi*cos(C(k,p))/2+beta(1,q)/2)));
     g(q,k)=abs(g(q,k));
     g(q,k)=g(q,k)*g(q,k);
    end
end
G=g/g(1,135);
for q=1:8
for k=1:241
    G1(q,k)=10*log10(G(q,k));
end
end
C=C';
C1=C*(180/pi);
plot(C1,G1(1,:));
hold on;
 plot(C1,G1(2,:));
 hold on;
 plot(C1,G1(3,:));
hold on;
plot(C1,G1(4,:));
hold on;
 plot(C1,G1(5,:));
hold on;
plot(C1,G1(6,:));
hold on;
plot(C1,G1(7,:));
hold on;
 plot(C1,G1(8,:));
hold on;
```

```
t=importdata('C:\kush0temp\kush02.mat');
a0=sim(net,in);
E=a0-t;
rmse=sqrt(sum(E.*E)/2000);
hist(E);
xlabel('Difference between Expected and Estimated SoI DoA (degrees)');
ylabel('Number of tests');
title('Uniform illumination with source separation 0.5 degrees ,
rmse=0.408');
w0=net.Iw{1,1};
w1=net.Lw{2,1};
w2=net.Lw{3,2};
w3=net.b{1};
w4=net.b{2};
w5=net.b{3};
w6=net.inputs{1}.processSettings{3};
w7=net.outputs{3}.processSettings{2};
```

8. Code for DoA Estimation of a single user in presence of 39 interferers, methodology applied is cosine illumination of the antenna elements

```
clc;
close all;
clear all;
G00=1.0790e+03;
E=importdata('C:\kushrel\kush12.mat');
A=[0.09755 0.27779 0.41573 0.49039 0.41573 0.27779 0.09755]
beta=[0 45 90 135 -45 -90 -135];
beta=beta*(pi/180);
theta=pi/6:pi/360:(5*pi)/6;
for j=1:241
 a(1,j)=theta(1,j);
end
A=a;
l=zeros(5000,7);
for k=1:5000
B(k,:)=randint(1,40,[1,241]);
for j=1:40
C(k,j)=A(1,B(k,j));
end
for q=1:7
  for p=1:40
      g(q,p)=0;
      for N=1:7
     g1(q,p)=E(B(k,p),1)*i*exp((pi*cos(C(k,p))+beta(1,q))*(N-
1)*i)*A(1,N);
     g(q,p)=g(q,p)+g1(q,p);
      end
     g(q,p)=abs(g(q,p));
     g(q,p)=g(q,p)*g(q,p);
  end
end
 G=(g/G00);
 C(k,:)=C(k,:)*(180/pi);
for m=1:7
```

```
    for n=1:39
        l(k,m)=l(k,m)+G(m,n)*128;
    end
l(k,m)=l(k,m)+G(m,40)*128*128;
end
L(k,:)=l(k,:);
end
C1=C(:,40);
L=L;
inputs=L';
targets=C1';
 net=newff(inputs,targets,[20 30]);
 net = train(net,inputs,targets);
in=importdata('C:\kush0temp\kush03.mat');
t=importdata('C:\kush0temp\kush04.mat');
a0=sim(net,in);
E=a0-t;
rmse=sqrt(sum(E.*E)/2000);
hist(E);
xlabel('Difference between Expected and Estimated SoI DoA (degrees)');
ylabel('Number of tests');
title('Cosine illumination with source separation 0.5 degrees ,
rmse=0.2168');
```

9. Code for DoA Estimation methodology applied with cosine illumination of the antenna elements plus sector beam with SeIR varying from 0 dB to -10 dB. Signal power is 0 dB while Interference power is varying from 0dB to 10 dB.

```
clc;
close all;
clear all;
G00=1.0790e+03;
E=importdata('C:\kushrel\kush12.mat');
A=[0.09755 0.27779 0.41573 0.49039 0.41573 0.27779 0.09755]
beta=[0 45 90 135 -45 -90 -135];
beta=beta*(pi/180);
theta=pi/6:pi/360:(5*pi)/6;
for j=1:241
 a(1,j)=theta(1,j);
end
A=a;
l=zeros(25000,7);
t=zeros(25000,1);
for k=1:25000
    r = 0.1 + (0.9).*rand(1,1);
B(k,:)=randint(1,40,[1,241]);
for j=1:40
C(k,j)=A(1,B(k,j));
end
for q=1:7
  for p=1:40
      g(q,p)=0;
      for N=1:7
```

```
g1(q,p)=E(B(k,p),1)*i*exp((pi*cos(C(k,p))+beta(1,q))*(N1)*i)*A(1,N);
     g(q,p)=g(q,p)+g1(q,p);
      end
     g(q,p)=abs(g(q,p));
     g(q,p)=g(q,p)*g(q,p);
     g0(1,p)=(E(B(k,p),1)*68.0769)';
  end
end
G=(g/G00);
C(k,:)=C(k,:)*(180/pi);
for m=1:7
    for n=1:39
        l(k,m)=l(k,m)+G(m,n)*256;
    end
l(k,m)=l(k,m)+G(m,40)*256*256*r;
end
for m=1:7
L(k,m)=l(k,m)+256;
end
   for n=1:39
        t(k,1)=t(k,1)+g0(1,n)*256;
    end
t(k,1)=t(k,1)+g0(1,40)*256*256*r;
t(k,1)=t(k,1)+256;
for j=1:7
    W(k,j)=L(k,j)/t(k,1);
end
end
C1=C(:,40);
inputs=W';
targets=C1';
net=newff(inputs,targets,[20 30]);
net = train(net,inputs,targets);
a = sim(net,inputs);
```

10. Code for DoA Estimation methodology applied with cosine illumination of the antenna elements plus sector beam with SeIR varying from -5 dB to 5 dB. Signal power is 0 dB while Interference power is varying from -5 dB to 5 dB.

```
clc;
close all;
clear all;
E=importdata('C:\kushrel\kush12.mat');
A=[0.09755 0.27779 0.41573 0.49039 0.41573 0.27779 0.09755]
beta=[0 45 90 135 -45 -90 -135];
beta=beta*(pi/180);
theta=pi/6:pi/360:(5*pi)/6;
for j=1:241
 a(1,j)=theta(1,j);
end
A=a;
l=zeros(25000,7);
t=zeros(25000,1);
```

5. Code for generation of orthogonal beams produced by uniform illumination of eight microstrip patches fed by an 8x8 Butler Matrix

```
clc;
close all;
clear all;
E=importdata('C:\kushrel\kush12.mat');
theta0=30:0.5:150;
theta=theta0*(pi/180);
 beta=[22.5 -157.5 112.5 -67.5 67.5 -112.5 157.5 -22.5];
 beta=beta*(pi/180);
for j=1:241
 a(j,1)=theta(1,j);
end
C=a;
for q=1:8
    for k=1:241
   p=1

g(q,k)=E(k,p)*exp(3.5*(pi*cos(C(k,p))+beta(1,q))*i)*(sin(4*(pi*cos(C(k,
p))+beta(1,q)))/(sin(pi*cos(C(k,p))/2+beta(1,q)/2)));
     g(q,k)=abs(g(q,k));
     g(q,k)=g(q,k)*g(q,k);
    end
end
G=g/g(1,135);
for q=1:8
for k=1:241
    G1(q,k)=10*log10(G(q,k));
end
end
C=C';
C1=C*(180/pi);
plot(C1,G1(1,:));
hold on;
 plot(C1,G1(2,:));
 hold on;
 plot(C1,G1(3,:));
hold on;
plot(C1,G1(4,:));
hold on;
 plot(C1,G1(5,:));
hold on;
plot(C1,G1(6,:));
hold on;
plot(C1,G1(7,:));
hold on;
 plot(C1,G1(8,:));
hold on;
```

6. Code for generation of seven beams produced by cosine illumination of eight microstrip patches fed by an 8x8 Butler Matrix plus a sector beam.

```
clc;
close all;
clear all;
G00=1.0790e+03;
E=importdata('C:\kushrel\kush12.mat');
A=[0.09755 0.27779 0.41573 0.49039 0.41573 0.27779 0.09755]
beta=[0 45 90 135 -45 -90 -135];
beta=beta*(pi/180);
theta=pi/6:pi/360:(5*pi)/6;
for j=1:241
 a(j,1)=theta(1,j);
end
C=a;
p=1
for k=1:241
    E0(k,p)=(E(k,p))*68.0769;
end
for q=1:7
   for k=1:241
      g(q,k)=0;
      for N=1:7
     g1(q,k)=E(k,p)*i*exp((pi*cos(C(k,p))+beta(1,q))*(N-1)*i)*A(1,N);
     g(q,k)=g(q,k)+g1(q,k);
      end
     g(q,k)=abs(g(q,k));
     g(q,k)=g(q,k)*g(q,k);
     g(8,k)=((E(k,p))*68.0769)';
   end
end
G=g/G00;
for k=1:241
    E1(k,p)=10*log10(E0(k,p));
end
for q=1:8
for k=1:241
    G1(q,k)=10*log10(G(q,k));
end
end
C=C';
C1=C*(180/pi);
plot(C1,G1(1,:));
hold on;
plot(C1,G1(2,:));
hold on;
plot(C1,G1(3,:));
hold on;
 plot(C1,G1(4,:));
hold on;
  plot(C1,G1(5,:));
hold on;
```

```
plot(C1,G1(6,:));
hold on;
plot(C1,G1(7,:));
hold on;
plot(C1,G1(8,:));
```

7. Code for DoA Estimation of a single user in presence of 39 interferers, methodology applied is uniform illumination of the antenna elements

```
clc;
close all;
clear all;
G0=1.5558e+04;
E=importdata('C:\kushrel\kush12.mat');
beta=[22.5 -157.5 112.5 -67.5 67.5 -112.5 157.5 -22.5];
beta=beta*(pi/180);
theta=pi/6:pi/1800:(5*(pi/6));
for j=1:241
 a(1,j)=theta(1,j);
end
A=a;
for k=1:25000
B(k,:)=randint(1,40,[1,241]);
for j=1:40
C(k,j)=A(1,B(k,j));
end
for q=1:8
  for p=1:40

g(q,p)=E(B(k,p),1)*sin(4*(pi*cos(C(k,p))+beta(1,q)))/(sin(pi*cos(C(k,p)
)/2+beta(1,q)/2));
     g(q,p)=abs(g(q,p));
     g(q,p)=g(q,p)*g(q,p);
  end
  end
G=g/G0;
C(k,:)=C(k,:)*(180/pi);
l=zeros(25000,8);
for m=1:8
    for n=1:39
        l(k,m)=l(k,m)+G(m,n)*128;
    end
l(k,m)=l(k,m)+G(m,40)*128*128;
end
L(k,:)=l(k,:);
end
C1=C(:,40);
L=L;
inputs=L';
targets=C1';
net=newff(inputs,targets,[20 30]);
net = train(net,inputs,targets);
a=sim(net,inputs);
in=importdata('C:\kush0temp\kush01.mat');
```

```
t=importdata('C:\kush0temp\kush02.mat');
a0=sim(net,in);
E=a0-t;
rmse=sqrt(sum(E.*E)/2000);
hist(E);
xlabel('Difference between Expected and Estimated SoI DoA (degrees)');
ylabel('Number of tests');
title('Uniform illumination with source separation 0.5 degrees ,
rmse=0.408');
w0=net.Iw{1,1};
w1=net.Lw{2,1};
w2=net.Lw{3,2};
w3=net.b{1};
w4=net.b{2};
w5=net.b{3};
w6=net.inputs{1}.processSettings{3};
w7=net.outputs{3}.processSettings{2};
```

8. Code for DoA Estimation of a single user in presence of 39 interferers, methodology applied is cosine illumination of the antenna elements

```
clc;
close all;
clear all;
G00=1.0790e+03;
E=importdata('C:\kushrel\kush12.mat');
A=[0.09755 0.27779 0.41573 0.49039 0.41573 0.27779 0.09755]
beta=[0 45 90 135 -45 -90 -135];
beta=beta*(pi/180);
theta=pi/6:pi/360:(5*pi)/6;
for j=1:241
 a(1,j)=theta(1,j);
end
A=a;
l=zeros(5000,7);
for k=1:5000
B(k,:)=randint(1,40,[1,241]);
for j=1:40
C(k,j)=A(1,B(k,j));
end
for q=1:7
  for p=1:40
      g(q,p)=0;
      for N=1:7
     g1(q,p)=E(B(k,p),1)*i*exp((pi*cos(C(k,p))+beta(1,q))*(N-
1)*i)*A(1,N);
     g(q,p)=g(q,p)+g1(q,p);
      end
     g(q,p)=abs(g(q,p));
     g(q,p)=g(q,p)*g(q,p);
  end
end
 G=(g/G00);
 C(k,:)=C(k,:)*(180/pi);
for m=1:7
```

```
    for n=1:39
        l(k,m)=l(k,m)+G(m,n)*128;
    end
l(k,m)=l(k,m)+G(m,40)*128*128;
end
L(k,:)=l(k,:);
end
C1=C(:,40);
L=L;
inputs=L';
targets=C1';
 net=newff(inputs,targets,[20 30]);
 net = train(net,inputs,targets);
in=importdata('C:\kush0temp\kush03.mat');
t=importdata('C:\kush0temp\kush04.mat');
a0=sim(net,in);
E=a0-t;
rmse=sqrt(sum(E.*E)/2000);
hist(E);
xlabel('Difference between Expected and Estimated SoI DoA (degrees)');
ylabel('Number of tests');
title('Cosine illumination with source separation 0.5 degrees ,
rmse=0.2168');
```

9. Code for DoA Estimation methodology applied with cosine illumination of the antenna elements plus sector beam with SeIR varying from 0 dB to -10 dB. Signal power is 0 dB while Interference power is varying from 0dB to 10 dB.

```
clc;
close all;
clear all;
G00=1.0790e+03;
E=importdata('C:\kushrel\kush12.mat');
A=[0.09755 0.27779 0.41573 0.49039 0.41573 0.27779 0.09755]
beta=[0 45 90 135 -45 -90 -135];
beta=beta*(pi/180);
theta=pi/6:pi/360:(5*pi)/6;
for j=1:241
 a(1,j)=theta(1,j);
end
A=a;
l=zeros(25000,7);
t=zeros(25000,1);
for k=1:25000
    r = 0.1 + (0.9).*rand(1,1);
B(k,:)=randint(1,40,[1,241]);
for j=1:40
C(k,j)=A(1,B(k,j));
end
for q=1:7
  for p=1:40
      g(q,p)=0;
      for N=1:7
```

```
g1(q,p)=E(B(k,p),1)*i*exp((pi*cos(C(k,p))+beta(1,q))*(N1)*i)*A(1,N);
     g(q,p)=g(q,p)+g1(q,p);
      end
     g(q,p)=abs(g(q,p));
     g(q,p)=g(q,p)*g(q,p);
     g0(1,p)=(E(B(k,p),1)*68.0769)';
  end
end
G=(g/G00);
C(k,:)=C(k,:)*(180/pi);
for m=1:7
    for n=1:39
        l(k,m)=l(k,m)+G(m,n)*256;
    end
l(k,m)=l(k,m)+G(m,40)*256*256*r;
end
for m=1:7
L(k,m)=l(k,m)+256;
end
   for n=1:39
        t(k,1)=t(k,1)+g0(1,n)*256;
    end
t(k,1)=t(k,1)+g0(1,40)*256*256*r;
t(k,1)=t(k,1)+256;
for j=1:7
    W(k,j)=L(k,j)/t(k,1);
end
end
C1=C(:,40);
inputs=W';
targets=C1';
net=newff(inputs,targets,[20 30]);
net = train(net,inputs,targets);
a = sim(net,inputs);
```

10. Code for DoA Estimation methodology applied with cosine illumination of the antenna elements plus sector beam with SeIR varying from -5 dB to 5 dB. Signal power is 0 dB while Interference power is varying from -5 dB to 5 dB.

```
clc;
close all;
clear all;
E=importdata('C:\kushrel\kush12.mat');
A=[0.09755 0.27779 0.41573 0.49039 0.41573 0.27779 0.09755]
beta=[0 45 90 135 -45 -90 -135];
beta=beta*(pi/180);
theta=pi/6:pi/360:(5*pi)/6;
for j=1:241
 a(1,j)=theta(1,j);
end
A=a;
l=zeros(25000,7);
t=zeros(25000,1);
```

```
for k=1:25000
    B(k,:)=randint(1,40,[1,241]);
for j=1:40
C(k,j)=A(1,B(k,j));
end
for q=1:7
  for p=1:40
      g(q,p)=0;
      for N=1:7
     g1(q,p)=E(B(k,p),1)*i*exp((pi*cos(C(k,p))+beta(1,q))*(N-
1)*i)*A(1,N);
     g(q,p)=g(q,p)+g1(q,p);
      end
     g(q,p)=abs(g(q,p));
     g(q,p)=g(q,p)*g(q,p);
     g0(1,p)=(E(B(k,p),1)*68.0769)';
  end
end
 G=(g);
C(k,:)=C(k,:)*(180/pi);
for m=1:7
    for n=1:39
        r = 0.3 + (2.9).*rand(1,1);
        l(k,m)=l(k,m)+G(m,n)*256*r;
    end
l(k,m)=l(k,m)+G(m,40)*256*256;
end
for m=1:7
L(k,m)=l(k,m)+256;
end
   for n=1:39
       r = 0.3 + (2.9).*rand(1,1);
        t(k,1)=t(k,1)+g0(1,n)*256*r;
    end
t(k,1)=t(k,1)+g0(1,40)*256*256;
t(k,1)=t(k,1)+256;
for j=1:7
    W(k,j)=L(k,j)/t(k,1);
end
end
C1=C(:,40);
inputs=W';
targets=C1';
net=newff(inputs,targets,[14 28]);
net = train(net,inputs,targets);
an = sim(net,inputs);
in=importdata('C:\kush0temp\k029.mat');
t=importdata('C:\kush0temp\k030.mat');
a0=sim(net,in);
E=a0-t;
rmse=sqrt(sum(E.*E)/2000);
hist(E);
xlabel('Difference between expected and estimated SoI DoA (degrees)');
ylabel('Number of tets');
```

```
title('Cosine illumination plus sector beam with Selr varying from -5
dB to 5 dB rmse=0.3815');
```

11. Code for detecting source in a desired sector like (0^0-10^0 in a total area of 120^0 using a 16 element antenna array having faults at a maximum of two positions)

a) Training Phase

```
clc;
close all;
clear all;
s1=sign(randn(6,100));
n=sqrt(0.1)*randn(16,100);
theta=-60:60;
for j=1:121
 a(1,j)=theta(1,j);
end
A=a;
for k=1:50000
B1(k,:)=randint(1,6,[1,121]);
C(k,1)=A(1,B1(k,1));
C(k,2)=A(1,B1(k,2));
C(k,3)=A(1,B1(k,3));
C(k,4)=A(1,B1(k,4));
C(k,5)=A(1,B1(k,5));
C(k,6)=A(1,B1(k,6));
for j=1:6
C1(k,j)=C(k,j)*(pi/180);
end
end
for k=1:50000
 j=1
    t1(k,j)=0;
    t2(k,j)=0;
    t3(k,j)=0;
    t4(k,j)=0;
    t5(k,j)=0;
    t6(k,j)=0;
    t7(k,j)=0;
    t8(k,j)=0;
    t9(k,j)=0;
    t10(k,j)=0;
    t11(k,j)=0;
    t12(k,j)=0;
end
for k=1:50000
for j=1:6
    if (C(k,j)>=-60 && C(k,j)<-50)
        t1(k,1)=1;
    end
end
for j=1:6
    if (C(k,j)>=-50 && C(k,j)<-40)
        t2(k,1)=1;
```

```
    end
end
for j=1:6
    if (C(k,j)>=-40 && C(k,j)<-30)
        t3(k,1)=1;
    end
end
for j=1:6
    if (C(k,j)>=-30 && C(k,j)<-20)
        t4(k,1)=1;
    end
end
for j=1:6
    if (C(k,j)>=-20 && C(k,j)<-10)
        t5(k,1)=1;
    end
end
for j=1:6
    if (C(k,j)>=-10 && C(k,j)<0)
        t6(k,1)=1;
    end
end
for j=1:6
    if (C(k,j)>=0 && C(k,j)<10)
        t7(k,1)=1;
    end
end
for j=1:6
    if (C(k,j)>=10 && C(k,j)<20)
        t8(k,1)=1;
    end
end
for j=1:6
    if (C(k,j)>=20 && C(k,j)<30)
        t9(k,1)=1;
    end
end
for j=1:6
    if (C(k,j)>=30 && C(k,j)<40)
        t10(k,1)=1;
    end
end
for j=1:6
    if (C(k,j)>=40 && C(k,j)<50)
        t11(k,1)=1;
    end
end
for j=1:6
    if (C(k,j)>=50 && C(k,j)<60)
        t12(k,1)=1;
    end
end
end
for k=1:50000
    for j=1:6
```

```
l(k,j)=pi*sin(C1(k,j));
    end
end
D=C;
D1=C1;
clear C*;
clear C1*;
z=1;
for k=1:15
for m=k+1:16
a9(1,z)=k;
a9(2,z)=m;
z=z+1;
end
end
for k=1:50000
    G1(k,:)=randint(1,1,[1,120]);
for b=1:6
for j=1:16
    s(b,j)=exp(l(k,b)*(j-1)*i);
end
s11(:,k)=a9(:,G1(k,:));
s(b,s11(:,k))=0;

end
M=s.';
X=M*s1+n;
RXX=X*X'/100;
Rxx1=RXX(1,1:16);
Rxx2=RXX(2,2:16);
Rxx3=RXX(3,3:16);
Rxx4=RXX(4,4:16);
Rxx5=RXX(5,5:16);
Rxx6=RXX(6,6:16);
Rxx7=RXX(7,7:16);
Rxx8=RXX(8,8:16);
Rxx9=RXX(9,9:16);
Rxx10=RXX(10,10:16);
Rxx11=RXX(11,11:16);
Rxx12=RXX(12,12:16);
Rxx13=RXX(13,13:16);
Rxx14=RXX(14,14:16);
Rxx15=RXX(15,15:16);
Rxx16=RXX(16,16);
Rxx(k,:)=[real(Rxx1)   imag(Rxx1)   real(Rxx2)   imag(Rxx2)   real(Rxx3)
imag(Rxx3)   real(Rxx4)   imag(Rxx4)   real(Rxx5)   imag(Rxx5)   real(Rxx6)
imag(Rxx6)   real(Rxx7)   imag(Rxx7)   real(Rxx8)   imag(Rxx8)   real(Rxx9)
imag(Rxx9)  real(Rxx10)  imag(Rxx10)  real(Rxx11)  imag(Rxx11)  real(Rxx12)
imag(Rxx12)  real(Rxx13)  imag(Rxx13)  real(Rxx14)  imag(Rxx14)  real(Rxx15)
imag(Rxx15)  real(Rxx16)  imag(Rxx16)];
end
Rxxx=Rxx';
[Y1,PS]=removeconstantrows(Rxxx);
[pn,ps1] = mapstd(Y1);
[ptrans,ps2] = processpca(pn,0.02);
```

```
net7 = newff(ptrans,t7',[20 30]);
net7 = train(net7,ptrans,t7');
Y7=sim(net7,ptrans);
w0=net7.Iw{1,1};
w1=net7.Lw{2,1};
w2=net7.Lw{3,2};
w3=net7.b{1};
w4=net7.b{2};
w5=net7.b{3};
w6=net.inputs{1}.processSettings{3};
w7=net.outputs{3}.processSettings{2};
save('D:\kush2elemfaileddoadet1\kush2elemfaileddoadet0.mat,'w0');
save('D:\kush2elemfaileddoadet1\kush2elemfaileddoadet1.mat','w1');
save('D:\kush2elemfaileddoadet1\kush2elemfaileddoadet2.mat','w2');
save('D:\kush2elemfaileddoadet1\kush2elemfaileddoadet3.mat','w3');
save('D:\kush2elemfaileddoadet1\kush2elemfaileddoadet4.mat','w4');
save('D:\kush2elemfaileddoadet1\kush2elemfaileddoadet5.mat','w5');
save('D:\kush2elemfaileddoadet1\kush2elemfaileddoadet6.mat','w6');
save('D:\kush2elemfaileddoadet1\kush2elemfaileddoadet7.mat','w7');
save('D:\kush2elemfaileddoadet1\kush2elemfaileddoadet8.mat','PS');
save('D:\kush2elemfaileddoadet1\kush2elemfaileddoadet9.mat','ps1');
save('D:\kush2elemfaileddoadet1\kush2elemfaileddoadet10.mat','ps2');
```

b) Testing phase

```
clc;
close all;
clear all;
s1=sign(randn(6,100));
n=sqrt(0.1)*randn(16,100);
w0=importdata('D:\kush2elemfaileddoadet1\kush2elemfaileddoadet0.mat');
w1=importdata('D:\kush2elemfaileddoadet1\kush2elemfaileddoadet1.mat');
w2=importdata('D:\kush2elemfaileddoadet1\kush2elemfaileddoadet2.mat');
w3=importdata('D:\kush2elemfaileddoadet1\kush2elemfaileddoadet3.mat');
w4=importdata('D:\kush2elemfaileddoadet1\kush2elemfaileddoadet4.mat');
w5=importdata('D:\kush2elemfaileddoadet1\kush2elemfaileddoadet5.mat');
w6=importdata('D:\kush2elemfaileddoadet1\kush2elemfaileddoadet6.mat');
w7=importdata('D:\kush2elemfaileddoadet1\kush2elemfaileddoadet7.mat');
PS=importdata('D:\kush2elemfaileddoadet1\kush2elemfaileddoadet8.mat');
ps1=importdata('D:\kush2elemfaileddoadet1\kush2elemfaileddoadet9.mat');
ps2=importdata('D:\kush2elemfaileddoadet1\kush2elemfaileddoadet10.mat')
;
k=1;
l=1;
o=1;
p=1;
q=1;
r=1;
s=1;
t=1;
for a=-60:1:45
C(k,1)=a;
k=k+1;
end
for b=-57:1:48
```

```
    C(l,2)=b;
    l=l+1;
end

for c=-54:1:51
    C(o,3)=c;
    o=o+1;
end
for d=-51:1:54
    C(p,4)=d;
    p=p+1;
end
for e=-48:1:57
    C(q,5)=e;
    q=q+1;
end
for f=-45:1:60
    C(r,6)=f;
    r=r+1;
end
C1=C*(pi/180);
for k=1:106
    for j=1:6
l(k,j)=pi*sin(C1(k,j));
    end
end
for k=1:106
for b0=1:6
for j=1:16
    s(b0,j)=exp(l(k,b0)*(j-1)*i);
end
s(b0,7)=0;
 s(b0,8)=0;
end
M=s.';
X=M*s1+n;
RXX=X*X'/100;
Rxx1=RXX(1,1:16);
Rxx2=RXX(2,2:16);
Rxx3=RXX(3,3:16);
Rxx4=RXX(4,4:16);
Rxx5=RXX(5,5:16);
Rxx6=RXX(6,6:16);
Rxx7=RXX(7,7:16);
Rxx8=RXX(8,8:16);
Rxx9=RXX(9,9:16);
Rxx10=RXX(10,10:16);
Rxx11=RXX(11,11:16);
Rxx12=RXX(12,12:16);
Rxx13=RXX(13,13:16);
Rxx14=RXX(14,14:16);
Rxx15=RXX(15,15:16);
Rxx16=RXX(16,16);
Rxx(k,:)=[real(Rxx1)   imag(Rxx1)   real(Rxx2)   imag(Rxx2)   real(Rxx3)
imag(Rxx3)   real(Rxx4)   imag(Rxx4)   real(Rxx5)   imag(Rxx5)   real(Rxx6)
```

```
imag(Rxx6)  real(Rxx7)  imag(Rxx7)  real(Rxx8)  imag(Rxx8)  real(Rxx9)
imag(Rxx9)  real(Rxx10)  imag(Rxx10)  real(Rxx11)  imag(Rxx11)  real(Rxx12)
imag(Rxx12)  real(Rxx13)  imag(Rxx13)  real(Rxx14)  imag(Rxx14)  real(Rxx15)
imag(Rxx15)  real(Rxx16)  imag(Rxx16)];
end
Rxxxa=Rxx';
Y1a = removeconstantrows('apply',Rxxxa,PS);
pna = mapstd('apply',Y1a,ps1);
ptransa = processpca('apply',pna,ps2);
v1=mapminmax('apply',ptransa,w6);
for j=1:106
v2=tansig(w0*v1(:,j)+w3);
v3=tansig(w1*v2+w4);
v4=purelin(w2*v3+w5);
v5(1,j)=mapminmax('reverse',v4,w7);
end
v5;
Z(1:56,1)=C(1:56,6);
Z(57:91,1)=C(72:106,1);
Z1(1:56,1)=v5(1,1:56);
Z1(57:91,1)=v5(1,72:106);
for j=1:60
A(j,1)=0;
end
 for j=61:71
A(j,1)=1;
end
for j=72:121
A(j,1)=0;
end
theta=-60:60;
plot(Z,Z1,'r:','linewidth',4);
hold on;
plot(theta,A,'k');
xlabel('AoA in degrees');
ylabel('Network Output');
title('element failed');
legend('Ann response','Desired Response');
```

12. Code for DoA Estimation Phase in sector (0^0-10^0)

a)Training Phase

```
clc;
close all;
clear all;
s1=sign(randn(6,100));
n=sqrt(0.1)*randn(16,100);
theta=0:10;
for j=1:11
 a(1,j)=theta(1,j);
end
A=a;
theta1=-60:-1;
for j=1:60
```

```
 a1(1,j)=theta1(1,j);
end
theta2=11:60;
k=1;
for j=61:110
 a1(1,j)=theta2(1,k);
 k=k+1;
end
A1=a1;
for k=1:5000
B1(k,:)=randint(1,1,[1,11]);
B2(k,:)=randint(1,5,[1,110]);
C(k,1)=A(1,B1(k,1));
C(k,2)=A1(1,B2(k,1));
C(k,3)=A1(1,B2(k,2));
C(k,4)=A1(1,B2(k,3));
C(k,5)=A1(1,B2(k,4));
C(k,6)=A1(1,B2(k,5));
for j=1:6
C1(k,j)=C(k,j)*(pi/180);
end
end
D=C;
clear C*;
clear C1*;
z=1;
for k=0:9
for m=k+1:10
a9(1,z)=k;
a9(2,z)=m;
z=z+1;
end
end
clear B1*;
clear B2*;
for k=1:8000
B1(k,:)=randint(1,1,[1,55]);
B2(k,:)=randint(1,4,[1,110]);
C(k,1)=a9(1,B1(k,:));
C(k,2)=a9(2,B1(k,:));
C(k,3)=A1(1,B2(k,1));
C(k,4)=A1(1,B2(k,2));
C(k,5)=A1(1,B2(k,3));
C(k,6)=A1(1,B2(k,4));
end
for k=1:8000
    for j=1:6
    D(k+5000,j)=C(k,j);
    end
end
clear B1*;
clear B2*;
z=1;
for k=0:8
for m=k+1:9
```

```
for n=m+1:10
a10(1,z)=k;
a10(2,z)=m;
a10(3,z)=n;
z=z+1;
end
end
end
for k=1:12000
B1(k,:)=randint(1,1,[1,165]);
B2(k,:)=randint(1,4,[1,110]);
C(k,1)=a10(1,B1(k,:));
C(k,2)=a10(2,B1(k,:));
C(k,3)=a10(3,B1(k,:));
C(k,4)=A1(1,B2(k,1));
C(k,5)=A1(1,B2(k,2));
C(k,6)=A1(1,B2(k,3));
end
for k=1:12000
    for j=1:6
    D(k+13000,j)=C(k,j);
    end
end


for k=1:25000
    for j=1:3
t1(k,j)=20;
    end
 end
for k=1:25000
 for j=1:3
   if (D(k,j)>-1 && D(k,j)<11)
       t1(k,j)=D(k,j);
   end
 end
end
D1=D*pi/180;
for k=1:25000
    for j=1:6
l(k,j)=pi*sin(D1(k,j));
    end
end
z=1;
for k=1:16
for m=k:16
a12(1,z)=k;
a12(2,z)=m;
z=z+1;
end
end
for k=1:4500
     G1(k,:)=randint(1,1,[1,136]);
for b=1:6
for j=1:16
```

```
    s(b,j)=exp(1(k,b)*(j-1)*i);
end
s11(:,k)=a12(:,G1(k,:));
s(b,s11(:,k))=0;
end
M=s.';
X=M*s1;
RXX=X*X'/100;
Rxx1=RXX(1,1:16);
Rxx2=RXX(2,2:16);
Rxx3=RXX(3,3:16);
Rxx4=RXX(4,4:16);
Rxx5=RXX(5,5:16);
Rxx6=RXX(6,6:16);
Rxx7=RXX(7,7:16);
Rxx8=RXX(8,8:16);
Rxx9=RXX(9,9:16);
Rxx10=RXX(10,10:16);
Rxx11=RXX(11,11:16);
Rxx12=RXX(12,12:16);
Rxx13=RXX(13,13:16);
Rxx14=RXX(14,14:16);
Rxx15=RXX(15,15:16);
Rxx16=RXX(16,16);
Rxx(k,:)=[real(Rxx1)   imag(Rxx1)   real(Rxx2)   imag(Rxx2)   real(Rxx3)
imag(Rxx3)   real(Rxx4)   imag(Rxx4)   real(Rxx5)   imag(Rxx5)   real(Rxx6)
imag(Rxx6)   real(Rxx7)   imag(Rxx7)   real(Rxx8)   imag(Rxx8)   real(Rxx9)
imag(Rxx9)  real(Rxx10)  imag(Rxx10)  real(Rxx11)  imag(Rxx11)  real(Rxx12)
imag(Rxx12)  real(Rxx13)  imag(Rxx13)  real(Rxx14)  imag(Rxx14)  real(Rxx15)
imag(Rxx15)  real(Rxx16)  imag(Rxx16)];
end
Rxxx=Rxx';
[Y1,PS]=removeconstantrows(Rxxx);
[pn,ps1] = mapstd(Y1);
[ptrans,ps2] = processpca(pn,0.02);
[tn,ts] = mapminmax(t1');
net = newff(ptrans,tn,[20 30]);
net = train(net,ptrans,tn);
Y=sim(net,ptrans);
a = mapminmax('reverse',Y,ts);
w0=net.Iw{1,1};
w1=net.Lw{2,1};
w2=net.Lw{3,2};
w3=net.b{1};
w4=net.b{2};
w5=net.b{3};
w6=net.inputs{1}.processSettings{3};
w7=net.outputs{3}.processSettings{2};
save('D:\kush2faildoadetfinal0\kush2faildoadet0.mat','w0');
save('D:\kush2faildoadetfinal0\kush2faildoadet1.mat','w1');
save('D:\kush2faildoadetfinal0\kush2faildoadet2.mat','w2');
save('D:\kush2faildoadetfinal0\kush2faildoadet3.mat','w3');
save('D:\kush2faildoadetfinal0\kush2faildoadet4.mat','w4');
save('D:\kush2faildoadetfinal0\kush2faildoadet5.mat','w5');
save('D:\kush2faildoadetfinal0\kush2faildoadet6.mat','w6');
```

```
save('D:\kush2faildoadetfinal0\kush2faildoadet7.mat','w7');
save('D:\kush2faildoadetfinal0\kush2faildoadet8.mat','PS');
save('D:\kush2faildoadetfinal0\kush2faildoadet9.mat','ps1');
save('D:\kush2faildoadetfinal0\kush2faildoadet10.mat','ps2');
save('D:\kush2faildoadetfinal0\kush2faildoadet11.mat','ts');
```

b) Testing Phase

```
clc;
close all;
clear all;
s1=sign(randn(6,100));
n=sqrt(0.1)*randn(16,100);
w0=importdata('D:\kush2faildoadetfinal0\kush2faildoadet0.mat');
w1=importdata('D:\kush2faildoadetfinal0\kush2faildoadet1.mat');
w2=importdata('D:\kush2faildoadetfinal0\kush2faildoadet2.mat');
w3=importdata('D:\kush2faildoadetfinal0\kush2faildoadet3.mat');
w4=importdata('D:\kush2faildoadetfinal0\kush2faildoadet4.mat');
w5=importdata('D:\kush2faildoadetfinal0\kush2faildoadet5.mat');
w6=importdata('D:\kush2faildoadetfinal0\kush2faildoadet6.mat');
w7=importdata('D:\kush2faildoadetfinal0\kush2faildoadet7.mat');
PS=importdata('D:\kush2faildoadetfinal0\kush2faildoadet8.mat');
ps1=importdata('D:\kush2faildoadetfinal0\kush2faildoadet9.mat');
ps2=importdata('D:\kush2faildoadetfinal0\kush2faildoadet10.mat');
ts=importdata('D:\kush2faildoadetfinal0\kush2faildoadet11.mat');
theta1=-60:-1;
for j=1:60
 a1(1,j)=theta1(1,j);
end
theta2=11:60;
k=1;
for j=61:110
 a1(1,j)=theta2(1,k);
 k=k+1;
end
A1=a1;
for k=1:90
B23(k,:)=randint(1,4,[1,104]);
C(k,3)=A1(1,B23(k,1));
C(k,4)=A1(1,B23(k,2));
C(k,5)=A1(1,B23(k,3));
C(k,6)=A1(1,B23(k,4));
end
z1=0:8;
z2=2:10;
w=1;
for k=1:9
    C(k,1)=z1(1,w);
    C(k,2)=z2(1,w);
    w=w+1;
end
w=1;
for k=10:18
    C(k,1)=z1(1,w);
    C(k,2)=z2(1,w);
```

```
    w=w+1;
end
w=1;
for k=19:27
    C(k,1)=z1(1,w);
    C(k,2)=z2(1,w);
    w=w+1;
end
w=1;
for k=28:36
    C(k,1)=z1(1,w);
    C(k,2)=z2(1,w);
    w=w+1;
end
w=1;
for k=37:45
    C(k,1)=z1(1,w);
    C(k,2)=z2(1,w);
    w=w+1;
end
w=1;
for k=46:54
    C(k,1)=z1(1,w);
    C(k,2)=z2(1,w);
    w=w+1;
end
w=1;
for k=55:63
    C(k,1)=z1(1,w);
    C(k,2)=z2(1,w);
    w=w+1;
end
w=1;
for k=64:72
   C(k,1)=z1(1,w);
   C(k,2)=z2(1,w);
   w=w+1;
end
w=1;
for k=73:81
    C(k,1)=z1(1,w);
    C(k,2)=z2(1,w);
    w=w+1;
end
w=1;
for k=82:90
    C(k,1)=z1(1,w);
    C(k,2)=z2(1,w);
    w=w+1;
end
s1=sign(randn(6,100));
n=sqrt(0.1)*randn(16,100);
C1=C*pi/180;
for k=1:90
    for j=1:6
```

```
l(k,j)=pi*sin(C1(k,j));
    end
end
for k=1:90
for b=1:6
for j=1:16
    s(b,j)=exp(l(k,b)*(j-1)*i);
end
 s(b,8)=0;
 s(b,9)=0;
end
M=s.';
X=M*s1+n;
RXX=X*X'/100;
Rxx01=RXX(1,1:16);
Rxx02=RXX(2,2:16);
Rxx3=RXX(3,3:16);
Rxx4=RXX(4,4:16);
Rxx5=RXX(5,5:16);
Rxx6=RXX(6,6:16);
Rxx7=RXX(7,7:16);
Rxx8=RXX(8,8:16);
Rxx9=RXX(9,9:16);
Rxx10=RXX(10,10:16);
Rxx11=RXX(11,11:16);
Rxx12=RXX(12,12:16);
Rxx13=RXX(13,13:16);
Rxx14=RXX(14,14:16);
Rxx15=RXX(15,15:16);
Rxx16=RXX(16,16);
Rxx(k,:)=[real(Rxx01)  imag(Rxx01)  real(Rxx02)  imag(Rxx02)  real(Rxx3)
imag(Rxx3)  real(Rxx4)  imag(Rxx4)  real(Rxx5)  imag(Rxx5)  real(Rxx6)
imag(Rxx6)  real(Rxx7)  imag(Rxx7)  real(Rxx8)  imag(Rxx8)  real(Rxx9)
imag(Rxx9)  real(Rxx10)  imag(Rxx10)  real(Rxx11)  imag(Rxx11)  real(Rxx12)
imag(Rxx12)  real(Rxx13)  imag(Rxx13)  real(Rxx14)  imag(Rxx14)  real(Rxx15)
imag(Rxx15)  real(Rxx16)  imag(Rxx16)];
end
Rxxxa=Rxx';
Y1a=removeconstantrows('apply',Rxxxa,PS);
pna=mapstd('apply',Y1a,ps1);
ptransa=processpca('apply',pna,ps2);
v1=mapminmax('apply',ptransa,w6);
for j=1:90
v2=tansig(w0*v1(:,j)+w3);
v3=tansig(w1*v2+w4);
v4=purelin(w2*v3+w5);
v5=mapminmax('reverse',v4,w7);
v6(:,j)=mapminmax('reverse',v5,ts);
end
v6;
C0=C(:,1:3);
E=v6-C0';
E1=E(1,:);
E2=E(2,:);
rmse1=sqrt(sum(E1.*E1)/90);
```

```
rmse2=sqrt(sum(E2.*E2)/90);
% Ya=sim(net,ptransa);
% a0=mapminmax('reverse',v5,ts);
N=1:90;
plot(N,v6(2,:),'r:','linewidth',4)
hold on;
plot(N,(C(:,2))','k')
hold on;
plot(N,v6(1,:),'b:')
hold on;
plot(N,(C(:,1))','k')
xlabel('points');
ylabel('AoA');
title('element failed');
legend('Source1 Ann response','Source1 Desired Response', 'Source 2 Ann
response', 'Source 2 Desired Response',-1);
```