NEURAL NETWORK BASED BEAMFORMING FOR LINEAR AND CYLINDRICAL ARRAY APPLICATIONS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MURAT GÜREKEN

IN PARTIAL FULLFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCES
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

MAY 2009

Approval of the thesis:

NEURAL NETWORK BASED BEAMFORMING FOR LINEAR AND CYLINDRICAL ARRAY APPLICATIONS

submitted by **MURAT GÜREKEN** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen _____
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. İsmet Erkmen _____
Head of Department, **Electrical and Electronics Engineering Dept., METU**

Prof. Dr. Gülbin Dural _____
Supervisor, **Electrical and Electronics Engineering Dept., METU**

Examining Committee Members:

Prof. Dr. Kemal Leblebicioğlu _____
Electrical and Electronics Engineering Dept., METU

Prof. Dr. Gülbin Dural _____
Electrical and Electronics Engineering Dept., METU

Assoc. Prof. Dr. Sencer Koç _____
Electrical and Electronics Engineering Dept., METU

Assoc Prof. Özlem Aydın Çivi _____
Electrical and Electronics Engineering Dept., METU

Dr. Selçuk Çaylar _____
Turkish Air Force

Date : May 8, 2009

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Murat Güreken

Signature :

ABSTRACT

NEURAL NETWORK BASED BEAMFORMING FOR LINEAR AND CYLINDRICAL ARRAY APPLICATIONS

Güreken, Murat

M. Sc., Department of Electrical and Electronics Engineering

Supervisor: Prof. Dr. Gülbin Dural

May 2009, 106 pages

In this thesis, a Neural Network (NN) based beamforming algorithm is proposed for real time target tracking problem. The algorithm is performed for two applications, linear and cylindrical arrays.

The linear array application is implemented with equispaced omnidirectional sources. The influence of the number of antenna elements and the angular seperation between the incoming signals on the performance of the beamformer in the linear array beamformer is studied, and it is observed that the algorithm improves its performance by increasing both two parameters in linear array beamformer.

The cylindrical array application is implemented with twelve microstrip patch antenna (MPA) elements. The angular range of interest is divided into twelve sectors. Since three MPA elements are used to form the beam in each sector, the input size of the neural network (NN) is reduced in cylindrical array. According to the reduced size of NN, the training time of the beamformer is decreased. The reduced size of the NN has no degradation in forming the beams to the desired directions.

The angular separation between the targets is an important parameter in cylindrical array beamformer.

Keywords: Beamforming, Neural Network, Direction of Arrival, Cylindrical Array, Linear Array

ÖZ

YAPAY SİNİR AĞLARI TEMELLİ DOĞRUSAL VE SİLİNDİR ANTEN DİZİLERİNDE HUZME ŞEKİLLENDİRME UYGULAMALARI

Güreken, Murat

Yüksek Lisans, Elektrik Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Gülbin Dural

Mayıs 2009, 106 sayfa

Bu tez çalışmasında Yapay Sinir Ağları kullanılarak gerçek zamanlı hedef takibi için geliştirilen bir huzme şekillendirme algoritması önerilmektedir. Algoritma, doğrusal ve silindirik anten dizisi olmak üzere iki uygulamada denenmiştir.

Doğrusal dizi uygulaması eşit aralıklı yönsüz noktasal kaynaklar kullanılarak yapılmıştır. Dizideki anten eleman sayısı ve anten dizisine gelen işaretler arasındaki açısal farkın algoritma üzerindeki etkisi incelenmiş, hem anten eleman sayısı hem de açısal farkı arttırmanın algoritmanın performansını arttırdığı görülmüştür.

Silindirik dizi uygulaması on iki elemanlı mikroşerit yama anten (MPA) elemanlarıyla gerçekleştirilmiştir. Açısal bölge on iki sektöre ayrılmıştır. Her bir sektöre gelen işaretlere huzmeyi yönlendirmek için üç adet MPA elemanı kullanılarak, Yapay Sinir Ağları girdilerinin boyutu küçültülmüştür. Boyutun küçülmesi Yapay Sinir Ağlarının eğitiminin hızlanmasında önemli bir faktör

olmuştur. Boyutun küçülmesi huzme formunda herhangi bir bozulmaya yol açmamıştır.

Hedefler arasındaki açısal fark, silindirik anten dizisinin performansında belirleyici etmenlerden biridir.

Anahtar Kelimeler: Huzme şekillendirme, Yapay Sinir Ağları, Geliş Yönü, Silindirik Dizi, Doğrusal Dizi

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my supervisor Prof. Dr. Gülbin Dural for her encouragements, guidance, advice, criticism and insight throughout the research. I would also like to thank Dr. Selçuk Çaylar for his comments and contributions.

I would like to thank ASELSAN Inc. for facilities provided for the competition of this thesis, and also to my mother for her encouragement.

TABLE OF CONTENTS

ABSTRACT iv

ÖZ vi

ACKNOWLEDGEMENTS viii

TABLE OF CONTENTS ix

LIST OF TABLES xi

LIST OF FIGURES xii

CHAPTERS

1 INTRODUCTION 1

2 FIXED BEAMFORMING 6

2.1 Signal Model 7

2.2 Delay-and-Sum Beamforming 11

2.3 Null-Steering Beamforming 13

2.4 Optimal Beamforming 15

2.5 Optimal Beamforming Using Reference Signal 17

2.6 Beam-Space Beamforming Process 19

2.7 Broad-Band Beamforming 20

2.8 Partitioned Realization 23

2.9 Frequency-Domain Beamforming 25

2.10 Digital Beamforming 26

2.11 Eigenstructure Method 29

3 ADAPTIVE BEAMFORMING 31

3.1 Sample Matrix Inversion Algorithm 32
3.2 Least Mean Square Algorithm 32
3.2.1 Unconstrained LMS Algorithm 33
3.2.2 Normalized LMS Algorithm 35
3.2.3 Constrained LMS Algorithm 36
3.3 Recursive Least Square Algorithm 39
3.4 Constant Modulus Algorithm 40
3.5 Conjugate Gradient Method 41
4 NEURAL NETWORK APPROACH FOR BEAMFORMING 43
4.1 Radial Basis Function Neural Network 45
4.2 Linear Array 47
4.2.1 Formulation 48
4.2.2 Algorithm 51
4.3 Cylindrical Array With Microstrip Patch Antenna Elements 54
4.3.1 Formulation 55
4.3.2 Algorithm 64
5 SIMULATIONS 68
5.1 Linear Array Resuls 69
5.1.1 Single Target 70
5.1.2 Two Targets 80
5.2 Cylindrical Array Results 88
5.2.1 Single Target 90
5.2.2 Multiple Targets 92
6 CONCLUSION 99
REFERENCES 102
APPENDIX 104-106

LIST OF TABLES

TABLES

Table 5-1 Summary of Linear Array Application for different number of antenna elements 73

Table 5-2 Summary of Linear Array Application for different SNR values 78

Table 5-3 Summary of Two Target Linear Array Application for different angular seperation values 84

Table 5-4 Summary of Two Target Linear Array Application for different SNR values 87

LIST OF FIGURES

FIGURES

Figure 2-1 Linear Array 7
Figure 2-2 Narrow-Band Beamformer 8
Figure 2-3 Delay-and-Sum Beamformer 12
Figure 2-4 Narrow-Band Beamformer with Reference Signal 18
Figure 2-5 Beam-Space Beamformer 19
Figure 2-6 Broad-Band Beamformer 21
Figure 2-7 Partitioned Processer 24
Figure 2-8 Frequency Domain Beamformer 26
Figure 2-9 Analog Beamformer 27
Figure 2-10 Digital Beamforming [1] 28
Figure 2-11 Sampled Digital Beamformer [1] 29
Figure 4-1 Radial Basis Function Neural Network 46
Figure 4-2 Linear Antenna Array 47
Figure 4-3 Neural Network Beamformer 51
Figure 4-4 Cylindrical Microstrip Patch Antenna Element [4] 55
Figure 4-5 The Polar Plot of The Electric Field Pattern of One Microstrip Patch Antenna 58
Figure 4-6 The Plot of The Electric Field Pattern of One Microstrip Patch Antenna 59
Figure 4-7 Total Electric Field Pattern of CMPA 61
Figure 4-8 The geometry of twelve-element cylindrical array [5] 62
Figure 4-9 Sectors of CMPA 65
Figure 4-10 The Neural Network Beamformer Architecture 65
Figure 5-1 Linear Array Geometry 69
Figure 5-2 Array Pattern for a signal coming from -30^{o} with $an = 3$ and $SNR = 30\ dB$ 71

Figure 5-3 Array Pattern for a signal coming from -30^o with $an=5$ and $SNR=30\ dB$ 71

Figure 5-4 Array Pattern for a signal coming from -30^o with $an=10$ and $SNR=30\ dB$ 72

Figure 5-5 Array Pattern for a signal coming from -30^o with $an=15$ and $SNR=30\ dB$ 72

Figure 5-6 3-dB Beamwith change for Linear Array Application by changing the number of antenna elements 74

Figure 5-7 Maximum Sidelobe Lavel change for Linear Array Application by changing the number of antenna elements 74

Figure 5-8 Array Pattern for a signal coming from 20^o with $an=10$ and $SNR=5\ dB$ 75

Figure 5-9 Array Pattern for a signal coming from 20^o with $an=10$ and $SNR=10\ dB$ 76

Figure 5-10 Array Pattern for a signal coming from 20^o with $an=10$ and $SNR=15\ dB$ 76

Figure 5-11 Array Pattern for a signal coming from 20^o with $an=10$ and $SNR=20\ dB$ 77

Figure 5-12 Array Pattern for a signal coming from 20^o with $an=10$ and $SNR=30\ dB$ 77

Figure 5-13 Array Pattern for 5 Different Noisy Signal Cases for Single Target.... 79

Figure 5-14 RMS error change for different SNR value..... 79

Figure 5-15 Sidelobe level change for different SNR values 80

Figure 5-16 Signals coming from 0^o and 1^o with $an=10$ and $SNR=30\ dB$ 81

Figure 5-17 Signals coming from 0^o and 3^o with $an=10$ and $SNR=30\ dB$ 81

Figure 5-18 Signals coming from 0^o and 5^o with $an=10$ and $SNR=30\ dB$ 82

Figure 5-19 Signals coming from 0^o and 10^o with $an=10$ and $SNR=30\ dB$ 82

Figure 5-20 Signals coming from 0^o and 15^o with $an=10$ and $SNR=30\ dB$ 83

Figure 5-21 Signals coming from 0^o and 20^o with $an=10$ and $SNR=30\ dB$ 83

Figure 5-22 Array Response for two targets in 10^o and 25^o, $an = 10$ and $SNR = 5\ dB$ 85

Figure 5-23 Array Response for two targets in 10^o and 25^o, $an = 10$ and $SNR = 10\ dB$ 85

Figure 5-24 Array Response for two targets in 10^o and 25^o, $an = 10$ and $SNR = 15\ dB$ 86

Figure 5-25 Array Response for two targets in 10^o and 25^o, $an = 10$ and $SNR = 20\ dB$ 86

Figure 5-26 Array Response for two targets in 10^o and 25^o, $an = 10$ and $SNR = 30\ dB$ 87

Figure 5-27 Array Pattern for 5 Different Noisy Signal Cases for Two Targets..... 88

Figure 5-28 Cylindrical Array Pattern and Geometry..... 89

Figure 5-29 Array Pattern for signal coming from 140^o, $SNR = 10\ dB$ 90

Figure 5-30 Array Pattern for signal coming from 140^o, $SNR = 15\ dB$ 91

Figure 5-31 Array Pattern for signal coming from 140^o, $SNR = 20\ dB$ 91

Figure 5-32 Array Pattern for signal coming from 140^o, $SNR = 30\ dB$ 92

Figure 5-33 Array Response for targets in 285^0 and 290^0, $SNR = 30\ dB$ 93

Figure 5-34 Array Response for targets in 285^0 and 295^0, $SNR = 30\ dB$ 94

Figure 5-35 Array Response for targets in 285^0 and 300^0, $SNR = 30\ dB$ 94

Figure 5-36 Array Response for targets in 285^0 and 305^0, $SNR = 30\ dB$ 95

Figure 5-37 Array Response for targets in 285^0 and 315^0, $SNR = 30\ dB$ 95

Figure 5-38 Array Pattern for targets in 25^0, 125^0, 210^0 and 300^0, $SNR = 30\ dB$ 97

Figure 5-39 Array Pattern for targets in 345^0, 15^0, 170^0 and 190^0, and for interfering signals in 0^0 and 180^0, $SNR = 30\ dB$ 98

Figure A-1 DOA Neural Network Architecture 105

Figure A-2 DOA Result of One Target in 2^0 106

Figure A-3 DOA Neural Network Architecture 106

CHAPTER 1

INTRODUCTION

Improving technologies in Global Positioning, mobile communication and Radar technologies demand faster real time target tracking. According to these improving technologies, new DOA and beamforming algorithms are developed.

Early works show that a good beamforming plays an important role in target tracking problem. Several beamforming works proposed in the literature serves for real time beamforming problem [1], [3], [4] , [8] .

Beamforming is an antenna array processing technique for shaping the antenna array beams to the directions of one or more desired signals and to put nulls in the directions of interfering signals [2], [7].

The main idea of the beamformer is to direct the antenna pattern to the desired signal's direction and to attenuate the interference signal(s). The output provided by each antenna element is weighted to distinguish the spatial properties of a signal of interest from noise and interference.

The beamforming term takes its name from the early forms of antenna arrays that were used to generate pencil beams [2]. The antenna arrays with pencil beams receive signals from a specific direction and attenuate signals coming from interfering directions. Since this type of arrays have narrow beam widths, beamforming has been extended to rich scattering scenarios.

A brief discussion over adaptive beamforming algorithms is studied in [1]. Beamforming techniques are divided in two main groups in [1]: fixed beamforming and adaptive beamforming. In fixed beamforming, the interference is not cancelled but mitigated with a reasonable cost. Adaptive beamforming requires complex signal processing algorithms to steer the main lobe towards the desired direction and to cancel the interfering signals. Adaptive beamforming applications leads to effective performance, but is more expensive and needs quite implementation efforts.

A narrowband beamformer operates L signals received on antenna elements. The antenna weights are $w = [w_1, w_2, \cdots, w_L]^T$ and the received signals are $x = [x_1, x_2, \cdots, x_L]^T$.

$$y(t) = \sum_{i=1}^{L} w_i^* \cdot x_i(t) \tag{1.1}$$

The weights can be applied at the Radio Frequency (RF) stage, realizing a beamforming network with analog devices. This choice is rather costly due to the high quality required for the RF components. Analog beamforming needs precise phase shifters and selective power dividers. Most of the analog beamformers are used to form a unique lobe towards a desired direction, while multiple lobes are usually difficult to realize [1].

An array of N antenna elements, in which the weights can be modified both in amplitude and phase, provides $N-1$ degrees of freedom. The one dimensional constraint regarding the desired direction reduces this number to $N-2$, which represents the number of directions that can be cancelled.

The digital beamforming is similar to analog beamforming in case that both techniques adjust the antenna weights. In digital beamforming, the received RF signals are downconverted to Intermediate Frequency (IF) and then digitized by Analog-to-Digital Converters (ADCs). The DownConverter (DC) is adopted to

simplify the digitization process, which gets more complex as the frequency increases. [1]

Improving mobile communication and Global Positioning Sytem (GPS) technologies needs faster beamforming algorithms are needed. Since the number of users and the interfering signals increases, the communication systems require to track the users continuously while they are moving, and to put nulls in the directions of interferences by forming the beams of the antenna arrays. In this point, the Neural Networks have an increasing demand of implementation in DOA and beamforming.

An early work of Neural Network (NN) Direction of Arrival (DOA) approach is given in [4] and [5]. In this work, NN is used for target tracking problem. Multiple sources tracking algorithm is proposed here to train the NN for different numbers of targets and antenna elements. A summary of DOA with NN is given in the Appendix.

One of the adaptive beamforming algorithms proposed in [1] is NN beamforming. According to its fast convergence, NN applications are having more important role in DOA and beamforming applications. NN beamforming is a type of adaptive beamforming which has a training and a performance phase. The NN is trained for suitable input signals and output weight pairs. The trained NN is then used for beamforming.

The early works of A.H. Zooghby have several NN DOA and beamforming applications, given in [3] and [10]. In both works Radial Basis Function Neural Network (RBFNN) is used as NN. Part of the thesis work related with NN beamforming application presented in this thesis is closely related to the work given by [3].

The proposed NN algorithm studied in this thesis uses Radial Basis Function Neural Networks (RBFNN) for trainning of each input and output pair. The NN takes the

incoming signals of antenna elements as inputs of the training set. The outputs of the training set are the optimum weights, assigned to each antenna array elements.

Two antenna array applications are practiced for the proposed beamforming algorithm: linear array with omnidirectional equispaced elements, and cylindrical array with microstrip patch antenna elements.

In the first beamforming problem, a total angular range of $\left[-90^0, 90^0\right]$ is covered by linear array of isotropic sources. The NN is trained for the cases of single target and two targets exist in the range.

The performance of the linear array beamforming NN is examined according to the number of antenna elements in the array and according to the angular separation between the targets for two target case for different values of SNR.

In the second application, cylindrical array application of twelve-elements array with directive microstrip patch antennas are taken into consideration to cover a total angular range of 360^0. The 360^0 of total range is divided into twelve sectors, being 30^0 of each. This approach fastens the algorithm computationally. By this consideration, beamforming algorithm works for 30^o, instead of full range.

Cylindrical array NN beamforming performance is analyzed for different SNR values and different amounts of angular separation between multiple targets for single and multiple targets applications. Interfering signal is included in cylindrical array simulations.

An introduction to beamforming, the evolution of the beamforming algorithms and an introduction to the beamforming applications studied in literature and the organization of this thesis are presented in Chapter 1.

The fixed beamforming are given in Chapter 2. The main ideas and the main constraints of the algorithms are discussed in this chapter.

The adaptive beamforming algorithm types, including the Neural Network beamforming, are given in Chapter 3.

The Neural Network based beamforming algorithm, studied for this thesis is discussed in Chapter 4. The proposed algorithm for linear array and for cylindrical arrays are presented. The angular seperation between the incoming signals are discussed and the effect of the number of antenna elements for linear array with different SNR's are discussed in this chapter.

Simulation results for the proposed algorithms in linear and cylindrical arrays are given in Chapter 5.

Chapter 6 covers the conclusion and the proposals for the future work.

CHAPTER 2

FIXED BEAMFORMING

Beamforming is a signal processing approach of beam steering application of antenna arrays. The main idea is to direct the antenna pattern to the desired signal's direction and to attenuate the interference signal. The beamforming is effective when the directions of the desired and the interference signals are different.

The main idea of beamforming can be expressed as;

$$y(t) = \sum_{i=1}^{L} w_i^* \cdot x_i(t) \tag{2.1}$$

where $y(t)$ is the output of the antenna array.

* denotes the complex conjugate of w, w_i are the weights which are applied to each antenna element to shape the beam and x_i are the coming signals from the sources. Both of these variables are complex variables. Desired signals are tracked by shaping the main beam.

Both the phase and the amplitudes are controlled to steer the antenna pattern to the desired location.

2.1 Signal Model

An array with L omnidirectional elements is assumed to exist in an environment with M uncorrelated point sources. The time, $\tau_l(\theta_i)$ needed to arrive from the i^{th} source to the l^{th} element is

$$\tau_l(\theta_i) = \frac{d}{c}(l-1)\cos\theta_i \tag{2.2}$$

which is given in Figure 2-1.

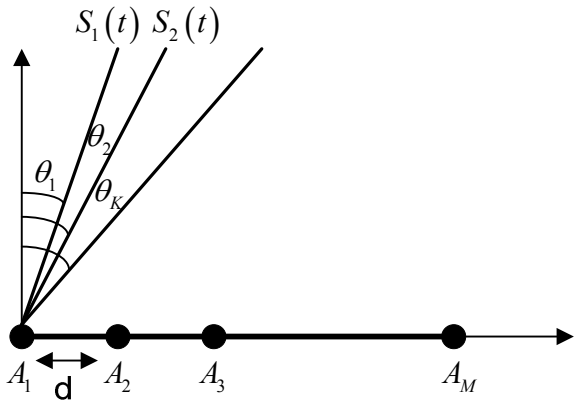

Figure 2-1 Linear Array

The signal induced on the reference element of the array is,

$$m_i(t)\, e^{j2\pi f_0 t} \tag{2.3}$$

Here, $m_i(t)$ is the modulating function. Modulating function shows the characterization of the induced signals.

The signal which arrived $\tau_l(\phi_i,\theta_i)$ seconds before to the l^{th} element according to the reference element is expessed as,

$$m_i(t)\, e^{j2\pi f_0\left(t+\tau_l(\phi_i,\theta_i)\right)} \tag{2.4}$$

Here it is assumed that the signal has narrow bandwidth and the modulating function does not change in $\tau_l(\phi_i,\theta_i)$ seconds.

The total induced signal on to the l^{th} element in the presence of M sources is

$$x_l = \sum_{i=1}^{M} m_i(t)\, e^{j2\pi f_0\left(t+\tau_l(\phi_i,\theta_i)\right)} + n_l(t) \tag{2.5}$$

$n_l(t)$ is a zero mean random noise with σ_n^2 variance, applied to the l^{th} element.

The narrow-band beamformer concept is outlined in Figure 2-2.

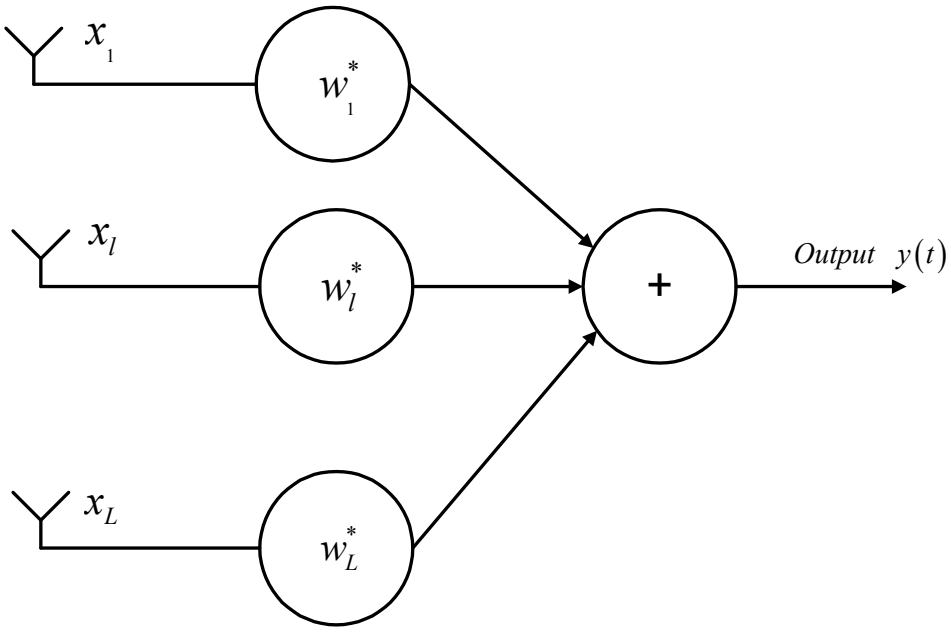

Figure 2-2 Narrow-Band Beamformer

As it is seen in Figure 2-2, signal induced in each antenna element is multiplied by the weight of that antenna element. The weighted antenna signals are summed and array output is formed.

The array output is given by

$$y(t) = \sum_{l=1}^{L} w_l^* x_l(t) \tag{2.6}$$

The weights are formed as,

$$\underline{w} = \left[w_1, w_2, \ldots, w_L \right]^T \tag{2.7}$$

and the signals induced on antenna elements are

$$\underline{x}(t) = \left[x_1(t), x_2(t), \cdots, x_L(t) \right]^T \ . \tag{2.8}$$

The beamformer, designed by the vectors described above, is

$$y(t) = \underline{w}^H \underline{x}(t) \tag{2.9}$$

The correlation matrix is the correlation between the i^{th} and the j^{th} element of antenna array. It is expressed by the expectation operation,

$$R = E\left\{ \underline{x}(t), \underline{x}^H(t) \right\} \tag{2.10}$$

The steering vector, $\underline{s_i}$ associated from the direction (ϕ_i, θ_i) by the i^{th} source is

$$\underline{s_i} = \left[e^{j2\pi f_0 \tau_1(\phi_i, \theta_i)}, \cdots, e^{j2\pi f_0 \tau_N(\phi_i, \theta_i)} \right]^T \tag{2.11}$$

The matrix notation of correlation matrix R can be formed as follows

$$R = ASA^H + \sigma_n^2 I \quad , \tag{2.12}$$

where I is the identity matrix. The matrix A, $[L \times M]$ is composed of M steering vectors

$$A = [\underline{s}_1, \underline{s}_2, \cdots, \underline{s}_M] \tag{2.13}$$

S matrix, $[M \times M]$ in (2.14) expresses the correlation between sources, which takes the values 1 or 0

$$S_{ij} = \begin{cases} p_i, & i = j \\ 0, & i \neq j \end{cases} \tag{2.14}$$

here p_i denotes the variances of the modulating function $m_i(t)$.

The correlation function can be written in terms of eigenvalues and eigenvectors which is the spectral decomposition of R. The eigenvalues are composed of two groups. The first one is the group of the eigenvalues of the directional sources and the second is the group of the eigenvalues of the white noises.

Eigenvalues are expressed as λ_l and the eigenvectors as $\underline{U}_l$,

$$R = \Sigma \Lambda \Sigma^H \quad , \tag{2.15}$$

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & & & 0 \\ 0 & \ddots & & & \\ & & \lambda_l & & \\ & & & \ddots & 0 \\ 0 & & & 0 & \lambda_L \end{bmatrix} , \tag{2.16}$$

$$\Sigma = [\underline{U}_1 \cdots \underline{U}_L] \tag{2.17}$$

By the above expressions, R is formed

$$R = \sum_{l=1}^{M} \lambda_l \underline{U}_l \underline{U}_l^H + \sigma_n^2 I \tag{2.18}$$

According to the selection of weights, beamforming is divided into the following types.

2.2 Delay-and-Sum Beamforming

It is the simplest beamformer. Each antenna element is multiplied by the same weight values to steer the beam to direction (ϕ_0, θ_0). This direction is called as look direction and it is assumed to be known before. The array weights can be written in terms of steering vector, $\underline{s}_0$ in look direction as

$$\underline{w}_c = \frac{1}{L}\underline{s}_0 \tag{2.19}$$

Considering that there is a signal source with power p_s and modulating function $m_s(t)$, the expression of the signal source induced on the l^{th} element of the antenna array is

$$x_{ls}(t) = m_s(t)e^{j2\pi f_0\left(t+\tau_l(\phi_0,\theta_0)\right)} \tag{2.20}$$

The induced signal expression can be rewritten including $\underline{s}_0$

$$x_s(t) = m_s(t)e^{j2\pi f_0 t}\underline{s}_0 \tag{2.21}$$

The output vector is

$$y(t) = \underline{w}_c^H \underline{x}_s(t) \tag{2.22}$$

$$= m_s(t)e^{j2\pi f_0 t} \tag{2.23}$$

The beamforming concept is equivalent to the steering array mechanically, except it is done electronically by phase shifters.

The delay-and-sum beamforming design is shown in Figure 2-3.

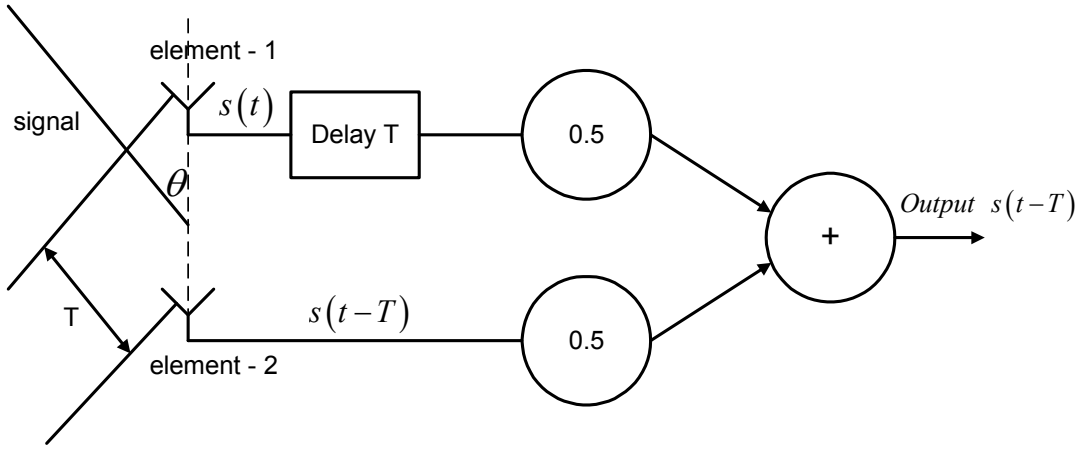

Figure 2-3 Delay-and-Sum Beamformer

The antenna array of Figure 2-3 consists of two elements. The wave arrival time difference between two elements is given as

$$T = \frac{d}{c}\cos\theta \ . \tag{2.24}$$

The signal arriving to the first antenna element is $s(t)$, and the signal arriving to the second element is $s(t-T)$. Signal induced to the first element is delayed with T, and no delay is applied to the signal induced to the second element. The signals become in phase. Each signal is multiplied by 0.5 and then summed to obtain antenna output.

Delay-and-sum beamformer works well in an environment with white noise only. It fails in the presence of directional interferences. The correlation function of the white noise is

$$R_N = \sigma_n^2 I \tag{2.25}$$

The output noise power is

$$P_N = \underline{w}_c^H R_N w_c \quad , \tag{2.26}$$

$$= \frac{\sigma_n^2}{L} \tag{2.27}$$

It is observed that the output power is decreased by the ratio L. The SNR of input and output are given by

$$\begin{aligned} &input \quad SNR = \frac{p_s}{\sigma_n^2} \\ &output \quad SNR = \frac{p_s L}{\sigma_n^2} \end{aligned} \quad , \tag{2.28}$$

In the absence of directional interferences, the beamformer leads a gain value of L.

The beamformer output to a signal coming from direction (ϕ_I, θ_I), denoting $\underline{s}_I$ as the steering vector coming from direction (ϕ_I, θ_I), is

$$\underline{w}_c^H \underline{s}_I = \frac{1}{L} \underline{s}_0^H \underline{s}_I \quad . \tag{2.29}$$

2.3 Null-Steering Beamforming

The main idea of this beamformer is to generate the desired antenna beam, i.e. put a null in the directions of interferences.

Delay-and-sum beamformer is used ot estimate the signal coming. It is delayed and summed by the beamformer. The output of the signal is substracted from each antenna element. By this process, strong interferences can be cancelled successfully [1],[2].

The weights are selected by the following constraints to put a unity response in the direction of desired signals, and to put nulls in the direction of interfering signals

$$\begin{aligned} \underline{w}^H \underline{s}_0 &= 1 \\ \underline{w}^H \underline{s}_i &= 0 \end{aligned} \qquad i = 1, 2, \ldots, k \tag{2.30}$$

The weight expression is given by

$$\underline{w}^H A = \underline{e}_1^T \tag{2.31}$$

A is composed of steering vectors,

$$A = \left[s_0, s_1, \cdots, s_k \right], \tag{2.32}$$

and e_1 is the constraint vector which is composed of zeros, and a one in the first element.

$$\underline{e}_1 = \left[1, 0, \cdots, 0 \right]^T \quad . \tag{2.33}$$

A is expected to be invertible, means that each steering vector is linearly independent from each other. The weight vector is

$$\underline{w}^H = \underline{e}_1^T A^{-1} \tag{2.34}$$

The first row of matrix A^{-1} gives the weight vector.

If A is not a square matrix, means that less than $L-1$ nulls are needed, the weight vector is given as

$$\underline{w}^H = \underline{e}_1^T A^H \left(A A^H \right)^{-1} \quad . \tag{2.35}$$

2.4 Optimal Beamforming

The optimal beamformer does not require the knowledge of the directions and the power levels of the interferences ad background noises. However, the knowledge of the desired signal is required. The weights are calculated to maximize the SNR [1].

The weight vector expression is

$$\underline{w} = \mu_0 R_N^{-1} \underline{s}_0 \quad , \tag{2.36}$$

R_N is the correlation matrix of noise and does not contain any information of signal from direction (ϕ_0, θ_0). The constant μ_0 is

$$\mu_0 = \frac{1}{\underline{s}_0^H R_N^{-1} \underline{s}_0} \quad . \tag{2.37}$$

Weight equation is obtained by putting μ_0 in weight's equation

$$\underline{w} = \frac{R_N^{-1} \underline{s}_0}{\underline{s}_0^H R_N^{-1} \underline{s}_0} \quad . \tag{2.38}$$

The noise level can not be powerful enough in real world applications. R, the correlation matrix of noise plus signal is used instead of R_N. Weight equation becomes

$$\underline{w} = \frac{R^{-1} \underline{s}_0}{\underline{s}_0^H R^{-1} \underline{s}_0} \quad . \tag{2.39}$$

There are two constraints while calculating the weights

$$\begin{aligned} &\min_{\underline{w}} imize \quad \underline{w}^H R \underline{w} \quad , \\ &subject\ to \quad \underline{w}^H \underline{s}_0 = 1 \quad . \end{aligned} \tag{2.40}$$

The objective of this beamformer is to minimize the mean output power and put a unity response in the look direction.

In this process, the power of noise is minimized and output signal stays constant., which yields maximized value of the output SNR.

The output SNR is given by

$$SNR = p_s \underline{s}_0^H R_N^{-1} \underline{s}_0 \tag{2.41}$$

When there is no interfering signal, the beamformer behaves as conventional (delay-and-sum) beamformer. The weights in this special case is

$$\underline{w} = \frac{\underline{s}_0}{L} \quad . \tag{2.42}$$

The output SNR in this case, assuming the array gain $G = L$

$$SNR = \frac{p_s L}{\sigma_n^2} \quad . \tag{2.43}$$

In another case, it is assumed that there is only one interfering signal with power p_I. The SNR and the antenna gain, G are given as,

$$SNR = \frac{p_s L}{\sigma_n^2} \frac{\rho + \frac{\sigma_n^2}{p_I L}}{1 + \frac{\sigma_n^2}{p_I L}} \quad , \tag{2.44}$$

$$G = \frac{p_I L}{\sigma_n^2} \frac{\left(1 + \frac{\sigma_n^2}{p_I}\right)\left(\rho + \frac{\sigma_n^2}{p_I L}\right)}{1 + p_I} \quad , \tag{2.45}$$

Here ρ is

$$\rho = 1 - \frac{\underline{s}_0^H \underline{s}_I \underline{s}_I^H \underline{s}_0}{L^2} \quad . \tag{2.46}$$

Another case is that the interference is much more stronger than the background noise, $p_I \gg \sigma_n^2$. SNR and the antenna gain, G are as follows

$$SNR \cong \frac{p_s L \rho}{\sigma_n^2} \quad , \tag{2.47}$$

$$G \cong \frac{p_I L \rho}{\sigma_n^2} \quad . \tag{2.48}$$

To have an appropriate performance from the optimal beamformer, there has to be less than $L-1$ interferences.

2.5 Optimal Beamforming Using Reference Signal

Optimal beamforming using reference signal is another application of narrow-band beamformer which uses a reference signal to obtain weights, shown in Figure 2-4.

The weights are adjusted by the error signal

$$\varepsilon(t) = r(t) - \underline{w}^H \underline{x}(t) \tag{2.49}$$

where $r(t)$ is the reference signal. The main constraint for calculating the weights is to minimize the MSE between the array output and the reference signals

$$\begin{aligned} MSE &= E\left\{\left|\varepsilon(t)^2\right|\right\} \\ &= E\left\{\left|r(t)\right|^2\right\} + \underline{w}^H R \underline{w} - 2\underline{w}^H E\left\{\underline{x}(t) r(t)\right\} \end{aligned} \tag{2.50}$$

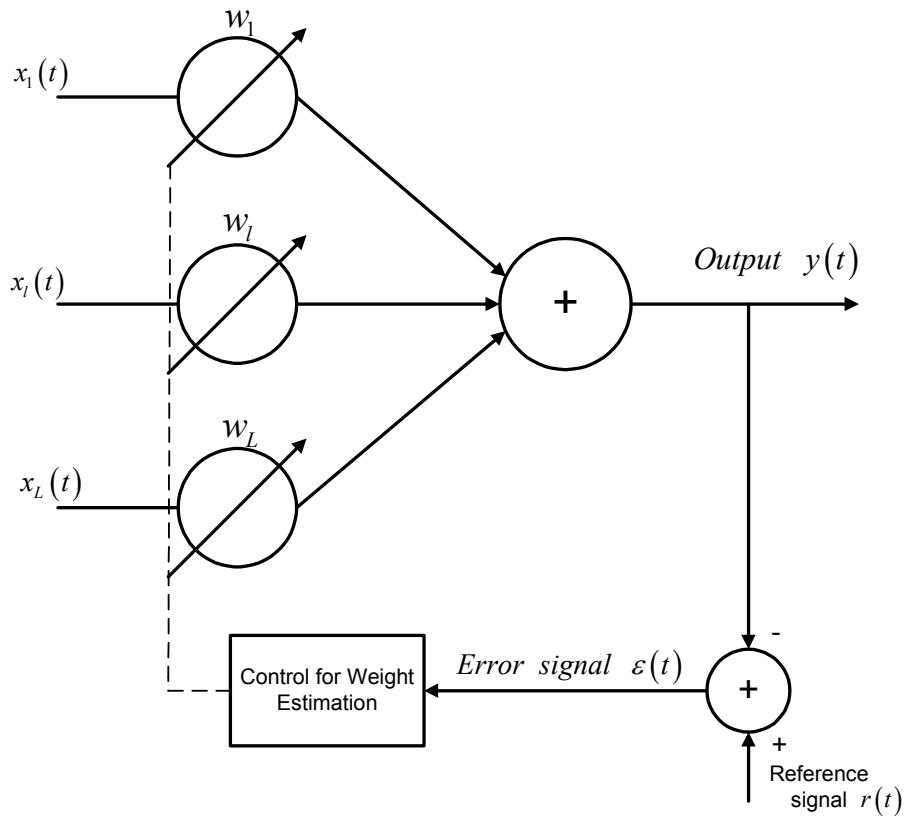

Figure 2-4 Narrow-Band Beamformer with Reference Signal

$\underline{z} = E\{\underline{x}(t) r(t)\}$ shows the correlation between the array output and reference signals.

The weights are

$$w_{MSE} = R^{-1}\underline{z} \quad . \tag{2.51}$$

The MMSE is

$$MMSE = E\left\{\left|r(t)\right|^2\right\} - \underline{z}^H R^{-1}\underline{z} \quad . \tag{2.52}$$

Beamformer is successful for obtaining a weak signal in the presence of strong interfering signal by taking the reference signal, zero to get rid of the strong interference signal. It has no effect on the desired signal. [1]

2.6 Beam-Space Beamforming Process

The beamformers discussed in the previous sections are the element-space beamformers.

Beam-space beamformers are composed of two parts shown in Figure 2-5.

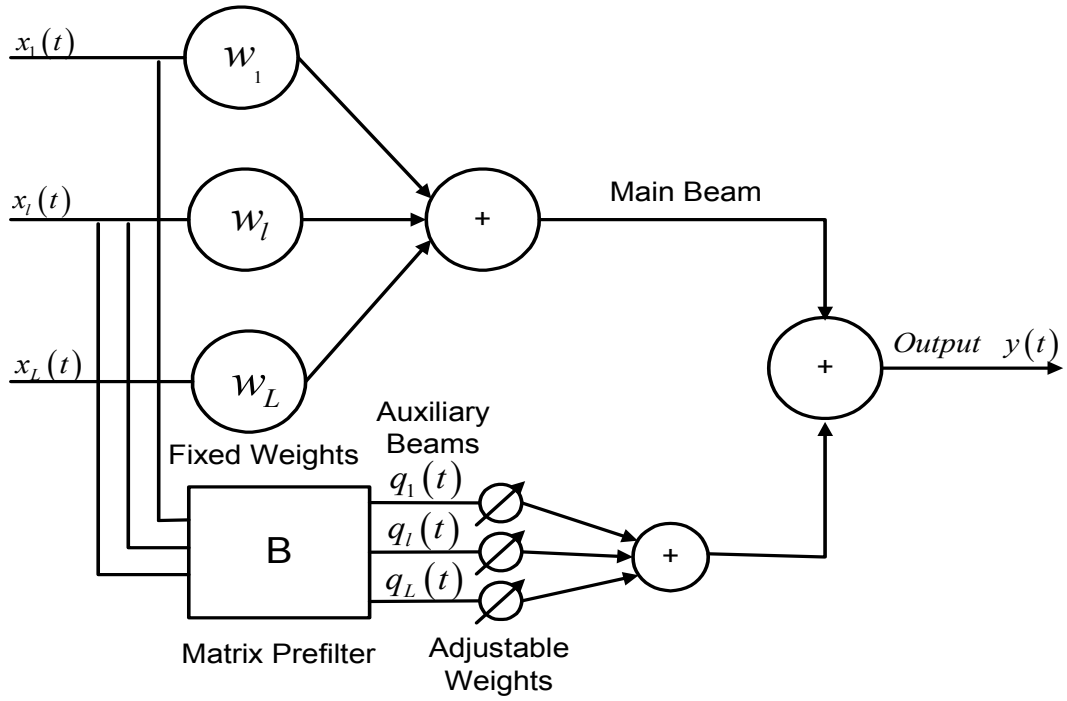

Figure 2-5 Beam-Space Beamformer

The first part generates multiple beams. The beams are weighted and summed in the second stage. The weights are not adaptive, they are fixed.

In this beamformer, there are L beams, where L is the number of antenna elements. One beam, the main beam, is in the direction of the desired signal. $L-1$ secondary beams are substracted from the main beam to cancel the interfering effects from the main beam. There is no information of desired signal in secondary (auxillary) beams. So, in substracting process no information of desired signal is lost.

The main beam pattern has a sinc shape, $\sin Lx / \sin x$. Since there is no information of desired signal in the auxillary beams, these beams have nulls in the look direction.

The $M-1$ auxillary beams are expressed as

$$\underline{q} = \underline{x}^H (t) B \tag{2.53}$$

B is the block matrix.

$$\underline{s}_0^H B = 0 \tag{2.54}$$

$\underline{s}_0$ is the steering vector in the look direction.

The beamformer has high performance when the number of interfering signals is less than the number of antenna elements, $M-1 < L$. The calculation is less than it is in the element-space beamformers. $M-1$ weights are required in beam-space beamformer, comparing to L weights required in the element-space beamformers.

The main idea is to cancel the maximum interference and to maximize the output SNR.

2.7 Broad-Band Beamforming

Broad-band beamforming shown in Figure 2-6 is an element-space beamformer.

It is seen from the figure that the incoming signal from the direction (ϕ_0, θ_0) is delayed by an amount of $T_l(\phi_0, \theta_0)$,

$$T_l(\phi_0, \theta_0) = T_0 + \tau_l(\phi_0, \theta_0) \tag{2.55}$$

where $\tau_l(\phi_0, \theta_0)$ is the time of signal arriving to the l^h array element according to the reference array element.

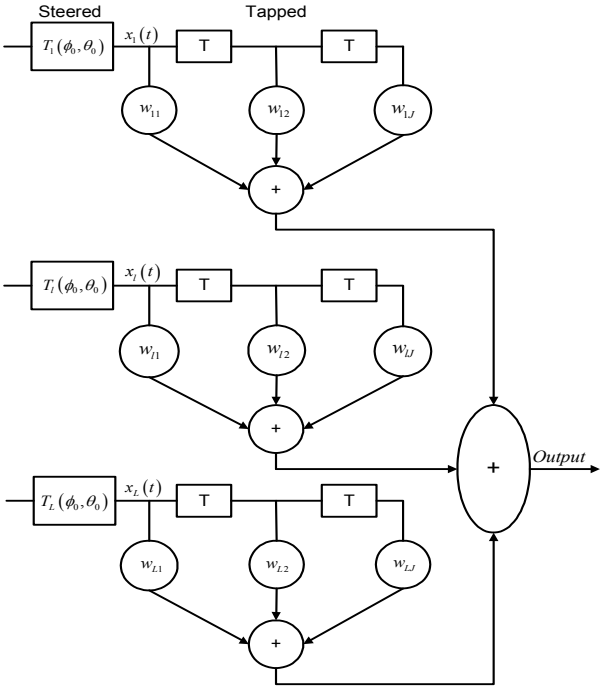

Figure 2-6 Broad-Band Beamformer

The output signal $x(t)$ in terms of induced signal $s(t)$ is

$$x_l(t) = s\left(t + \tau_l(\phi, \theta) - T_l(\phi_0, \theta_0)\right) \tag{2.56}$$

The broad-band is Finite Length Impulse Response (FIR) filter. It is designed to steer the beam to the look direction by adjusting the coefficients of the filter. The weights are

$$\underline{w} = \left[\underline{w}_1, \underline{w}_2, \cdots, \underline{w}_J\right]^T \tag{2.57}$$

The weight vector has size of $\left[LJ \times 1\right]$. The mean output power is

$$P\left(\underline{w}\right) = \underline{w}^T R \underline{w} \ . \tag{2.58}$$

The correlation between the $\left(l-1\right)^{th}$ tap on the m^{th} channel and the $\left(k-1\right)^{th}$ tap on the n^{th} channel outputs

$$\left(R_{m,n}\right)_{l,k} = \rho\left[\left(m-n\right)T + T_l\left(\phi_0,\theta_0\right) - T_k\left(\phi_0,\theta_0\right) + \tau_k\left(\phi,\theta\right) - \tau_l\left(\phi,\theta\right)\right] \tag{2.59}$$

$\rho\left(t\right)$ is the expected value operator,

$$\rho\left(\tau\right) = E\left\{s\left(t\right)s\left(t+\tau\right)\right\} \tag{2.60}$$

$$\rho\left(\tau\right) = \int_{-\infty}^{\infty} S\left(f\right)e^{j2\pi f\tau}df \ . \tag{2.61}$$

The following constraints are taken into account for the requirement of interference cancellation and of putting appropriate response in the look direction

$$\begin{aligned} \min_{\underline{w}} imize \quad & \underline{w}^T R \underline{w} \quad , \\ subject\ to \quad & C^T \underline{w} = \underline{F} \quad . \end{aligned} \tag{2.62}$$

$\underline{F}$ $\left[J \times 1\right]$ denotes the frequency responce in look direction and C $\left[LJ \times J\right]$ is

$$C = \begin{bmatrix} \underline{1} & & 0 \\ & \underline{1} & \\ & \ddots & \\ 0 & & \underline{1} \end{bmatrix} \quad . \tag{2.63}$$

Weight vector under these constraints is

$$\underline{w} = R^{-1}C\left(C^T R^{-1} C\right)^{-1} \underline{F} \quad . \tag{2.64}$$

F_j is equal to the sum of L weights before j^{th} delay. In order to direct the beam to the look direction, the sum of weights is zero, except the weight near the middle of filter.

The steered array pattern can have a broader shape by taking derivative constraints into account. In this constraint, the derivative of power pattern with respect to ϕ and θ is equal to zero.

Another constraint uses the known parameters of desired signal, which is correlation constraint

$$\begin{aligned} \min imize \quad & \underline{w}^T R \underline{w} \quad , \\ subject\ to \quad & \underline{r_d}^T \underline{w} = \rho_0 \quad . \end{aligned} \tag{2.65}$$

ρ_0 is a constant value and $\underline{r_d}$ specifies the correlation between the array output and desired signal.

2.8 Partitioned Realization

Partitioned processor is a beam-space processor, shown in Figure 2-7.

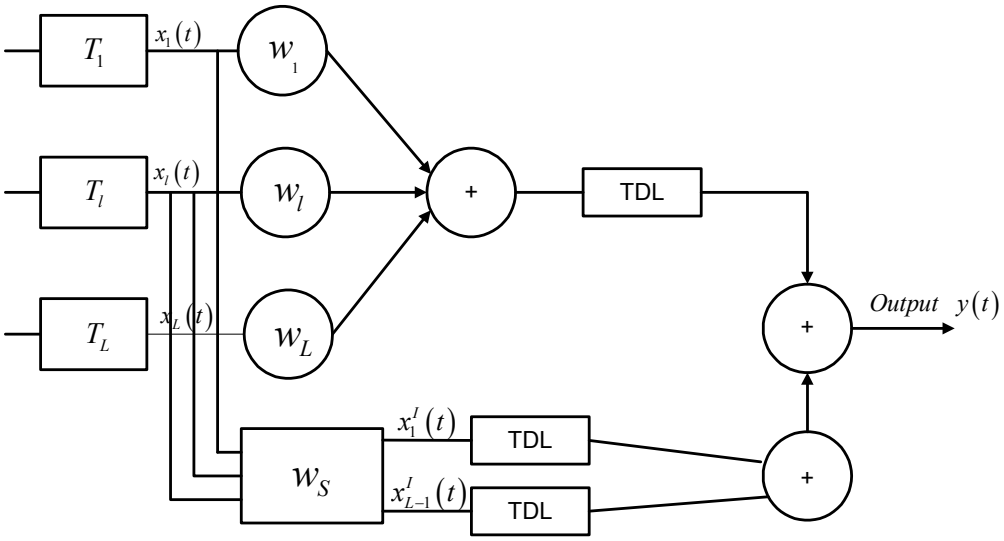

Figure 2-7 Partitioned Processer

The main constraint is to direct a unitary response in the look direction. The first time delays control the induced signal from the look direction to put the signals in each antenna element in phase. [1], [2]

After the time delays, the beamformer is divided into two parts. In the first part, the delayed signals are multiplied by fixed weights, by selecting appropriate FIR filter coefficients. The output of the first part is

$$y_c(t) = \sum_{k=0}^{J-1} F_{k+1} y(t - Tk) \quad , \tag{2.66}$$

$$y(t) = \frac{\underline{x}^T(t)\underline{1}}{L} \quad . \tag{2.67}$$

In the second part of the beamformer, the signal coming from the look direction is blocked by W_s matrix. The sum of each row of W_s is equal to zero. The output of W_s matris is

$$\underline{x}'(t) = W_s \underline{x}(t) \quad . \tag{2.68}$$

$L-1$ output beams are adjusted by FIR filter coefficients, $\underline{a}_k$ in TDL box. The output is

$$y_a(t) = \sum_{k=0}^{J-1} \underline{a}_k^T \underline{x}'(t-kT) \quad . \tag{2.69}$$

the constraint in choosing the coefficients is

$$\min_{\underline{a}_k} imize \quad E\left\{\left[y_c(t) - y_a(t)\right]^2\right\} \quad . \tag{2.70}$$

FBW, the ratio of the bandwidth to the center frequency, is an important parameter that specifies the performance of the beamformer.

To have a broader beam, a larger ratio of largest eigenvalue of the correlation matrix to the smallest value of the correlation matrix is required.

2.9 Frequency-Domain Beamforming

Frequency-domain beamformer is an element-space beamformer type, shown in Figure 2-8.

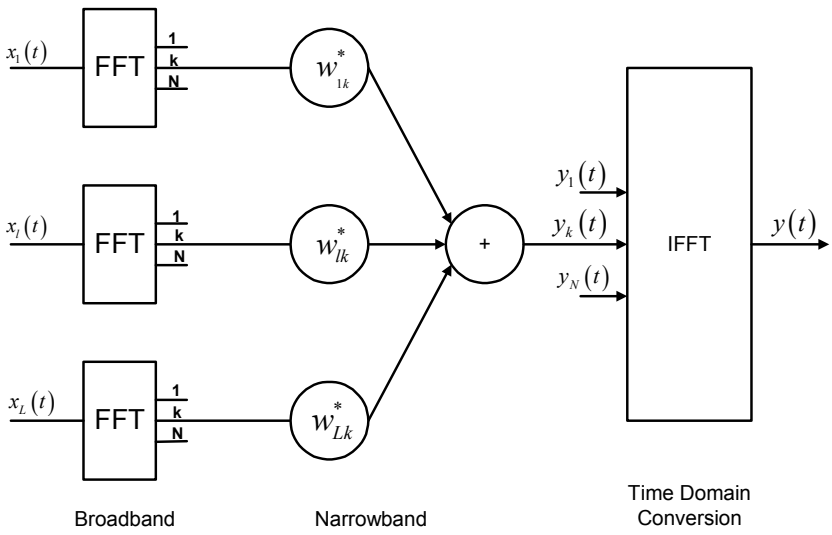

Figure 2-8 Frequency Domain Beamformer

Induced signal in each array element is passed to Fast Fourier Transform (FFT) process. Each frequency bin is weighted and summed.

The weights are adjusted independently to minimize the mean output power of each frequency bin which serves a faster calculation.

The performances of the time- and frequency-domain beamformers are the same for the constraint of signals in each frequency bin being independent.

2.10 Digital Beamforming

Analog beamformig structure is explained under delay-and-sum beamformer part, shown in Figure 2-9.

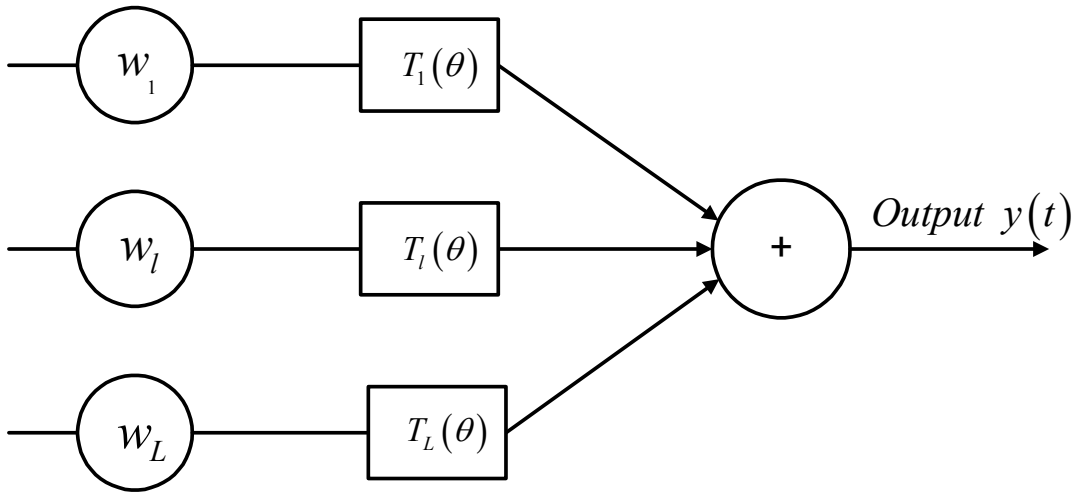

Figure 2-9 Analog Beamformer

The output of the beamformer is

$$y(t) = \sum_{i=1}^{L} w_i x_i \left(t - \tau_i(\theta)\right) \quad . \tag{2.71}$$

By delaying the induced signal on each array element, signals become in phase and then weighted to steer the beam in the look direction.

In digitial beamforming, weighted signals are sampled and stored. The appropriate samples are used to shape the beam. For delay process, a constant value, Δ is used. The arrival time of the induced signal from angle θ_2 to the i^{th} element is

$$\tau_i(\theta_2) = (i-1)\Delta \quad . \tag{2.72}$$

For a signal induced from angle θ_3, shown with B, needs to be delayed $(L-i)\Delta$ seconds, shown in Figure 2-10. For signals coming from angle θ_1, no delay is needed.

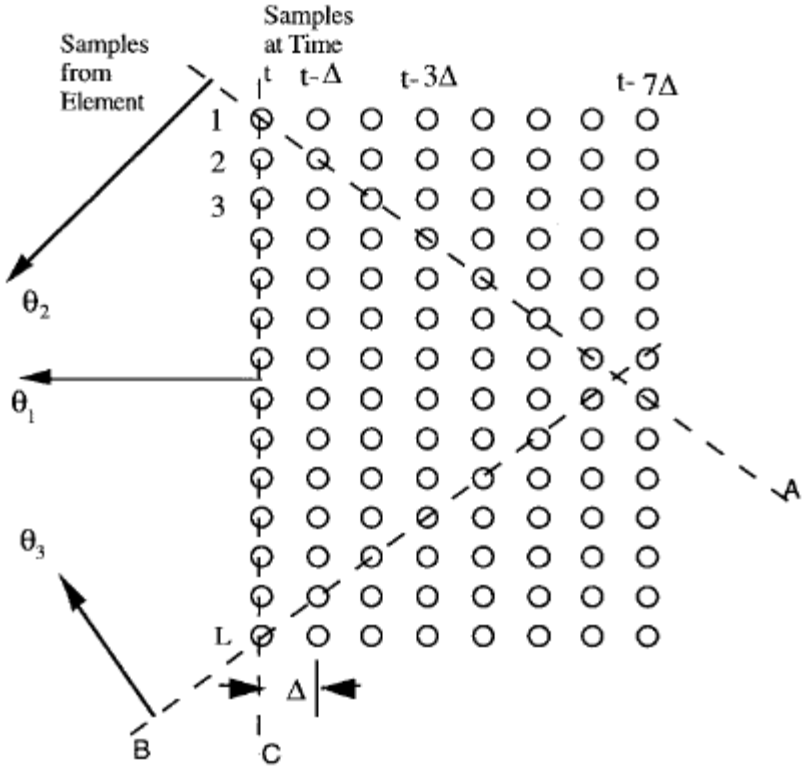

Figure 2-10 Digital Beamforming [1]

As it is seen from the Figure 2-10, the steering directions are limited in $[\theta_2, \theta_3]$ range. To enlarge the direction range, sampling is increased by changing the sampling interval to $\Delta/2$, in Figure 2-11. Larger direction range is seen in Figure 2-10 with θ_4 and θ_5.

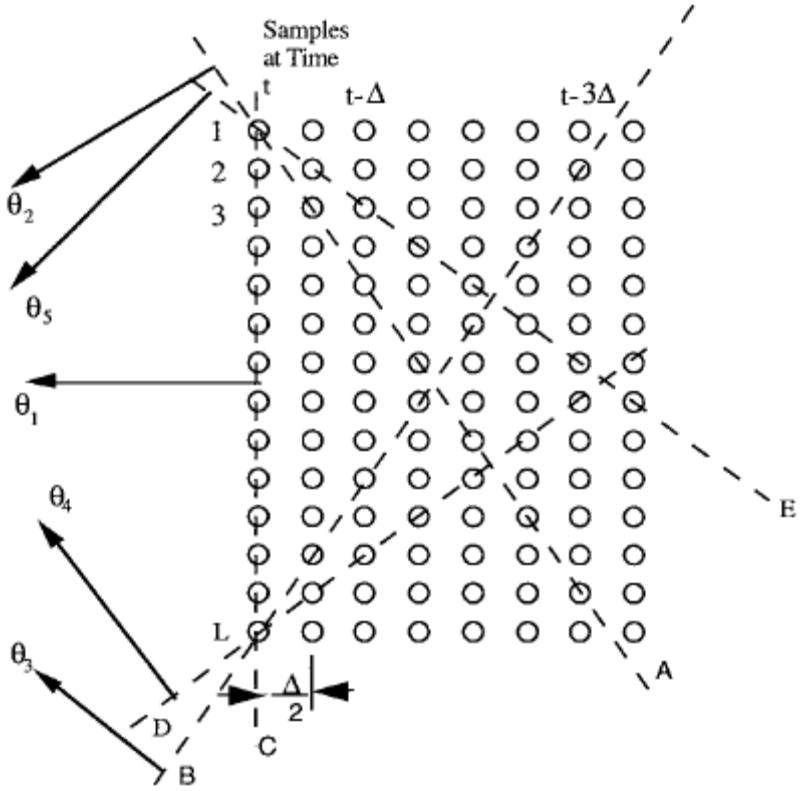

Figure 2-11 Sampled Digital Beamformer [1]

High sampling rates require larger storage capacity, faster input-output devices, ADC's. Digital interpolation is applied for increasing sampling rates by zero padding.

Increasing the array size satisfies narrower beams. In application, it is not possible to increase the array size. Instead of that, extrapolating is applied in digital beamforming.

2.11 Eigenstructure Method

Eigenvalues of correlation matrix, R are seperated into two parts as noise and signal eigenvalues. The eigenvectors, corresponded to each part are calculated.

The eigenvectors of R are orthogonal to each other, assuming that there are L spaces. This set is seperated into signal and noise subspaces.

Weight vector is adjusted to have an approriate response in look direction and to cancel the interferences. It is mentioned above that the weight vector of the desired signal is orthogonal to the steering vector of the interfering signal.

CHAPTER 3

ADAPTIVE BEAMFORMING

Adaptive bemaformers update the weights of the antenna array according to the desired and/or the interfering signals in the environment.

There are many adaptive beamforming schemes, which are referred to adaptive algorithms [1]. Some of them are presented in this chapter. The characteristics, such as the speed of adaption and the mean and variance of the estimated weights are given.

The correlation matrices used for estimating weight vector are not considerable in application.

Since the correlation matrix of noise (R_N) and the correlation matrix of signal (R) are not available, optimal weights are calculated by using the known parameters of signal and array output.

Adaptive algorithms are accomplished to adjust the optimal weights and to shape the appropriate array pattern.

Some of the adaptive algorithms and their parameters are discussed below.

3.1 Sample Matrix Inversion Algorithm

Sample Matrix Inversion (SMI) algorithm estimates the array weights by replacing with its estimate. [1], [2]

N samples of array signal $\underline{x}(n)$ are used to adjust R,

$$R(n) = \frac{1}{N}\sum_{n=0}^{N-1} \underline{x}(n)\underline{x}^{H}(n) \quad . \tag{3.1}$$

The algorithm enables to update R according to the new signal samples,

$$R(n+1) = \frac{nR(n) + \underline{x}(n+1)\underline{x}^{H}(n+1)}{n+1} \quad , \tag{3.2}$$

according to the new samples, weights are also updated, $\underline{w}(n+1)$. R^{-1} is required in calculation of optimal weights

$$R^{-1}(n) = R^{-1}(n-1) - \frac{R^{-1}(n-1)\underline{x}(n)\underline{x}^{H}(n)R^{-1}(n-1)}{1+\underline{x}^{H}(n)R^{-1}(n-1)\underline{x}(n)} \quad . \tag{3.3}$$

Increasing the number of samples concludes a good approximation of R and optimal weight, $n \to \infty$, $R(n) \to R$ and $\underline{w}(n) \to \underline{w}_{MSE}$.

3.2 Least Mean Square Algorithm

The Least Mean Square (LMS) algorithm can be divided into two sets: constraint and unconstrained LMS algorithms.

In constrained algorithm, the weight updates are adjusted under some considerations [1].

In unconstrained algorithm, the weights are updated according to the reference signal, without any direction information of the signal [1], [2].

Gradient of quadratic surface estimation is referred to weight updates. According to the estimation, weights are updated by an amount of step size. Small step size success in calculating appropriate optimal weight values. On the other hand, large step size values success in faster convergence, but result fluctuations around the optimal weights.

3.2.1 Unconstrained LMS Algorithm

Unconstrained LMS is the first type of LMS adaptive algorithm.

The updated weight expression is

$$\underline{w}(n+1) = \underline{w}(n) - \mu \underline{g}\left(\underline{w}(n)\right) \quad , \tag{3.4}$$

μ is a scalar value, known as step size. It reflects how fast the algorithm approaches to optimum weights.

$\underline{g}\left(\underline{w}(n)\right)$ is the unbiased estimate of MSE

$$MSE\left(\underline{w}(n)\right) = E\left\{\left|r(n+1)\right|^2\right\} + \underline{w}^H(n) R \underline{w}(n) - 2\underline{w}^H(n)\underline{z} \tag{3.5}$$

$$\nabla_{\underline{w}} MSE\left(\underline{w}\right)|_{\underline{w}=\underline{w}(n)} = 2R\underline{w}(n) - 2\underline{z} \quad . \tag{3.6}$$

The output of the beamformer is

$$y(n) = \underline{w}^H(n)\underline{x}(n+1) \quad . \tag{3.7}$$

It is seen in equation (3.7) that $\underline{w}(n)$ is used in $(n+1)^{th}$ iteration.

For real applications, R and z are changed with noisy values

$$\underline{g}\left(\underline{w}(n)\right) = 2\underline{x}(n+1)\underline{x}^{H}(n+1)\underline{w}(n) - 2\underline{x}(n+1)\underline{r}(n+1)$$

$$= 2\underline{x}(n+1)\varepsilon^{*}\left(\underline{w}(n)\right) \tag{3.8}$$

The error between the array and the reference signal, $\varepsilon\left(\underline{w}(n)\right)$, is

$$\varepsilon\left(\underline{w}(n)\right) = \underline{w}^{H}(n)\underline{x}(n+1) - \underline{r}(n+1) \tag{3.9}$$

$\mu < 1/\lambda_{\max}$ makes the algorithm behave stable and converge to optimal weight. $\lambda_{\max}$ is the maximum eigenvalue of R.

The convergence speed is represented by the L eigenvectors and l^{th} eigenvalue of R,

$$\tau_l = \frac{1}{2\mu\lambda_l} \quad . \tag{3.10}$$

Larger eigenvalue denotes small convergence speed, τ_l. Large eigenvalues refer to the signal eigenvalues. Smaller eigenvalues denotes large τ_l, which refers to weak signals, noise. For large eigenvalues, the convergence speed of the algorithm takes small values. The algorithm cancels strong signals first.

LMS algorithm is not an appropriate algorithm for nonstationary environment, since it has a slow convergence speed.

The covariance matix of the weights is given as

$$k_{\underline{w}\underline{w}}(n) = E\left\{\left(\underline{w}(n) - \overline{\underline{w}}\right)\left(\underline{w}(n) - \overline{\underline{w}}\right)\right\} \quad , \tag{3.11}$$

$\overline{\underline{w}}$ is the expected value of $\underline{w}(n)$. Algorithm minimizes the MSE given in equation to calculate the optimum weights

$$MSE\left(\underline{w}(n)\right) = MMSE + \underline{V}^{H}(n)R\underline{V}(n) \quad , \tag{3.12}$$

$$\underline{V}(n) = \underline{w}(n) - \underline{w}_{MSE} \quad . \tag{3.13}$$

Equation (3.13) gives the error between estimated and optimal weights. $\underline{V}(n)$ is expected to be equal to zero for infinite iterations.

The difference between the estimated weight in LMS and the optimal weight is named as misadjusment. The gradient step size μ and misadjustment M are given as follows

$$0 < \mu < \frac{1}{4\lambda_{\max}} \quad , \tag{3.14}$$

$$\eta(\mu) \square \sum_{i=1}^{L} \frac{2\mu\lambda_i}{1-2\mu\lambda_i} < 1 \quad , \tag{3.15}$$

$$M = \frac{\eta(\mu)}{1-\eta(\mu)} \quad . \tag{3.16}$$

Increasing μ causes an increase in misadjustment. While increasing μ, the algorithm converges faster to the optimal weights but fluctuates so much around the optimal weights and causes a great miasadjustment. Decreasing μ decreases the convergence speed of the algorithm. For a nonstationary environment, estimated weights stay behind the optimal weights.

3.2.2 Normalized LMS Algorithm

This algorithm is a type of the constant-step-size LMS algorithm [1], [2]. The data-dependent step size is given as,

$$\mu(n) = \frac{\mu_0}{\underline{x}^H(n)\underline{x}(n)} \quad . \tag{3.17}$$

The algorithm convergence to the optimal weight better, and no eigenvalue is needed to be calculated.

3.2.3 Constrained LMS Algorithm

Constrained LMS is the last type of adaptive LMS algorithm. According to the constrained LMS algorithm, the updated optimal weight is

$$\underline{w}(n+1) = P\left\{\underline{w}(n) - \mu \underline{g}\left(\underline{w}(n)\right)\right\} + \frac{\underline{s}_0}{\underline{s}_0^H \underline{s}_0} \tag{3.18}$$

$$P = I - \frac{\underline{s}_0 \underline{s}_0^H}{L} \quad . \tag{3.19}$$

μ is the step size, s_0 is the steering matrix in the look direction. $g\left(\underline{w}(n)\right)$ is the estimate of the gradient of $\underline{w}^H(n) R \underline{w}(n)$ with respect to $\underline{w}(n)$,

$$\underline{g}\left(\underline{w}(n)\right) = \nabla_{\underline{w}}\left(\underline{w}^H R \underline{w}\right)|_{\underline{w}=\underline{w}(n)} = 2R\underline{w}(n) \quad . \tag{3.20}$$

For applications, the above expressions have to be replaced with standard LMS algorithm. Assuming the noisy environment

$$R \to \underline{x}(n+1)\underline{x}^H(n+1) \quad , \tag{3.21}$$

$$\underline{g}\left(\underline{w}(n)\right) = 2\underline{x}(n+1) y^*\left(\underline{w}(n)\right) \quad . \tag{3.22}$$

$y\left(\underline{w}(n)\right)$ is the array output. The gradient step size and the convergence speed are as follows

$$0 < \mu < \frac{1}{2\lambda_{\max}(PRP)} \quad , \tag{3.23}$$

$$\tau_l = \frac{-1}{\ln\left[1-2\mu\lambda_l\left(PRP\right)\right]} \approx \frac{1}{2\mu\lambda_l\left(PRP\right)} \quad . \tag{3.24}$$

λ_l and $\lambda_{\max}$ are the l^{th} and the maximum eigenvalues of PRP. R is

$$R = p_s \underline{s}_0 \underline{s}_0^H + R_N \tag{3.25}$$

$$P\underline{s}_0 = 0 \quad , \tag{3.26}$$

Convergence speed depends on the eigenvalues of $PRP = PR_N P$. Eigenvalues depend on the direction and power of directional sources.

The variance of the gradient and the misadjusment are given as,

$$V_{\underline{g}}\left(\underline{w}(n)\right) = 4\underline{w}^H(n) R \underline{w}(n) R \tag{3.27}$$

$$M = \frac{\mu\sum_{i=1}^{L-1}\frac{1}{1-\mu\lambda_i\left(PRP\right)}}{1-\mu\sum_{i=1}^{L-1}\frac{1}{1-\mu\lambda_i\left(PRP\right)}} \quad . \tag{3.28}$$

The constrainted LMS algorithm as expressed above is

$$\underline{w}(n+1) = P\underline{w}(n) + \frac{\underline{s}_0}{\underline{s}_0^H \underline{s}_0} - \mu P \underline{g}\left(\underline{w}(n)\right), \tag{3.29}$$

where $\underline{g}\left(\underline{w}(n)\right)$ is

$$\underline{g}\left(\underline{w}(n)\right) = \underline{x}(n+1)\underline{x}^H(n+1)\underline{w}(n) \quad , \tag{3.30}$$

and $\underline{x}(n)$ is

$$\underline{x}(n) = m_s(n)\underline{s}_0 + \underline{x}_N(n) \quad . \tag{3.31}$$

$m_s(n)$ is the modulating function. $\underline{x}_N(n)$ is the array receiver vector.

$$P\underline{g}\left(\underline{w}(n)\right) = P\underline{x}_N(n+1)\underline{x}_N^H(n+1)\underline{w}(n) + m_s^*(n+1)P\underline{x}_N(n+1)\underline{s}_0^H\underline{w}(n) \quad (3.32)$$

$m_s^*(n+1)$ is a random variable which has the variance of the signal power in the look direction.

Weight estimation is sensitive to the signal power in standard LMS algorithm. It is better to choose a low step size in an environment where strong signals exist.

The standard LMS algorithm adjusts the optimal weights by the new samples of correlation matrix. However, the recursive algorithm uses previous samples also.

$$R(n+1) = \frac{nR(n) + \underline{x}_N(n+1)\underline{x}_N^H(n+1)}{n+1} \quad (3.33)$$

The correlation matrix is used to make an estimation of

$$\underline{g}\left(\underline{w}(n)\right) = 2R(n+1)\underline{w}(n) \quad . \quad (3.34)$$

Variance of the estimated gradient is

$$V\underline{g}\left(\underline{w}(n)\right) = \frac{4}{(n+1)^2}\underline{w}^H(n)R\underline{w}(n)R \quad . \quad (3.35)$$

The gradient is decreased with a ratio of $(n+1)^2$, so the recurcive LMS algorithm is less sensitive to the signal power than the standard LMS algorithm.

The improved LMS algorithm's correlation function of a linear array, equispaced elements is,

$$R \equiv \begin{bmatrix} r_0 & r_1 & \cdots & r_{L-1} \\ r_H^* & \ddots & & \\ \vdots & & \ddots & \\ r_{L-1}^* & & & r_0 \end{bmatrix} \quad , \tag{3.36}$$

r_i are the L correlation lags

$$r_i(n) = \frac{1}{N_i} \sum_l x_l(n) x_{l+i}^*(n) \qquad i = 0,1,\cdots,L-1 \quad . \tag{3.37}$$

The improved LMS algorithm works better in the presence of strong signals. Increasing the step size has no effect on algorithm's stability.

3.3 Recursive Least Square Algorithm

Recursive Least Square (RLS) algorithm uses the inverse of correlation matrix, $R^{-1}(n)$. This approximation have a better performance in a large eigenvalue spread. The weight update is

$$\underline{w}(n) = \underline{w}(n-1) - R^{-1}(n)\underline{x}(n)\varepsilon^*\left(\underline{w}(n-1)\right) \tag{3.38}$$

and $R(n)$ is

$$R(n) = \delta_0 R(n-1) + \underline{x}(n)\underline{x}^H(n) \quad ,$$

$$= \sum_{k=0}^{n} \delta_n^{n-k} \underline{x}(k)\underline{x}^H(k) \quad . \tag{3.39}$$

δ_0 is the forgetting factor, $1/1-\delta_0$ is the memory. The update of the correlation matrix is given as

$$R^{-1}(n) = \frac{1}{\delta_0}\left[R^{-1}(n-1) - \frac{R^{-1}(n-1)\underline{x}(n)\underline{x}^H(n)R^{-1}(n-1)}{\delta_0 + \underline{x}^H(n)R^{-1}(n-1)\underline{x}(n)} \right] \tag{3.40}$$

where

$$R^{-1}(0)=\frac{1}{\varepsilon_0}I, \qquad \varepsilon_0>0 \quad . \tag{3.41}$$

The constraint of the RLS algorithm for convergence, independent of the eigenvalues of R, is to minimize

$$J(n)=\sum_{k=0}^{n}\delta_0^{n-k}\left|\varepsilon(k)\right|^2 \quad . \tag{3.42}$$

According to the convergence speed, RLS is the most efficient algorithm.

3.4 Constant Modulus Algorithm

The constrained of the gradient-based algorithm, Constant Modulus Algorithm (CMA) is to minimize the following

$$J(n)=\frac{1}{2}E\left\{\left(\left|y(n)\right|^2-y_0^2\right)^2\right\} \quad . \tag{3.43}$$

The weight update is

$$\underline{w}(n+1)=\underline{w}(n)-\mu\underline{g}\left(\underline{w}(n)\right) \quad , \tag{3.44}$$

y_0 is the desired amplitude when there is no interfering signal, $y(n)=\underline{w}^H(n)\underline{x}(n+1)$ is the array output. The gradient cost function $\underline{g}\left(\underline{w}(n)\right)$ is

$$\underline{g}\left(\underline{w}(n)\right)=2\varepsilon(n)\underline{x}(n+1) \tag{3.45}$$

and

$$\varepsilon(n)\square\left(\left|y(n)\right|^2-y_0^2\right)y(n) \tag{3.46}$$

Under the above considerations, the weight update is

$$\underline{w}(n+1) = \underline{w}(n) - 2\mu\varepsilon(n)\underline{x}(n+1) \quad . \tag{3.47}$$

3.5 Conjugate Gradient Method

The algorithm tries to solve the following equation

$$A\underline{w} = \underline{b} \tag{3.48}$$

$\underline{w}$ is the weight vector, A matrix is composed of the sampled signals from array elements. $\underline{b}$ has the samples of desired signal. The algorithm is trying to decrease the error between the desired and the array output under a determined value. The error variable, named as the residual vector is

$$\underline{r} = \underline{b} - A\underline{w} \tag{3.49}$$

The weight update is

$$\underline{w}(n+1) = \underline{w}(n) - \mu(n)\underline{g}(n) \quad . \tag{3.50}$$

where $\underline{g}(n)$, the direction vector and $\mu(n)$,step size are as

$$\underline{g}(n) = A^H \underline{r}(n) \ , \tag{3.51}$$

$$\mu(n) = \frac{\left|A^H \underline{r}(n)\right|^2}{\left|A^H \underline{g}(n)\right|^2} \quad . \tag{3.52}$$

The residual update and direction vector update expressions

$$\underline{r}(n+1) = \underline{r}(n) + \mu(n) A\underline{g}(n) \ , \tag{3.53}$$

$$\underline{g}(n+1) = A^H \underline{r}(n+1) - \alpha(n)\underline{g}(n) \ , \tag{3.54}$$

where

$$\alpha(n) = \frac{\left|A^H \underline{r}(n+1)\right|^2}{\left|A^H \underline{r}(n)\right|^2} \quad . \tag{3.55}$$

The error surface, $\underline{r}^H(n)\underline{r}(n)$ is minimized in L iteration.

CHAPTER 4

NEURAL NETWORK APPROACH FOR BEAMFORMING

Neural Network algorithms have been more popular in signal processing applications. According to the improving mobile communication, Global Positioning Sytem (GPS) and Radar technologies, faster beamforming algorithms are needed. Since the number of users and the interfering signals increases, the communication systems require to track the users continuously while they are moving, and to put nulls in the directions of interferences. Neural Networks (NN) have good performances in accordance of these needs, and NN can easily be implemented for these applications.

The main idea of NN applications is to define input and output pairs for the training phase. The inputs of the training phase have to be chosen carefully, since the NN is going to make an optimization for a new, unseen input according to the trained input and output pairs.

In this thesis work, beamforming applications are applied with NN. The main idea of this beamformer is to direct the antenna array patterns to the desired signal directions and to put nulls in the directions of interferences.

The inputs are chosen as the correlation matrices of the incoming signals from sources. The inputs have all the possibilities of the direction of arrival (DOA)

information of the incoming signals. The outputs are the weights of the antenna elements with respect to each correlation matrices. These weights are used to steer and shape the antenna pattern.

Radial Basis Function Neural Network (RBFNN) is used for training phase. A brief information of RBFNN is described in Section 4.1.

Two applications are studied for NN beamforming. First one is the linear antenna array application RBFNN's and the second application is over cylindrical array with cylindrical microstrip patch antenna elements via RBFNN.

A part of linear array application in this thesis is parallel to the work given in [3] of A. H. Zooghby, C. G. Christodoulou, and M. Georgiopoulos. The network is composed of the correlation matrices of the incoming signals to the linear array. The outputs are the optimum weights of each antenna element. The weights are calculated according to the paper in [3].

The application is implemented for an angular range of interest of $\left[-90^0, 90^0\right]$. Since all the possibilies of single target or multiple targets in any direction are taken into consideration for training, the algorithm has the knowledge of the DOA information of the whole angular range of interest.

In the performance phase, the correlation matrix of an incoming signal from any direction is given as the input to the NN. The direction of the incoming signal is not necessary to be known. Optimum weights are derived by training NN.

Second approach is implemented for a cylindrical array application. The cylindrical array has twelve Circular Microstrip Patch Array (CMPA) elements. Since the antenna elements are directive and the array allows a full coverage of 360^0, this array is used as a performance application of NN beamformer.

The main idea of the NN beamforming idea is identical with the linear array application. The main difference is that the angular range of 360^0 is divided into

twelve sectors, having an angular coverage of 30^0. Each CMPA element is placed in the middle of the corresponding sector, as mentioned in Figure 4-9.

The inputs of the NN are the correlation matrices of all possible incoming signals form all directions of 360^0. The outputs are the calculated weights, according to the formulation given in [3].

The main advantage of cylindrical array application over the linear array one is that the NN in cylindrical array needs less time for training. In the first implementation, there are more possibilities for a whole range of 180^0. In cylindrical array application, the possibilities are limited with 30^0 for each twelve sectors.

For an incoming signal to any sector, three array elements are activated. For $\text{sec}tor-i$, the activated elements are $(i-1)^{th}$, i^{th} and $(i+1)^{th}$ elements. The beamforming for an angle included in $\text{sec}tor-i$ is implemented by just three antenna elements and only for 30^0 of interest.

The DOA information of the incoming signal is not necessary to be known. The NN optimizes the beam for the signal in the performance phase. The array pattern is shaped according to the new signal.

Section 4.1 involves an introduction to RBFNN. In Section 4.2, the formulation and the algorithm of the linear array approach is given. Section 4.3 presents the formulation and the algorithm of CMPA application. The input and output pairs for training phase of each two implementation are described briefly in the following chapters.

4.1 Radial Basis Function Neural Network

The Radial Basis Function Neural Network (RBFNN) is a three layered feed-forward network. Since RBFNN network has fast learning speed and needs less

iterations for converging to the target values, it is used for the beamforming applications of this thesis work.

The main idea of RBFNN is to make a junction between the inputs and outputs of any size. This junction is a multifunction of input. The network tries to fit the [input, output] pairs for this multifunction. [4], [10]

The architecture of the RBFNN is shown in Figure 4-1.

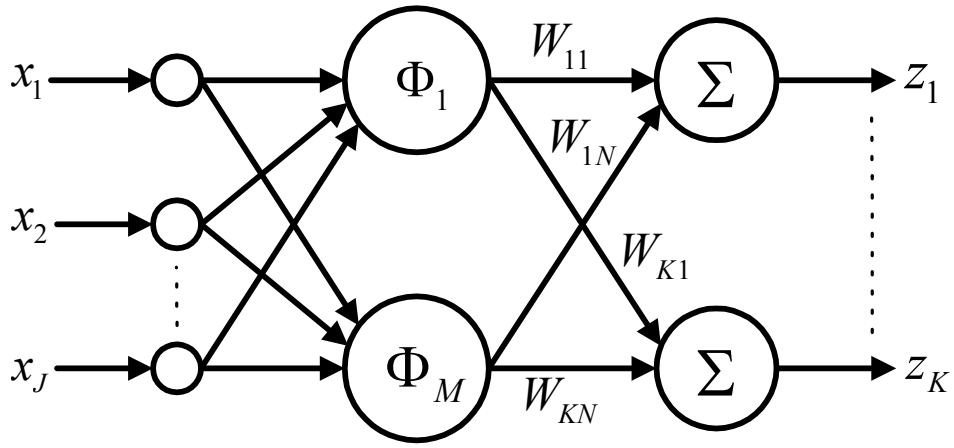

Figure 4-1 Radial Basis Function Neural Network

The mapping between the input-hidden layers is nonlinear. There is a linear combination between hidden and outpul layers.

The mapping function is expressed as

$$F(x) = \sum_{1}^{N} w_i \varphi\left(\|x - x_i\|\right) \quad . \tag{4.1}$$

N is the number of functions. x_i are the centers of the radial basis function. φ can be considered as

$$\varphi(x) = e^{\frac{-x^2}{2\sigma^2}} \quad , \tag{4.2}$$

$\varphi(x)$ is called as the transfer function of the neural network's input-output pairs.

In this thesis, two MATLAB commands of RBFNN are used for training and performance phases of the beamformer.

"newrb" is the training MATLAB command of the RBFNN. The inputs of this command are the inputs, outputs, the mean squared error goal, spread of the RBFNN and maximum number of neurons. The inputs are the correlation matrices, and the outputs are the weights.

"sim" is the performance MATLAB command. The inputs of this command are trained NN, input, network targets. The command returns the corresponding output weights of given input correlation matrix.

4.2 Linear Array

The linear array beamforming approach is taken from [1], shown in Figure 4-2.

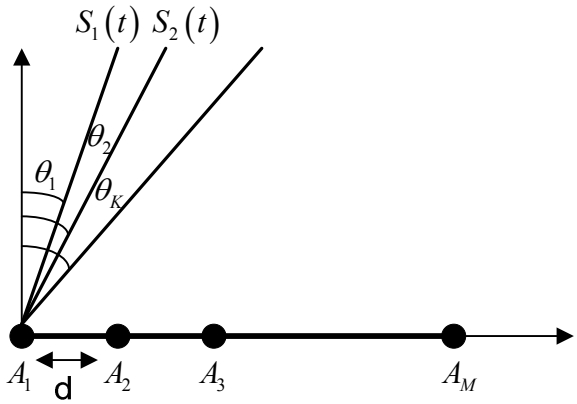

Figure 4-2 Linear Antenna Array

It is assumed that there are M isotropic equispaced antenna elements, with distance d. There are K number of sources coming from angles θ_i, $i = 1,2\cdots,K$ which is $\left[-90^0, 90^0\right]$. The sources are in the far field.

The correlation matrices are calculated according to the K incoming signals. The number of incoming signals are 181, which includes 0^0. The distance between the antenna elements are taken as $\lambda/2$.

The performance analysis of this application is examined by changing the number of antenna elements, the SNR value and for multiple target applications, the angular separation between the incoming signals.

In Section 4.2.1, the formulation of the electric field, correlation matrix and the weight calculations are given. These informations are used for training and the performance phase of the linear array beamforming NN. The algorithms of the training and performance phases are described in Section 4.2.2.

4.2.1 Formulation

Assuming that there are K antenna elements, the induced signal to each antenna element is calculated by,

$$X_i(t) = \sum_{m=1}^{K} S_m(t) e^{-j(i-1)k_m} + n_i(t) \quad , \tag{4.3}$$

S_m are the signals coming from each signal source, $n_i(t)$ is a zero mean, statistically independent white noise, with variance σ^2, k_m is given as

$$k_m = \frac{w_0 d}{c} \sin\theta_m \tag{4.4}$$

d , is the distance between the array elements, w_0 is the angular frequency, c is the speed of light.

Equation (4.3) can be rewriten in matrix form as

$$X(t) = AS(t) + N(t) \tag{4.5}$$

X and N are $[M \times 1]$-sized vectors. S is a $[K \times 1]$-sized vector. A is a $[M \times K]$-sized matrix. This matrix is the steering matrix for linear array. Steering matrix is composed of the following,

$$A_{im} = e^{-j(i-1)k_m} \tag{4.6}$$

where k_m is given in equation (4.4).

After deriving the induced signal to the linear array, the correlation matrix of each incoming signal is calculated. As mentioned earlier, the correlation matrices are used for deriving the inputs of the NN for training and for the performance phases.

The correlation function is derived from the induced signals on each array element, given as

$$\begin{aligned} R &= E\left\{X(t)X(t)^H\right\} \\ &= AE\left\{S(t)S(t)^H\right\}A^H + E\left\{N(t)N(t)^H\right\} \end{aligned} \tag{4.7}$$

The first row of the correlation matrix is taken into account for calculating the Z vector, [5]. The derived Z vector is then given as the input of NN.

The correlation function and vector-b are given as,

$$R = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \text{ and } b = \begin{bmatrix} R_{11} \\ R_{12} \\ R_{13} \end{bmatrix} \tag{4.8}$$

Z vector is obtained from b

$$Z = \frac{b}{\|b\|} \tag{4.9}$$

The elements of the correlation matrix R are complex values. Since the NN does not accept the complex values, real and imaginary parts of each element of the matrix is considered seperately. The size of Z vector in equation (4.9) is $[M \times 1]$. Resizing the vector, by seperating it into its real and imaginary parts changes the size of Z to $[2M \times 1]$. Z vector is given as the input to the RBFNN. The next step is to calculate the outputs of the NN.

The outputs are the optimum weights of the linear array elements for corresponding DOA's. The formulation of weight calculation is given in [1]. The weights minimize the signals received from interferences and maximizes the array response for the desired signal directions.

The optimum weights are calculated as,

$$\hat{w}_{opt} = R^{-1} S_d \left[S_d^H R^{-1} S_d \right]^{-1} r \tag{4.10}$$

The main constraint of the optimum weights is to minimize the mean output power.

R is the correlation matrix, defined above.

$$R = E\left\{ X(t) X(t)^H \right\} \tag{4.11}$$

S_d is the steering matrix of the desired signals. The information of the directions of each desired signal is given in this matrix. Assuming that the scanned azimuth angle is $[-\theta, \theta]$, and the number of the desired signals is V, the desired steering matrix S_d has a size of $\left[(2\theta+1) \times V\right]$. In this thesis work, assuming that the number of desired signals is two, the size of S_d is $[181 \times 2]$.

The steering matrix of the desired signals is expressed as,

$$S_d = \left[S_d(\theta_1) \quad S_d(\theta_2) \quad \cdots \quad S_d(\theta_V) \right] . \tag{4.12}$$

$S_d(\theta_i)$ is

$$S_d(\theta_i) = \left[1 \quad e^{-jk_i} \quad e^{-j2k_i} \quad \cdots \quad e^{-j(M-1)k_i} \right] \tag{4.13}$$

After the training phase, the performance phase is accomplished to obtain the optimum weights for new incoming signals.

For a new incoming signal of $S(t)$, the correlation matrix, b and Z vectors are calculated according to the equations (4.7), (4.8) and (4.9). The outputs are derived by performing the trained NN. The outputs are the weights which shapes the beams.

The array response of the incoming signal is derived by multiplying the weights with each antenna element for each direction, given by the following formulations,

$$Array\ Pattern = w_{opt}^{H} X \tag{4.14}$$

4.2.2 Algorithm

The algorithm is composed of a neural network with multiple input and output pairs. These multiple pairs are trained by RBFNN, as shown in Figure 4-3.

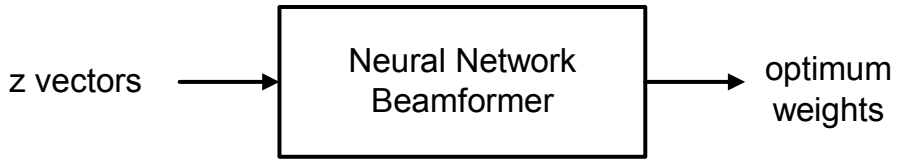

Figure 4-3 Neural Network Beamformer

The algorithm can be considered for two cases being single target and multiple target cases.

First case deals with single target presence in the angular range of interest of $\left[-90^0, 90^0\right]$. The total number of possibilities that a single target can be present in any angle of the angular range is 181^0. There are 181 cases of correlation matrices and 181 cases of weight vectors with 1^0 steps resolution.

The second case deals with multiple targets and/or target+interfering signals. The number of all possibilities of two signals presence simultaneously is 16290. The number of all possibilities of single signal+single interference signal presence simultaneously is 32580.

The increase in the number of possibilities results an increase in the training time. The cases of three and more targets are not implemented. Eventhough the main idea is similar, more training time is required.

The training phase and the performance phase of the algorithm are presented in the following sections.

4.2.2.1 Training Phase

The training phase is composed of calculating the input and the output pairs of the neural network. As mentioned above, the inputs are the Z vectors.

Z vectors are calculated from the correlation matrix, as given in equations (4.8)-(4.9). The optimum weigths are derived from the equation (4.10).

It is assumed that the number of array elements is M and the number of desired signals is K and the angular range of interest is $\left[-90^0, 90^0\right]$. The number of antenna elements is given as an.

Assuming that the number of possibilities of the inputs is No_Pos, there are No_Pos number of correlation matrices of size $an \times an$, No_Pos number of Z vectors of size $[2M \times 1]$ and No_Pos number of weight vectors of size $an \times 1$.

The Z vector is calculated for all training sets of number No_Pos. For all training sets, No_Pos number of optimum weights are derived.

The optimum weight formulation contains the steering matrix of desired signals. Assuming that the total number of desired and/or interfering signals are No_des_int. The size of S_d is $[181, No\_des\_\text{int}]$. The term r in the equation (4.10) is the characteristic parameter that determines if the signals are interfering or desired signals. If there are two desired signals, $r = [1 \quad 1]$. If there is a single desired and a single interfering signal, $r = [1 \quad 0]$.

According to the above calculations, optimum weigths are calculated. The input and output pairs are obtained, $\left(Z_{2M \times No\_Pos}, w_{opt,an \times No\_Pos}\right)$.

In MATLAB, "**newrb"** command is used to train the NN for the given input and output pairs. The trained network is saved in the memory as data file. This data file is used for beamforming in the performance phase.

4.2.2.2 Performance Phase

In the performance phase, it is assumed that there comes a signal from an unknown direction. According to the incoming signal, the correlation matrix R, b vector and the Z vector are calculated.

The calculated Z vector is presented as the input of the trained network which is saved at the end of the training phase.

The NN tries to make an optimization for the input Z vector and outputs w_{opt}. The performance phase of NN is implemented by "**sim**" command in MATLAB.

The derived w_{opt} is then used to obtain the array pattern that is formed for desired and interfering signals by equation (4.14).

4.3 Cylindrical Array With Microstrip Patch Antenna Elements

Cylindrical Microstrip Patch Antenna (CMPA) is a performance application of the NN beamformer. There are twelve CMPA elements, placed on a cylinder. This array structure has several advantages. The first advantage of this array is the total coverage of 360^0. According to the architecture of the array, the antenna elements cover the full range in azimuth. The second advantage is that CMPA elements have enough directivity for forming the beams of the array to the directions of the desired signals and to put nulls in the directions of interferences.

The third advantage is that the angular range is divided into twelve sectors. Assuming that a signal is coming from an angle coresponds to $sector-i$, the activated elements are $(i-1)^{th}$, i^{th} and $(i+1)^{th}$ elements. The beamforming for an angle included in $sector-i$ is implemented by just three antenna elements and only for 30^o of interest. This reduces the size and the time needed for the training phase of the NN.

Single element geometry of the array is given in Figure 4-4.

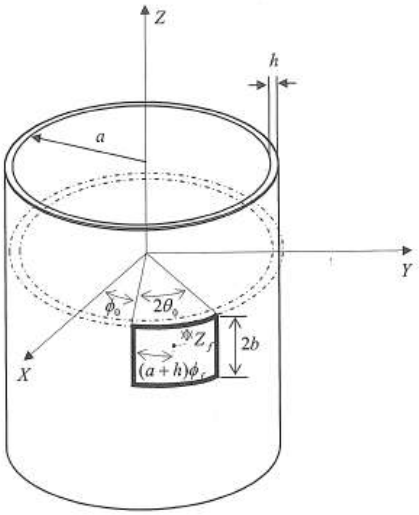

Figure 4-4 Cylindrical Microstrip Patch Antenna Element [4]

a is the radius of the cylinder, h is the height of the substrate. $2\theta_0 a$ and $2b$ are the dimensions of the patch. The patch is placed on the angle ϕ_0. Z_f and ϕ_f gives the position of the coaxial probe feed.

The performance analysis of this application is examined by changing the SNR value and the angular separation between the incoming signals for multiple target case.

In Section 4.3.1, the formulation of the electric field, the correlation and the weight calculations are given. The algorithms of the training and performance phases are described in Section 4.3.2. These informations are used for training and the performance phase of the linear array beamforming NN.

4.3.1 Formulation

The NN of the cylinrical array application is composed of Z and w_{opt} vectors, similar to the linear array NN architecture.

The calculations of the induced signals, electrical field of single patch antenna elements, the total array pattern, input and output pairs are described in the following sections.

4.3.1.1 Single Patch Element

The formulation and the pattern of single patch antenna element of the cylindrical array is given in this section.

The field equation is given as

$$\frac{1}{\rho^2}\frac{\partial^2(\rho E\phi)}{\partial\rho\partial\phi}-\frac{1}{\rho^2}\frac{\partial^2 E_\rho}{\partial\phi^2}-\frac{\partial^2 E_\rho}{\partial z^2}+\frac{\partial^2 E_z}{\partial z\partial\rho}-k^2E_\rho=0 \tag{4.15}$$

$$k^2=w^2\mu\in \tag{4.16}$$

The electric field is composed of E_ρ component only in the cavity model.

By using the cavity model approximation and by the assumption of $h<<a$, E_ρ yields ([5]),

$$E_\rho=\psi_{mn}=E_0\cos\left[\frac{m\pi}{2\theta_0}(\phi-\phi_0)\right]\cos\left(\frac{n\pi z}{2b}\right) \tag{4.17}$$

$$k^2=k_{mn}^2=\left(\frac{m\pi}{2(a+h)\theta_0}\right)^2+\left(\frac{n\pi}{2b}\right)^2 \quad . \tag{4.18}$$

The magnetic currents radiating along the cylinder are

$$\overline{M}=E_\rho\ \hat{p}\times\hat{n} \quad . \tag{4.19}$$

The electric field components are

$$E_\theta = E_0 \frac{e^{-jk_0 r}}{\pi r} \sin\theta \sum_{p=-\infty}^{\infty} e^{jp\phi} j^{p+1} f_p\left(-k_0 \cos\theta\right) \quad , \tag{4.20}$$

$$E_\phi = -\frac{E_0 k_0}{w\mu_0} \frac{e^{-jk_0 r}}{\pi r} \sin\theta \sum_{p=-\infty}^{\infty} e^{jp\phi} j^{p+1} g_p\left(-k_0 \cos\theta\right) \quad , \tag{4.21}$$

f_p and g_p are

$$f_p(u) = \frac{jw\in_0 \overline{M}_\phi(p,u)}{\left(k_0^2 - u^2\right) H_p^{(2)}\left(a\sqrt{k_0^2 - u^2}\right)} \quad , \tag{4.22}$$

$$g_p(u) = \frac{jw\in_0 \overline{M}_\phi(p,u)}{\left(k_0^2 - u^2\right) H_p^{(2)'}\left(a\sqrt{k_0^2 - u^2}\right)} \left[-\overline{M}_z(p,u) + \frac{pu\overline{M}_\phi(p,u)}{a\left(k_0^2 - u^2\right)} \right] \quad , \tag{4.23}$$

where

$$\overline{M}_\phi(p,u) = \frac{1}{2\pi} \int_0^{2\pi} d\phi \int_{-\infty}^{\infty} dz M_\phi(a,\phi,z) e^{-jp\phi} e^{-juz} \quad , \tag{4.24}$$

$$\overline{M}_z(p,u) = \frac{1}{2\pi} \int_0^{2\pi} d\phi \int_{-\infty}^{\infty} dz M_z(a,\phi,z) e^{-jp\phi} e^{-juz} \quad , \tag{4.25}$$

$H_p^{(2)}$ is the Hankel function of second kind. The electric fields can be rewritten as

$$E_{\theta,mn} = \frac{E_0 h e^{-jk_0 r}}{2\pi^2 r \sin\theta} \left[1 - (-1)^n e^{-jk_0 b\cos\theta}\right] \sum_{p=-\infty}^{\infty} \frac{e^{jp(\phi-\phi_0)} j^{p+1} I(\theta_0, m, -p)}{H_p^{(2)}\left(k_0 a \sin\theta\right)} \quad , \tag{4.26}$$

$$E_{\phi,mn} = -j \frac{E_0 h e^{-jk_0 r}}{2\pi^2 r a} I(b, n, -k_0 \cos\theta) \sum_{p=-\infty}^{\infty} \frac{e^{jp(\phi-\phi_0)} j^{p+1}}{H_p^{(2)'}\left(k_0 a \sin\theta\right)} \left[1 - (-1)^n e^{-j2p\theta_0}\right]$$

$$-j\frac{E_0 h e^{-jk_0 r}}{2\pi^2 ra}\frac{\cos\theta}{k_0\sin^2\theta}\left[1-(-1)^n e^{-j2k_0 b\cos\theta}\right]\sum_{p=-\infty}^{\infty}\frac{e^{jp(\phi-\phi_0)}j^{p+1}pI(\theta_0,m,-p)}{H_p^{(2)'}(k_0 a\sin\theta)},\quad(4.27)$$

I 's are

$$I(b,n,-k_0\cos\theta)=\int_{-2b}^{0}\cos\left(\frac{n\pi z}{2b}\right)e^{-juz}dz\quad,\qquad(4.28)$$

$$I(\theta_0,m,-p)=\int_{0}^{2\theta_0}\cos\left(\frac{m\pi\phi}{2\theta_0}\right)e^{-jp\phi}d\phi\quad.\qquad(4.29)$$

The electric field pattern of one microstrip patch antenna is shown in Figure 4-5 and Figure 4-6.

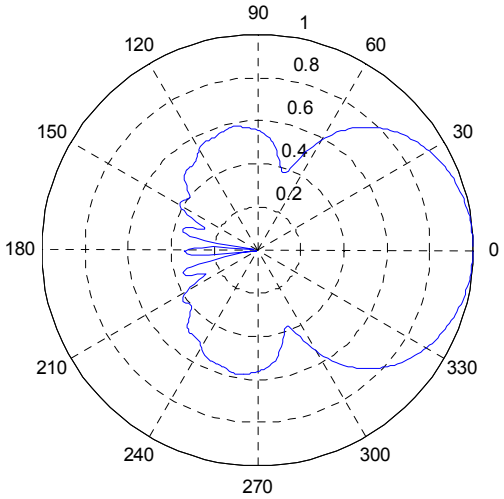

Figure 4-5 The Polar Plot of The Electric Field Pattern of One Microstrip Patch Antenna

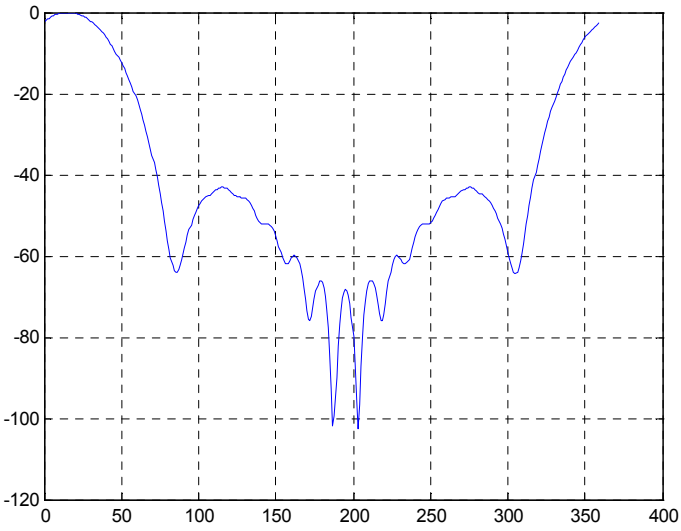

Figure 4-6 The Plot of The Electric Field Pattern of One Microstrip Patch Antenna

The dimensions of the patch are taken as $L = 38.3mm$, $2\theta_0 = 18.3696^o$, $w = 42mm$. Cylinder has a radius $a = 131mm$. $\theta_0 = \phi_0 = 18.3696^o / 2$. Height of the substrate $h = 1.600$, the permittivity $\in_r = 4.4$.

4.3.1.2 Cylindrical Patch Array

Assuming that there are three antenna elements in each sector beamforming calculation, the induced signal to each sector is calculated by equation (4.3).

There are twelve microstrip patch antennas placed on the cylinder, Figure 4-8. Each of them are similar and the antennas are centered beginning from 0^0. The n^{th} element is positioned on

$$\phi_n = \frac{2\pi n}{12} \; in \; radians \qquad (4.30)$$

The total signal induced on each antenna element of the twelve-element-patch array is

$$X_i(t) = \sum_{m=1}^{K} S_m(t) E_T(\phi_m) e^{-jk_m} + n_i(t) \quad i = 1,2,\cdots,12 \tag{4.31}$$

$n_i(t)$ is a statistically independent white noise with zero mean.

X and N are $[3\times 1]$-sized vectors. S is a $[3\times 30]$-sized matrix. K is the number of angular range. A is a $[3\times K]$-sized matrix, composed of

$$E_T = E_\theta + E_\phi \tag{4.32}$$

$$k_m = \frac{\omega_0}{c} a\cos\left(\phi_m - \frac{2\pi(i-1)}{12}\right) \tag{4.33}$$

a is the radius of the cylinder, c is the speed of light in free-space, and ω_0 is the angular center frequency of the signal.

The total induced signal can be represented in matrix form

$$X(t) = AS(t) + N(t) \tag{4.34}$$

X and N are $[3\times 1]$-sized vectors. S is a $[3\times 30]$-sized matrix. K is the number of antenna elements. A is a [3xK]-sized matrix, composed of

$$A_{im} = E_T(\phi_m) e^{-jk_m} \tag{4.35}$$

Total electric field pattern of the array is given in Figure 4-7.

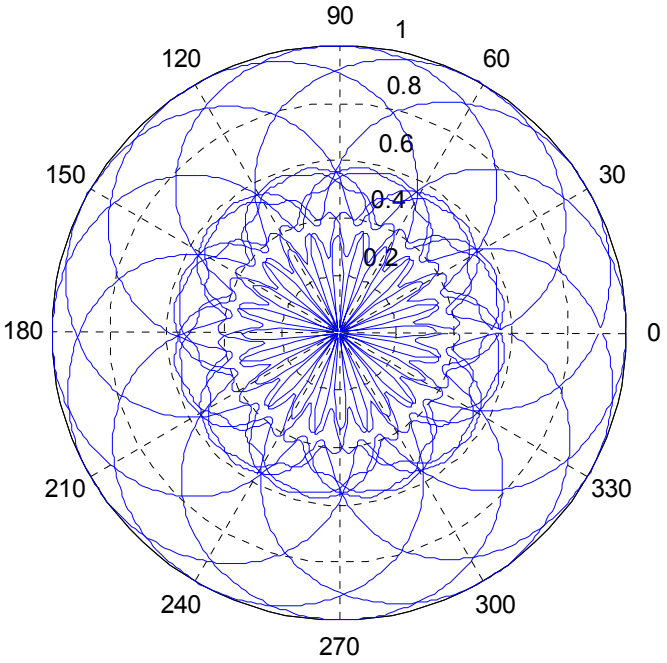

Figure 4-7 Total Electric Field Pattern of CMPA

The schematical expression of twelve-element cylindrical microstrip patch array is in Figure 4-8.

The angular range is divided into 12 sectors. One element is in the middle of each sector. As it is seen from Figure 4-8, each sector is affected by array elements in that sector and in the adjacent sectors, which have more than 20 dB signal level according to their radiation patterns.

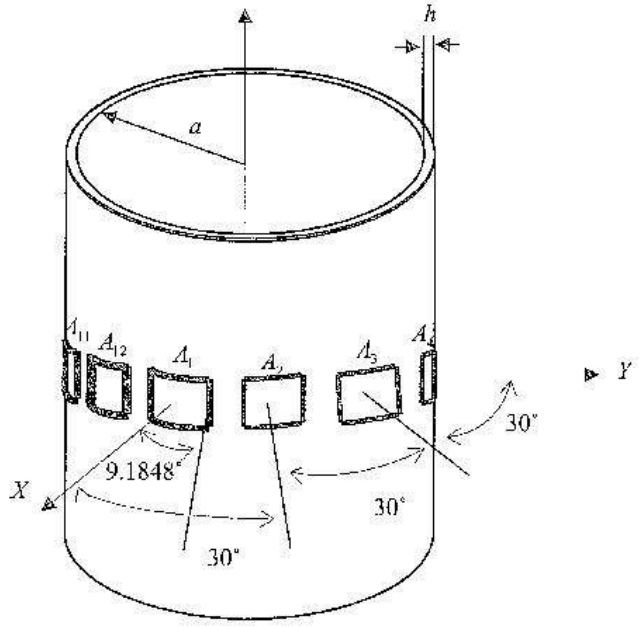

Figure 4-8 The geometry of twelve-element cylindrical array [5]

Since sectors are formed by three antennas, each sector consists of radiation patterns of three array elements. There are four sector groups, changing according to the incoming signal's angle of arrival. For example, it is assumed that the signal is coming from θ_3. Signal coming from θ_3 activates the Sector-3. Since each sector is affected by array elements in that and in adjacent sectors, the second, third and the fourth array elements are grouped. The X vector, the steering matrix and A matrix are calculated by these three array elements.

The second, third and the fouth elements are used for the signals coming to the Sector-3, in a 30^0 of range. If a new signal comes from θ_4 simultaneously, which is in Sector-4's angular range, the algorithm does not take this signal into account. It's because of that Sector-4's sector group includes third, fourth and fifth array elements. The third and the fouth array elements are used for Sector-3 for signal(s) coming form θ_3.

According to the above assumption, the new signal has to come from angle θ_6 simultaneously, which is in Sector-6. This sector group is composed of the fifth, sixth and the seventh array elements. They are not used for Sector-3.

The sector group is composed of three array elements and an angular range of 30^0 is in interest. The electric fields, the correlation matrices and the optimum weights are calculated for these two constraints.

The input of the network is the correlation function, as it is in the linear array assumption. The correlation function is derived from the induced signals on each array element, given as

$$R = E\left\{X(t)X(t)^H\right\} \ ,$$

$$= AE\left\{S(t)S(t)^H\right\}A^H + E\left\{N(t)N(t)^H\right\} \quad . \tag{4.36}$$

The first row of the correlation matrix is taken into account in further calculations for network inputs [5]. A vector, b is obtained from the correlation matrix.

$$R = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \text{ and } b = \begin{bmatrix} R_{11} \\ R_{12} \\ R_{13} \end{bmatrix} \tag{4.37}$$

Z vector is obtained from b

$$Z = \frac{b}{\|b\|} \tag{4.38}$$

The outputs are the optimum weigths of the array elements. It is given as

$$\hat{w}_{opt} = R^{-1}S_d\left[S_d^H R^{-1} S_d\right]^{-1} r \quad . \tag{4.39}$$

The main constraint of the optimum weights is to minimize the mean output power.

R is the correlation matrix, as mentioned above.

$$R = E\left\{ X(t) X(t)^H \right\} \quad , \tag{4.40}$$

S_d is the steering matrix of the desired signals. Assuming that the number of the desired signals is V, the desired steering matrix S_d is

$$S_d = \left[S_d(\theta_1) \;\; S_d(\theta_2) \;\; \cdots \;\; S_d(\theta_V) \right] \quad , \tag{4.41}$$

where $S_d(\theta_i)$ is

$$S_d(\theta_i) = \left[1 \;\; e^{-jk_i} \;\; e^{-j2k_i} \;\; \cdots \;\; e^{-j(M-1)k_i} \right] \quad . \tag{4.42}$$

The Radial Basis Function Neural Network (RBFNN) is used for the training phase. The performance phase is accomplished to obtain the optimum weights for unseen correlation matrix inputs after the training phase.

4.3.2 Algorithm

The network is composed of $\left[Z, w_{opt} \right]$, input-output pairs.

The angular range is divided into twelve sectors, as $\left[0, \; 30, \; \cdots, 330 \right]$. According to Figure 4-8 and Figure 4-9, each angular sector is affected by the radiation pattern of three antenna elements. For Sector i, $(i-1)^{th}$, i^{th} and $(i+1)^{th}$ elements are taken in consideration.

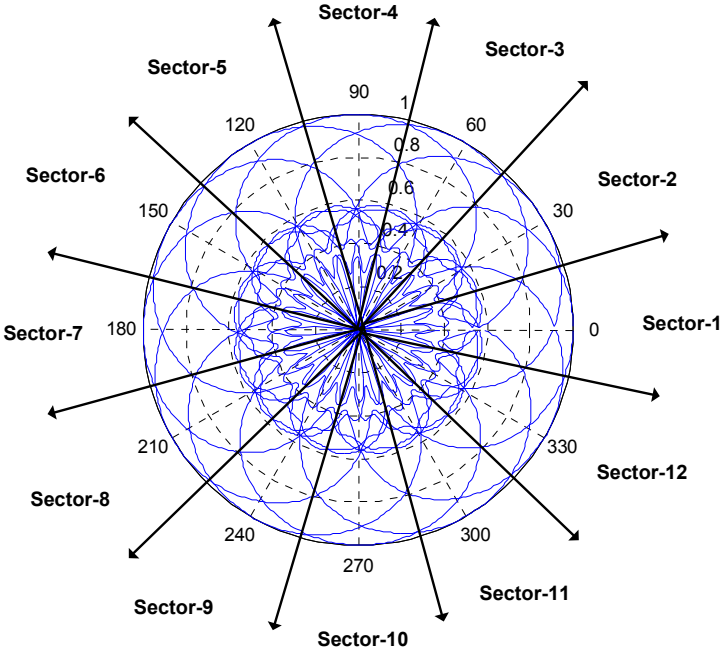

Figure 4-9 Sectors of CMPA

The RBFNN network is used for training phase. The algorithm scheme is given in Figure 4-10.

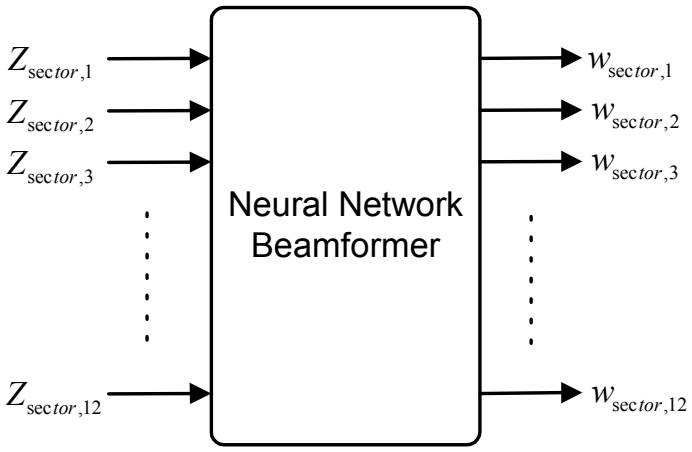

Figure 4-10 The Neural Network Beamformer Architecture

4.3.2.1 Training Phase

The training phase includes the input and output pairs which are calculated in equations (4.37), (4.38) and (4.39) respectively.

The training phase has less number of input and output pairs than linear array application for multiple signal presence simultaneously.

For single target case, the number of pairs is $30 \times 12 = 360$. For two target presence, the number of input-output pairs is $435 \times 12 = 5220$, which is 32580 in linear array case.

Z vectors are calculated from the correlation matrix, as given in equations (4.37) and (4.38). The optimum weigths are derived from the equation (4.39).

The number of array elements for beamforming in each sector is three and the number of desired signals is K and the angular range of interest is $\left[0^0, 360^0\right]$.

It is assumed that the number of possibities of the inputs is No_Pos. There are No_Pos number of correlation matrices of size 3×3, No_Pos number of Z vectors of size $\left[6 \times 1\right]$ and No_Pos number of weight vectors of size 3×1. The Z vector is calculated for all training sets of number No_Pos. For all training sets, No_Pos number of optimum weights are derived.

The optimum weight formulation contains the steering matrix of desired signals. Assuming that the total number of desired and/or interfering signals are No_des_int. The size of S_d is $\left[360, No\_des\_\text{int}\right]$. The term r in equation (4.39) is characteristic parameter that determines if the signals are interfering or desired signals. If there are two desired signals, $r = \left[1 \quad 1\right]$. If there is a single desired and a single interfering signal, $r = \left[1 \quad 0\right]$. Z vectors are given as the input of the RBFNN network. The output of the network is w_{opt}.

The input and output pairs are obtained, $\left(Z_{6\times No\_Pos}, w_{opt,3\times No\_Pos}\right)$. This training set makes the NN free from the DOA knowledge of the incoming signal.

In MATLAB, "**newrb"** command is used to train the NN for the given input and output pairs. The trained network is saved in the memory as data file. This data file is used for beamforming in the performance phase.

4.3.2.2 Performance Phase

In the performance phase, it is assumed that there comes a signal from an unknown direction. According to the incoming signal, the correlation matrix R, b vector and the Z vector are calculated from the induced signals by cylindrical microstrip patch antennas in each sector group.

The calculated Z vector is presented as the input of the trained network which is saved at the end of the training phase.

The NN tries to make an optimization for the input Z vector and outputs w_{opt} for each sector. The performance phase of NN is implemented by "**sim**" command in MATLAB.

The derived w_{opt} is then used to obtain the array pattern that is formed for desired and interfering signals by the equation (4.14).

CHAPTER 5

SIMULATIONS

In this chapter, some simulations over linear and cylindrical array applications are implemented to demonstrate the performance of the NN beamformer.

The simulations are divided into two main groups: linear antenna array and cylindrical antenna array applications.

The performances of the beamformers are examined according to the number of array antenna elements, the angular seperation between targets for multiple target simulations and SNR values.

The first implementation is composed of linear array simulations. The performance is analyzed by changing the number of isotropic equispaced antenna elements, the SNR value and the angular seperation between antenna elements. The distance between the antenna elements is $\lambda/2$. The angular range of interest is $\left[-90^0, 90^0\right]$. The operation frequency is 1.8 GHz.

The second implementation consists twelve element Cylindrical Microstrip Patch Antenna (CMPA). This array has a full coverage of 360^0. The geometry and the array pattern are given in Figure 4-7 and Figure 4-8. The performance is examined according to the SNR value and angular seperation between targets. The angular range of interest is $\left[0^0, 360^0\right]$. The operating frequency is 1.8 GHz.

The simulation results of linear array are given in Section 5.1 and the simulation results of cylindrical array are given in Section 5.2.

5.1 Linear Array Resuls

Linear array simulations are implemented by assuming an array with equispaced antenna elements, as shown in Figure 5-1.

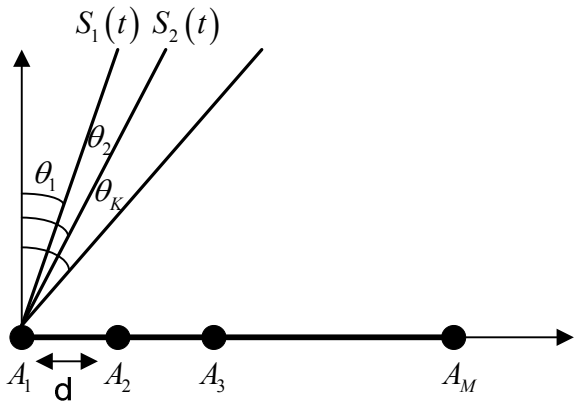

Figure 5-1 Linear Array Geometry

Linear array simulation results are divided into two groups: single target and two targets and interference cases.

In Section 5.1.1, simulation results are applied by the assumption of single target presence simultaneously in the angular range of interest $\left[-90^0, 90^0\right]$. The simulation results are analyzed for different number of antenna elements and for different SNR values.

In Section 5.1.2, simulation performances are given for the case of two target presence simultaneously.

5.1.1 Single Target

In the following simulations, the Neural Network is trained for all possibilities of single target presence in $\left[-90^0, 90^0\right]$. The criterions of performance analysis are number of array antenna elements and SNR value.

In chapter 5.1.1.1, the number of antenna elements will be changed. In chapter 5.1.1.2, the SNR effect over the beamformer will be shown.

5.1.1.1 Effect Of The Number Of Antenna Elements

The number of antenna elements is an important criteria for the performance of beamformer. As it will be seen in the following figures, increasing the number of antenna elements results a narrower array pattern for the signal coming from the desired direction.

The simulations are implemented for a single signal coming from -30^0. The number of antenna elements (an) are chosen as 3, 5, 10 and 15. The SNR value is $30\ dB$.

The simulation results are given in Figure 5-2-Figure 5-5.

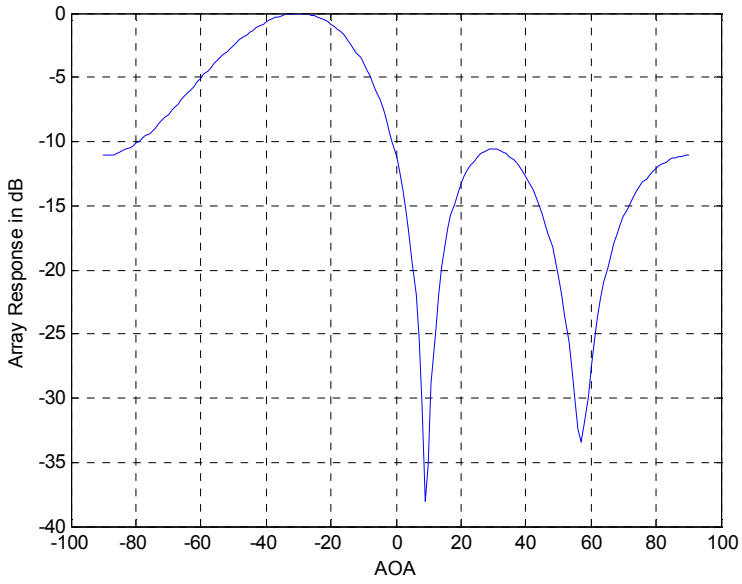

Figure 5-2 Array Pattern for a signal coming from -30^{o} with $an = 3$ and $SNR = 30\ dB$

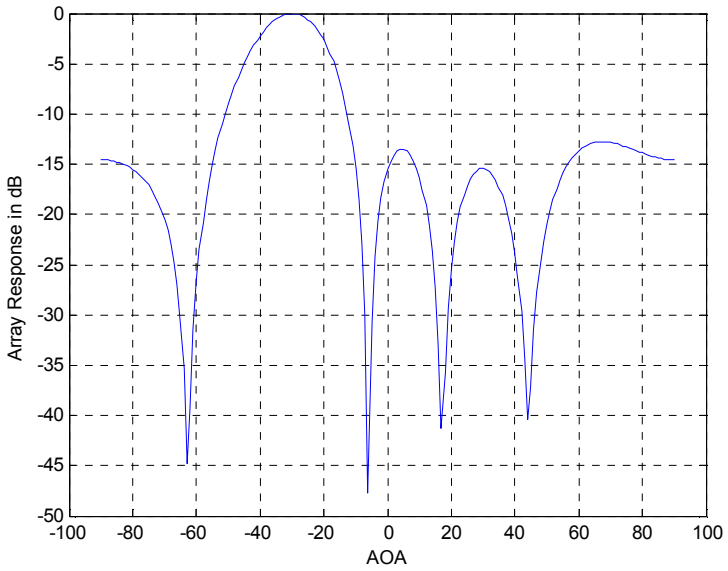

Figure 5-3 Array Pattern for a signal coming from -30^{o} with $an = 5$ and $SNR = 30\ dB$

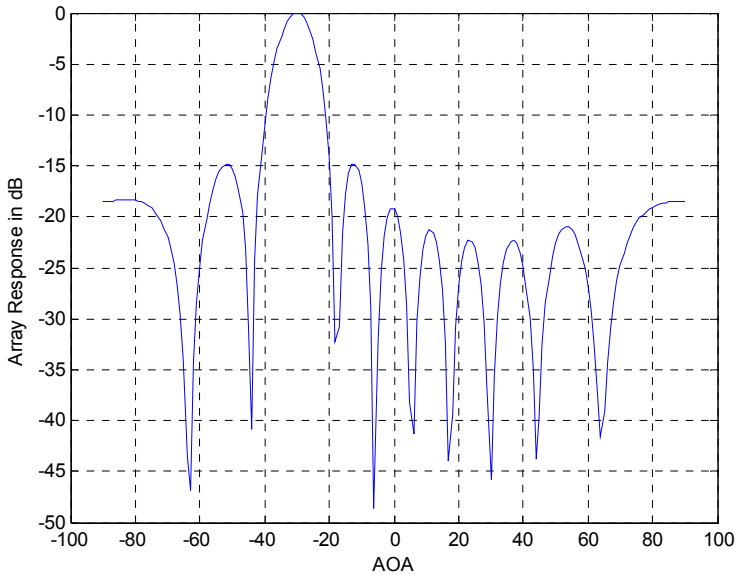

Figure 5-4 Array Pattern for a signal coming from -30^{o} with $an = 10$ and $SNR = 30\ dB$

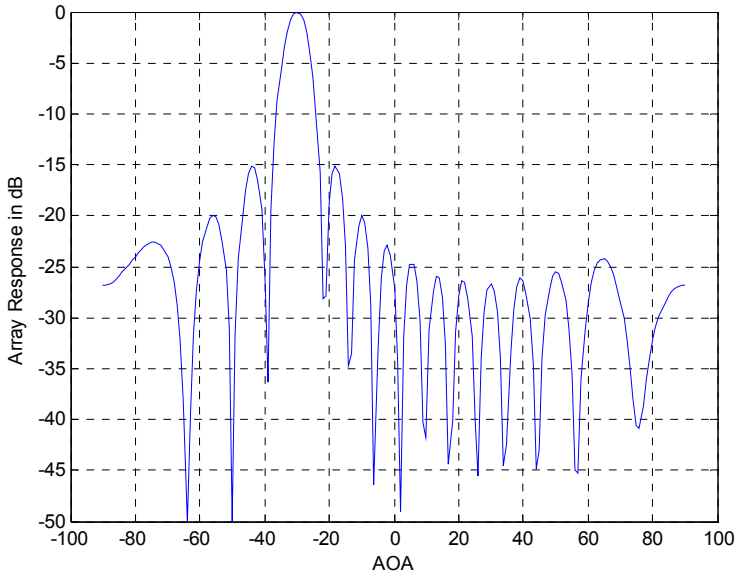

Figure 5-5 Array Pattern for a signal coming from -30^{o} with $an = 15$ and $SNR = 30\ dB$

As it is expected from the antenna array theory, increasing the number of antenna elements, the beamwidth of the antenna array becomes narrower, and the sidelobe levels decreases. These results improve the perfomance of the beamformer. Signals coming from the sidelobes are suppressed 15 dB in Figure 5-5.

A summary of these results are given in Table 5-1.

Number of antenna Elements	**Sidelobe Level**	**3-dB Beamwidth**
3	$-1\ dB$	40^o
5	$-13.2\ dB$	23.5^o
10	$-14.5\ dB$	11.5^o
15	$-15\ dB$	8^o

Table 5-1 Summary of Linear Array Application for different number of antenna elements

3-dB beamwidth of the array pattern decreases by increasing the number of antenna elements. As it is mentioned above, the sidelobe levels are suppressed more for higher numbers of antenna elements.

In Figure 5-6, it is seen that the array pattern is narrower for higher number of antenna elements. The beamwidth decreases to 11.5^o for $an = 10$.

In Figure 5-7, the sidelobe levels are figured for different numbers of antenna elements. The sidelobe level goes to $-14.5\ dB$ for $an = 10$.

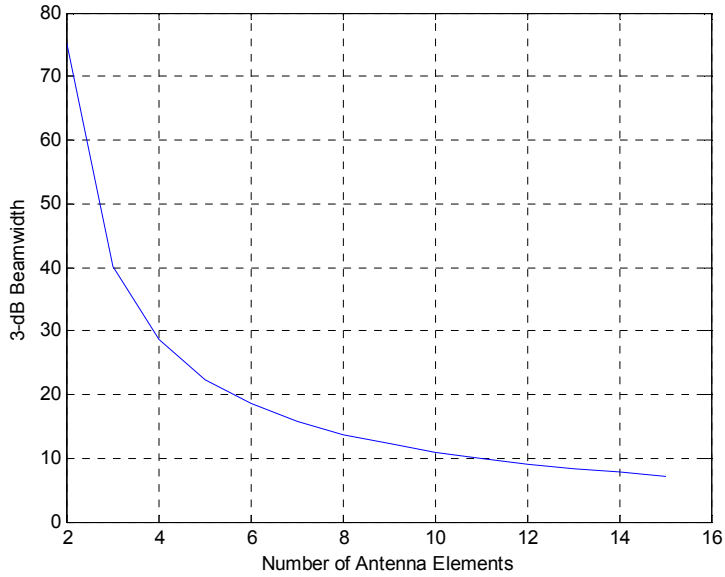

Figure 5-6 3-dB Beamwith change for Linear Array Application by changing the number of antenna elements

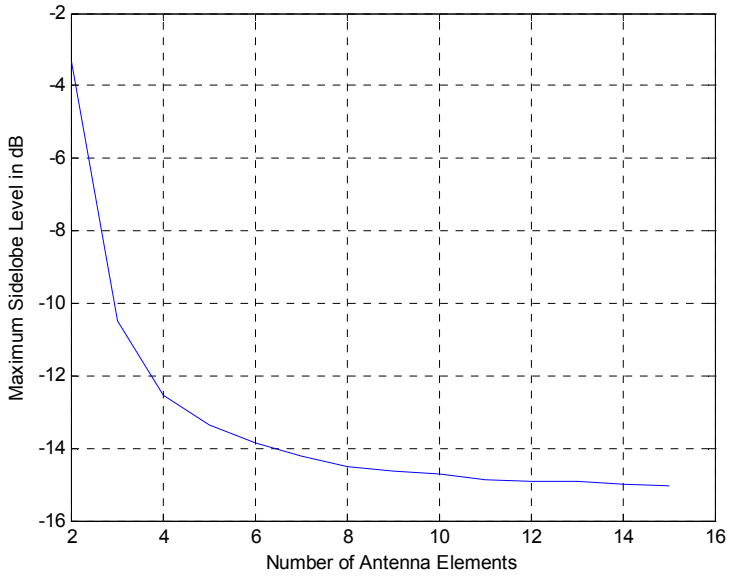

Figure 5-7 Maximum Sidelobe Lavel change for Linear Array Application by changing the number of antenna elements

5.1.1.2 SNR Effect

The SNR value is another important criteria for the performance of beamformer. The array has a good performence begining with $SNR = 10\ dB$.

The simulations are implemented for a single signal coming from 20^0. The number of antenna elements is 10.

The performance is analyzed with changing the SNR value by following values: $5\ dB$, $10\ dB$, $15\ dB$, $20\ dB$ and $30\ dB$. It is assumed that the noise is white Gaussian noise with zero-mean. The results are given in Figure 5-8-Figure 5-12.

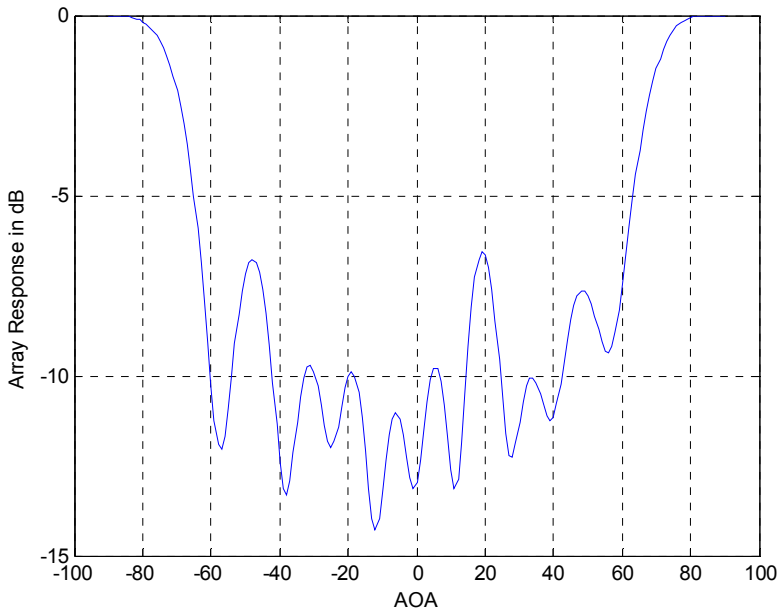

Figure 5-8 Array Pattern for a signal coming from 20^o with $an = 10$ and $SNR = 5\ dB$

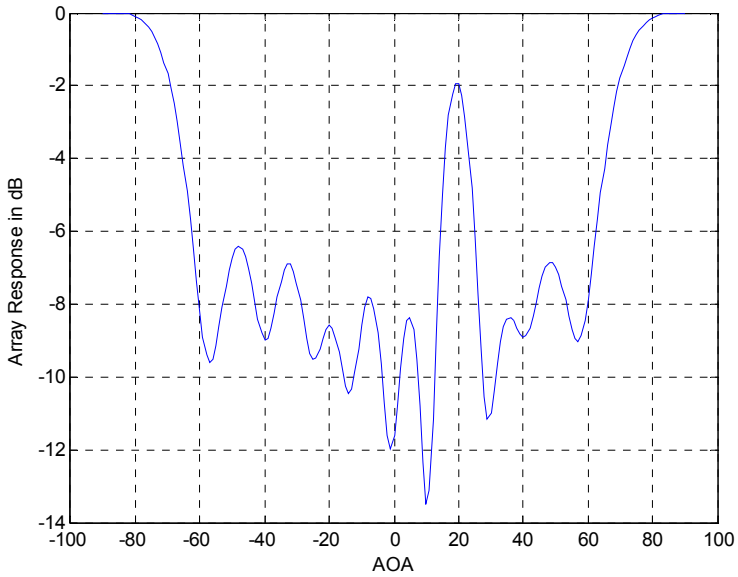

Figure 5-9 Array Pattern for a signal coming from 20^{o} with $an = 10$ and $SNR = 10\ dB$

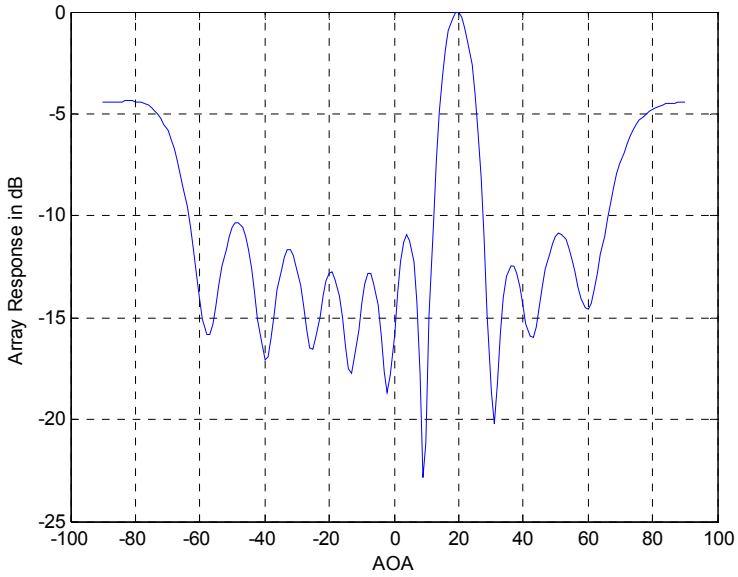

Figure 5-10 Array Pattern for a signal coming from 20^{o} with $an = 10$ and $SNR = 15\ dB$

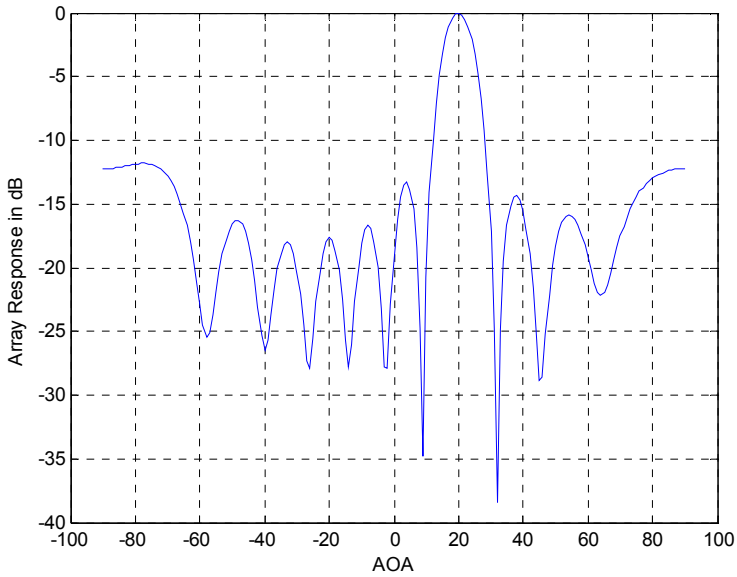

Figure 5-11 Array Pattern for a signal coming from 20^o with $an = 10$ and $SNR = 20\ dB$

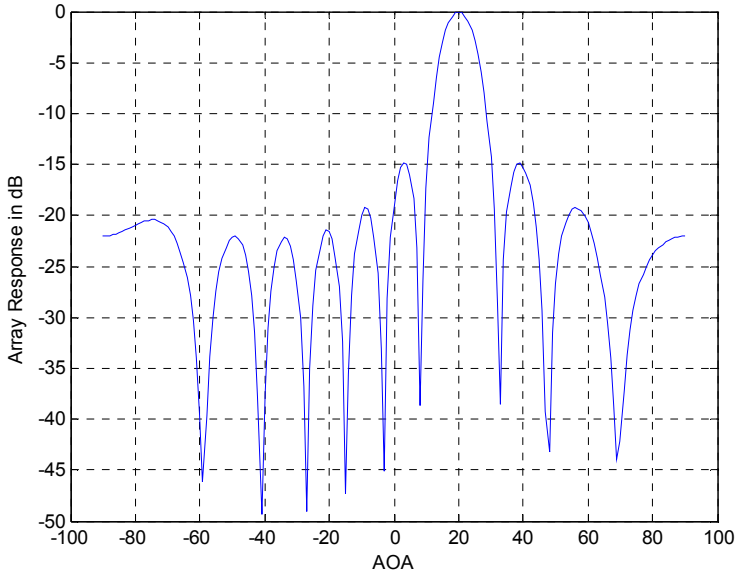

Figure 5-12 Array Pattern for a signal coming from 20^o with $an = 10$ and $SNR = 30\ dB$

It is observed that the array has higher sidelobe suppression values and better beam shapes for higher SNR values to direct the pattern in the direction of the desired signal. A summary of these results are given in Table 5-2.

SNR value $(an = 10)$	**RMS error**	**Sidelobe Level**
10 *dB*	0.50	0 *dB*
15 *dB*	0.25	-4.6 *dB*
20 *dB*	0.1	-12 *dB*
30 *dB*	0	-20.5 *dB*

Table 5-2 Summary of Linear Array Application for different SNR values

In Figure 5-13, the performance of the NN beamformer is shown for five different noisy signals with 10 *dB* of SNR.

In Figure 5-14, the RMS error is shown for different SNR values. The RMS error is calculated according to the position of the maximum peak level of the array pattern. The error is defined as the error of the position of the maximum peak level of the array pattern according to the direction of the incoming signal.

In Figure 5-15, the sidelobe levels are seen for different SNR values. The sidelobe level goes to -20.5 *dB* for $SNR = 30$ *dB* .

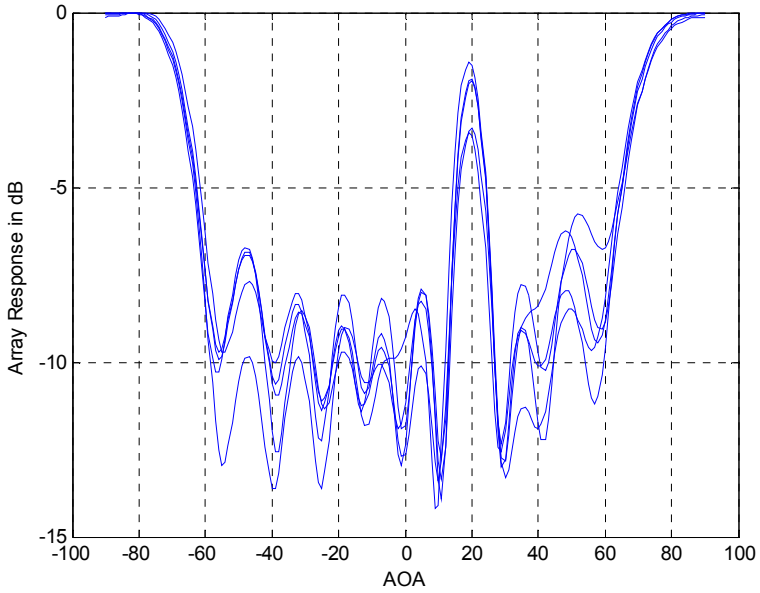

Figure 5-13 Array Pattern for 5 Different Noisy Signal Cases for Single Target

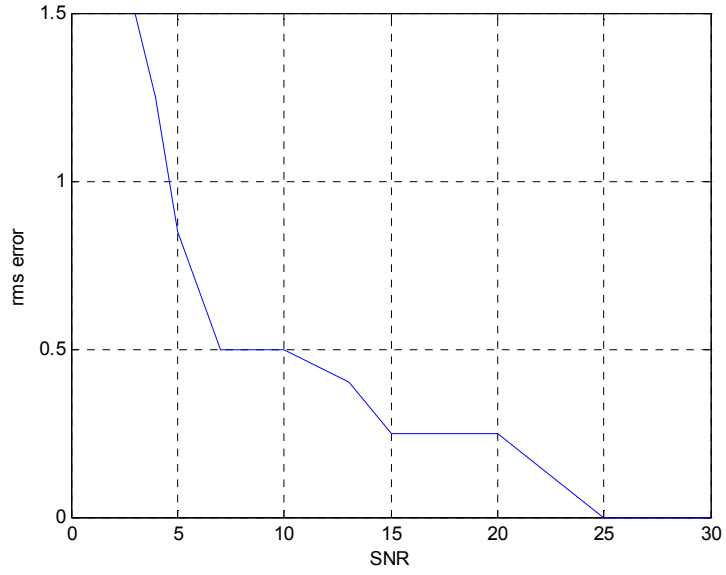

Figure 5-14 RMS error change for different SNR value

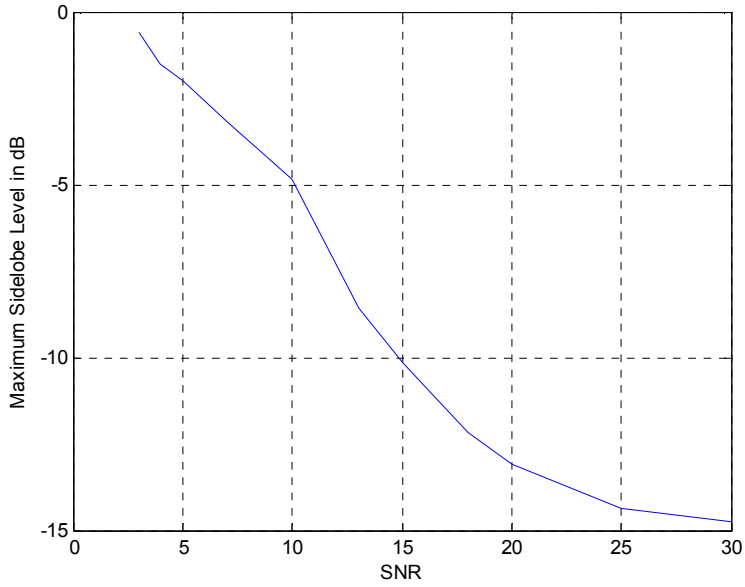

Figure 5-15 Sidelobe level change for different SNR values

5.1.2 Two Targets

In the following simulations, the Neural Network is trained for all possibilies of two targets present in 180^0 angular range of interest. The criterions of performance analysis are angular seperation between targets and SNR value. The number of antenna elements is taken as 10.

5.1.2.1 Angular Seperation Effect

The simulations for this chapter, are implemented for two signals present simultaneously. First signal comes from 0^0, while second signal comes from 1^o, 3^o, 5^o, 10^o, 15^o or 20^o respectively. The number of antenna elements is 10 and the SNR value is $30\,dB$.

The simulation results are shown in Figure 5-16 - Figure 5-21.

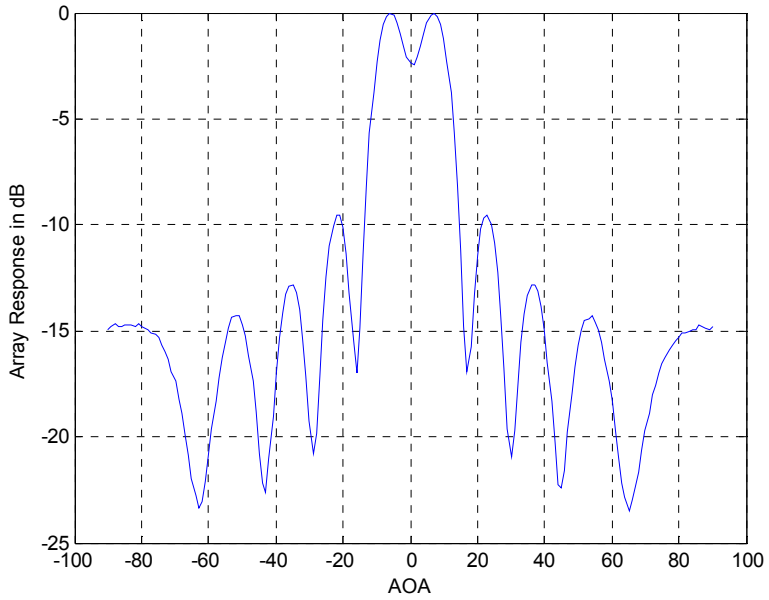

Figure 5-16 Signals coming from 0^o and 1^o with $an = 10$ and $SNR = 30\ dB$

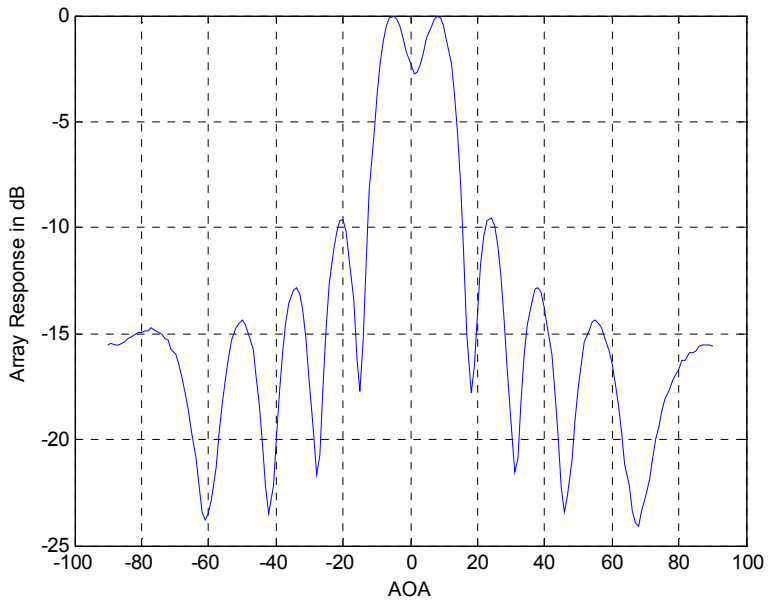

Figure 5-17 Signals coming from 0^o and 3^o with $an = 10$ and $SNR = 30\ dB$

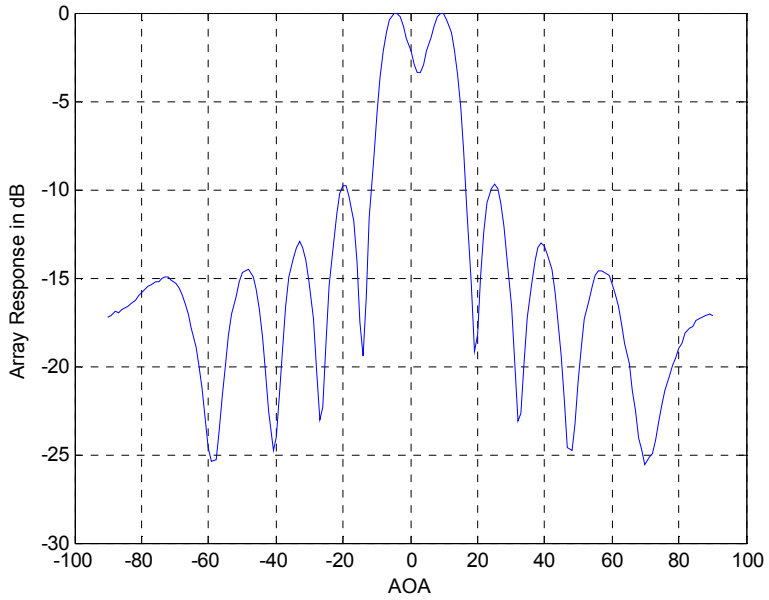

Figure 5-18 Signals coming from 0^o and 5^o with $an = 10$ and $SNR = 30\,dB$

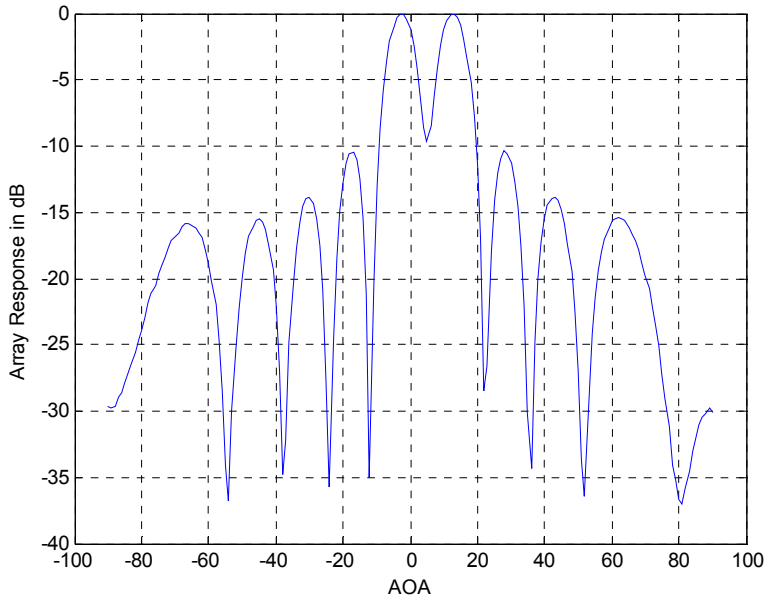

Figure 5-19 Signals coming from 0^o and 10^o with $an = 10$ and $SNR = 30\,dB$

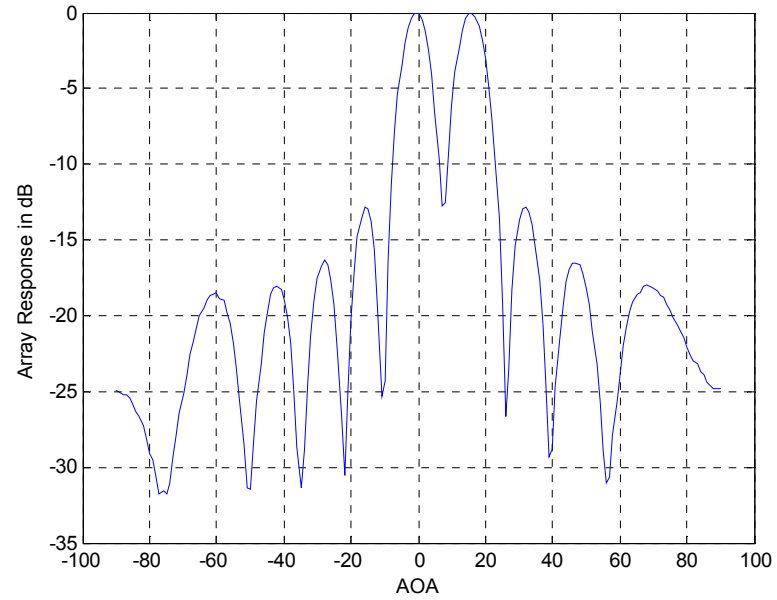

Figure 5-20 Signals coming from 0^o and 15^o with $an = 10$ and $SNR = 30\ dB$

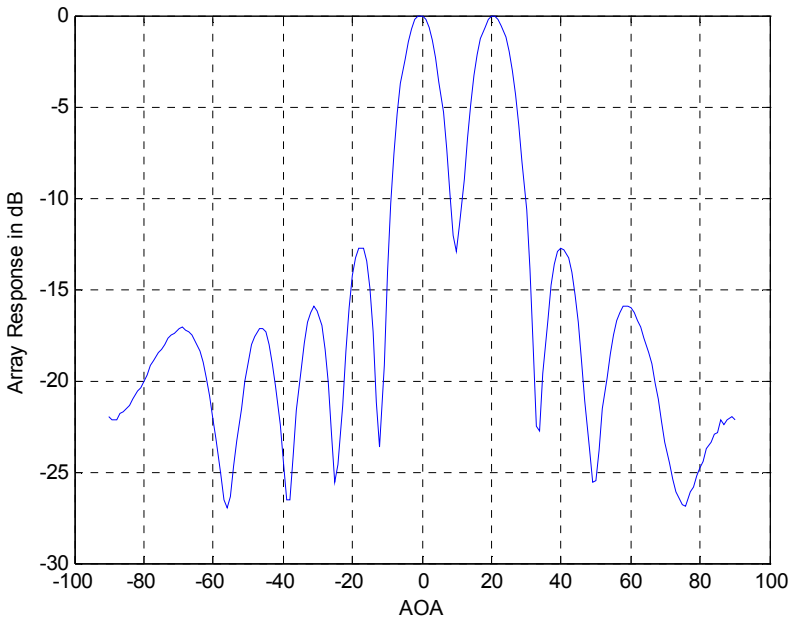

Figure 5-21 Signals coming from 0^o and 20^o with $an = 10$ and $SNR = 30\ dB$

The simulation results for 1^0, 3^0 and 5^0 angular seperations do not have peak levels of array response in the directions of desired signals. On the other hand, the desired signals directions are covered with 3-dB beamwidth of the array pattern. The simulation results for higher angular seperations have good performances.

The summary of simulation results are in Table 5-3.

Angular Seperation	**Sidelobe level**	**RMS error**
1^o	$-9.5\ dB$	6^o
3^o	$-9.7\ dB$	5^o
5^o	$-9.9\ dB$	3.5^o
10^o	$-10.5\ dB$	3^o
15^o	$-13.0\ dB$	1^o
20^o	$-13.5\ dB$	0.7^o

Table 5-3 Summary of Two Target Linear Array Application for different angular seperation values

5.1.2.2 SNR Effect

The simulations are implemented for two signals presence simultaneously for SNR criteria. Signals are coming from 10^0 and 25^0. The number of antenna elements is 10. The SNR values are taken as the following values: $5\ dB$, $10\ dB$, $15\ dB$, $20\ dB$ and $30\ dB$. The simulation results are presented in Figure 5-22 - Figure 5-26.

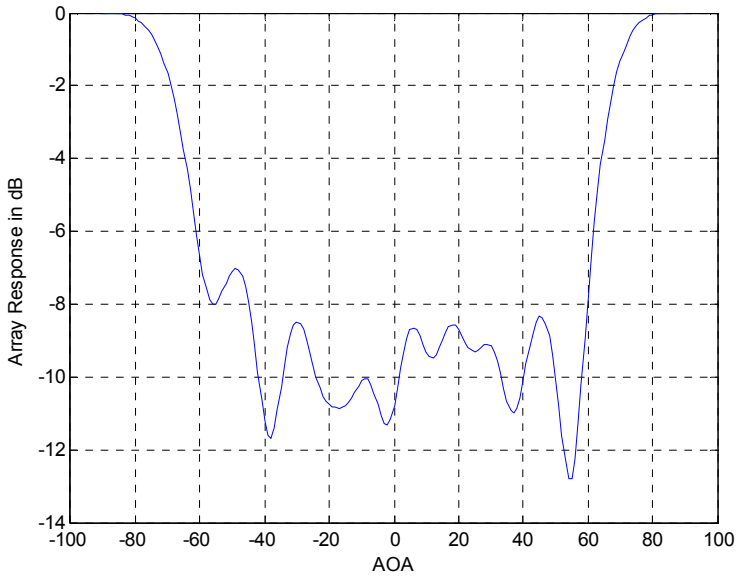

Figure 5-22 Array Response for two targets in 10^o and 25^o, $an = 10$ and $SNR = 5\ dB$

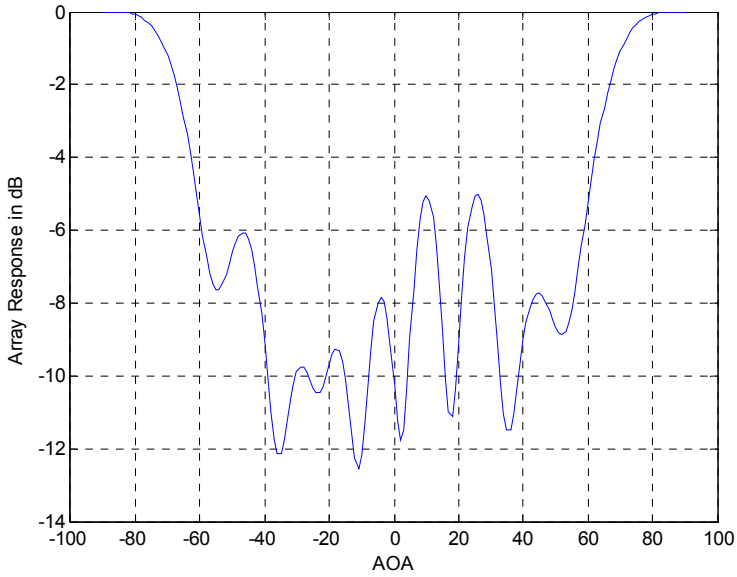

Figure 5-23 Array Response for two targets in 10^o and 25^o, $an = 10$ and $SNR = 10\ dB$

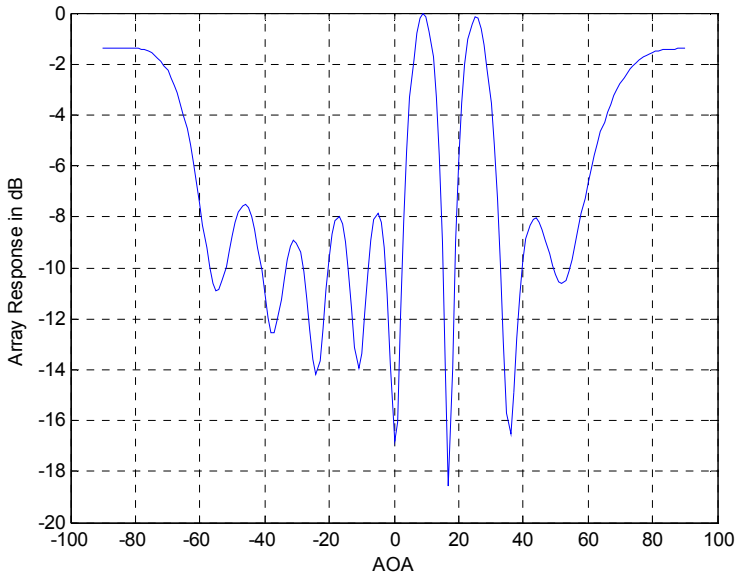

Figure 5-24 Array Response for two targets in 10^o and 25^o, $an = 10$ and $SNR = 15\ dB$

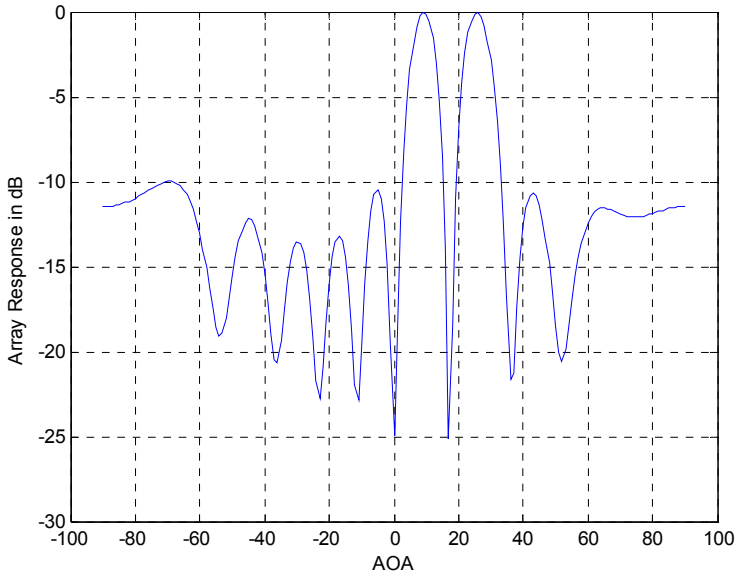

Figure 5-25 Array Response for two targets in 10^o and 25^o, $an = 10$ and $SNR = 20\ dB$

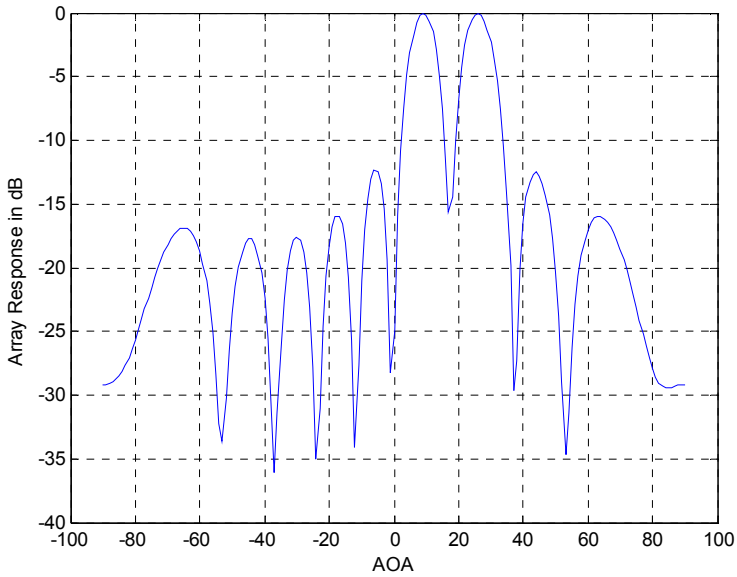

Figure 5-26 Array Response for two targets in 10^o and 25^o, $an = 10$ and $SNR = 30\ dB$

The beamformer has good performance beginning from $SNR = 10\ dB$. The sidelobe levels are decreasing by the increase in SNR value, as shown in Table 5-4.

SNR Level	**Sidelobe level**
10	0 *dB*
15	$-1.7\ dB$
20	10 *dB*
30	12.4 *dB*

Table 5-4 Summary of Two Target Linear Array Application for different SNR values

In Figure 5-27Figure 5-13, the performance of the NN beamformer is shown for five different noisy signals with 15 *dB* of SNR.

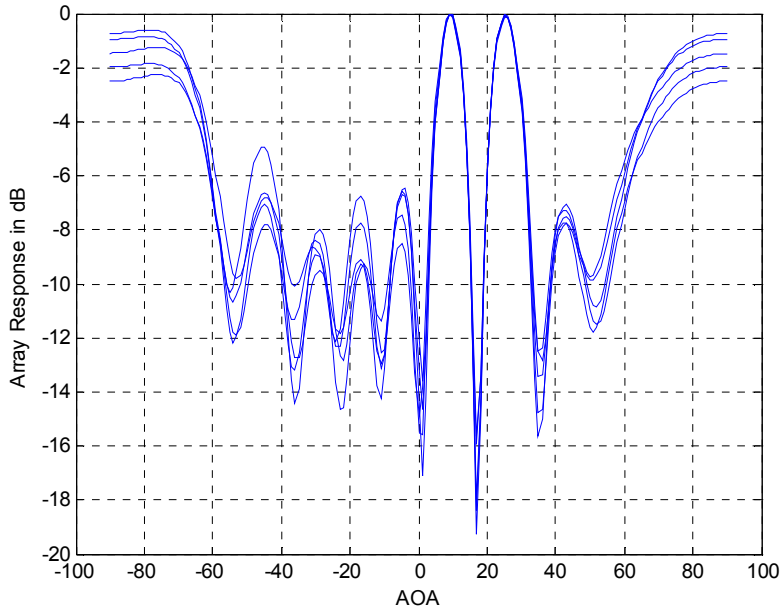

Figure 5-27 Array Pattern for 5 Different Noisy Signal Cases for Two Targets

5.2 Cylindrical Array Results

Cylindrical array is composed of twelve CMPA elements. The geometry of the array gives the benefit of having a full coverage of 360^0 as shown in Figure 5-28.

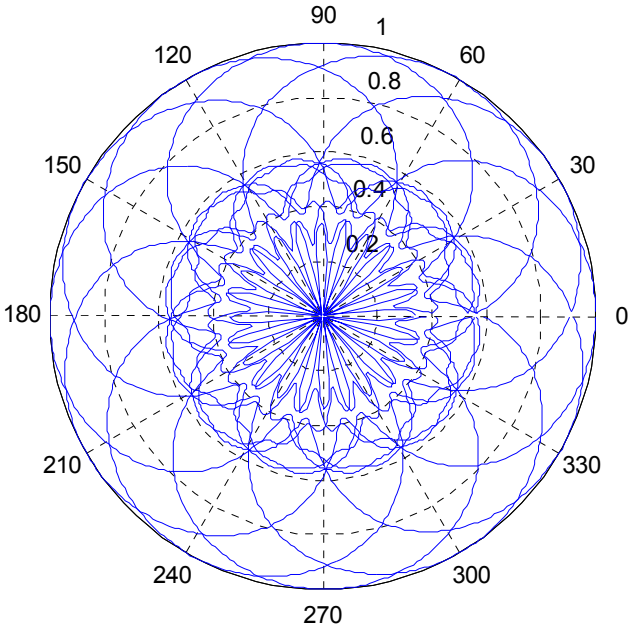

Figure 5-28 Cylindrical Array Pattern and Geometry

The main advantage of this array for NN is its less training time because that 360^0 is divided into sectors, 30^0 each.

The simulations are divided into two: single target and multiple targets. For multiple target simulations, the directions targets are assumed to be in the same sector and/or in different sectors of the antenna array.

First simulations are implemented for one target coming to one sector in Section 5.2.1. Second part simulations are for multiple target signals coming in one or more sectors, as mentioned in Section 5.2.2.

5.2.1 Single Target

The following simulations are over one target beamforming applications. The NN is trained for all possibilities of single target present in angular coverage of 30^0 in each sector. The number of all posibilities is only 30 for each sector. Twelve sectors are trained parallel to each other in one NN. This helps for a faster training.

In the performance phase, there is a signal coming from 140^0, 6^{th} sector in Figure 4-9, with an angular coverage of $\left[135^0, 165^0\right]$ is activated according to Figure 4-7.

SNR values are taken as, $10\ dB$, $15\ dB$, $20\ dB$ and $30\ dB$.

The simulation results are presented in Figure 5-29 - Figure 5-32.

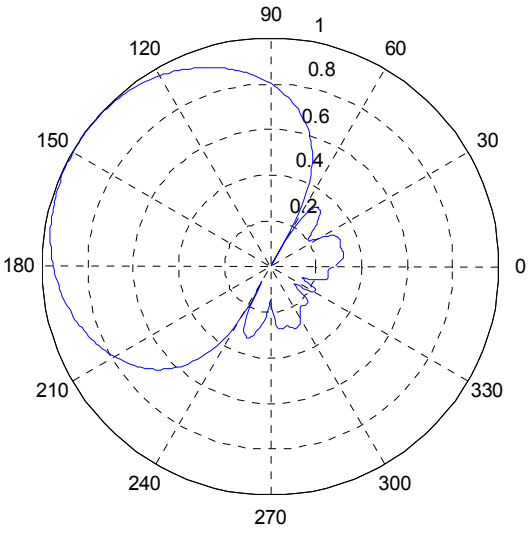

Figure 5-29 Array Pattern for signal coming from 140^o, $SNR = 10\ dB$

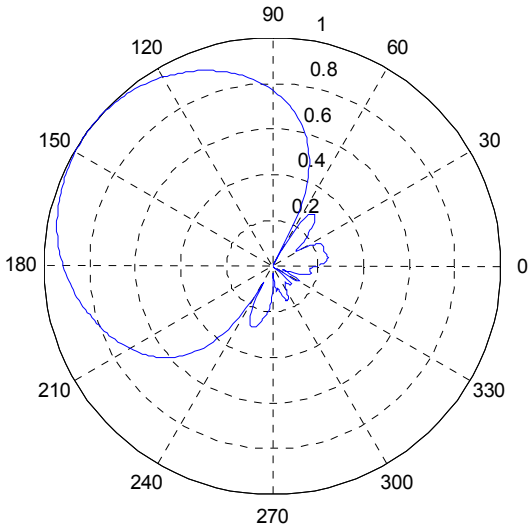

Figure 5-30 Array Pattern for signal coming from 140^{o}, $SNR = 15\ dB$

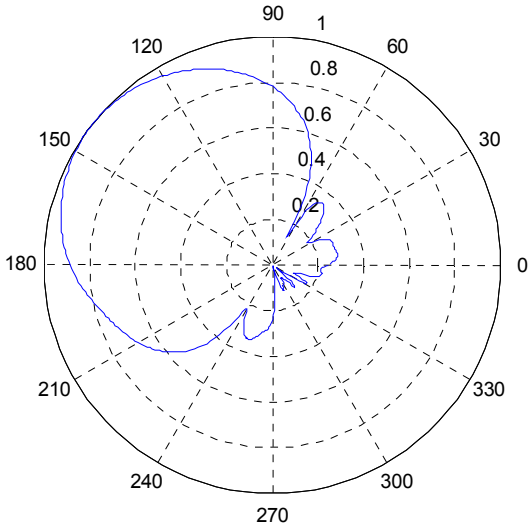

Figure 5-31 Array Pattern for signal coming from 140^{o}, $SNR = 20\ dB$

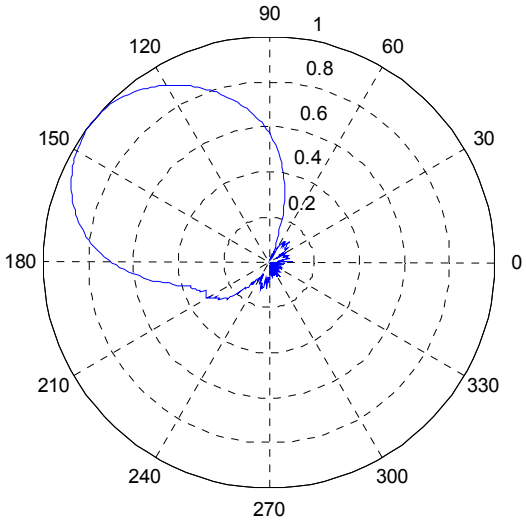

Figure 5-32 Array Pattern for signal coming from 140^{o}, $SNR = 30\ dB$

The simulation results shows that the beamformer perfomance is good beginnig from $SNR = 10\ dB$. The array has very low sidelobe levels in other sectors than in the sector of target. The pattern is formed with three antenna elements.

5.2.2 Multiple Targets

Multiple target simulations need more time for training, since there are more possibilites of multiple target presence in each sectors.

In the following simulations, it is assumed that there are more than a single target, with a $30\ dB$ SNR value. Three different type of simulations are presented here. In Section 5.2.2.1, there are two targets in one sector. The effect of angular seperation change in array pattern is taken into consideration in this section. In Section 5.2.2.2, there are multiple targets in different sectors and there are interferences which are nulled by NN beamformer.

5.2.2.1 Angular Seperation Effect

In this simulation, two targets are considered within the same sector. The NN is trained for all possibilities of being two targets simultaneously in a sector. By changing the angular seperation between the two targets, the performance of the NN beamformer is analyzed.

There exist two targets in 11^{th} sector, as shown in Figure 4-9, which has an angular coverage of $\left[285^0, 315^0\right]$. The simulation is presented to show the performance of the Neural Network beamformer under different angular seperation values between two targets.

The first target is assumed to be at 285^0 while the second target is at 290^0, 295^0, 300^0, 305^0 and 315^0 respectively.

The simulations are presented in Figure 5-33 - Figure 5-37.

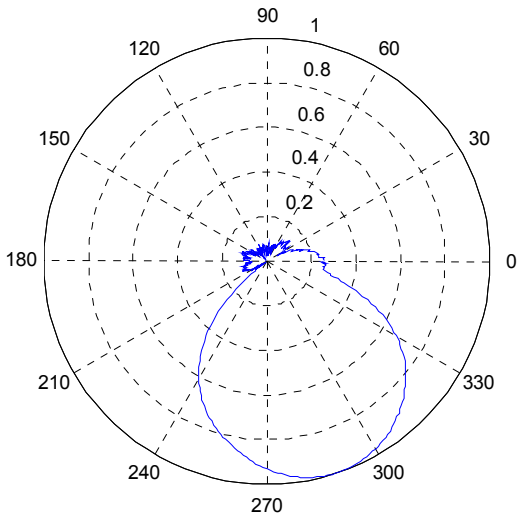

Figure 5-33 Array Response for targets in 285^0 and 290^0, $SNR = 30\ dB$

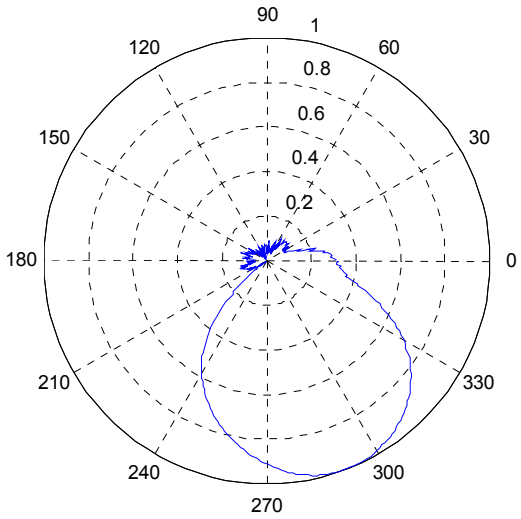

Figure 5-34 Array Response for targets in 285^0 and 295^0, $SNR = 30\ dB$

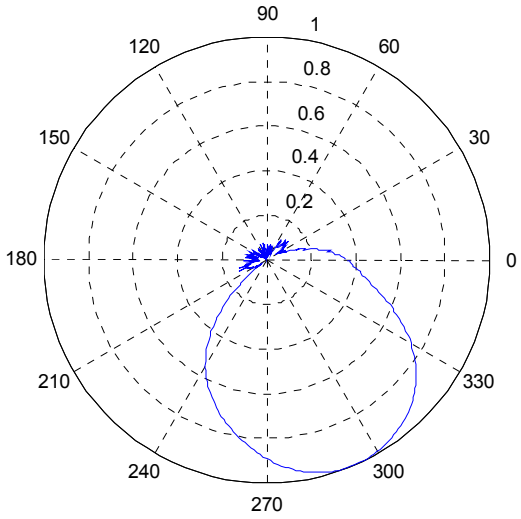

Figure 5-35 Array Response for targets in 285^0 and 300^0, $SNR = 30\ dB$

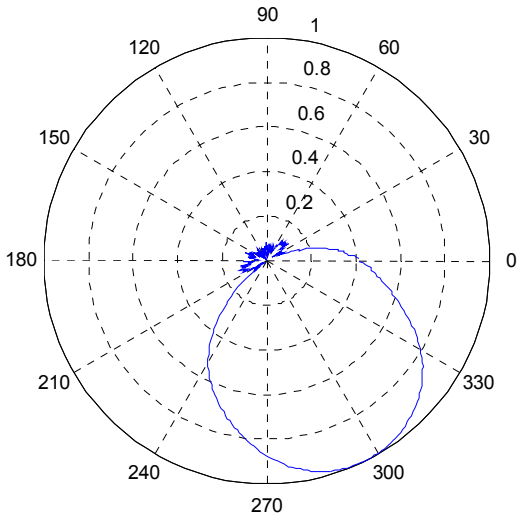

Figure 5-36 Array Response for targets in 285^0 and 305^0, $SNR = 30\ dB$

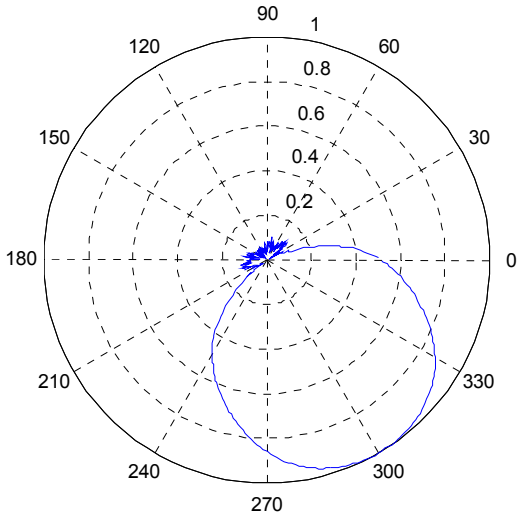

Figure 5-37 Array Response for targets in 285^0 and 315^0, $SNR = 30\ dB$

Increasing the angular seperation causes a wider pattern in these simulations. Sidelobe levels are very low in the other sectors.

5.2.2.2 Targets in Different Sectors

In this part, two simulations are implemented to show the performance of the NN beamformer for target signals coming from different sectors.

First simulation is composed of four targets in four different sectors. Three antenna elements in each sector forms the beam to the direction of incoming signal. SNR is $30\ dB$.

Second simulation consists two targets and a single interfering signal in each of two different sectors. The NN is trained for all possibilities of being two targets in each sector of 30^0, with SNR of $30\ dB$.

In the first simulation, the targets are coming from 25^0, 125^0, 210^0 and 300^0 to the 2^{nd}, 5^{th}, 8^{th} and 11^{th} sectors respectively, as shown in Figure 5-38.

The array pattern is directed to each four targets simultaneouly by directing the peaks of each pattern to the directions of incoming signals.

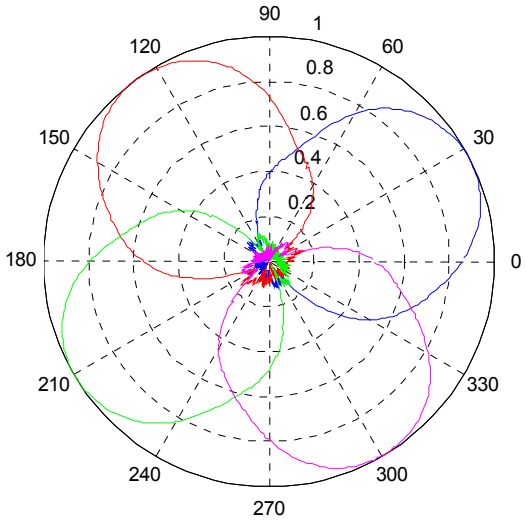

Figure 5-38 Array Pattern for targets in 25^0, 125^0, 210^0 and 300^0, $SNR = 30\ dB$

In the second simulation, 1^{st}, and 7^{th} sectors are activated. This simulation shows the performance of NN beamformer for interference presence. The NN is trained for all possibilities of being two targets and an interfering signal in each sector.

Signals coming to the 1^{st} sector are from 345^0 and 15^0. The interfering signal is coming from 0^0.

Signals coming to the 7^{th} sector are from 170^0 and 190^0. The interfering signal is coming from 180^0, shown in Figure 5-39.

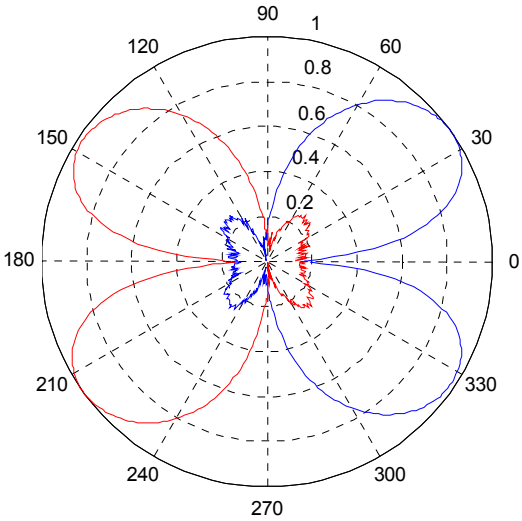

Figure 5-39 Array Pattern for targets in 345^0, 15^0, 170^0 and 190^0, and for interfering signals in 0^0 and 180^0, $SNR = 30\ dB$

It is seen that the beamformer puts nulls successfully in the directions of interferences with a trade of having the peak values of each pattern not directly pointed in the directions of targets.

The pattern is not exactly directed to the directions of targets, but the targets are in 3-dB beamwidth coverage of each four beams.

CHAPTER 6

CONCLUSION

In this thesis, a Neural Network (NN) algorithm for beamforming problem is proposed. Two array applications are implemented to show the performance of the proposed beamformer.

The NN's are trained for each problem to shape the patterns of the arrays to the directions of targets and to put nulls in the directions of interfering signals. The RBFNN's are used for the training phase of the NN . The effectiveness of this NN helps for fast convergence in training phase. The performance phases give good responses for each application.

Two different array applications are studied in this work. First application is an implementation of linear array as shown in Figure 5-1. An angular coverage of 180^0 degrees is handled with this array. The second application is composed of a cylindrical array with twelve antenna elements as shown in Figure 4-8.

The linear antenna array is composed of omnidirectional antennas with an equal distance. An angular range of 180^0 is trained for forming the beams. The performance of the linear array beamformer is examined by changing the number of antenna elements of the array and by changing the angular seperation between the targets for different SNR values.

The number of the antenna elements and the angular seperation between the targets influence the performance of the beamforming algorithm. The algorithm works better for larger number of antenna elements and for a sufficient angular seperation. It is seen that the angular seperation between multiple targets is an important criteria for beamformer performance.

The second application is studied on cylindrical array, composed of twelve microstrip patch antenna (MPA) elements. The total angular range of 360^0 is covered by the MPA elements. The total angular range is divided into twelve sectors. The directive MPA elements help for faster training times of NN for beamforming. Three antenna elements used for shaping the beams in each sector. Since the total angular range of 360^0 is considered in terms of twelve sectors of 30^0 each, the proposed algorithm performs faster in the cylindrical array application. It is seen in the cylindrical array application that the beamformer works well for sufficient angular seperation between the targets and interference.

The simulations are performed for simultaneous target presence in the angular range of interest. The NN beamformer is examined under this consideration. The results shows that the beamformer has satisfactory performance for two different array applications.

The main need for good perfomance of this beamformer algorithm is to choose the exact input and output pairs for the training phase. The performance phase has fast responses for each incoming signal.

The proposed beamformer can easily be implemented in real world applications with its flexible algorithm and fast training performance. Since the beamformer performance phase has a sufficient speed for forming the beam according to its training set, this beamformer can be used for tracking applications.

The advantage of the NN beamforming algorithm over the adaptive beamforming algorithms is that the NN beamformer has fast convergence speed. NN beamformer

calculates the weights of the antenna array elements offline for the incoming signal direction possibilities. The calculated weights and the correlation matrices form a lookup table for the NN. The lookup table is used for a new incoming signal to give the new weights, instead of calculating the weights every time of a new signal arrival. This approach provides faster responses.

As a proposal for future works, modulated signals and different types of jamming signals can be implemented as the inputs of the NN beamformer to analyze the performance of the beamformer in the presence of modulation and jamming signals.

REFERENCES

[1] Lal C. Godara, "Application of antenna arrays to mobile communications, part II: beam-forming and direction-of-arrival considerations", *Proceedings of the IEEE*, vol. 85, No. 8, pp. 1195-1245, August 1997.

[2] Barry D. Van Veen, Kevin M. Buckley, "Beamforming: A Versatile Approach to Spatial Filtering", IEEE ASSP Magazine, 4-24, April 1998.

[3] A. H. Zooghby, C. G. Christodoulou, and M. Georgiopoulos, "Neural Network-Based Adaptive Beamforming for One- and Two-Dimensional Antenna Arrays", IEEE Transactions on Antennas and Propagation, vol. 46, No.12, pp. 1891-1893, December 1998.

[4] Çaylar, S., K. Leblebicioğlu, and G. Dural, "A new neural network approach to the target tracking problem with smart structures", Radio Sci., 41, RS5004, doi:10.1029/2005RS003301., October 2006.

[5]Çaylar, S., ***Target Tracking with Smart Antennas***, Ankara: Middle East Technical University, Ph. D. Thesis, September 2007.

[6] Koç, A.T., ***Direction Finding with a Uniform Circular Array via Single Snapshot Processing***, Ankara: Middle East Technical University, Ph. D. Thesis, January 1996.

[7] 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 Processing, vol. 82, pp. 547-561, 2002.

[8] Yupeng Chen; Chaohuan Hou; "High resolution adaptive bearing estimation using a complex-weighted neural network", 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing Proc., vol. 2, pp. 317-320, March 1992.

[9] H. L. Southall, J. A. Simmers, and T. H. O'Donnell, "Direction finding in phased arrays with a neural network beamformer," *IEEE Trans. Antennas Propagat.*, vol. 43, pp. 1369-1374, Dec. 1995.

[10] 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.

[11] Mozingo and Miler, *Introduction to Adaptive Arrays*. New York: Wiley, 1980.

[12] H. L. Southall, "Array Antenna Applications for Neural Beamforming," *IEEE*, pp. 40-44, 1997.

[13] Niemand, P., ***Null Synthesis and Implementation of Cylindrical Microstrip Patch Arrays***, Pretoria:University of Pretoria, Ph.D. Thesis, Nov. 2004.

APPENDIX A

DOA WITH NEURAL NETWORK

This section summarizes the DOA with NN. Formulations given are presented in [5]. Thsi study is implemented to understand the application of Neural Networks before implementing the NN beamforming application.

The NN has three stage in the DOA algorithm and the angular range of interest is divided into subsectors. In each subsector, a NN is trained and performed.

The architecture of the DOA NN algorithm is summarized in Figure A-1.

First stage is the Detection stage. In this part, according to the direction of incoming signal, the appropriate subsector is activated. The input of the NN is the correlation matrix of incoming signal,the output is 1 or 0.

The second stage is the Filtering stage. In this part, the NN is filtered and only the activated NN's are trained. This approach presents a faster training time in place of training the whole angular range of interest. The inputs are the same with the first stage, the outputs are the correlation matrices of the incoming signals to the activated sectors.

The third stage is DOA Estimation stage. This part decides the DOA of the incoming signal. The inputs of this sector are the outputs of the second stage. The outputs are the DOA informations of the incoming signals.

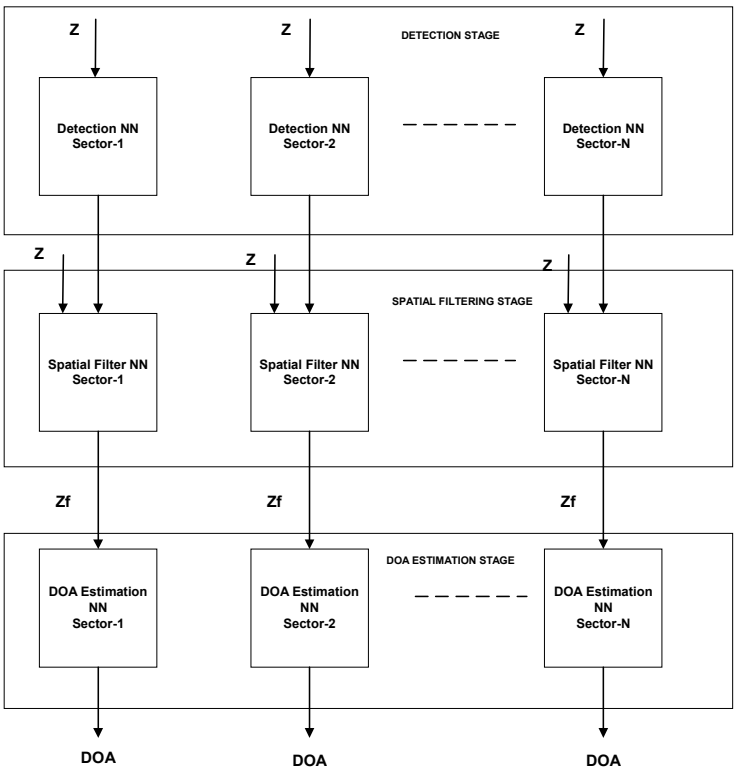

Figure A-1 DOA Neural Network Architecture

In the performance stage, the corelation matrix of the incoming signal is presented to the input of Detection stage, and the DOA information is observed at the output of the DOA Estimation stage.

The simlation is implemented under $SNR = 30\,dB$. The angular range is 30 degrees. This range is divided into three subsectors, each with 10^0.

In the first simulation, the incoming signal is assumed to come from 2^0 to the first subsector. The DOA result is given Figure A-2.

In the second simulation, the signal is assumed to come from 25^0, to the third subsector. The DOA result is given in Figure A-3.

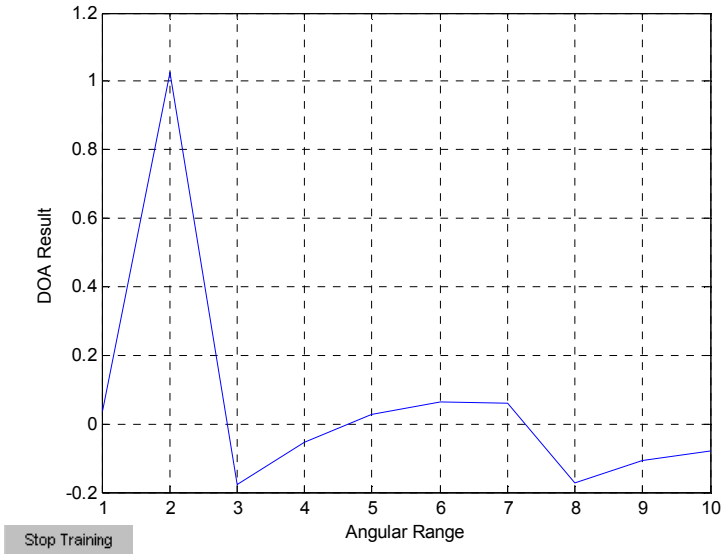

Figure A-2 DOA Result of One Target in 2^0

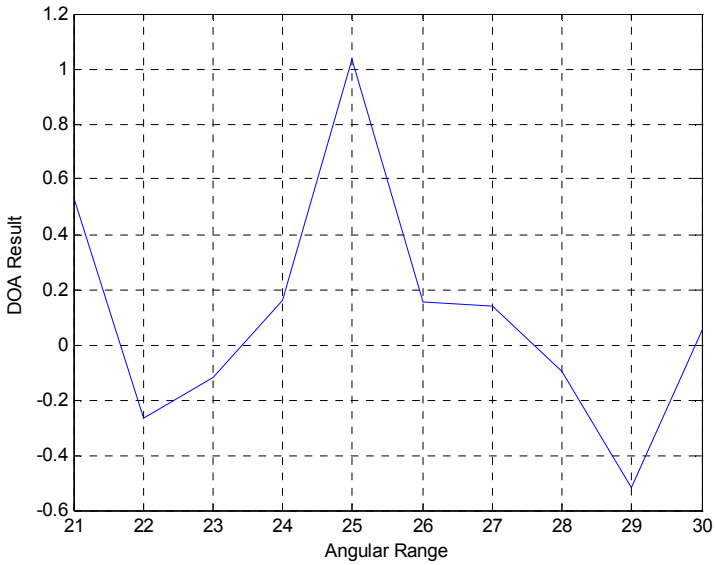

Figure A-3 DOA Result of One Target in 25^o