



ADDIS ABABA UNIVERSITY  
SCHOOL OF GRADUATE STUDIES  
FACULTY OF TECHNOLOGY  
ELECTRICAL AND COMPUTER ENGINEERING  
DEPARTMENT

Performance of Beamforming Techniques Using Uplink Channel Characteristics  
in Cellular CDMA Communication Systems

By  
**Degebassa Dida**

A thesis submitted to the school of Graduate studies of Addis Ababa University  
in partial fulfillment of the requirements for the degree of

Masters of Science

In

Communication Engineering

September 2007

Addis Ababa, Ethiopia

ADDIS ABABA UNIVERSITY  
SCHOOL OF GRADUATE STUDIES  
FACULTY OF TECHNOLOGY  
ELECTRICAL AND COMPUTER ENGINEERING  
DEPARTMENT

Performance of Beamforming Techniques Using Uplink Channel  
Characteristics in Cellular CDMA Communication Systems

BY  
**Degebassa Dida**

Advisor

Dr.-ing. Hailu Ayele

ADDIS ABAABA UNIVERSITY  
SCHOOL OF GRADUATE STUDIES

Performance of Beamforming Techniques Using Uplink Channel  
Characteristics in Cellular CDMA Communication Systems

BY  
**Degebassa Dida**

FACULTY OF TECHNOLOGY

APPROVAL BY BOARD OF EXAMINERS

---

Chairman Department of Graduate  
Committee

---

Signature

---

Advisor

---

Signature

---

Examiner

---

Signature

---

External Examiner

---

Signature

## **Acknowledgement**

I would like to express my deepest appreciation and sincerest gratitude to my advisor Dr.ing. Hailu Ayele for his, supervision, knowledge, and persistent encouragements during my M.Sc thesis work.

I would like to thank all my senior colleagues in particular Ato Fikadu Wakijira for his advice. I thank Ato Mengash Alemu and Ato Hailu Dejene for providing the Laptop for this thesis work. Finally, I would like to thank my family; my wife w/o Genet Dejene for their love and patience and my parents for their continuous encouragement and moral support.

## Table of contents

|                                                       |      |
|-------------------------------------------------------|------|
| Abstract.....                                         | iv   |
| Acknowledgment.....                                   | vi   |
| Table of Contents.....                                | vii  |
| List of Figures.....                                  | x    |
| List of Tables.....                                   | xii  |
| Glossary of Acronyms.....                             | xiii |
| CHAPTER ONE                                           |      |
| 1. INTRODUCTION .....                                 | 1    |
| 1.1 Problem overview .....                            | 2    |
| 1.2 objective of the study .....                      | 4    |
| 1.3 Motivation for the study.....                     | 4    |
| 1.4 Methodology .....                                 | 5    |
| 1.5 Thesis outline .....                              | 6    |
| CHAPTER TWO                                           |      |
| 2. BACKGROUND OF THE CDMA UPLINK CHARACTERISTICS..... | 7    |
| 2.1 Introduction.....                                 | 7    |
| 2.2 Base-station receiver .....                       | 8    |
| 2.3 Uplink CDMA Transmission channel Model .....      | 8    |
| 2.3.1 Path Loss .....                                 | 8    |
| 2.3.2 Shadowing.....                                  | 9    |
| 2.3.3 Rayleigh Fading .....                           | 9    |
| 2.3.4 Channel Attenuation.....                        | 9    |
| 2.4 CDMA uplink channel waveform.....                 | 10   |
| 2.5 Encoding CDMA Uplink Data.....                    | 11   |
| 2.6 Digital Beamforming in uplink CDMA .....          | 11   |
| 2.7 Demodulation.....                                 | 12   |
| 2.7.1 Single Element Demodulation .....               | 12   |

|                                                           |    |
|-----------------------------------------------------------|----|
| 2.7.2 Multiple Element Demodulations .....                | 14 |
| CHAPTER THREE                                             |    |
| 3. DIGITAL BEAMFORMING AND ANTENNA ARRAY TECHNIQUES ..... | 17 |
| 3.1 Introduction.....                                     | 17 |
| 3.2 Fundamentals of digital beamforming .....             | 17 |
| 3.2.1 Beamforming weight vector and beampattern .....     | 18 |
| 3.3 Antenna array processing.....                         | 18 |
| 3.3.1 Uniform linear antenna array .....                  | 18 |
| 3.3.2 Circular antenna array .....                        | 19 |
| 3.3.3 Planar antenna array.....                           | 19 |
| 3.4 Digital Beamforming (DBF).....                        | 19 |
| 3.4.1 Element- space beamforming.....                     | 20 |
| 3.4.2 Beam-space beamforming.....                         | 21 |
| 3.4.3 DBF with Multiple Access Schemes .....              | 21 |
| 3.4.4 Digital Beamforming with CDMA .....                 | 22 |
| 3.5 Narrowband Beamforming.....                           | 22 |
| 3.6 Wideband Beamforming.....                             | 23 |
| 3.7 Switched Beamforming.....                             | 23 |
| 3.8 Adaptive Beamforming.....                             | 23 |
| 3.8.1 Adaptive Criteria.....                              | 24 |
| 3.8.2 Adaptive beamforming for Uplink.....                | 24 |
| 3.9 Adaptive Beamforming Algorithms .....                 | 25 |
| 3.9.1 Trained Adaptive Algorithms .....                   | 27 |
| 3.9.1.1 The LMS algorithm.....                            | 27 |
| 3.9.1.2 Recursive Least Square Algorithms.....            | 29 |
| 3.9.2 Blind Adaptive Algorithms.....                      | 30 |
| 3.9.2.1 Constant Modulus Algorithm (CMA).....             | 30 |
| 3.9.2.2 Decision-Directed Algorithm (DDA).....            | 32 |
| CHAPTER FOUR                                              |    |
| 4. COVERAGE AND CAPACITY ESTIMATION IN CDMA .....         | 33 |
| 4.1 Introduction.....                                     | 33 |

|                                                      |    |
|------------------------------------------------------|----|
| 4.2 Coverage.....                                    | 33 |
| 4.3 Capacity.....                                    | 34 |
| 4.4 Maximum Signal-to- Interference Noise Ratio..... | 34 |
| CHAPTER FIVE                                         |    |
| 5. ANALYSIS AND SIMULATION .....                     | 36 |
| 5.1 Digital Beamforming algorithms.....              | 36 |
| 5.2 System and Traffic Model.....                    | 37 |
| 5.3 Simulation model and Parameters.....             | 38 |
| 5.4 Simulation results and discussion.....           | 39 |
| 5.4.1 LMS Adaptive beamforming Algorithm.....        | 39 |
| 5.4.2 CMA for Adaptive Beamforming.....              | 50 |
| 5.4.3 Performance measure.....                       | 57 |
| 6. CONCLUSION AND FUTURE WORKS.....                  | 60 |
| 6.1 Conclusions.....                                 | 60 |
| 6.2 Future work.....                                 | 61 |

## LIST OF FIGURES

|     |                                                                                                                                                                                                                                                                                 |    |
|-----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 2.1 | Channel power gain (dB) vs distance in Km ...                                                                                                                                                                                                                                   | 10 |
| 2.2 | CDMA reverse link data encoding process showing “per frame”.....                                                                                                                                                                                                                | 11 |
| 3.1 | Uniform spaced linear antenna array (ULA).....                                                                                                                                                                                                                                  | 19 |
| 3.2 | Element-space digital beamformer.....                                                                                                                                                                                                                                           | 21 |
| 3.3 | Beam-space digital beamformer .....                                                                                                                                                                                                                                             | 21 |
| 3.4 | Classification of adaptive array algorithms.....                                                                                                                                                                                                                                | 25 |
| 3.5 | Adaptive beamforming configuration.....                                                                                                                                                                                                                                         | 26 |
| 3.6 | Signal-flow graph representation of the LMS algorithm.....                                                                                                                                                                                                                      | 29 |
| 3.7 | Signal-flow graph representation of the RLS algorithm.....                                                                                                                                                                                                                      | 30 |
| 5.1 | Adaptive Beamforming algorithm Flow chart .....                                                                                                                                                                                                                                 | 37 |
| 5.2 | System model.....                                                                                                                                                                                                                                                               | 38 |
| 5.3 | LMS algorithm for an adaptive array with 6 –element antennas and 5<br>interferers for antenna noise figure equal to 0.1, $\mu=0.01$ , and $d=0.5\lambda$ .....                                                                                                                  | 40 |
| 5.4 | LMS algorithm for an adaptive array with 6-element antennas and 5<br>interferers for antenna noise figure equal to 0.01, $\mu=0.01$ , and $d=0.5\lambda$ .....                                                                                                                  | 41 |
| 5.5 | Array factor for LMS algorithm for an adaptive array with 6-element<br>antennas and 5 interferers for antenna noise figure equal to 0.01, $\mu=0.01$ ,<br>and $d=0.5\lambda$ .....                                                                                              | 42 |
| 5.6 | LMS algorithm for an adaptive array with 4-element antennas<br>and 3 interferers for antenna noise figure equal to 0.01, $\mu=0.01$ , $d=0.5\lambda$<br>a) Amplitude and Phase estimation versus time of iteration.....<br>(b) Array factor (dB) versus theta (degrees).....    | 43 |
| 5.7 | LMS algorithm for an adaptive array with 6-element antennas and 5<br>interferers for antenna noise figure equal to 0.1 and $\lambda =d$ and $\mu = 0.01$<br>a) Amplitude and Phase estimation versus time of iteration.....<br>b) Array factor (dB) versus theta (degrees)..... | 44 |
| 5.8 | LMS algorithm for an adaptive array with 6 –element antennas and<br>4 interferers for antenna noise figure equal to 0.1 and $d=0.25\lambda$ and $\mu = 0.01$ .....                                                                                                              | 46 |
| 5.9 | LMS algorithm for an adaptive array with 6 –element antennas and 5 interferers                                                                                                                                                                                                  |    |

|                                                                                                                                                                     |    |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| for antenna noise figure equal to 0.1 and $0.5 \lambda = d$ and $\mu = 0.001$                                                                                       |    |
| a) Amplitude and Phase estimation versus time of iteration.....                                                                                                     | 47 |
| b) Array factor (dB) versus theta (degrees).....                                                                                                                    | 48 |
| 5.10 LMS algorithm for an adaptive array with 4-element antennas and 2 interferers for<br>Antenna noise figure equal to 0.01, $\mu=0.01$ , $d=0.5\lambda$           |    |
| a) Amplitude and Phase estimation versus time of iteration.....                                                                                                     | 49 |
| b) Array factor (dB) versus theta (degrees).....                                                                                                                    | 49 |
| 5.11 CMA algorithms for an adaptive array with 6-element antennas and 5<br>interferers for antenna noise figure equal to 0.01, $\mu =0.01$ , $d=0.5\lambda$ ....    | 51 |
| 5.12 CMA algorithm for an adaptive array with 6-element antennas and 5<br>interferers for antenna noise figure equal to 0.1. $\mu =0.01$ , $d=0.5\lambda$ .....     | 51 |
| 5.13 CMA algorithm for an adaptive array with 4-element antennas and 3<br>interferers for antenna noise figure equal to 0.01 $d=0.5\lambda$ , .                     | 52 |
| 5.14 CMA algorithms for an adaptive array with 4-element antennas and 2<br>interferers for antenna noise figure equal to 0.1 and $d =0.25\lambda$ and $\mu = 0.01$  |    |
| a) Amplitude and Phase estimation versus time of iteration.....                                                                                                     | 53 |
| b) Array factor (dB) versus theta (degrees).....                                                                                                                    | 53 |
| 5.15 CMA algorithm for an adaptive array with 4-element antennas and 2 interferers<br>for antenna noise figure equal to 0.01 and $\lambda =d$ and $\mu = 0.01$      |    |
| a) Amplitude and Phase estimation versus time of iteration.....                                                                                                     | 54 |
| b) Array factor (dB) versus theta (degrees).....                                                                                                                    | 55 |
| 5.16 CMA algorithm for an adaptive array with 4-element antennas and 3 interferers<br>for antenna noise figure equal to 0.01 and $0.5 \lambda =d$ and $\mu = 0.001$ |    |
| a) Amplitude and Phase estimation versus time of iteration.....                                                                                                     | 56 |
| b) Array factor (dB) versus theta (degrees).....                                                                                                                    | 56 |
| 5.17 Flow chart for the Performance evaluation algorithm .....                                                                                                      | 58 |
| 5.18 SINR vs BER.....                                                                                                                                               | 59 |

## **LIST OF TABLES**

|                                 |    |
|---------------------------------|----|
| 5.1 Simulation parameters.....  | 39 |
| 5.2 Summary of performance..... | 59 |

## Glossary of Acronyms

|                         |                                                   |
|-------------------------|---------------------------------------------------|
| A                       | Area                                              |
| AC                      | access channel                                    |
| ADC                     | Analog to digital converter                       |
| AF                      | Array factor                                      |
| AOA                     | Angle of arrival                                  |
| AWGN                    | Adaptive white Gaussian noise                     |
| BER                     | Bit error rate                                    |
| bpf                     | Bit per frame                                     |
| bps                     | Bit per second                                    |
| BS                      | Base station                                      |
| CDMA                    | Code division multiple access                     |
| CMA                     | Constant modulus algorithm                        |
| $cpf$                   | Chip per frame                                    |
| $d$                     | Inter- element antenna spacing                    |
| $E_b/I_o$               | Energy per bit per interference                   |
| $d(k)$                  | Desired signal                                    |
| DBF                     | Digital beamforming                               |
| DDA                     | Decision Directed Algorithm                       |
| DOA                     | Direction of arrival                              |
| DSP                     | Digital signal processing                         |
| $E^2\{\varepsilon(k)\}$ | Mean square error                                 |
| $E_b/N_o$               | Energy per bit per Noise                          |
| ESPRIT                  | Estimation of signal parameter Identification via |
| $f_c$                   | Carrier frequency                                 |
| FDMA                    | Frequency division multiple access                |
| FFT                     | Fast Fourier Transform                            |
| $G$                     | Processing gain                                   |
| $G(\theta, \phi)$       | Beam pattern                                      |
| Hz                      | Hertz                                             |
| I                       | in- phase                                         |
| IS-95 CDMA              | Interim-95 Code Division multiple access          |
| $K$                     | Number of antenna element                         |
| LMS                     | Least mean square                                 |
| LOS                     | Line-of-sight                                     |
| LS                      | Least square                                      |
| MISO                    | Multiple input single output                      |
| MMIC                    | Monolithic microwave integrated circuit           |
| MSs                     | Mobile stations                                   |
| MT                      | Mobile terminal                                   |
| MUSIC                   | Multiple Signal Classification                    |

|                |                                                     |
|----------------|-----------------------------------------------------|
| $N$            | Number of users in the cell                         |
| $n(t)$         | Wide-sense stationary zero-mean Gaussian            |
| $N_M$          | Number of mobiles                                   |
| $N_{pole}$     | Pole capacity                                       |
| $N_{TH}$       | Power spectral density                              |
| PCS            | Personal communication system                       |
| $Pdf$          | Probability density function                        |
| pg             | Processing gain                                     |
| PN             | Pseudo- Noise                                       |
| PSD            | Power spectrum distribution                         |
| $P_t$          | Limited power                                       |
| Q              | Quadrature phase                                    |
| QoS            | Quality of service                                  |
| $R_b$          | Bit rate                                            |
| RF             | Radio frequency                                     |
| RLS            | Recursive least square                              |
| RTC            | Reverse traffic channel                             |
| $R_n(\tau)$    | Noise with autocorrelation                          |
| $r_{xd}$       | Cross correlation                                   |
| $R_{xx}$       | Covariance matrix                                   |
| SCORE          | Spectral self- COherence REstoral                   |
| $S_k$          | Shadowing multiplicative attenuation                |
| SINR           | Signal interference- plus- Noise ratio              |
| SMI            | Sample mark inversion                               |
| SNR            | Signal- to -noise ratio                             |
| SOI            | Signal of interest                                  |
| $T_c$          | Period of one PN chip                               |
| TDMA           | Time division multiple access                       |
| UMTS           | Universal mobile telephone system                   |
| $W$            | System bandwidth                                    |
| $w_p^*$        | Complex beamforming weight                          |
| $w_{cpf}$      | Walsh chips per frame                               |
| $w_{fpf}$      | Walsh function per frame                            |
| $w_{opt}$      | Optimum weight                                      |
| $x_i(k)$       | Array element out put                               |
| $X_k$          | Data signal                                         |
| $x_P$          | Weight sum of the elements of $x$                   |
| $y(k)$         | Array out put at time k                             |
| $\alpha$       | Path loss exponent                                  |
| $\beta$        | Inter-cell interference                             |
| $\beta_k$      | Signal envelope for $k^{th}$ mobile at the receiver |
| $\beta_{k(t)}$ | Signal strength for the $k^{th}$ mobile             |
| $\gamma$       | Propagation loss exponent                           |
| $\eta_k$       | Signal attenuation over the transmission channel    |
| $\lambda$      | Wave length                                         |
| $\xi$          | Normal distribution                                 |
| $\rho_k$       | Path loss attenuation factor                        |

|                        |                                                                    |
|------------------------|--------------------------------------------------------------------|
| $\tau_k$               | Time delay                                                         |
| $\omega_i$             | $i^{th}$ entry of beamforming weight vector                        |
| $F(\theta)$            | Array factor                                                       |
| $\tau_{kp}$            | Time delay of the $i^{th}$ mobiles signal relative to the $p^{th}$ |
| $\sigma_n^2$           | Variance of the background thermal noise                           |
| $\sigma_s^2$           | Variance of the shadowing random variable                          |
| $\theta_{Q_k}$         | Quadrature angle used for generating a $\pm 1$ component in        |
| $\theta_{I_k}$         | in phase angle used for generating a $\pm 1$ component             |
| $\mathcal{E}(k)$       | Error function                                                     |
| $\omega(o)$            | Initial weight vector                                              |
| $ y(n) $               | Instantaneous amplitude of the array output                        |
| $\overline{\rho}(D_i)$ | Path loss                                                          |
| $H$                    | Hermitian transpose                                                |
| $\mu$                  | Step size/convergence rate                                         |

## **Abstract**

The demand of code division multiple access (CDMA) cellular communication has grown remarkably in the past few years. Due to this reason mobile communication networks should use the limited resources of the system in efficient and convenient manner. One of the techniques is digital beamforming, in which energy is dynamically adjusted to the desired direction. In beamforming, we multiply a complex-valued weighting vector by the outputs of antenna array and sum these outputs to generate a signal for each user. As each user will have a unique weight, we can select the weights to greatly decrease interference from other users, and therefore increase system capacity.

In this thesis, the uplink CDMA cellular systems is modeled and investigated to show how different beamforming algorithms are effective in improving system performance and capacity using adaptive antenna arrays at base station. It was initially thought that only the uplink would be the capacity limiting link in CDMA cellular systems but this assumption turned out to be incorrect and the capacity in actual fact is limited by the both links. To determining uplink CDMA system capacity and the effect of inter-element antenna array spacing, step size and number of antenna elements on the overall system capacity using adaptive antenna array elements.

Adaptive beamforming is more complex, but highly efficient where the radiation pattern is constructed dynamically in which interferers are blocked and beam is formed in the direction of users. The adaptive algorithms such as, LMS and CMA are investigated to estimate error and to compute array factor for CDMA system, because the spreading codes can be used as a reference for beamforming.

Results of comparison show that the LMS algorithm performs better than the CMA algorithm under the same condition. The performance obtained from using beamforming techniques using six (6) adaptive antenna array element results in an improvement by about 30% on average over the non beamforming techniques from the simulation result obtained.

# CHAPTER ONE

## 1. INTRODUCTION

The growth of telecommunications, from the wired phone to personal communication services (PCS), is resulting in the accessibility of wireless services that were not formerly considered realistic. In providing different types of wireless services, capacity and reliability are limited by three major impairments: multipath fading, delay spread, and co-channel interference [29]. In mobile communications, the fast increasing number of cellular phones demands an ever rising capacity for future mobile communications systems. In order to meet the needs of an ever increasing demand for mobile communications, the spread spectrum communication scheme was introduced, which is called as code division multiple access (CDMA). CDMA is interference limited multiple access technique. To enhance its cell capacity, frequency reuse and cell sectorization are used. But, in this work, we look into another capacity enhancement technique called as digital beamforming (DBF).

DBF represents a quantum step in antenna performance and complexity. It is based on deep-rooted theoretical concepts that are now becoming practically usable, mainly as a consequence of current major advances in areas such as monolithic microwave integrated circuit (MMIC) and digital signal processing techniques. DBF technology has reached a satisfactory level of maturity, such that it can become practical for communications in improving system performance. A common DBF antenna system consists of three major components: the antenna array, the digital transceivers and digital signal processor.

Digital beamforming is based on capturing the radio frequency (RF) signals at every antenna element and converting them into two streams of binary baseband signals (i.e. in-phase (I) and quadrature-phase (Q)). These digital baseband signals represent the amplitudes and phases of signals received at each component of the array. The beamforming is carried out by weighting each of the digital signals, thus adjusting their amplitudes and phases such that when added together they form the desired beam. Beamforming is a classical method of processing temporal sensor array measurements for signal estimation, interference cancellation, and spectrum estimation. The main concept of beamforming is, conveniently weighting and combining the signals received from multiple sensors or antennas.

Beamforming can be implemented to suppress interference by using antenna arrays at the base station, thereby increasing the system capacity.

The major benefit to be gained from digital beamforming is critically added flexibility without the degradation in signal-to-noise ratio (SNR) of the system. Therefore, all of the information arriving at the antenna interface is captured in the digital streams and driven by beamforming instructions software routines to add wide-ranging flexibility in the types of beams such as, scanned beams, multiple beams, shaped beams and beams with steered nulls [4]. Digital beamforming process has an upper level of information by its flexibility and permits a number of attractive features beyond the capabilities of predictable phased arrays. Also, DBF systems are capable of carrying out real-time calibration of antenna systems in the digital domain. As a consequence study of different DBF techniques and evaluation of their performance has become an important area of research.

## **1.1 Problem overview**

In wireless cellular system the method of enhancing system capacity and coverage are significant issues because of the increased demand for wireless connectivity. Thus it is important to develop a system that can handle the increased number of users. In [3], [6] methods to increase system capacity, develop spectrum efficiency in digital cellular spread spectrum code division multiple access (CDMA) scheme are discussed. Code division multiple access (CDMA) communications system is interference limited, as all of the users share the same bandwidth with each mobile's signal being spread by a unique individual pseudo-noise (PN) chip sequence and the received signal correlated at the base station with each user's known sequence, to recover the original data.

A direct way to increase system capacity is to create the number of cells which increases infrastructure costs and requires more frequent hand-offs between adjacent cells. In cellular CDMA system significantly increased capacity for a given amount of bandwidth is a research area aimed at finding methods by which the multiple access interference (MAI) can be reduced. By increasing capacity, a given area can be covered with fewer base stations, thereby reducing the overall cost of the cellular communications systems. In addition, by

avoiding the need for subdividing cells, the logical problems of performing hand-offs between adjacent cells as users move around are reduced. One way of accomplishing this task is through the use of digital beamforming based on antenna arrays.

Digital beamforming improves the CDMA system capacity by reducing the interference. This can be realized using a multi-element antenna array at the base station and a suitable set of beamforming coefficients. It also takes a weighted sum of the antenna array outputs to maximize the ratio of the desired mobile's SINR. DBF is a form of spatial filtering which reduces the effect of interference and thus boosts the overall cell capacity. This system contains an array of antenna elements at the receiving end, A/D, N point FFT (optional), beam selectors, weight multiplier, and finally summer circuit elements assuming as there is an arriving signal.

Digital beamforming method improves the signal-to-noise ratio (SNR) of a received signal through the use of an antenna array. In [7],[8],[6], signals arriving from the desired direction multiply a complex-valued weighting vector to the outputs of antenna array and sum these outputs to generate a signal for each user. These weights are chosen to co-phase the desired signal, while randomizing the phases of the signals arriving from other directions. These signals form a moveable beam pattern that can be steered to a desired direction that tracks the mobile stations (MSs) as they move. This allows the antenna system to focus radio frequency (RF) energy on a particular mobile station and minimizes the impact of interference [5].

In this work, uplink transmission is the primary area of focus where it is not realistically practical to implement beamforming at the mobile stations due to the necessary antenna arrays and processing requirements. In different literature, [4], [5], [7] the application of different beamforming scenarios that maximizes the SNR for mobiles simultaneously by raising the beam pattern in the direction of mobile users and lower it in other direction have been studied. In a cellular system, the desired and the interfering signals originate from different spatial locations. This spatial separation is exploited by a beamformer which can be regarded as a spatial filter separating the desired signal from the interference. The signals

from different antenna elements are weighted and summed to optimize the quality of the signal. There are different DBF that have been developed & suggested over the last few years. Each of such methods does not have identical performance and need to establish their relative merits using different performance criteria.

## **1.2 Objective of the study**

The objective of this work is to analyze the feasibility and benefits of enhancing the cell capacity performance using the DBF techniques. The underlying objective is to reduce the interference signal and maximize the reception of the desired signal which leads to the capacity improvement. In addition to this, the cell coverage is increased and direction of arrival (DOA) is estimated using the DBF techniques.

The specific objectives are:

- ❖ To investigate the performance of DBF schemes in estimating the desired signal and interference, so as to maximize signal to interference noise ratio (SINR).
- ❖ Study the impact of DBF on the CDMA system capacity and coverage enhancement through the use of adaptive algorithm error minimization and effective rejection of interferers.
- ❖ To investigate the relative merit of different techniques (algorithms) employed in DBF especially the adaptive algorithms; Least mean square (LMS) and constant modulus algorithm (CMA).

## **1.3 Motivation for the study**

Wireless cellular network consists of a large number of cells, mobile users with various movement patterns: (speed and directions), base stations and a number of mobile terminals (MTs). With current technology, it is now possible to implement very sophisticated and computationally intensive digital signal processing algorithms for practical cellular communications. In addition, comprehensive investigations have been conducted on the use of a smart antenna array at the base station for optimal directional reception and transmission in order to further increase cell capacity. To further increase cell capacity, it has been demonstrated that adaptive or self-adjusted antenna array beamformers hold great potential for improving SINR and hence achieving higher cell capacities [9].

The main purpose of implementing beamforming in communications systems is to combat multipath, extend coverage by the base station and reuse frequency channels within the cell. Digital beamforming based on the antenna array will be one key method in satisfying this task. In beamforming, we apply complex-valued weighting vectors to the outputs of the antenna array and sum the results to generate a signal for every user. As each user has unique weights, we may select the weights to greatly decrease the interference from other users, and therefore increase system capacity, which initiate our motivation.

## **1.4 Methodology**

As discussed in the above subsections, the methodology of this work will have a specific emphasis on the beamforming techniques to maintain QoS, increasing system capacity and improve the network coverage. Therefore, beamforming is used only at the base station, and the capacity is partially determined by the uplink, since the received signal data is processed to get the required beamforming information. Beamforming can also be implemented on the downlink using feedback information from the mobile station [3].

The wireless channel used for the mobile cellular network is a fading channel; due to mobility effect the Gaussian channel model is not practical. The fast fading effect is most serious than the path loss in mobile communication. For the time varying signal environments, statistics change with time as the target mobile and interferers move around the cell. For the time-varying signal propagation environment, a recursive update of the weight vector is needed to track a moving mobile so that adaptive beamforming algorithms is used in mobile communication systems to determine the optimal weight vectors of array antenna elements dynamically, based on different performance criteria such, Mean square error (MSE), the Maximum Likelihood (ML), Maximum Signal to Interference Noise Ratio (SINR). The weighting coefficient is a function of the signal's direction of arrival and the antenna array geometry. The optimum weighting coefficients are obtained from the Wiener solution [9]. In beamforming there are two approaches: switched and adaptive beamforming. Element-space beamforming and beam-space beamforming techniques are discussed in different papers [4], [33]. Beamforming can be implemented with different types of

algorithms: adaptive beamforming algorithms and switched beamforming algorithms. Switched beamforming algorithm is less efficient compared to the adaptive algorithms [33]. They only implement steering the direction of arrival of a desired signal and this algorithm is not used in strong interference cancellation. We will try to implement LMS & CMA adaptive algorithms in the simulation study, in order to see how the beamforming enhances the system capacity in CDMA cellular system. The performance measurements are made through simulation study of array factor, error estimation over time change, angle of arrival, steering angles and SINR vs BER. The simulation study implemented is with MATLAB codes and analysis done by detailed discussion of simulation results obtained.

## **1.5. Thesis Outline**

Chapter 2 provides further details regarding the technologies involved in the system under consideration. Chapter two also presents the details of CDMA uplink characteristics. Chapter 3 is the literature review area and further details of the problem addressed in the pervious chapters. Also this chapter includes a description of the different beamforming techniques, algorithms, antenna array processing techniques, and its associated requirements, assumption, and specifications. In this chapter we also discuss about adaptive beamforming approaches in detail.

Cell capacity and coverage problems for the uplink CDMA cellular communication systems are presented in chapter 4. Chapter 5 is the main part of this work which includes simulation study, analysis and discussion of the simulation results. In this chapter we also discuss system model, simulation model and simulation results. Finally chapter 6 summarizes the conclusions reached at the end of this work and indicates potential future work. It also briefly addresses some of the limitation of the work and identifies areas for further work by other researcher.

# **CHAPTER TWO**

## **BACKGROUND OF THE CDMA UPLINK CHARACTERISTICS**

### **2.1 Introduction**

Code division multiple access (CDMA) is a multiple access method where different users share the same physical medium, which is, the same frequency band, at the same time. The main component of CDMA is the spread spectrum method that uses high rate signature pulses to enhance the signal bandwidth for a given data rate. In a CDMA scheme, the different users can be identified and, separated at the receiver by means of their characteristic individual signature pulses, that is, by their individual codes [32], [38].

The most prominent applications of CDMA is in mobile communication systems like cdmaOne (IS-95), UMTS or cdma2000. To apply CDMA in a mobile radio environment, specific additional methods need to be implemented in all the different systems. Methods such as power control and soft handover have to be applied to control the interference by other users and to be able to separate the users by their respective codes [32].

The second generation CDMA ordinary air interface standard proposed in [10] was officially adopted as the North American digital cellular. Beginning field trials of IS-95 have verified that a CDMA cellular system can increase capacity by 10 to 13 times over that of an analog cellular system under practical channel environment. This can more improve the received signal quality using antenna array at the base station that will in turn increase the cell capacity. It is not realistically practical to implement beamforming at the mobiles due to the necessary antenna arrays and computational complexity. Even though, beamforming is used only at the base station, the uplink condition is one of the factors in capacity enhancement while the received signal data can be directly processed to acquire the required beamforming information.

## 2.2 Base-station receiver

For uplink reception, the base station uses dual antenna diversity in each of the  $120^\circ$  sector to offer path diversity. But each sector uses different frequency to reduce co-channel interference; therefore handoffs between sectors are required for higher performance, but this will result in too many handoffs [40]. This problem can be overcome using adaptive antenna arrays at base station. At base station antenna array, multipath signals arriving with more than one chip delay ( $T_c$ ), which will fade independently since PN sequences have nearly zero correlation for time offsets greater than one chip.

## 2.3 Uplink CDMA Transmission Channel Model

This section deals with how the uplink transmission channel between the mobiles and base station is modeled. This mostly involves quantities specifying the reduction of signal strength over the transmission channel because of factors such as path loss, shadowing, and Rayleigh fading.

### 2.3.1 Path Loss

Path loss is defined as an overall reduction of signal strength with distance between the transmitter and receiver. Measurements show [11] that the average received signal power decreases logarithmically with distance. The common path loss  $\bar{\rho}(D_k)$  can be expressed as

$$\bar{\rho}(D_k) = \bar{\rho}(D_0) + 10 \alpha \log_{10} \left( \frac{D_k}{D_0} \right), \quad \text{dB} \quad (2.1)$$

Where  $\alpha$  is the path loss exponent and indicates the rate of signal power attenuation with distance.  $D_0$  is the close-in reference distance that is determined from measurements close to the transmitter,  $D_k$  is the distance between the  $i^{th}$  MS and the BS and  $\bar{\rho}(D_0)$  is the average path loss at close-in reference distance [11]. The value of  $\alpha$  depends on the explicit propagation situation, in free space  $\alpha = 2$ .

Path loss is higher for non- line- of -sight (NLOS) than line- of -sight (LOS) path. As seen from equation (2.1) the power of the signal falls as a function of distance. Therefore, the

amplitude of the signal envelope should be multiplied by  $\sqrt{\rho_k}$ , where  $\rho_k$  is the power of the signal. The path loss quantity is updated dynamically as mobiles move inside the cell.

### 2.3.2 Shadowing

Shadowing or slow fading represents the variation of average path loss. Shadowing is caused by various obstacles such as trees and buildings in the transmission path of a mobile. A commonly accepted shadowing model is to use a lognormal random variable [5] such as:

$$S_k = 10^{\xi/10} \quad (2.2)$$

where  $\xi$  has an  $N(0, \sigma_s^2)$  distribution. Typical value used for  $\sigma_s$  is 8dB.

A new value of  $S_k$  is chosen for each mobile at the start of each call by that mobile, and  $S_k$  remains constant for the duration of each call. Thus considering shadowing, the overall path loss  $\rho(D_k)$  is given by

$$\rho(D_k) = \bar{\rho}(D_k) + S_k, \text{ (dB)} \quad (2.3)$$

Where  $\bar{\rho}(D_k)$  is given in (2.1) and  $S_k$  is a lognormal random variable.

### 2.3.3 Rayleigh Fading

Rayleigh fading is a statistical model for the effect of a propagation environment on a radio signal, such as that used by wireless devices. It assumes that the magnitude of a signal that has passed through such a transmission medium (also called a communications channel) will vary randomly, or fade, according to a Rayleigh distribution. It is a reasonable model for signal propagation as well as the effect of heavily built-up urban environments on radio signals. Rayleigh fading is most applicable when there is no line of sight between the transmitter and receiver. If there is a line of sight, Rician fading is more applicable. Due to multipath propagation it can also be integrated with the channel model. This form of fading can be modeled as either uncorrelated or correlated. Rayleigh fading is a reasonable model when there are many objects in the environment that scatter the radio signal before it arrives at the receiver. An introduction to the characteristics and methods for overcoming Rayleigh amplitude fading ( $R_k$ ) in mobile digital communication systems can be found in [12].

### 2.3.4 Channel Attenuation

As the signal travels to the base station, its strength is attenuated by the factors discussed above. Thus, the envelope is multiplied by the following value to decide its magnitude as observed at the base station [39].

$$\eta_k = \sqrt{\rho_k S_k} R_k \quad (2.4)$$

The path loss and shadowing terms influence the power of a signal. A typical channel power gain due to the three fading effects is illustrated in figure (2.1) below:

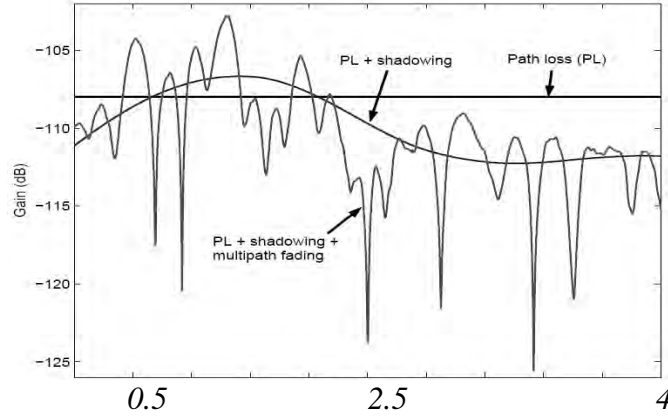


Figure 2.1: Channel power gain (dB) vs distance (km)

## 2.4 CDMA Uplink channel waveform

The CDMA uplink channel employs access channel (AC) and Reverse traffic channel (RTC). The access channel is used by the mobile to initiate communication with the base station and to respond to paging channel messages [38].

The CDMA uplink uses PN spread spectrum modulation with 1.25MHz bandwidth using a length  $2^{15} - 1$  short code sequence of chip rate 1.2288Mcps. Signals from different mobiles are distinguished at the base station by the use of a very long PN ( $2^{42} - 1$ ) sequence with a user address determined by time offset [40]. The uplink data stream is then split into in-phase (I) and quadrature (Q) phase streams where each is further modulated with the short codes at chip rate 1.2288Mcps and then modulated by a carrier frequency ( $f_c$ ). To maximize system capacity, the uplink imposes power control where the mobile transmission powers

are adjusted so that the received signal-to-noise ratios  $E_b / N_0$  at the base station are equal for each mobile station.

## 2.5 Encoding CDMA Uplink Data

The source data is assumed to consist of sequences of random bits representing condensed voice signals that are generated as necessary. In order to provide security against errors, the data bits under goes a multi-level encoding process as shown in figure 2.2 below.

The data can be transmitted at a bit rate  $R_B=9600\text{bps}$  which includes overhead. These bits are divided into frames with  $F$  (50) frames per second and  $B_T=192\text{bits}$  per frame. The frame bits are first passed through a rate 1/3 convolutional encoder with a constraint length of 9. This produces 576 encoded bits per frame. The next stage involves block interleaving the encoded bits using a 32X18 inter leavers matrix. The resulting bits are then grouped into sets of  $N_B=6$  to form binary number which select one of  $2^{N_B}=64$  different Walsh functions. This yields a total of 96 Walsh functions per frame. Each Walsh functions have a sequence of 64 chips associated with it which results in a total of 6144 Walsh chips per frame. Then Walsh chips multiplied with the spreading factor 4 which results in a total of 24576cpf.

The resulting signal is then split into in-phase and quadrature factors with each factor being further encoded by a different short code sequence[39], every with a period of  $2^{15}$  chips. The in-phase and quadrature components are modulated using PSK (Phase Shift Keying) with a carrier frequency of  $f_c$ .

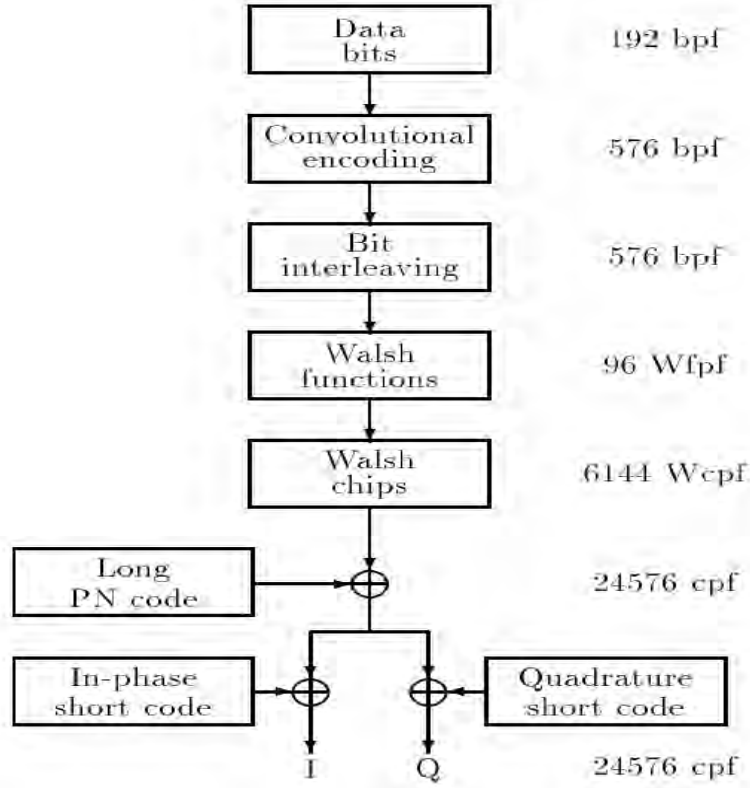


Figure 2.2: CDMA reverse link data encoding process showing “per frame”

## 2.6 Digital Beamforming for Uplink CDMA

Digital beamforming can be utilized to increase the capacity of a cell [13]. By using a multi-element antenna array at the base station and an appropriate set of beamforming coefficients, it is possible to take the weighted sum of the antenna element outputs that maximizes the ratio of the desired mobile’s signal power- to- interference plus noise power (SINR). This is basically a type of spatial filtering that reduces interference from other mobiles and therefore boosts overall cell capacity. Such process discussed here how it affects the uplink signals. For a multi-element base station antenna array, the received signal vector  $x(t)$  given [3] as:

$$x(t) = \sum_{k=1}^{N_M} \beta_k(t) a_k + n(t) \quad (2.5)$$

Where  $\beta_k(t)$  represents the signal strength from the  $k^{th}$  mobile,  $a_k$  is the corresponding array vector and  $n(t)$  represents wide-sense stationary zero-mean Gaussian noise with autocorrelation is:

$$R_n(\tau) = \begin{cases} \sigma_n^2 I, & \tau = 0 \\ 0, & \tau \neq 0 \end{cases} \quad (2.6)$$

It is necessary to reduce the complex vector  $x$  to a scalar quantity that can be used in the rest of the decoding process. This can be accomplished by taking a weighted sum of the elements of  $x$ .

$$x_p = w_p^* x \quad (2.7)$$

Each elements of the signal with the beamforming weight vector  $w_p$  induces a phase shift in the equivalent entry of the signal vector  $x$ .

## 2.7 Demodulation

In this subsection we are interested in the demodulation process since we are assuming the beamforming effect at the receiver part rather than at the transmitter side to recover the original data and also due to the reason that implementing multiple antenna elements at base station is simpler than at mobile station. It is simpler to think of the demodulation of the signal from a single antenna element first and then to generalize this result to a multi-element antenna array.

### 2.7.1 Single Element Demodulation

Consider the PSK modulated in-phase and quadrature signal components originating from the  $k^{th}$  mobile [38, 39]:

$$x_{I_k}(t) = \cos(2\pi f_c t + \theta_{I_k}(t)), \text{ and } x_{Q_k}(t) = \sin(2\pi f_c t + \theta_{Q_k}(t)) \quad (2.8)$$

Where  $\theta_{I_k}(t)$  and  $\theta_{Q_k}(t)$  is the phase angles used to modulate the carrier signal which are limited to the following values:

$$\theta_{I_k}, \theta_{Q_k} \in \{0, \pi\} \quad (2.9)$$

The combined signal can be written as:

$$X_k(t) = x_{I_k}(t) + x_{Q_k}(t) = \cos(2\pi f_c t + \theta_{I_k}(t)) + \sin(2\pi f_c t + \theta_{Q_k}(t)) \quad (2.10)$$

Introducing a random time delay  $\tau_k$  which will cause a phase shift, due to the lack of time synchronization between the mobiles on the uplink which is assumed to be uniformly distributed over the interval  $[0, T_c)$ .

$$X_k(t) = \cos(2\pi f_c(t - \tau_k) + \theta_{I_k}(t - \tau_k)) + \sin(2\pi f_c(t - \tau_k) + \theta_{Q_k}(t - \tau_k)) \quad (2.11)$$

$\theta_{I_k}$  and  $\theta_{Q_k}$  have a much lower frequency than carrier frequency  $f_c$ , so their dependence on time and the effects of the delay  $\tau_k$  on these terms can be ignored during this analysis.

$$\begin{aligned} X_k(t) &= \cos(2\pi f_c(t - \tau_k) + \theta_{I_k}) + \sin(2\pi f_c(t - \tau_k) + \theta_{Q_k}) \\ &= \{\cos(\theta_{I_k})\cos(2\pi f_c(t - \tau_k)) - \sin(\theta_{I_k})\sin(2\pi f_c(t - \tau_k))\} + \\ &\quad \{\sin(\theta_{Q_k})\cos(2\pi f_c(t - \tau_k)) + \cos(\theta_{Q_k})\sin(2\pi f_c(t - \tau_k))\} \\ &= \{\cos(\theta_{I_k}) + \sin(\theta_{Q_k})\}\cos(2\pi f_c(t - \tau_k)) + \{-\sin(\theta_{I_k}) + \cos(\theta_{Q_k})\}\sin(2\pi f_c(t - \tau_k)) \end{aligned} \quad (2.12)$$

Equation (2.12) can also be written in phasor format to simplify the rest of the analysis.

$$\begin{aligned} X_k(t) &= \text{Re}\left\{\left[\cos(\theta_{I_k}) + \sin(\theta_{Q_k})\right] - j\left[-\sin(\theta_{I_k}) + \cos(\theta_{Q_k})\right]\exp(j2\pi f_c(t - \tau_k))\right\} \\ &= \text{Re}\left\{\left[\cos(\theta_{I_k}) - j\cos(\theta_{Q_k})\right]\exp(-j2\pi f_c(t - \tau_k))\right\} \\ &= \text{Re}\left\{\left[\cos(\theta_{I_k}) - j\cos(\theta_{Q_k})\right]\exp(-j2\pi f_c\tau_k)\exp(j2\pi f_c t)\right\} \end{aligned} \quad (2.13)$$

The sin terms in (2.13) will evaluate as zero due to the constraints given in (2.9) and can be dropped from the equation. The cos terms will evaluate as  $\pm 1$ .

To demodulate the in-phase component, equation (2.13) is multiplied by  $\cos(2\pi f_c t)$  and low-pass filtered.

$$\cos(2\pi f_c t) = 1/2\{\exp(j2\pi f_c t) + \exp(-j2\pi f_c t)\} \quad (2.14)$$

The final in-phase demodulated result [39]:

$$\begin{aligned} D_{I_k}(t) &= \text{Re}\left\{\left[\cos(\theta_{I_k}) - j\cos(\theta_{Q_k})\right]\exp(-j2\pi f_c\tau_k)\exp(j2\pi f_c t)\left[1/2\{\exp(j2\pi f_c t) + \exp(-j2\pi f_c t)\}\right]\right\} \\ &= \text{LPF} \quad \text{Re}\left\{\frac{1}{2}[\cos(\theta_{I_k}) - j\cos(\theta_{Q_k})]\exp(-j2\pi f_c\tau_k)\right\} \\ &= \left\{\frac{1}{2}[\cos(\theta_{I_k})\cos(2\pi f_c\tau_k) - \cos(\theta_{Q_k})\sin(2\pi f_c\tau_k)]\right\} \end{aligned} \quad (2.15)$$

To eliminate the quadrature signal from the in-phase, the time delay  $\tau_k$  of the effective phase multiplied by  $2\pi$ , to recover the in-phase signal exactly. Otherwise, the quadrature signal will contribute interference.

To demodulate the quadrature factor, equation (2.13) is multiplied by  $\sin(2\pi f_c t)$  and low-pass filtered where:

$$\sin(2\pi f_c t) = -j \frac{1}{2} \{ \exp(j2\pi f_c t) - \exp(-j2\pi f_c t) \} \quad (2.16)$$

This gives the demodulated quadrature signal:

$$\begin{aligned} U_{Q_k}(t) &= \text{Re} \left\{ [\cos(\theta_{I_k}) - j \cos(\theta_{Q_k})] \exp(-j2\pi f_c \tau_k) \exp(j2\pi f_c t) \left[ -j \frac{1}{2} \{ \exp(j2\pi f_c t) - \exp(-j2\pi f_c t) \} \right] \right\} \\ &\stackrel{LPF}{=} \text{Re} \left\{ \frac{1}{2} [j \cos(\theta_{I_k}) + \cos(\theta_{Q_k})] \exp(-j2\pi f_c \tau_k) \right\} \\ &= \frac{1}{2} [\cos(\theta_{I_k}) \sin(2\pi f_c \tau_k)] + \frac{1}{2} [\cos(\theta_{Q_k}) \cos(2\pi f_c \tau_k)] \end{aligned} \quad (2.17)$$

The phase shift due to the time delay is an integral multiple of  $2\pi$ ; otherwise there will be interference from the in-phase component. Therefore, equations (2.15) and (2.17) represent the demodulation of a signal from a single mobile. As a result, it is essential to sum the demodulation results for each mobile in the system in order to obtain the total demodulated signal as well as interfering signals from the others

## 2.7.2 Multiple Element Demodulations

Let consider a received signal from mobile 'k' in the form of (2.13).

$$X_k(t) = \text{Re} \{ [\cos(\theta_{I_k}) - j \cos(\theta_{Q_k})] \exp(-j2\pi f_c \tau_k) \exp(j2\pi f_c t) \} \quad (2.18)$$

For the  $i^{th}$  antenna element, there is a recognized phase offset  $\alpha_{ki}$  relative to the center of the array. This phase offset depends on the angle of arrival of the signal and the geometry of the antenna array, and the value of this quantity comes directly from the array response vector. Therefore, with this additional phase offset the received signal at the  $i^{th}$  array element can be written as:

$$X_{ki}(t) = \text{Re} \{ [\cos(\theta_{I_k}) - j \cos(\theta_{Q_k})] \exp(-j\alpha_{ki}) \exp(-j2\pi f_c \tau_k) \exp(j2\pi f_c t) \} \quad (2.19)$$

Let  $w_{pi}$  be the  $i^{th}$  entry in the beamforming weight vector for the  $p^{th}$  mobile which is the mobile of interest. This represents a phase shift of the corresponding element of the signal vector, so  $w_{pi}$  can be written as:

$$w_{pi} = \exp(-j\phi_{pi}) \quad (2.20)$$

$$w_{pi}^* = \exp(j\phi_{pi}) \quad (2.21)$$

Thus, by combining equations (2.7), (2.19) and (2.21), the final complex quantity to be demodulated can be obtained as:

$$\begin{aligned}
X_p(t) &= \sum_{i=1}^{N_A} w_{pi}^* \sum_{k=1}^{N_M} X_{ki}(t) \\
&= \sum_{i=1}^{N_A} \sum_{k=1}^{N_M} \text{Re} \left\{ [\cos(\theta_{I_k}) - j \cos(\theta_{Q_k})] \exp(j\phi_{pi}) \exp(-j\alpha_{ki}) \exp(-j2\pi f_c \tau_{kp}) \exp(j2\pi f_c t) \right\} \\
&= \sum_{i=1}^{N_A} \sum_{k=1}^{N_M} \text{Re} \left\{ [\cos(\theta_{I_k}) - j \cos(\theta_{Q_k})] \exp(j(\phi_{pi} - \alpha_{ki} - 2\pi f_c \tau_{kp})) \exp(j2\pi f_c t) \right\} \quad (2.22)
\end{aligned}$$

$X_p(t)$  represents the received signal vector containing signals from all of the mobiles after having undergone beamforming with the weights for the  $p^{th}$  mobile where  $\tau_{kp}$  represents the comparative time delay of the  $k^{th}$  mobile's signal while  $\tau_{pp}$  is assumed to be zero.

The demodulated in-phase and quadrature components should be:

$$\begin{aligned}
D_{I_p}(t) &= \frac{1}{2} \sum_{i=1}^{N_A} \sum_{k=1}^{N_M} \beta_k \omega_k(t) \{ \cos(\theta_{I_k}) \cos(\phi_{pi} - \alpha_{ki} - 2\pi f_c \tau_{kp}) + \cos(\theta_{Q_k}) \sin(\phi_{pi} - \alpha_{ki} - 2\pi f_c \tau_{kp}) \} \quad (2.23) \\
&= \frac{1}{2} \sum_{i=1}^{N_A} \sum_{k=1}^{N_M} \beta_k \omega_k(t) \{ C_{I_k} \cos(\alpha_{ip} - \phi_{ki} - 2\pi f_c \tau_{kp}) + C_{Q_k} \sin(\alpha_{ip} - \phi_{ki} - 2\pi f_c \tau_{kp}) \}
\end{aligned}$$

$$\begin{aligned}
D_{Q_p}(t) &= \frac{1}{2} \sum_{i=1}^{N_A} \sum_{k=1}^{N_M} \beta_k \omega_k(t) \{ -\cos(\theta_{I_k}) \sin(\phi_{pi} - \alpha_{ki} - 2\pi f_c \tau_{kp}) + \cos(\theta_{Q_k}) \cos(\phi_{pi} - \alpha_{ki} - 2\pi f_c \tau_{kp}) \} \quad (2.24) \\
&= \frac{1}{2} \sum_{i=1}^{N_A} \sum_{k=1}^{N_M} \beta_k \omega_k(t) \{ -C_{I_k} \sin(\phi_{pi} - \alpha_{ki} - 2\pi f_c \tau_{kp}) + C_{Q_k} \cos(\phi_{pi} - \alpha_{ki} - 2\pi f_c \tau_{kp}) \}
\end{aligned}$$

Where  $\beta_k$  is the received signal strength as included in equation (2.5),  $\omega_i(t)$  is the current Walsh chip value for the  $k^{th}$  mobile, and  $C_{I_k}$  and  $C_{Q_k}$  represent the current in-phase and quadrature PN chip values which are  $\pm 1$ .

# **CHAPTER THREE**

## **DIGITAL BEAMFORMING AND ANTENNA ARRAY PROCESSING TECHNIQUES**

### **3.1 Introduction**

Beamforming is a classical method of processing temporal sensor array measurements for signal estimation, interference cancellation, estimating source direction, and spectrum estimation. Beamforming is a technique that utilizes an array of sensor elements to focus a receiver channel on a specific signal of interest (SOI) or to transmit signal in a specified direction. It is a spatial filtering used to distinguish the spatial properties between a SOI and the noise and interference signals. Digital beamforming consists of the spatial filtering of a signal where the phase shifting, amplitude scaling, and summer circuit implemented to obtain the desired signal.

Beamforming has been extensively used in wireless systems that utilize a fixed set of antenna elements in an array [1]. Allowing for receive beamforming in uplink transmissions, the signals from these antenna elements are combined to form a movable beam pattern that can be steered to a desired direction that tracks mobile stations (MS) as they move. This allows the antenna system to focus radio frequency (RF) resources on a particular mobile station and minimize the interference [14]. When beamforming is used at the mobile station, the transmit beam pattern can be adjusted to reduce the interference to unintended receivers. At a base station, receive beamforming for each desired user could be implemented separately without affecting the performance of other links [15].

### **3.2 Fundamentals of Digital Beamforming**

Digital beamforming is based on capturing the RF signals and converting them into two streams of binary baseband I and Q signals which jointly represent the amplitudes and phases of signals received at the elements of the array as discussed in chapter one above. In

DBF the direction of beam can be realized by forming an array with a number of elemental radiators. As the directivity of the antenna increases, the gain also increases.

The increase in directivity means that the antenna receives less interference from signals that are not along its path of maximum gain or directivity.

### 3.2.1 Beamforming weight vector

As discussed before, a beamformer is a spatial signal processor, which produces an output with an emphasized desired signal compared to the input, which is the received signal at the array elements. It accomplishes this by applying a complex beamforming weight vector  $w = [w_1 w_2 w_3 \dots w_N]^T$  to the input signal vector:

$$y(k) = \sum_{i=1}^N w_i x_i(k) \quad (3.1)$$

where  $i$  is array element index, and  $k$  is the time index of the received signal sample being considered [4].

## 3.3 Antenna array processing

An antenna array system is composed of a collection of spatially separated antenna elements. The antenna array system has the ability to dynamically adjust the combining mechanism in order to improve system performance as presented in the following subsections.

### 3.3.1 Uniform Linear Antenna Array

In uniform linear array the spacing between the elements are equal. Figure 3.1 below shows a  $K$  element uniform linear array (ULA). The inter-element spacing between the arrays is  $d$  and a plane wave arrives at the array from a direction  $\theta$  with respect to the array normal and the wave front impinging on the first element travels an additional  $d \sin \theta$  distance to arrive at the second element. By setting the phase of the signal at the origin, the phase lead of the signal at element  $k$  relative to that at element 0 is  $\kappa k d \sin \theta$ , where  $\kappa = \frac{2\pi}{\lambda}$ , where  $\kappa$  is

propagation constant in free space and  $\lambda$  is wave length. Adding the entire component outputs together gives [4] array factor F:

$$F(\theta) = V_0 + V_1 e^{j\kappa d \sin \theta} + V_2 e^{j2\kappa d \sin \theta} + \dots = \sum_{k=0}^{K-1} V_k e^{j\kappa K d \sin \theta} \quad (3.2)$$

in terms of vector inner product:  $F(\theta) = V^T v$

Where  $V = [V_0 V_1 \dots V_{K-1}]^T$  the weight vector and  $v = \begin{bmatrix} 1 e^{j\kappa d \sin \theta} \dots e^{j(K-1)\kappa d \sin \theta} \end{bmatrix}^T$  is the array propagation vector that contains the information on the angle of arrival of the signal.

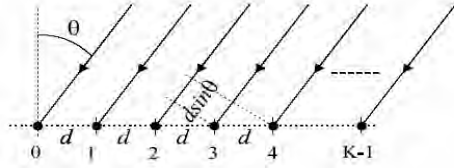


Figure 3.1 Uniform spaced linear antenna array

### 3.3.2 Circular Antenna Array

A circular array consisting of  $L$  identical isotropic elements evenly spaced in a circle of radius  $R$ . Each element is weighted with complex weight  $V_l$  for  $l = 0, 1, \dots, L-1$ . Since the  $L$  elements are equally spaced around the circle of radius  $R$ , the azimuth angle of the  $l^{th}$  element is given as  $\phi_l = 2l\pi / L$ . If a plane wave impinges upon the array in the direction of  $(\theta, \phi)$  in the coordinate system, the relative phase at the  $l^{th}$  element with respect to the center of the array is given as [27]

$$\beta_l = -\kappa R \cos(\phi - \phi_l) \sin \theta \quad (3.3)$$

### 3.3.3 Planar Antenna Array

In addition to placing elements along a line to form a linear array, one can position them on a plane to form a planar array. Planar arrays provide additional variables which can be used to control and shape the array's beam pattern. The main beam of the array can be steered towards any point in its half space [4], [27]. One of the common configurations of planar arrays is the rectangular array, where the elements are placed along a rectangular grid.

### 3.4 Digital Beamforming (DBF)

An antenna can be considered to be a device that converts spatiotemporal signals into temporal signals, thereby making them available to a wide variety of signal processing techniques. In this way all of the desired information that is being carried by these signals can be extracted.

A major advantage of digital beamforming lies in the fact that once the RF information is captured in the form of a digital stream, the conversion of the RF signal at each antenna elements into two streams of binary baseband signals representing I and Q channels. The digital baseband signals then represent the amplitudes and phases of signal received at each element of the array. The process of beamforming involves weighting these digital signals, thereby adjusting their amplitudes and phases such that when added together they form the desired beam. It is possible to take a weighted sum of the antenna element outputs which maximize power in a given direction by implementing a multi-element antenna array at receiving end.

#### 3.4.1 Element-space beamforming

The process represented by equation (3.1) is referred to as element-space beamforming, where the data signals  $X_k$ ,  $k=0, 1 \dots K-1$ , from the array elements is directly multiplied by set of weights to form a beam at a desired angle. By multiplying the data signals with different sets of weights, it is possible to form a set of beams with pointing angles directed anywhere in the field defined by the elements used in the array. The setup for generating an arbitrary number of simultaneous beams from  $K$  antenna elements is shown in figure 3.2. By selecting appropriate values for the weighting vectors, one can implement beam steering, adaptive nulling and beam shaping [18].

$$y(\theta_i) = \sum_{k=0}^{K-1} w_k^{i*} x_k \quad (3.4)$$

Where  $y(\theta_i)$  = output of the beamformer,  $x_k$  = sample from the  $k^{th}$  array element,

$w_k^i$  = weight for forming beam at angle  $\theta_i$

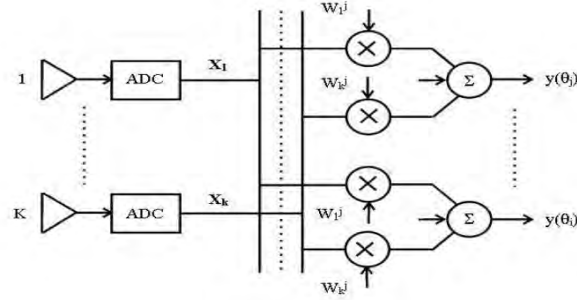


Fig. 3.2– Element-space digital beamformer

### 3.4.2 Beam-space beamforming

The output signals from the array elements can be processed by a multiple-beam beamformer to form a suite of orthogonal beams. The output of each beam can then be weighted and the result combined to produce a desired output. This process is referred to as beam-space beamforming. Figure 3.3 shows the implementation of a weighted FFT-based beamformer [18]. The output of beam-space digital beamformer is given

$$y_i = w_1^i v(\zeta_k) + w_2^i v(\zeta_{k+2}) = \sum_{m=1}^{M_I} w_m^i v(\zeta_{i(m)}) \quad (3.5)$$

Where  $i(m)$  is the selected beam index  $i(1)=k ; i(2)=k+2$  and  $M_I$  is the number of orthogonal beams that are required to form the  $i^{th}$  desired beam

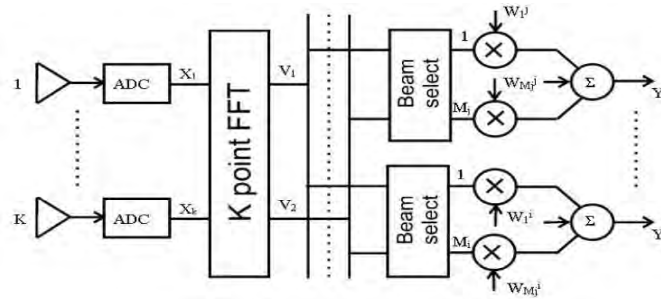


Fig. 3.3 – Beam-space digital beamformer

### 3.4.3 DBF with Multiple Access Schemes

Digital beamforming applied jointly with different multiple access schemes in a cellular communication to enhance system capacity. The multiple schemes such as, frequency division multiple access (FDMA), time division multiple access (TDMA) and code division

multiple access (CDMA) systems. In this work we are only interested in code division multiple access (CDMA), and the reader may refer [4] for FDMA and TDMA schemes.

#### 3.4.4 Digital Beamforming with CDMA

CDMA systems employ the spread-spectrum modulation; where, each user's digital waveform is spread over the entire frequency spectrum that is reserved for all users of the network [4]. The sender of the signal can be identified since the transmitted signal is modulated with a unique code. The entire CDMA frequency band is covered by a continuous beamforming. Implementing synchronous or asynchronous CDMA system affects the chosen configuration of DBF. Synchronous CDMA system is characterized by time aligning the information bit duration for each user at the BS, whereas in asynchronous CDMA system, there is no any user signal synchronization process [17].

#### 3.5 Narrowband Beamforming

A signal is considered to be narrowband if all the frequency components of the signal as they travel across the array undergo only a phase shift and not a change in the magnitude. In other words, the signal bandwidth is very small compared to the carrier frequency. If the signal bandwidth is very small compared to the carrier frequency then the phase shifts undergone by the frequency components at the two edges of the band are almost equal. For a narrowband signal, narrowband array processing is the appropriate choice [3]. A narrowband array has one weight per element followed by a linear combiner.

$$y(t) = w^H(t)u(t) \quad (3.6)$$

where superscript  $H$  denotes the Hermitian (complex conjugate) transpose and, is known as the beamformer weight vector.

$$w(t) = \begin{bmatrix} w_0(t) \\ w_1(t) \\ \vdots \\ w_{M-1}(t) \end{bmatrix} \quad (3.7)$$

### 3.6 Wideband Beamforming

A signal is considered to be wideband if all the frequency components of the signal as they travel across the array undergo a phase shift with a change in the magnitude. In other words, the signal bandwidth is not very small compared to the carrier frequency. If the signal bandwidth is not very small compared to the carrier frequency then the phase shifts undergone by the frequency components at the two edges of the band are not equal. Therefore the phase shift line appears not to be flat across the bandwidth. For a wideband signal, wideband array processing is the appropriate choice [33].

Then, the output for a wideband adaptive beamformer may be expressed as,

$$y(t) = \bar{w}^H(t) \bar{u}(t) \quad (3.8)$$

which is identical in form to the array output of the narrowband array. The wideband beamformer is more complex than the narrowband beamformer.

### 3.7 Switched Beamforming

Switched beamforming provides non-uniform coverage when the MS is moving from one beam to another [33], there might be a call drop due to no coverage zone in between two beams. In addition when the MS moves from one beam to another intra-cell handover takes place, the frequency of which directly depends on the beam width. Adaptive beamforming overcome these two problems, as the beam follows the MSs to where ever they move. It is the simpler approach where the direction of arrival (DOA) of signal is detected in that direction by multiplying pre-computed complex vector. These algorithms such as: MUSIC and ESPRIT algorithms. These algorithms restrict their applicability in a wireless mobile communications because they cannot perform steer nulls and suppressing [28] strong interference.

### 3.8 Adaptive Beamforming

Adaptive Beamforming is a technique in which an array of antennas is exploited to achieve maximum reception in a specified direction by estimating the signal arrival from a desired direction while signals of the same frequency from other directions are rejected [4]. This is achieved by varying the weights of each of the sensors (antennas) used in the array. It

basically uses the idea that, though the signals emanating from different transmitters occupy the same frequency channel, they still arrive from different directions. This spatial separation is exploited to separate the desired signal from the interfering signals. In adaptive beamforming the optimum weights are iteratively computed using complex algorithms based upon different criteria. Most of the algorithms are concerned with the maximization of the signal to noise ratio (SNR).

### **3.8.1 Adaptation Criteria**

The optimum weights using different criteria are all given by the solution of the Wiener equation. These optimum criteria are as valid as in communications as in other applications. Though different forms of criteria appear in the [4], it can be shown that they all stem from the wiener-Hopf equation. This is because the wiener solution provides the upper limit on the theoretical adaptive beamforming steady-state performance. Some of the most frequently used performance criteria are the Mean Square Error (MSE), the Maximum Likelihood (ML), Maximum Signal to Noise Ratio (MSNR) and Maximum Signal to Interference Noise Ratio (SINR), [33]. These performance criteria are usually expressed as cost functions and the weights are adapted iteratively until the cost functions converge to a minimum value. Once the cost function is minimized we can be assured that the performance criterion is met and the algorithm is said to have converged. There are several factors that are to be considered while choosing an adaptive algorithm like the convergence rate of the algorithm, complexity and robustness. The convergence rate is the number of iterations required for the convergence of the algorithm. In mobile environments, convergence rate is important to converge to optimum before the channel conditions change [34].

### **3.8.2 Adaptive Beamforming for Uplink**

Adaptive beamforming for reverse link has studied by several researchers [4], [5], [7]. Adaptive beamforming is traditionally used for reception in remote sensing, radar, and sonar systems. The spatial channel information is available on the uplink, thereby making it much easier to develop beamforming techniques and algorithms to deal with multipath and co-channel interference.

### 3.9 Adaptive Beamforming Algorithms

The basic adaptive algorithms that have been investigated for beamforming in mobile communications include the trained and blind adaptive beamforming algorithms as classified in Figure 3.4 under adaptive array (beamforming) algorithms[4, 34]. Adaptive beamformers have additional capability to steer nulls in the direction of interfering signals. An adaptive beamformer comprised of  $N$  antenna elements can effectively reject  $N-1$  interfering signals [33].

Non-blind adaptive algorithms need statistical knowledge of the transmitted signal in order to converge to a weight solution. This is typically accomplished through the use of a pilot training sequence sent over the channel to the receiver to identify the desired user. On the other hand, blind adaptive algorithms do not need any training [4]. They attempt to restore some type of characteristic of the transmitted signal in order to separate it from other users in the surrounding environment.

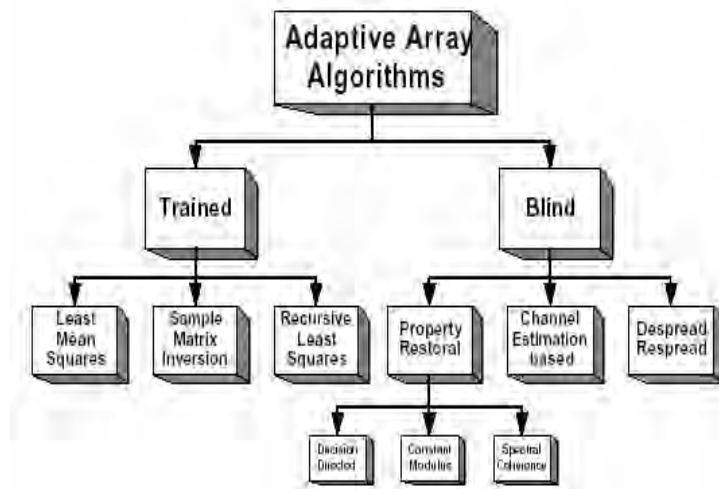


Figure 3.4 – Classification of adaptive array algorithms

A basic adaptive beamformer is shown in Figure 3.5. The weight vector  $\mathbf{w}$  is calculated using the statistics of signal  $X_N(k)$  arriving from the antenna array.

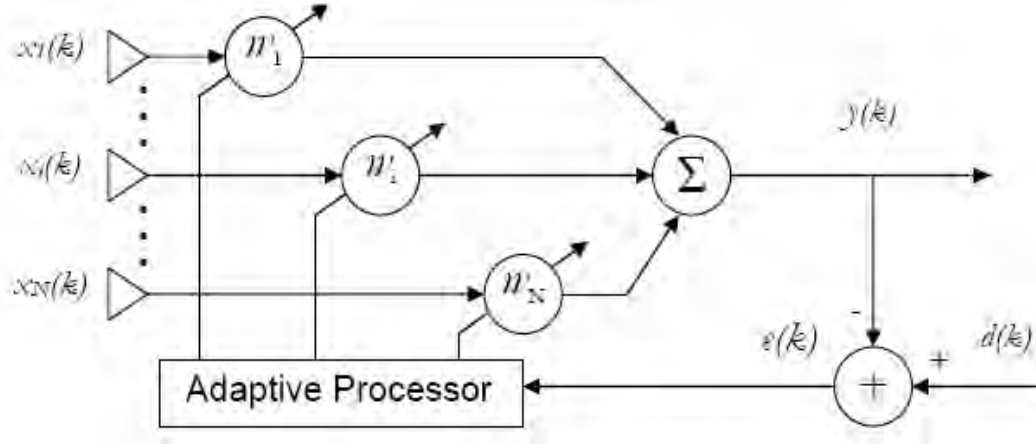


Figure 3.5 adaptive beamforming configuration

Through a feedback loop the weights,  $w_1, \dots, w_N$  are updated by the time sampled error signal:

$$e(k) = d(k) - y(k) \quad (3.9)$$

where: The training sequence,  $d(k)$ , is the desired signal and  $y(k)$  is the output of the adaptive array

$$y(k) = w^H x(k) \quad (3.10)$$

The feedback system attempts to direct the weights at each element to their optimal weights,  $w_{opt}$ . The adaptive processor adjusts the weight vector to minimize the mean square error (MSE) of the error signal,  $e(k)$ , given by:

$$E[e(k)^2] = E[d(k) - y(k)]^2 \quad (3.11)$$

The output of the array  $y(k)$  with variable element weights is the weighted sum of the received signals at the array elements and the noise at the receivers connected to each element. The weights are iteratively computed based on the array output, a reference signal that approximates the desired signal, and previous weights. The reference signal  $d(k)$  is approximated to the desired signal using a training sequence or a spreading code, which is known at the receiver. The format of the reference signal varies and depends upon the system where adaptive beamforming is implemented. The reference signal usually has a good correlation with the desired signal and the degree of correlation influences the accuracy and the convergence of the algorithm [21, 22].

### 3.9.1 Trained Adaptive Algorithms

Trained adaptive beamforming algorithms use a finite set of training symbols to adapt the weights of the array and maximize the SINR. First, a training signal, which is known to both the transmitter and receiver, is transmitted by the transmitter. The beamformer in the receiver uses the information of the training signal to compute the optimal weight vector. After the training period, data is sent and the beamformer uses the weight vector computed previously to process the receiver signal. The trained adaptive algorithms drawback is the excessive utilization of transmission time and wastage of bandwidth. The trained adaptive algorithms are categorized based on their adaptation criteria and they are the LMS, SMI and RLS methods [4, 21, 33]. For simulation study we use LMS from the trained adaptive algorithm.

#### 3.9.1.1 The LMS algorithm

The LMS algorithm can be considered as the most common adaptive algorithm for continuous adaptation. It uses the steepest-descent method and recursively computes and updates the weight vector. Due to the steepest-descent the updated vector will propagate to the vector which causes the least mean square error (MSE) between the beamformer output and the reference signal. The LMS algorithm is simple to implement. But the dynamic range over which it operates is quite limited. Since the received signals in a mobile radio system vary by more than 20dB [19], power control is required if the LMS algorithm is to be used. The normalized LMS algorithm can be used to overcome the dynamic range limitation. The following derivation for the LMS algorithm is found in [22]. The MSE is defined by:

$$\epsilon^2(k) = [d^*(k) - w^H x(k)]^2 \quad (3.12)$$

$d^*(k)$  is the complex conjugate of the desired signal. The signal  $x(k)$  is the received signal from the antenna elements, and  $w^H x(k)$  is the output of the beamformer antenna and H is the Hermetian operator. The expected value of both sides leads to [19, 22]:

$$E\{\epsilon^2(k)\} = E\{d^2(k)\} - 2w^H r + w^H R w \quad (3.13)$$

In this relation the r and R are defined as [26, 27]:

$$r = E\{d^*(k)x(k)\} \quad (3.14)$$

$$R = E\{x(k)x(k)^H\} \quad (3.15)$$

$R$  is referred to as the covariance matrix and  $r$  is the cross-correlation. If the gradient of the weight vector  $w$  is zero, the MSE is at its minimum [22]. This leads to:

$$\nabla_w(E\{\varepsilon^2(k)\}) = -2r + 2Rw = 0 \quad (3.16)$$

The solution of (3.16) is called the Wiener-Hopf equation for the optimum Wiener solution:

$$w_{opt} = R^{-1}r \quad (3.17)$$

The LMS algorithm converges to this optimum Wiener solution. The basic iteration is based on the following simple recursive relation [19, 22]:

$$w(k+1) = w(k) + 1/2\mu(-\nabla(E\{\varepsilon^2\})) \quad (3.18)$$

combining (3.16) with (3.18) gives:

$$w(k+1) = w(k) + \mu(r - Rw(k)) \quad (3.19)$$

The measurement of the gradient vector is not possible, and therefore the instantaneous estimate is used as defined by (3.20) and (3.21).

$$\hat{R}(k) = x(k)x^H(k) \quad (3.20)$$

$$\hat{r}(k) = d^*(k)x(k) \quad (3.21)$$

By rewriting (3.19) using the instantaneous estimates, the LMS algorithm can be written in its final form (3.22).

$$\begin{aligned} \hat{w}(k+1) &= \hat{w}(k) + \mu x(k) \left( d^*(k) - x^H(k) \hat{w}(k) \right) \\ &= \hat{w}(k) + \mu x(k) \varepsilon^*(k) \end{aligned} \quad (3.22)$$

One of the issues on the use of the instantaneous error is concerned with the gradient vector, which is not the true error gradient. The gradient is stochastic and therefore the estimated vector will never be the optimum solution. The LMS cannot filter the system noise, as it is not correlated for all given antennas. The interferers are cancelled by placing nulls in the direction of the interferers. The LMS algorithm can be the best choice for systems with a relatively small number of antenna elements.

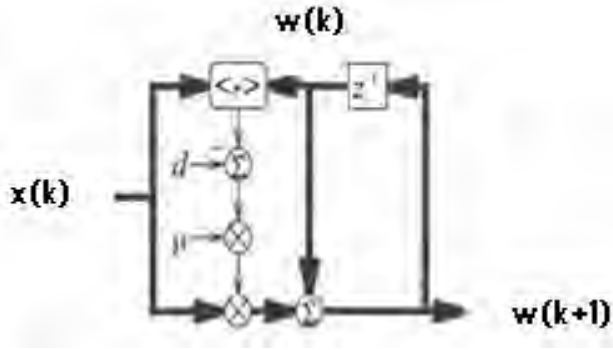


Figure 3.6 LMS algorithm Signal-flow graph

### 3.9.1.2 Recursive Least Square Algorithms

RLS is a recursive version of the LS approach, where the inversion of the matrix is carried out using a recursion. An important feature of RLS algorithm is that the inversion of the covariance matrix is replaced at each step by a simple scalar division. This feature reduces the computational complexity while maintaining a similar performance. For a fast fading channel, a value slightly below unity should be chosen. A value of 0.95 was reported to be reasonable [20].

RLS requires computations of the order of  $M^2$  where  $M$  is the number of antenna array elements. RLS operates on a sample-by-sample basis and hence to process  $N$  samples, RLS requires computations on the order of  $M^2 XN$  operations. In contrast to the LMS algorithm, a RLS adaptive algorithm approximates the Wiener solution directly [16]. The RLS algorithm uses weighted sums for estimating  $R_{xx}$  and  $r_{xd}$  using the following equations,

$$\hat{R}_{xx} = \sum_{i=1}^N \gamma^{n-1} x(i) x^H(i) \quad (3.23)$$

$$\hat{r}_{xd} = \sum_{i=1}^N \gamma^{n-1} d(i)^* x(i) \quad (3.24)$$

The matrix inversion obtained recursively and the weight update equation is given in[4]

$$w(n) = w(n-1) + q(n)[d^*(n) - w^H(n-1)x(n)] \quad (3.25)$$

Where

$$q(n) = \frac{\gamma^{-1} R_{xx}^{-1}(n-1)x(n)}{1 + \gamma^{-1} x^H(n) R_{xx}^{-1}(n-1)x(n)} \quad (3.26)$$

Where

$$R_{xx}^{-1}(n) = \gamma^{-1}[R_{xx}^{-1}(n-1) - q(n)x(n)R_{xx}^{-1}(n-1)] \quad (3.27)$$

Recalling that the statistically optimum Wiener solution is

$$w = R_{xx}^{-1}r_{xd} \quad (3.28)$$

The RLS algorithm converges faster than the LMS algorithm but at the expense of computational complexity. RLS requires an initial estimate of  $R_{xx}^{-1}$  matrix and a reference signal.

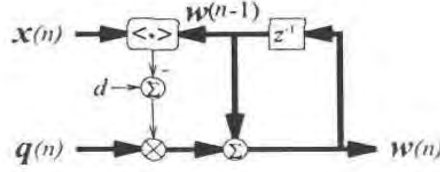


Figure 3.7 Signal-flow graph of the RLS algorithm.

### 3.9.2 Blind Adaptive Algorithms

Adding a known training sequence prior to data transmission leads to additional overhead, which limits the channel bandwidth available for data. The techniques generally referred to as blind adaptive algorithms adapt by attempting to restore some known property of the received signal. It uses statistical information of the signal. Blind algorithms can be classified as property restoral algorithms, channel estimation algorithms and despreads and respread algorithms. The algorithms have been studied by a number of authors [4, 24, 25].

#### 3.9.2.1 Constant Modulus Algorithm (CMA)

The CMA gives a single beamformer vector. This is sufficient for blind equalization applications, where the received signal consists of several temporal shifts of the same CM signal, and we do not have to recover all of them [35]. The CM algorithm is applicable to constant modulus signals and adjusts the weight vector of the adaptive array to minimize the variation of the envelope at the output of the array. After the algorithm converges, the array can steer a beam in the direction of the signal of interest (SOI), and nulls in the directions of the interference. The only disadvantage of CM algorithm is that it converges to the strongest user in the channel. Godard was the first to make use of constant modulus algorithm (CMA) property to carry out blind equalization in two-dimensional digital communications systems [23]. The out put of a DBF array is given as:

$$y(k) = w^H(k)x(k) \quad (3.29)$$

Assuming that the transmitted signal  $S(k)$  has a constant envelope, the array output  $y(k)$  should have a constant envelope as well. This can be accomplished by adjusting the array weight vector  $w$  in such a way as to minimize the cost function, which measures the signal modulus variation and is given in a general sense by

$$\mathcal{E} = D\{f[S(k)], f[y(k)]\} \quad (3.30)$$

Where  $D$  and  $f$  are some defined specific distance metrics. In particular, the updated value of the weight vector at time  $k+1$  is computed by using the simple recursive relation

$$w(k+1) = w(k) + (1/2)\mu[-\nabla(E\{\mathcal{E}(k)\})] \quad (3.31)$$

Where the gradient  $\nabla(E\{\mathcal{E}(k)\})$  can be found as:

$$\nabla(E\{\mathcal{E}(k)\}) = -E\{[r_p(k) - |y(k)|^2]y(k)x(k)\} \quad (3.32)$$

In reality, an exact measurement of the gradient vector is not possible, since this would require a prior knowledge of  $E[|S(k)|^p]$ . The strategy is to scale  $|S(k)|$  to unity and to use the instantaneous estimate,

$$\nabla(E\{\mathcal{E}(k)\}) \cong -[1 - |y(k)|^2]y(k)x(k) \quad (3.33)$$

The weight vector can then updated as

$$w(k+1) = w(k) + \mu [1 - |y(k)|^2]y(k)x(k) \quad (3.34)$$

The error is different and defined by [4] as:

$$\mathcal{E}(k) = (1 - |y(k)|^2)y^*(k) \quad (3.35)$$

The CM algorithm can be found in many derived forms. The error function for a derived version is given as:

$$\mathcal{E}(k) = \frac{y(k)}{|y(k)|} - y(k) \quad (3.36)$$

We can express the updating equation as

$$w(k+1) = w(k) + \mu x(k)\mathcal{E}^*(k) \quad (3.37)$$

The advantage of the CM algorithm is the fact that it only needs the instantaneous amplitude of the array output  $|y(k)|$  and therefore no synchronization is required. Due to this property, the CM algorithm is relatively simple to implement.

### **3.9.2.2 Decision-Directed Algorithm (DDA)**

In the decision- directed algorithm, the tab weights of the adaptive equalizer are adjusted for an adaptive process and widely used for adaptive equalization to combat intersymbol interference (ISI) in digital communications. Although the decision-directed algorithm widely used in adaptive equalization, it is only applied to diversity combining and beamforming [4].The decision-directed beamforming for wireless communications has been studied by a number of authors [29], [30].

## **CHAPTER FOUR**

### **COVERAGE AND CELL CAPACITY ESTIMATION IN DBF**

#### **4.1 Introduction**

The fundamental capability of the antenna array that is of most interest to the wireless is the ability to improve signal reception, the ability to estimate the angle of arrival of signals of interest [17]. The ability to improve signal reception can be used to improve various aspects of the wireless network.

#### **4.2 Coverage**

The antenna array is used to increase the percentage of the area of the cell that will have an acceptable signal level which is referred to as coverage. In general, for low bit rate services such as voice and for small cell loading (small throughput situations), the uplink is the limiting link for coverage. The reason is mainly due to the limitation of the mobile terminal transmit power. This is despite other factors in favor of the uplink, such as receive antenna diversity and better receiver design available at the base station [3]. When radio energy propagates in a cellular environment, the received signal level degrades as the distance between transmitter and receiver increases. This received signal has to exceed the inherent noise level in the radio receiver by a certain margin in order to be successfully demodulated. The ratio of the received signal level and this noise is called the signal-to-noise ratio (SNR). Cellular systems are deployed with a certain “average” or nominal SNR target. CDMA systems operate at these much lower SNRs by introducing large redundancy into transmitted data through a process known as “spreading.” In practice, the range of a CDMA base station is limited more by interference among the users in the system than by receiver noise.

Adaptive beamforming can increase the cell coverage range significantly through antenna gain and interference rejection as shown under chapter 5. In particular, the coverage range can be improved by a factor of  $M^{1/\gamma}$  for noise limited environments, where  $M$  denotes the number of antenna element and  $\gamma$  is the propagation loss exponent. Generally, there will be fewer sites required with adaptive antennas employed at base stations. A large coverage area can be achieved with base station antennas at a greater height above terrain. However, this

system trade-off between antenna height and coverage area can be eased by using the number of antenna elements as a design parameter [31].

### **4.3 Capacity**

Cell capacity can be defined as the maximum number of users per cell that can be supported with a given level of performance. For voice, this shows a BER of  $10^{-3}$  or less. The two key factors in determining which link is limiting the system capacity are the base station transmit power and uplink target load factor. As we have seen from practical point of view, there is a trade-off between system coverage and capacity; that is, increasing the sector throughput (in other words, the capacity or number of users) shrinks the coverage [33]. Normally, the initial system design is based on the coverage requirement, which is why the initial target uplink load factor is low so that the coverage area is maximized. This would lead to an uplink capacity limited scenario as the maximum uplink load is reached before the base station runs out of transmit power.

The capacity can be improved by frequency reuse, cell sectorization. But in case of CDMA systems, instead of frequency reuse, we implement code reuse having the same fundamental concept. Depending upon the number of antenna elements, propagation environment and the number of dynamic channel allocation, adaptive beamforming increases the number of available voice channel through communication links [4]. The output of adaptive beamformer results an increase in bit rate, due to its SIR improvement. The expected minimum rate improvement is  $10\log M$  dB, which depends on the modulation format under noise limited environment [4].

### **4.4. Maximum signal-to-Interference Noise ratio**

Beamforming uses to maximize the desired signal while minimizing a large number of interfering signals arriving from many directions in the uplink of a CDMA system [7]. This is accomplished through the use of antenna arrays to increase the received SINR results in a reduced BER in digital systems. For voice or video applications the reduced BER yields a corresponding improvement in the received signal quality. In general, for noise limited

channels, the minimum signal thresholds are reduced by  $10\log_{10}M$ , where  $M$  is the number of antenna elements in the array.

When the number of interfering signals is large, the beamformer can not cancel all of them adequately, and can result insignificant gain for the noise component. The solution of this limitation is a statistically optimal maximum SINR beamformer which maximizes the output signal to interference and noise power ratio. This method requires knowledge of both the desired signal's and the interference correlation matrix.

The SINR at the beamformer output is given [35]

$$y = w^H x = w^H (as + \eta) = w^H as + w^H \eta = y_s + y_\eta \quad (4.1)$$

Where  $y_s = w^H (as)$  the desired signal is output and  $y_\eta = w^H \eta$  is the interference output. The average output SINR is then [1]

$$SINR = E \left\{ \frac{|y_s|^2}{|y_\eta|^2} \right\} = E \left\{ \frac{w^H as(as)^H w}{w^H \eta \eta^H w} \right\} = \frac{w^H R_s w}{w^H R_i w} \quad (4.2)$$

Where  $R_s = E\{as(as)^H\}$  and  $R_i = E\{\eta \eta^H\}$ . The optimum weight vector  $w_{opt}$  is the vector which maximizes (4.2).

The relation between the SIR and BER in adaptive antenna array system is given in [4] as,

$$BER = Q(\sqrt{3N_s SIR}) \quad (4.3)$$

$$\text{Where } Q(x) = \frac{1}{2\pi} \int_x^\infty e^{-t^2/2} dt = \frac{1}{2} \operatorname{erfc}\left(\frac{x}{\sqrt{2}}\right) = \frac{1}{2} \operatorname{erfc}\left(\sqrt{\frac{3N_s SIR}{2}}\right), \quad x \geq 0$$

Where  $N_s$  is spreading factor which varies from 4 to 256, for voice  $N_s = 256$  assuming the noise limited environment. The performance measure signal to interference-plus-noise ratio (SINR) optimum beamforming weight are given by Wiener solution [35]

$$x(k) = \underset{\text{signal}}{x_s(k)} + \underset{\text{interference}}{x_i(k)} + \underset{\text{noise}}{x_n(k)} \quad (4.4)$$

## **CHAPTER FIVE**

### **ANALYSIS AND SIMULATION**

#### **5.1 Digital Beamforming Algorithms**

In mobile cellular communication frequency reuse and cell sectorization are widely implemented in order to increase cell capacity and coverage. But due to signal power radiated in directions and throughout the cell area further increment in capacity and coverage is not practically reliable [33]. Thus, Beamforming is one to minimize these constraints. There are mainly two approaches to Beamforming: switched and adaptive beamforming based on their complexity and efficiency.

The simpler one is switched BF where the direction of arrival (DOA) of the signal is detected and corresponding beam is formed in that direction by multiplying complex array factor (AF). The DOA in the switched beamforming can be computed using algorithms such as MUSIC and ESPRIT but these algorithms do not suppressing strong interfering signals [33].

Adaptive beamforming is more complex, but highly efficient where the radiation pattern is constructed dynamically in which interferers are blocked by placing nulls and the beam is formed in the direction of users. Beam steering implemented in the direction of the user as it moves using fully adaptive antenna array. The complex weights are computed by using adaptive algorithms: LMS, RLS, SMI, CMA, and DDA which are discussed under chapter 3 above. Thus, the required pattern is formed by multiplying the computed weight with the signal from the antenna array. In this work, we try to emphasis on LMS and CMA of the beamforming algorithms from the discussed beamforming algorithms, in the simulation study for the performance measurement. The adaptive algorithms LMS & CMA are less complex and simple to implement relative to the others.

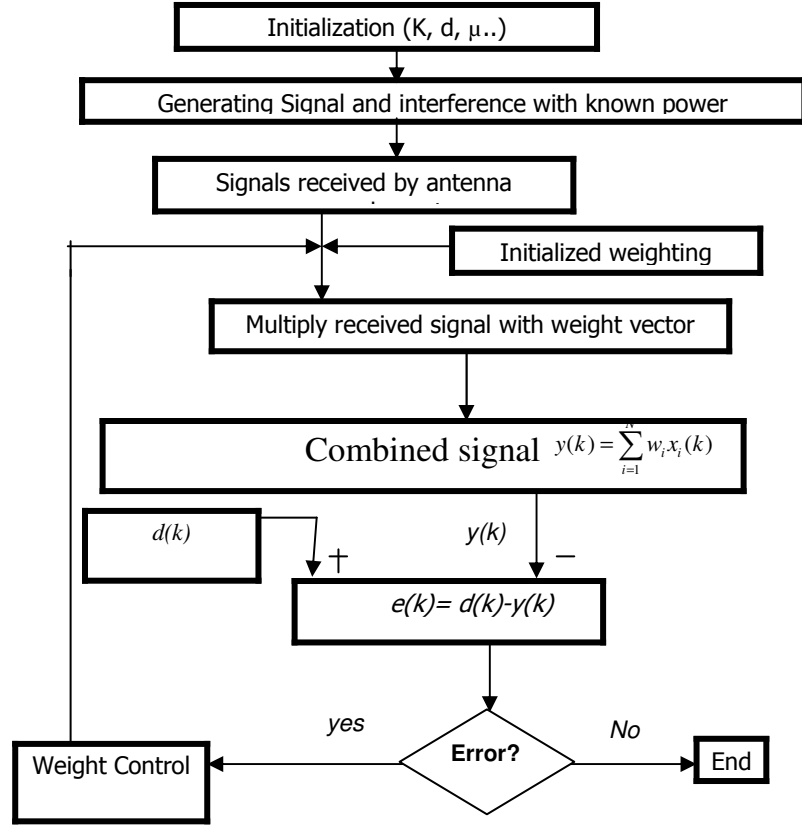


Figure 5.1 Adaptive Beamforming Algorithm Flow chart

## 5.2 System and Traffic Model

In CDMA, although there is no hard limit on the number of mobile users served, there is a practical limit on the number of simultaneous users in a cell to control the interference among users having the same pilot signal. The system capacity bound with acceptable QoS of CDMA systems supporting voice and data traffic in the reverse link is given in many literatures [34]. We assume that antenna array is used at the base station only, and at the mobile station we have a single antenna, which results a channel configuration of multiple input single output as shown in figure 5.1. The service area of a mobile cellular network is assumed to be covered by many identical regular hexagonal cells and mobile stations are uniformly distributed in each cell with a random speed  $V$  which has a probability density function (pdf)  $f_v(v)$ . Mobile stations move randomly in any direction with equal probability over  $[0, 2\pi)$  angle of arrival. In mobile cellular system implementing beamforming algorithm, it is assumed that the bandwidth separation is  $\theta_b$ , which is further assumed as the minimum separation angle between two consecutive mobile stations using the same physical

channel [34]. The inter-antenna elements spacing ( $d$ ) is in terms of signal wavelength ( $\lambda$ ) and we use few number of array element antenna: 4 and 6 to emphasis on some of the beamforming algorithm effective with few number of antenna elements..

### 5.3 Simulation model and parameters

In order to achieve the goal of this thesis work, a model of adaptive beamforming system is created and the performance of the system is analyzed using system model of figure 5.2. In adaptive beamforming system, the BS can form multiple independent beams to serve the users.

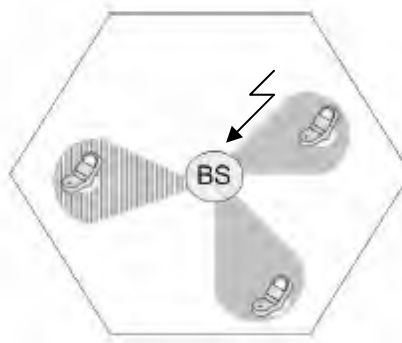


Figure 5.2 System model

The simulation has been performed using MATLAB codes. From the Beamforming algorithm discussed above we choose for the simulation study only two of them from adaptive algorithm: LMS and CMA. The simulating codes have loops to update the weight vectors that are used to estimate the desired signal. The parameters used in this simulation study are: antenna noise figure (base station system noise), the step size ( $\mu$ ), antenna spacing ( $d$ ), and antenna array element. The efficiency of these algorithms is dependent on the selected values for the above parameters. In addition to these parameters there are other parameters which are considered to be fixed with constant values. Mainly the simulation model for these two algorithms looks for specific system model and analysis model as discussed in the next sub sections. The error is a result of the extra 'system' noise that is added to all antennas. The interference signals are Gaussian white noise, zero mean with a standard deviation of 1.

The following assumption is considered to simplify the simulation environment:

- ❖ Perfect power control.
- ❖ A beam formed by the array can be steered in any direction in the azimuth plane.

Table 5.1 simulation parameters

|                                        |                                              |
|----------------------------------------|----------------------------------------------|
| Inter- antenna Spacing (d)             | $0.25\lambda$ , $0.5\lambda$ , and $\lambda$ |
| Step size ( $\mu$ )                    | 0.1, 0.01 and 0.001                          |
| Number of antenna array elements       | 4 and 6                                      |
| System noise (antenna)                 | 0.1 and 0.01,                                |
| No of interference in the system       | 2, 3, 4 and 5                                |
| Desired signal AOA                     | 20 degree                                    |
| Steering angles for null interferences | -82, -40, -19, -10, 0, and 40 (degree)       |
| Bit rate                               | 100                                          |
| No of samples (iteration time)         | 210                                          |
| System band width (W)                  | 1.25MHz                                      |
| Bit-error rate (BER)                   | $10^{-3}$                                    |

## 5.4 Simulation results and discussion

### 5.4.1 LMS Adaptive Beamforming Algorithm

The LMS is discussed in subsection 3.9.1.1; an adaptive array is simulated in MATLAB by using the LMS algorithm explained above. When an array of 6 element antennas are used, there is a maximum of 5 nulls that can eliminate the interferer and when 4 element antennas are used, there is maximum of 3 nulls that can reject interferers. Synchronization of the LMS algorithm usually involves the use of correlators in the digital beamformer to align the desired vector with the incoming signal. The interferers are cancelled by placing nulls in the direction of the interferers.

Figure 5.3 below is the simulation result obtained using 6 antenna elements and assuming five interfering signals. The inter-antenna spacing is equal to half wavelength. From this figure one can observe that, LMS adaptive beamforming weighted signal  $|y|$  starts from

zero and estimate the normalized desire signal magnitude. The error amplitude also is minimized as the number of iteration (samples) becomes greater. It means LMS error rapidly decreases up to 60 samples and then very slowly. The reason why the LMS error start from one and weighted signal from zero is that, trained adaptive algorithm uses reference signals and initialized the weight vector to zero.

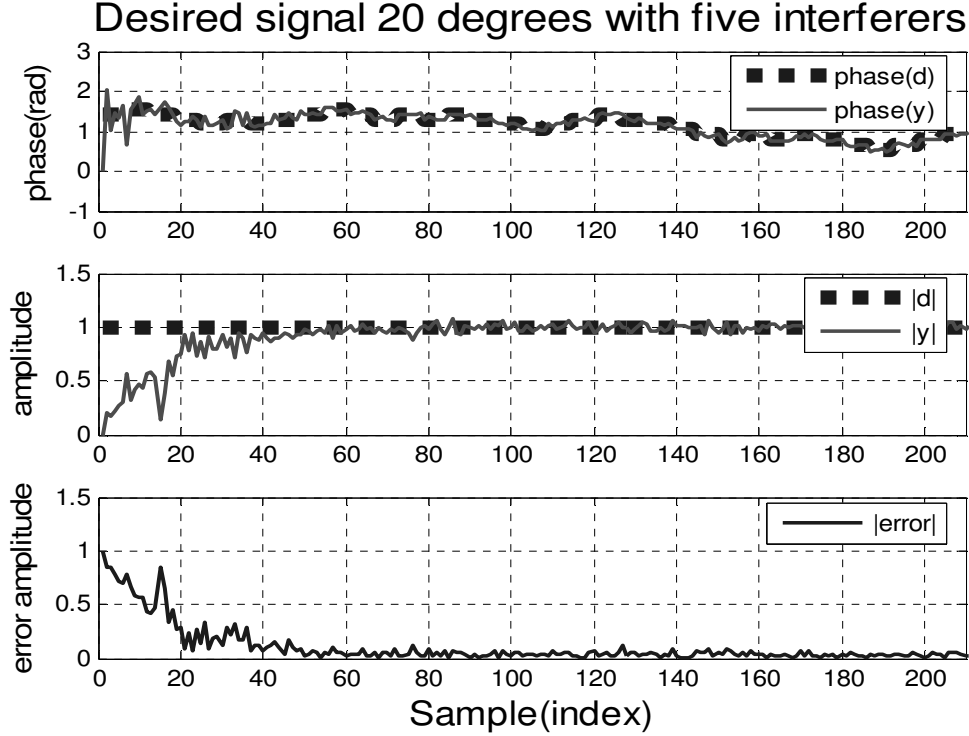


Figure 5.3 *LMS algorithm for an adaptive array with 6-element antenna and 5 interferers for antenna noise figure equal to 0.1,  $\mu=0.01$ , and  $d=0.5\lambda$ , Amplitude and Phase estimation versus time of iteration.*

Figure 5.4 is similar with 5.3 but their difference lies on the simulation parameter value. Comparing them, we can observe that figure 5.4 is more efficient in estimation of the desired signal after some samples and it highly minimize LMS error. In this simulation study we use the system (antenna) noise figure smaller than in case of figure 5.3. This shows it has a direct effect on the performance of beamforming algorithms and that is discussed under table 5.1 why we choice it as simulation parameter.

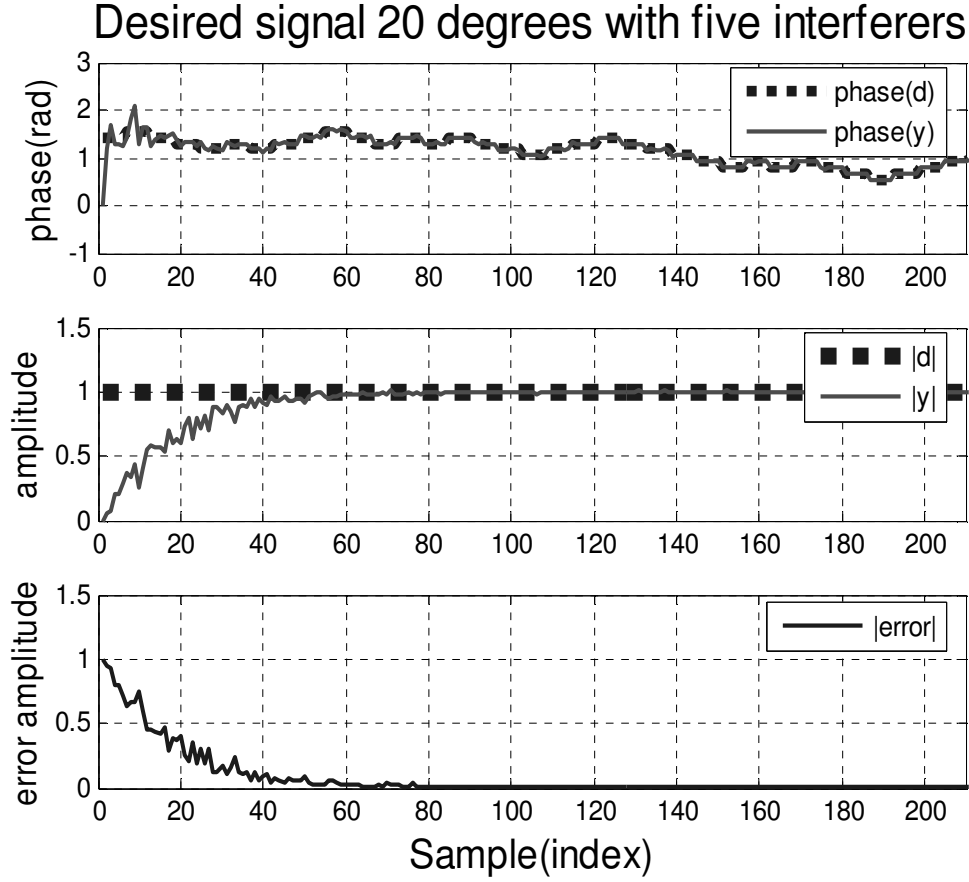


Figure 5.4 LMS algorithm for an adaptive array with 6-element antenna and 5 interferers for antenna noise figure equal to 0.01,  $\mu=0.01$ , and  $d=0.5\lambda$ , Amplitude and Phase estimation versus time of iteration.

Figure 5.5 below is the array factor versus angle. From this figure we see that the desired signal is steered at 20 degree and we have five nulls where strong interfering signals steered to be rejected. As shown by the simulation result, the desired signal properly received with array factor of 0dB. The simulation performed assuming one desired signal and 5 interfering signals, this 6 antenna arrays can reject 5 strong interfering signals. This simulation result shows that the LMS algorithm is able to iteratively update the weights to force deep nulls at the direction of the interferers and achieve maximum beam in the direction of the desired signal. The nulls can be seen deep at below -45dB level from the maximum observed at 20 degree.

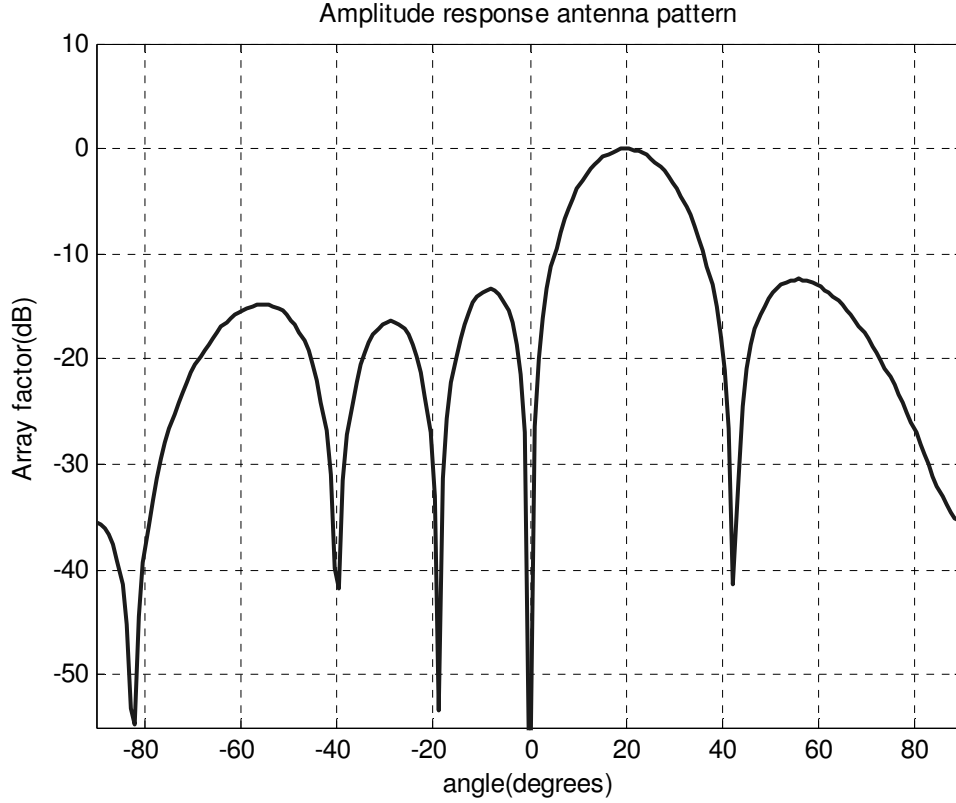
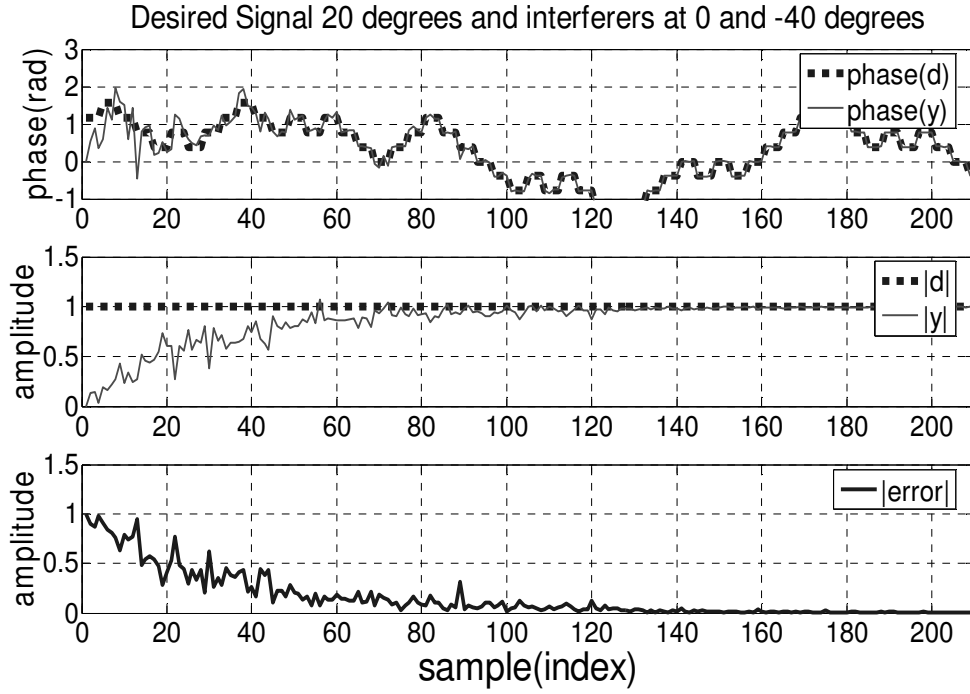
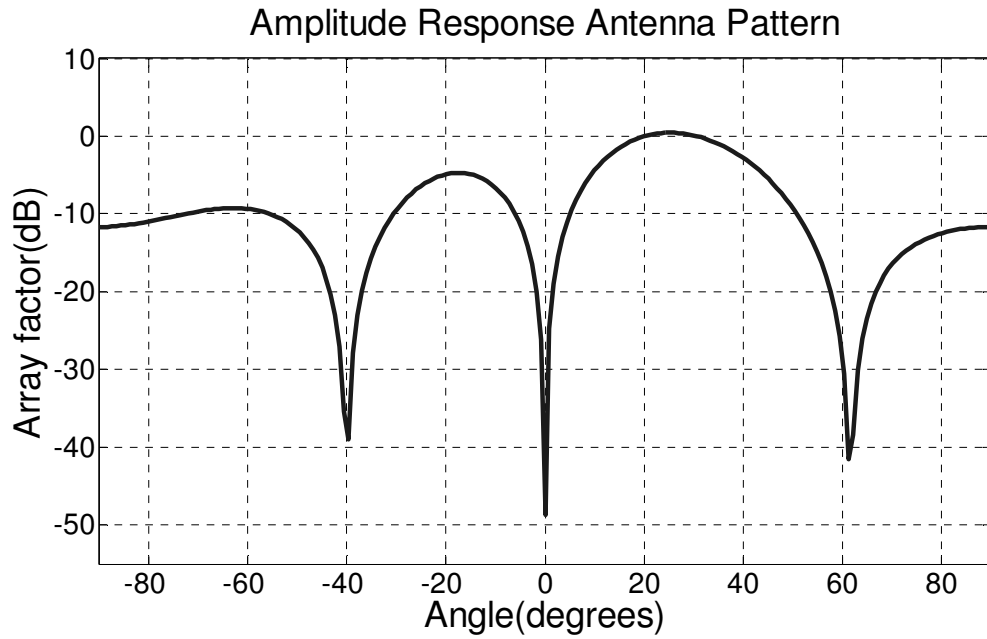


Figure 5.5 Array factor for LMS algorithm for an adaptive array with 6-element antenna and 5 interferers for antenna noise figure equal to 0.01,  $\mu=0.01$ , and  $d=0.5\lambda$ . Array factor (dB) versus theta (degrees)

The simulation result given below under 5.6 (a) and (b) is using the same simulating parameter with the result discussed above by figure 5.4 and 5.5, but they are differing in number of antenna used. Here we use 4-element antenna to maximally reject three interfering signals. The weighted signal estimate the desired signal efficiently after 60 samples as given by figure 5.4 where as the estimation is attain best result after 120 samples when 4 antenna elements are used as shown by figure 5.6(a) below. Thus, from this two simulation results as the number of antenna element at base station increase, the LMS algorithm for digital beamforming performance increases. In other words, adaptation of the LMS algorithm is very high and converges rapidly. As the number of antenna array element increases the width of the main lobe decreases, the number of the side lobe increases and the number of nulls in the pattern increases. Therefore, using more antenna at base station maximize the performance of digital beamforming by rejecting more interferences which maximize the SINR.



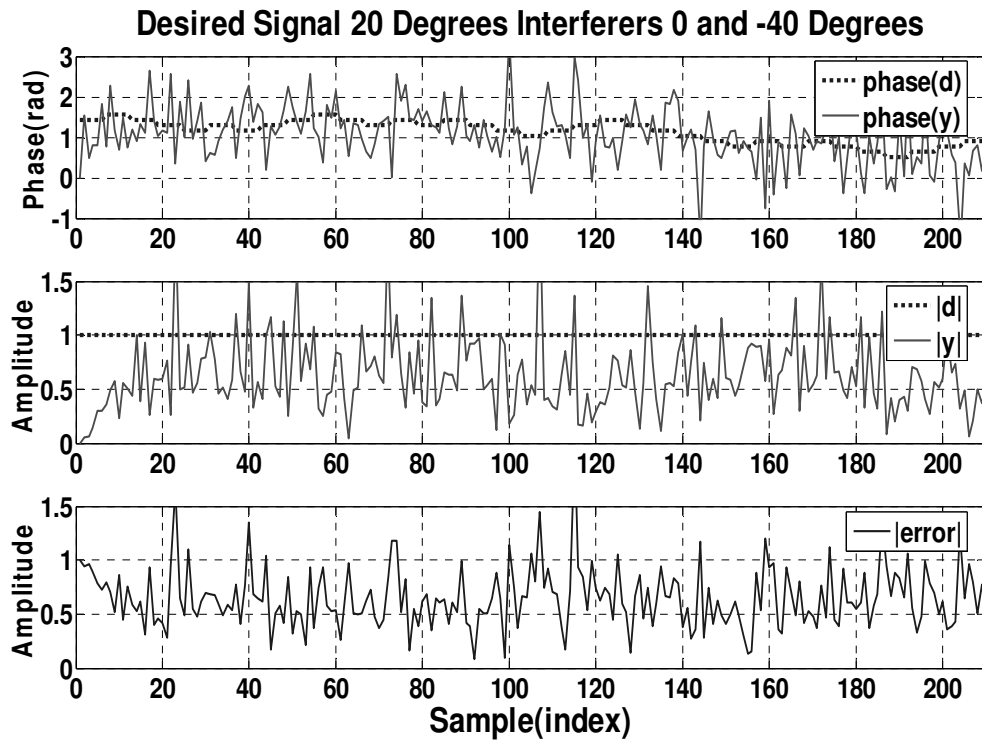
(a) Amplitude and Phase estimation versus time of iteration



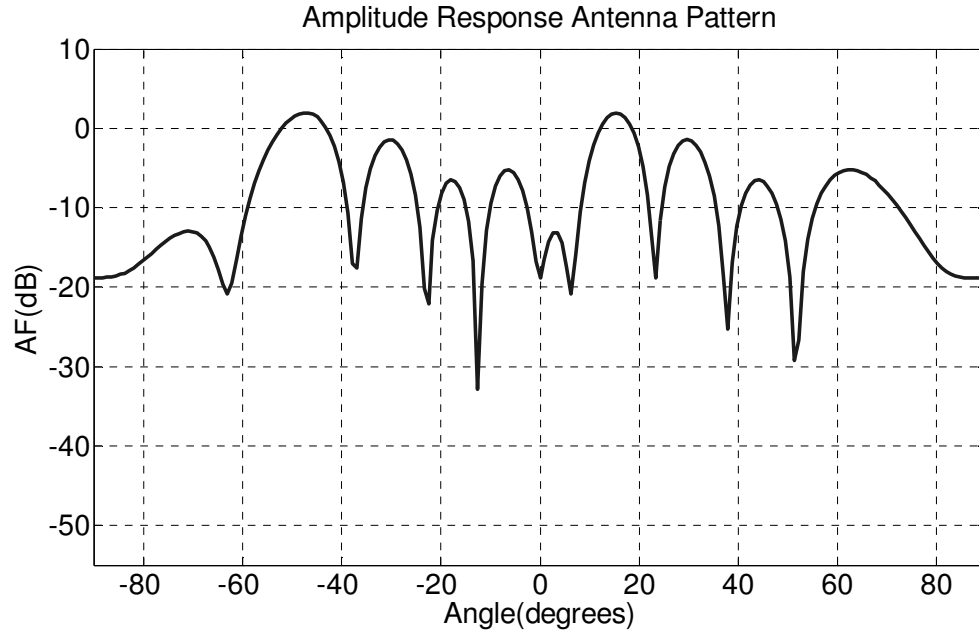
(b) Array factor (dB) versus theta (degrees)

Figure 5.6 LMS algorithm for an adaptive array with 4-element antenna and 3 interferers for antenna noise figure equal to 0.01,  $\mu=0.01$ ,  $d=0.5\lambda$ .

From figure 5.7 below we can see how the inter antenna element spacing affects the performance of the adaptive beamforming algorithms. In this simulation study we assume that the inter-antenna element spacing distance is full wavelength. The LMS algorithm for CDMA system capacity improvement is strongly affected by the antenna array arrangement, from this simulation result the normalized weighted signal magnitude is about 0.5 or half of the desired signal and the over all LMS error obtained averages to over 0.5 or it results in over 50% error. Thus, using this value deteriorate the performance of LMS algorithm as we can see from figure 5.7 this result can not accepted. Increasing inter antenna element spacing causes the main lobe in undesired direction, which causes grating lobe.



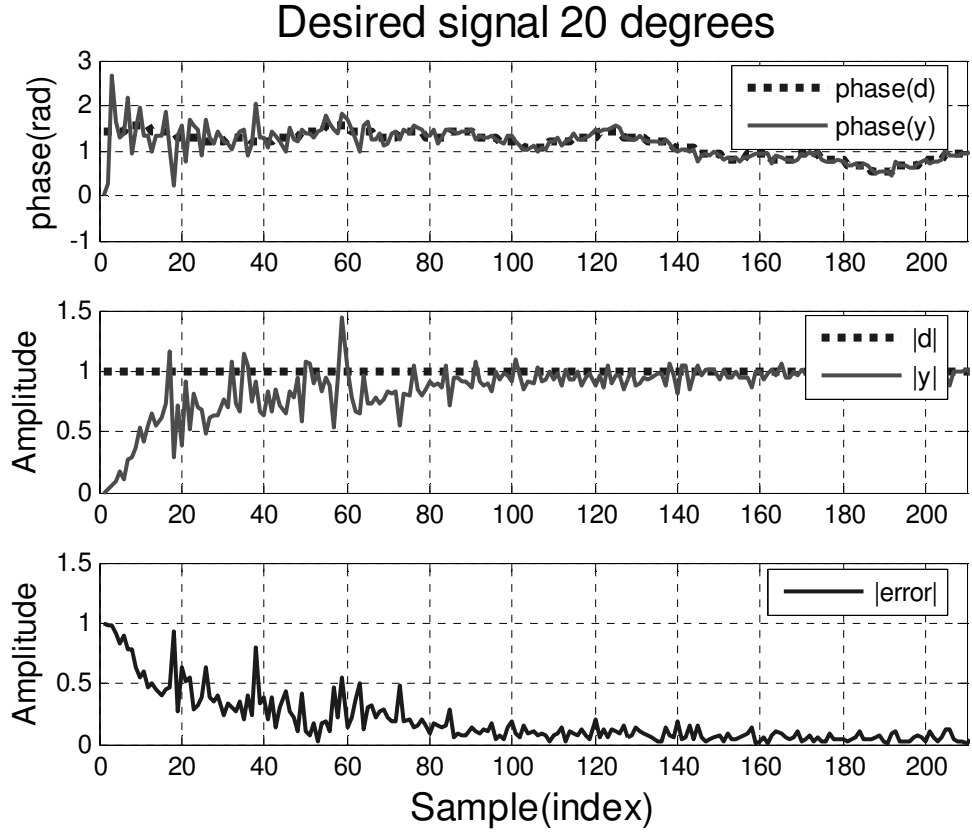
(a) Amplitude and Phase estimation versus time of iteration



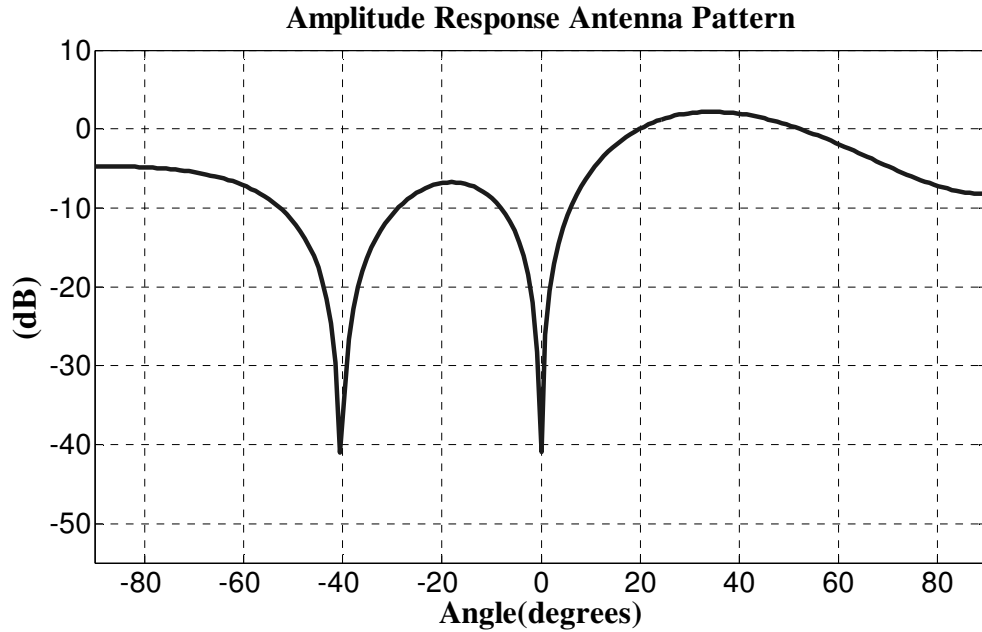
(b) Array factor (dB) versus theta (degrees)

**Fig.5.7** *LMS algorithm for an adaptive array with 6-element antenna and 5 interferers for antenna noise figure equal to 0.1 and  $\lambda=d$ . and  $\mu = 0.01$  Array factor (dB) versus theta (degrees)*

From figure 5.8 below the inter antenna element spacing distance is one fourth wavelength ( $0.25\lambda$ ). From this simulation result the normalized weighted signal magnitude is very slowly adapt to the desired signal and the LMS error obtained is also decreases slowly. But comparing with the case where inter-antenna element spacing distance is full wavelength ( $d=\lambda$ ), the result obtained by this simulation study is more practical and acceptable. Also from the simulation result obtained for the array factor, only two interfering signals would be rejected even though from figure 5.5 using the same numbers of antenna element five strong interfering signals are rejected. As seen from figure 5.8(b) the inter-element spacing decreases the number of rejected interferences are limited so, the system capacities and coverage becomes reduced. Thus using such value deteriorates the performance of beamforming algorithm used to enhance the system performance.



a) Amplitude and Phase estimation versus time of iteration

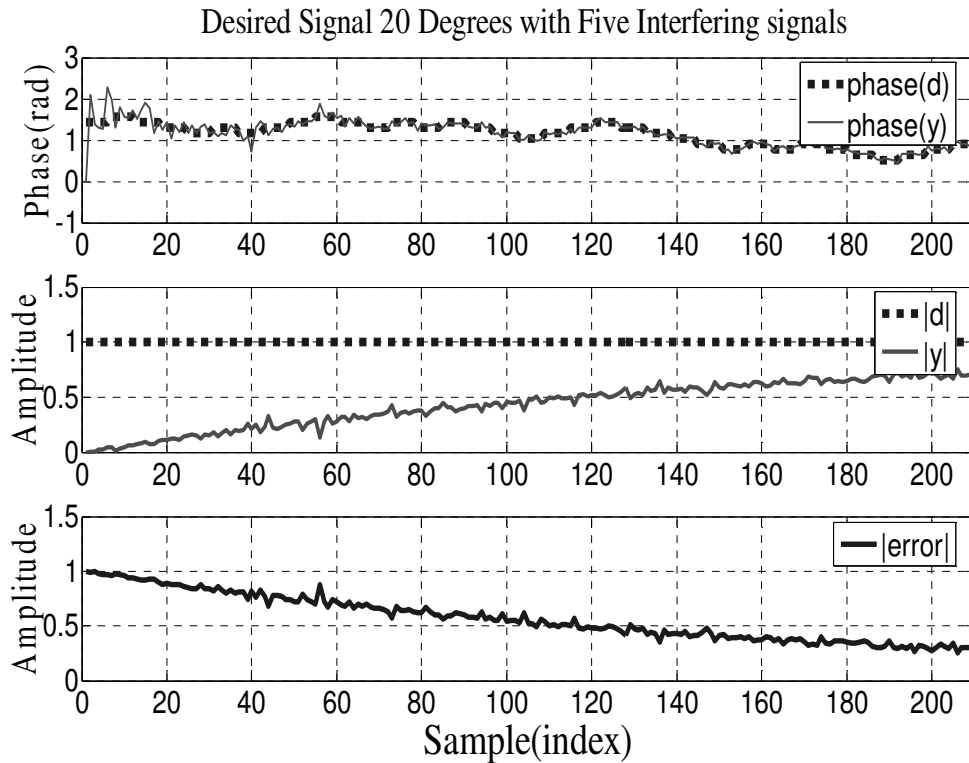


b) Array factor (dB) versus theta (degrees)

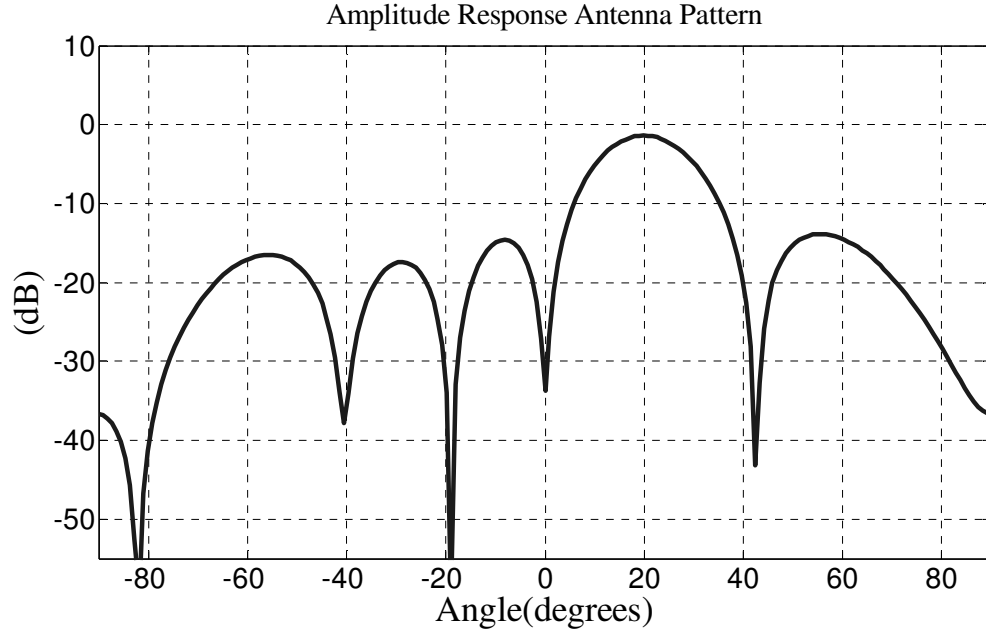
**Fig.5.8** LMS algorithm for an adaptive array with 6-element antenna and 4 interferers for antenna noise figure equal to 0.1 and  $d=0.25\lambda$  and  $\mu = 0.01$

The simulating parameters implemented for the result given by figure 5.9 below is the same with parameters that result figure 5.4, only differ by the step size  $\mu$ . Here  $\mu = 0.001$ , using this value and keeping other parameters the same we get the result shown by figure 5.9 below. From this simulation result the normalized weighted signal magnitude is very slow to adapt toward the desired signal and the LMS error obtained is also very slowly decreases. As one can see over the considered sample the normalized weighted signal magnitude is under 0.6 and the LMS error is above 0.4.

When number of sample (at number of iterating time) is 120 the weighted signal normalized amplitude is 0.5 and LMS error is also 0.5. Based on the simulation result given by figure 5.9 to implement small step size the LMS adaptive beamforming algorithm must require very large number of reference signal, which need more bandwidth. Since the wireless bandwidth is highly limited, it is not economical to use more reference signals as it wastes more scarce resources. Thus using such value deteriorates the performance of beamforming algorithm used to enhance the system performance.



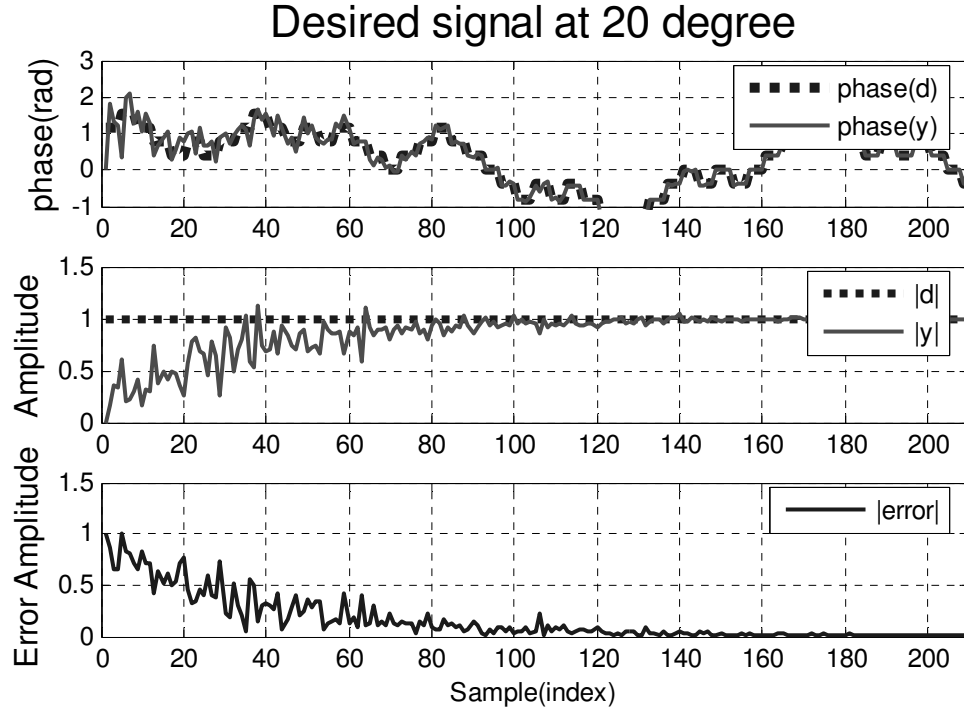
a) Amplitude and Phase estimation versus time of iteration



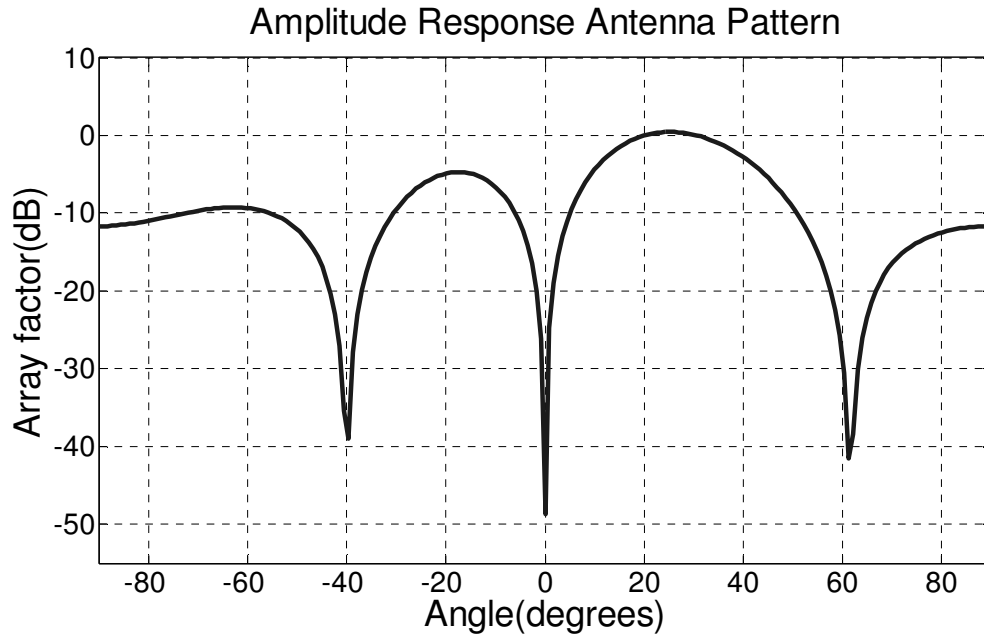
b) Array factor (dB) versus theta (degrees)

**Fig.5.9** LMS algorithm for an adaptive array with 6-element antenna and 5 interferers for antenna noise figure equal to 0.01 and  $0.5 \lambda = d$  and  $\mu = 0.001$

The simulation result given by figure 5.10 below is uses the same simulation parameters with figure 5.4, but they only differ in the number of antenna element and assumed number of interfering signals. In this simulation study 4-element antenna arrays and 2 interfering signals are considered. The maximum interference that can be rejected using 4 antenna elements is 3. From the result obtained we can see that there is no more shift in phase between the desired signals and weighted. The normalized weighted amplitude is slow to adapt to the desire signal and the LMS error is decreasing slowly. After 140 samples the result obtained from this simulation relatively the same with the result obtained from figure 5.4 after 60 samples.



a) Amplitude and Phase estimation versus time of iteration



b) Array factor (dB) versus theta (degrees)

**Fig.5.10** LMS algorithm for an adaptive array with 4-element antenna and 2 interferers for antenna noise figure equal to 0.01,  $\mu=0.01$ ,  $d=0.5\lambda$ .

### 5.4.2 CMA for Adaptive Beamforming

In reality, an exact measurement of the gradient vector is not possible, since this would require a prior knowledge of  $E[|s(n)|^p]$ . The strategy is to scale  $|s(n)|$  to unity and to use the instantaneous estimate is discussed in sub section 3.9.2.1. The algorithm is simulated by MATLAB and the result of the simulation can be found in Figure 5.11 and 5.12 below. The algorithm converges slower than the LMS algorithm, as we can observe from figure 5.4 and figure 5.11 and figure 5.3 and figure 5.12. The simulation was done with a relatively low system noise. The interferers are the same as in the LMS experiment with the same angle of arrival of the signals.

The signals of the interferers arrive at an angle of  $-10$ ,  $-40$  and  $58$  degrees. The signal to be received arrives at an angle of  $20$  degrees as shown in figure 5.13 below. Figure 5.13 shows the amplitude response of the adaptive array factor, where three interferers are rejected. The results demonstrate the concept of the CM algorithm. From the comparison of these two algorithms we observe a difference in initialization of weighting vectors. During the efforts to simulate the CM algorithm it was clear that the algorithm is less stable than the LMS algorithm. Simulations of the algorithm using the error defined in (3.35) have not resulted in stable results. The advantage of the CM algorithm is the fact that it only needs the instantaneous amplitude of the array output  $|y(k)|$  and therefore no synchronization is required where as LMS require synchronization as discussed in the previous section. Due to this property, the CM algorithm is relatively simple to implement.

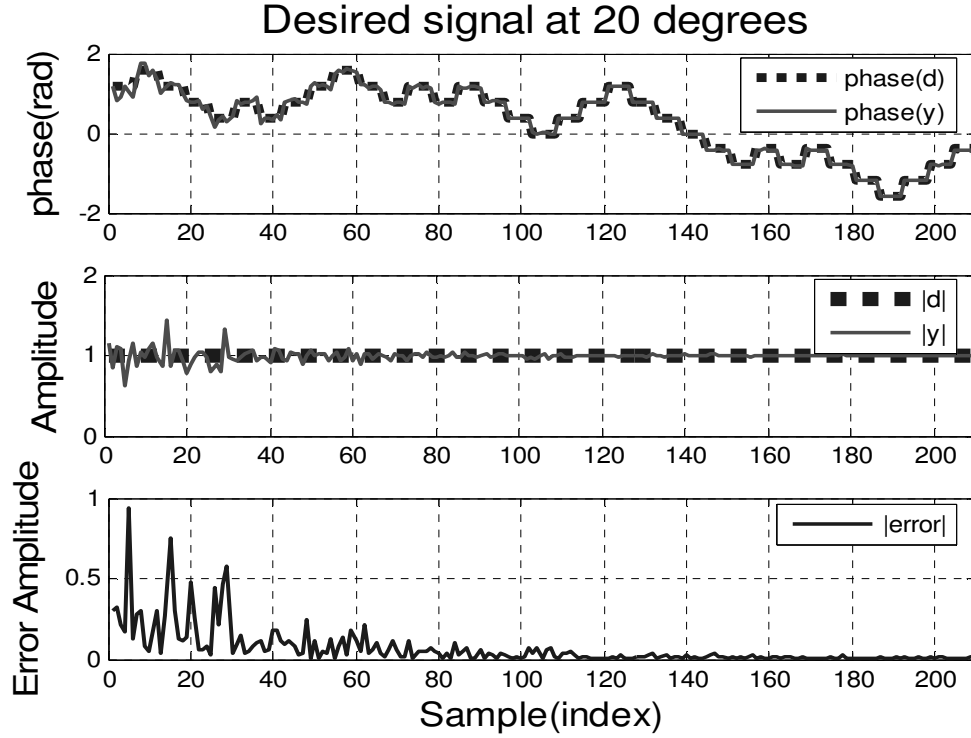


Figure 5.11 CMA algorithms for an adaptive array with 6-element antenna and 5 interferers for antenna noise figure equal to 0.01,  $\mu=0.01$ ,  $d=0.5\lambda$ . Amplitude and Phase estimation versus time of iteration

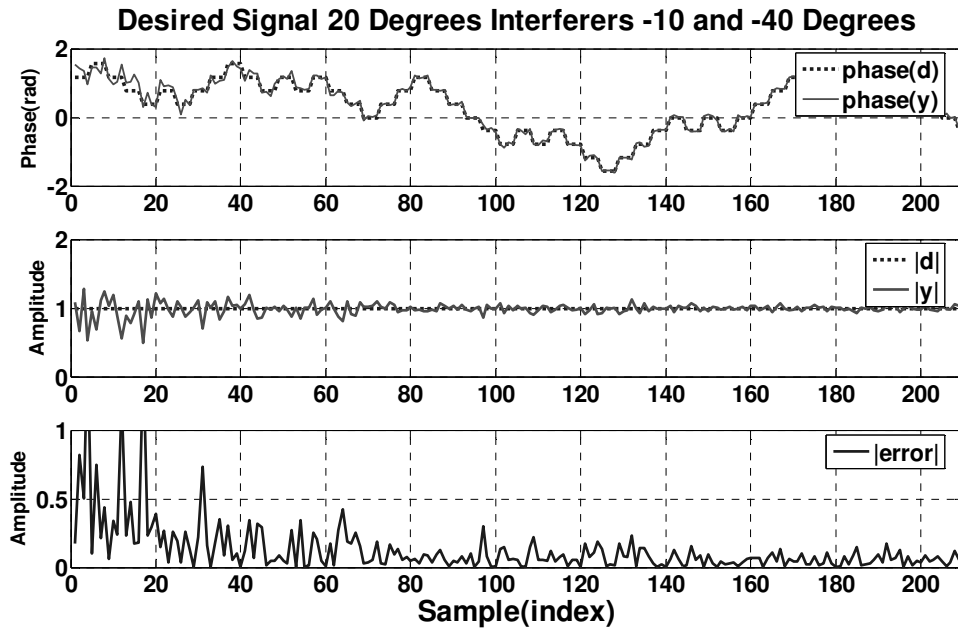


Figure 5.12 CMA algorithm for an adaptive array with 6-element antenna and 5 interferers for antenna noise figure equal to 0.1,  $\mu=0.01$ ,  $d=0.5\lambda$ . Amplitude and Phase estimation versus time of iteration

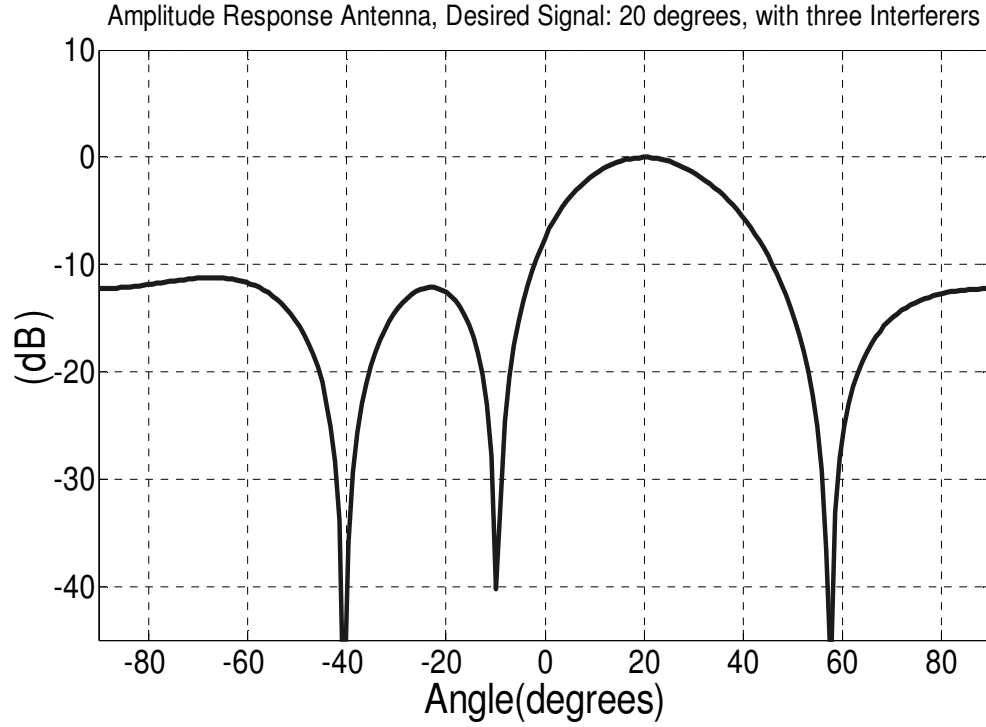
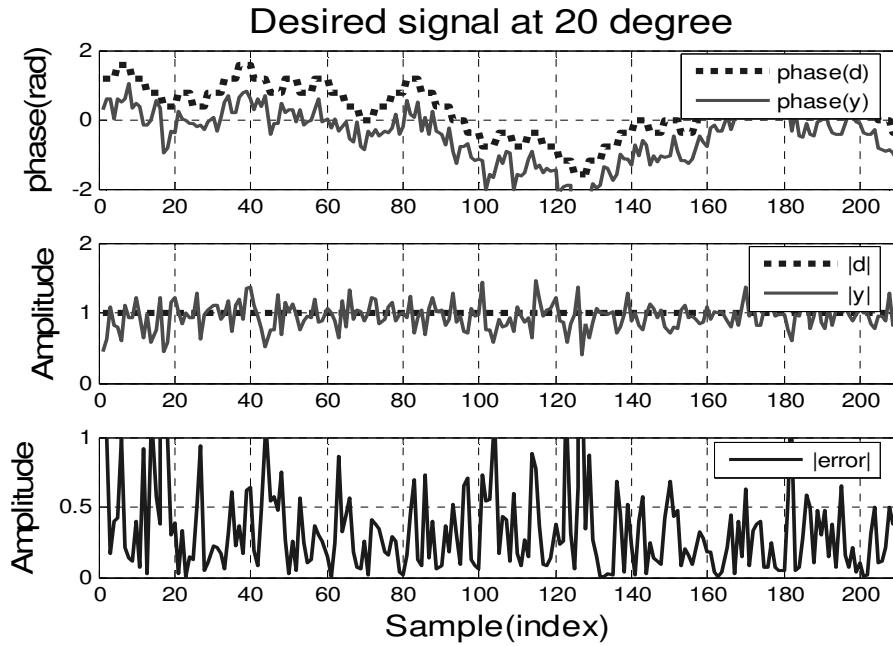
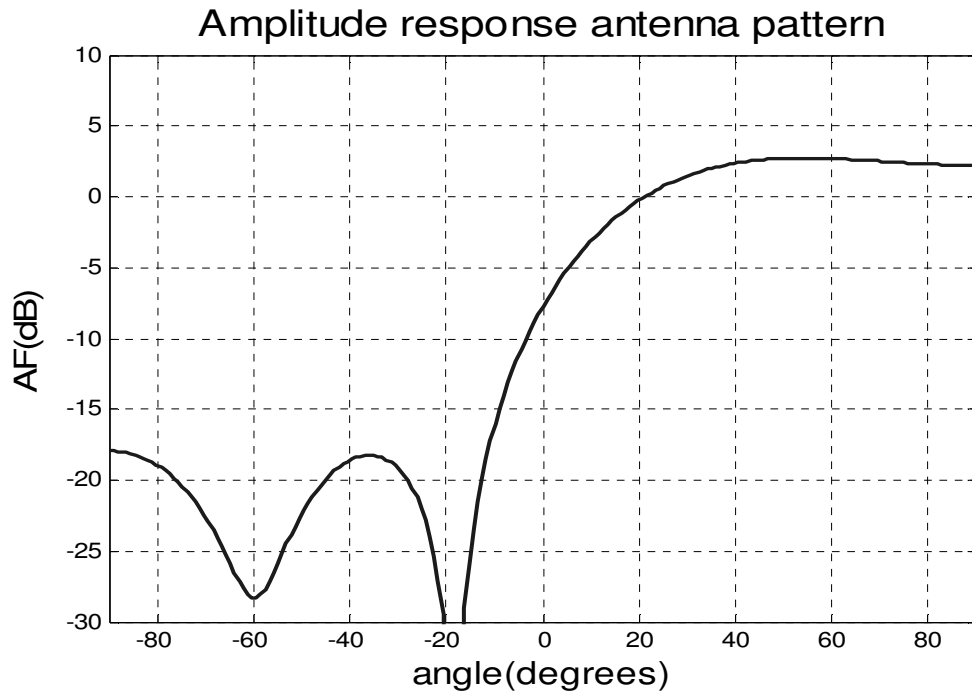


Figure 5.13 CMA algorithm for an adaptive array with 4-element antennas and 3 interferers for antenna noise figure equal to 0.01,  $\mu=0.01$ ,  $d=0.5\lambda$ . Array factor (dB) versus theta (degrees)

Figure 5.14 (a) below is the simulation result obtained for CMA algorithm using the same parameters with the parameters used for simulating the figure 5.11 only differ in the inter-antenna spacing distance. As one can observe from the figure 5.14 (a) the phase of the desired signal and weighted signal have no any point in common, which introduce more error. Comparing this with the desired signal and weighted signal phase given by figure 5.11 and the error magnitude for the two cases, we can realize that the change in inter-antenna element spacing affects the performance of the CMA algorithm for digital beamforming to enhance system capacity. As explained above the inter element antenna spacing  $d=0.25\lambda$  is not suitable distance to implement this algorithm and it can not properly reject the interferences.



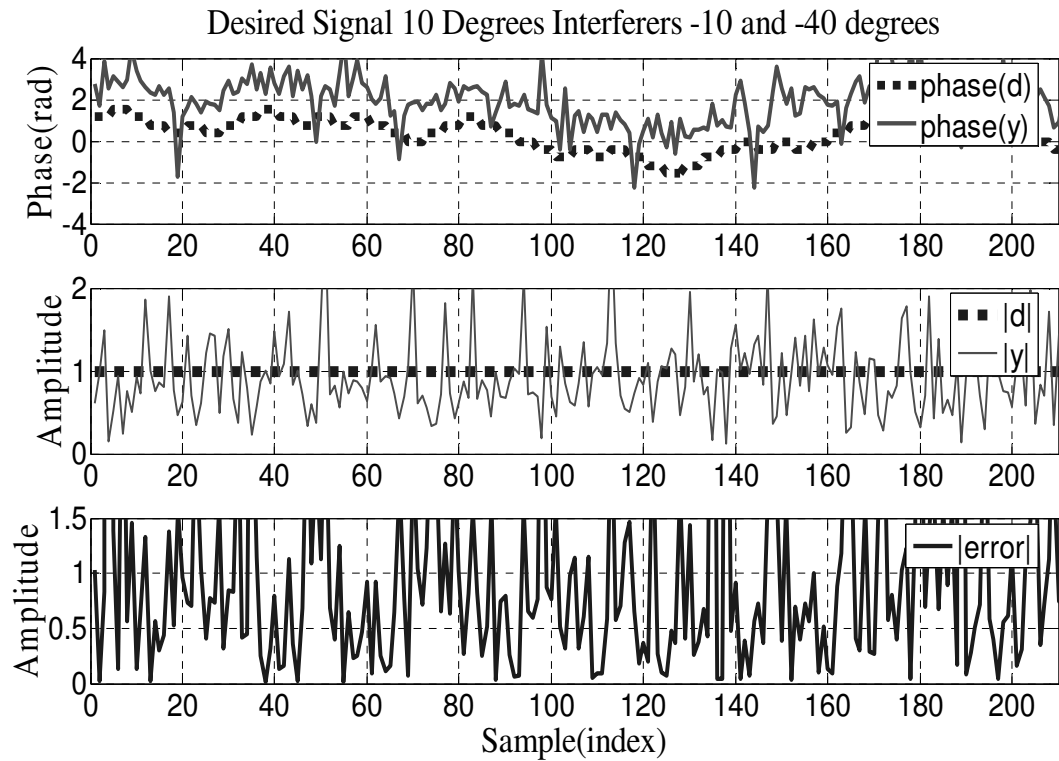
a) Amplitude response and Phase estimation versus time of iteration



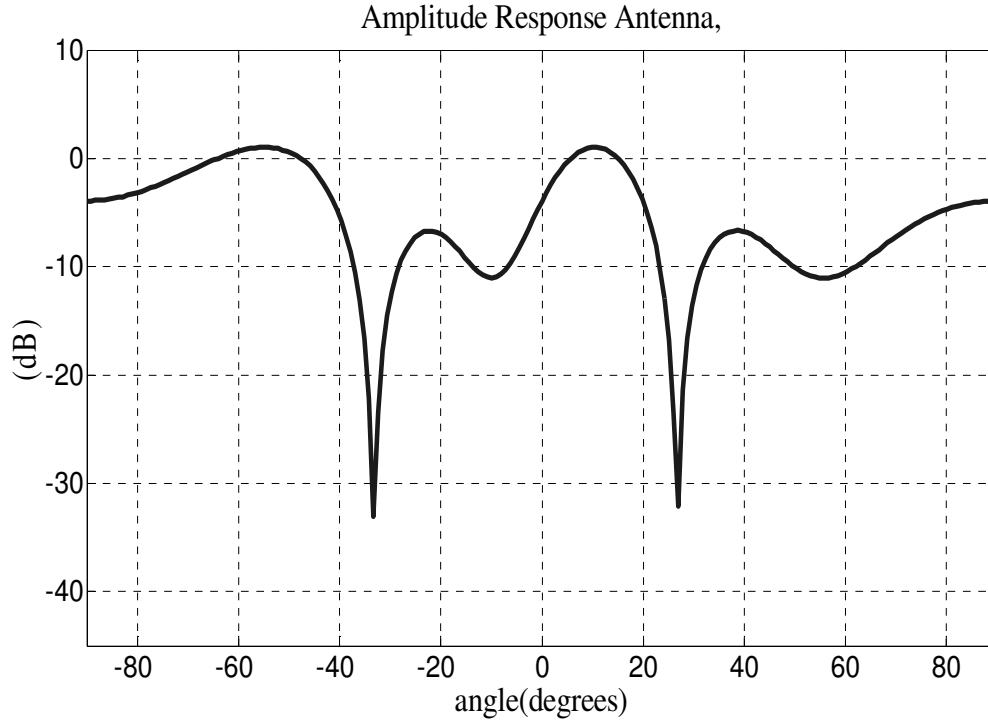
b) Array factor (dB) versus theta (degrees)

**Figure 5.14** CMA algorithms for an adaptive array with 4-element antenna and 2 interferers for antenna noise figure equal to 0.01 and  $d = 0.25\lambda$  and  $\mu = 0.01$

From figure 5.15 (a) and (b) shown below we see that as the CMA for digital beamforming affected by the magnitude of inter-antenna element spacing comparing with figure 5.11 give above. Using  $d = \lambda$  to enhance the system performance by implementing digital beamforming concepts as observed from figure 5.7, 5.8 for LMS algorithm and 5.11 for CMA algorithms. The CMA error magnitude is highly fluctuating randomly and averaged to around 0.75 or 75% and the weighted signal is highly oscillating and become unstable. Also the phase of the weighted signal is not follow the phase shift of desired signal. Therefore, the result obtained using such parameter is not practical as it shifts the desired signal arrival angle and mismatch the nulls with interfering signals as we can see from figure 5.15 (b) below.



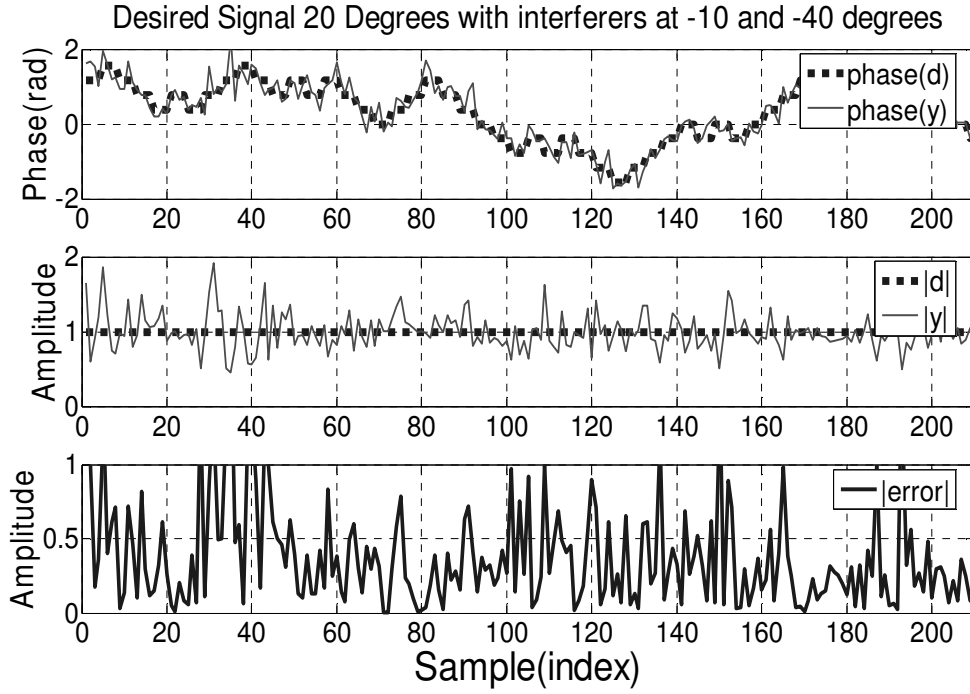
(a) Amplitude and Phase estimation versus time of iteration



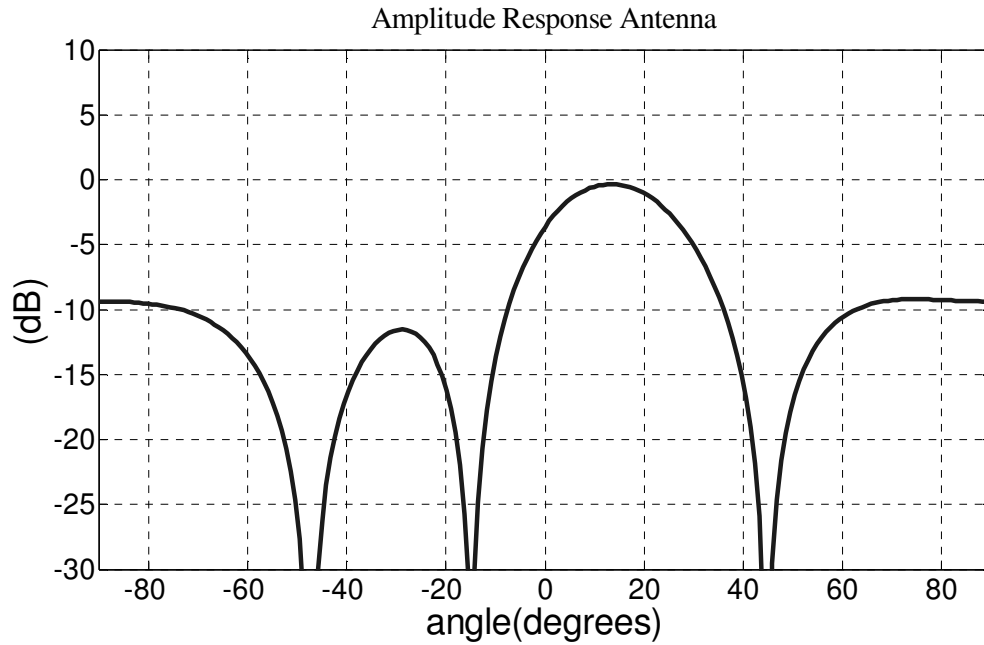
(b) Array factor (dB) versus theta (degrees)

**Figure 5.15** CMA algorithm for an adaptive array with 4-element antenna and 2 interferers for antenna noise figure equal to 0.01 and  $\lambda = d$  and  $\mu = 0.01$

As discussed above in figure 5.9 using very small adaptation step size affects the beamforming algorithms performance. From these two simulation result for different adaptive algorithm we observe that in order to adapt to the desired signal the adaptation step size should be reasonable. But comparing to the result obtained from figure 5.9 the result obtained from simulation result given by figure 5.16 is highly unstable and it cannot show any improvement in error amplitude and estimation of desired signal. These indicate that the LMS algorithm converges as the number of sample increases while CMA cannot do this.



(a) Amplitude and Phase estimation versus time of iteration



**b) Array factor (dB) versus theta (degrees)**

**5.16 CMA algorithm for an adaptive array with 4 antennas and 3 interferers for antenna noise figure equal to 0.01 and  $0.5\lambda = d$  and  $\mu = 0.001$**

### **5.4.3. Maximization signal-to-Interference Noise ratio**

From figure 5.17 performance evaluation algorithm flow chart we obtain the weight vector which is used to maximize SINR of the adaptive beamforming. From figure 5.18 shown below we see that the performance increases as the number of adaptive antenna array elements increase while the BER is becoming lower and lower. The simulation of SINR vs BER for CMA, LMS with array element antenna number 4, 6 and for the same antenna array with out using the beamforming concept is shown in figure 5.18. Thus the use of adaptive antennas improves the performance of mobile communications systems in terms of SINR vs BER as obtained from the simulation result given under figure 5.18 below. The signal reception at  $\text{BER}=10^{-3}$  with 6-element antenna array (LMS) more performance than the others. Therefore, using more element antenna array the signal reception quality improved with the beamforming algorithm.

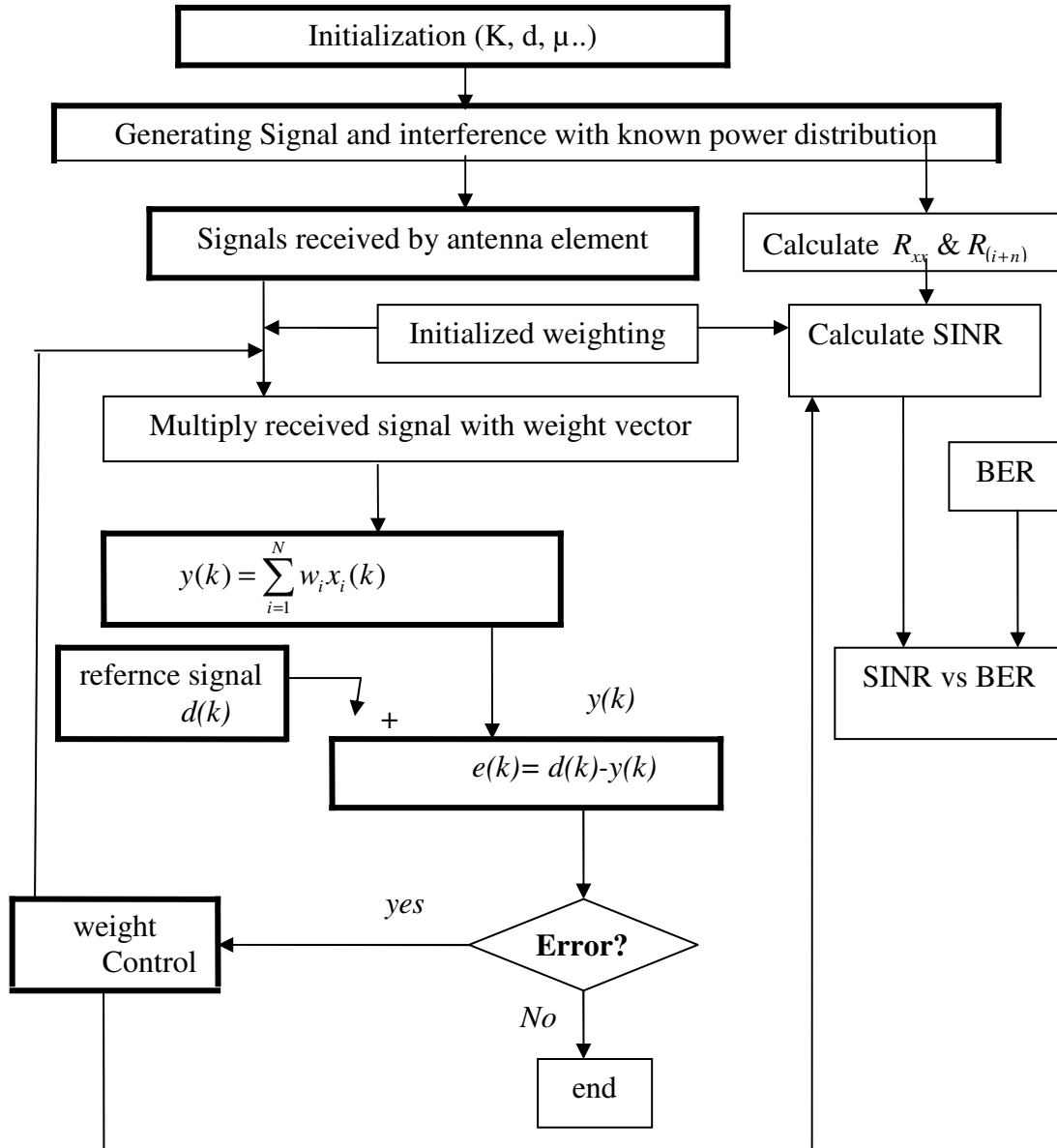


Fig 5.17 Flow chart for the Performance evaluation algorithm

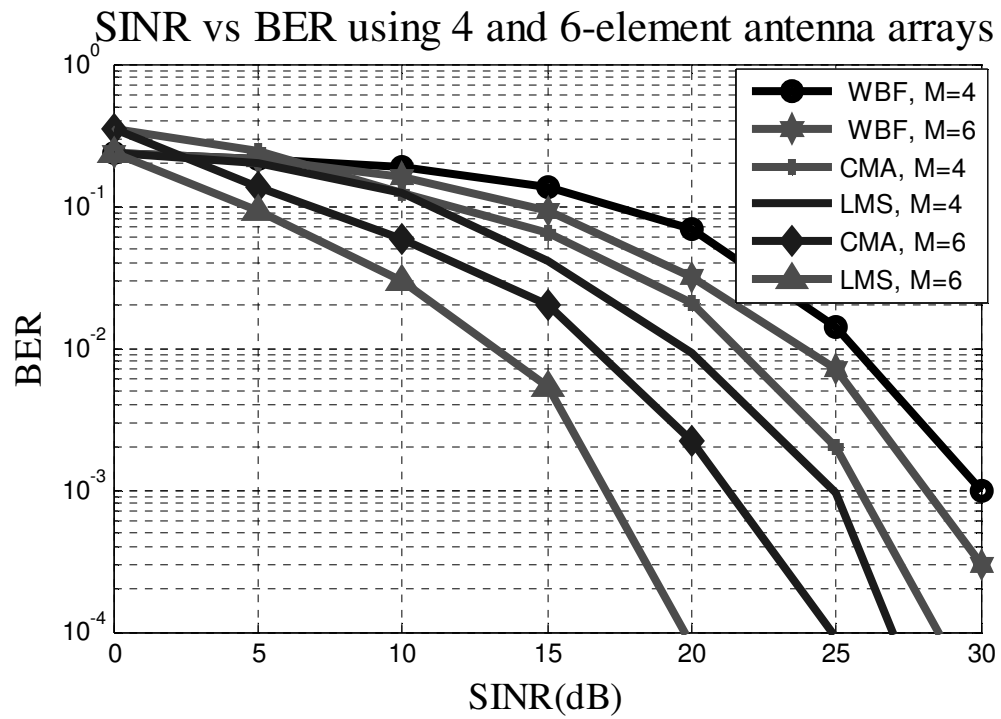


Figure 5.18 SINR vs. BER

Table.2 Summary of performance

| Algori<br>thm | Parameter           |                       |                       |                 |                                     |           | Performa<br>nce (%) |
|---------------|---------------------|-----------------------|-----------------------|-----------------|-------------------------------------|-----------|---------------------|
|               | Array<br>element(M) | Element<br>spacing(d) | Step<br>size( $\mu$ ) | System<br>noise | Converge<br>nce rate<br>(iteration) | BER       |                     |
| LMS           | 4                   | $0.5\lambda$          | 0.01                  | 0.01            | 120                                 | $10^{-3}$ | 12.5                |
|               | 6                   | $0.5\lambda$          | 0.01                  | 0.01            | 60                                  |           | 30                  |
| CMA           | 4                   | $0.5\lambda$          | 0.01                  | 0.01            | -                                   |           | 5.35                |
|               | 6                   | $0.5\lambda$          | 0.01                  | 0.01            | 140                                 |           | 23.2                |

## CHAPTER SIX

### CONCLUSION AND FUTURE WORKS

#### 6.1 Conclusions

In this work, we study about the digital beamforming to enhance the mobile cellular network system capacity and coverage. For this we have discuss about different beamforming types, different antenna array geometry and different beamforming algorithms. Techniques for increasing flexibility of the network to deal with new services and traffic characteristics are the requirements and implementation challenges. The CDMA is one of the multiple access mechanism implemented in wireless networks to improve the system capacity over other access methods. But CDMA capacity is highly affected by interferences. To overcome this problem we try to null the interferers and receive the desire signal at its maximum value, which results in maximization of SINR of the system. Once the SINR can be maximized it is true that the CDMA system capacity increased or within a given base station we can have more users. Also the CDMA system coverage increased as the strong interferers are rejected by the deep nulls to the direction of strong interferers and maximum toward the desired signal. These results to the system from the digital beamforming techniques implemented at the base station.

The inter-element spacing between the antenna elements is an important factor in the design of an antenna array: a distance of more than  $\lambda/2$  between the elements (where  $\lambda$  is the wavelength of the incoming or outgoing signals) produces grating lobes. If the inter element spacing between the antenna elements is less than  $\lambda/2$ , the mutual coupling between the elements can not be neglected any more. That's why the optimum element spacing for beamforming application distance,  $d = \lambda/2$ . We overviewed the concept of beamforming and discussed the techniques used in solving the problems associated with the wireless personal communication. We discussed how multibeam antennas are capable of increasing the capacity of CDMA systems. In the LMS, we need a reference signal  $d(k)$  to which we want to have the output converge. In CMA, however, the reference signal is not necessary, we use the a priori information that  $|y| = 1$  in the absence of interfering signals.

The simulation result for the two adaptive algorithms which are implemented to test the performance improvement of the digital beamforming concept show that the number of interferers to be rejected, the level of the maximum and null, the error magnitude, the estimation (adaptation) of weighted signal to desired signal and the phase shift varies with variations of antenna number, step size, system noise, and inter antenna element spacing. The results obtained from the analysis in this thesis are valuable for mobile cellular network planning and dimensioning. From the simulation study performed, the LMS algorithm performs better than the CMA algorithm in all assumed condition with some computational complexity. Therefore, these algorithms have been shown to give a significant performance improvement at reasonable computational costs over algorithms which only consider the array manifold (over the switched beamforming algorithms). The simulation study for SINR vs BER shows the performance measure of a CDMA communication link, which give us an insight about the performance improvement resulted from beamforming techniques.

## **6.2 Future Work**

There are several issues that can be addressed as a future work related to this research areas. This can be extended to the downlink CDMA mobile communication. A very interesting future work, which can be addressed, is to implement this concept in the mobile station by using multi antenna element at the mobile station in order to provide better quality and increase the system capacity and coverage.

Future studies could also be performed on a higher chip rate bandwidth CDMA system which would provide a higher processing gain and possible multimedia wireless communication such as high data rate video transmissions. Also using larger number of antenna elements to get more precise and fast estimation results by using RLS and perturbation-RLS (PRLS) adaptive beam forming algorithms. This concept can be implemented in higher information bit transmission rates to overcome the effect of a harsh propagation environment with small coherence time where the fading effect over an information bit is relatively flat.

## Reference

1. Jack H. Winters, "Smart Antennas for Wireless Systems," *IEEE Personal Communications, Vol. 1, pp. 23-27, February 1998.*
2. T. E. Curtis, "Digital beamforming for sonar system," *IEEE Proc. Pt. F, vol. 127, pp. 257 – 265, Aug. 1980.*
3. A. F. Naguib and A. Paulraj, and T. Kailath. "Capacity improvement with base station antenna arrays in cellular CDMA." *IEEE Transaction on Vehicular Technology Vol.43 No.3 pp. 691-698, August 1994.*
4. J. Litva, "Digital Beamforming in wireless communications", *Artech House Publishers, 1996.*
5. Jin Yu, Yu-Dong Yao, Jinyun Zhang and Andreas F.Molish. "Reverse-Link Capacity of power-controlled CDMA systems with beamforming" *IEEE Transaction Vehicular Technology, Vol.53 No. 5, September 2004.*
6. Geoffrey W.K.Colman. "An Investigation into the Capacity of cellular CDMA communication systems with Beamforming in Environments with Scatter." *PhD Thesis, Dept. of Electrical and Computer Engineering, Queen's University Kingston Ontario, Canada, Sep.1998.*
7. Tao Luo. "Beamforming in the uplink and Downlink Channels of a cellular CDMA communication System". *Master Thesis, Dept. of Electrical and Computer Engineering, Queen's University Kingston Ontario, Canada, February 1998.*
8. A. Mark Earnshaw. "An Investigation into improving performance of cellular CDMA communication systems with digital beamforming." *PhD .Thesis, Dept. of Electrical and Computer Engineering, Queen's University Kingston Ontario, Canada, October 1997 .*
9. A. F. Naguib and A. Paulraj, "Recursive Adaptive Beamforming for wireless CDMA," *IEEE International Conference on communications, vol.3, no. 3, pp. 1515-1519, 1995.*
10. Qualcomm Inc., "Mobile station base compatibility standard for dual-mode wideband spread system," *1993.*

11. S. Ariyavisitakul, "Signal and interference statistics of a CDMA system with feedback power control-part II," *IEEE Transactions on Communications*, vol. 42, no. 2, 3, 4,.
12. B. Sklar, "Rayleigh fading channels in mobile digital communication systems" *Part I and II: Characterization*," *IEEE Communications Magazine*, vol. 35, no. 7, pp. 90-109, 1997.
13. A. F. Naguib and A. Paulraj, "performance of CDMA cellular networks with base-station antenna arrays," *International Zurich Seminar on Digital Communications*, pp. 87-100, 1994.
14. J. C. Liberti and T. S. Rappaport, "Smart Antennas for Wireless Communications: S-95 and Third Generation CDMA Applications," *NJ, Prentice Hall*, 1999.
15. F. Rashid-Farrokhi, K. J. R. Liu and L. Tassiulas, "Transmit beamforming and power control for cellular wireless systems," *IEEE J. Select. Areas Commun.*, vol. 16, no. 8, pp. 1437-1450, Oct. 1998.
16. B.D. Van Veen and K.M. Buckley, "Beamforming: A Versatile Approach to Spatial Filtering," *IEEE ASSP Magazine*, April 1988.
17. R. T. Compton, "Adaptive Antennas: Concepts and Applications," *Prentice Hall. Englewood Cliffs, NJ*, 1988.
18. Ma, M. T. "Theory and application of antenna arrays." *John Wiley & Sons, New York* 1974
19. J. H. Winters, "Optimum combating in digital mobile radio with co-channel interference," *IEEE Trans. Veh. Technol.*, vol. 33, pp. 144-155, Aug. 1984
20. J. fernandez, I. R. Corden, and M. Barrett, "Adaptive array algorithms for optimal combining in digital mobile communications systems," in *IEEE Eighth Int. Conf. Antenna and propagate.*, *Heriot-Watt University, U.K.*, 1993.
21. Y. Wang and J. R. Cruz, "Adaptive antenna arrays for cellular CDMA Communication systems," in *Proc. IEEE Int'l. Conf. Acoustics, Speech and Signal Processing, Detroit*, pp. 1725-1728, 1995.
22. D. H. Johnson and Dan E. Dudgeon, "Array signal processing: concepts and techniques," *P T R Prentice-Hall*, 1993.
23. D. N. Godard, "Self-recovering equalization and carrier tracking in a two-dimensional data communication system," *IEEE Trans. Comm.*, vol.28, pp. 1867-1875, 1980.

24. V. Kezys and J. Litva, "A versatile intelligent antenna testbed for wireless applications," in *Wireless'95 Proc., Calgary, Alberta, 1995*.
25. A. Swindlehurst, S. Daas, and J. Yang, "Analysis of a decision *directed beamformer*," *IEEE Trans. Signal Processing*, vol. 43, pp. 2920-2927, Dec. 1995.
26. S. Bellini, "Busgang techniques for blind equalization," in *Globecom, Houston, pp. 1634-1640, 1993*.
27. R. C. Hansen, "Phased Array Antennas," *John Wiley and Sons, New York, 1997*
28. M. D. Zoltowski, G. M. Kautz, and S. D. Silverstein, "Beamspace root-MUSIC," *IEEE Trans. Signal Processing*, vol. 41, pp. 344-364, Jan. 1993.
29. K. S. Gilhousen, I. M. Jacobs, R. Padovani, A. J. Viterbi, L. A. Weaver, Jr., and C. E. Wheatly III, "On the capacity of a cellular CDMA system," *IEEE Transactions on Vehicular Technology*, vol. 40, pp. 303-312, May 1991.
30. A. M. Viterbi and A. J. Viterbi, "Erlang Capacity of a Power Controlled CDMA System," *IEEE Journal on Selected Areas in Communication*, vol. 11, pp. 892-900, August 1993.
31. Kyoung Il Kim et al., "Handbook of CDMA System Design, Engineering and Optimization", *Prentice Hall, 2000*.
32. H. Schulze and L. Christian, "Theory and Applications of OFDM and CDMA: Wideband Wireless Communications," *John Wiley & Sons, England, 2005*.
33. A. E. Zooghyby. "Smart Antenna Engineering", *Artech House, Boston, 2005*
34. H. Liu and Q-A. Zeng, "Modeling and performance analysis of mobile communication systems using adaptive beamforming technique," *Proceeding of the 3rd Wireless Telecommunications Symposium, Pomona, CA, pp. 33-38 (2004)*.
35. J. Li and P. Stoica. "Robust Adaptive Beamforming", *John Wiley & Sons, Inc., Hoboken, New Jersey, 2006*
36. Samuel C. Yang "CDMA RF system engineering." *Artech House mobile communications library, 1998*.
37. Kamil Sh. Zigangirov . "Theory of code division multiple access communication". A *John Wiley & sons, inc., publication, 2004*.
38. Jhong Sam Lee Leonard E. Miller, "CDMA Systems Engineering Hand book", *Artech house, Boston, London, 1998*.

39. A.Mark Earnshaw & Steven D .Blostien, “ A chip-level IS-95-Compliant cellular CDMA Simulator: Design, Implementation, and Analysis, Image Processing and communication Lab”, *Department of Electrical & Computer Engineering, Queen's University*.
40. K.Garg . Kenneth Smolik Joseph E. Wilkes ‘ Applications of CDMA in Wireless/Personal Communications’ *Prentice Hall PTR, 1997*

I, the undersigned, declare that this thesis is my original work, has not been presented for a degree in this or any other university, and all sources of materials used for the thesis have been fully acknowledged.

Name: Degebassa Dida

Signature: \_\_\_\_\_

Place: Addis Ababa

Date of submission: \_\_\_\_\_

This thesis has been submitted for examination with my approval as a university advisor.

Dr.-Ing Hailu Ayele

Advisor's Name

\_\_\_\_\_  
Signature

% 'LMSarray.m'

## **%adaptive array demonstrator LMS algorithm for DBF A**

```
theta_x = 20 %degrees, direction of signal x
theta_x = (2*pi/360)*theta_x
theta_n1 = 0; %degrees, direction of noise source 1
theta_n1 = (2*pi/360)*theta_n1
theta_n2 = -40 %degrees, direction of noise source 2
theta_n2 = (2*pi/360)*theta_n2
%theta_n3 = -19 %degrees, direction of noise source 3
%theta_n3 = (2*pi/360)*theta_n3
%theta_n4 = -82 %degrees, direction of noise source 4
%theta_n4 = (2*pi/360)*theta_n4
%theta_n5 = 41 %degrees, direction of noise source 5
%theta_n5 = (2*pi/360)*theta_n5
theta = pi*[-1:0.005:1];
arraylength = 4; %nr of antennas
bitrate = 100;
fsim = 4*bitrate; %simulation frequency
Ts = 1/fsim; %simulation sample period
message = [1 -1 1 1 1 -1 1 1 -1 -1 1 1 1 -1 1 1 1 1 -1 -1 -1 1 1 1 1 1 -1 1 -1 1 1 1 -1 -1 -1 -1
1 -1 1 1 -1 -1 -1 -1 1 1 1 1 -1 1 1 1 1 1 -1 1 -1 1 1 1 -1 -1 -1 -1
1 -1 1 1 -1 -1 -1 -1 1 1 1 1 -1 1 1 1 1];
message = upsample(message, fsim/bitrate); %upsample message
t = Ts:Ts:(length(message)/fsim); %timeline
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%generate a complex MSK signal
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
mesint=(cumsum(message))/8;
Q = cos(pi*mesint);
I = sin(pi*mesint);
plot(Q)
hold on;
plot(I,'r')
signal_x = I+j*Q; %the signal to be received
% the undesired complex noise singals-> uniform phase, gaussian amplitude distribution
signal_n1 = normrnd(0,1,1,length(t)).*exp (j*(unifrnd(-pi,pi,1,length(t))));
signal_n2 = normrnd(0,1,1,length(t)).*exp (j*(unifrnd(-pi,pi,1,length(t))));
noise = zeros(arraylength, length(t));
% system noise for every antenna
for i = 0:arraylength-1,
noise(i+1,:) = normrnd(0,0.01,1,length(t)).*exp (j*(unifrnd(-pi,pi,1,length(t))));
end;
%K = 2*pi/ lambda
%d = 0.5*lambda
%Kd means K*d equal to pi
Kd = pi;
```

```

% array responses for the desired signal x and noise n1 and n2
arrayvec_x = zeros(1,arraylength);
arrayvec_n1 = zeros(1,arraylength);
arrayvec_n2 = zeros(1,arraylength);
for k = 0:arraylength-1,
arrayvec_x(k+1) = exp(j*k*Kd*sin(theta_x));
end;
for k = 0:arraylength-1,
arrayvec_n1(k+1) = exp(j*k*Kd*sin(theta_n1));
end;
for k = 0:arraylength-1,
arrayvec_n2(k+1) = exp(j*k*Kd*sin(theta_n2));
end;
x = zeros(arraylength, length(t));
n1 = zeros(arraylength, length(t));
n2 = zeros(arraylength, length(t));
% received signal from signal source x
for i = 0:arraylength-1,
x(i+1,:) = signal_x .* arrayvec_x(i+1);
end;
% received signal from noise source n1
for i = 0:arraylength-1,
n1(i+1,:) = signal_n1 .* arrayvec_n1(i+1);
end;
% received signal from noise source n2
for i = 0:arraylength-1,
n2(i+1,:) = signal_n2 .* arrayvec_n2(i+1);
end;
%total received signal is the
%+- -+
%| signal received at antenna 1 |
%| signal received at antenna 2 |
%| | |
%| signal received at antenna n |
%+- -+
signal_ns = (noise + n1+n2+x);
% define weight vector
w = zeros(1,arraylength);
mu = 0.01;
y = zeros(1,length(t));%output
e = zeros(1,length(t));%error
%%%%%%%%%%
%%
%LMS Algorithm
%%%%%%%%%%
%%%%%%%%%%
for repeat=1:1,% use repeat for more convergence on the same message

```

```

for i=0:length(t)-1,
y(i+1) = w * signal_ns(:,i+1);
e(i+1) = signal_x(i+1)-y(i+1);
w = w + mu *e(i+1)*(signal_ns(:,i+1))';
end;
end;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Plot all figures
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
close all;
subplot 311, (plot(phase(signal_x),':'))
grid on;
ylabel('phase(rad)');
xlabel('sample(index)');
axis([0 210 -1 3]);
title('desired signal 25 degrees interferers 0 and -40 degrees')
hold on;
plot(phase(y),'r');
legend('phase(d)','phase(y)')
grid on;
hold off;
subplot 312, plot(abs(signal_x),':');
hold on;
plot(abs(y),'r');
grid on;
legend('ldl', 'lyl')
ylabel('amplitude');
xlabel('sample(index)');
axis([0 210 0 1.5]);
hold off;
subplot 313, plot(abs(e));
legend('error!');
grid on;
ylabel('amplitude');
xlabel('sample(index)');
axis([0 210 0 1.5]);
for k = 0:arraylength-1,
arrayvec(k+1,:) = exp(j*k*Kd*sin(theta));
end;
%calculate response of array
F = w*arrayvec;
figure
plot(((theta/(2*pi))*360), 20*log10(abs(F)));
grid on;
title('amplitude response antenna pattern');

```

```

ylabel('(dB)');
xlabel('angle(degrees)');
axis([-90,+90,-55,10]);
hold on;
dbgrens = 40
grid on;
%figure; plot(((theta/(2*pi))*360), phase(F)); %plot phase response if
%desired
%'LMSarray.m'
%adaptive array demonstrator LMS algorithm for DBF B
theta_x = 20 %degrees, direction of signal x
theta_x = (2*pi/360)*theta_x
theta_n1 = 0; %degrees, direction of noise source 1
theta_n1 = (2*pi/360)*theta_n1
theta_n2 = -40 %degrees, direction of noise source 2
theta_n2 = (2*pi/360)*theta_n2
%theta_n3 = -19 %degrees, direction of noise source 3
%theta_n3 = (2*pi/360)*theta_n3
%theta_n4 = -82 %degrees, direction of noise source 4
%theta_n4 = (2*pi/360)*theta_n4
%theta_n5 = 41 %degrees, direction of noise source 5
%theta_n5 = (2*pi/360)*theta_n5
theta = pi*[-1:0.005:1];
arraylength = 4; %nr of antennas
bitrate = 100;
fsim = 4*bitrate; %simulation frequency
Ts = 1/fsim; %simulation sample period
message = [1 -1 1 1 1 -1 1 -1 -1 -1 1 1 -1 1 -1 1 1 1 -1 -1 -1 1 1 1 1 1 -1 1 -1 1 1 1 1 -1 -1 -1 -1
1 -1 1 -1 -1 -1 -1 -1 1 1 1 -1 1 -1 1 1];
message = upsample(message, fsim/bitrate); %upsample message
t = Ts:Ts:(length(message)/fsim); %timeline
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%generate a complex MSK signal
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
mesint=(cumsum(message))/8;
Q = cos(pi*mesint);
I = sin(pi*mesint);
plot(Q)
hold on;
plot(I,'r')
signal_x = I+j*Q; %the signal to be received
% the undesired complex noise singals-> uniform phase, gaussian amplitude distribution
signal_n1 = normrnd(0,1,1,length(t)).*exp (j*(unifrnd(-pi,pi,1,length(t))));
signal_n2 = normrnd(0,1,1,length(t)).*exp (j*(unifrnd(-pi,pi,1,length(t))));
noise = zeros(arraylength, length(t));

```

```

% system noise for every antenna
for i = 0:arraylength-1,
noise(i+1,:) = normrnd(0,0.01,1,length(t)).*exp (j*(unifrnd(-pi,pi,1,length(t))));
end;
%K = 2*pi/ lambda
%d = 0.5*lambda
%Kd means K*d equal to pi
Kd =1.5* pi;
% array responses for the desired signal x and noise n1 and n2
arrayvec_x = zeros(1,arraylength);
arrayvec_n1 = zeros(1,arraylength);
arrayvec_n2 = zeros(1,arraylength);
for k = 0:arraylength-1,
arrayvec_x(k+1) = exp(j*k*Kd*sin(theta_x));
end;
for k = 0:arraylength-1,
arrayvec_n1(k+1) = exp(j*k*Kd*sin(theta_n1));
end;
for k = 0:arraylength-1,
arrayvec_n2(k+1) = exp(j*k*Kd*sin(theta_n2));
end;
x = zeros(arraylength, length(t));
n1 = zeros(arraylength, length(t));
n2 = zeros(arraylength, length(t));
% received signal from signal source x
for i = 0:arraylength-1,
x(i+1,:) = signal_x .* arrayvec_x(i+1);
end;
% received signal from noise source n1
for i = 0:arraylength-1,
n1(i+1,:) = signal_n1 .* arrayvec_n1(i+1);
end;
% received signal from noise source n2
for i = 0:arraylength-1,
n2(i+1,:) = signal_n2 .* arrayvec_n2(i+1);
end;
%total received signal is the
%+- -+
%| signal received at antenna 1 |
%| signal received at antenna 2 |
%| | |
%| signal received at antenna n |
%+- -+
signal_ns = (noise + n1+n2+x);
% define weight vector
w = zeros(1,arraylength);
mu = 0.01;

```

```

y = zeros(1,length(t));%output
e = zeros(1,length(t));%error
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%LMS Algorithm
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for repeat=1:1,% use repeat for more convergence on the same message
for i=0:length(t)-1,
y(i+1) = w * signal_ns(:,i+1);
e(i+1) = signal_x(i+1)-y(i+1);
w = w + mu *e(i+1)*(signal_ns(:,i+1))';
end;
end;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Plot all figures
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
close all;
subplot 311, (plot(phase(signal_x),'.'))
grid on;
ylabel('phase(rad)');
%xlabel('sample(index)');
axis([0 210 -1 3]);
title('desired signal 25 degrees interferers 0 and -40 degrees')
hold on;
plot(phase(y),'r');
legend('phase(d)','phase(y)')
grid on;
hold off;
subplot 312, plot(abs(signal_x),'.');
hold on;
plot(abs(y),'r');
grid on;
legend('|d|', '|y|')
ylabel('amplitude');
%xlabel('sample(index)');
axis([0 210 0 1.5]);
hold off;
subplot 313, plot(abs(e));
legend('lerror');
grid on;
ylabel('amplitude');
xlabel('sample(index)');
axis([0 210 0 1.5]);
for k = 0:arraylength-1,

```

```

arrayvec(k+1,:) = exp(j*k*Kd*sin(theta));
end;
%calculate response of array
F = w*arrayvec;
figure
plot(((theta/(2*pi))*360), 20*log10(abs(F)));
grid on;
title('amplitude response antennne pattern');
ylabel('(dB)');
xlabel('angle(degrees)');
axis([-90,+90,-55,10]);
hold on;
dbgrens = 40
grid on;
%figure; plot(((theta/(2*pi))*360), phase(F)); %plot phase response if
%desired

```

% 'LMSarray.m'

## **%adaptive array demonstrator LMS algorithm for DBF A**

```

theta_x = 20 %degrees, direction of signal x
theta_x = (2*pi/360)*theta_x
theta_n1 = 0; %degrees, direction of noise source 1
theta_n1 = (2*pi/360)*theta_n1
theta_n2 = -40 %degrees, direction of noise source 2
theta_n2 = (2*pi/360)*theta_n2
theta_n3 = -19 %degrees, direction of noise source 3
theta_n3 = (2*pi/360)*theta_n3
theta_n4 = -82 %degrees, direction of noise source 4
theta_n4 = (2*pi/360)*theta_n4
theta_n5 = 41 %degrees, direction of noise source 5
theta_n5 = (2*pi/360)*theta_n5
theta = pi*[-1:0.005:1];
arraylength = 4; %nr of antennas
bitrate = 100;
fsim = 4*bitrate; %simulation frequency
Ts = 1/fsim; %simulation sample period
message = [1 -1 1 1 1 -1 1 -1 -1 1 1 -1 1 1 -1 1 1 1 -1 -1 -1 1 1 1 1 -1 1 -1 1 1 1 -1 -1 -1 -1
1 -1 1 -1 -1 -1 -1 -1 1 1 1 -1 1 -1 1 1];
message = upsample(message, fsim/bitrate); %upsample message
t = Ts:Ts:(length(message)/fsim); %timeline
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%generate a complex MSK signal
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
mesint=(cumsum(message))/8;
Q = cos(pi*mesint);

```

```

I = sin(pi*mesint);
plot(Q)
hold on;
plot(I,'r')
signal_x = I+j*Q; %the signal to be received
% the undesired complex noise singals-> uniform phase, gaussian amplitude distribution
signal_n1 = normrnd(0,1,1,length(t)).*exp (j*(unifrnd(-pi,pi,1,length(t))));
signal_n2 = normrnd(0,1,1,length(t)).*exp (j*(unifrnd(-pi,pi,1,length(t))));
noise = zeros(arraylength, length(t));
% system noise for every antenna
for i = 0:arraylength-1,
noise(i+1,:) = normrnd(0,0.01,1,length(t)).*exp (j*(unifrnd(-pi,pi,1,length(t))));
end;
%K = 2*pi/ lambda
%d = 0.5*lambda
%Kd means K*d equal to pi
Kd = pi;
% array responses for the desired signal x and noise n1 and n2
arrayvec_x = zeros(1,arraylength);
arrayvec_n1 = zeros(1,arraylength);
arrayvec_n2 = zeros(1,arraylength);
for k = 0:arraylength-1,
arrayvec_x(k+1) = exp(j*k*Kd*sin(theta_x));
end;
for k = 0:arraylength-1,
arrayvec_n1(k+1) = exp(j*k*Kd*sin(theta_n1));
end;
for k = 0:arraylength-1,
arrayvec_n2(k+1) = exp(j*k*Kd*sin(theta_n2));
end;
x = zeros(arraylength, length(t));
n1 = zeros(arraylength, length(t));
n2 = zeros(arraylength, length(t));
% received signal from signal source x
for i = 0:arraylength-1,
x(i+1,:) = signal_x .* arrayvec_x(i+1);
end;
% received signal from noise source n1
for i = 0:arraylength-1,
n1(i+1,:) = signal_n1 .* arrayvec_n1(i+1);
end;
% received signal from noise source n2
for i = 0:arraylength-1,
n2(i+1,:) = signal_n2 .* arrayvec_n2(i+1);
end;
%total received signal is the
%+- -+

```

```

%| signal received at antenna 1 |
%| signal received at antenna 2 |
%| | |
%| signal received at antenna n |
%+- -+
signal_ns = (noise + n1+n2+x);
% define weight vector
w = zeros(1,arraylength);
mu = 0.01;
y = zeros(1,length(t));%output
e = zeros(1,length(t));%error
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%LMS Algorithm
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for repeat=1:1,% use repeat for more convergence on the same message
for i=0:length(t)-1,
y(i+1) = w * signal_ns(:,i+1);
e(i+1) = signal_x(i+1)-y(i+1);
w = w + mu *e(i+1)*(signal_ns(:,i+1))';
end;
end;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Plot all figures
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
close all;
subplot 311, (plot(phase(signal_x),':'))
grid on;
ylabel('phase(rad)');
%xlabel('sample(index)');
axis([0 210 -1 3]);
title('desired signal 25 degrees interferers 0 and -40 degrees')
hold on;
plot(phase(y),'r');
legend('phase(d)','phase(y)')
grid on;
hold off;
subplot 312, plot(abs(signal_x),':');
hold on;
plot(abs(y),'r');
grid on;
legend('|d|', '|y|')
ylabel('amplitude');
%xlabel('sample(index)');

```

```

axis([0 210 0 1.5]);
hold off;
subplot 313, plot(abs(e));
legend('lerror!');
grid on;
ylabel('amplitude');
xlabel('sample(index)');
axis([0 210 0 1.5]);
for k = 0:arraylength-1,
arrayvec(k+1,:) = exp(j*k*Kd*sin(theta));
end;
%calculate response of array
F = w*arrayvec;
figure
plot(((theta/(2*pi))*360), 20*log10(abs(F)));
grid on;
title('amplitude response antennne pattern');
ylabel('(dB)');
xlabel('angle(degrees)');
axis([-90,+90,-55,10]);
hold on;
dbgrens = 40
grid on;
%figure; plot(((theta/(2*pi))*360), phase(F)); %plot phase response if
%desired
%'LMSarray.m'

```

### **%adaptive array demonstrator LMS algorithm for DBF C**

```

theta_x = 20 %degrees, direction of signal x
theta_x = (2*pi/360)*theta_x
theta_n1 = 0; %degrees, direction of noise source 1
theta_n1 = (2*pi/360)*theta_n1
theta_n2 = -40 %degrees, direction of noise source 2
theta_n2 = (2*pi/360)*theta_n2
theta_n3 = -19 %degrees, direction of noise source 3
theta_n3 = (2*pi/360)*theta_n3
theta_n4 = -82 %degrees, direction of noise source 4
theta_n4 = (2*pi/360)*theta_n4
theta_n5 = 41 %degrees, direction of noise source 5
theta_n5 = (2*pi/360)*theta_n5
theta = pi*[-1:0.005:1];
arraylength = 4; %nr of antennas
bitrate = 100;
fsim = 4*bitrate; %simulation frequency
Ts = 1/fsim; %simulation sample period
message = [1 -1 1 1 1 -1 1 -1 -1 -1 1 1 -1 1 -1 1 1 1 -1 -1 -1 1 1 1 1 1 -1 1 -1 1 1 1 -1 -1 -1 -1
1 -1 1 -1 -1 -1 -1 1 1 1 -1 1 -1 1 1 1];
message = upsample(message, fsim/bitrate); %upsample message

```

```

t = Ts:Ts:(length(message)/fsim); %timeline
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%generate a complex MSK signal
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
mesint=(cumsum(message))/8;
Q = cos(pi*mesint);
I = sin(pi*mesint);
plot(Q)
hold on;
plot(I,'r')
signal_x = I+j*Q; %the signal to be received
% the undesired complex noise singals-> uniform phase, gaussian amplitude distribution
signal_n1 = normrnd(0,1,1,length(t)).*exp (j*(unifrnd(-pi,pi,1,length(t))));
signal_n2 = normrnd(0,1,1,length(t)).*exp (j*(unifrnd(-pi,pi,1,length(t))));
noise = zeros(arraylength, length(t));
% system noise for every antenna
for i = 0:arraylength-1,
noise(i+1,:) = normrnd(0,0.01,1,length(t)).*exp (j*(unifrnd(-pi,pi,1,length(t))));
end;
%K = 2*pi/ lambda
%d = 0.5*lambda
%Kd means K*d equal to pi
Kd = pi;
% array responses for the desired signal x and noise n1 and n2
arrayvec_x = zeros(1,arraylength);
arrayvec_n1 = zeros(1,arraylength);
arrayvec_n2 = zeros(1,arraylength);
for k = 0:arraylength-1,
arrayvec_x(k+1) = exp(j*k*Kd*sin(theta_x));
end;
for k = 0:arraylength-1,
arrayvec_n1(k+1) = exp(j*k*Kd*sin(theta_n1));
end;
for k = 0:arraylength-1,
arrayvec_n2(k+1) = exp(j*k*Kd*sin(theta_n2));
end;
x = zeros(arraylength, length(t));
n1 = zeros(arraylength, length(t));
n2 = zeros(arraylength, length(t));
% received signal from signal source x
for i = 0:arraylength-1,
x(i+1,:) = signal_x .* arrayvec_x(i+1);
end;
% received signal from noise source n1
for i = 0:arraylength-1,

```

```

n1(i+1,:) = signal_n1 .* arrayvec_n1(i+1);
end;
% received signal from noise source n2
for i = 0:arraylength-1,
n2(i+1,:) = signal_n2 .* arrayvec_n2(i+1);
end;
%total received signal is the
%+- -+
%| signal received at antenna 1 |
%| signal received at antenna 2 |
%| | |
%| signal received at antenna n |
%+- -+
signal_ns = (noise + n1+n2+x);
% define weight vector
w = zeros(1,arraylength);
mu = 0.01;
y = zeros(1,length(t));%output
e = zeros(1,length(t));%error
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%LMS Algorithm
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for repeat=1:1,% use repeat for more convergence on the same message
for i=0:length(t)-1,
y(i+1) = w * signal_ns(:,i+1);
e(i+1) = signal_x(i+1)-y(i+1);
w = w + mu *e(i+1)*(signal_ns(:,i+1))';
end;
end;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Plot all figures
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
close all;
subplot 311, (plot(phase(signal_x),''))
grid on;
ylabel('phase(rad)');
xlabel('sample(index)');
axis([0 210 -1 3]);
title('desired signal 25 degrees interferers 0 and -40 degrees')
hold on;
plot(phase(y),'r');
legend('phase(d)','phase(y)')
grid on;

```

```

hold off;
subplot 312, plot(abs(signal_x), ':');
hold on;
plot(abs(y), 'r');
grid on;
legend('ldl', 'lyl')
ylabel('amplitude');
%xlabel('sample(index)');
axis([0 210 0 1.5]);
hold off;
subplot 313, plot(abs(e));
legend('lerror!');
grid on;
ylabel('amplitude');
xlabel('sample(index)');
axis([0 210 0 1.5]);
for k = 0:arraylength-1,
arrayvec(k+1,:) = exp(j*k*Kd*sin(theta));
end;
%calculate response of array
F = w*arrayvec;
figure
plot(((theta/(2*pi))*360), 20*log10(abs(F)));
grid on;
title('amplitude response antennne pattern');
ylabel('(dB)');
xlabel('angle(degrees)');
axis([-90,+90,-55,10]);
hold on;
dbgrens = 40
grid on;
%figure; plot(((theta/(2*pi))*360), phase(F)); %plot phase response if
%desired
% 'LMSarray.m'

```

### **%adaptive array demonstrator LMS algorithm for DBF D**

```

theta_x = 20 %degrees, direction of signal x
theta_x = (2*pi/360)*theta_x
theta_n1 = 0; %degrees, direction of noise source 1
theta_n1 = (2*pi/360)*theta_n1
theta_n2 = -40 %degrees, direction of noise source 2
theta_n2 = (2*pi/360)*theta_n2
theta_n3 = -19 %degrees, direction of noise source 3
theta_n3 = (2*pi/360)*theta_n3
theta_n4 = -82 %degrees, direction of noise source 4
theta_n4 = (2*pi/360)*theta_n4
theta_n5 = 41 %degrees, direction of noise source 5
theta_n5 = (2*pi/360)*theta_n5

```

```

theta = pi*[-1:0.005:1];
arraylength = 4; %nr of antennas
bitrate = 100;
fsim = 4*bitrate; %simulation frequency
Ts = 1/fsim; %simulation sample period
message = [1 -1 1 1 1 -1 1 -1 -1 -1 1 1 -1 1 -1 1 1 1 -1 -1 -1 1 1 1 1 -1 1 -1 1 1 1 -1 -1 -1 -1
1 -1 1 -1 -1 -1 -1 -1 1 1 1 -1 1 -1 1 1];
message = upsample(message, fsim/bitrate); %upsample message
t = Ts:Ts:(length(message)/fsim); %timeline
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%generate a complex MSK signal
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
mesint=(cumsum(message))/8;
Q = cos(pi*mesint);
I = sin(pi*mesint);
plot(Q)
hold on;
plot(I,'r')
signal_x = I+j*Q; %the signal to be received
% the undesired complex noise singals-> uniform phase, gaussian amplitude distribution
signal_n1 = normrnd(0,1,1,length(t)).*exp (j*(unifrnd(-pi,pi,1,length(t))));
signal_n2 = normrnd(0,1,1,length(t)).*exp (j*(unifrnd(-pi,pi,1,length(t))));
noise = zeros(arraylength, length(t));
% system noise for every antenna
for i = 0:arraylength-1,
noise(i+1,:) = normrnd(0,0.01,1,length(t)).*exp (j*(unifrnd(-pi,pi,1,length(t))));
end;
%K = 2*pi/ lambda
%d = 0.5*lambda
%Kd means K*d equal to pi
Kd = 1.5*pi;
% array responses for the desired signal x and noise n1 and n2
arrayvec_x = zeros(1,arraylength);
arrayvec_n1 = zeros(1,arraylength);
arrayvec_n2 = zeros(1,arraylength);
for k = 0:arraylength-1,
arrayvec_x(k+1) = exp(j*k*Kd*sin(theta_x));
end;
for k = 0:arraylength-1,
arrayvec_n1(k+1) = exp(j*k*Kd*sin(theta_n1));
end;
for k = 0:arraylength-1,
arrayvec_n2(k+1) = exp(j*k*Kd*sin(theta_n2));
end;
x = zeros(arraylength, length(t));

```

```

n1 = zeros(arraylength, length(t));
n2 = zeros(arraylength, length(t));
% received signal from signal source x
for i = 0:arraylength-1,
x(i+1,:) = signal_x .* arrayvec_x(i+1);
end;
% received signal from noise source n1
for i = 0:arraylength-1,
n1(i+1,:) = signal_n1 .* arrayvec_n1(i+1);
end;
% received signal from noise source n2
for i = 0:arraylength-1,
n2(i+1,:) = signal_n2 .* arrayvec_n2(i+1);
end;
%total received signal is the
%+- -+
%| signal received at antenna 1 |
%| signal received at antenna 2 |
%| | |
%| signal received at antenna n |
%+- -+
signal_ns = (noise + n1+n2+x);
% define weight vector
w = zeros(1,arraylength);
mu = 0.01;
y = zeros(1,length(t));%output
e = zeros(1,length(t));%error
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%LMS Algorithm
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for repeat=1:1,% use repeat for more convergence on the same message
for i=0:length(t)-1,
y(i+1) = w * signal_ns(:,i+1);
e(i+1) = signal_x(i+1)-y(i+1);
w = w + mu *e(i+1)*(signal_ns(:,i+1))';
end;
end;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Plot all figures
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
close all;
subplot 311, (plot(phase(signal_x),''))
grid on;

```

```

ylabel('phase(rad)');
xlabel('sample(index)');
axis([0 210 -1 3]);
title('desired signal 25 degrees interferers 0 and -40 degrees')
hold on;
plot(phase(y),'r');
legend('phase(d)','phase(y)')
grid on;
hold off;
subplot 312, plot(abs(signal_x), ':');
hold on;
plot(abs(y),'r');
grid on;
legend('ldl', 'lyl')
ylabel('amplitude');
xlabel('sample(index)');
axis([0 210 0 1.5]);
hold off;
subplot 313, plot(abs(e));
legend('error!');
grid on;
ylabel('amplitude');
xlabel('sample(index)');
axis([0 210 0 1.5]);
for k = 0:arraylength-1,
    arrayvec(k+1,:) = exp(j*k*Kd*sin(theta));
end;
%calculate response of array
F = w*arrayvec;
figure
plot(((theta/(2*pi))*360), 20*log10(abs(F)));
grid on;
title('amplitude response antennne pattern');
ylabel('(dB)');
xlabel('angle(degrees)');
axis([-90,+90,-55,10]);
hold on;
dbgrens = 40
grid on;
%figure; plot(((theta/(2*pi))*360), phase(F)); %plot phase response if
%desired

```

% 'LMSarray.m'

### **%adaptive array demonstrator LMS algorithm for DBF E**

```

theta_x = 20 %degrees, direction of signal x
theta_x = (2*pi/360)*theta_x

```

```

theta_n1 = 0; %degrees, direction of noise source 1
theta_n1 = (2*pi/360)*theta_n1
theta_n2 = -40 %degrees, direction of noise source 2
theta_n2 = (2*pi/360)*theta_n2
theta_n3 = -19 %degrees, direction of noise source 3
theta_n3 = (2*pi/360)*theta_n3
theta_n4 = -82 %degrees, direction of noise source 4
theta_n4 = (2*pi/360)*theta_n4
theta_n5 = 41 %degrees, direction of noise source 5
theta_n5 = (2*pi/360)*theta_n5
theta = pi*[-1:0.005:1];
arraylength = 4; %nr of antennas
bitrate = 100;
fsim = 4*bitrate; %simulation frequency
Ts = 1/fsim; %simulation sample period
message = [1 -1 1 1 1 -1 1 -1 -1 -1 1 1 -1 1 1 1 1 -1 -1 -1 1 1 1 1 1 -1 1 -1 1 1 1 -1 -1 -1 -1
1 -1 1 1 -1 -1 -1 -1 1 1 1 1 -1 1 1 1 1];
message = upsample(message, fsim/bitrate); %upsample message
t = Ts:Ts:(length(message)/fsim); %timeline
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%generate a complex MSK signal
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
mesint=(cumsum(message))/8;
Q = cos(pi*mesint);
I = sin(pi*mesint);
plot(Q)
hold on;
plot(I,'r')
signal_x = I+j*Q; %the signal to be received
% the undesired complex noise singals-> uniform phase, gaussian amplitude distribution
signal_n1 = normrnd(0,1,1,length(t)).*exp (j*(unifrnd(-pi,pi,1,length(t))));
signal_n2 = normrnd(0,1,1,length(t)).*exp (j*(unifrnd(-pi,pi,1,length(t))));
noise = zeros(arraylength, length(t));
% system noise for every antenna
for i = 0:arraylength-1,
noise(i+1,:) = normrnd(0,0.1,1,length(t)).*exp (j*(unifrnd(-pi,pi,1,length(t))));
end;
%K = 2*pi/ lambda
%d = 0.5*lambda
%Kd means K*d equal to pi
Kd = 1.5*pi;
% array responses for the desired signal x and noise n1 and n2
arrayvec_x = zeros(1,arraylength);
arrayvec_n1 = zeros(1,arraylength);
arrayvec_n2 = zeros(1,arraylength);

```

```

for k = 0:arraylength-1,
arrayvec_x(k+1) = exp(j*k*Kd*sin(theta_x));
end;
for k = 0:arraylength-1,
arrayvec_n1(k+1) = exp(j*k*Kd*sin(theta_n1));
end;
for k = 0:arraylength-1,
arrayvec_n2(k+1) = exp(j*k*Kd*sin(theta_n2));
end;
x = zeros(arraylength, length(t));
n1 = zeros(arraylength, length(t));
n2 = zeros(arraylength, length(t));
% received signal from signal source x
for i = 0:arraylength-1,
x(i+1,:) = signal_x .* arrayvec_x(i+1);
end;
% received signal from noise source n1
for i = 0:arraylength-1,
n1(i+1,:) = signal_n1 .* arrayvec_n1(i+1);
end;
% received signal from noise source n2
for i = 0:arraylength-1,
n2(i+1,:) = signal_n2 .* arrayvec_n2(i+1);
end;
%total received signal is the
%+- -+
%| signal received at antenna 1 |
%| signal received at antenna 2 |
%| |
%| signal received at antenna n |
%+- -+
signal_ns = (noise + n1+n2+x);
% define weight vector
w = zeros(1,arraylength);
mu = 0.01;
y = zeros(1,length(t));%output
e = zeros(1,length(t));%error
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%LMS Algorithm
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for repeat=1:1,% use repeat for more convergence on the same message
for i=0:length(t)-1,
y(i+1) = w * signal_ns(:,i+1);
e(i+1) = signal_x(i+1)-y(i+1);
w = w + mu *e(i+1)*(signal_ns(:,i+1))';

```

```

end;
end;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Plot all figures
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
close all;
subplot 311, (plot(phase(signal_x),':'))
grid on;
ylabel('phase(rad)');
%xlabel('sample(index)');
axis([0 210 -1 3]);
title('desired signal 25 degrees interferers 0 and -40 degrees')
hold on;
plot(phase(y),'r');
legend('phase(d)','phase(y)')
grid on;
hold off;
subplot 312, plot(abs(signal_x),':');
hold on;
plot(abs(y),'r');
grid on;
legend('ldl', 'lyl')
ylabel('amplitude');
%xlabel('sample(index)');
axis([0 210 0 1.5]);
hold off;
subplot 313, plot(abs(e));
legend('error!');
grid on;
ylabel('amplitude');
xlabel('sample(index)');
axis([0 210 0 1.5]);
for k = 0:arraylength-1,
arrayvec(k+1,:) = exp(j*k*Kd*sin(theta));
end;
%calculate response of array
F = w*arrayvec;
figure
plot(((theta/(2*pi))*360), 20*log10(abs(F)));
grid on;
title('amplitude response antennne pattern');
ylabel('(dB)');
xlabel('angle(degrees)');
axis([-90,+90,-55,10]);
hold on;

```

```

dbgrens = 40
grid on;
%figure; plot(((theta/(2*pi))*360), phase(F)); %plot phase response if
%desired
%'LMSarray.m'

```

## **%adaptive array demonstrator LMS algorithm for DBF F**

```

theta_x = 20 %degrees, direction of signal x
theta_x = (2*pi/360)*theta_x
theta_n1 = 0; %degrees, direction of noise source 1
theta_n1 = (2*pi/360)*theta_n1
theta_n2 = -40 %degrees, direction of noise source 2
theta_n2 = (2*pi/360)*theta_n2
theta_n3 = -19 %degrees, direction of noise source 3
theta_n3 = (2*pi/360)*theta_n3
theta_n4 = -82 %degrees, direction of noise source 4
theta_n4 = (2*pi/360)*theta_n4
theta_n5 = 41 %degrees, direction of noise source 5
theta_n5 = (2*pi/360)*theta_n5
theta = pi*[-1:0.005:1];
arraylength = 4; %nr of antennas
bitrate = 100;
fsim = 4*bitrate; %simulation frequency
Ts = 1/fsim; %simulation sample period
message = [1 -1 1 1 1 -1 1 -1 -1 -1 1 1 -1 1 -1 1 1 1 -1 -1 -1 1 1 1 1 1 -1 1 -1 1 1 1 -1 -1 -1 -1
1 -1 1 -1 -1 -1 -1 1 1 1 -1 1 -1 1 1];
message = upsample(message, fsim/bitrate); %upsample message
t = Ts:Ts:(length(message)/fsim); %timeline
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%generate a complex MSK signal
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
mesint=(cumsum(message))/8;
Q = cos(pi*mesint);
I = sin(pi*mesint);
plot(Q)
hold on;
plot(I,'r')
signal_x = I+j*Q; %the signal to be received
% the undesired complex noise singals-> uniform phase, gaussian amplitude distribution
signal_n1 = normrnd(0,1,1,length(t)).*exp (j*(unifrnd(-pi,pi,1,length(t))));
signal_n2 = normrnd(0,1,1,length(t)).*exp (j*(unifrnd(-pi,pi,1,length(t))));
noise = zeros(arraylength, length(t));
% system noise for every antenna
for i = 0:arraylength-1,
noise(i+1,:) = normrnd(0,0.01,1,length(t)).*exp (j*(unifrnd(-pi,pi,1,length(t))));

```

```

end;
%K = 2*pi/ lambda
%d = 0.5*lambda
%Kd means K*d equal to pi
Kd = 2*pi;
% array responses for the desired signal x and noise n1 and n2
arrayvec_x = zeros(1,arraylength);
arrayvec_n1 = zeros(1,arraylength);
arrayvec_n2 = zeros(1,arraylength);
for k = 0:arraylength-1,
arrayvec_x(k+1) = exp(j*k*Kd*sin(theta_x));
end;
for k = 0:arraylength-1,
arrayvec_n1(k+1) = exp(j*k*Kd*sin(theta_n1));
end;
for k = 0:arraylength-1,
arrayvec_n2(k+1) = exp(j*k*Kd*sin(theta_n2));
end;
x = zeros(arraylength, length(t));
n1 = zeros(arraylength, length(t));
n2 = zeros(arraylength, length(t));
% received signal from signal source x
for i = 0:arraylength-1,
x(i+1,:) = signal_x .* arrayvec_x(i+1);
end;
% received signal from noise source n1
for i = 0:arraylength-1,
n1(i+1,:) = signal_n1 .* arrayvec_n1(i+1);
end;
% received signal from noise source n2
for i = 0:arraylength-1,
n2(i+1,:) = signal_n2 .* arrayvec_n2(i+1);
end;
%total received signal is the
%+- -+
%| signal received at antenna 1 |
%| signal received at antenna 2 |
%| | |
%| signal received at antenna n |
%+- -+
signal_ns = (noise + n1+n2+x);
% define weight vector
w = zeros(1,arraylength);
mu = 0.01;
y = zeros(1,length(t));%output
e = zeros(1,length(t));%error

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%LMS Algorithm
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for repeat=1:1,% use repeat for more convergence on the same message
for i=0:length(t)-1,
y(i+1) = w * signal_ns(:,i+1);
e(i+1) = signal_x(i+1)-y(i+1);
w = w + mu *e(i+1)*(signal_ns(:,i+1))';
end;
end;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Plot all figures
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
close all;
subplot 311, (plot(phase(signal_x),':'))
grid on;
ylabel('phase(rad)');
%xlabel('sample(index)');
axis([0 210 -1 3]);
title('desired signal 25 degrees interferers 0 and -40 degrees')
hold on;
plot(phase(y),'r');
legend('phase(d)','phase(y)')
grid on;
hold off;
subplot 312, plot(abs(signal_x),':');
hold on;
plot(abs(y),'r');
grid on;
legend('ldl', 'lyl')
ylabel('amplitude');
%xlabel('sample(index)');
axis([0 210 0 1.5]);
hold off;
subplot 313, plot(abs(e));
legend('error!');
grid on;
ylabel('amplitude');
xlabel('sample(index)');
axis([0 210 0 1.5]);
for k = 0:arraylength-1,
arrayvec(k+1,:) = exp(j*k*Kd*sin(theta));
end;

```

```

%calculate response of array
F = w*arrayvec;
figure
plot(((theta/(2*pi))*360), 20*log10(abs(F)));
grid on;
title('amplitude response antenne pattern');
ylabel('(dB)');
xlabel('angle(degrees)');
axis([-90,+90,-55,10]);
hold on;
dbgrems = 40
grid on;
%figure; plot(((theta/(2*pi))*360), phase(F)); %plot phase response if
%desired

%FILE CMAarray.m
%adaptive array demonstrator CM algorithm for Digital Beamforming 1
typeofinterferer1='noise'; %set to 'noise' for guassian type of interferer
%set to 'signl' for MSK type of interferer
theta_x = 10 %degrees
theta_x = (2*pi/360)*theta_x
theta_n1 = -10; %degrees
theta_n1 = (2*pi/360)*theta_n1
theta_n2 = -40 %degrees
theta_n2 = (2*pi/360)*theta_n2
theta = pi*[-1:0.005:1];
arraylength = 4; %nr of antennas
bitrate = 200;
fsim = 4*bitrate; %simulation frequency
Ts = 1/fsim;
message = [1 -1 1 1 1 -1 1 -1 -1 -1 1 1 -1 1 1 1 -1 -1 -1 1 1 1 1 -1 1 -1 1 1 1 -1 -1 -1 -1
1 -1 1 -1 -1 -1 -1 1 1 1 -1 1 -1 1 1];
message = upsample(message, fsim/bitrate); %upsample message
t = Ts:Ts:(length(message)/fsim); %timeline
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%generate a complex MSK signal
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
mesint=(cumsum(message))/8;
Q = cos(pi*mesint);
I = sin(pi*mesint);
hold on;
signal_x = I+j*Q; %the signal to be received
signal_ref = signal_x(1:end);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

%generate 2 interferers n1 and n2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
% if the interfere n1 needs to be gaussian noise -> uniform phase, normal amplitude
distribution
if (typeofinterferer1 == 'noise'),
    signal_n1 = normrnd(0,1,1,length(t)).*exp (j*(unifrnd(-pi,pi,1,length(t))));
end;
% if testing with an MSK signal as interferer is required: noise source n1 will be MSK
signal
if (typeofinterferer1 == 'signal'),
    message_n1 = [-1 -1 1 -1 1 -1 -1 1 -1 -1 -1 1 1 -1 -1 1 1 1 1 -1 1 -1 1 -1 1 -1 -1 -1 1 1 -1 1 -
    1 -1 -1 1 -1 1 -1 -1 -1 -1 1 1 1 -1 1 -1 1 -1 -1];
    message_n1 = upsample(message_n1, fsim/bitrate); %upsample message
    t = Ts:Ts:(length(message)/fsim); %timeline
    mesint_n1=(cumsum(message_n1))/8;
    Qn1 = cos(pi*mesint_n1);
    In1 = sin(pi*mesint_n1);
    signal_n1 = In1+j*Qn1; %the signal to be received
end;
%noise source 2 is gaussian
signal_n2 = normrnd(0,1,1,length(t)).*exp (j*(unifrnd(-pi,pi,1,length(t))));
noise = zeros(arraylength, length(t));
% system noise for every antenna
for i = 0:arraylength-1,
    noise(i+1,:) = normrnd(0,0.1,1,length(t)).*exp (j*(unifrnd(-pi,pi,1,length(t))));
end;
%lambda = sym('lambda');
%K = 2*pi/ lambda
%d = 0.5*lambda
Kd = pi;
%alpha = -Kd*sin(theta_nul);
% array responses for the desired signal x and noise n1 and n2
arrayvec_x = zeros(1,arraylength);
arrayvec_n1 = zeros(1,arraylength);
arrayvec_n2 = zeros(1,arraylength);
for k = 0:arraylength-1,
    arrayvec_x(k+1) = exp(j*k*Kd*sin(theta_x));
end;
for k = 0:arraylength-1,
    arrayvec_n1(k+1) = exp(j*k*Kd*sin(theta_n1));
end;
for k = 0:arraylength-1,
    arrayvec_n2(k+1) = exp(j*k*Kd*sin(theta_n2));
end;
x = zeros(arraylength, length(t));
n1 = zeros(arraylength, length(t));

```

```

n2 = zeros(arraylength, length(t));
% received signal from signal source x
for i = 0:arraylength-1,
x(i+1,:) = signal_x .* arrayvec_x(i+1);
end;
% received signal from noise source n1
for i = 0:arraylength-1,
n1(i+1,:) = signal_n1 .* arrayvec_n1(i+1);
end;
% received signal from noise source n2
for i = 0:arraylength-1,
n2(i+1,:) = signal_n2 .* arrayvec_n2(i+1);
end;
%total received signal is the
%+- -+
%| signal received at antenna 1 |
%| signal received at antenna 2 |
%| | |
%| signal received at antenna n |
%+- -+
w = zeros(1,arraylength);
signal_ns = (noise + n1+n2+x);
%start weight vector, may be in the direction of the desired signal:
w = (1/4)*[1, exp(-j*pi*sin(theta_x)),exp(-j*2*pi*sin(theta_x)),exp(-j*3*pi*sin(theta_x))];
%start vector may be any type:
%w = [0.1 0.1 0.1 0.1]
wstart = w;
%mu = 0.007;
mu = 0.001;
y = zeros(1,length(t));
e = zeros(1,length(t));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%Constant Modulus Algorithm
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
for repeat=1:1,
for i=0:length(signal_ref)-1,
y(i+1) = w * signal_ns(:,i+1);
%e(i+1) = 1 - y(i+1)
e(i+1) = y(i+1)/((abs(y(i+1))))^2-y(i+1); %SATO's principle for updating the error
w = w + mu*e(i+1)*(signal_ns(:,i+1))';
end;
end;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%Plot all figures

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
close all;
subplot 311, (plot(phase(signal_x),':'))
ylabel('phase(rad)');
title('desired signal 10 degrees interferers -10 and -40 degrees')
grid on;
axis([0 210 -2 2])
hold on;
plot(phase(y),'r');
legend('phase(d)','phase(y)')
hold off;
subplot 312, plot(abs(signal_x),':');
hold on;
grid on;
axis([0 210 0 2])
plot(abs(y),'r');
legend('ldl', 'lyl')
ylabel('amplitude');
%xlabel('sample(index)');
hold off;
subplot 313, plot(abs(e));
grid on;
legend('lerrorl');
ylabel('amplitude');
xlabel('sample(index)');
axis([0 210 0 1]);
for k = 0:arraylength-1
arrayvec(k+1,:) = exp(j*k*Kd*sin(theta));
end;
%calculate response of array
F = w*arrayvec;
% F2 = wstart*arrayvec;
figure
plot(((theta/(2*pi))*360), 20*log10(abs(F)));
%hold on;
%plot(((theta/(2*pi))*360), 20*log10(abs(F2)), 'r');
title('amplitude response antenne, desired signal: 10 degrees, interferers: -10 and -40
degrees');
ylabel('(dB)');
xlabel('angle(degrees)');
axis([-90,+90,-30,10]);
grid on;
hold on;
dbgrens = 40

%FILE CMAarray.m

```

## **%adaptive array demonstrator CM algorithm for Digital Beamforming 2**

```

typeofinterferer1='noise'; %set to 'noise' for gaussian type of interferer
%set to 'signl' for MSK type of interferer
theta_x = 10 %degrees
theta_x = (2*pi/360)*theta_x
theta_n1 = -10; %degrees
theta_n1 = (2*pi/360)*theta_n1
theta_n2 = -40 %degrees
theta_n2 = (2*pi/360)*theta_n2
theta = pi*[-1:0.005:1];
arraylength = 4; %nr of antennas
bitrate = 200;
fsim = 4*bitrate; %simulation frequency
Ts = 1/fsim;
message = [1 -1 1 1 1 -1 1 -1 -1 -1 1 1 -1 1 1 1 1 -1 -1 -1 1 1 1 1 1 -1 1 -1 1 1 1 -1 -1 -1 -1
1 -1 1 -1 -1 -1 -1 -1 1 1 1 1 -1 1 -1 1 1];
message = upsample(message, fsim/bitrate); %upsample message
t = Ts:Ts:(length(message)/fsim); %timeline
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%generate a complex MSK signal
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
mesint=(cumsum(message))/8;
Q = cos(pi*mesint);
I = sin(pi*mesint);
hold on;
signal_x = I+j*Q; %the signal to be received
signal_ref = signal_x(1:end);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%generate 2 interferers n1 and n2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% if the interfere n1 needs to be gaussian noise -> uniform phase, normal amplitude
distribution
if (typeofinterferer1 == 'noise'),
    signal_n1 = normrnd(0,1,1,length(t)).*exp (j*(unifrnd(-pi,pi,1,length(t))));
end;
% if testing with an MSK signal as interferer is required: noise source n1 will be MSK
signal
if (typeofinterferer1 == 'signl'),
    message_n1 = [-1 -1 1 1 -1 1 -1 -1 1 -1 -1 1 1 1 -1 -1 1 1 1 1 -1 1 -1 1 -1 1 -1 -1 -1 1 1 -1 1 -
1 -1 -1 1 1 -1 -1 -1 -1 1 1 1 1 -1 1 -1 1 -1];
    message_n1 = upsample(message_n1, fsim/bitrate); %upsample message
    t = Ts:Ts:(length(message)/fsim); %timeline
    mesint_n1=(cumsum(message_n1))/8;

```

```

Qn1 = cos(pi*mesint_n1);
In1 = sin(pi*mesint_n1);
signal_n1 = In1+j*Qn1; %the signal to be received
end;
%noise source 2 is gaussian
signal_n2 = normrnd(0,1,1,length(t)).*exp (j*(unifrnd(-pi,pi,1,length(t))));
noise = zeros(arraylength, length(t));
% system noise for every antenna
for i = 0:arraylength-1,
noise(i+1,:) = normrnd(0,0.01,1,length(t)).*exp (j*(unifrnd(-pi,pi,1,length(t))));
end;
%lambda = sym('lambda');
%K = 2*pi/ lambda
%d = 0.5*lambda
Kd = pi;
%alpha = -Kd*sin(theta_nul);
% array responses for the desired signal x and noise n1 and n2
arrayvec_x = zeros(1,arraylength);
arrayvec_n1 = zeros(1,arraylength);
arrayvec_n2 = zeros(1,arraylength);
for k = 0:arraylength-1,
arrayvec_x(k+1) = exp(j*k*Kd*sin(theta_x));
end;
for k = 0:arraylength-1,
arrayvec_n1(k+1) = exp(j*k*Kd*sin(theta_n1));
end;
for k = 0:arraylength-1,
arrayvec_n2(k+1) = exp(j*k*Kd*sin(theta_n2));
end;
x = zeros(arraylength, length(t));
n1 = zeros(arraylength, length(t));
n2 = zeros(arraylength, length(t));
% received signal from signal source x
for i = 0:arraylength-1,
x(i+1,:) = signal_x .* arrayvec_x(i+1);
end;
% received signal from noise source n1
for i = 0:arraylength-1,
n1(i+1,:) = signal_n1 .* arrayvec_n1(i+1);
end;
% received signal from noise source n2
for i = 0:arraylength-1,
n2(i+1,:) = signal_n2 .* arrayvec_n2(i+1);
end;
%total received signal is the
%+- -+
%| signal received at antenna 1 |

```

```

%| signal received at antenna 2 |
%| | |
%| signal received at antenna n |
%+- -+
w = zeros(1,arraylength);
signal_ns = (noise + n1+n2+x);
%start weight vector, may be in the direction of the desired signal:
w = (1/4)*[1, exp(-j*pi*sin(theta_x)),exp(-j*2*pi*sin(theta_x)),exp(-j*3*pi*sin(theta_x))];
%start vector may be any type:
%w = [0.1 0.1 0.1 0.1]
wstart = w;
%mu = 0.007;
mu = 0.001;
y = zeros(1,length(t));
e = zeros(1,length(t));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%Constant Modulus Algorithm
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
for repeat=1:1,
for i=0:length(signal_ref)-1,
y(i+1) = w * signal_ns(:,i+1);
%e(i+1) = 1 - y(i+1)
e(i+1) = y(i+1)/((abs(y(i+1))))^2-y(i+1); %SATO's principle for updating the error
w = w + mu*e(i+1)*(signal_ns(:,i+1));
end;
end;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%Plot all figures
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
close all;
subplot 311, (plot(phase(signal_x),':'))
ylabel('phase(rad)');
title('desired signal 10 degrees interferers -10 and -40 degrees')
grid on;
axis([0 210 -2 2])
hold on;
plot(phase(y),'r');
legend('phase(d)','phase(y)')
hold off;
subplot 312, plot(abs(signal_x),':');
hold on;
grid on;
axis([0 210 0 2])

```

```

plot(abs(y),'r');
legend('ldl', 'lyl')
ylabel('amplitude');
%xlabel('sample(index)');
hold off;
subplot 313, plot(abs(e));
grid on;
legend('lerror!');
ylabel('amplitude');
xlabel('sample(index)');
axis([0 210 0 1]);
for k = 0:arraylength-1
arrayvec(k+1,:) = exp(j*k*Kd*sin(theta));
end;
%calculate response of array
F = w*arrayvec;
% F2 = wstart*arrayvec;
figure
plot(((theta/(2*pi))*360), 20*log10(abs(F)));
%hold on;
%plot(((theta/(2*pi))*360), 20*log10(abs(F2)), 'r');
title('amplitude response antennae, desired signal: 10 degrees, interferers: -10 and -40
degrees');
ylabel('(dB)');
xlabel('angle(degrees)');
axis([-90,+90,-30,10]);
grid on;
hold on;
dbgrens = 40

```

```

%FILE CMAarray.m

```

### **%adaptive array demonstrator CM algorithm for Digital Beamforming 3**

```

typeofinterferer1='noise'; %set to 'noise' for gaussian type of interferer

```

```

%set to 'signl' for MSK type of interferer

```

```

theta_x = 10 %degrees

```

```

theta_x = (2*pi/360)*theta_x

```

```

theta_n1 = -10; %degrees

```

```

theta_n1 = (2*pi/360)*theta_n1

```

```

theta_n2 = -40 %degrees

```

```

theta_n2 = (2*pi/360)*theta_n2

```

```

theta = pi*[-1:0.005:1];

```

```

arraylength = 4; %nr of antennas

```

```

bitrate = 200;

```

```

fsim = 4*bitrate; %simulation frequency

```

```

Ts = 1/fsim;

```

```

message = [1 -1 1 1 1 -1 1 -1 -1 -1 1 1 -1 1 -1 1 1 1 -1 -1 -1 1 1 1 1 -1 1 -1 1 1 1 -1 -1 -1 -1
1 -1 1 1 -1 -1 -1 -1 1 1 1 -1 1 -1 1 1 1];

```

```

message = upsample(message, fsim/bitrate); %upsample message
t = Ts:Ts:(length(message)/fsim); %timeline
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%generate a complex MSK signal
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
mesint=(cumsum(message))/8;
Q = cos(pi*mesint);
I = sin(pi*mesint);
hold on;
signal_x = I+j*Q; %the signal to be received
signal_ref = signal_x(1:end);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%generate 2 interferers n1 and n2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% if the interfere n1 needs to be gaussian noise -> uniform phase, normal amplitude
distribution
if (typeofinterferer1 == 'noise'),
signal_n1 = normrnd(0,1,1,length(t)).*exp (j*(unifrnd(-pi,pi,1,length(t))));
end;
% if testing with an MSK signal as interferer is required: noise source n1 will be MSK
signal
if (typeofinterferer1 == 'signl'),
message_n1 = [-1 -1 1 -1 1 -1 -1 1 -1 -1 -1 1 1 -1 -1 1 1 1 1 -1 1 -1 1 -1 1 -1 -1 -1 1 1 -1 1 -
1 -1 -1 1 -1 1 -1 -1 -1 1 1 1 -1 1 -1 1 -1 -1];
message_n1 = upsample(message_n1, fsim/bitrate); %upsample message
t = Ts:Ts:(length(message)/fsim); %timeline
mesint_n1=(cumsum(message_n1))/8;
Qn1 = cos(pi*mesint_n1);
In1 = sin(pi*mesint_n1);
signal_n1 = In1+j*Qn1; %the signal to be received
end;
%noise source 2 is gaussian
signal_n2 = normrnd(0,1,1,length(t)).*exp (j*(unifrnd(-pi,pi,1,length(t))));
noise = zeros(arraylength, length(t));
% system noise for every antenna
for i = 0:arraylength-1,
noise(i+1,:) = normrnd(0,0.01,1,length(t)).*exp (j*(unifrnd(-pi,pi,1,length(t))));
end;
%lambda = sym('lambda');
%K = 2*pi/ lambda
%d = 0.5*lambda
Kd = 1.5*pi;
%alpha = -Kd*sin(theta_nul);

```

```

% array responses for the desired signal x and noise n1 and n2
arrayvec_x = zeros(1,arraylength);
arrayvec_n1 = zeros(1,arraylength);
arrayvec_n2 = zeros(1,arraylength);
for k = 0:arraylength-1,
arrayvec_x(k+1) = exp(j*k*Kd*sin(theta_x));
end;
for k = 0:arraylength-1,
arrayvec_n1(k+1) = exp(j*k*Kd*sin(theta_n1));
end;
for k = 0:arraylength-1,
arrayvec_n2(k+1) = exp(j*k*Kd*sin(theta_n2));
end;
x = zeros(arraylength, length(t));
n1 = zeros(arraylength, length(t));
n2 = zeros(arraylength, length(t));
% received signal from signal source x
for i = 0:arraylength-1,
x(i+1,:) = signal_x .* arrayvec_x(i+1);
end;
% received signal from noise source n1
for i = 0:arraylength-1,
n1(i+1,:) = signal_n1 .* arrayvec_n1(i+1);
end;
% received signal from noise source n2
for i = 0:arraylength-1,
n2(i+1,:) = signal_n2 .* arrayvec_n2(i+1);
end;
%total received signal is the
%+- -+
%| signal received at antenna 1 |
%| signal received at antenna 2 |
%| |
%| signal received at antenna n |
%+- -+
w = zeros(1,arraylength);
signal_ns = (noise + n1+n2+x);
%start weight vector, may be in the direction of the desired signal:
w = (1/4)*[1, exp(-j*pi*sin(theta_x)),exp(-j*2*pi*sin(theta_x)),exp(-j*3*pi*sin(theta_x))];
%start vector may be any type:
%w = [0.1 0.1 0.1 0.1]
wstart = w;
%mu = 0.007;
mu = 0.001;
y = zeros(1,length(t));
e = zeros(1,length(t));

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%Constant Modulus Algorithm
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
for repeat=1:1,
for i=0:length(signal_ref)-1,
y(i+1) = w * signal_ns(:,i+1);
%e(i+1) = 1 - y(i+1)
e(i+1) = y(i+1)/((abs(y(i+1))))^2-y(i+1); %SATO's principle for updating the error
w = w + mu*e(i+1)*(signal_ns(:,i+1))';
end;
end;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%Plot all figures
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
close all;
subplot 311, (plot(phase(signal_x),':'))
ylabel('phase(rad)');
title('desired signal 10 degrees interferers -10 and -40 degrees')
grid on;
axis([0 210 -2 2])
hold on;
plot(phase(y),'r');
legend('phase(d)','phase(y)')
hold off;
subplot 312, plot(abs(signal_x),':');
hold on;
grid on;
axis([0 210 0 2])
plot(abs(y),'r');
legend('ldl', 'lyl')
ylabel('amplitude');
%xlabel('sample(index)');
hold off;
subplot 313, plot(abs(e));
grid on;
legend('errorl');
ylabel('amplitude');
xlabel('sample(index)');
axis([0 210 0 1]);
for k = 0:arraylength-1
arrayvec(k+1,:) = exp(j*k*Kd*sin(theta));
end;
%calculate response of array

```

```

F = w*arrayvec;
% F2 = wstart*arrayvec;
figure
plot(((theta/(2*pi))*360), 20*log10(abs(F)));
%hold on;
%plot(((theta/(2*pi))*360), 20*log10(abs(F2)), 'r');
title('amplitude response antenne, desired signal: 10 degrees, interferers: -10 and -40
degrees');
ylabel('(dB)');
xlabel('angle(degrees)');
axis([-90,+90,-30,10]);
grid on;
hold on;
dbgrens = 40
%FILE CMAarray.m
%adaptive array demonstrator CM algorithm for Digital Beamforming 4
typeofinterferer1='noise'; %set to 'noise' for gaussian type of interferer
%set to 'signl' for MSK type of interferer
theta_x = 10 %degrees
theta_x = (2*pi/360)*theta_x
theta_n1 = -10; %degrees
theta_n1 = (2*pi/360)*theta_n1
theta_n2 = -40 %degrees
theta_n2 = (2*pi/360)*theta_n2
theta = pi*[-1:0.005:1];
arraylength = 4; %nr of antennas
bitrate = 200;
fsim = 4*bitrate; %simulation frequency
Ts = 1/fsim;
message = [1 -1 1 1 1 -1 1 -1 -1 1 1 -1 1 1 -1 1 -1 1 1 1 1 -1 1 -1 1 1 1 1 -1 -1 -1 -1
1 -1 1 -1 -1 -1 -1 1 1 1 -1 1 -1 1 1];
message = upsample(message, fsim/bitrate); %upsample message
t = Ts:Ts:(length(message)/fsim); %timeline
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%generate a complex MSK signal
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
mesint=(cumsum(message))/8;
Q = cos(pi*mesint);
I = sin(pi*mesint);
hold on;
signal_x = I+j*Q; %the signal to be received
signal_ref = signal_x(1:end);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%generate 2 interferers n1 and n2

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% if the interfere n1 needs to be gaussian noise -> uniform phase, normal amplitude
distribution
if (typeofinterferer1 == 'noise'),
    signal_n1 = normrnd(0,1,1,length(t)).*exp (j*(unifrnd(-pi,pi,1,length(t))));
end;
% if testing with an MSK signal as interferer is required: noise source n1 will be MSK
signal
if (typeofinterferer1 == 'signl'),
    message_n1 = [-1 -1 1 -1 1 -1 -1 1 -1 -1 -1 1 1 -1 -1 1 1 1 1 -1 1 -1 1 -1 1 -1 -1 -1 1 1 -1 1 -
    1 -1 -1 1 -1 1 -1 -1 -1 -1 1 1 1 -1 1 -1 1 -1 -1 -1 1 1 -1 1 -];
    message_n1 = upsample(message_n1, fsim/bitrate); %upsample message
    t = Ts:Ts:(length(message)/fsim); %timeline
    mesint_n1=(cumsum(message_n1))/8;
    Qn1 = cos(pi*mesint_n1);
    In1 = sin(pi*mesint_n1);
    signal_n1 = In1+j*Qn1; %the signal to be received
end;
%noise source 2 is gaussian
signal_n2 = normrnd(0,1,1,length(t)).*exp (j*(unifrnd(-pi,pi,1,length(t))));
noise = zeros(arraylength, length(t));
% system noise for every antenna
for i = 0:arraylength-1,
    noise(i+1,:) = normrnd(0,0.1,1,length(t)).*exp (j*(unifrnd(-pi,pi,1,length(t))));
end;
%lambda = sym('lambda');
%K = 2*pi/ lambda
%d = 0.5*lambda
Kd = 1.5*pi;
%alpha = -Kd*sin(theta_nul);
% array responses for the desired signal x and noise n1 and n2
arrayvec_x = zeros(1,arraylength);
arrayvec_n1 = zeros(1,arraylength);
arrayvec_n2 = zeros(1,arraylength);
for k = 0:arraylength-1,
    arrayvec_x(k+1) = exp(j*k*Kd*sin(theta_x));
end;
for k = 0:arraylength-1,
    arrayvec_n1(k+1) = exp(j*k*Kd*sin(theta_n1));
end;
for k = 0:arraylength-1,
    arrayvec_n2(k+1) = exp(j*k*Kd*sin(theta_n2));
end;
x = zeros(arraylength, length(t));
n1 = zeros(arraylength, length(t));
n2 = zeros(arraylength, length(t));

```

```

% received signal from signal source x
for i = 0:arraylength-1,
x(i+1,:) = signal_x .* arrayvec_x(i+1);
end;
% received signal from noise source n1
for i = 0:arraylength-1,
n1(i+1,:) = signal_n1 .* arrayvec_n1(i+1);
end;
% received signal from noise source n2
for i = 0:arraylength-1,
n2(i+1,:) = signal_n2 .* arrayvec_n2(i+1);
end;
%total received signal is the
%+- -+
%| signal received at antenna 1 |
%| signal received at antenna 2 |
%| | |
%| signal received at antenna n |
%+- -+
w = zeros(1,arraylength);
signal_ns = (noise + n1+n2+x);
%start weight vector, may be in the direction of the desired signal:
w = (1/4)*[1, exp(-j*pi*sin(theta_x)),exp(-j*2*pi*sin(theta_x)),exp(-j*3*pi*sin(theta_x))];
%start vector may be any type:
%w = [0.1 0.1 0.1 0.1]
wstart = w;
%mu = 0.007;
mu = 0.001;
y = zeros(1,length(t));
e = zeros(1,length(t));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Constant Modulus Algorithm
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for repeat=1:1,
for i=0:length(signal_ref)-1,
y(i+1) = w * signal_ns(:,i+1);
%e(i+1) = 1 - y(i+1)
e(i+1) = y(i+1)/((abs(y(i+1))))^2-y(i+1); %SATO's principle for updating the error
w = w + mu*e(i+1)*(signal_ns(:,i+1))';
end;
end;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Plot all figures

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
close all;
subplot 311, (plot(phase(signal_x),':'))
ylabel('phase(rad)');
title('desired signal 10 degrees interferers -10 and -40 degrees')
grid on;
axis([0 210 -2 2])
hold on;
plot(phase(y),'r');
legend('phase(d)','phase(y)')
hold off;
subplot 312, plot(abs(signal_x),':');
hold on;
grid on;
axis([0 210 0 2])
plot(abs(y),'r');
legend('ldl', 'lyl')
ylabel('amplitude');
%xlabel('sample(index)');
hold off;
subplot 313, plot(abs(e));
grid on;
legend('lerrorl');
ylabel('amplitude');
xlabel('sample(index)');
axis([0 210 0 1]);
for k = 0:arraylength-1
arrayvec(k+1,:) = exp(j*k*Kd*sin(theta));
end;
%calculate response of array
F = w*arrayvec;
% F2 = wstart*arrayvec;
figure
plot(((theta/(2*pi))*360), 20*log10(abs(F)));
%hold on;
%plot(((theta/(2*pi))*360), 20*log10(abs(F2)), 'r');
title('amplitude response antenne, desired signal: 10 degrees, interferers: -10 and -40
degrees');
ylabel('(dB)');
xlabel('angle(degrees)');
axis([-90,+90,-30,10]);
grid on;
hold on;
dbgrens = 40

```

```

%FILE CMAarray.m
%adaptive array demonstrator CM algorithm for Digital Beamforming 5
typeofinterferer1='noise'; %set to 'noise' for gaussian type of interferer
%set to 'signl' for MSK type of interferer
theta_x = 10 %degrees
theta_x = (2*pi/360)*theta_x
theta_n1 = -10; %degrees
theta_n1 = (2*pi/360)*theta_n1
theta_n2 = -40 %degrees
theta_n2 = (2*pi/360)*theta_n2
theta = pi*[-1:0.005:1];
arraylength = 4; %nr of antennas
bitrate = 200;
fsim = 4*bitrate; %simulation frequency
Ts = 1/fsim;
message = [1 -1 1 1 1 -1 1 -1 -1 -1 1 1 -1 1 1 1 1 -1 -1 -1 1 1 1 1 1 -1 1 -1 1 1 1 1 -1 -1 -1 -1
1 -1 1 -1 -1 -1 -1 -1 1 1 1 1 -1 1 -1 1 1];
message = upsample(message, fsim/bitrate); %upsample message
t = Ts:Ts:(length(message)/fsim); %timeline
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%generate a complex MSK signal
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
mesint=(cumsum(message))/8;
Q = cos(pi*mesint);
I = sin(pi*mesint);
hold on;
signal_x = I+j*Q; %the signal to be received
signal_ref = signal_x(1:end);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%generate 2 interferers n1 and n2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% if the interfere n1 needs to be gaussian noise -> uniform phase, normal amplitude
distribution
if (typeofinterferer1 == 'noise'),
signal_n1 = normrnd(0,1,1,length(t)).*exp (j*(unifrnd(-pi,pi,1,length(t))));
end;
% if testing with an MSK signal as interferer is required: noise source n1 will be MSK
signal
if (typeofinterferer1 == 'signl'),
message_n1 = [-1 -1 1 -1 1 -1 -1 1 -1 -1 -1 1 1 1 -1 -1 1 1 1 1 -1 1 -1 1 -1 1 -1 -1 -1 1 1 -1 1 -
1 -1 -1 1 -1 -1 -1 -1 -1 1 1 1 -1 1 -1 1 -1];
message_n1 = upsample(message_n1, fsim/bitrate); %upsample message
t = Ts:Ts:(length(message)/fsim); %timeline

```

```

mesint_n1=(cumsum(message_n1))/8;
Qn1 = cos(pi*mesint_n1);
In1 = sin(pi*mesint_n1);
signal_n1 = In1+j*Qn1; %the signal to be received
end;
%noise source 2 is gaussian
signal_n2 = normrnd(0,1,1,length(t)).*exp (j*(unifrnd(-pi,pi,1,length(t))));
noise = zeros(arraylength, length(t));
% system noise for every antenna
for i = 0:arraylength-1,
noise(i+1,:) = normrnd(0,0.01,1,length(t)).*exp (j*(unifrnd(-pi,pi,1,length(t))));
end;
%lambda = sym('lambda');
%K = 2*pi/ lambda
%d = 0.5*lambda
Kd = 2*pi;
%alpha = -Kd*sin(theta_nul);
% array responses for the desired signal x and noise n1 and n2
arrayvec_x = zeros(1,arraylength);
arrayvec_n1 = zeros(1,arraylength);
arrayvec_n2 = zeros(1,arraylength);
for k = 0:arraylength-1,
arrayvec_x(k+1) = exp(j*k*Kd*sin(theta_x));
end;
for k = 0:arraylength-1,
arrayvec_n1(k+1) = exp(j*k*Kd*sin(theta_n1));
end;
for k = 0:arraylength-1,
arrayvec_n2(k+1) = exp(j*k*Kd*sin(theta_n2));
end;
x = zeros(arraylength, length(t));
n1 = zeros(arraylength, length(t));
n2 = zeros(arraylength, length(t));
% received signal from signal source x
for i = 0:arraylength-1,
x(i+1,:) = signal_x .* arrayvec_x(i+1);
end;
% received signal from noise source n1
for i = 0:arraylength-1,
n1(i+1,:) = signal_n1 .* arrayvec_n1(i+1);
end;
% received signal from noise source n2
for i = 0:arraylength-1,
n2(i+1,:) = signal_n2 .* arrayvec_n2(i+1);
end;
%total received signal is the
%+- -+

```

```

%| signal received at antenna 1 |
%| signal received at antenna 2 |
%| | |
%| signal received at antenna n |
%+- -+
w = zeros(1,arraylength);
signal_ns = (noise + n1+n2+x);
%start weight vector, may be in the direction of the desired signal:
w = (1/4)*[1, exp(-j*pi*sin(theta_x)),exp(-j*2*pi*sin(theta_x)),exp(-j*3*pi*sin(theta_x))];
%start vector may be any type:
%w = [0.1 0.1 0.1 0.1]
wstart = w;
%mu = 0.007;
mu = 0.001;
y = zeros(1,length(t));
e = zeros(1,length(t));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Constant Modulus Algorithm
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for repeat=1:1,
for i=0:length(signal_ref)-1,
y(i+1) = w * signal_ns(:,i+1);
%e(i+1) = 1 - y(i+1)
e(i+1) = y(i+1)/((abs(y(i+1))))^2-y(i+1); %SATO's principle for updating the error
w = w + mu*e(i+1)*(signal_ns(:,i+1))';
end;
end;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Plot all figures
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
close all;
subplot 311, (plot(phase(signal_x),'.'))
ylabel('phase(rad)');
title('desired signal 10 degrees interferers -10 and -40 degrees')
grid on;
axis([0 210 -2 2])
hold on;
plot(phase(y),'r');
legend('phase(d)','phase(y)')
hold off;
subplot 312, plot(abs(signal_x),'.');
hold on;
grid on;

```

```

axis([0 210 0 2])
plot(abs(y),'r');
legend('ldl', 'lyl')
ylabel('amplitude');
%xlabel('sample(index)');
hold off;
subplot 313, plot(abs(e));
grid on;
legend('lerrorl');
ylabel('amplitude');
xlabel('sample(index)');
axis([0 210 0 1]);
for k = 0:arraylength-1
arrayvec(k+1,:) = exp(j*k*Kd*sin(theta));
end;
%calculate response of array
F = w*arrayvec;
% F2 = wstart*arrayvec;
figure
plot(((theta/(2*pi))*360), 20*log10(abs(F)));
%hold on;
%plot(((theta/(2*pi))*360), 20*log10(abs(F2)), 'r');
title('amplitude response antenne, desired signal: 10 degrees, interferers: -10 and -40
degrees');
ylabel('(dB)');
xlabel('angle(degrees)');
axis([-90,+90,-30,10]);
grid on;
hold on;
dbgrens = 40

```