Simulation of Adaptive Circular Array for CDMA Systems

Qun Zheng

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Applied Science at
Concordia University
Montréal, Québec, Canada

July 2002

ABSTRACT

Simulation of Adaptive Circular Array for CDMA Systems

Qun Zheng

In this thesis, we study the performance of adaptive Uniformly Spaced Circular Array (UCA) and Uniformly Spaced Linear Array (ULA) utilized in the base station of CDMA systems. In these systems, the desired signal becomes much stronger than the interference signals after chip correlation by a factor of the processing gain. Further, simple blind algorithms and the real-time implementation can be applied to improve bit error rate in various interference scenario. The performance of several algorithms is compared in AWGN channel, Rayleigh Flat Fading channel, and Frequency Selective Fading channel considering angle spread, phase errors and mobile tracking. In wireless communications, propagation effects of the transmitted signals result in delay spread, angle spread, Doppler spread and multipath in mobile radio channel. Different antenna array geometries may cause different responses to the received signals. Our study shows that the UCA is less sensitive to the angle spread than the ULA does, consequently the UCA exhibits better performance than the ULA in the multipath environment with angle spread, which is almost certainly the case in the uplink of CDMA communication systems.

ACKNOWLEDGEMENTS

I would like to sincerely thank my advisor, Dr. A.K. Elhakeem, for his support and patient guidance throughout this research, and for providing me with continuous feedback and encouragement during the period I worked in this thesis.

I would like to thank all my friends who constantly shared their wisdom and joyful moments with me during my studies at Concordia University. Special thank Mr. Li zhiyun for some helpful discussion and suggestion to this thesis.

I would also like to thank my wife, parents and sister for their support and encouragement through every step of my life.

TABLE OF CONTENTS

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Smart antenna technology applied at the base station represent an important technological innovation capable of increasing capacity and range of a wireless system. Along with Code Division Multiple Access (CDMA) technique, smart antenna will play an important role in the future of wireless telecommunications.

Smart antennas are a new technology for wireless systems that use a fixed set of antenna elements in an array. The signals from these antenna elements are combined to form a movable beam pattern that can be steered, using either digital signal processing, or RF hardware, to a desired direction that tracks mobile units as they move. This allows the smart antenna system to focus Radio Frequency resources on a particular subscriber, while minimizing the impact of noise, interference, and other effects that can degrade signal quality.

Code Division Multiple Access (CDMA) is a new wireless technology that allows multiple radio subscribers to share the same frequency band at the same time by assigning each user a unique code. The technology makes very efficient use of limited spectral resources and allows robust communication over time-varying radio channels. Both smart antenna technology and CDMA promise to revolutionize the field of wireless communications [1].

Smart antenna systems can include both adaptive antenna and switched beam technology [1]. In this thesis, we study the performance of adaptive antenna with Uniformly Spaced Circular Array (UCA) and Uniformly Spaced Linear Array (ULA)

applied in the base station of the CDMA systems. The goal of this research is to compare the performance of several adaptive algorithms and array geometries and finally find the way that makes the real-time implementation achievable in the base station of the CDMA systems.

1.2 Thesis Organization

This thesis is organized as follows. Chapter 2 reviews the fundamentals of adaptive antenna arrays, the antenna array beamforming technique and the spatial filtering with the effect of the number of antenna elements and the effect of antenna element spacing. Chapter 3 introduces the adaptive beamforming algorithms, especially of those that will be used in our simulations. Chapter 4 introduces the adaptive antenna techniques for CDMA systems; we also review the multipath fading channel and the CDMA spatial processors which are crucial to our simulations. Chapter 5 presents the simulation results under various communication channels and some analysis of these results. Chapter 6 concludes this thesis and gives suggestions for future work.

Chapter 2
Fundamentals of Antenna Arrays

2.1 Introduction

Smart Antennas offer a broad range of ways to improve wireless system performance. In general, smart antennas have the potential to provide enhanced range and reduced infrastructure costs in early deployments, enhanced link performance as the system is built-out, and increased long-term system capacity [1].

Smart antenna systems can include both adaptive antenna and switched beam technologies. Adaptive antenna is an array of antennas, which is able to change its antenna pattern dynamically to adjust to noise, interference, and multipath, and it can adjust their pattern to track portable users. Adaptive antennas are used to enhance received signals and may also be used to form beams for transmission. The structure of an adaptive antenna processor is shown in Figure 2-1.

Switched Beam Systems use a number of fixed beams at an antenna site. The receiver selects the beam that provides the greatest signal enhancement and interference reduction. Switched beam systems may not offer the degree of performance improvement offered by adaptive systems, but they are often much less complex and are easier to retrofit to existing wireless technologies. This thesis will focus on Adaptive Antenna Array.

This chapter reviews and explores conventional antenna array processing concepts and adaptive beamforming techniques. Section 2.2 introduces antenna array concept and the array response model. The Antenna Array Beamforming and Spatial Filtering

capability are presented in Section 2.3. Then the Adaptive Antenna Array systems and the criteria for adaptive algorithms are reviewed in Section 2.4.

2.2 Antenna Array and Response Model

An antenna array consists of a set of antenna elements that are spatially distributed at known locations with reference to a common fixed point [6]. The antenna elements can be arranged in various geometries, with linear, circular and planar arrays being very common. In the case of a linear array, the centers of the elements of the array are aligned along a straight line. If the spacing between the array elements is equal, it is called a *Uniformly Spaced Linear Array* (ULA). A circular array is one in which the centers of the array elements lie on a circle. If the spacing between the array elements is equal, it is called a *Uniformly Spaced Circular Array* (UCA). In the case of a planar array, the centers of the array elements lie on a single plane. Both the linear array and circular array are special cases of the planar array. In this thesis, the cases of the most commonly used array patterns – Uniformly Spaced Linear Array (ULA) and Uniformly Spaced Circular Array (UCA) are studied.

The radiation pattern of an array is determined by the radiation pattern of the individual elements, their orientation and relative positions in space, and the amplitude and phase of the feeding currents. If each element of the array is an isotropic point source, then the radiation pattern of the array

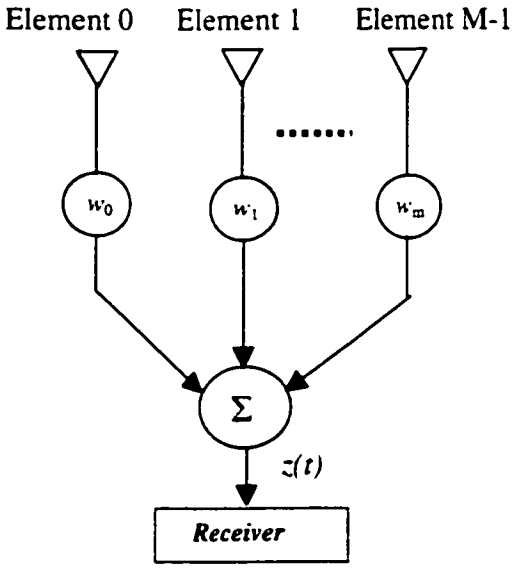


Figure 2-1 Antenna Array with Beamforming capability

will depend solely on the geometry and feeding current of the array.

2.2.1 Array Response Model

As shown on Figure 2-1, we assume that the antenna array include M elements. Each branch of the array has a weighting element, w_m, which has both a magnitude and a phase associated with it.

We define the *weight vector* as $\mathbf{w} = [w_0 \ldots w_{M-1}]^H$, where the superscript H represents the *Hermitian transpose.*

Consider a plane wave incident on the array from an angle ϕ (for simplicity we assume that the source of signal is in the same plane with the antenna array). We represent the modulation of the plane wave using the baseband complex envelope, *s(t).* Assume that all of the array elements are noiseless isotropic antennas which have uniform gain in all directions. The signal received at antenna element *m* is

$$u_m(t) = As(t)e^{-j\omega_c\tau_m} \tag{2.1}$$

where A is the arbitrary gain constant, ω_c is the carrier frequency and τ_m represent the time delays of the wave on m^{th} element relative to a reference point.

The signals from each antenna element can be grouped in a *data vector*

$$\boldsymbol{u}(t) = [u_0(t) \ldots u_{M-1}(t)]^T \tag{2.2}$$

Then the array output, *z(t),* can be expressed as

$$z(t) = \sum_{m=0}^{M-1} w_m u_m(t) = As(t)\sum_{m=0}^{M-1} w_m e^{-j\omega_c\tau_m} = As(t)\mathbf{w}^H\mathbf{a}(\phi) = \mathbf{w}^H\boldsymbol{u}(t) \tag{2.3}$$

where $\mathbf{u}(t) = As(t)\mathbf{a}(\phi)$.

The vector $\mathbf{a}(\phi)$ is called the *steering vector* in direction ϕ, and it describes the phase of the signal available at each antenna element relative to the phase of the signal at the reference element. The *steering vector* is

$$\mathbf{a}(\phi) = [a_0(\phi) \quad a_1(\phi) \quad \ldots \quad a_{M-1}(\phi)]^T \tag{2.4}$$

At certain carrier frequency ω_c, the *steering vector* can be expressed in terms of time delay as

$$\mathbf{a}(\phi) = [\, e^{-j\omega_c \tau_0} \quad e^{-j\omega_c \tau_1} \quad \ldots \, e^{-j\omega_c \tau_{M-1}} \,]^T \tag{2.5}$$

The value of each element in the vector will depend on the position of the antenna element relative to the reference element, i.e., it depends on the array pattern.

To simplify the analysis of antenna arrays, we make the follow assumptions [1]:

- The spacing between array elements is small enough that there is no amplitude variation between the signals received at different elements.
- There is no mutual coupling between elements.
- There are a finite number of signals.
- The bandwidth of the signal incident on the array is small compared with the carrier frequency.

2.2.2 Uniformly Spaced Linear Array

The Uniformly Spaced Linear Array (ULA) is illustrated in Figure 2-2.

As shown in sector 2.2.1, at certain carrier frequency ω_c, the *steering vector* is only decided by the time delay on each element.

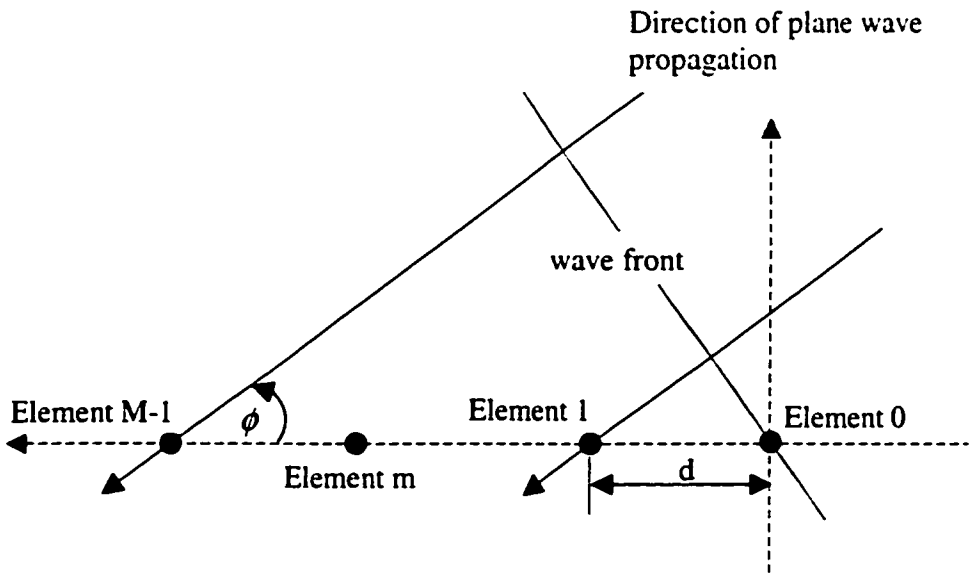


Figure 2-2 Uniformly Spaced Linear Array

In the case of ULA, we take element 0 as the reference element, i.e. $\tau_0 = 0$. The time delays on m^{th} elements is calculated by Equation (2.6),

$$\tau_m = m \cdot \frac{d}{c} \cdot \cos\phi \tag{2.6}$$

where d is the equal antenna spacing between adjacent antenna elements, c is the speed of light, 3×10^8 m/s, ϕ is DOA (Direction of Arrival).

2.2.3 Uniformly Spaced Circular Array

The Uniformly Spaced Circular Array (UCA) is illustrated in Figure 2-3.

We take element 0 as the reference element.

The distance from m^{th} antenna to the reference element is

$$a_m = R(\cos(\phi) - \cos(\vartheta_m - \phi)) \tag{2.7}$$

where R is the radius of the circle, ϑ_m is the angle of m^{th} element relative to x-axis, ϕ is DOA.

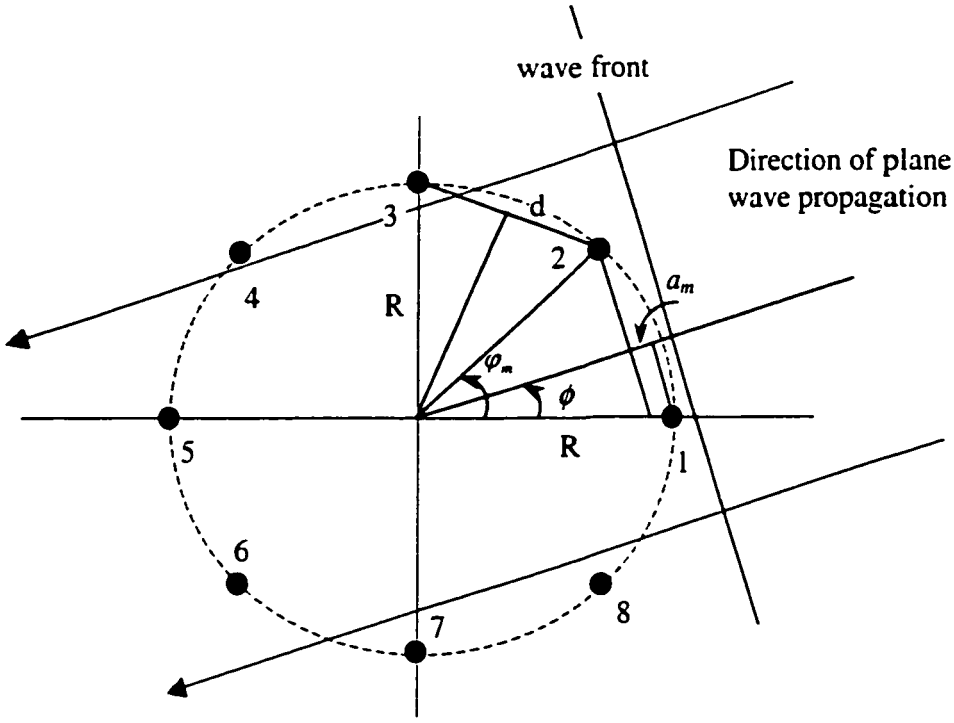


Figure 2-3 8-element Uniformly Spaced Circular Array

If we consider M elements uniformly distribute on the circle, then

$$d = 2 \cdot R \cdot \sin\frac{\pi}{M} \tag{2.8}$$

$$\vartheta_m = \frac{2\pi(m-1)}{M} \tag{2.9}$$

hence

$$a_m = \frac{d}{2 \sin \pi / M}\left[\cos(\phi) - \cos(\frac{2\pi(m-1)}{M} - \phi)\right] \tag{2.10}$$

The time delays on m^{th} elements is

$$\tau_m = a_m/c \tag{2.11}$$

where c is the speed of light.

2.3 Antenna Array Beamforming and Spatial Filtering

Systems designed to receive spatially propagating signals often encounter the presence of interference signals. The desired and interfering signals usually originate from different spatial locations. This spatial separation can be exploited to separate signal from interference using a spatial filter at the receiver. A beamformer is a processor used in conjunction with an array of sensors (i. e., antenna elements in an adaptive array) to provide a versatile form of spatial filtering. The sensor array collects spatial samples of propagating wave fields, which are processed by the beamformer. Typically a beamformer linearly combines the spatially sampled time series from each sensor to obtain a scalar output time series in the same manner that an FIR filter linearly combines temporally sampled data.

The spatial passband and stopband of an antenna array can be affected by the choice of array geometry, spatial aperture, and antenna weights. The *angular resolution* of an antenna array can be used to evaluate its ability to accurately attenuate powers from directions other than the desired direction. High *angular resolution* improves the maximum output signal-to-interference-plus-noise ratio (SINR) when the angular separation between the desired and undesired signals is small. The reciprocal of the 3 dB beamwidth/spatial passband of the main lobe (the principal lobe) of the antenna array's beam pattern is the array's *angular resolution*. In other words, the narrower the main lobe, the better the spatial filtering capability of the antenna array. Note that the main lobe of an array's directivity pattern is called a beam. A directivity pattern is also called the beam pattern [23].

2.3.1 Spatial Filtering

The frequency response of an FIR filter with tap weights w^*_i, i = 1,..., M and a sampling period T_s is given by

$$H(e^{j2\pi f}) = \sum_{i=1}^{M} \mathbf{w}_i^* e^{-j2\pi f T_s(i-1)} \tag{2.12}$$

where $H(e^{j2\pi f})$ represents the response of the filter to a complex sinusoid of frequency f. For the harmonic retrieval problem, if we want to extract the signal with frequency f_i, we need to find a set of complex weights such that the frequency response of the filter has a higher gain at f_i and lower gains at other frequencies. For the beamforming problem, what we want to extract is the signal from certain direction ϕ and refuse the signal other than that.

The *array response* of a weighted antenna array, which represents the relative sensitivity of a response to signals coming from DOA ϕ and having an array response vector of $\mathbf{a}(\phi)$, can be defined as

$$A(\Phi) = \mathbf{w}^*\mathbf{a}(\phi) = \sum_{m=1}^{M} w_m^* e^{-j\omega_c \tau_m} = \sum_{m=1}^{M} r_m e^{-j\omega_c(\tau_m - \hat{\tau}_m)} \tag{2.13}$$

here we set $w_m = r_m e^{j\omega_c \hat{\tau}_m}$, the complex weight with *shading* r_m and a time delay of $\hat{\tau}_m$, which are applied to the m^{th} antenna element's output. Note that $\hat{\tau}_m$ is a value corresponding to certain DOA ϕ_0. τ_m is the time delay of a plane-wave impinging on the array measure from the array reference element, and M is the number of antenna array elements.

The *beam pattern* is defined as the magnitude of the array response $A(\Phi)$ and is given by

$$G(\Phi) = |A(\Phi)| \tag{2.14}$$

The *normalized beam pattern* can be expressed as

$$G(\Phi) = 10 \log_{10}\left\{\frac{|A(\Phi)|^2}{(\sum_{m=1}^{M} r_m)^2}\right\} \tag{2.15}$$

since A(Φ) maximized when $\tau_m = \hat{\tau}_m$.

Using the *beam pattern*, we can now explore the effects on the beam pattern (spatial filtering capability) caused by different choices for the antenna array spacing d, array geometry (UCA or ULA) and the number of antenna elements used. The antennas considered here are omnidirectional and the array response time delays τ_m of an UCA and ULA follows Equations (2.6) and (2.11), respectively. The *beam pattern*, showing the relative sensitivity of the spatial array response to incoming DOAs ϕ, are evaluated over a range of $0^\circ \leq \phi \leq 360^\circ$.

Now we consider the beam pattern with $r_m = 1$ for m = 1,2,..., M.

For the case of ULA, recall equation (2.6) and we adjust w_m with

$$\hat{\tau}_m = m \cdot \frac{d}{c} \cdot \cos \phi_0 \tag{2.16}$$

the phase of beam pattern (2.13) can be written as

$$\omega_c(\tau_m - \hat{\tau}_m) = 2 \cdot \pi \cdot m \cdot \frac{d}{c} \cdot (\cos \phi - \cos \phi_0) \tag{2.17}$$

where ϕ_0 is the DOA of the desired signal, i.e. the direction of which we want to have the maximum antenna gain.

Similarly, for the case of UCA, recall equation (2.11) and we can adjust w_m with

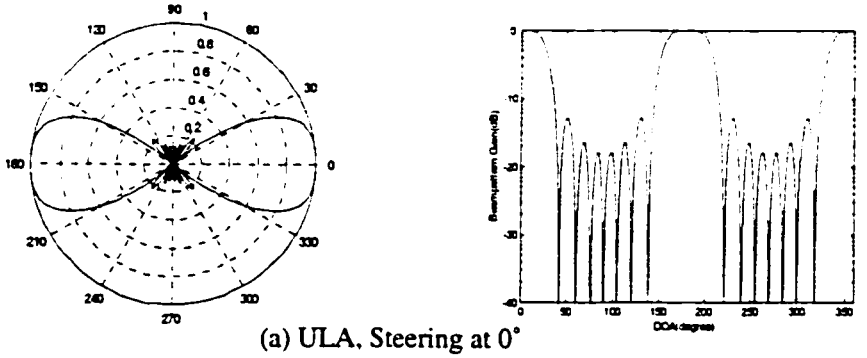
$$\hat{\tau}_m = \frac{d}{2 \cdot c \cdot \sin \pi / M}\left[\cos(\phi_0) - \cos(\frac{2\pi(m-1)}{M} - \phi_0)\right] \tag{2.18}$$

Hence from equation (2.13) to (2.18), we can investigate the beam pattern of ULA and UCA under different conditions.

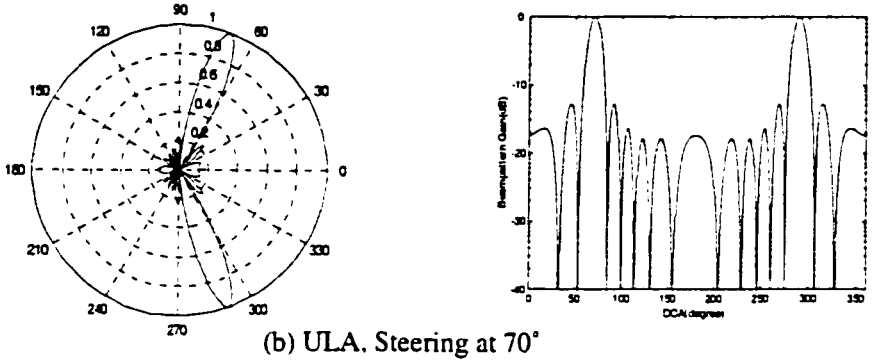
Figure 2-4 and 2-5 show the beam steering ability of a UCA and a ULA in polar coordinate and Semi-logarithmic coordinate, respectively, when the steering angle ϕ_0 equals 0° and 70°. Here the spacing between the antenna elements d is kept constant at one-half of the carrier wave length λ and the number of antenna elements is 8. In the figures, lobes which have a gain level much below that of the *mainlobe* are called *sidelobes*, and lobes which have the same gain level as the mainlobe are called *grating lobes*, which are usually undesirable.

Note that the beam of a UCA retains the same 3 dB beamwidth while it is steering; however, the beamwidth of a ULA gradually increases as the beam steers away from broadside. It can be reasoned that the directivity gain pattern of a ULA, in response to a DOA, ϕ, with the time delays $\hat{\tau}_m$ being designed to steer at ϕ_0, can be written as (by using the equation $\sum_{m=0}^{M-1} a^m = (1 - a^M)/(1 - a)$)

$$A(\Phi) = \sum_{m=1}^{M} e^{-j\omega_c(\tau_m - \hat{\tau}_m)} = \frac{1 - e^{-j\frac{2\pi}{\lambda}Md(\cos\phi - \cos\phi_0)}}{1 - e^{-j\frac{2\pi}{\lambda}d(\cos\phi - \cos\phi_0)}}$$

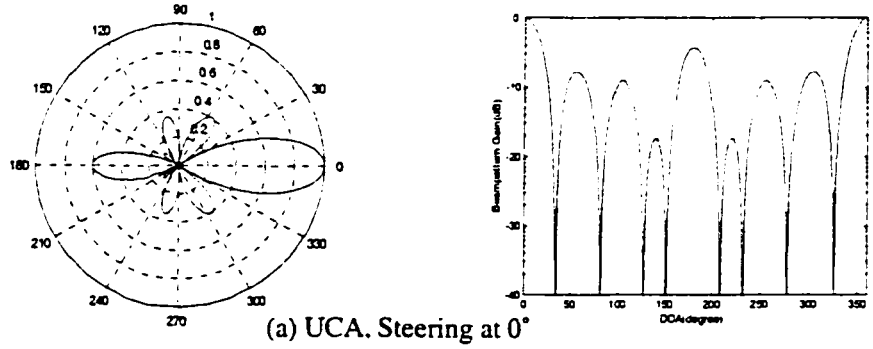


(a) ULA, Steering at 0°

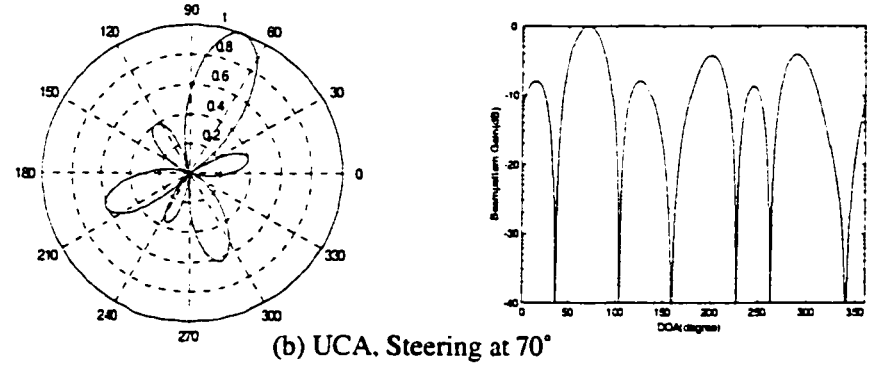


(b) ULA, Steering at 70°

Figure 2-4: Beam pattern of 8-element ULA, with d = 0.5λ



(a) UCA, Steering at 0°



(b) UCA, Steering at 70°

Figure 2-5: Beam pattern of 8-element UCA, with d = 0.5λ

$$= \frac{\sin\frac{M\pi d}{\lambda}(\cos\phi - \cos\phi_0)}{\sin\frac{\pi d}{\lambda}(\cos\phi - \cos\phi_0)} \cdot e^{-j\frac{\pi}{\lambda}Md(\cos\phi-\cos\phi_0)} \tag{2.22}$$

From equation (2.22), we see that the normalized beam pattern value depends on the difference between the cosine of the angles (a change of sign will occur when the DOAs are off-broadside) rather than on the difference between the angles themselves (when ϕ_0 is close to broadside, $\cos\phi_0 \approx 0$). Therefore, the array pattern plots have different shapes depending on the value of the steered angle ϕ_0. Thus, a ULA beam pattern has a larger main beamlobe asymmetry for a larger off-broadside steering angle ϕ_0.

2.3.2 Effect of Number of Antenna Elements

Now let's observe the beam pattern of ULA and UCA in different number of antenna elements. The antenna spacing is still kept constant at $\lambda/2$ and the azimuth of the signal (steering angle ϕ_0) is fixed at 70° for the convenience of observation.

Figure 2.6 and 2.7 show the beam patterns of 3, 6, 12, and 16-element ULA and UCA respectively. It is clear that increasing the number of antenna elements decreases the beam width of the main lobe, which improves the resolution. High *array resolution* improves the maximum output signal-to-noise ratio (SNR) when the angular separation between the desired and the undesired signals is small. High-resolution capability also implies sharp array pattern nulls, thereby reducing the array ability to place a broad null on clustered interference sources [11]. By observing the figures, we can find that the number of nulls (slots between two lobes) basically less than the number of antenna elements in the scope of 180° for ULA and 360° for UCA. In general, an M-element

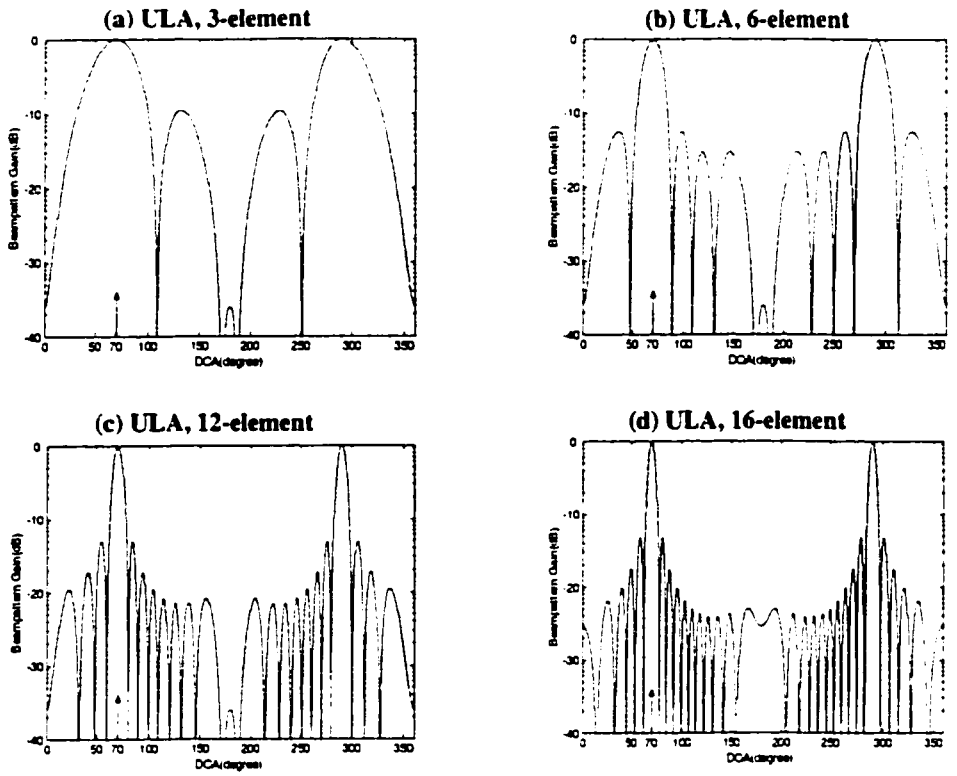


Figure 2-6: Effect of Number of Elements, ULA, with d = 0.5λ, ϕ_0=70°

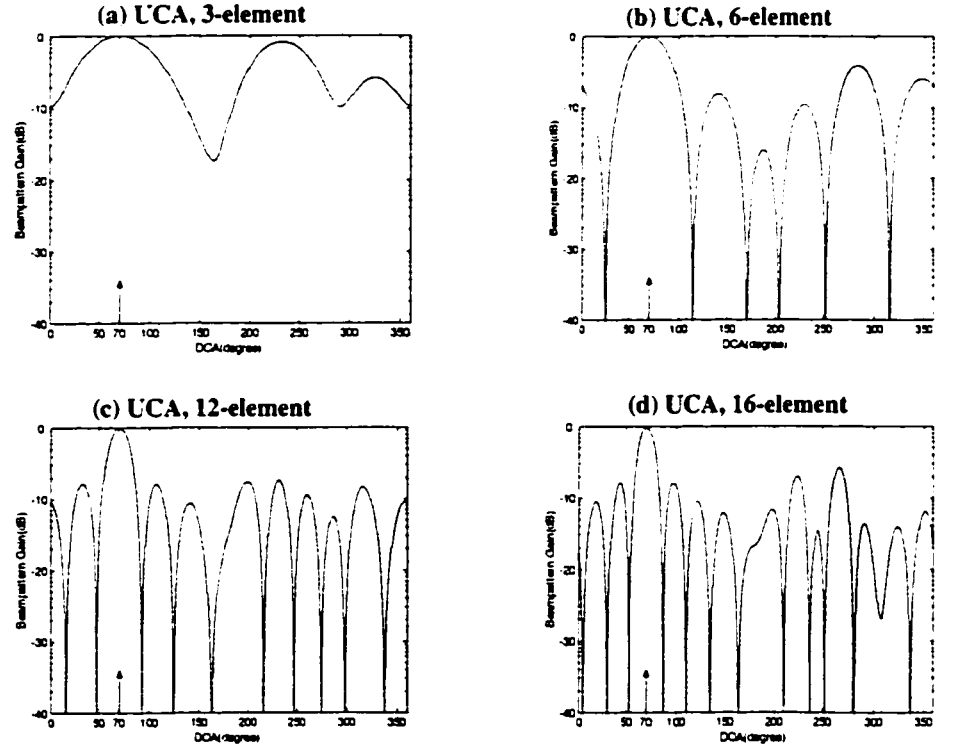


Figure 2-7: Effect of Number of Elements, UCA, with d = 0.5λ, ϕ_0=70°

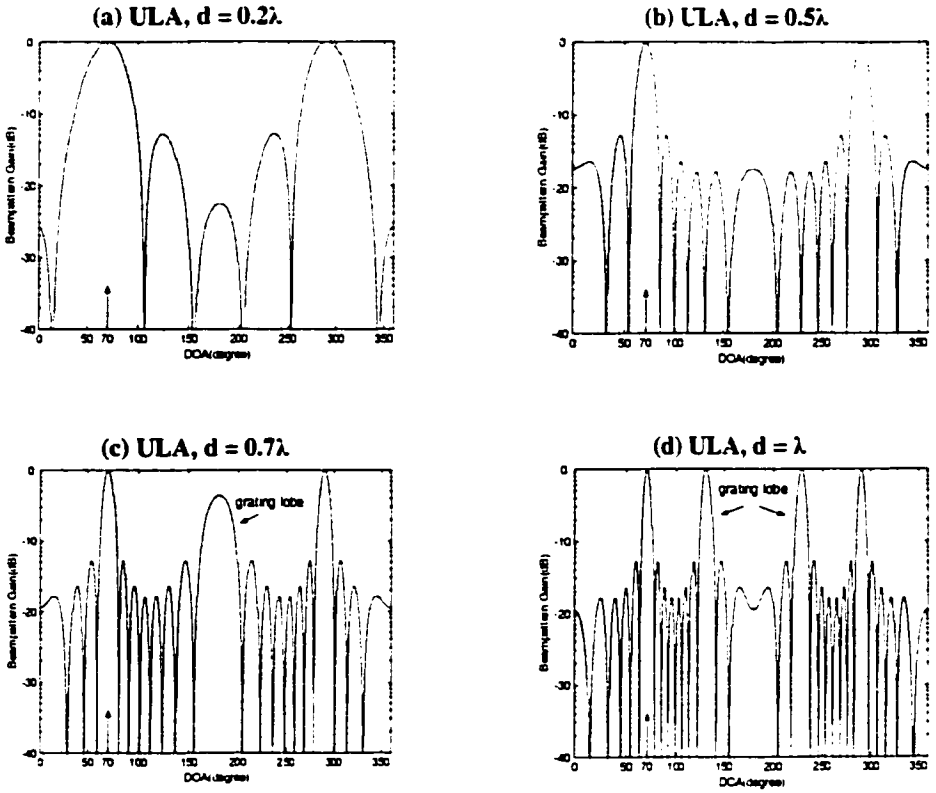


Figure 2-8: Effect of Elements Spacing, ULA, 8-element, ϕ_0=70°

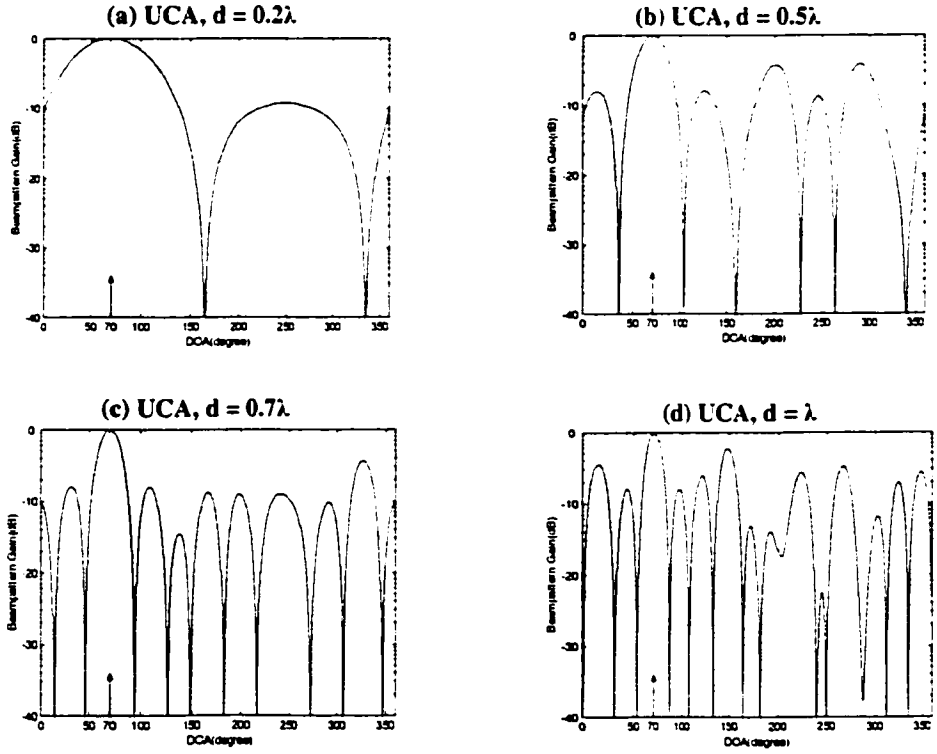


Figure 2-9: Effect of Elements Spacing, UCA, 8-element, ϕ_0=70°

linear array has M-1 degrees of freedom so that up to M-1 array beam pattern nulls can be independently adjusted for array operation [5][11]. This feature means that the capability for the antenna array to null out the interfering signals is limited by the number of antenna elements.

From above we see that different array geometries, such ULA and UCA, produces different beam patterns, i.e. mainlobe beamwidth and sidelobe levels. The mainlobe beamwidth and sidelobe levels of ULA are obviously less than that of UCA. But note that there is a counterpart that is symmetric to the mainlobe in linear array (ULA).

2.3.3 Effect of Antenna Element Spacing

Now the effect of the *spacing* between the antenna elements on beam pattern is considered.

Figure 2.8 and 2.9 show the beam patterns of different antenna element spacing for 8-element ULA and UCA, in which the azimuth of the signal is still fixed at 70°. Note that the absolute separation between antenna elements is not important, while its relative value to the carrier wavelengths is crucial [12]. These plots show that resolution increases as the spacing increases. However, the trades-off is that the grating lobes are becoming more significant. These effects will lead to a degradation of the weighted output SINR since interfering signals will be passed through the grating lobes with a high spatial gain. To avoid grating lobe for omnidirectional antennas, a maximum element spacing of $\lambda/2$ is usually used and such an array is called a "filled array". Filled arrays, however, need a larger number of antenna elements to obtain resolution than is available with a wider spatial aperture. Also, the antenna spacing should not be less than $\lambda/4$, which is

considered to be inefficient since there is very little difference in the directional pattern as compared to that of a single element antenna [11].

2.4 Adaptive Antenna Array Systems

In a mobile communication system, the mobile is generally moving; therefore the DOAs of the received signals in the base station are time varying. Also, due to the time-varying wireless channel between the mobile and the base station, and the existence of the cochannel interference, multipath, and noise, the parameters of each impinging signal are varied with time. For a beamformer with constant weights, the resulting beampattern cannot track these time-varying factors. However, an adaptive array may change its patterns automatically in response to the signal environment. An adaptive array is an antenna system that can modify its beampattern or other parameters, by means of internal feedback control while the antenna system is operating. Adaptive arrays are also known as adaptive beamformers, or adaptive antennas. A simple adaptive array is shown in Figure 2.10 [5], in which d_i represents an estimate or replica of the desired signal for the i^{th} user at the array output.

In Figure 2.10, the complex weights $w_1,..., w_m$ are adjusted by the adaptive control processor. The method used by the adaptive control processor to change the weights is called the adaptive algorithm. Most adaptive algorithms are derived by first creating a performance criterion, and then generating a set of iterative equations to adjust the weights such that the performance criterion is met. Some of the most frequently used performance criteria include minimum mean squared error (MSE), maximum signal-to-interference-and-noise ratio (SINR), maximum likelihood (ML), minimum noise variance, minimum output power, maximum gain, etc. [12]. These criteria are often

expressed as cost functions, which are typically inversely associated with the quality of the signal at the array output. As the weights are iteratively adjusted, the cost function becomes smaller and smaller. When the cost function is minimized, the performance criterion is met and the algorithm is said to have converged [4].

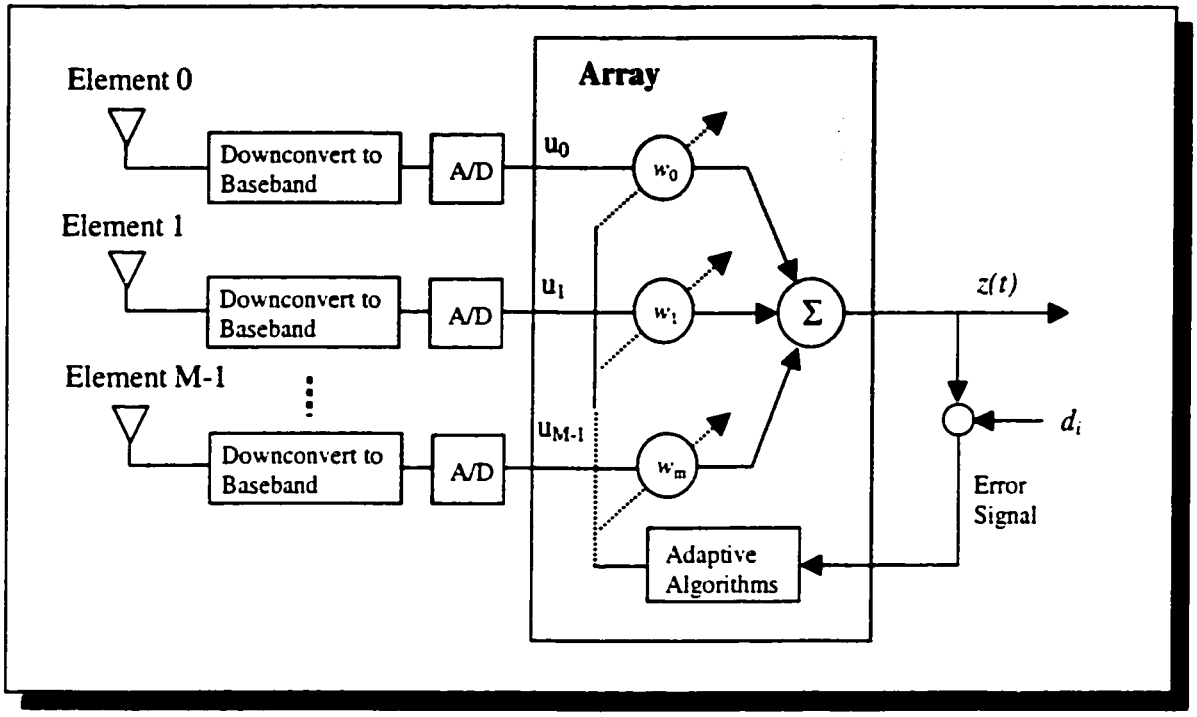


Figure 2-10 An Adaptive Antenna Structure

For one adaptive array, there may exist several adaptive algorithms that could be used to adjust the weight vector. The choice of one algorithm over another is determined by various factors [15]:

- Rate of convergence. This is defined as the number of iterations required for the algorithm, in response to stationary input, to converge to the optimum solution. A fast rate of convergence allows the algorithm to adapt rapidly to a stationary environment of unknown statistics.

- Tracking. When an adaptive algorithm operates in a nonstationary environment, the algorithm is required to track statistical variations in the environment.
- Robustness. In one context, robustness refers to the ability of the algorithm to operate satisfactorily with ill-conditioned input data. The term robustness is also used in the context of numerical behavior.
- Computational requirements. Here the issues of concern include (a) the number of operations (i. e., multiplications, divisions, and additions/subtractions) required to make one complete iteration of the algorithm, (b) the size of memory locations required to store the data and the program, and (c) the investment required to program the algorithm on a computer or a DSP processor.
- Structure. This refers to the structure of information flow in the algorithm, determining the manner in which it is implemented in hardware form. For example, and algorithm whose structure exhibits high modularity, parallelism, or concurrency is well suited for implementation using very large-scale integration (VLSI).

Since there exists a mapping between the narrowband beamformer and the FIR filter, most of the adaptive algorithms used by the adaptive filter [15] may be applied to the adaptive beamformer. However, some of the adaptive beamforming algorithms also have their unique aspects that an adaptive filter algorithm does not possess. A thorough survey of adaptive beamforming algorithms is given in Chapter 3.

2.5 Summary

In this chapter we introduced some basic concepts related to antenna array and adaptive beamforming. Signal model of antenna array system are established. The effect

of number of antenna elements and element spacing on antenna array spatial filtering capability are investigated. In summary, the antenna array spatial filtering resolution increases as the number of antenna elements increases. As antenna element spacing increases, resolution will also increase, but this trade-offs with the presence of grating lobes. The resolution of a ULA decreases as it steers off-broadside, while a UCA appears to have the same resolution as it steers around its full angular range. Also, different types of array geometries provide different resolutions and beam patterns. Finally the choice of adaptive algorithms for the adaptive array systems should follow some principles.

Chapter 3

Adaptive Beamforming Algorithms

3.1 Introduction

This chapter provides a survey of adaptive algorithms. Most of the algorithms described in this chapter will be simulated.

As we described in Chapter 2, since the mobile environment is time-variable, the solution for the weight vector must be updated or adapted periodically. Typically, the weight vector computed at one cycle differs from the weight vector computed at the previous update by only a small amount. Also, since the data required to estimate the optimal solution is noisy, it is desirable to use an update technique which uses previous solutions for the weight vector to smooth the estimate of the optimal response, thus reducing the effects of noise. For these reasons, an *adaptive algorithm* is used to update the weight vector periodically. The adaptive algorithm operates either in a block mode or iterative mode. In block processing techniques, a new solution is calculated periodically using estimates of statistics obtained from the most recently available block of data. In iterative algorithms, at each iteration n, the current weight vector, $\bar{w}(n)$, is adjusted by an incremental value to form a new weight vector, $\bar{w}(n+1)$, which approximates the optimum solution [5].

Most of the adaptive algorithms may be categorized into two classes according to whether a training signal is used or not. One class of these algorithms is the non-blind adaptive algorithm in which a training signal is used to adjust the array weight vector.

Another technique is to use a blind adaptive algorithm, which does not require a training signal [4].

Since the non-blind algorithms use a training signal, during the training period, data cannot be sent over the radio channel. This reduces the spectral efficiency of the system.

3.2 Non-blind Adaptive Algorithms

In a non-blind adaptive algorithm, a training signal, $d(n)$, which is known to both the transmitter and receiver, is sent from the transmitter to the receiver during the training period. The beamformer in the receiver uses the information of the training signal to compute the optimal weight vector, w_{opt}. After the training period, data is sent and the beamformer uses the weight vector computed previously to process the received signal. If the radio channel and the interference characteristics remain constant from one training period until the next, the weight vector w_{opt} will contain the information of the channel and the interference, and their effect on the received signal will be compensated in the output of the array.

3.2.1 Wiener Solution

Most of the non-blind algorithms try to minimize the mean-squared error between the desired signal $d(t)$ and the array output $z(t)$. Let $z(n)$ and $d(n)$ denote the sampled signal of $z(t)$ and $d(t)$ at time instant t_n, respectively. Then the error signal is given by [15]

$$e(n) = d(n)-z(n) \tag{3.1}$$

the estimation error $e(n)$ is the sample value of a random variable. To optimize the spatial filter, we choose to minimize the mean-square value of the estimation error $e(n)$. We define the cost function as the mean-squared error

$$J = E[e(n)e^*(n)] = E[|e(n)|^2] \tag{3.2}$$

where E denotes the statistical expectation operator. Substituting equation (3.1) and (2.3) into equation (3.2), we have

$$\begin{aligned} J &= E[|d(n) - z(n)|^2] \\ &= E[\{d(n) - \mathbf{w}^H\mathbf{u}(n)\}\{d(n) - \mathbf{w}^H\mathbf{u}(n)\}^*] \\ &= E[|d(n)|^2 - d(n)\mathbf{u}^H(n)\mathbf{w} - \mathbf{w}^H\mathbf{u}(n)d^*(n) + \mathbf{w}^H\mathbf{u}(n)\mathbf{u}^H(n)\mathbf{w}] \\ &= E[|d(n)|^2] - \mathbf{p}^H\mathbf{w} - \mathbf{w}^H\mathbf{p} + \mathbf{w}^H\mathbf{R}\mathbf{w} \end{aligned} \tag{3.3}$$

where

$$R = E[\mathbf{u}(n)\mathbf{u}^H(n)] \tag{3.4}$$

and

$$\mathbf{p} = E[\mathbf{u}(n)d^*(n)] \tag{3.5}$$

In equation (3.3), R is the M-by-M correlation matrix of the input data vector u(n), and p is the M-by-1 cross-correlation vector between the input data vector and the desired signal d(n).

The gradient of a function of a complex vector, is defined as

$$\nabla f(\mathbf{w}) = \begin{bmatrix} \frac{\partial}{\partial a_0} f(\mathbf{w}) \\ \cdots \\ \frac{\partial}{\partial a_{M-1}} f(\mathbf{w}) \end{bmatrix} + j \begin{bmatrix} \frac{\partial}{\partial b_0} f(\mathbf{w}) \\ \cdots \\ \frac{\partial}{\partial b_{M-1}} f(\mathbf{w}) \end{bmatrix} \tag{3.6}$$

where $w_i = a_i + jb_i$. In general, we minimize a vector function by determining a location where the gradient of the function goes to zero, i.e.,

$$\nabla(J) \mid \mathbf{w}_{opt} = 0 \tag{3.7}$$

Substituting equation (3.3) into equation (3.7), we have

$$-2\mathbf{p} + 2\mathbf{R}\mathbf{w}_{opt} = 0 \tag{3.8}$$

we find that the solution for $\mathbf{w}_{opt}$ which minimizes J is

$$\mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{p} \tag{3.9}$$

The optimum weight vector $\mathbf{w}_{opt}$ in equation (3.9) is called the *Wiener solution*. From equation (3.9), we see that the computation of the optimum weight vector $\mathbf{w}_{opt}$ requires knowledge of two quantities: (1) the correlation matrix R of the input data vector $\mathbf{u}(n)$, and (2) the cross-correlation vector $\mathbf{p}$ between the input data vector $\mathbf{u}(n)$ and the desired signal $d(n)$.

3.2.2 Method of Steepest-Descent

Rather than solving (3.9) directly, adaptive techniques are often used with an iterative approach which provides an updated weight vector, $\mathbf{w}_{opt}$, after each computation. This is known as *method of steepest-descent* [10]. To find the optimum weight vector $\mathbf{w}_{opt}$ by the steepest-descent method we proceed as follows:

1. Begin with an initial value $\mathbf{w}(0)$ for the weight vector, which is chosen arbitrarily. Typically, $\mathbf{w}(0)$ is set equal to a column vector of an M-by-M identity matrix.

2. Using this initial or present guess, compute the gradient vector, the real and imaginary parts of which are defined as the derivative of the mean-squared error $J(n)$, evaluated with respect to the real and imaginary parts of the tap-weight vector $\mathbf{w}(n)$ at time n (i.e., the n^{th} iteration).

3. Compute the next guess at the weight vector by making a change in the initial or present guess in a direction opposite to that of the gradient vector.

4. Go back to step 2 and repeat the process.

It is intuitively reasonable that successive corrections to the weight vector in the direction of the negative of the gradient vector should eventually lead to the minimum mean-squared error J_{min}, at which the weight vector assumes its optimum value $\mathbf{w}_{opt}$.

Let w(n) denote the value of the weight vector at time n. According to the method of steepest-descent, the update value of the weight vector at time n + 1 is computed by using the simple recursive relation

$$\mathbf{w}(n + 1) = \mathbf{w}(n) + \frac{1}{2}\mu[-\nabla(J(n))] \tag{3.10}$$

where μ is a positive real-valued constant. The factor 1/2 is used merely for convenience. From equation (3.8) we have

$$\nabla(J(n)) = -2\mathbf{p} + 2\mathbf{R}\mathbf{w}(n) \tag{3.11}$$

Substituting equation (3.11) into (3.10), we obtain

$$\mathbf{w}(n + 1) = \mathbf{w}(n) + \mu[\mathbf{p} - \mathbf{R}\mathbf{w}(n)], \quad n = 0,1,2,... \tag{3.12}$$

We observe that the parameter μ controls the size of the incremental correction applied to the weight vector as we proceed from one iteration cycle to the next. We therefore refer to μ as the *step-size parameter* or weighting constant. Equations (3.12) describe the mathematical formulation of the steepest-descent method.

3.2.3 Least-Mean-Squares Algorithm

If it were possible to make exact measurements of the gradient vector $\nabla(J(n))$ at each iteration, and if the step-size parameter μ is suitably chosen, then the weight vector computed by using the steepest-descent method would indeed converge to the optimum Wiener solution. In reality, however, exact measurements of the gradient vector are not possible since this would require prior knowledge of both the correlation matrix $\mathbf{R}$ of the input data vector and the cross-correlation vector $\mathbf{p}$ between the input data vector and the desired signal.

Consequently, the gradient vector must be estimated from the available data. In other words, the weight vector is updated in accordance with an algorithm that adapts to the incoming data. One such algorithm is the *least-mean-squares (LMS) algorithm* [15][1]. A significant feature of the LMS algorithm is its simplicity: it does not require measurements of the pertinent correlation functions, nor does it require matrix inversion.

Using equation (3.1), (3.4), (3.5), the gradient vector in equation (3.12) may be written in another form

$$\begin{aligned} \nabla(J(n)) &= -2E[\mathbf{u}(n)d^*(n) - \mathbf{u}(n)\mathbf{u}^H(n)\mathbf{w}(n)] \\ &= -2E[\mathbf{u}(n)\{d(n) - z(n)\}^*] \\ &= -2E[\mathbf{u}(n)e^*(n)] \end{aligned} \tag{3.13}$$

To develop an estimate of the gradient vector $\nabla(J(n))$, the most obvious strategy is to substitute the expected value in equation (3.13) with the instantaneous estimate,

$$\hat{\nabla}(J(n)) = -2\mathbf{u}(n)e^*(n) \tag{3.14}$$

Substituting this instantaneous estimate of the gradient vector into equation (3.10), we have

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu\mathbf{u}(n)e^{*}(n) \tag{3.15}$$

Equivalently, we may write the result in the form of three basic relations as follows:

$$z(n) = \mathbf{w}^{H}(n)\mathbf{u}(n) \tag{3.16}$$

$$e(n) = d(n) - z(n) \tag{3.17}$$

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu\mathbf{u}(n)e^{*}(n) \tag{3.18}$$

The algorithm described by equation (3.16) to (3.18) is the complex form of the adaptive *least-mean-square (LMS) algorithm*. The LMS algorithm is a member of a family of stochastic gradient algorithms since the instantaneous estimate of the gradient vector is a random vector that depends on the input data vector x(k).

The LMS algorithm requires only 2M complex multiplications and 2M+1 complex additions per iteration, where M is the number of weights (elements) used in the adaptive array.

The step-size parameter μ have to satisfy

$$0 < \mu < \frac{2}{\sum_{k=0}^{M-1} E[|u(n-k)|^{2}]} \tag{3.19}$$

Refer to [15] for detailed analysis of LMS algorithm.

3.2.4 Normalized Least-Mean-Squares Algorithm

In the standard form of LMS algorithm, the correction $\mu\mathbf{u}(n)e^{*}(n)$ applied to the weight vector w(n) at iteration n+1 is directly proportional to the input vector u(n).

Therefore, when **u**(n) is large, the LMS algorithm experiences a gradient noise amplification problem. to overcome this difficulty, we may use the normalized LMS algorithm, which is the companion to the ordinary LMS algorithm. In particular, the correction applied to the weight vector **w**(n) at iteration n+1 is "normalized" with respect to the squared Euclidean norm of the input vector **u**(n) at iteration n [15].

By setting

$$\mu(n) = \frac{\tilde{\mu}}{\|\mathbf{u}(n)\|^2} \tag{3.20}$$

we may view the normalized LMS algorithm as an LMS algorithm with a *time-varying step-size parameter*. To overcome the numerical difficulties caused by small u(n), we can slightly modify equation (3.20) and finally we can get the recursion equation for NLMS algorithm

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\tilde{\mu}}{a + \|\mathbf{u}(n)\|^2}\mathbf{u}(n)e^*(n) \tag{3.21}$$

where $a > 0$, and $0 < \tilde{\mu} < 2$.

The normalized LMS algorithm exhibits a rate of convergence that is potentially faster than that of the standard LMS algorithm for both uncorrelated and correlated input data.

3.2.5 Recursive Least-Squares Algorithm

Unlike the LMS algorithm which uses the method of steepest-descent to update the weight vector, the recursive *least-squares (RLS) algorithm* [15] uses the *method of least squares* to adjust the weight vector. In the method of least squares, we choose the weight vector **w**(n), so as to minimize a cost function that consists of the sum of error squares over a

time window. In the method of steepest-descent, on the other hand, we choose the weight vector to minimize the ensemble average of the error squares.

In the exponentially weighted RLS algorithm, at time n, the weight vector is chosen to minimize the cost function

$$\varepsilon(n) = \sum_{i=1}^{n} \lambda^{n-i} |e(i)|^2 \tag{3.22}$$

where *e(i)* is defined in equation (3.1), and is a positive constant close to, but less than, one. When λ equals 1, we have the ordinary method of least squares. λ determines how quickly the previous data are de-emphasized. In a stationary environment, λ should be equal to 1, since all data past and present should have equal weight.

The optimum value of the weight vector, w(n), for which the cost function $\varepsilon(n)$ of equation (3.22) attains its minimum value is defined by the normal equations written in matrix form:

$$\Phi(n)w(n) = z(n) \tag{3.23}$$

The M-by-M correlation matrix $\Phi(n)$ is now defined by

$$\begin{aligned} \Phi(n) &= \sum_{i=1}^{n} \lambda^{n-i} \mathbf{u}(i)\mathbf{u}^H(i) \\ &= \lambda\left[\sum_{i=1}^{n-1} \lambda^{n-1-i} \mathbf{u}(i)\mathbf{u}^H(i)\right] + \mathbf{u}(n)\mathbf{u}^H(n) \\ &= \lambda\Phi(n-1) + \mathbf{u}(n)\mathbf{u}^H(n) \end{aligned} \tag{3.24}$$

The M-by-1 cross-correlation vector **z**(n) between the tap inputs of the transversal filter and the desired response is correspondingly defined by

$$\begin{aligned}\mathbf{z}(n) &= \sum_{i=1}^{n} \lambda^{n-i}\mathbf{u}(i)d^{*}(i) \\ &= \lambda\mathbf{z}(n-1) + \mathbf{u}(n)d^{*}(n)\end{aligned} \tag{3.25}$$

The recursive equation for updating the least-squares estimate weight vector w(n) is

$$\mathbf{w}(n) = \Phi^{-1}(n)\mathbf{z}(n) \tag{3.26}$$

In order to get $\Phi^{-1}(n)$, we use the matrix inversion lemma. Let **A** and **B** be two positive-definite M-by-M matrices related by

$$\mathbf{A} = \mathbf{B}^{-1} + \mathbf{C}\mathbf{D}^{-1}\mathbf{C}^{H} \tag{3.27}$$

where D is another positive-definite N-by-M matrix, and C is an M-by-N matrix. According to the *matrix inversion lemma*, we may express the inverse of the matrix A as follows:

$$\mathbf{A}^{-1} = \mathbf{B} - \mathbf{B}\mathbf{C}(\mathbf{D} + \mathbf{C}^{H}\mathbf{B}\mathbf{C})^{-1}\mathbf{C}^{H}\mathbf{B} \tag{3.28}$$

Now let

$$\begin{aligned}\mathbf{A} &= \Phi(n) \\ \mathbf{B} - 1 &= \lambda\Phi(n-1) \\ \mathbf{C} &= \mathbf{u}(n) \\ \mathbf{D} &= 1\end{aligned}$$

Then substituting these definitions in equation (3.28), we obtain following recursive equation for the inverse of the correlation matrix:

$$\Phi^{-1}(n) = \lambda^{-1}\Phi^{-1}(n-1) - \frac{\lambda^{-2}\Phi^{-1}(n-1)\mathbf{u}(n)\mathbf{u}^{H}(n)\Phi^{-1}(n-1)}{1 + \lambda^{-1}\mathbf{u}^{H}(n)\Phi^{-1}(n-1)\mathbf{u}(n)} \tag{3.29}$$

For convenience of computation, let

$$\mathbf{P}(n) = \Phi^{-1}(n) \tag{3.30}$$

and

$$\mathbf{k}(n) = \frac{\lambda^{-1}\mathbf{P}(n-1)\mathbf{u}(n)}{1 + \lambda^{-1}\mathbf{u}^H(n)\mathbf{P}(n-1)\mathbf{u}(n)} \tag{3.31}$$

hence (3.28) can be written as:

$$\mathbf{P}(n) = \lambda^{-1}\mathbf{P}(n-1) - \lambda^{-1}\mathbf{k}(n)\mathbf{u}^H(n)\mathbf{P}(n-1) \tag{3.32}$$

From equation (3.30) and (3.31) we can get

$$\mathbf{k}(n) = \mathbf{P}(n)\mathbf{u}(n) \tag{3.33}$$

Use all these equations to (3.25), finally we can get the desired recursive equation for updating the weight vector:

$$\begin{aligned} \mathbf{w}(n) &= \mathbf{w}(n-1) + \mathbf{k}(n)[d^*(n) - \mathbf{u}^H(n)\mathbf{w}(n-1)] \\ &= \mathbf{w}(n-1) + \mathbf{k}(n)\xi^*(n) \end{aligned} \tag{3.34}$$

where $\xi(n)$ is a *priori estimation error* defined by

$$\xi(n) = d(n) - \mathbf{w}^H(n-1)\mathbf{u}(n) \tag{3.35}$$

Equations (3.31), (3.35), (3.34), and (3.32), collectively and in that order, constitute the RLS algorithm.

We can initialize the algorithm by setting

$$\mathbf{P}(0) = \delta^{-1}\mathbf{I}$$

where I is the M -by-M identity matrix, and δ is a small positive constant.

An important feature of the RLS algorithm is that it utilizes information contained in the input data, extending back to the instant of time when the algorithm is initiated. The resulting rate of convergence is therefore typically an order of magnitude faster than the simple LMS algorithm. This improvement in performance, however, is achieved at the expense of a large increase in computational complexity. The RLS algorithm requires $3M^2 + 4M + 2$ complex multiplications and $4M^2 - 1$ complex additions per iteration, where M is the number of weights used in the adaptive array.

3.3 Blind Adaptive Algorithms

Other techniques have been developed which do not require a training sequence. These techniques, generally referred to as *blind adaptive algorithms*, adapt by attempting to restore some known property to the received signal [1]. Some of the blind algorithms need complex matrix inversion, which is too computational complex to be used in practice. This section will focus on two simple blind algorithms that will be used in our simulation, and also give a brief introduction to other kinds of blind algorithms can be used in adaptive antenna array.

3.3.1 Constant Modulus Algorithm

Some communication signals such as phase-shift keying (PSK), frequency-shift keying (FSK), and analog FM signals have a constant envelope. This constant envelope may be distorted when the signal is transmitted through the channel. The *constant modulus algorithm* (CMA) [20][21][1] 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 desired signal.

The CMA tries to minimize the cost function

$$J(n) = E\left[\left|\,|z(n)|^p - |\alpha|^p\,\right|^q\right] \tag{3.36}$$

where α is the desired signal amplitude at the array output, and $z(n) = \mathbf{w}_n^H \mathbf{u}_n$. An adaptive array which uses the CM cost-function will attempt to drive the signal at the array output to have a constant envelope with the specified amplitude, α. The exponents p and q are each equal to either 1 or 2. Using different values of p and q, it is possible to develop several different steepest-descent algorithms which have different convergence characteristics and complexity.

For PSK case, $\alpha = 1$, and we use J with $p = 1$, $q = 2$. With the cost function of this 1-2 form, the CMA minimizes the function

$$J(n) = E\left[\,|\,|z(n)| - 1\,|^2\right] \tag{3.37}$$

The gradient vector is given by [4]

$$\begin{aligned}
\nabla(J(n)) &= 2\frac{\partial J(n)}{\partial \mathbf{w}^*(n)} \\
&= 2E\left[(|z(n)| - 1)\frac{\partial |z(n)|}{\partial \mathbf{w}^*(n)}\right] \\
&= 2E\left[(|z(n)| - 1)\frac{\partial \left\{z(n)z^*(n)\right\}^{\frac{1}{2}}}{\partial \mathbf{w}^*(n)}\right] \\
&= 2E\left[|(z(n)| - 1)\frac{\partial \left\{\mathbf{w}^H(n)\mathbf{u}(n)\mathbf{u}(n)\mathbf{w}(n)\right\}^{\frac{1}{2}}}{\partial \mathbf{w}^*(n)}\right]
\end{aligned}$$

$$
\begin{aligned}
&= E\left[(|z(n)| - 1)\left\{\mathbf{w}^H(n)\mathbf{u}(n)\mathbf{u}(n)\mathbf{w}(n)\right\}^{-\frac{1}{2}} \cdot \frac{\partial\left\{\mathbf{w}^H(n)\mathbf{u}(n)\mathbf{u}^H(n)\mathbf{w}(n)\right\}}{\partial \mathbf{w}^*(n)}\right] \\
&= E\left[(|z(n)| - 1)\frac{\mathbf{u}(n)\mathbf{u}(n)^H\mathbf{w}(n)}{|z(n)|}\right] \\
&= E\left[(1 - \frac{1}{|z(n)|})\mathbf{u}(k)\mathbf{z}^*(n)\right] \\
&= E\left[\mathbf{u}(n)\left(z(n) - \frac{z(n)}{|z(n)|}\right)^*\right]
\end{aligned} \tag{3.38}
$$

Ignoring the expectation operator in equation (3.38), the instantaneous estimate of the gradient vector can be written as

$$
\hat{\nabla}(J(n)) = \mathbf{u}(n)\left(z(n) - \frac{z(n)}{|z(n)|}\right)^* \tag{3.39}
$$

Using the method of steepest-descent, and replacing the gradient vector with its instantaneous estimate, we can update the weight vector by

$$
\begin{aligned}
\mathbf{w}(n+1) &= \mathbf{w}(n) - \mu\hat{\nabla}(J(n)) \\
&= \mathbf{w}(n) - \mu\mathbf{u}(n)\left(z(n) - \frac{z(n)}{|z(n)|}\right)^*
\end{aligned} \tag{3.40}
$$

where μ is the step-size parameter.

Now, the CMA algorithm can be described by the following three equations

$$
z(n) = \mathbf{w}^H(n)\mathbf{u}(n) \tag{3.41}
$$

$$
e(n) = z(n) - \frac{z(n)}{|z(n)|} \tag{3.42}
$$

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu\mathbf{u}(n)e^{*}(n) \tag{3.43}$$

From equation (3.42) we see that when the output of the array has a unity magnitude, i.e., $|(z(n)| = 1$, the error signal becomes zero. Comparing the above three equations with equation (3.16), (3.17), and (3.18), we see that the CMA is very similar to the LMS algorithm, and the term $\frac{z(n)}{|(z(n)|}$ in CMA plays the same role as the desired signal d(n) in the LMS algorithm. However, the reference signal d(t) must be sent from the transmitter to the receiver and must be known for both the transmitter and receiver if the LMS algorithm is used. The CMA algorithm does not require a reference signal to generate the error signal at the receiver.

Several properties of the constant modulus algorithm are discussed in [20]. One of the properties is the *phase roll*. Let $z(n) = m \cdot e^{j\varphi}$ in (3.42), we have

$$e(n) = (m \pm 1) \cdot e^{j\varphi} \tag{3.44}$$

note $|z(n)| = m$. From (3.44) we can see that CMA uses only the amplitude information of the array output in the control of the error and the phase information φ is not utilized. This means that in CMA, if a weight vector $\mathbf{w}$ generates an array output with constant modulus, so does its phase-shifted version [20],

$$\mathbf{w}_r = e^{j\varphi}\mathbf{w} \tag{3.45}$$

The behavior of CMA is discussed in detail in [20][21][22][23].

3.3.2 Algorithms based on Eigenvector Estimation

This class of blind algorithms is presented in [30]-[35] and is applied to the base station of CDMA systems. The algorithms are designed to compute the eigenvector corresponding to the largest eigenvalue of an autocovariance matrix. The characteristic of this algorithm is its low computational complexity after some simplification.

The autocovariance matrix of the received signal is computed as follows [31][32][33]:

$$\mathbf{R}uu(n) = f\mathbf{R}uu(n-1) + \mathbf{u}(n)\mathbf{u}^{H}(n) \tag{3.46}$$

where the forgetting factor f lies in $0 \leq f \leq 1$, and $\mathbf{u}(n)$ is the received signal vector at n^{th} snapshot.

When the desired signal is sufficiently larger than each of the interferers, the maximum eigenvector $\mathbf{e}_1$ of the autocovariance matrix $\mathbf{R}uu(n)$ can be approximated as [31]

$$\mathbf{e}_1 \approx \mathbf{a}(\theta_1) \tag{3.47}$$

where $\mathbf{a}(\theta_1)$ is *steering vector* as described in (2.8) and θ_1 is the arrival angle of the desired signal, which changes at every snapshot as the signal source moves. When there are more than one comparable unresolvable strong paths (will be explained in Chapter 4), the maximum eigenvector is

$$\mathbf{e}_1 \approx \sum_{i=1}^{K} \varsigma_i a(\theta_i) \tag{3.48}$$

where K is the number of the strong multipaths and ς_i is a constant that is determined by the propagation characteristics of the i^{th} path. In this case, the array provides a beampattern of

multiple main-beams each of which is generated along the direction of each strong path.

From (3.47) and (3.48), it can be observed that the maximum eigenvector e_1 of the autocovariance matrix $\mathbf{R}uu(n)$ forms the suboptimal weight vector, once the desired signal transmitted from the target subscriber is sufficiently larger than each interferer at the receiving array. The more dominant is the desired signal, the closer the eigenvector is to the target steering vector. In Chapter 4, we will see that this condition can be satisfied in post-correlated CDMA signal.

The adaptive procedure of this blind algorithm is based on the maximization of the following functional [31]

$$J(\mathbf{w}, \gamma) = \mathbf{w}^H \mathbf{R}uu\mathbf{w} + \gamma(1 - \mathbf{w}^H \mathbf{w}) \tag{3.49}$$

where γ is Lagrange multiplier. The maximum eigenvector can be found by searching for a vector $\mathbf{w}$ that maximizes the functional $J(\mathbf{w}, \gamma)$ shown in (3.49) with a constraint $\mathbf{w}^H\mathbf{w} = 1$ [31][33]. In order to find the target eigenvector in an iterative way, starting from an initial guess $\mathbf{w}(0)$, the weight vector is updated as follows:

$$\mathbf{w}(n + 1) = \mathbf{w}(n) + \frac{1}{2}\mu[\nabla(J(n))] \tag{3.50}$$

From (3.49), the gradient ∇ can be computed as

$$\nabla = 2(\mathbf{R}uu\mathbf{w} - \gamma\mathbf{w}) \tag{3.51}$$

which states that the solution for the extreme value of the functional $J(\mathbf{w}, \gamma)$ is an eigenvalue for γ and corresponding eigenvector for $\mathbf{w}$ (by observing that $\nabla = 0$). From

(3.50) and (3.51) we have

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu(\mathbf{R}uu\mathbf{w} - \gamma\mathbf{w}) \tag{3.52}$$

Noting that the constraint should be satisfied at each iteration step, i.e., $\mathbf{w}^H(n+1)\cdot\mathbf{w}(n+1) = 1$, the n^{th} snapshot $\gamma(n)$ can be found to be the solution of the following quadratic equation:

$$\begin{aligned}\mu\gamma^2(n) - 2[1 + \mu\mathbf{w}^H(n)\mathbf{R}uu(n)\mathbf{w}(n)]\gamma(n) + \mu\mathbf{w}^H(n)\mathbf{R}^2uu(n)\mathbf{w}(n) \\ + 2\mathbf{w}^H(n)\mathbf{R}uu(n)\mathbf{w}(n) = 0\end{aligned} \tag{3.53}$$

In order to reduce the computational complexity, the autocovariance matrix can be approximated with the instantaneous signal vector at each snapshot as:

$$\mathbf{R}uu(n) = \mathbf{u}(n)\mathbf{u}^H(n) \tag{3.54}$$

From (3.53) and (3.54), the Lagrange multiplier can be found as

$$\gamma(n) = \frac{b \pm \sqrt{b^2 - ac}}{a} \tag{3.55}$$

where

$$\begin{aligned} a &= \mu \\ b &= \mu\|z\|^2 + 1 \\ c &= \|z\|^2(2 + \mu\|u\|^2) \end{aligned} \tag{3.56}$$

and

$$z(n) = \mathbf{w}^H(n)\mathbf{u}(n) \tag{3.57}$$

Substituting (3.54) and (3.55) into (3.52), we can finally get the weight update equation.

Later on, we abbreviate the algorithm described above BEE (Blind Eigenvector Estimation) algorithm.

3.3.3 Least-Squares CMA

The Least_Squares Constant Modulus Algorithm (LS-CMA) was proposed by Agee using an extension of the method of nonlinear least squares (Gauss's Method) [30].

Gauss's method exploits the property that a (real-argument) cost function with the form

$$F(\mathbf{w}) = \sum_{n=1}^{N} \phi_n^2(\mathbf{w}) = \left\|\Phi(\mathbf{w})\right\|_2^2 \tag{3.58}$$

has a partial Taylor-series expansion with sum-of-squares form

$$F(\mathbf{w} + \Delta) \approx \left\|\Phi(\mathbf{w}) + \mathbf{J}^T(\mathbf{w})\Delta\right\|_2^2 \tag{3.59}$$

where $\mathbf{J}(\mathbf{w})$ is the Jacobian of $\Phi(\mathbf{w})$,

$$J(w) = [\nabla\phi_1(\mathbf{w}), \nabla\phi_2(\mathbf{w}),...\nabla\phi_N(\mathbf{w})] \tag{3.60}$$

Gauss's method updates w by the offset Δ which minimizes this sum-of-squares, resulting in optimization algorithm and we expand it to the complex-argument expression:

$$\mathbf{w}_{k+1} = \mathbf{w}_k - [\mathbf{J}(\mathbf{w}_k)\mathbf{J}^H(\mathbf{w}_k)]^{-1}\mathbf{J}(\mathbf{w}_k)\Phi(\mathbf{w}_k) \tag{3.61}$$

The LS-CMA is derived by applying equation (3.61) to the "1-2" constant modulus cost function (i.e., $p = 1$, $q = 2$ in (3.36)):

$$F(\mathbf{w}) = \sum_{n=1}^{N} (|z(n)| - 1)^2 = \| [|z(n)| - 1] \|_z^2 \tag{3.62}$$

Substituting equation (3.62) into (3.61) and assuming a linearly-independent set of input data yields the LS-CMA,

$$\begin{aligned} \mathbf{w}_{k+1} &= \mathbf{w}_k - (\mathbf{U} \cdot \mathbf{U}^H)^{-1}\mathbf{U}(\mathbf{z}_k - \boldsymbol{\delta}_k) \\ &= (\mathbf{U} \cdot \mathbf{U}^H)^{-1}\mathbf{U}\boldsymbol{\delta}_k \end{aligned} \tag{3.63}$$

where $\mathbf{U}$ is the input data matrix and $\mathbf{z}_k$ and $\boldsymbol{\delta}_k$ are the output-data and complex-limited output-data vectors,

$$\begin{aligned} \mathbf{U} &= [\mathbf{u}(1), \mathbf{u}(2),...\mathbf{u}(N)] \\ \mathbf{z}(k) &= [z_k(n)] = \mathbf{w}^H(k)\mathbf{U}(k) \\ \boldsymbol{\delta}_k &= [\delta_k(n)] = [z_k(n) / |z_k(n)|] \end{aligned} \tag{3.64}$$

Unlike the steepest descent CMA algorithms discussed in the previous section, which update on a sample-by-sample basis, the dynamic LS-CMA approach updates on a block-by-block basis.

The LS-CMA algorithm converges significantly (10 to 100 times) faster than the steepest descent CMA [30]. But as we can see in equation (3.63), the operation of this algorithm needs the computation of the matrix inversion, which presents very high computational complexity.

3.3.4 Other Blind Beamforming Algorithms

Other LS-CMA based algorithms include Multitarget least Squares Constant Modulus Algorithm (MT-LSCMA), Multitarget Decision-Directed Algorithm (MT-

DD), Least Squares De-spread Re-spread Multitarget Array (LS-DRMTA) and least Squares De-spread Re-spread Multitarget Constant Modulus Algorithm (LS-DRMTCMA). All of these algorithms are developed for CDMA systems and are presented in detail in [1] and [4], we also give a brief introduction of them in the next Chapter.

Another different kinds of algorithms are based on Estimation of Direction of Arrivals (DOAs) of received signals. In the DOA-estimation-based algorithms, the DOAs of the received signals are first determined by using prior knowledge of the array response, i. e., the array manifold or special array structure. The high-resolution techniques for DOA estimation include MUSIC and ESPRIT [15]. After the DOAs are estimated, an optimum beamformer is then constructed from the corresponding array response to extract the desired signals from interference and noise [16]. The performance of this technique strongly depends on the reliability of the prior spatial information, e.g., the array manifold. In many situations of practical interest, this information is not available. Even if this information is available, the cost is very expensive and the information may be inaccurate. The computational complexity of these algorithms is also very high. Another disadvantage of this technique is that the number of DOAs that the algorithm can estimate is limited by the number of array elements.

In a wireless communication system, especially in a code division multiple access (CDMA) system, the number of users in a radio channel may be greater than the number of array elements, and if the multipaths of each user's signal are taken into account, the total number of signals impinging on the array will far exceed the number

of array elements, and the DOA estimation algorithm in this case will fail. Furthermore, this approach does not exploit the known temporal structure of the incoming signals [4].

Chapter 4

CDMA Communication Systems and Adaptive Antenna Techniques

4.1 Introduction

In Section 4.2, we briefly review the concept of spread spectrum and Code Division Multiple Access (CDMA) techniques. Section 4.3 introduce signal model of Antenna Array in Fading Channel and the types of fading channels. Section 4.4 reviews IS-95 CDMA Reverse link and Section 4.5 gives our CDMA beamforming structure and its application on IS-95 CDMA Reverse link.

4.2 Spread Spectrum and Code Division Multiple Access

Traditionally, in radio communication systems, the carrier was modulated with user data using techniques that minimize the transmitted bandwidth to conserve spectrum resources. This was because radio systems were designed so that only a single channel occupied a given frequency band. If signals are transmitted in multiple non-overlapping frequency bands, they do not interfere with each other and the signals may each be recovered, provided that the power levels are high enough relative to the noise which is always present in the channel. In a spread spectrum system, rather than trying to minimize the bandwidth of the modulated signal, the goal is to create a modulated signal that uses a large amount of bandwidth. There are two main types of spread spectrum systems: direct sequence (DS) and frequency hop (FH) [1][5]. In this thesis, we only discuss the DS spread spectrum systems.

4.2.1 Direct-Sequence Spread Spectrum

In DS systems, a narrowband signal is multiplied by a signal with a much larger bandwidth, which is called the spreading signal. The spreading signal is chosen to have properties which facilitate demodulation of the transmitted signal by the intended receiver, and which make demodulation by an unintended receiver as difficult as possible. These same properties will also make it possible for the intended receiver to discriminate between the communication signal and jamming. If the bandwidth of the spreading signal is large relative to the data bandwidth, the spread-spectrum transmission bandwidth is dominated by the spreading signal and is nearly independent of the data signal.

The spreading signal is comprised of symbols that are defined by a pseudo random sequence which is known to both the transmitter and receiver. These symbols are called *chips*. Typically, the rate of chips in the spreading code, R_c, is much greater than the symbol rate, R_d, of the original data sequence. The pseudo random chip sequence is also called a *Pseudo Noise* (PN) sequence because the power spectral density of the PN-sequence looks approximately like white noise filtered to have the same spectral envelope as a chip. The spreading factor, or processing gain, is defined as the ratio of the chip rate to the data symbol rate, or $N = R_c / R_d$ [1].

Let's consider an information signal $b(t)$, which is a digital signal composed of a sequence of symbols, b_j, each of duration T_s. The data signal is given by

$$b(t) = \sum_{j=-\infty}^{\infty} b_j p_{T_s}(t - jT_s) \tag{4.1}$$

where $p_{T_s}(t)$ is the pulse shape of the bit. We assume a rectangular pulse shape:

$$p_{T_s}(t) = \begin{cases} 1 & 0 \le t \le T_s \\ 0 & otherwise \end{cases} \tag{4.2}$$

This signal is multiplied by a PN sequence *a(t)*:

$$a(t) = \sum_{j=-\infty}^{\infty} a_j p_{T_c}(t - jT_c) \tag{4.3}$$

where a_j is the j^{th} chip of a discrete periodic PN sequence assigned to the user and takes on values of $a_j \in \{\pm 1\}$ with a duration of T_c. $p_{T_c}(t)$ has a same function with $p_{T_s}(t)$.

The transmitted signal *s(t)* is then (if we assume the initial phase is 0)

$$s(t) = a(t)b(t)\cos(\omega_c t) \tag{4.4}$$

Assume that the channel does not distort the signal in any way, so that the received signal, *r(t)*, consists of a weighted version of the transmitted signal with added white Gaussian noise, *n(t)*:

$$r(t) = As(t) + n(t) = Aa(t)b(t)\cos(\omega_c t) + n(t) \tag{4.5}$$

At the receiver, a local replica of the chip sequence, $a(t - \tau_0)$, is generated, where τ_0 is a random time offset between 0 and MT_c. To despread the signal at the receiver, the local chip sequence must be synchronized with the received signal. This means that the timing offset, τ_0, must be set to zero. One means of doing this is through the use of a Delay Locked Loop (DLL). Similarly, a Phase Locked Loop (PLL) may be used to create a replica of the carrier, $\cos(\omega_c t)$. Now assume that $a(t)a^*(t) = 1$, using these two quantities, a decision

statistic is formed by multiplying the received signal by the local PN sequence and the local oscillator and integrating the result over one data symbol:

$$\begin{aligned} Z_j &= \int_{jTs}^{(j+1)Ts} r(t)a^*(t)\cos(\omega_c t)dt \\ &= Ab_j\left(\frac{T_s}{2} + \frac{1}{4\omega_c}(\sin(2\omega_c(j+1)T_s) - \sin(2\omega_c jT_s))\right) + \eta \end{aligned} \tag{4.6}$$

where η represents the influence of channel noise on the decision statistic, and

$$\eta = \int_{jTs}^{(j+1)Ts} n(t)a^*(t)\cos(\omega_c t)dt \tag{4.7}$$

Assuming that the carrier frequency is large relative to the reciprocal of the bit period. i.e., $\omega_c >> 1 / T_s$, then

$$Z_j = \frac{Ab_jT_s}{2} + \eta \tag{4.8}$$

Therefore, the decision statistic, Z_j, is an estimate, $\hat{b}_j$, of the transmitted data symbol b_j.

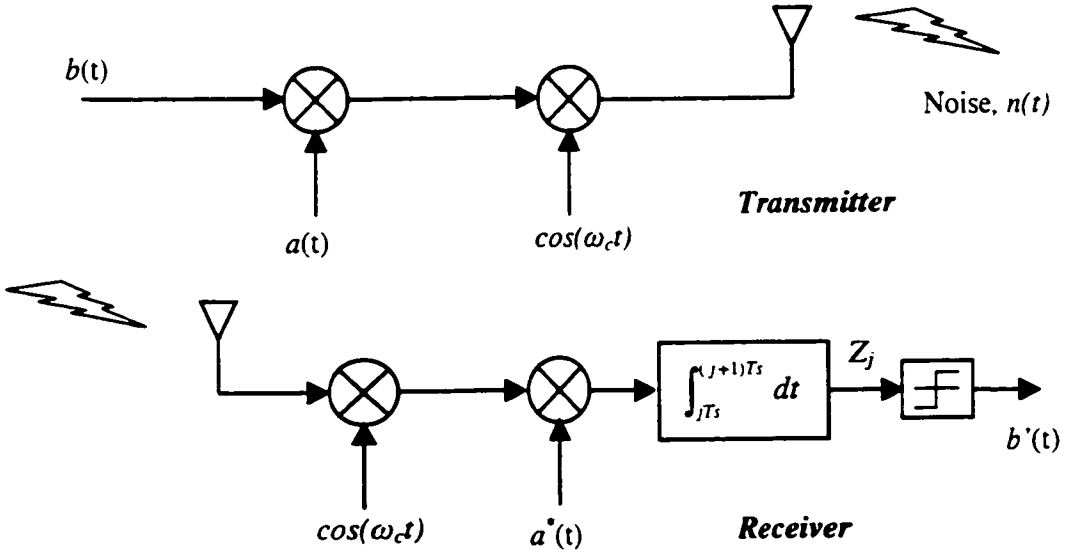


Figure 4-1 A DS spread spectrum transmitter and receiver

Figure 4-1 illustrate a DS spread spectrum transmitter and receiver.

If the channel noise is additive white Gaussian noise with two-sided power spectral density $N_o/2$, then the variable η is a zero mean Gaussian random variable $u_\eta = 0$ with variance $\sigma_\eta^2 = \frac{N_o T_s}{4}$ [1].

Spreading spectrum techniques have the following advantages:

1. Small spectral density of the signal makes it difficult for a potential eavesdropper to detect the presence of the signal.

2. Significantly mitigate the effects of fading which can degrade the performance of narrowband systems.

3. Many DS signals can be overlaid on top of each other in the same frequency band.

4.2.2 Multiple Access Interference in DS-CDMA Systems

In the uplink of CDMA system, the signals from different users interfere with each other, resulting in higher bit error rates as compared with orthogonal CDMA. In this case, every user contributes interference, called Multiple Access Interference (MAI).

Consider a CDMA system in which K users occupy the same frequency band at the same time. These users may share the same cell, or some of the K users may communicate with other base stations.

The signal received at the base station from user k is given by

$$s_k(t - \tau_k) = \sqrt{2P_k}\, a_k(t - \tau_k) b_k(t - \tau_k) \cos(\omega_c t + \phi_k) \tag{4.9}$$

where $b_k(t)$ is the data sequence for user k, $a_k(t)$ is the spreading sequence for user k, τ_k is the delay of user k relative to a reference user 0, P_k is the received power of user k, and ϕ_k is the

phase offset of user k relative to a reference user 0, and $\tau_0 = 0$ and $\phi_0 = 0$.

Assume that both $a_k(t)$ and $b_k(t)$ are binary sequences having values of −1 or +1, and we have $T_b = T_s$. At the receiver, the signal available at the input to the correlator is given by

$$r_0(t) = \sum_{k=0}^{K-1} s_k(t - \tau_k) + n(t) \tag{4.10}$$

Then the received signal is mixed down to baseband, multiplied by the PN sequence of the desired user (assume user 0) and integrated over one bit period. Thus, assuming that the receiver is delay and phase synchronized with user 0, the decision statistic for user 0 is given by

$$Z_0 = \int_{jTb}^{(j+1)Tb} r_0(t) a_1^*(t) \cos(\omega_c t) dt \tag{4.11}$$

For simplicity, assume j = 0 in (4.11), and substituting (4.9) and (4.10) into (4.11),

$$Z_0 = \int_0^{Tb} \left[\left(\sum_{k=0}^{K-1} \sqrt{2P_k} a_k(t - \tau_k) b_k(t - \tau_k) \cos(\omega_c t + \phi_k) \right) + n(t) \right] a_0^*(t) \cos(\omega_c t) dt \tag{4.12}$$

which may be expressed as

$$Z_0 = I_0 + \eta + \zeta \tag{4.13}$$

where I_0 is the contribution to the decision statistic from the desired user ($k = 0$), ζ is the multiple access interference, and η is the noise contribution. As shown in [1], the contribution from the desired user is given by

$$I_0 = \sqrt{2P_0} \int_{t=0}^{Tb} |a_0(t)|^2 b_0(t) \cos^2(\omega_c t) dt \approx \sqrt{\frac{P_0}{2}} b_{0,0} T_b \tag{4.14}$$

ζ is the summation of K-1 terms, I_k,

$$\zeta = \sum_{k=1}^{K-1} I_k \tag{4.15}$$

each of which is given by

$$I_k = \int_{t=0}^{Tb} \sqrt{2P_k} b_k(t - \tau_k) a_k(t - \tau_k) a_0^*(t) \cos(\omega_c t + \phi_k) \cos(\omega_c t) dt \tag{4.16}$$

and the total variance of interference plus noise term is

$$\sigma_\xi^2 = \sigma_\zeta^2 + \sigma_\eta^2 = \frac{NT_c^2}{6} \sum_{k=1}^{K-1} P_k + \frac{N_0 T_b}{4} \tag{4.17}$$

where N is the spreading factor.

In Appendix A of [1] it is shown that the Multiple Access Interference can be approximated as a Gaussian random variable with the variance given by Equation (4.17). This concept and Equation (4.17) will be used in our simulation.

In CDMA systems, if every user transmits at the same power level, a few users closest to the base station may contribute more Multiple Access Interference that they prevent other uplink signals from being successfully received at the base station. This is the *near-far* effect.

To solve this problem, *power control* is used in CDMA systems. Power control forces all users to transmit the minimum amount of power needed to achieve acceptable signal quality at the base station.

4.3 Multipath Fading Channel Model

Unlike wired channels whose parameters are stationary and predictable, wireless channels place fundamental limitations on the performance of a mobile communications system due to the dynamic time-varying nature of the transmission path and the relative

speed between the transmitter and the receiver. The transmitted signal interacts with various objects in the physical region before reaching the base station. These objects include buildings, hills, streets, trees, and moving vehicles. The collection of objects in any given physical region describes the propagation environment. The propagation of radio waves through this environment is a complicated process that involves diffraction, refraction, and multiple reflections. A good understanding of the wireless channel is important in designing effective techniques for signal reception and transmission.

The wireless propagation environment can be characterized by three key effects which impair the transmitted signal: *path loss, shadowing, and multipath fading*. Path loss is the loss in power of the radiated signal as it travels to the base station. It is a function of the distance between the mobile and base station. Shadowing refers to the attenuation in signal power caused by large obstructions, such as buildings and hills that are positioned between the mobile and base station. Shadowing may cause large variations in signal power as a function of location.

Multipath propagation plays a central role in determining the nature of the wireless channel. The transmitted signal is scattered by numerous physical objects in its path to the base station. Each path arrives from a different direction, with some attenuation and propagation delay. The base station receives a superposition of these paths. The multiple paths add up constructively or destructively, causing variations in the received signal amplitude. This phenomenon is known as multipath fading.

4.3.1 Signal Model of Antenna Array in Fading Channel

A discrete model is used to characterize the channel [19][1][38][39], where each multipath component is considered to be a plane wave, arriving from a discrete direction at a

discrete time delay. For a particular subscriber, the channel between the portable transmitter and the base station receiver is modeled using the Vector Channel Impulse Response,

$$\boldsymbol{h}(\varsigma,t) = \sum_{i=0}^{L-1} \mathbf{a}(\theta_i, \phi_i)\eta_i(t)\delta(\varsigma - \varsigma_i) \tag{4.18}$$

where η_i, ς_i and (θ_i, ϕ_i) are the complex amplitude, path delay, and Direction-Of-Arrival (DOA) of the i^{th} multipath component. There are a total of L multipah components. The complex amplitude of the i^{th} component is written as a function of time and may be expressed as

$$\eta_i(t) = \frac{1}{\sqrt{K_i}} \sum_{k=1}^{K_i} e^{j2\pi(f_i \cos\varphi t - f_c\varsigma_i)} \tag{4.19}$$

where K_i is the number of scattered components at the i^{th} path, f_i is the Doppler shift of i^{th} path due to the motion of either a mobile unit or scatterers in the environment, and φ_i is the direction of the k^{th} component of the i^{th} path. In general, all of the variables of the variables of the Vector Channel Impulse Response can vary with time, the position of the user, and the velocity of the user. As the user moves over a small local area, a distance of several wavelengths, we consider the number of multipath components, L, to be fixed, and for each component, the DOA (θ_i, ϕ_i), the Doppler shift (f_i), the phase offset (φ_i), and the delay (ς_i), are assumed to be approximately constant. For simplicity, we assume that multipath components arrive at the base station in the horizontal plane, $\theta = \pi/2$, so that the azimuthal direction, ϕ, completely specifies the DOA.

The vector channel impulse response relates the transmitted signal to the signal received at each antenna element of the array. Given a transmitted baseband complex

envelope signal, $s(t)$, the received data vector is (refer to Figure 2-1)

$$\boldsymbol{u}(t) = [u_0(t) \ldots u_{M-1}(t)]^T = s(t) * \boldsymbol{h}(\varsigma, t) + \boldsymbol{n}(t)$$

$$= \int_{-\infty}^{t} s(\lambda)h(t-\lambda, t)d\lambda + n(t)$$

$$= \sum_{i=0}^{L-1} \mathbf{a}(\phi_i)\eta_i(t)s(t - \varsigma_i) + n(t) \tag{4.20}$$

where $\boldsymbol{n}(t)$ is the vector representing the noise introduced at each antenna element.

4.3.2 Types of Fading Channel

Depending on the relation between the signal parameters (such as bandwidth, symbol period, etc.) and the channel parameters, different transmitted signals will undergo different types of fading. First les us briefly review some important channel parameters [9].

Time dispersion parameters: *mean excess delay* and *RMS (root-mean-square) delay spread* are multipath channel parameters. The *mean excess delay* is defined to be

$$\bar{\tau} = \frac{\sum_k a_k^2 \tau_k}{\sum_k a_k^2} = \frac{\sum_k P(\tau_k)\tau_k}{\sum_k P(\tau_k)} \tag{4.21}$$

where a_k is the amplitude of the kth path, τ_k is the delay of kth path, and $P(\tau_k)$ is the power of the kth path. The RMS delay spread is

$$\sigma_\tau = \sqrt{\overline{\tau^2} - (\bar{\tau})^2} \tag{4.22}$$

Reference [10][1] give us another measure of the difference in time delays of multipath components, the ensemble average *delay spread*, which is given by

$$\sigma_\tau = E[(\tau_k - \bar{\tau})^2] \tag{4.23}$$

It is assumed that the delays of the multipath components are independent, with mean $\bar{\tau}$.

Coherence Bandwidth: *Coherence Bandwidth* is the range of frequencies over which two frequency components have strong potential for amplitude correlation. If the coherence bandwidth is defined as the bandwidth over which the frequency correlation functions above 0.5, then the coherence bandwidth is approximately

$$Bc \approx \frac{1}{5\sigma_\tau} \tag{4.24}$$

Doppler Spread and Coherence Time: *Doppler spread* and *coherence time* are parameters which describe the time varying nature of the channel in a small-scale region. *Doppler spread* B_D is a measure of the spectral broadening caused by the time rate of change of the mobile radio channel and is defined as the range of frequencies over which the received Doppler spectrum is essentially non-zero. If the baseband signal bandwidth is much greater than B_D, the effects of *Doppler spread* are negligible at the receiver. *Coherence time* is the time domain dual of *Doppler spread* and is used to characterize the time varying nature of the frequency dispersiveness of the channel in the time domain. The *Doppler spread* and the *coherence time* are inversely proportional to one another. That is,

$$T_{CT} \approx \sqrt{\frac{9}{16\pi f_m^2}} = \frac{0.423}{f_m} \tag{4.25}$$

where f_m is the maximum Doppler shift given by $f_m = vf_c/c$ (v is the speed of mobile, f_c is the carrier frequency and c is the speed of light). The definition of coherence time implies that

two signals arriving with a time separation greater than T_{CT} is affected differently by the channel.

The types of Small-Scale Fading are summarized in Table 4-1 [9].

Table 4-1 Types of small-scale fading

Based on multipath time delay spread	
Flat Fading	**Frequency Selective Fading**
BW of signal << Coherence bandwidth $B_S << B_C$	BW of signal > Coherence bandwidth $B_S > B_C$
Delay spread << Symbol period $\sigma_\tau << T_S$	Delay spread > Symbol period $\sigma_\tau > T_S$
Based on Doppler spread	
Fast Fading	**Slow Fading**
High Doppler spread	Low Doppler spread
Coherence time < Symbol period $T_{CT} < T_S$ and $B_S < B_D$	Coherence time >> Symbol period $T_{CT} >> T_S$ and $B_S >> B_D$
Channel variations faster than baseband signal variations	Channel variations slower than baseband signal variations

For a CDMA signal, we say that two multipath components are uncorrelated if they are resolvable by more than a chip duration [1], or $|\tau_1 - \tau_2| > T_c$, where T_c is the chip duration. From (4.23) we can express the expected value of the squared difference in delays between two multipath components as $E[(\tau_1 - \tau_0)^2] = 2\sigma_\tau^2$. Therefore, two multipath components are expected to be uncorrelated if $2\sigma_\tau^2 > T_c^2$.

A CDMA signal is in *Flat Fading* Channels if the delay spread is small so that $\sigma_\tau < T_c/4\pi$. In flat fading channels, phase coherence among the correlated multipath components allows them to interact either constructively or destructively, leading to potentially deep fades. If the delay spread is large relative to the chip period, $\sigma_\tau > T_c/4\pi$, *Frequency Selective Fading* may occur in which fading in one region of the signal spectrum does not imply fading in other regions of the signal spectrum and multipath components tend to be uncorrelated.

In this thesis, both the cases of Flat Fading and Frequency Selective Fading are simulated. In mobile radio channels, the Rayleigh distribution is commonly used to describe the statistical time varying nature of the received envelope of a Flat Fading signal, or the envelope of an individual multipath component. Therefore, the Rayleigh random variable will be used to simulate the flat fading channel, and a complicated multipath model with Angle Spread will be used for the simulation of the Frequency Selective Fading. More details about simulations will be introduced in Chapter 5.

4.4 IS-95 CDMA Reverse link

Interim Standard 95 (IS-95) specifies a wireless telephony system that uses direct-sequence spread-spectrum (DS/SS) as a multiple access technique. It was introduced by Qualcomm Corporation, and it was designed to operate in the same frequency band as the U.S. analog cellular system (AMPS), in which full duplex operation is achieved by using frequency division duplexing. The basic waveform, coding, and interference suppression features of such systems are outlined as follows [5]:

- Each channel is spread across a bandwidth of about 1.25MHz and filtered for spectral containment.

- The chip rate Rc of the PN code is 1.2288 Mchips/s. The nominal data rate, known as Rate Set 1 (RS1), is 9.6 kbits/s, making the processing gain Gp = Rc/R = 128. An extension to the original IS-95 Introduced Rate Set 2 (RS2) at 14.4 kbits/s.
- The data modulation is binary phase-shift keying (BPSK), with quadrature phase-shift keying (QPSK) spreading. (Each quadrature component of the carrier wave is a BPSK signal modulated with the same data.)
- Convolutional coding with Viterbi decoding is used.
- Interleavers with a 20-ms time span are used for time diversity.
- Path diversity is exploited with a Rake receiver, and spatial diversity is implemented with two receive antennas per cell sector.
- Orthogonal code multiplexing is used for channelization.
- Power control is used to minimize transmitted power and thereby reduce interference.

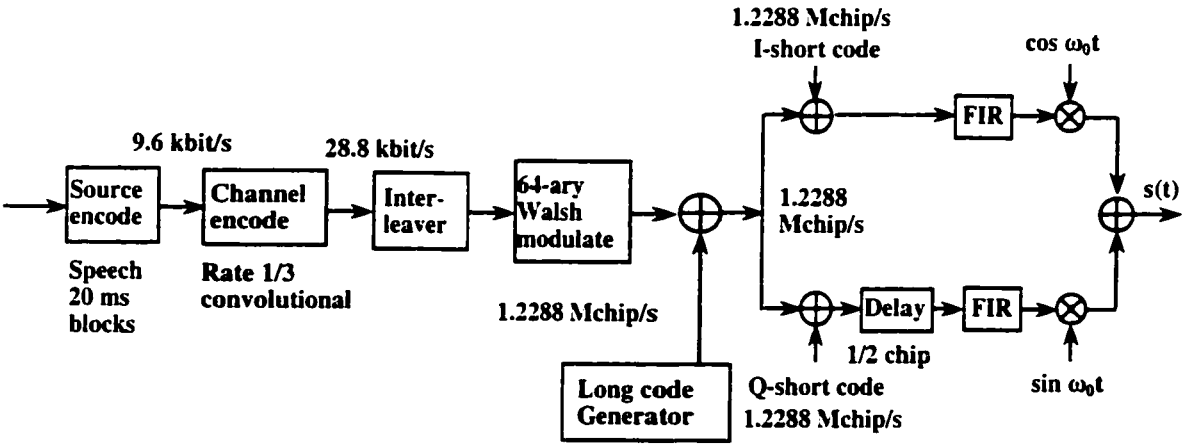


Figure 4-2 Simplified CDMA reverse-traffic channel with full-rate speech

The forward link comprises four types of channels: pilot, synchronization, paging, and traffic. The reverse link comprises two types of channels: access and traffic.

In our simulation, we assume that only the base station is equipped with an antenna array. Therefore only the reverse link (mobile to base station) of IS-95 standard is discussed.

Figure 4-2 depicts a simplified block diagram of IS-95 reverse-traffic channel transmission.

In IS-95, the reverse link does not support a pilot channel, since one would be required for each mobile unit. Thus, the reverse-channel signal is demodulated noncoherently at the base station. Since the reverse channel is less robust than the forward channel, a more powerful rate 1/3 convolutional code is used to improve performance. Following the interleaver, each group of six modulation symbols is replaced with a single 64-chip Walsh Function. Walsh Functions have the property that each function is exactly orthogonal to all of the others. In 64-ary orthogonal modulation, each set of six bits, at a rate of 28.8 kbps, into the Walsh Code Modulator block, produces 64 Walsh chips at a rate of 307.2 kbps. The Walsh chips are then direct-sequence spread with the mobile's long code sequence at 1.2288Mcps.

The long code is used for the purpose of: 1, Channelization – The base station separates reverse channel traffic by using the long code. 2, Privacy, scrambling, and spreading. The long code is a particular type of Pseudo Noise sequence called a *Maximal Length Sequence* (M-sequence). An M-sequence can be generated by a Linear Feedback Shift Register. The sequence produced by this generator polynomial has a length of 2^{42}-1 chips.

The final stage in generating the baseband reverse link signal is the *Quadrature Spreading* operation. After spreading by the long code, the signal is split into In-phase and Quadrature components where each is further modulated with the short codes at chip rate

1.2288Mcps and then modulated with a carrier frequency. The M-sequences generated by these polynomials have length 2^{15}-1. An extra zero is inserted at the end of the short code sequence to make the total length 2^{15} = 32768 chips.

4.5 CDMA Spatial Processors

Spatial filtering at the CDMA base station receiver can take many forms, as shown in [1]. Spatial processing systems for the base station may be classified as either single-user or multi-user systems. In multi-user systems, beams are formed for all users simultaneously.

4.5.1 Pre-correlation Beamformer Structure

A coherent-combining array receiver for a CDMA system is shown in Figure 4-3 [1]. The metrics used for weight update are possibilities include examination of the constellation at the input to the *Walsh Chip Matched Filter*, examination of the Walsh Chip Matched Filter output, or others.

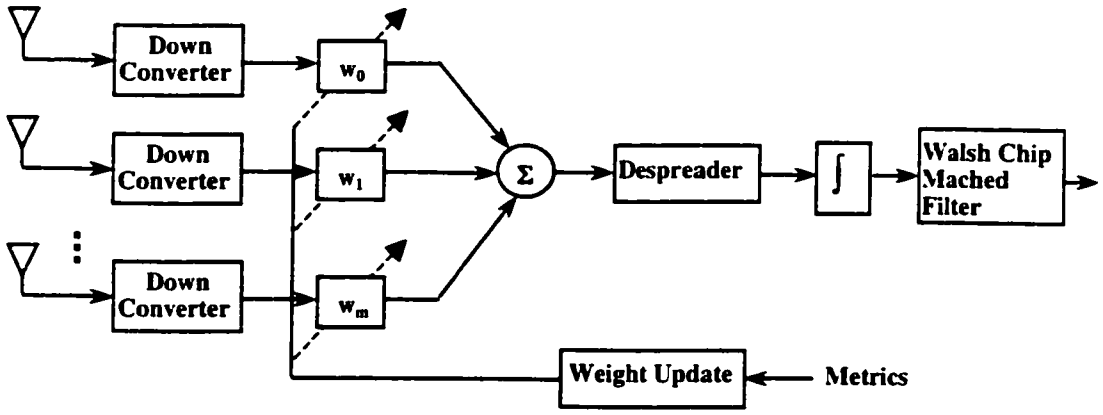


Figure 4-3 A coherent-combining receiver for a CDMA system

In this structure, the beamforming weight update is performed before despreading, i.e., in chip signal. The advantage of doing so is it requires only one despreading module for

each spatial filtering receiver. But it also brings out extra computational load in the algorithms, which can be seen below.

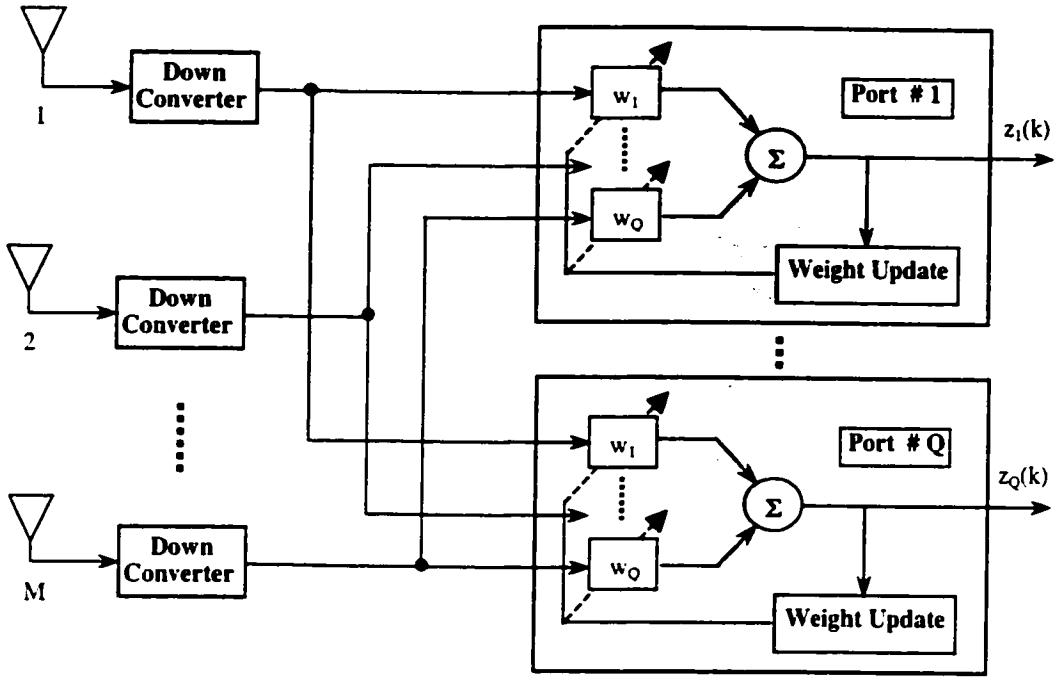


Figure 4-4 A multitarget adaptive beamformer

In a CDMA mobile communication system, multiple users occupy the same frequency band. The beamformer in the base station attempts to form a beam directed to each user, so that for one desired user, the interference from other directions is reduced. For a system with p users, the beamformer will generate p complex weight vectors. Figure 4-4 shows a structure of a multitarget adaptive beamformer with M antenna elements and Q output ports. In this figure, $z_1(k)$, ..., $z_Q(k)$ are the outputs of ports 1,...,Q, respectively, and w_1,...,w_Q are the weight vectors for port 1,...,Q. For some algorithms, the number of output ports, Q, must

be less than or equal to the number of antenna elements, M, but for other algorithms, it can be greater than the number of antenna elements and equal to the number of users [1][4].

If the structure depicted in Figure 4-3 is used in each port of Figure 4-4, some extra procedure has to be performed in order to extract the desired signal correctly.

First, some important blind algorithms (such as CMA based) utilizes only the a priori information that the original signals have a constant envelope, for a CDMA system with all the transmitted signals having the constant modulus property before despreading, if steps are not taken, all the weight vectors in different ports may converge to the same beam pattern. To avoid this, *Gram-Schmit orthogonalization* (GSO) have to be performed. If two weight vectors intend to converge to one with the same beam pattern, the absolute value of their correlation coefficient will become larger. The correlation coefficient between two weight vectors w_i and w_j is defined by

$$\rho_{ij} \equiv \frac{w_i^H w_j}{\|w_i\| \|w_j\|} \tag{4.26}$$

By comparing this correlation coefficient with certain threshold and adjusting the weight, we can prevent different weight vectors from converging to the same spatial response.

Second, after the algorithm converges, a sorting procedure must be performed to relate the port outputs to each user's signal. In a CDMA system, the PN sequence assigned to each user can be utilized in the sorting procedure.

The algorithms can be used in above structure include *Multitarget least Squares Constant Modulus Algorithm* (MT-LSCMA) and *Multitarget Decision-Directed Algorithm*

(MT-DD). Other two algorithms called *Least Squares De-spread Re-spread Multitarget Array* (LS-DRMTA) and *least Squares De-spread Re-spread Multitarget Constant Modulus Algorithm* (LS-DRMTCMA), which are developed by Rong and Rappaport, utilize the despreading signal and do not need to perform the GSO procedure and sorting procedure. But matrix inversion still needs to be calculated in the algorithms. All the algorithms mentioned above are described in detail in [1] and [4].

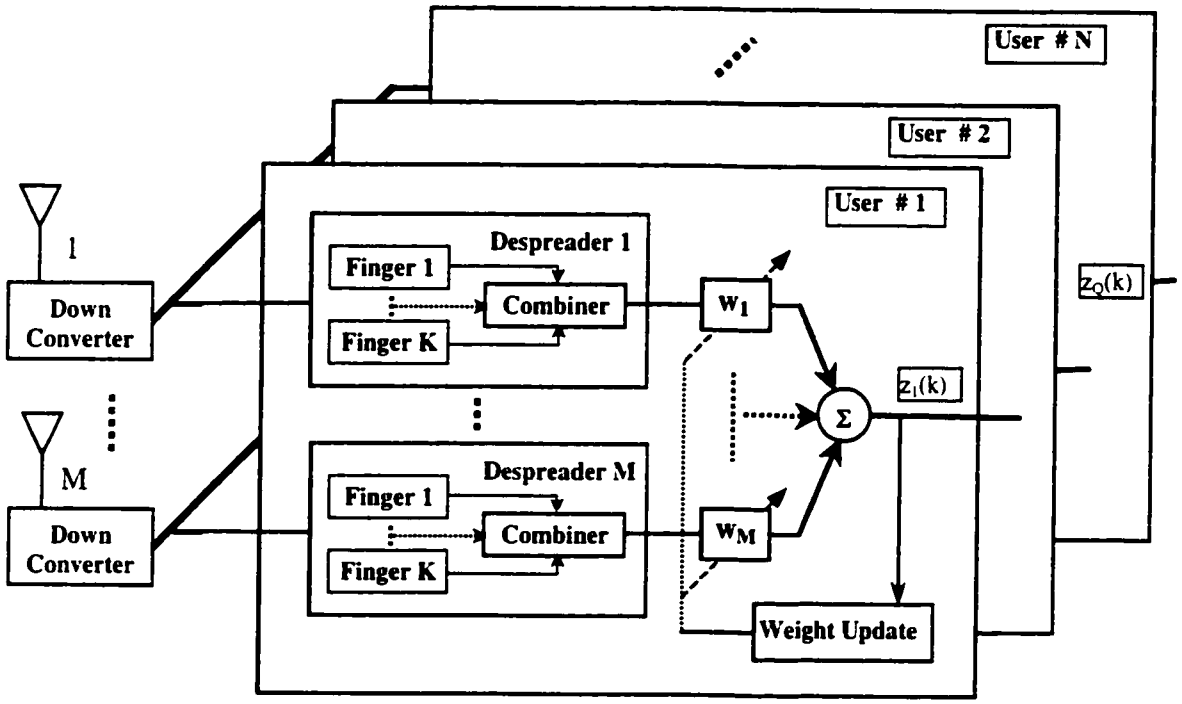


Figure 4-5 A multitarget adaptive beamformer performed after Despreading

4.5.2 Post-correlation Beamformer Structure

Another spatial filtering structure that can be used in CDMA base station is shown in Figure 4-5 [33][38].

The structure shown in Figure 4-5 differs from previous structure (Figure 4-3) not only in hardware but also in the algorithms.

In this model, because the weight update is performed after chip correlation (despreading), the desired signal becomes stronger than each interferer by a factor of the processing gain with the aid of a good performance in power control and chip code synchronization [33].

Because of the feature of the post-correlation signal, the BEE and CMA algorithms that we mentioned in section 3.3.2 can be used in this model. These two blind algorithms have the significant advantage of simplicity. The computational load for computing a weight is only in the order of M, where M is the number of antenna elements. Therefore they are easily implemented in real time with current technique.

It is important to observe that in the post-correlation signal, the power difference between the desired signal and the interferers is more than 10dB (depends on the processing gain). Therefore the algorithms performed in this model does not null out the interferers because they cannot be clearly detected. In other words, the beampattern simply maximizes the gain along the direction of the desired signal source. Most importantly, the algorithms can work normally whether or not the number of antenna elements is greater than that of interferers. As we presented in section 2.3.2, the capability for the antenna array to null the interfering signals is limited by the number of antenna elements. But note that the number of interferes in a normal CDMA signal environment is at least in the order of tens and it is never realistic to have that many antenna elements in the cell site of a given mobile communication system. It means that the conventional arrays based on the nulling procedure are unlikely to be applicable in practical CDMA systems [33].

In this thesis, some algorithms performed for the model shown in Figure 4-5 will be simulated.

In above model a 2D-RAKE Receiver structure is exploited to improve the receiver performance in frequency-selective fading channel. For the CDMA IS-95 standard, a one-chip path differential delay corresponds to a differential path distance of about 250 meters. Therefore, signals arriving at the base station with a differential delay of more than one chip will very likely be coming from different directions. This effect is especially pronounced in heavily built-up urban area. Therefore, angle-of-arrival diversity techniques can also be used for the optimal RAKE receiver in an IS-95 system.

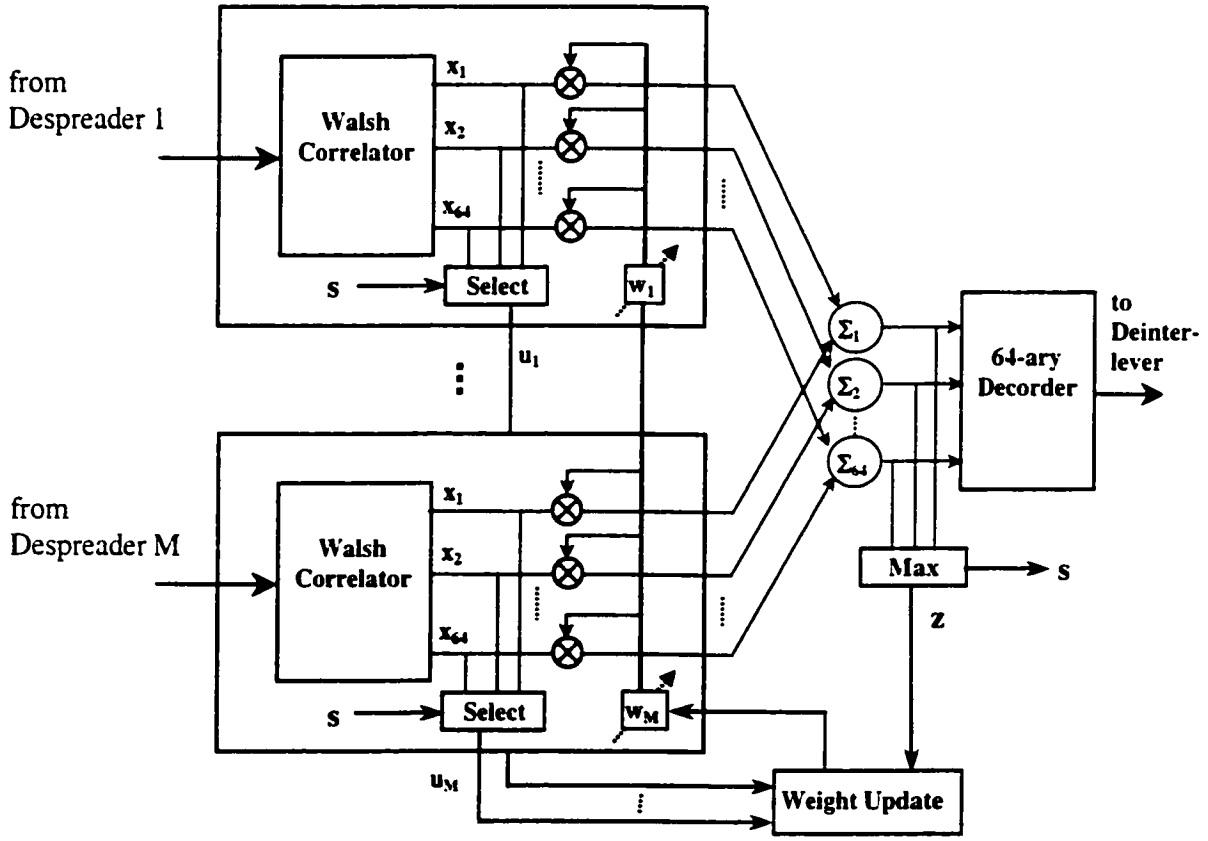


Figure 4-6 Application of Adaptive Array in the IS95 Reverse link

The 2D RAKE can exploit the temporal and spatial structure of the individual paths arriving at the receiver and finally coherently combining the paths in order to further improve the SINR. Note that it is also possible to reverse the order of the RAKE Fingers and combiner (not the despreader) and the beamforming [22][38], relative to Figure 4-5, i.e. by first beamforming each of the paths from the array, then delaying each by the estimated channel delay and combing them.

Figure 4-6 shows an application of the above model in IS-95 CDMA reverse link [33][38]. As we mentioned in the previous section, in 64-ary orthogonal modulation, each set of six bits, at a rate of 28.8 kbps, into the Walsh Code Modulator block, produces 64 Walsh chips at a rate of 307.2 kbps at the transmitter. At the receiver, after despreading the signal at a rate of 307.2 kbps is feed to the 64-ary Walsh Correlator, after which 64 complex post-correlation signals $x_1,....,x_{64}$ are produced. The value that cause a maximum output after multiply the weight, which is most likely from the desired signal, is selected with the help of the beamformer output signal "s" from the 64 post-correlation signal. Signal "s" is actually the index of maximum output. The selected value combined with the output of the beamformer is used for the weight updating. The output values of the beamformer are finally employed by the 64-ary decoder for the recovering of the set of six bits to the rate of 28.8 kbps.

Chapter 5

Simulation Results and Discussion

5.1 Introduction

In this chapter, we will simulate and compare the performance of several algorithms under different conditions of CDMA wireless communication system. The algorithms will be simulated include both non-blind and blind algorithms that we presented in Chapter 3. The CDMA receiver structure used in our simulation is shown in Figure 4-5. As we explained in Section 4.5.2, this structure allows us to perform some very simple blind algorithms that real time operation is possible in current technique, although it increase some hardware cost. But note that the complicated algorithms are not only difficult to be performed in real time but also significantly increase the hardware cost if they are implemented in hardware – such as in VLSI. The intention is not to null out the interferers, rather maximizing the gain along the direction of the desired signal source. As we mentioned in Section 4.5, it is not possible to null out all the interferers in a CDMA system because the amount of interferers is always much larger than the number of antenna elements.

In the simulation, the following cases will be considered:

1. AWGN channel.
2. Flat Fading channel.
3. Frequency Selective Fading channel with *angle spread* (Multipath case).
4. Phase Offset and Mobile Tracking.

The comparison of the algorithms will focus on the BER performance of the algorithms operated in different situations.

5.2 Simulation Program and General Issues of the Simulated Algorithms

In this section, we will briefly introduce our simulation program and some issues of the algorithms, i.e. the convergence rate, the phase offset and the computational complexity of the algorithms.

5.2.1 Introduction to Simulation Program and Parameters Setting

Our simulation program is developed in C++ programming language. The output of the program is the MATLAB M-file that can be directly run by MATLAB and plots the figures. All of the simulation figures shown in this chapter are produced by this program.

The system we consider in the simulation program is direct sequence CDMA (DS-CDMA) and the modulation scheme is binary phase-shift keying (BPSK). The program is designed to perform following functions:

- Array Type
  - Conventional Receiver
  - Uniformly Spaced Linear Array (ULA)
  - Uniformly Spaced Circular Array (UCA)
  - Two Dimension Planar Array
- Channel Type
  - AWGN Channel
  - Rayleigh Flat Fading Channel
  - Frequency Selective Channel

- Receiver Type
  - Without RAKE Receiver
  - With RAKE Receiver
- Algorithm Type
  - LMS Algorithm
  - NLMS Algorithm
  - RLS Algorithm
  - CMA Algorithm
  - NCMA Algorithm
  - BEE Algorithm
- Mobile Tracking
- Phase Offset
- Output Type
  - BER Performance versus E_b/N_o
  - BER Performance versus Number of Users
  - Algorithms Convergence Curve
  - Phase Offset Response Curve
  - Tracking Curve
  - Beam pattern in Rectangular Coordinate and Polar Coordinate
  - Signal waveform Recovering Process

The parameters used in the program can be easily set in a configuration file. The parameters include: Processing Gain, Number of Antenna Elements, Carrier Frequency,

Element Spacing, Number of Interferes, Direction of Desired User and Multipaths, Power Ratio between the Multipaths, Delay of Chips between the Multipaths, Vehicle Speed, Sampling Rate of the Signal, Frame Size, Weight Updating Block Size, Number of Training Sequence, Signal-to-Noise-Ratio, Number of Interferers, Algorithms Factors, Phase Offset, Tracking Step and several program control parameters.

The combination of above cases can produce many outputs representing different situation. In this chapter we only give some typical results and analyses. To avoid any misunderstanding, we will fix the Processing Gain at 31 and Carrier Frequency at 1.9 GHz in every cases of the simulation (although these parameters can be easily changed in the configuration file). And we use M-sequences for the simulation of CDMA chip signals. All of the BER performances in the following simulations are obtained by sending and comparing a very large number of bits.

5.2.2 Summary of the simulated algorithms

The algorithms used in the simulation program are summarized as following:

1. Non blind Algorithms:

a. Least Mean Square (LMS)

Initialization: If prior knowledge on the weight vector $\hat{w}_n$ is available, use it to select an appropriate value for $\hat{w}_0$. Otherwise, set $\hat{w}_0 = 0$.

Weight Update:

$$z_n = \hat{\mathbf{w}}^H{}_n \cdot \mathbf{u}_n$$

$$e_n = d_n - z_n$$

$$\hat{\mathbf{w}}_{n+1} = \hat{\mathbf{w}}_n + \mu \cdot \mathbf{u}_n \cdot e^*_n$$

Where μ is step-size parameter.

b. Normalized Least Mean Square (NLMS)

Initialization: If prior knowledge on the weight vector $\hat{w}_n$ is available, use it to select an appropriate value for $\hat{w}_0$. Otherwise, set $\hat{w}_0 = 0$.

Weight Update:

$$z_n = \hat{\mathbf{w}}^H{}_n \cdot \mathbf{u}_n$$

$$e_n = d_n - z_n$$

$$\hat{\mathbf{w}}_{n+1} = \hat{\mathbf{w}}_n + \frac{\tilde{\mu}}{a + \|\mathbf{u}_n\|^2} \cdot \mathbf{u}_n \cdot e^*_n$$

Where $\tilde{\mu}$ is adaptation constant, $0 < \tilde{\mu} < 2$.

c. Recursive Least Square (RLS)

Initialize the algorithm by setting $\mathbf{P}_0 = \delta^{-1}\mathbf{I}$, δ is small positive constant, $\hat{w}_0 = 0$.

Weight Update:

$$k_n = \frac{\lambda^{-1}\mathbf{P}_{n-1}\mathbf{u}_n}{1 + \lambda^{-1}\mathbf{u}^H{}_n\mathbf{P}_n\mathbf{u}_n}$$

$$\xi_n = d_n - \hat{\mathbf{w}}^H{}_n\mathbf{u}_n$$

$$\hat{\mathbf{w}}_n = \hat{\mathbf{w}}_{n-1} + k_n\xi^*_n$$

$$\mathbf{P}_n = \lambda^{-1}\mathbf{P}_{n-1} - \lambda^{-1}k_n\mathbf{u}^H{}_n\mathbf{P}_{n-1}$$

Where λ is forgetting factor, $0 < \lambda < 1$.

2. Blind Algorithms:

a. Constant Modulus Algorithm (CMA)

Initialization: If prior knowledge on the weight vector $\hat{w}_n$ is available, use it to select an appropriate value for $\hat{w}_0$. Otherwise, set $\hat{w}_0 = 1$.

Weight Update:

$$z_n = \hat{\mathbf{w}}^H{}_n \cdot \mathbf{u}_n$$

$$e_n = 2 \cdot \left(z_n - \frac{z_n}{|z_n|} \right)$$

$$\hat{\mathbf{w}}_{n+1} = \hat{\mathbf{w}}_n + \mu \cdot \mathbf{u}_n \cdot e^*{}_n$$

Where μ is step-size parameter.

b. Normalized Constant Modulus Algorithm (NCMA)

The step-size parameter μ is chosen similar to NLMS.

c. Blind Eigenvector Estimation Algorithm (BEE)

Initialization: If prior knowledge on the weight vector $\hat{w}_n$ is available, use it to select an appropriate value for $\hat{w}_0$. Otherwise, set $\hat{w}_0 = 1$.

Weight Update:

$$z_n = \hat{\mathbf{w}}^H{}_n \cdot \mathbf{u}_n$$

$$a = \mu, \quad b = \mu \|z\|^2 + 1, \quad c = \|y\|^2 \left(2 + \mu \|u\|^2 \right)$$

$$\gamma = \left[b - \left(b^2 - a \cdot c\right)^{1/2}\right] / a$$

$$\hat{\mathbf{w}}_{n+1} = \left[1 - \mu\gamma\right] \cdot \hat{\mathbf{w}}_n + \mu \cdot z^*_n \cdot \mathbf{u}_n$$

In the non-blind algorithms (LMS, NLMS and RLS), the desired signal must be supplied using either a training sequence or decision direction. In the training sequence approach, a brief data sequence is transmitted which is known by the receiver. The receiver uses an adaptive algorithm to estimate the weight vector during the training period, then holds the weights constant while information is being transmitted. This technique requires that the environment be stationary from one training period to the next, and it reduces channel throughput by requiring the use of channel symbols for training. Especially in CDMA IS-95 reverse link, there is no pilot signal transmitted, hence it is not possible to use the algorithms with training sequence.

The blind algorithms (CMA, BEE) do not require training sequences. Therefore they are preferred to be used in wireless communication systems. Here the simulation of the non-blind algorithms is only for the comparison purpose.

5.2.3 Beam pattern of the Algorithms

In Chapter 1, we defined the *beam pattern* as the magnitude of the *array response* $A(\Phi)$, $G(\Phi) = |A(\Phi)|$, where

$$A(\Phi) = \mathbf{w}^* \mathbf{a}(\phi) = \sum_{m=1}^{M} w_m^* e^{-j\omega_c \tau_m} = \sum_{m=1}^{M} r_m e^{-j\omega_c (\tau_m - \hat{\tau}_m)}$$

here we let $w_m = e^{j\omega_c \hat{\tau}_m}$, in which $\hat{\tau}_m$ is estimated by Equations (2.16) and (2.18) for ULA and UCA, respectively. But in these two equations, the direction of arrival ϕ_0 must be

known. As we mentioned in Section 3.3.4, because there are some disadvantages with the algorithms based on Estimation of Direction of Arrivals, the algorithms that only utilize the array input signal are preferred.

Let us first examine the beam pattern produced by the algorithms which are summarized in Section 5.2, then investigate the convergence behaviour and complexity of these algorithms.

Figure 5-1 shows the Beam pattern of 8-element ULA, with d = 0.5λ, E_b/N_o = 4 dB, produced by LMS, RLS, CMA and BEE algorithms, respectively. The direction of the desired signal is from 70°. Comparing Figure 5-1 with Figure 2-4, we see that there are no obviously differences between the beam patterns produced by the algorithms and by the given weight ($w_m = e^{j\omega_c \tau_m}$). The main lobe width and side lobes are in the same level. But note that in these figures, the algorithms adjust the maximum gain to the direction of the desired signal even though the direction is not known. Actually through further simulations we find that the beam patterns with the effect of number of antenna elements and the effect of antenna element spacing made by the algorithms are similar to what we investigated in Section 2.3.2 and 2.3.3, as shown in the sample Figure 5-2 and 5-3, and the cases in UCA are also similar.

Observing Figure 5-1 we can find that for the beam patterns generated by different algorithms, the main lobe width and side lobes are also in the same level (after algorithms converging), apart from some difference between the shapes. This is true even between the blind (CMA, BEE) and non-blind (LMS, RLS) algorithms. Therefore we can infer that in this CDMA receiver structure, the capability of different algorithms to extract the desired signal from certain direction is similar after the algorithms converging. But the features and

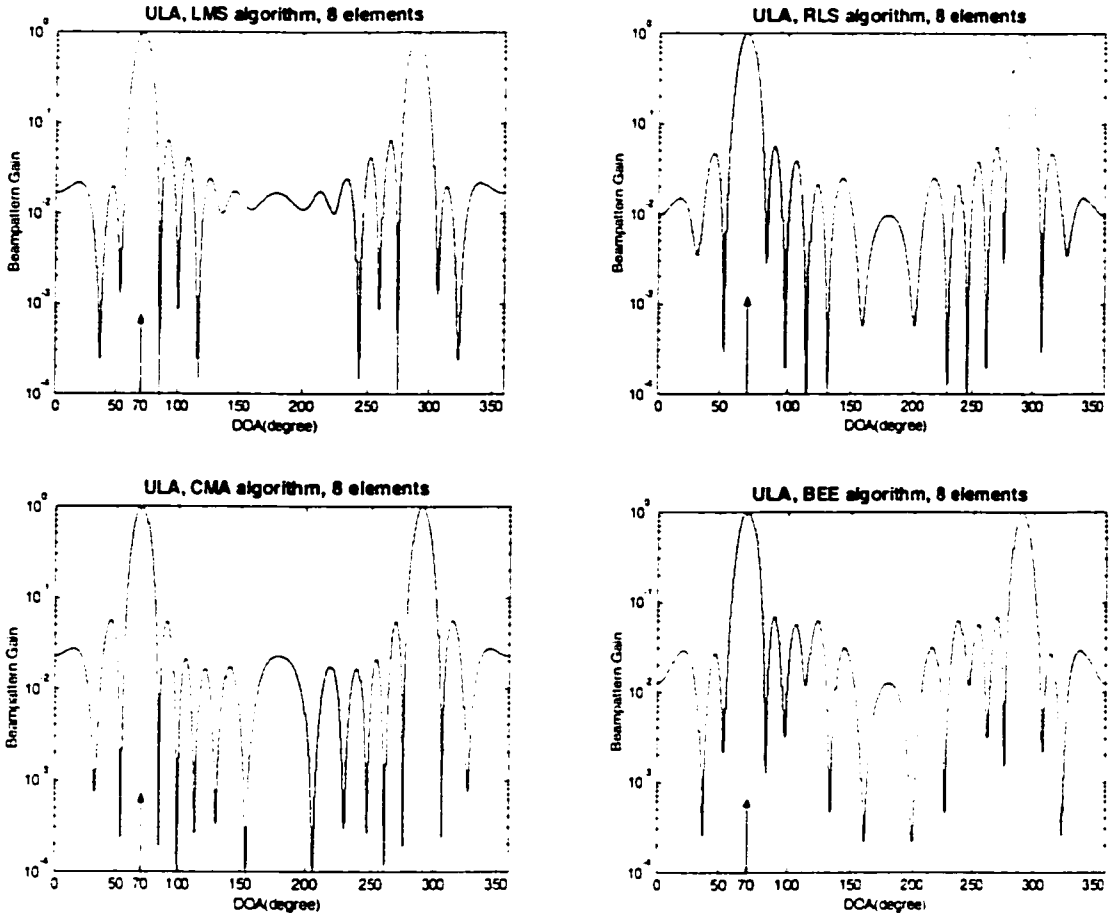


Figure 5-1: Beam pattern of 8-element ULA, with d = 0.5λ, produced by LMS, RLS, CMA and BEE algorithms, respectively. Eb/No = 4 dB. Steering at 70°.

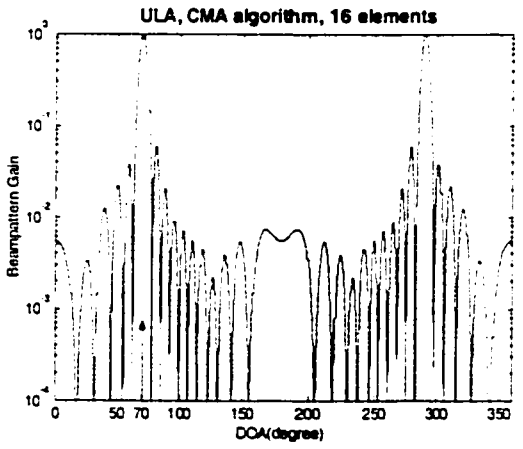


Figure 5-2: 16-element ULA, with d = 0.5λ, CMA agorithms, Eb/No = 4 dB.

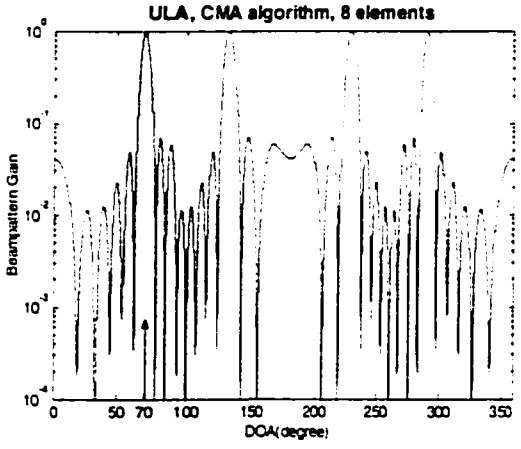


Figure 5-3: 8-element ULA, with d = λ, CMA agorithms, Eb/No = 4 dB.

convergence speed of the algorithms are different; certain algorithm certainly has more advantages over the others in the practical utilization.

5.2.4 Convergence Rate

Now let us compare the convergence behaviour of the algorithms that we summarized in Section 5.2. The convergence rate is actually closely connected to the parameters of the algorithms, such as the step-size parameter μ in LMS and CMA algorithms. The parameters are selected in such a way that they are as large as possible to get a faster convergence rate but must be smaller enough to keep the algorithms converge even in a worse case. In our simulation, the least signal-to-noise-ratio is considered as –6dB. Therefore, the parameters in our simulation are selected as the largest value to keep the algorithm work normally at E_b/N_o = -6dB in AWGN channel.

Several methods can be used to determine the convergence rate of an algorithm. Here we use the BER (Bit Error Rate) performance of the algorithm to determine the convergence rate. Before an algorithm converging, a high BER is expected because the maximum gain has not adjusted to the direction of desired user, and we think the algorithm has converged when the change of BER become stable after some iterations.

Figure 5-4 shows a comparison of convergence rate of different algorithms in UCA, with 4 antenna elements, d = 0.5λ, E_b/N_0 = 0 dB. From the figure we see that with the parameters we selected according to above criteria, the RLS algorithm has a fastest convergence rate. LMS algorithm converges relatively slowly. The blind algorithm CMA also has a good performance in convergence, and NCMA present a superior rate of convergence over the CMA. These curves are generated with μ = 0.01 for CMA and LMS,

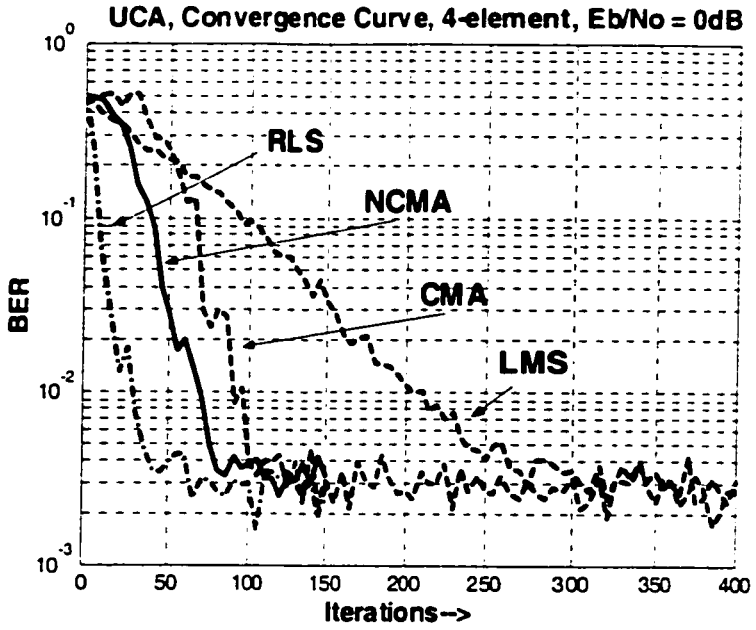


Figure 5-4: Comparison of Convergence rate of different algorithms in UCA, 4-element, with d = 0.5λ, E_b/N_0 = 0 dB.

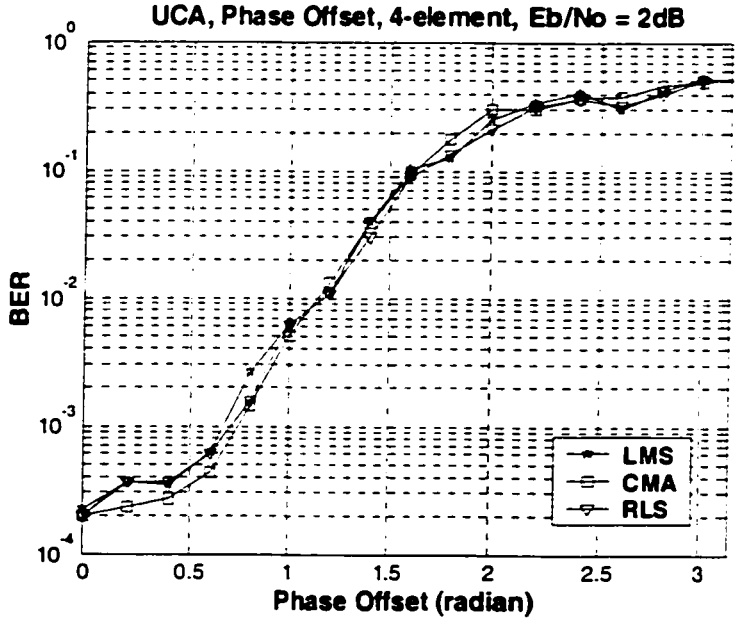


Figure 5-5: Effect of total Phase Offset, UCA, 4-element, with d = 0.5λ, E_b/N_0 = 2 dB.

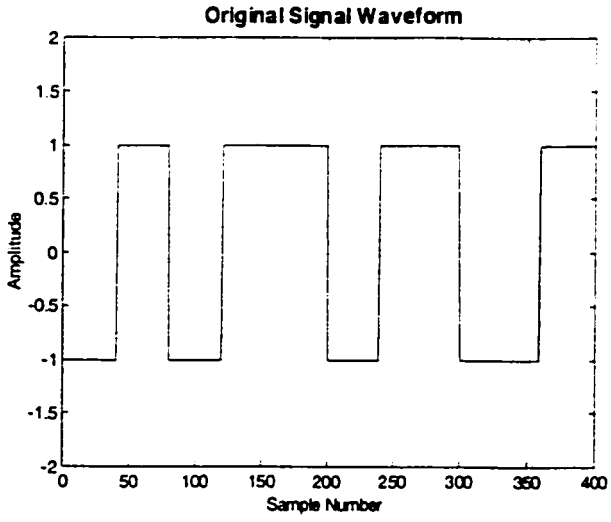


Figure 5-6: Original Signal Waveform at Transmitter

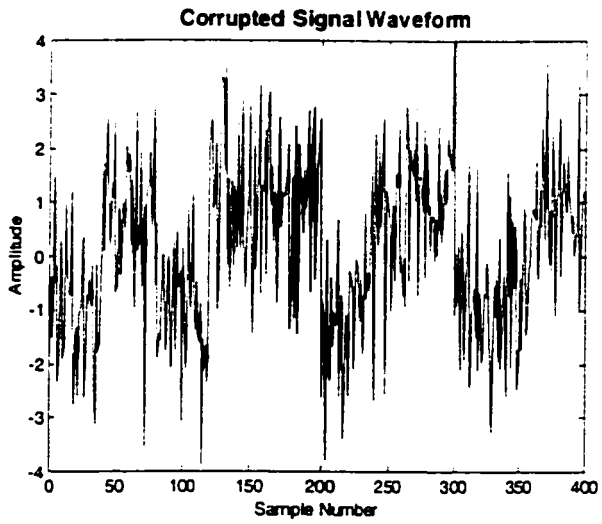


Figure 5-7: Corrupted Signal Waveform at Array Input, E_b/N_o = -4dB.

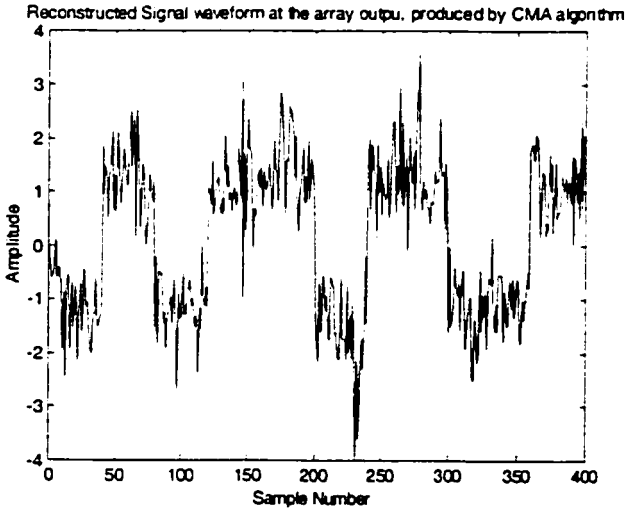


Figure 5-8: Reconstructed Signal Waveform at Array Output, produced by CMA algorithm

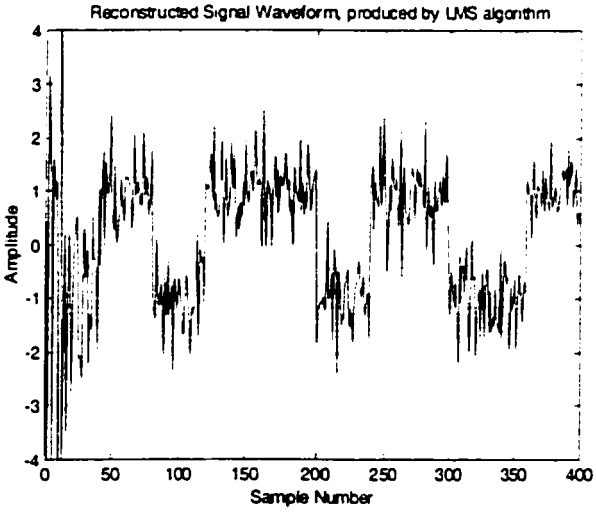


Figure 5-9: Reconstructed Signal Waveform at Array Output, produced by LMS algorithm

$\tilde{\mu}$ = 0.3, a = 2 for NCMA. The number of iterations for an algorithm to converge shown in Figure 5-4 is: LMS > 300, CMA > 100, NCMA > 80, RLS > 50

The convergence rate of ULA is not shown here since it is similar with UCA.

5.2.5 Phase Offset

In Figure 5.5, the effect of total phase offset is illustrated. In this case, we simulate the effect of phase offset by shifting the input signals in radian. From the figure we see that the ability to compensate for the phase error is nearly equal for all algorithms.

5.2.6 Signal Reconstruction

Figure 5-6 to 5-9 show a signal reconstructing process by an antenna array in AWGN channel, where E_b/N_o = -4dB and a 8 elements antenna is employed. In order to illustrate the process clearly, we don't send the data randomly. Instead, we send an "artificial" waveform in which the first 40 data bits are all "-1", next 40 data bits are all "1", and so on, as shown in Figure 5-6. Figure 5-7 shows a corrupted signal waveform at the array input due to the Additive White Gaussian Noise. We see that the corrupted signal is difficult to be detected. Figure 5-8 and 5-9 show the reconstructed signal waveform at the array output generated by CMA algorithm and LMS algorithm, respectively. We see that at the array output, the signal is reconstructed and can be detected by hard decision. Note that the signal fluctuates at the beginning of the samples shown in Figure 5-8 and 5-9, especially in the case of LMS algorithm. This is because the array is likely to produce some random output before the converging, and the slow convergence rate of LMS algorithm causes a longer adaptation time.

5.2.7 Computational Complexity of the Algorithms

The computational complexity of the algorithms is a very important issue for the real

time application. In wireless communication, the mobile user often moves with a high speed, therefore the weight update of the algorithm should be performed fast enough to follow the change of the environment and the direction of the user. The number of multiplications and additions required per iteration of an algorithm is summarized below (M is the number of elements used in the adaptive array):

LMS algorithm 2M complex multiplications, 2M+1 complex additions

RLS algorithm $3M^2 + 4M + 2$ complex multiplications, $4M^2$-1 complex additions.

CMA algorithm 3M complex multiplications, 3M -1 complex additions.

BEE algorithm 2M+3 complex multiplications, 2M -1 complex additions, and several real number multiplications and additions.

From above we see that the RLS algorithm is the most complicated of these algorithms. Other three algorithms are in the same computational complexity.

5.3 BER performance in AWGN channel

In this section, we will compare the BER performance of different algorithms under different conditions. We will compare the BER performance versus E_b/N_o and versus number of interferences of different algorithms with different number of antenna elements. We will also consider the BER performance in the timing offset case, which is due to the imperfect synchronization of the PN sequence in the receiver.

5.3.1 BER Performance versus E_b/N_o

Figure 5-10 and 5-11 show the BER performance versus E_b/N_o of different algorithms with different number of antenna elements in the case of ULA and UCA, respectively. From this figure we can get following conclusions:

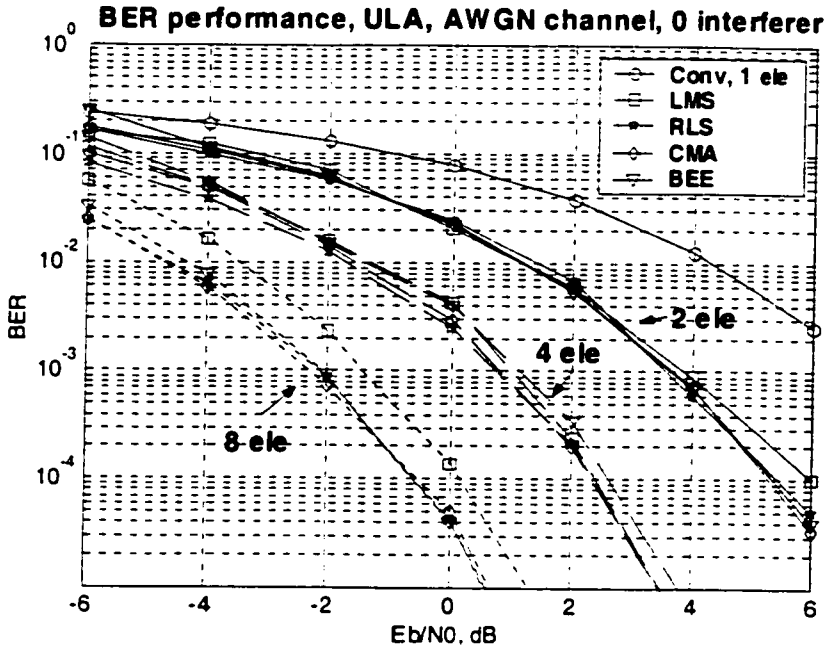


Figure 5-10: BER performance in AWGN channel, ULA, 2,4 and 8 antenna elements, with d = 0.5λ, no interference signal.

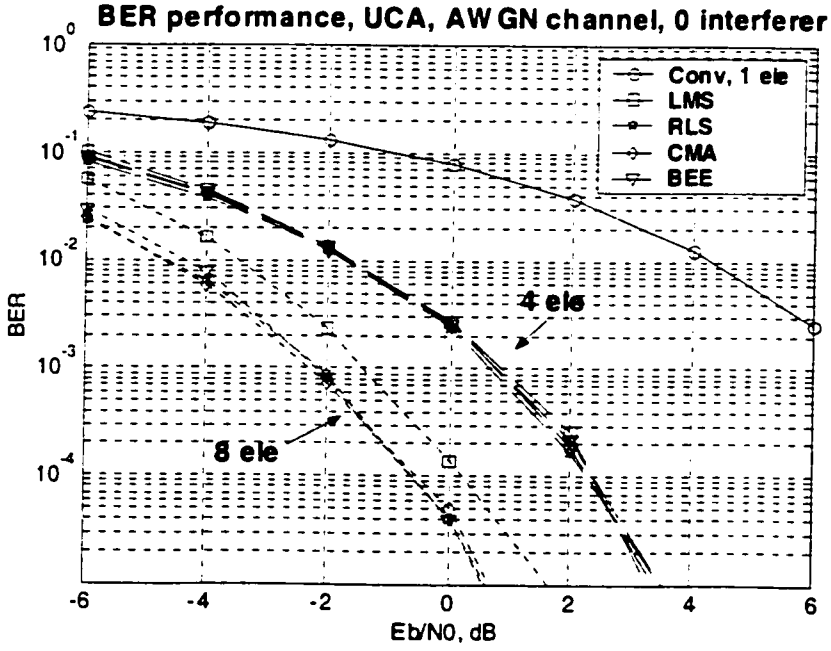


Figure 5-11: BER performance in AWGN channel, UCA, 4 and 8 antenna elements, with d = 0.5λ, no interference signal.

a). The more the number of antenna elements, the better the BER performance. This is reasonable because the width of the main-beam of the beampattern decreases as the antenna elements increases, but the side lobe level remains almost unchanged as the number of antenna elements changes, as shown in Figure 2-6 and Figure 5-2. We also can prove this theoretically as follows [1].

As described in Chapter 1, if we consider the noise contribution at each antenna element, the data vector is equal to

$$\mathbf{u}(t) = s(t)\mathbf{a}(\phi) + \mathbf{n}(t) \tag{5.1}$$

the vector $\mathbf{n}(t)$ contains the noise contributed at each antenna element. Each element of $\mathbf{n}(t)$ is a complex Gaussian random variable with variance σ_n^2.

We can use Equation (3.9) to find an optimal solution for the weight vector to extract $s(t)$,

$$\begin{aligned} \mathbf{w} &= \mathbf{R}^{-1}\mathbf{p} \\ &= \left(E[|s(t)|^2]\mathbf{a}(\phi)\mathbf{a}^H(\phi) + \sigma_n^2\mathbf{I}\right)^{-1} E[|s(t)|^2]\mathbf{a}(\phi) \end{aligned} \tag{5.2}$$

With some rearranging, we have

$$w = \frac{E[|s(t)|^2]}{\sigma_n^2}\mathbf{a}(\phi)(1 - \mathbf{a}^H(\phi)\mathbf{w}) \tag{5.3}$$

so $w \propto \mathbf{a}(\phi)$, where the proportionality constant is unimportant. For simplicity, we set

$$\mathbf{w} = \mathbf{a}(\phi) \tag{5.4}$$

Then the output of the antenna array is

$$\begin{aligned} z(t) &= \mathbf{w}^H\mathbf{u}(t) = s(t)\mathbf{a}^H(\phi)\mathbf{a}(\phi) + \mathbf{a}^H(\phi)\mathbf{n}(t) \\ &= Ms(t) + \mathbf{a}^H(\phi)\mathbf{n}(t) \end{aligned} \tag{5.5}$$

The power of the desired signal component of z(t) is

$$P_z = M^2 E[|s(t)|^2] \tag{5.6}$$

The power of the noise component of z(t) is

$$\begin{aligned} Nz &= E[\mathbf{a}^H(\phi)\mathbf{n}(t)\mathbf{n}^H(t)\mathbf{a}(\phi)] \\ &= \mathbf{a}^H(\phi)E[\mathbf{n}(t)\mathbf{n}^H(t)]\mathbf{a}(\phi) \\ &= \mathbf{a}^H(\phi)\sigma_n^2\mathbf{I}\mathbf{a}(\phi) = M\sigma_n^2 \end{aligned} \tag{5.7}$$

Therefore, the signal to noise ratio of z(t) is

$$\gamma_z = M \frac{E[|s(t)|^2]}{\sigma_n^2} \tag{5.8}$$

If, instead of using the array of M elements, only the signal at a single array element, $u_o(t)$, is used, then

$$u_o(t) = s(t) + n(t) \tag{5.9}$$

The signal-to-noise ratio of $u_o(t)$ is

$$\gamma_z = \frac{E[|s(t)|^2]}{\sigma_n^2} \tag{5.10}$$

Comparing (5.8) and (5.10), we see that an M-element array can achieve an SNR improvement of

$$G = 10\log(M)\ \text{dB} \tag{5.11}$$

in Additive White Gaussian Noise (AWGN) with no interference or multipath. In our simulation, because the algorithms cannot reach the optimal solution for weight updating, the SNR improvement will be worse than the theoretical value.

b). The BER performances of these algorithms are comparable. In Section 5.2.2 we explained this point by observing the beam pattern (Figure 5-1) of the algorithms. Through the simulation, we find that the BER performance is also closely related to the parameters of the algorithms. Improper parameters adjustments tend to lower the BER performance. This is not contradictive with the criteria of which we select parameters for faster convergence rate, because the algorithm with a normally worked parameter can always form a good beam pattern and produce a good BER performance.

One exception of the performance is in the case of LMS algorithm. The BER performance of LMS algorithm in 8 elements case is obviously worse than other algorithms, no matter what step-size parameter μ is selected. We find that it is difficult to get a satisfied performance simultaneously in higher E_b/N_o and lower E_b/N_o case from LMS algorithm. The NLMS algorithm can adjust the step-size parameter automatically according to the input signal; the performance of NLMS algorithm is a little better than that of LMS algorithm.

c). Comparing Figure 5-10 and Figure 5-11, we find that the BER performance in the case of UCA is almost the same with that in the case of ULA for every algorithm in AWGN channel. Because 2 elements ULA and UCA give completely the same array pattern, in Figure 5-11 we just show the performances with 4 and 8 antenna elements.

5.3.2 BER Performance versus Number of Interferers

In Section 4.2.2 we analyzed the Multiple Access Interference in DS-CDMA Systems and in Appendix A of [1] it is shown that the Multiple Access Interference can be approximated as a Gaussian random variable with the variance given by Equation (4.17). In this section we will use this concept to simulate the BER performance versus the Number of Interferers. As we mentioned before that in our receiver structure each interferer only

contributes a small value to the data vector after the despreading, and the algorithms performed are not intended to null out the interferers but simply maximizes the gain along the direction of the desired signal source. Therefore the direction of the interference signal is not important in this structure. Note that the previous assumption hold under the assumption of the perfect power control is applied in the CDMA system.

Figure 5-12 shows the BER performance for CDMA reverse link with conventional receiver (no antenna array used) when the number of users changes. The Processing Gain is 31 and 63 respectively; perfect power control is applied such that all users have the same power level. From the figure we see that the BER performance decrease when number of users increase, and larger Processing Gain reduce the interference level which will cause better BER performance. Note that in later simulation we still fix the Processing Gain at 31.

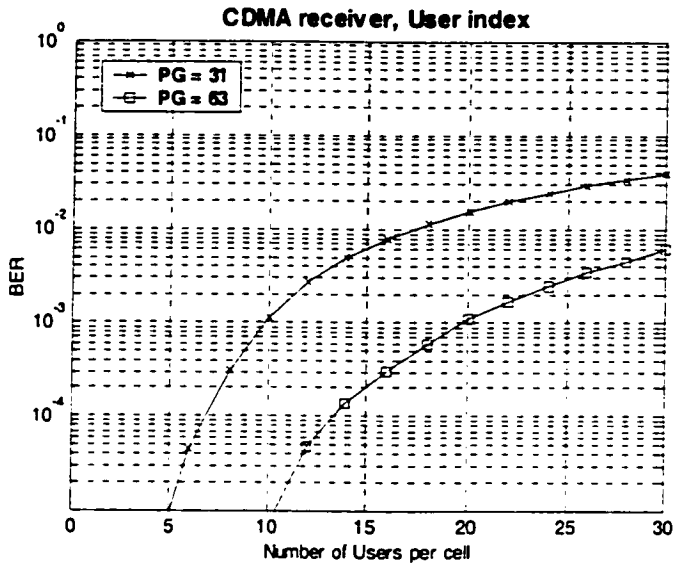


Figure 5-12: BER performance for CDMA reverse link, with PG = 31, 63

Figure 5-13 shows the BER performance versus Number of Interferers in AWGN channel. In this figure, RLS, CAM and BEE algorithms are performed in ULA case with 3 antenna elements and 6 antenna elements, E_b/N_o = 4dB. As we mentioned in last section, the performance in UCA is almost the same with that of ULA in AWGN channel, we will not

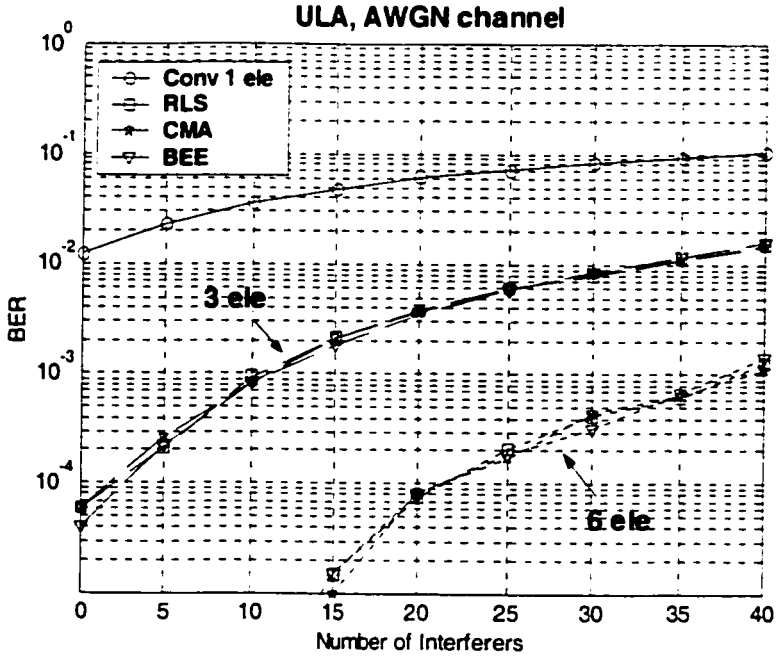


Figure 5-13: BER performance vs. Number of Interferers, Eb/No = 4dB, 3 and 6 elements.

show the performance in UCA case. Compared with the conventional receiver, we see that even use the array with 3 elements, 10 dB BER performance improvement still can be reached when number of interferers is greater than the Processing Gain. The more antenna elements are employed, the more improvement we can get. The three algorithms still have a same performance in this case.

5.3.3 BER Performance in Timing Offset case

In a CDMA system, due to the low SNR and high interference level, the synchronization of the PN sequence in the receiver cannot be performed perfectly; therefore a timing offset exists between the local generated PN sequence and received PN sequence. The timing offset will degrade the performance of the system, because the timing offset finally lower the power level of the desired signal after despreading while the noise and

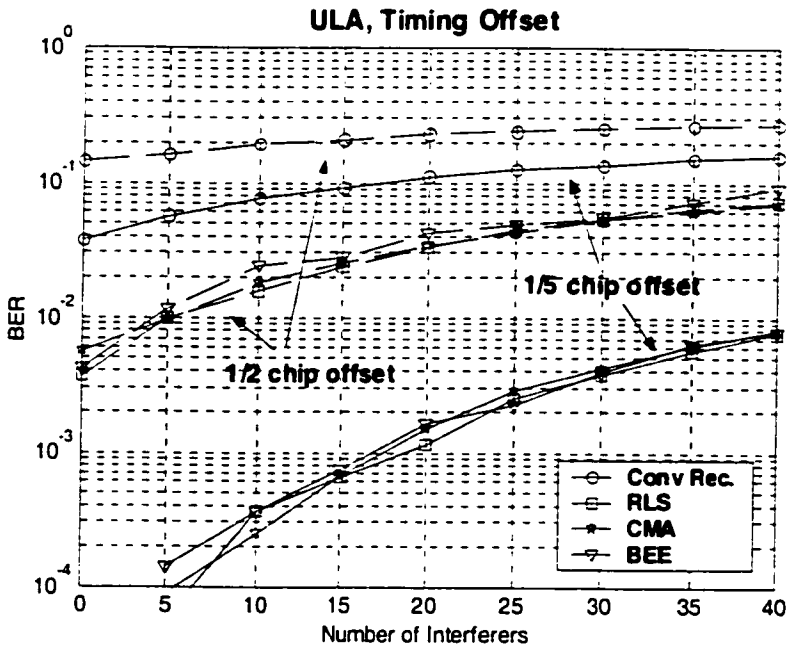


Figure 5-14: BER performance, Timing Offset, Eb/No = 4dB, ULA 6 elements.

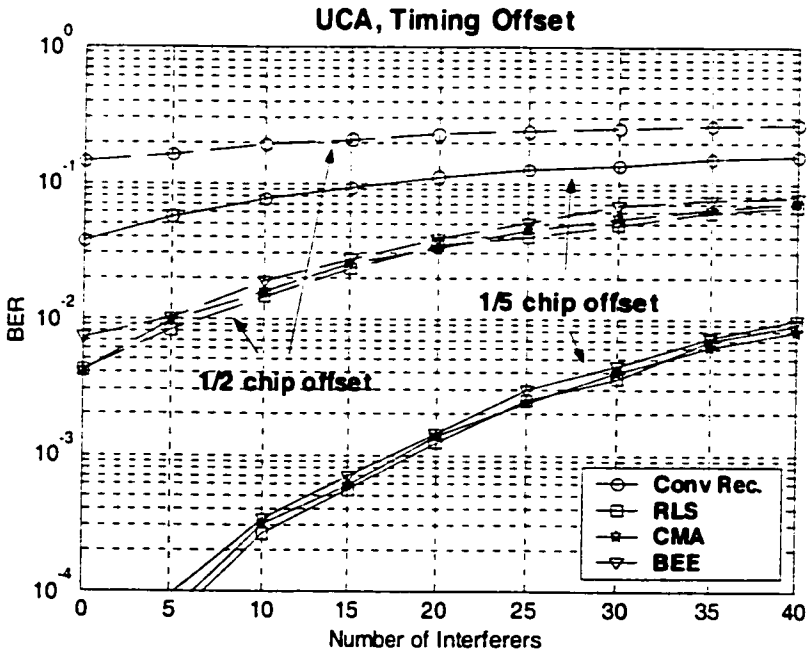


Figure 5-15: BER performance, Timing Offset, Eb/No = 4dB, UCA 6 elements.

interference level will not be changed. In this section, we will examine the BER performance of the algorithms under different timing offset conditions. We will consider two timing offset cases, offset = 1/5 *chip* and offset = 1/2 *chip*. UCA will also be considered in this case.

Figure 5-14 shows the BER performance versus Number of Interferers with 1/5 chip offset and 1/2 chip offset. In this figure, RLS, CAM and BEE algorithms are performed in ULA case with 6 antenna elements, E_b/N_o = 4dB. The figure shows that the timing offset degrade the BER performance significantly, especially when it is large. When the timing offset is small, such as offset = 1/5 *chip*, the improvement of the performance is noticeable by using the adaptive array. But when the offset becomes larger, i.e. offset = 1/2 *chip*, although some improvement can still be obtained, the effect of using adaptive array will be degraded.

Figure 5-15 gives the performance in the UCA case, under the same condition with that in Figure 5-14. We see that the difference in BER performance between UCA and ULA is still neglectable in timing offset case.

5.4 BER Performance in Rayleigh Flat Fading channel

In Section 4.3 we introduced the signal model in fading channel. In this section we will present some simulation result of BER performance in Flat Fading channel. First let us review a mathematical expression of the Flat Fading channel.

Consider a transmitted signal $s(t) = A\ cos2\pi f_c t$ through a fading channel. The received signal can be expressed as (ignoring the effects of noise):

$$y(t) = A\sum_{i=1}^{N} a_i \cos(2\pi f_c t + \theta_i) \qquad (5.12)$$

Where a_i is the attenuation of the ith multipath component, θ_i is the phase-shift of the ith multipath component. a_i , θ_i are random variables. The above expression can be re-written as:

$$y(t) = A\left\{\left(\sum_{i=1}^{N} a_i \cos(\theta_i)\right)\cos(2\pi f_c t) - \left(\sum_{i=1}^{N} a_i \sin(\theta_i)\right)\sin(2\pi f_c t)\right\} \qquad (5.13)$$

We introduce two random processes $X_1(t)$ and $X_2(t)$, such that the above equation becomes:

$$y(t) = A\{X_1(t)\cos(2\pi f_c t) - X_2(t)\sin(2\pi f_c t)\} \qquad (5.14)$$

If the value of N is large (i.e, a large number of scattered waves are present), invoking the *Central-Limit Theorem*, we get approximate $X_1(t)$ and $X_2(t)$ to be *gaussian random variables* with zero mean and variance σ^2. The expression can be re-written as:

$$y(t) = AR(t)\cos(2\pi f_c t + \theta(t)) \qquad (5.15)$$

Where, The amplitude of the received waveform $R(t)$ is given by:

$$R(t) = \sqrt{X_1(t)^2 + X_2(t)^2} \qquad (5.16)$$

Since the processes $X_1(t)$ and $X_2(t)$ are gaussian, it can be shown that $R(t)$ has a *Rayleigh Distribution* with a probabilit density function (pdf) given by:

$$f_R(r) = \frac{r}{2\sigma^2} e^{\frac{-r^2}{2\sigma^2}}, \qquad r > 0 \qquad (5.17)$$

The phase of the received waveform $\theta(t)$ is given by

$$\theta(t) = \tan^{-1}\left(\frac{X_2(t)}{X_1(t)}\right) \qquad (5.18)$$

Since the processes $X_1(t)$ and $X_2(t)$ are gaussian, it can be shown that $\theta(t)$ has a *Uniform Distribution* with a probability density function given by:

$$f_\theta(\theta) = \frac{1}{2\pi}, \qquad -\pi \le \theta \le \pi \tag{5.18}$$

The distortion in the phase can be overcome by synchronization system and some modulation technique (such as differential modulation) in normal cases (when speed of the mobile is not too high). The amplitude distortion $R(t)$ severely degrades performance of digital communication systems over fading channel.

From above description we see that the *Rayleigh Random Variable* can be used to describe the statistical time varying nature of the received envelope of a Flat Fading signal. Actually, because the bandwidth of the spreading spectrum signal is much larger than the *Coherence bandwidth*, the Frequency Selective Fading is most likely suffered in CDMA communication. But one path of the Frequency Selective Fading sometimes can be well modeled as a Rayleigh Fading when a large number of scattered waves exist in that path [9][38]. Further, the typical flat fading channels cause deep fades, simulation of the Rayleigh Flat Fading channel can examine the adaptability of the algorithms in such condition.

As we know, fading channel is time varying and is described by Doppler Spread and Coherence Time, as we introduced in Section 4.3.2. In order to keep the current channel information, the period between two snapshots to update the weigh of the antenna array should be much less than the Coherence Time. The coherence time is rewritten as follow,

$$T_{CT} \approx \sqrt{\frac{9}{16\pi f_m^2}} = \frac{0.423}{f_m} \tag{5.19}$$

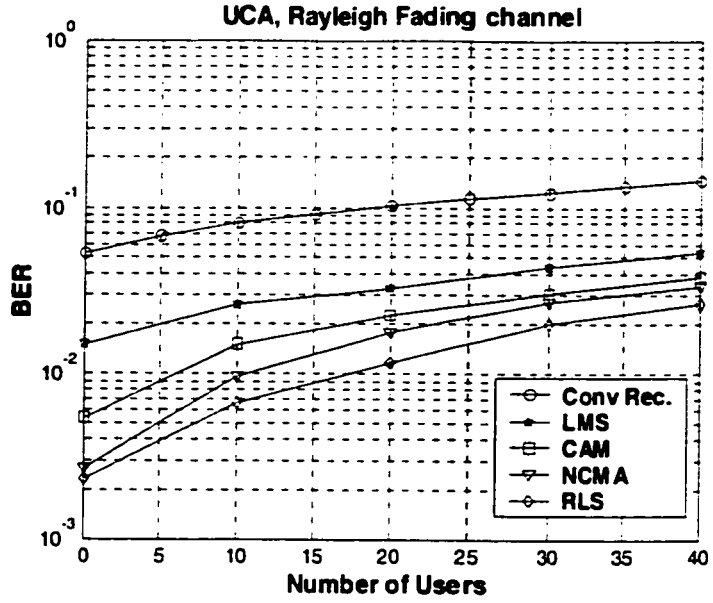


Figure 5-16: BER performance in Rayleigh Flat Fading channel. UCA, 8 elements, Eb/No = 8 dB.

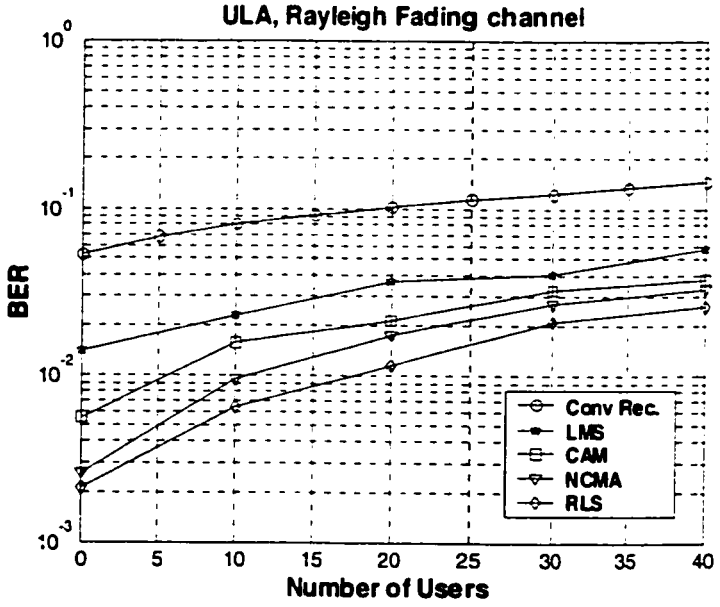


Figure 5-17: BER performance in Rayleigh Flat Fading channel, ULA, 8 elements, Eb/No = 8 dB.

where f_m is the maximum Doppler shift given by $f_m = vf_c/c$ (v is the speed of mobile, f_c is the carrier frequency and c is the speed of light). If we consider the worse case in which the speed of mobile is 150km/h, and f_c is 1.9GHz, we can get f_m= 264 Hz. Then T_{CT} = 1.6 ms, which means that the weight-updating period should be much less than this value.

If the adaptive antenna system is employed in IS-95 CDMA system, we know from Section 4.4 that after despreading the symbol rate is 28.8 kbps, corresponding a symbol period of 35 μs, which is much less than above Coherence time. If the definition in Table 4-1 is used, we say that the signal is suffering *Slow Fading*. Doppler phase offset caused by the motion of the mobile is assumed to be corrected by proper synchronization. The case of imperfect synchronization will be considered in next section.

For some simple blind algorithms (CMA, BEE etc.), the weight vector can be updated in each symbol period (35 μs in above case) after initial convergence. For the non-blind cases, the overhead must be considered. If the weight vector is updated in every 1 ms period and only one symbol is transmitted as training signal in this period (after initial convergence), we still have about 1/30 overhead. The situation will be worse in practice.

Figure 5-16 shows the BER performance in Rayleigh Flat Fading channel with 8 elements UCA, E_b/N_o = 8dB. In this case, NCMA outperforms CMA due to its better adaptability, but they are a little inferior to that of RLS algorithm. The performance of an algorithm in Rayleigh Fading channel basically depends on whether it can form an effective beam in deep fading. Any failure of the beam forming will cause extremely high bit error rate. Obviously, RLS algorithm has the strongest ability of doing this, NCMA next, and so on.

The case of ULA is shown in Figure 5-17. The performance of the algorithms in ULA is similar to that in UCA as usual.

5.5 BER Performance in Frequency Selective Fading channel

In Chapter 4 we explained that the bandwidth of CDMA signal is much large than the *Coherence bandwidth*, therefore the Frequency Selective Fading most likely occurs in CDMA communication environment. In such a case, the transmitted signal may arrive at the receiver through different paths with different time delays. This multipath will cause the Inter Symbol Interference (ISI) and degrade the BER performance of the system. However, we can use an adaptive array with 2D-RAKE receiver to combine and extract the path with the stronger power, therefore overcoming the ISI and improving the BER performance. In this section, we will examine the BER performance of the algorithms and the 2D-RAKE receiver under different multipath environment.

5.5.1 Signal Model

In Section 4.3.1, we introduced a discrete signal model and the Vector Channel Impulse Response. Based on these results we will analyze the received signal at each antenna element in detail and further apply it into our simulation. The received baseband signal at the m[th] antenna element of the array is

$$u_m(t) = \sum_{i=1}^{L}\left(\frac{1}{\sqrt{K_i}}\sum_{k=1}^{K_i} e^{j2\pi(f_i\cos\varphi - f_c\varsigma_{ki})}e^{-j\omega_c\tau_m}s_i(t-\varsigma_{ki})\right) + n_m(t) \qquad (5.20)$$

where L is the number of propagation paths of the desired user, K_i is the number of scattered components at the i^{th} path, f_i is the Doppler shift of i^{th} path due to the motion of either a mobile unit or scatterers in the environment, and φ_i is the direction of the k^{th} component of the i^{th} path. f_c and ω_c is carrier frequency and ς_{ki} is the delay of the k^{th} component of the i^{th} path, τ_m is the time delay on m[th] antenna element, which depend on the array pattern (such as

ULA or UCA). $n_m(t)$ represents the undesired terms consisting of interference and noise, which are assumed to be zero-mean white Gaussian.

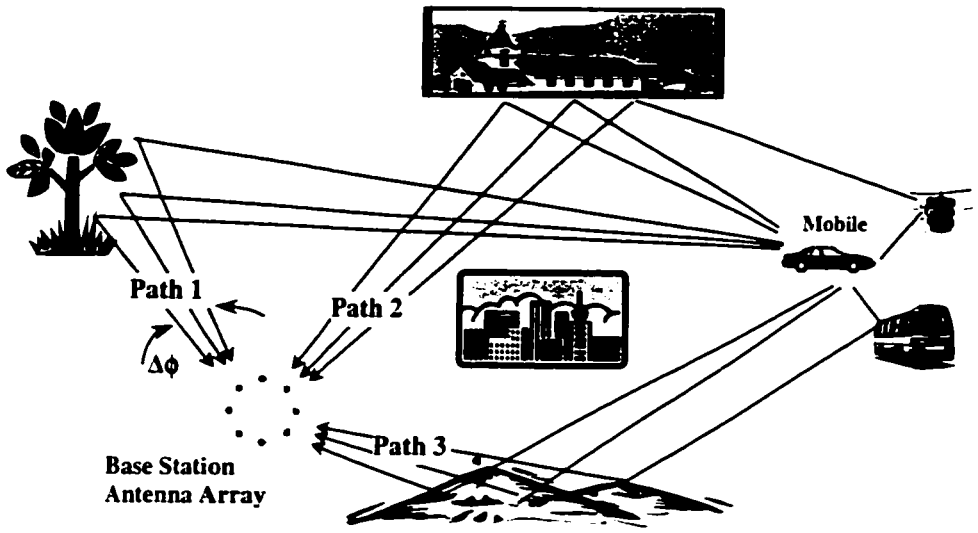


Figure 5-18: An illustration of multipath propagation channel

Figure 5-18 illustrates a multipath propagation channel that our signal model is based on. In this model, each propagation path consists of many scattered components. If our transmitted signal is narrowband signal, all of the propagation paths and their scattered components are correlated, that is the Delay spreads of the multipath signals are much less than the symbol period, the signal is actually in Flat Fading. But currently our transmitted signal is wideband CDMA signal, each propagation path is most likely resolvable by more than one chip duration, 2D-RAKE receiver can be used in combining the signals from different paths. The adaptive antenna array can also be used to combine the signals if the time deference of the propagation delays of the strong paths is happened less than one chip, as we will show in following simulation. Note that even if the propagation paths are resolvable, the scattered components in that path are assumed unresolvable. In another

words, the delay spread of scattered components in one propagation path is less than one chip duration, i.e. $\sigma_\tau < T_c/4\pi$, as we explained in Section 4.3.2.

Observing Figure 5-18, we see that the scattered components are actually incident from different angle. We assume that all components from the mobile arrive at the base station antenna array uniformly distributed within $\pm\Delta\phi$ of the mean angle of arrival ϕ [38]. The value of $\Delta\phi$ is called the *Angle Spread* around the mean angle of arrival (shown in Figure 5-18). The large *Angle Spread* will degrade the performance of the system. In next section we examine the effect of Angle Spread on different array pattern.

In the following simulations, we assume that the phase offset caused by imperfect synchronization is less than $\pi/4$.

5.5.2 Effect of Angle Spread

Angle Spread will cause the carrier phase delay at each antenna element to be different from one another [39][34]. When the carrier phase is compensated by multiplying the complex conjugate of the term $\sum_{k=1}^{K} e^{j2\pi(f_c \cos\phi - f_c\tau_k)}$ to the received signal induced at every antenna element, the carrier phase delay is not compensated precisely except for the reference antenna element, because the amount of carrier phase delay is different at each antenna element due to the angle spread. The error in compensating the carrier phase delay increases as the distance from the reference antenna element becomes larger.

Now we assume the spacing between adjacent antenna element equals d in both ULA and UCA, and 8 antenna elements are used, we see that the distance between the reference element to the farthest elements equals 7d in ULA, but only 2.6d in UCA. We can infer that in the Angle Spread case the performance of using UCA should be better than that of using

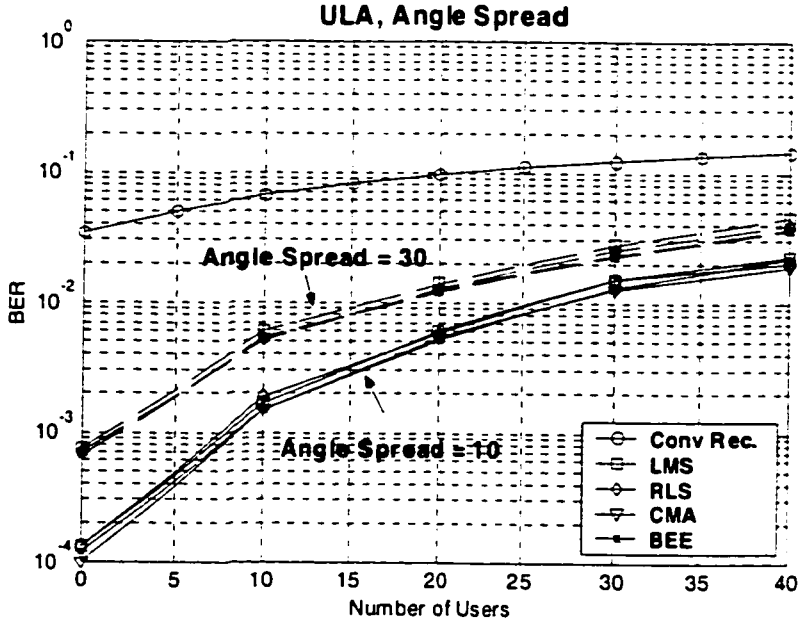


Figure 5-19: BER performance in Frequency Selective Fading with Angle Spread, ULA, 4 elements, Eb/No = 4 dB.

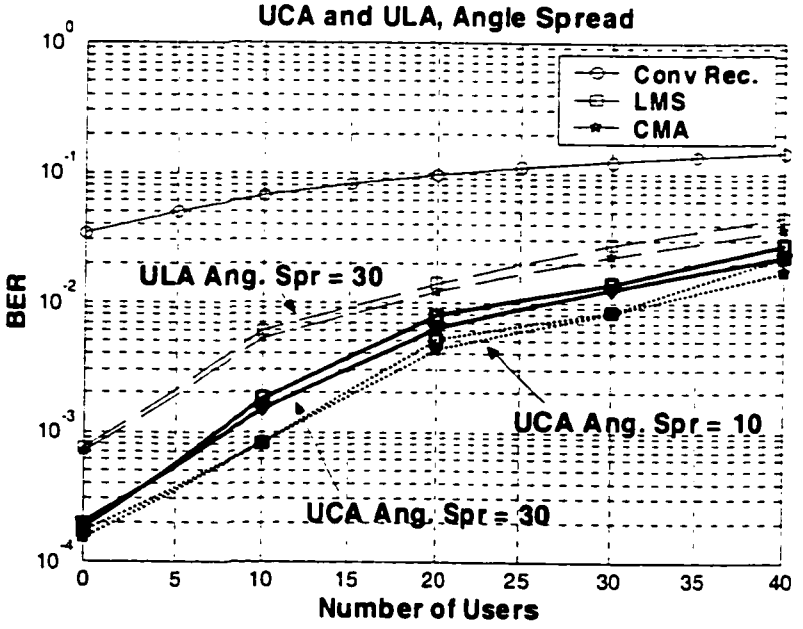


Figure 5-20: BER performance in Frequency Selective Fading with Angle Spread, ULA and UCA, 4 elements, Eb/No = 4 dB.

ULA, and the wider the angle spread, the more the performance improvement can be reached by using UCA compared with using ULA. This point will be proved by the simulation result.

In order to illustrate how the Angle Spread affects the BER performance, we assume that there is only one propagation path exist from mobile to the base station, and we also assume that there are 20 scatter components in the path. The term $2\pi(f_i \cos \varphi t - f_c \varsigma_i)$ modulo 2π can be modeled as i.i.d random variables uniformly distributed over $[0,2\pi]$ [38][40].

Figure 5-19 shows the comparison of the BER performance when Angle Spread = 10° and 30°. In this case the algorithms are performed under 4 antenna elements ULA, with E_b/N_o = 4dB. We see that the BER performance at Angle Spread = 10° is about 3 dB better than that of 30° Angle Spread in average, which means the performance degrade when the Angle Spread increase. The other algorithms perform similar in this case.

Figure 5-20 illustrates the BER performance in both UCA and ULA case, with 4 elements and E_b/N_o = 4dB. As shown in the figure, the case of using UCA conspicuously outperforms the case of ULA, as we inferred above. More improvement on the performance is expected if more antenna elements are used. This is reasonable because the distance difference between the reference element and the farthest element increase when the number of elements increases. Comparing Figure 5-20 with Figure 5-19, we also find that the UCA is less sensitive to the Angle Spread. An average of 3dB improvement can be get in ULA case when the Angle Spread is decreased from 30° to10°, while only 1dB improvement can be obtained in UCA case.

5.5.3 Antenna Array Signal Combining

In this section, we will consider 3 propagation paths arriving from different directions with 20 scatter components in each path, and the difference of propagation delays between the paths is less than one chip duration of the PN sequence. In this case, the antenna array will produce 3 main beams for each path and the gain of main beams basically depends on the relative power of the signals. Therefore in such case the desired signal propagated from different paths is combined by the antenna array. Depending on the delay difference and power difference of each path, we will divide this case into three sub-cases.

Table 5-1 Paths Parameters used in simulation

Path #	DOA of Paths	# of Scatter Rays	Angle Spread
1	45°	20	15°
2	85°	20	15°
3	135°	20	15°

The multipaths parameters used in our simulation are summarized in Table 5-1. We also assume that in following simulations the PN sequence is synchronized at path 1. We will consider three sub cases with different delay difference and power ration between the paths.

Case (a).

Delay Difference (chips) path 1 / path 2	Delay Difference (chips) path 1 / path 3	Power Ratio (dB) path 1 / path 2	Power Ratio (dB) path 1 / path 3
0	0	0	0

In this sub case we assume that the signals from 3 paths arrive at the array at same time with same power. An 8-element ULA with 15° Angle Spread polar coordinate beam pattern of **Case (a)** is shown in Figure 5-21. The beam pattern is generated by CMA algorithm at E_b/N_o = 4dB.

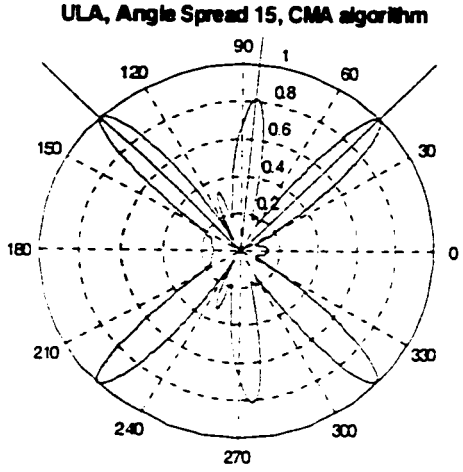


Figure 5-21: ULA, 8 elements, Angle Spread = 15°, Eb/No = 4 dB.

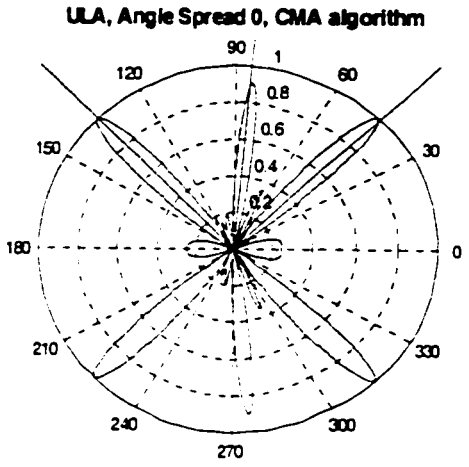


Figure 5-22: ULA, 8 elements, Angle Spread = 0°, Eb/No = 4 dB.

Figure 5-22 is produced under the same condition with Figure 5-21 except Angle Spread equals 0°. Comparing the two beam patterns, we see that the three main beam widths in 15° Angle Spread case are wider than 0° Angle Spread case. This means that the algorithm attempts to widen the beam width to include more scatters power in the beam when the Angle Spread becomes wider. The same situation exists in UCA, as shown in Figure 5-23 and Figure 5-24.

Figure 5-25 illustrates the BER performance of **Case (a)**, where 8-element UCA and ULA are compared with E_b/N_o = 8dB. From this figure we see that the performance of UCA significantly outperform ULA. As we remarked in Section 5.5.2, more improvement of using UCA can be obtained when we use 8-element array comparing to use 4-element array.

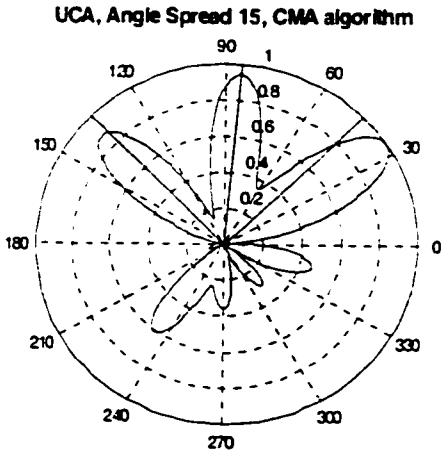


Figure 5-23: UCA, 8 elements, Angle Spread = 15°, Eb/No = 4 dB.

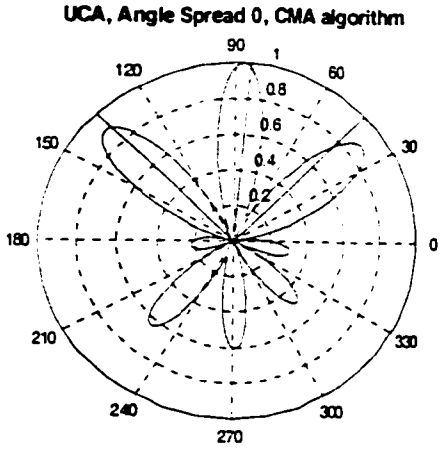


Figure 5-24: UCA, 8 elements, Angle Spread = 0°, Eb/No = 4 dB.

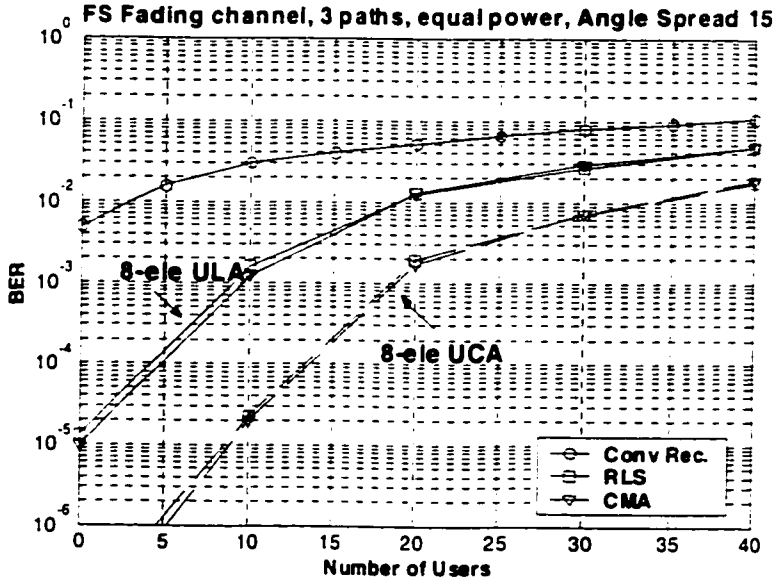


Figure 5-25: BER performance of FS Fading channel, with equal power 3 paths, 8-element UCA and ULA, Eb/No = 8 dB

The RLS and CMA algorithms are used to produce the figure. We see that their performances are similar. Another phenomenon should be noticed in this case is that the

BER performance is sensitive to the noise and interference level, as shown in the figure, the BER performance become worse sharply when the number of users increases.

Case (b).

Delay Difference (chip) path 1 / path 2	Delay Difference (chip) path 1 / path 3	Power Ratio (dB) path 1 / path 2	Power Ratio (dB) path 1 / path 3
0	0	2	4

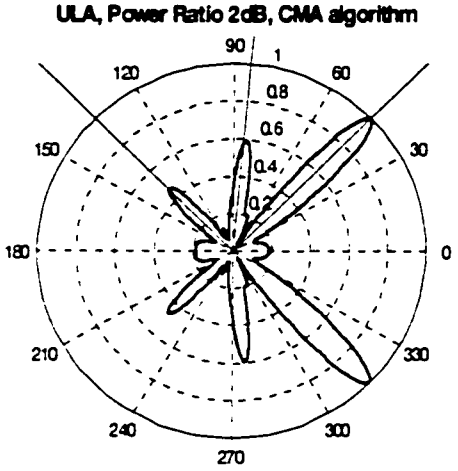


Figure 5-26: ULA, 8 elements, Angle Spread = 15°, power ratio 2 dB

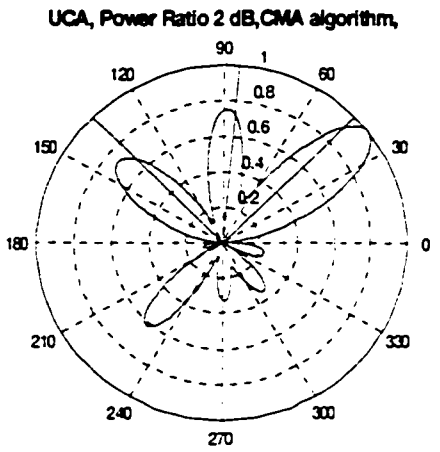


Figure 5-27: UCA, 8 elements, Angle Spread = 15°, power ratio 2 dB

In **Case (b)** we assume that the power ratio between path 1 and path 2 equals 2dB, and 4dB between path 1 and path 3. Here we assume that the received power equals one if no multipaths exists, and the power is divided according to the power ratio if multipahts is present. In another words, the summation of multipaths power is still equal to one.

The polar coordinate beam pattern of 8-element ULA and UCA with 15° Angle Spread and 2dB power ratio between paths are shown in Figure 5-26 and Figure 5-27. The beam

pattern is generated by CMA algorithm at E_b/N_o = 4dB. From the figures we see that gain of path 1 is greater than that of path 2 and so does it than path 3. This means that the gain of particular main beam actually depends on the power of the path which the main beam points to. If the power of particular path is much stronger than other paths, the algorithm will generate an overwhelming main-beam pointing to that path. This case is similar to a single path situation.

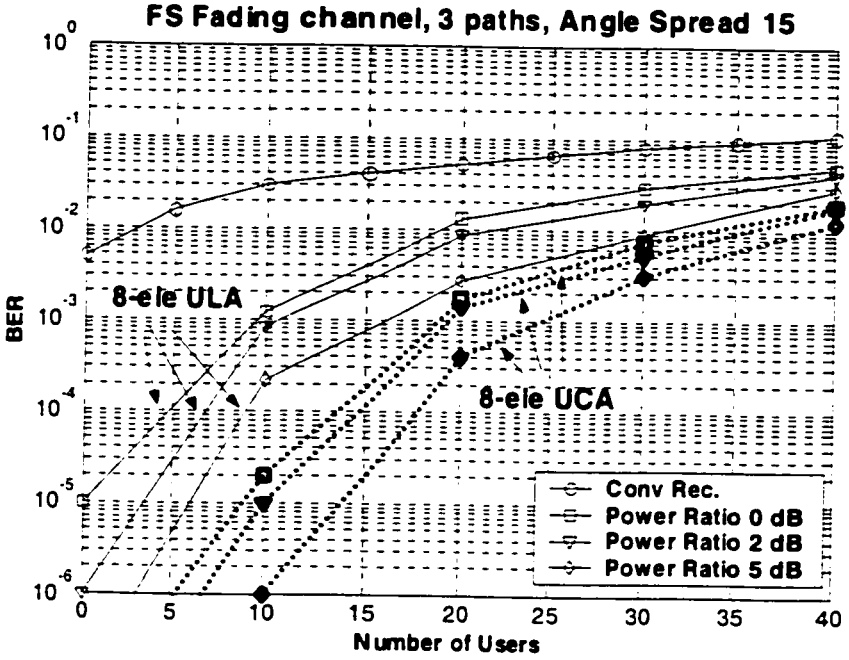


Figure 5-28: BER performance of FS Fading channel, with power difference, 8-element UCA and ULA, Eb/No = 8 dB

Figure 5-28 illustrates the BER performance with power ratio of 2dB and 5dB between path 1 and path 2, and the same ratios between path 2 and path 3. A equal power curve also shown for comparison. The figure is produced by CMA algorithm on 8-element ULA and UCA at E_b/N_o = 8dB. We see in the figure that as the power ratio between paths increase, the BER performance improves as well. The case with equal power ratio has a

worst performance. This is because the equal power ratio will cause equal main-beams which are pointing to each path. The main-beams not only increase the gain toward particular direction, they increase the interfering power as well. The more the number of main-beam, the more interfering power they bring in. In addition, with the unity total power assumption, the increased power ratio tend to force the gain concentrating on the path with maximum power, thus the noise introduced from multi-beam is decreased, making the BER performance improves.

Case (c).

Delay Difference (chip) path 1 / path 2	Delay Difference (chip) path 1 / path 3	Power Ratio (dB) path 1 / path 2	Power Ratio (dB) path 1 / path 3
1/4	2/4	0	0

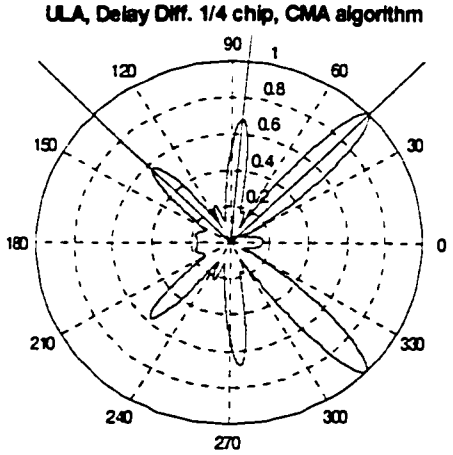


Figure 5-29: ULA, 8 elements, Angle Spread = 15°, delay diff. 1/4 chip

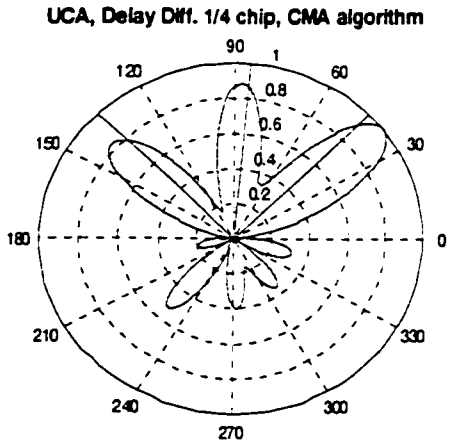


Figure 5-30: UCA, 8 elements, Angle Spread = 15°, delay diff. 1/4 chip

Now let us examine the case with delay difference existing between equal power paths. Here we assume that the delay difference between path 1 and path 2 equals 1/4 chip. and 2/4 chip between path 1 and path 3. The polar coordinate beam patterns of 8-element ULA and UCA with 1/4 chip delay difference between paths are shown in Figure 5-29 and Figure 5-30. The beam pattern is generated by CMA algorithm at E_b/N_o = 4dB. Because the autocorrelation of the PN sequence with its shifted version just degrades the amplitude of the post correlation signal, we know from case (b) that if the delay difference between path 1 and other paths is greater than or equals to chip duration, the algorithms will only form a single main-beam to path 1, and treat the other paths as interference. It is important to note that at this time the path of the single main-beam just own 1/3 of the total power, which is unlike case (b).

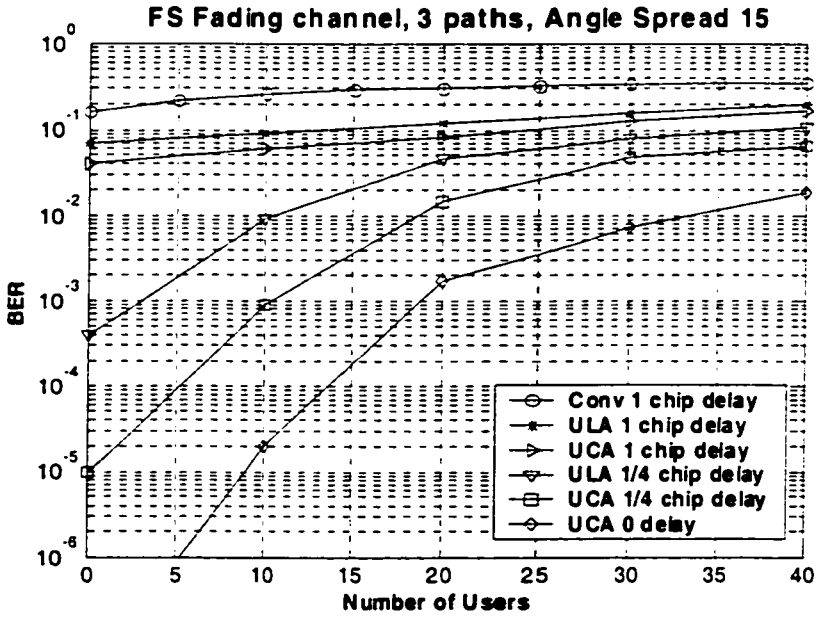


Figure 5-31: BER performance of FS Fading channel, with delay difference, 8-element UCA and ULA, Eb/No = 8 dB

The above remark hold under the assumption of the PN sequence being synchronized with path 1. In practical system the synchronization system most likely track to the path with strongest power.

Figure 5-31 shows the BER performance of this case. The figure is produced by CMA algorithm on 8-element ULA and UCA at E_b/N_o = 8dB. We see that the worst case exist in the case when the delay difference is greater than chip duration, and the less the delay difference, the better the performance.

From above we see that if the delay difference of the paths is longer than chip duration, the antenna array cannot combine the signals from different paths, which will cause some power loss. This situation seems to occur more frequently in CDMA communication because of the wideband property of CDMA signal. Therefore it is meaningful to use the 2D-RAKE receiver to collect the energy that cannot be combined by antenna array.

5.5.4 2D RAKE Receiver Combining

As we described in last section, the 2D-RAKE receiver can be used to combine the signal from different paths when delay difference of the paths is longer than chip duration. In order to perform the RAKE reception, the received signal, consisting of signals from all the users in a given cell, is correlated with the PN code sequence of each target user with a number of different time lags. Then for each value of the time lag, the received signal is despreaded. The number of distinct time lags in the despreading procedure for receiving the signal from each user is the number of fingers in the RAKE receiver [33]. The RAKE receiver structure used in our simulation is shown in Figure 4-5.

Figure 5-32 illustrates the BER performance of the 2D RAKE receiver. The figure is generated by CMA algorithm on 8-element ULA and UCA at E_b/N_o = 8dB and under the

assumption of a perfect synchronization is performed in the receiver. The case with one chip delay difference is also shown for comparison. Comparing with Figure 5-25, we see that the RAKE receiver with 3 fingers case is similar to the case with no delay difference and equal power. If a 2 fingers structure is used, only the powers from two paths can be collected in the receiver, cause the performance degrade comparing to 3 fingers case.

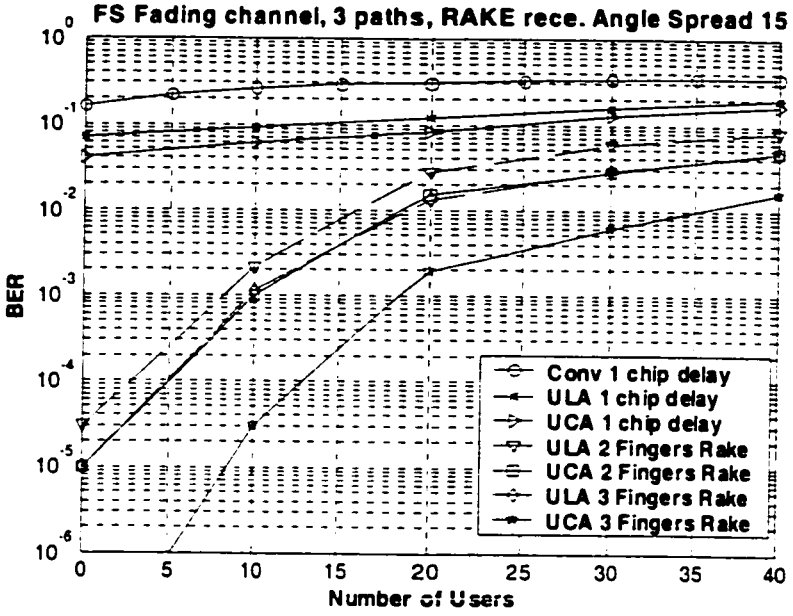


Figure 5-32: BER performance of FS Fading channel, with 2D-RAKE receiver, 8-element UCA and ULA, Eb/No = 8 dB

5.6 BER Performance in Mobile Tracking

In previous simulations we assumed that the angle of arrival was fixed in the process of communication. In order to examine the tracking capability of the algorithms, this section we will simulate the case where the angle of arrival is changing in the process of communication.

First let us estimate the time interval between the adjacent snapshots of an algorithm, i.e. the least time required per iteration for the algorithm. We consider an array consisting of

10 antenna elements. From 5.2.4 we examined the exact number of multiplicaitons and additions required per iteration for the algorithms (calculated by number of elements M), and we know that the total number of operations required to update the weigh would be less than 500 for LMS, CMA and BEE algorithm considering some additional operations that might possible be needed due to the restrictions on general-purpose processors [31][33]. Since the instruction cycle of a modern digital signal-processing chip is less than 70 ns (depends on the type of the chip), the lower bound of the time interval between snapshots would be less than 40 μs (The RLS algorithm will take longer due to its complexity). If the moving speed of the subscriber is 150km/h, the angle change of this subscriber in 40 μs is much less than 0.01°even if he is very close to base station, and 40 μs is also much less than the Coherence Time T_{CT} = 1.6 ms, which we calculated in Section 5.4.

In the simulation, we assume that there is only one 15° Angle Spread path exist.

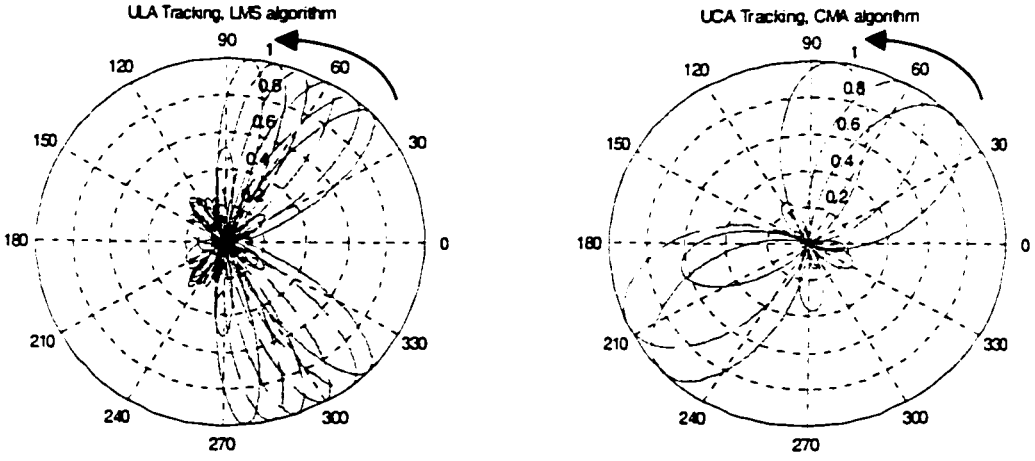


Figure 5-33: Beampattern Tracking Trace

Figure 5-33 shows the Beampattern tracking trace. The initial SOI is 45°, ending at 85°. 4 elements, Eb/No = 2dB, 5 interferences. In this case, the angle change is 0.02°

between two snapshots with a single iteration between snapshots.

Figure 5-34 shows the tracking capability of LMS algorithm and CMA algorithm. The tracking curve shown in the figure is produced in such a way that every snapshot the DOA changes one tracking step and performs one iteration of algorithm. We see that for CMA the performance does not degrade when the tracking step less than 1°. The tracking capability of LMS algorithm is worse than that of CMA when the tracking step is less than 2.4°. But note that in practice the angle change during adjacent snapshots is not possible great than 0.01° even in the worse case. This means that algorithms performed in our simulated structure satisfy the requirement of the real-time implementation.

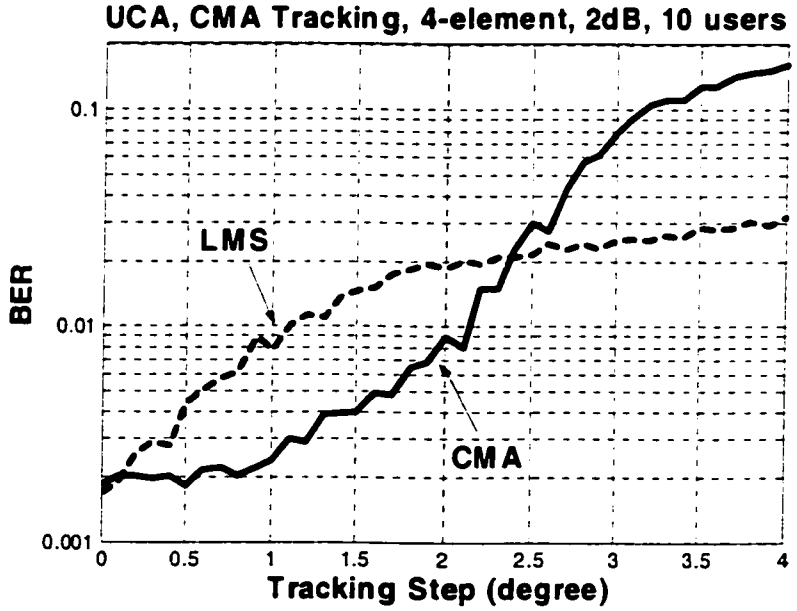


Figure 5-34: Tracking Capability

Chapter 6

Conclusions and Future Work

6.1 Conclusions

In this thesis, we performed a through comparison of several adaptive algorithms and two array geometries in the CDMA post-correlation beamformer under different conditions, e.g. the AWGN channel, the Flat Fading channel, the Frequency Selective Fading channel, and the phase offset and mobile tracking cases. The following conclusion can be obtained:

It was shown that the post-correlation beamformer has less computational complexity and yields superior bit error results. This structure makes some simple blind algorithms realized easily, thus the real-time implementation is obtainable. Most importantly, in this structure the algorithms can work normally whether or not the number of antenna elements is greater than the number of interference signals.

Another important result we get from the simulation is that the Uniformly Spaced Circular Array (UCA) is less sensitive to the angle spread than the Uniformly Spaced Linear Array (ULA) does; consequently the UCA exhibits better performance than the ULA in the multipath environment with angle spread, which is almost certainly the case in the uplink of CDMA communication systems.

The algorithms exhibit similar vulnerability to carrier phase error.

6.2 Suggestions of future work

1. The theoretical analysis of the different performance between the UCA and ULA is necessary.
2. It is important to perform the filed-test and compare the performance of CMA UCA and CMA ULA.

References

[1] JOSEPH.C.LIBERTI, JR & THEODORE S. RAPPAPORT, *SMART ANTENNAS FOR WIRELESS COMMUNICATIONS*, Prentice Hall, 1999

[2] S.U.Pillai, *Array Signal Processing*, Springer-Verlag, New York, 1989

[3] Z.Rong, THEODORE S. RAPPAPORT, "Simulation of MultiTarget Adaptive Algorithms for Wireless CDMA Systems", *Proc. IEEE Vehicular Technology Conf.*, Apr. 1996

[4] Z.Rong, "Simulation of Adaptive Array Algorithms for CDMA Systems", *Master's Thesis MPRG-TR-96-31*, Mobile & Portable Radio Research Group, Virginia Tech, Blacksburg, VA, Sept. 1996

[5] R.L. Peterson, R.E.Ziemer and D.E.Borth, *Introduction to Spread-Spectrum Communications*, Prentice Hall, 1995

[6] Fung.V. & T.S. RAPPAPORT, "Bit-Error Simulation for x/4 DQPSK Mobile Radio Communication using Two-Ray and Measurement-Based Impulse Response Model," *IEEE Journal on Selected Areas in Communication*, Vol. 11, No. 3, April 1993, pp 393.

[7] Simon Haykin, Ed., *Advances in Spectrum Analysis and Array Processing*, Prentice Hall, New Jersey, 1995.

[8] W. L. Stutzman and G. A. Thiele, *Antenna Theory and Design*, John Wiley & Sons, New York, 1981.

[9] THEODORE S. RAPPAPORT, *Wireless Communications Principles and Practice*, Prentice Hall, 1996

[10] W.C.Jakes, Ed., *Microwave Mobile Communicaiton*, IEEE Press, Piscataway, NJ, 1974

[11] Robert A.Monzingo, Thomas W.Miller, *Introduction to Adaptive Arrays*, John Wiley & Sons, New York, 1981.

[12] R. T. Compton, Jr., *Adaptive Antennas, Concept and Performance*, Prentice Hall, Englewood Cliffs, New Jersey, 1988.

[13] S. U. Pillai, *Array Signal Processing*, Springer-Verlag, New York, 1989.

[14] Edmond Nicolau, Dragos Zaharia, *Adaptive Arrays*, Editura Academiei, Bucharest, 1989.

[15] Simon Haykin, *Adaptive Filter Theory*, Prentice Hall, Upper Saddle River, New Jersey, 1991.

[16] J. Tsai and B. D. Woerner "The Fading Correlation Function of a Circular Antenna Array in Mobile Radio Environment", *Globecom 2001*.

[17] Kai Chang, *RF and Microwave Wireless Systems*, John Wiley & Sons, 2000.

[18] B. Suard, A. F. Naguib, G. Xu, and A. Paulraj, "Performance of CDMA mobile communication systems using antenna arrays" *Proc. IEEE ICASSP*, vol. 4, pp. 153, April 1993.

[19] S.Gummadi and Brian L. Evans, "Cochannel Signal Separation in Fading Channels Using a Modified Constant Modulus Array", gummadi@ece.utexas.edu

[20] John R. Treichler and Brian G. Agee, "A new approach to multipath correction of constant modulus signals", *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. ASSP-31, no. 2, pp. 459-471, April 1983.

[21] John R. Treichler and M. G. Larimore, "New processing techniques based on the constant modulus adaptive algorithm", *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. ASSP-33, no. 2, pp. 420-431, April 1985.

[22] B.H.Khalaj, A.Paulraj, and T.Kailath, "2D RAKE receivers for CDMA celluar systems", *IEEE International Conference on Communications*, pp. 400-404, 1994.

[23] Takeo Ohgane, "Characteristics of CMA adaptive array for selective fading compensation in digital land mobile radio communications", *Electronics and Communications in Japan*, Part 1, vol. 74, no. 9, pp. 43-53, 1991.

[24] M. G. Larimore and J. R. Treichler, "Convergence behavior of the constant modulus algorithm", *Proc. IEEE ICASSP*, pp. 13-16, April 1983.

[25] M. G. Larimore and J. R. Treichler, "Noise capture properties of the constant modulus algorithm", *Proc. IEEE ICASSP*, pp. 30.6.1-30.6.4, April 1985.

[26] J. O. Smith and B. Friedlander, "Global convergence of the constant modulus algorithm", *Proc. IEEE ICASSP*, pp. 30.5.1-30.5.4, April 1985.

[27] B. G. Agee, "Convergent behavior of modulus-restoring adaptive arrays in Gaussian interference environments", Proc. 22nd Asilomar Conf. on Signals, Systems and Computers, pp. 818-822, 1988.

[28] Y. Wang and J. R. Cruz, "Adaptive antenna arrays for cellular CDMA communication systems," *Proc. IEEE ICASSP*, pp. 1725, July 1995.

[29] R. Gooch and J. Lundell, "The CM array: An adaptive beamformer for constant modulus signals", *Proc. IEEE ICASSP*, vol. 4, pp. 2523-2526, April 1986.

[30] B. G. Agee, "The least-squares CMA: A new technique for rapid correction of constantmodulus signals", *Proc. IEEE ICASSP*, pp. 19.2.1-19.2.4, 1986.

[31] Seungwon Choi and Dongun Yun, "Design of an Adaptive Antenna Array for Tracking the Source of Maximum Power and Its Application to CDMA Mobile Communications", *IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY*. VOL.45.NO.5. SEPT. 1997.

[32] S. Choi, S. Ahn and T. K. Sarkar "A Linearized Power Method for Adaptive Beam-forming in a Multipath Fading CDMA Environment", Microwave and Optical Technology Letter, Dec.2001

[33] Seungwon Choi and Donghee Shim, "A Novel Adaptive Beam forming Algorithm for a Smart Antenna System in a CDMA Mobile Communication Environment", *IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY*. VOL.49.NO.5. SEPT. 2000.

[34] S. Choi, Jinho Choi, Heung Jae Im and B.Choi "A Novel Adaptive Beamforming Algorithm for Antenna Array CDMA systems with strong interferers",

[35] D.Shim and S.Choi. "A New Blind Adaptive Algorithm Based on Largange's Formula for a Smart Antenna System in CDMA Mobile Communications".

[36] M. Fujimoto, N. Kikuma, and N. Inagaki, "Performance of CMA adaptive array optimized by the Marquardt method for suppressing multipath waves," *Electronics and Communications in Japan*, Part 1, vol. 75, no. 9, pp. 89, 1992.

[37] A. Mathur, A. V. Keerthi, and J. J. Shynk, "Estimation of correlated cochannel signals using the constant modulus array " *Proc. IEEE Int. Conf. Commun.*, vol. 3, pp. 1525, June 1995.

[38] A.F. Naguib, "Adaptive antennas for CDMA wireless networks", Ph.D. dissertation, Dept. Elec. Eng. Stanford University, Aug. 1996.

[39] Heung-Jae Im, S. choi, B.choi, H.kim and J.choi, "A Survey of Essential Problems in the Design of Smart Antenna System",

[40] W.R.Braun and U.Dersch, "A Physical Mobile Radio Channel Model", *IEEE Trans. Veh. Tech., vol VT-40, pp. 472, May 1991.*

[41] K. Hilal and P. Duhamel, "A blind equalizer allowing soft transition between the constant modulus and decision-directed algorithm for PSK modulated signals" *Proc. IEEE Int. Conf. Commun.*, vol. 2, pp. 1144, 1993.