Adaptive Equalization for the Downlink of a 3G W-CDMA System

A Project Report

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

Bachelor of Technology

in

Electrical Engineering

by

S Sunil

under the guidance of

Dr. B. Srikrishna

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, MADRAS
May 2004

Certificate

This is to certify that the project entitled **Adaptive Equalization for the Downlink of a 3G W-CDMA System** by **S Sunil** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology**, is a bona-fide record of work carried out by him under my supervision and guidance at the Department of Electrical Engineering, Indian Institute of Technology Madras.

Place: Chennai [Dr. B. Srikrishna]
Date:

Acknowledgments

I express my sincere gratitude to my mentor and adviser, Dr. B. Srikrishna for his guidance, patience and encouragement throughout the development of this project. He was always available for stimulating discussions and provided the best support I could have desired. I am also indebted to him for the motivation and the freedom I enjoyed throughout. I also take this opportunity to thank the Intel lab administrator for providing me with an account in this wonderful research lab and the Intel lab scholars for providing a conducive atmosphere for research.

I would definitely like to thank Compo (Vinay) and Kakey (Ritesh) for helping me out with the theory, especially, during the earlier stages of this project. Working with them was such a pleasure. Finally, I would like to thank my parents and my brother, Sudhir who have been a constant source of support, advice and motivation over the last four years of my stay here at IIT Madras.

Place: Chennai [S Sunil]
Date:

Abstract

With novel features such as web browsing and online gaming being introduced into present-day mobiles, the downlink is definitely expected to carry significant traffic volumes in the near future. Even though orthogonal spreading codes are being used to distinguish between different users, multipath propagation and fading effects tend to destroy this orthogonality. The key idea in this report is to equalize the channel at the mobile receiver, thus approximately restoring the orthogonality among the different users and reducing the interference.

First, a theoretical model of the linear MMSE equalizer is presented. Linear Equalizers are known to work well for stationary channels. However, in most cases, the channel is time-varying in nature. In such cases, we need to incorporate adaptive equalization. To accommodate fading effects into the equalizer model, we present the adaptive MMSE equalizer. The adaptive equalizer is based on two standard converging algorithms - Least mean Squares and Recursive Least Squares.

The performance of the MMSE equalizer is compared against the performance of the RAKE filter in order to draw conclusions on its improvement in performance over the matched filter-based detectors. For stationary channels, the linear MMSE equalizer is shown to perform far superior compared to the RAKE receiver. The adaptive MMSE equalizer is first tested on static channels. Upon showing a considerable improvement in performance over the RAKE receiver, we test the same on fading channels, and the simulation studies prove that for low to moderate speeds (of upto 50 km/hr), adaptive equalization performs better compared to the RAKE receiver.

Table of Contents

List of Figures

CHAPTER 1

Introduction

1.1 Motivation

3G Wideband CDMA (3G W-CDMA), the third generation cellular standard, is based on direct sequence (DS) code division multiple access (CDMA), wherein each of the users uses the entire bandwidth allotted to him, but different users employ different spreading sequences. Each of the spreading code used is orthogonal with every other code, so that any user's signal does not interfere with the other users' signals. The primary traffic on present day cellular networks is voice and messages, so that the forward link or downlink (base-to-mobile) and the reverse link or uplink (mobile-to-base) carry similar traffic. With advanced features such as web browsing and online gaming being introduced in present-day mobiles, the downlink traffic is expected to increase significantly. The downlink is therefore expected to be the bottleneck in traditional cellular architectures, which currently allocate equal bandwidth resources to both the uplink and downlink. Even for the current cellular applications, the uplink has several advantages over the downlink: The base station transmits at a relatively higher power to mobiles that are far away, when compared to mobiles that are closer to the base station. Strong multipath components from such base stations could seriously impact the received signals of these closer mobiles. Thus, downlink power control could possibly lead to a near-far problem. Besides, the technology and receiver processing involved is definitely more sophisticated at the base station when than in a mobile. In view of enhancing the downlink, equalization appears to be a very attractive and convenient method, since the small number of interfering sources (the neighboring stations), together with the information available from the pilot signal, present some practical opportunities for improvement in performance.

The base station transmits the sum of the signals allotted for the different mobiles in the cell, with orthogonal spreading sequences assigned to different mobiles (users). The downlink is designed to be free of intra-cell interference under ideal conditions. However, the multipath propagation channel has a time-dispersive effect which de-

stroys the orthogonality among user codes, substantially degrading the performance of the matched filter based detectors. In the case of non-stationary channel conditions, fading effects also come into play. The transmitted signal arrives at the receiver from different paths with randomly time-varying nature. The mobility of the user also contributes significantly to the time-varying nature of the channel. This results in amplitude variations in the received signal, usually referred to as fading. Multipath fading is a significant limiting factor in the performance of existing mobile communication systems. The usual methods of multipath mitigation in CDMA (e.g."blind minimum output energy technique") rely on received signal cyclostationarity. In this application however, the scrambling code destroys the cyclostationarity and so an alternative means of multipath mitigation and interference cancellation is necessary. The key idea in this report is to equalize the channel at the mobile receiver, thus approximately restoring the orthogonality among the different users and reducing the interference.

In a given chip interval, the base station transmits the sum of the chips destined for each user, together with the chip for the pilot sequence. All elements in this sum observe the same channel, so that, a linear equalizer for the pilot sequence works for the desired user as well. The pilot sequence can thus play the role of a perpetual training sequence for adaptive equalization. The adaptation is based on the linear MMSE criterion. The idea of restoring orthogonality on the downlink through the use of equalization is especially useful since it leads to an adaptive receiver performing interference suppression even for CDMA systems with no spreading code periodicities.

The results of the MMSE adaptive equalizer is compared against that of the RAKE filter, in order to draw conclusions on the performance gain achieved by using the equalizer. Unlike the RAKE filter, which can be viewed as a matched filter, the MMSE equalizer is based on the exact computation of the MMSE solution and hence does not require any prior estimation of the channel parameters. In particular, the adaptive equalizer requires only coarse timing information in order to capture the energy from the desired chip in its observation window. Provided that the equalizer spans a sufficiently long observation window, the equalizer is robust against timing uncertainty and is capable of tracking timing variations.

1.2 Related Work

The idea of equalizing the downlink channel for the purpose of restoring orthogonality of the user spreading sequences was first proposed in [12]. A block equalizer is developed for a hybrid time division CDMA system. Equalization of the downlink for a DS-CDMA system was first independently introduced in [13], [14] and [15]. In [13], an adaptive MMSE equalizer using the pilot signal for tracking is presented, while [14] considers the zero-forcing equalizer. In [15], an MMSE solution for the downlink equalizer of a CDMA system is computed. The use of orthogonal spreading sequences, however, is not considered. An adaptive algorithm not requiring the pilot training sequence is presented in [16].

1.3 Organization of the Report

In *chapter 2*, some preliminary concepts in 3G W-CDMA transmission and reception are discussed. An insight into the Jakes model and the RAKE receiver is dealt with. *Chapter 3* deals with the Linear MMSE Equalizer. A theoretical model for the MMSE equalizer is derived. *Chapter 4* is fully dedicated to the Adaptive MMSE Equalizer. Convergence algorithms such as LMS and RLS are described and mathematically analyzed in detail. It is also seen as to how these algorithms could be applied to the case of an equalizer. In *chapter 5*, we present the simulation results and summarize the work done. We also discuss the further avenues that need to be explored.

CHAPTER 2

Preliminaries

2.1 3G W-CDMA (UMTS)

The Universal Mobile Telecommunications System (UMTS) is a visionary air interface standard that has evolved since late 1996 under the auspices of the European Telecommunications Standards Institute (ETSI). Around the turn of the century, several competing W-CDMA proposals agreed to merge into a single W-CDMA standard, which is now called UMTS. 3G systems promise unparalleled wireless access in ways that have never been possible before. Multi-megabit Internet access, communications using Voice over Internet Protocol (VoIP), voice-activated calls, unparalleled network capacity, and ubiquitous "always-on" access are just some of the advantages being touted by 3G developers.

The 3G W-CDMA standard has been designed for "always-on" packet-based wireless service, so that computers, entertainment devices and telephones may all share the same wireless network and be connected to the Internet, anytime, anywhere. W-CDMA will support high packet data rates, thereby allowing high quality data, multimedia, streaming audio, streaming video and broadcast-type services to consumers. W-CDMA provides public and private network features, as well as videoconferencing and *virtual home entertainment* (VHE).

W-CDMA requires a minimum spectrum allocation of 5 MHz, which is an important distinction from the other 3G standards. W-CDMA employs variable/selectable direct sequence spread spectrum chip rates that can exceed 16 megachips per second per user. With W-CDMA data rates from as low as 8 kbps to as high as 2 Mbps will be carried simultaneously on a single W-CDMA 5 MHz radio channel, and each channel will be able to support between 100 and 350 simultaneous voice calls at once, depending on antenna sectoring, propagation conditions, user velocity, and antenna polarizations.

2.2 HSDPA

High Speed Downlink Packet Access is a packet-based data service in W-CDMA downlink with data transmission up to 14.4 Mbps (and approximately twice this rate for MIMO systems) over a 5MHz bandwidth in W-CDMA downlink. Some of the features of HSDPA include Adaptive Modulation and Coding (AMC), Multiple-Input Multiple-Output (MIMO), Hybrid Automatic Request (HARQ), fast cell search, and advanced receiver design.

It is a concept within W-CDMA specifications whose main target is to increase user peak data rates and quality of service, and to generally improve spectral efficiency for downlink asymmetrical and bursty packet data services. When implemented, the HSDPA concept can co-exist on the same carrier as the Release'99 W-CDMA services, enabling a smooth and cost-efficient introduction of HSDPA into existing W-CDMA networks. Furthermore, a user can download packet-data over HSDPA, while at the same time having a speech call.

HSDPA is based on W-CDMA evolution, standardized as part of 3GPP Release 5 W-CDMA specifications. The new adaptive modulation and coding method of HSDPA, a fast scheduling function and fast retransmissions greatly improve its peak data rate and throughput, which enhances spectral efficiency. In addition to these benefits, users will perceive faster connections to services though shorter round trip times. As a result of these enhancements, operators using HSDPA will be able to support considerably higher numbers of high data rate users on a single radio carrier than is possible with any existing 3G technology. Information regarding 3G-CDMA and HSDPA is taken from [1].

2.3 System Model

2.3.1 Downlink Channel Model

The downlink channel model assumed consists of 3 main channels - CPICH, DPCH, HS-PDSCH and an interference simulator, OCNS. In the next few sections, we will briefly deal with each of these.

2.3.1.1 Common Pilot Channel (CPICH)

The CPICH is a fixed rate (30 kbps, SF = 256) downlink physical channel that carries a pre-defined bit sequence. The CPICH is broadcast over the entire cell. The pre-defined bit sequence as specified by the 3GPP standards [7] is $1 + i$. The same channelization code(SF = 256) is always used for the CPICH. Each radio frame is 10 ms long, each frame comprising 15 slots. *Figure 2.1* shows the frame structure of the CPICH. The pilot channel plays the role of a perpetual training sequence for tracking the channel.

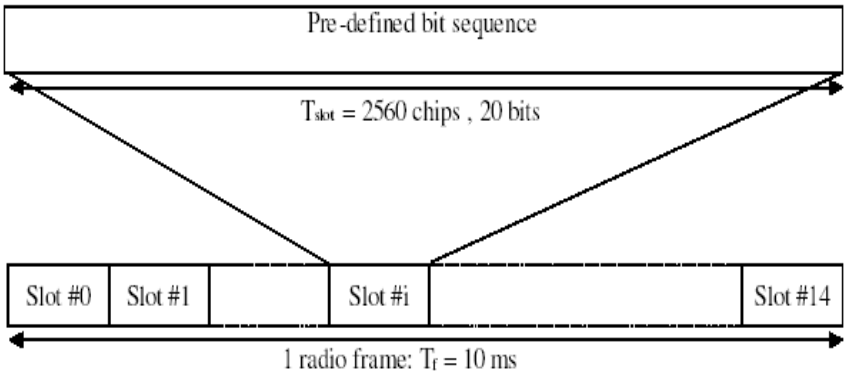

Figure 2.1: Frame Structure of the CPICH. [7]

2.3.1.2 Dedicated Physical Channel (DPCH)

The downlink DPCH can be seen as a time-multiplex of a downlink DPDCH and a downlink DPCCH. *Figure 2.2* shows the frame structure of the downlink DPCH. Each frame of length 10 ms is split into 15 slots, each of length $T_{slot} = 2560$ chips. The parameter k in the *Figure 2.2* determines the total number of bits per downlink DPCH slot. It is related to the spreading factor SF of the physical channel as $SF = \frac{512}{2^k}$. SF might thus range from 512 down to 4. The exact number of bits of the different downlink DPCH fields ($N_{pilot}, N_{TPC}, N_{TFCI}, N_{data1}, N_{data2}$) can vary depending on the spreading factor used. There are basically two types of DPCH - those that include TFCI (Transport Format Combination Indicator) (e.g. for several simultaneous services) and those that do not include TFCI (e.g. for fixed-rate services). It is the UTRAN (UMTS Terrestrial Radio Access Network) that determines if a TFCI should be transmitted and if it is mandatory for all UEs (User Equipments) to support the use of TFCI in the downlink.

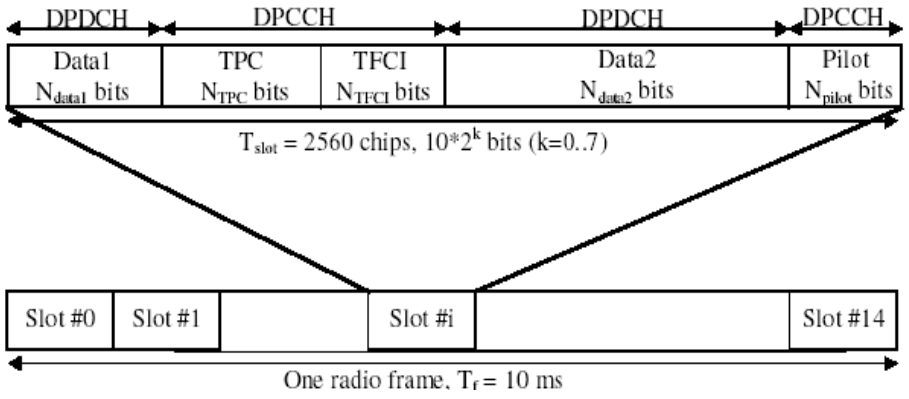

Figure 2.2: Frame Structure of the DPCH. [7]

2.3.1.3 High Speed Physical Downlink Shared Channel (HS-PDSCH)

The HS- PDSCH is used to carry the High Speed Downlink Shared Channel (HS-DSCH).

A HS-PDSCH corresponds to one channelization code of fixed spreading factor SF = 16 from the set of channelization codes reserved for HS-DSCH transmission. Multi-code transmission is allowed, which translates to UE being assigned multiple channelization codes in the same HS-PDSCH subframe, depending on its UE capability. *Figure 2.3* shows the slot and subframe structure of the HS-PDSCH. An HS-PDSCH may use QPSK or 16QAM modulation symbols. In *figure 2.3*, M is the number of bits per modulation symbols *i.e.* M = 2 for QPSK and M = 4 for 16QAM.

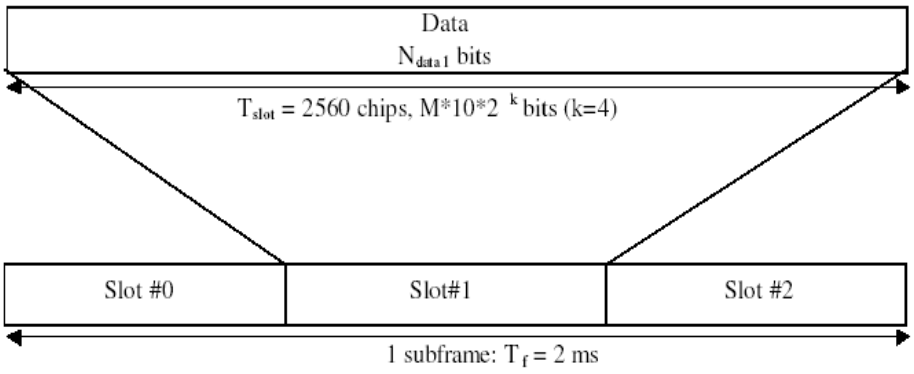

Figure 2.3: Subframe Structure of the HS-PDSCH. [7]

2.3.1.4 Orthogonal Channel Noise Simulator (OCNS)

It is a mechanism used to simulate the users or control signals on the other orthogonal channels of a downlink link. OCNS interference consists of 6 dedicated data channels as specified in *Table 2.1*. The selected codes are designed to have a single length-16 parent code.

Selected Channelization Codes at SF=128	Relative Power Level Setting (dB)
2	-6
3	-8
4	-8
5	-10
6	-7
7	-9

Table 2.1: OCNS definition for HSDPA receiver testing. [6]

2.3.1.5 Downlink Physical Channels connection set-up.

The downlink physical channels for HSDPA receiver testing for Single Link performance is given in *Table 2.2*

Physical Channel	Relative Setting of Ec	SF
CPICH	-10 dB	256
DPCH	-10 dB	128
HS-PDSCH	Test Specific	16
OCNS	Necessary power such that total transmit power spectral density (Ior) adds to one.	128

Table 2.2: Downlink physical channels set-up. [6]

2.3.1.6 Timing relationship between Physical Channels.

The CPICH is used as the timing reference for all the physical channels. The DPCH is aligned exactly with the CPICH. The HS-PDSCH is transmitted at an offset of $\tau = 1.33ms$ (2 time slots) or 5120 chips relative to the CPICH. A detailed description of the physical channels, their mapping and timing relationships can be found in [6] and [7].

2.3.2 Scrambling code

The scrambling code sequences are constructed by combining two real sequences into a complex sequence [8]. Each of the two real sequences are constructed as the position wise modulo 2 sum of 38400 chip segments of two binary m-sequences generated by means of two generator polynomials of degree 18. The resulting sequences thus constitute segments of a set of Gold sequences. The scrambling codes are repeated for every 10 ms radio frame. Let x and y be the two sequences respectively. The x sequence is constructed using the primitive (over GF(2)) polynomial $1 + X^7 + X^{18}$. The y sequence is constructed using the polynomial $1 + X^5 + X^7 + X^{10} + X^{18}$.

The m-sequences are constructed as follows :
Initial conditions:

- x is constructed with $x(0) = 1, x(1) = x(2) = \ldots = x(16) = x(17) = 0$.
- y is constructed with $y(0) = y(1) = \ldots = y(16) = y(17) = 0$

Recursive definition of subsequent symbols:

- $x(i + 18) = [x(i + 7) + x(i)]modulo(2)$, $i = 0 \ldots 2^{18} - 2$
- $y(i + 18) = [y(i + 10) + y(i + 7) + y(i + 5) + y(i)]modulo(2)$, $i = 0 \ldots 2^{18} - 2$

The n^{th} Gold sequence z_n, $n = 0, 1, 2 \ldots 2^{18} - 2$ is then defined as:

$$z_n(i) = [x(i + n)modulo(2^{18} - 1) + y(i)]modulo(2), i = 0 \ldots 2^{18} - 2 \tag{2.1}$$

These binary sequences are converted to real-valued sequences by the following transformation:

- $Z_n(i) = 1 - 2z_n(i)$

Finally, the n^{th} complex scrambling code is defined as:

$$S_{dl,n} = Z_n(i) + j(Z_n(i + 131072)modulo(2^{18} - 1)), i = 0, 1, \ldots 38399 \tag{2.2}$$

The pattern from phase 0 up to the phase of 38399 is repeated.

2.3.3 Pulse shaping

The transmit and receive pulse shaping filters are root raised cosines with roll-off $\alpha = 0.22$ [6].

$$RRC = \frac{\sin(\frac{\pi t}{T_c})(1-\alpha) + \frac{4\alpha t}{T_c}\cos(\frac{\pi t}{T_c})(1+\alpha)}{\frac{\pi t}{T_c}(1-\frac{4\alpha t}{T_c})^2} \tag{2.3}$$

where the roll-off factor $\alpha = 0.22$ and the chip duration is

$$T_c = \frac{1}{chiprate} \tag{2.4}$$

2.3.4 The Jakes Model

The Jakes fading model [9] is a deterministic method for simulating time-correlated Rayleigh fading waveforms and is still widely used today. The model assumes that N equal-strength rays arrive at a moving receiver with uniformly distributed arrival angles α_n, such that ray n experiences a Doppler shift $\omega_n = \omega_M \cos(\alpha_n)$, where $\omega_M = 2\pi f \frac{v}{c}$ is the maximum Doppler shift, v is the vehicle speed, f is the carrier frequency and c, the speed of light.

Using $\alpha_n = 2\pi\frac{n}{N}$, there is a quadrantal symmetry in the magnitude of the Doppler shift, except for angles 0 and π. As a result, the fading waveform can be modelled with $N_0 + 1$ complex oscillators, where $N_0 = \frac{(\frac{N}{2}-1)}{2}$. This gives the fading waveform as

$$T(t) = K\left\{\frac{1}{\sqrt{(2)}}[\cos(\alpha)+i\sin(\alpha)]\cos(\omega_M t+\theta_n)+\sum_{n=1}^{N_0}[\cos(\beta_n)+i\sin(\beta_n)]\cos(\omega_n t+\theta_n)\right\} \tag{2.5}$$

where K is a normalization constant, α (not the same as α_n) and β_n are phases, and θ_n are initial phases, usually set to zero. Setting $\alpha = 0$ and $\beta = \frac{\pi n}{N_0+1}$ gives zero cross correlation between the real and imaginary parts of $T(t)$.

To generate multiple uncorrelated waveforms, Jakes [4] suggests using phases $\theta_{n,j} = \beta_n + 2\pi\frac{(j-1)}{(N_0+1)}$, where $j = 1 \ldots N_0$ is the waveform index. However this gives almost uncorrelated waveforms j and k only when $\theta_{n,j} - \theta_{n,k} = i\pi + \frac{\pi}{2}$, for some integer i. Otherwise, the correlation between certain waveform pairs can be significant. A remedy for this problem is achieved by reformulating the Jakes model in terms of slightly different arrival angles.

To provide quadrantal symmetry for all Doppler shifts, which leads to equal power oscillators, the following arrival angles are used: $\alpha_n = 2\pi\frac{(n-0.5)}{N}$. This leads to the model:

$$T(t) = \sqrt{\frac{2}{N_0}}\left\{\sum_{n=1}^{N_0}[\cos(\beta_n) + i\sin(\beta_n)]\cos(\omega_n t + \theta_n)\right\} \tag{2.6}$$

where $N_0 = \frac{N}{4}$ and the normalization factor $\sqrt{\frac{2}{N_0}}$ gives $E(T(t)T^*(t)) = 1$.

2.3.5 The RAKE Receiver

In CDMA spread spectrum systems, the chip rate is typically much greater than the flat-fading bandwidth of the channel. CDMA spreading codes are designed to provide very low correlation between successive chips. Thus, propagation spread in the radio channel merely provides multiple versions of the transmitted versions of the signal at the receiver. if these multipath components are delayed in time by more than a chip duration, they appear like uncorrelated noise at a CDMA receiver. The spread spectrum processing gain makes uncorrelated noise negligible after despreading. However, since there is useful information in the multipath components, CDMA receivers may combine the time delayed versions of the original signal transmission in order to improve the SNR at the receiver. The RAKE receiver does exactly this.

It is a matched filter that uses a tap delay line to combine signals arriving over multipath propagation paths. In general, the RAKE receiver can be viewed as a filter matched to the convolution of the chip waveform with the channel, followed by the descrambling and despreading operations. In order to do this, the RAKE filter assumes that it has perfect knowledge of the channel coefficients and delays. The RAKE filter consists of several delay-taps (correlators) to separately detect the multipath components. The output of each tap is then separately weighed and all such weighed values are summed up to provide a better estimate of the transmitted signal.

CHAPTER 3

The Linear MMSE Equalizer

3.1 Introduction

An equalizer is usually a tapped FIR transversal filter, in which current and past values of the received signal are linearly weighted by the filter coefficients and summed to produce the output. Equalization compensates for Inter Symbol Interference (ISI) created by mutipath within time-dispersive channels. Equalization works on the principle of training and tracking. It is trained first with reference to a pre-known pilot signal. With the coefficients obtained after the training stage, the equalizer is used to track the input. [11] discusses equalization in detail.

Equalizers could be of chip-level or symbol-level, depending on whether they directly estimate chips or symbols respectively. The symbol-level equalizer is preferred over the other because of two reasons, the first one being that the equalizer would be implemented using the training signal at a higher SIR (Signal-to-Interference Ratio). Secondly, it reduces the time of computation and the complexity, since it estimates directly, the desired user's symbols instead of chips. Also, two implementations of the symbol-level equalizer are possible - the pre-despreading and the post-despreading implementations. For analytical purposes, it is convenient to consider the post-despreading implementation (*see Figure 3.1*) of the symbol level equalizer, where the despreading precedes equalization. The theoretical model of the linear MMSE equalizer is taken from [5].

Notation used:

- $L = L_1 + L_2 + 1$ is the length of the MMSE Equalizer.
- $\mathbf{h}^{(1)}(n) = [h^{(1)}(M_1), h^{(1)}(M_1 + 1), \ldots, h^{(1)}(M_2)]$ denotes the impulse response of the channel. Hence, the impulse response of the channel is assumed to have finite support on the interval $[M_1, M_2]$.
- $u^{(1)}(n)$ is the transmitted signal at time n from the base station.
- $\mathbf{u}^{(1)}(n) = [u^{(1)}(n - L_1 - M_2), u^{(1)}(n - L_1 - M_2 + 1), \ldots, u^{(1)}(n + L_2 - M_1)]$ is a vector of dimension $L + M_2 - M_1$.

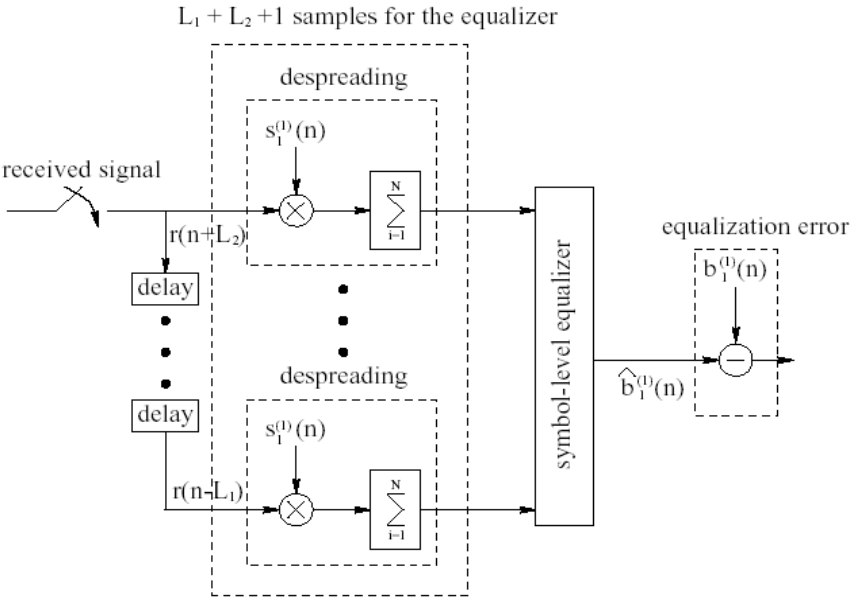

Figure 3.1: Conceptual diagram of the post-despreading implementation of the symbol-level equalizer. [5]

- $b_k(n)$ denotes the symbol sequence for mobile k expressed at chip rate. $b_k(n)$ is piecewise constant over symbol interval of length N chips, where N denoted the processing gain (N=Spreading Factor).
- $\mathbf{r}(n) = [r(n - L_1), \ldots, r(n-1), r(n), r(n+1), \ldots, r(n + L_2)]$ is the received block of samples used by the equalizer in order to estimate $u^{(1)}(n)$.
- $s_1^{(1)}$ is the desired user's spreading sequence.
- $\hat{u_1}^{(1)}(n)$ is the MMSE estimate of the desired user's signal at time n, $u_1^{(1)}(n)$ from the observation $r(n)$.

3.2 Theoretical Model for the MMSE Equalizer

The expression for the MMSE estimate(optimum Wiener filter coefficients) of the desired user's signal at time n, $\hat{u_1}^{(1)}(n)$, from the observation $r(n)$ is given by

$$\hat{u_1}^{(1)}(n) = (\mathbf{c}^{(1)}(n))^H \mathbf{r}(n) \tag{3.1}$$

where

$$(\mathbf{c}^{(1)}(n)) = (E_s\{\mathbf{r}(n)\mathbf{r}^H(n)\})^{-1}(E_s\{\mathbf{r}(n)u_1^{(1)}(n)^*\}) \tag{3.2}$$

is a vector of dimensions $L \times 1$, and E_s denotes the expectation over the user symbols and noise, conditioned on the user spreading sequences.

We first note that the received vector $\mathbf{r}(n)$ can be written in matrix form as

$$\mathbf{r}(n) = \mathbf{H}^{(1)}\mathbf{u}^{(1)}(n) + \mathbf{n}(n) \tag{3.3}$$

where $\mathbf{H}^{(1)}$ is a matrix of dimension $L \times (L + M_2 - M_1)$ of the form

$$\mathbf{H^{(1)}} = \begin{bmatrix} h^{(1)}(M_2) & h^{(1)}(M_2-1) & \cdots & h^{(1)}(M_1) & 0 & \cdots & 0 \\ 0 & h^{(1)}(M_2) & \cdots & h^{(1)}(M_1+1) & h^{(1)}(M_1) & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & h^{(1)}(M_2) & \cdots & h^{(1)}(M_1+1) & h^{(1)}(M_1) \end{bmatrix}$$

and $\mathbf{n}(n)$ is the additive white gaussian noise, which accounts for the interference from distant cells as well as the receiver thermal noise. The vector $\mathbf{n}$ also has dimensions $L \times 1$.

Let the length L vector $\mu_1^{(1)}(n)$ be defined as

$$\mu_1^{(1)}(n) \equiv E_s[\mathbf{r}(n)(u_1^{(1)}(n))^*]$$

If the user symbols are independent and identically distributed, so that

$$E[b_k(l)b_m(n)] = \delta(k-m)\delta(\lfloor l/N \rfloor - \lfloor n/N \rfloor)$$

where $\lfloor \cdot \rfloor$ denotes the floor operation, then the p^{th} element of the expectation vector $\mu_1^{(1)}(n)$ can be written as

$$\mu_{1,p}^{(1)}(n) = \sqrt{P_{(1)}^1} \sum_{l=M_1}^{M_2} h^{(1)}(l) s_1^{(1)}(n - L_1 + p - l)(s_1^{(1)}(n))^* \delta(\lfloor \frac{n - L_1 + p - l}{N} \rfloor - \lfloor \frac{n}{N} \rfloor) \tag{3.4}$$

for $0 \leq p \leq L_1 + L_2$. $P_j^{(1)}$ is the fraction of the total power allocated to the j^{th} user.

Let $\theta(n)$ denote the $L \times L$ correlation matrix, given by

$$\theta(n) \equiv E_s[\mathbf{r}(n)\mathbf{r}^H(n)] = \mathbf{H}^{(1)}\mathbf{U}^{(1)}(n)(\mathbf{H}^{(1)})^H + I_{oc}I_L$$

where

$$\mathbf{U}^{(1)}(n) = E_s[\mathbf{u}^{(1)}(n)(\mathbf{u}^{(1)}(n))^H]$$

and I_L is the $L \times L$ identity matrix.

The matrix $\mathbf{U}^{(1)}(n)$ is a square matrix of dimension $L + M_2 - M_1$, and has elements

$$U_{l,m}^{(i)}(n) = E_s[u^{(i)}(n - L_1 - M_2 + l)(u^{(i)}(n - L_1 - M_2 + m))^*] \tag{3.5}$$

for $0 \leq l, m \leq L_1 + L_2 + M_2 - M_1$ and 0 else. If as assumed above, the user symbol sequences are independent and identically distributed, the expectation in (3.5) can be computed as follows

$$E_s[u^{(1)}(m)u^{(1)}(n)^*] = \sum_{j=0}^{K^{(1)}} P_j^{(1)} s_j^{(1)}(m)(s_j^{(1)}(n))^* \delta(\lfloor \frac{m}{N} \rfloor - \lfloor \frac{n}{N} \rfloor) \tag{3.6}$$

Summarizing the above development, the MMSE equalizer for $u_1^{(1)}(n)$ from the observation $r(n)$ is given by

$$(\mathbf{c}^{(1)}(n)) = (E_s\{\mathbf{r}(n)\mathbf{r}^H(n)\})^{-1}(E_s\{\mathbf{r}(n)u_1^{(1)}(n)^*\}) = (\theta(n))^{-1}\mu_1^{(1)}(n) \tag{3.7}$$

From (3.7), we can see that the MMSE chip equalizer is computationally very complex and depends on the detailed parameters of the transmitted signal, the user and the base station. Besides, it needs to be computed at chip-rate, which makes it almost impossible to implement in practice. Hence, it is advisable to define an average MMSE chip equalizer, which will be substantially less complex and easier to adapt to varying conditions of the channel.

The Average MMSE Chip Equalizer is defined as

$$(\mathbf{c}^{(1)}(n)) = [E(E_s\{\mathbf{r}(n)\mathbf{r}^H(n)\})]^{-1}[E(E_s\{\mathbf{r}(n)u_1^{(1)}(n)^*\})] = \bar{\theta}^{-1}\bar{\mu}_1^{(1)} \tag{3.8}$$

By taking the expectations in (3.4) an (3.6), it is straightforward to show that

$$\bar{\mu}_1^{(1)} = \sqrt{P_1^1}h^{(1)} \tag{3.9}$$

where $\mathbf{h}^{(1)} = [h^{(1)}(-L_1), h^{(1)}(-L_1+1), \ldots h^{(1)}(L_2)]^T$, which is obtained by zero-padding the channel upto length L, and $\bar{\theta} = I_{or}\mathbf{H}^{(1)}(\mathbf{H}^{(1)})^H + I_{oc}I_L$.

The average MMSE chip equalizer can be calculated directly from the knowledge of the channel $h^{(1)}$, from which the channel matrix $\mathbf{H}^{(1)}$ can be computed, the sum powers received from the serving and interfering base station, I_{or} and I_{oc}. As a result, the complexity associated with direct computation of the average MMSE chip equalizer is not unreasonable.

For obtaining the average MMSE symbol level equalizer, minor changes need to be carried out. First, let $\mathbf{z}_1^{(1)}(j)$ denote the input vector to the equalizer corresponding to the j-th symbol of the desired user. The l-th element of this vector is given by

$$\mathbf{z}_1^{(1)}(j) = \sum_{m=0}^{N-1} \mathbf{r}(jN - L_1 + l + m)(\mathbf{s}_1^{(1)}(jN+m))^*$$

where $0 \leq l \leq L_1 + L_2$. Using the matrix notation defined in (3.3), vector $\mathbf{z}_1^{(1)}(j)$ can be expressed in a compact form as follows

$$\mathbf{z}_1^{(1)}(j) = \mathbf{H}^{(1)} \sum_{m=0}^{N-1} u^{(1)}(jN+m)(\mathbf{s}_1^{(1)}(jN+m))^* + \sum_{m=0}^{N-1} \mathbf{n}(jN+m)(\mathbf{s}_1^{(1)}(jN+m))^* \quad (3.10)$$

Let $\mathbf{d}_1^{(1)}(j)$ denote the MMSE symbol-level equalizer for symbol $b_1^{(1)}(jN)$, which is given by

$$\mathbf{d}_1^{(1)}(j) = (E_s[\mathbf{z}_1^{(1)}(j)(\mathbf{z}_1^{(1)}(j))^H])^{-1} E_s[\mathbf{z}_1^{(1)}(j)(b_1^{(1)}(jN))]^* \equiv (\tau_1^{(1)}(j))^{-1} \nu_1^{(1)}(j) \quad (3.11)$$

where $\tau_1^{(1)}(j)$ and $\nu_1^{(1)}$ are defined implicitly. The MMSE estimate of the desired symbol $b_1^{(1)}(jN)$ is given by the inner product $\langle \mathbf{d}_1^{(1)}(j), \mathbf{z}_1^{(1)}(j) \rangle$. A time-invariant average symbol-level equalizer is obtained by performing averaging of the terms in (3.11) over the user's spreading sequences. The average symbol-level equalizer $\mathbf{d}_1^{(1)}$ is given by

$$\mathbf{d}_1^{(1)} = (E[\tau_1{}^{(1)}(j)])^{-1} E[\nu_1^{(1)}(j)] = (\bar{\tau}_1^{(1)})^{-1} \bar{\nu}_1^{(1)} \quad (3.12)$$

The following are the properties of the average symbol-level equalizer. [5]

- The average symbol-level equalizer is the same, within a positive multiplicative constant, for the random and orthogonal spreading models.
- Analogous to the average chip equalizer, within a positive multiplicative constant, the average symbol-level equalizer is the same for all users transmitted from the same base station.
- Under both spreading models, the average symbol-level and chip-level equalizers are equal, within a positive multiplicative constant. More generally, the average symbol-level equalizer is the same regardless of the number of chips combined in the post-despreading implementation.

3.3 Adaptive Implementation

The theoretical model presented for the linear MMSE Equalizer works well for channels that do not change appreciably over time. However, in most cases, the channel is time-varying in nature. In such cases, we need to incorporate adaptive equalization. Adaptive equalizers are non-linear equalizers that work on standard converging algorithms. In such equalizers, the output is tracked with reference to the desired response. The working of such equalizers is mostly iterative in nature.

CHAPTER 4

The Adaptive MMSE Equalizer

As described in the previous chapters, the linear MMSE equalizer may work well only for channels that are static in nature. For fast-changing (fading) channels, an adaptive method needs to be implemented in the working of the equalizer. *Figure 4.1* shows the block diagram for such a concept. In adaptive equalizers, the input is tracked to a desired response, using a feedback system. The error obtained in estimation after each iteration is fed back to the system, in an attempt to reduce the error and hence obtain convergence. In this chapter, we describe adaptive equalizers, based on two important algorithms - Least Mean Squares (LMS) and Recursive Least Squares (RLS). The theory behind RLS and LMS algorithms and the mathematics involved, which are explained in the subsequent sections are adopted from [3].

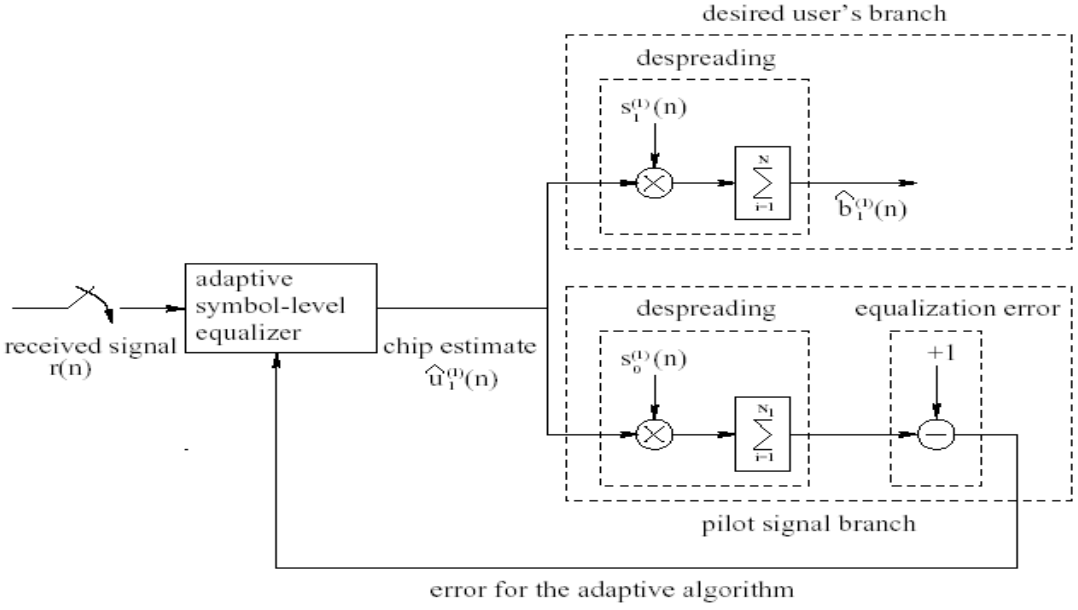

Figure 4.1: Conceptual diagram of the post-despreading implementation of the adaptive symbol-level equalizer. [5]

4.1 LMS Adaptive Filters

The LMS algorithm is an important member of the family of *stochastic gradient algorithms*. The term "stochastic gradient" is intended to distinguish the LMS algorithm from the method of steepest descent, which uses a deterministic gradient in a recursive computation of the Wiener filter for stochastic inputs. An important feature of the LMS is its simplicity. Moreover, it does not require measurements of the relevant correlation functions, nor does it require matrix inversion. Hence, it is memory-efficient. However, the convergence of LMS is often slow compared to the other standard algorithms. Besides, LMS is susceptible to problems with noise during periods when the input fails to excite the system. There exist various forms of the LMS algorithm, as described in [10].

4.1.1 Overview of the LMS Adaptation algorithm

During the LMS filtering process, the *desired response* $d(n)$ is supplied for processing, alongside the tap-input vector $\mathbf{u}(\mathbf{n})$. Given this input, the transversal filter with coefficients $\mathbf{h}(\mathbf{n})$, produces an output $(\hat{d}(n)|\mathbf{u}(n))$ used as an estimate of the desired response $d(n)$. Accordingly, we may define an estimation error $e(n)$ as the difference between the desired response and the actual filter output. We define $J(n) = |e(n)|^2 = e(n)e^*(n)$ as the cost function. With every iteration, the estimates of the tap weights $\mathbf{h}(n)$ are updated, according to the information obtained from the *gradient vector*, $\nabla J(n)$, the differential of the cost function $J(n)$, w.r.t the tap weights. The updating is done in a direction opposite the gradient. μ is denoted as the *step-size parameter*, which actually determines the rate of convergence.

If it were possible to make exact measurements of the gradient vectors $\nabla J(n)$ at each iteration n, and if the step-size parameter is suitably chosen, then the tap-weight vector computed by using the steepest-descent algorithm would indeed converge to the Optimum Wiener filter solution. In reality, however, exact measurements of the gradient vector are not possible, since that would require prior knowledge of both the correlation matrix $\mathbf{R}$ of the tap inputs and the cross-correlation vector $\mathbf{p}$ between the tap inputs and the desired response. Consequently, the gradient vector must be estimated from the available data when we operate in an unknown environment. The gradient vector can be written in terms of the correlation matrix $\mathbf{R}$ and the cross-correlation matrix $\mathbf{p}$ as

$$\nabla J(n) = -2\mathbf{p} + 2\mathbf{R}\mathbf{h}(n) \tag{4.1}$$

Since $\mathbf{p}$ and $\mathbf{R}$ are not known before hand, it is best to use instantaneous estimates for $\mathbf{R}$ and $\mathbf{p}$ and develop an instantaneous estimate of the gradient vector. $\hat{\mathbf{R}}(n)$ is the instantaneous estimate of the correlation matrix and is given by $\hat{\mathbf{R}}(n) = \mathbf{u}(n)\mathbf{u}^H(n)$. Likewise $\hat{\mathbf{p}}(n)$ is given by $\hat{\mathbf{p}}(n) = \mathbf{u}(n)d^*(n)$. These give

$$\hat{\nabla}J(n) = -2\mathbf{u}(n)d^*(n) + 2\mathbf{u}(n)\mathbf{u}^H(n)\hat{\mathbf{h}}(n) \tag{4.2}$$

Substituting the estimate of (4.2) for the gradient vector in the steepest descent algorithm[3], we obtain the recursive relation for updating the tap-weight vector:

$$\hat{\mathbf{h}}(n+1) = \hat{\mathbf{h}}(n) + \mu\mathbf{u}(n)[d^*(n) - \mathbf{u}^H(n)\hat{\mathbf{h}}(n)]^* \tag{4.3}$$

4.1.2 LMS algorithm applied to the Equalizer case

Let $\mathbf{u}(n)$ be the input vector to the LMS-based equalizer at time n. Correspondingly, let $\mathbf{h}(n) = [h(n-L_1)\dots h(n)\dots h(n+L_2)]$ be the L equalizer coefficients.
The output of the equalizer for symbol n (after descrambling and despreading) is given by

$$op(n) = \sum_{m=n}^{n+SF-1} \sum_{k=-L_1}^{L_2} u(m-k)h(k)c^*(m) \tag{4.4}$$

where $c(m)$ is the corresponding scrambling code. The despreading is done over the all-ones CPICH code ($SF = 256$). Hence, it is expected to get back $1 + i$ at the receiver, which then forms our $d(n)$. Hence, we can define the error-term $e(n)$ as

$$e(n) = d(n) - \sum_{m=n}^{n+SF-1} \sum_{k=-L_1}^{L_2} u(m-k)h(k)c^*(m)$$

Consequently, the gradient vector can be defined as

$$\nabla J(n) = \frac{\partial}{\partial h(k)} J(n) = -2 \times \left[\sum_{SF=m}^{m+256} u(m-k)\right] \cdot e(n) \tag{4.5}$$

The update equation for the LMS filter (i^{th} iteration) is given by

$$h_{i+1}(k) = h_i(k) + \mu\left[\sum_{SF=m}^{m+256} u(m-k)\right] \cdot e^*(n); -L_2 \leq k \leq L_1 \tag{4.6}$$

μ is the parameter that actually decides the rate of convergence as well as the steady state error. Higher the step-size value μ, relatively faster is the convergence, but the steady state error would also be high, and correspondingly for a low value

of μ, we have the opposite. Hence, it is required to appropriately choose μ, owing to a trade-off between the rate of convergence and the steady-state error. Also, for the stability of the system, $0 \leq \mu \leq \frac{2}{LS_{max}}$, where S_{max} is the maximum value of the power spectral density of the tap inputs $u(n)$ and the filter length L is moderate to large.

4.2 RLS Adaptive Filters

4.2.1 Overview of the RLS algorithm

In recursive implementation of the method of least squares, we start the computation with prescribed initial conditions and use the information contained in new data samples to update the old estimates. Thus, the length of observable data can be varied. We express the cost function to be minimized as

$$J(n) = \sum_{i=1}^{n} \beta(n,i)|e(i)|^2 \tag{4.7}$$

where $\beta(n,i)$ is defined as the weighting factor, and $e(n)$ defined as in the LMS case. The weighting factor $\beta(n,i)$ lies between 0 and 1. The use of the weighing factor in general, is intended to ensure that data in the distant past are "forgotten" in order to afford the possibility of following the statistical variations of the observable data when the filter operates in a non stationary environment. The weighting factor $\beta(n,i)$ is defined in terms of another parameter λ as

$$\beta(n,i) = \lambda^{n-i} \tag{4.8}$$

where λ is a positive constant close to, but less than unity. When $\lambda = 1$, we have the ordinary method of least squares. The inverse of $1 - \lambda$ is, roughly speaking, a measure of the *memory* of the algorithm. The special case $\lambda = 1$ corresponds to *infinite memory*.

The LMS algorithm as well as the method of least squares draw problems due to the fact that they cannot correctly map the input onto the output. To make the estimation problem "well-posed", some form of prior information about the input-output mapping is necessary. This in turn, means that the formulation of the cost function must be expanded to take prior information into account. We expand the cost function as the sum of two components as:

$$J(n) = \sum_{i=1}^{n} \lambda^{n-i}|e|^2 + \delta\lambda^n|\mathbf{h}(n)|^2 \tag{4.9}$$

- *The sum of weighted error squares*

$$\sum_{i=1}^{n} \lambda^{n-i}|e|^2 = \sum_{i=1}^{n} \lambda^{n-i}|d(n) - \mathbf{h}^H(n)\mathbf{u}(i)|^2,$$

  which is data dependent.

- *A regularizing term*

$$\delta\lambda^n|\mathbf{h}(n)|^2 = \delta\lambda^n\mathbf{h}^H(n)\mathbf{h}(n)$$

  where δ is a positive real number called the *regularization parameter*. The regularization parameter is under the designer's control. It should be set to a small value for a high prevalent SNR conditions (of the order of 10 dB) and to a large value for small SNR conditions.

We find that the effect of including the regularizing term $\delta\lambda^n|\mathbf{h}(n)|^2$ to a change in the formulation of the time-averaged correlation matrix $\mathbf{R}$ and the cross-correlation matrix $\mathbf{p}$ as well. The modified equation are:

$$\mathbf{R}(n) = \sum_{i=1}^{n} \lambda^{n-i}\mathbf{u}(i)\mathbf{u}^H(i) + \delta\lambda^n I \tag{4.10}$$

where I is the $L \times L$ identity matrix. $\mathbf{p}$ is written as:

$$\mathbf{p}(n) = \sum_{i=1}^{n} \lambda^{n-i}\mathbf{u}(i)d^*(i) \tag{4.11}$$

For obtaining recursive computations of $\mathbf{R}(n)$ and $\mathbf{p}(n)$, we may write

$$\mathbf{R}(n) = \lambda\left[\sum_{i=1}^{n-1} \lambda^{n-1-i}\mathbf{u}(i)\mathbf{u}^H(i) + \delta\lambda^{n-1}I\right] + \mathbf{u}(n)\mathbf{u}^H(n), \tag{4.12}$$

from which we have:

$$\mathbf{R}(n) = \lambda\mathbf{R}(n-1) + \mathbf{u}(n)\mathbf{u}^H(n) \tag{4.13}$$

Similarly, we may derive the following recursion for updating the cross-correlation vector between the tap inputs and the desired response.

$$\mathbf{p}(n) = \lambda\mathbf{p}(n-1) + \mathbf{u}(n)d^*(n) \tag{4.14}$$

We now define the *The Matrix Inversion Lemma* [3] which will help us find the inverse of the correlation matrix $\mathbf{R}$.

4.2.1.1 The Matrix Inversion Lemma (MIL)

Let A and B be two positive-definite $L \times L$ matrices related by

$$A = B^{-1} + CD^{-1}C^H,$$

where D is a positive-definite $N \times L$ matrix and C is an $L \times N$ matrix. According to the MIL,

$$A^{-1} = B - BC(D + C^H BC)^{-1}C^H B. \tag{4.15}$$

We assume that the correlation matrix $\mathbf{R}(n)$ is non-singular and make the following identifications:

$$A = \mathbf{R}(n); B^{-1} = \lambda \mathbf{R}(n-1); C = \mathbf{u}(n); D = 1$$

Using the above in the MIL, we obtain the recursive equation for the inverse of the correlation matrix as:

$$\mathbf{R}^{-1}(n) = \lambda^{-1}\mathbf{R}^{-1}(n-1) - \frac{\lambda^{-2}\mathbf{R}^{-1}(n-1)\mathbf{u}(n)\mathbf{u}^H(n)\mathbf{R}^{-1}(n-1)}{1 + \lambda^{-1}\mathbf{u}^H(n)\mathbf{R}^{-1}(n-1)\mathbf{u}(n)} \tag{4.16}$$

For convenience of computation, let

$$\mathbf{P}(n) = \mathbf{R}^{-1}(n) \tag{4.17}$$

and

$$\mathbf{k}(n) = \frac{\lambda^{-1}\mathbf{P}(n-1)\mathbf{u}(n)}{1 + \lambda^{-1}\mathbf{u}^H\mathbf{P}(n-1)\mathbf{u}(n)} \tag{4.18}$$

We may write

$$\mathbf{P}(n) = \lambda^{-1}\mathbf{P}(n-1) - \lambda^{-1}\mathbf{k}(n)\mathbf{u}^H(n)\mathbf{P}(n-1) \tag{4.19}$$

By rearranging (4.18), we obtain

$$\mathbf{k}(n) = \left[\lambda^{-1}\mathbf{P}(n-1) - \lambda^{-1}\mathbf{k}(n)\mathbf{u}^H(n)\mathbf{P}(n-1)\right]\mathbf{u}(n) \tag{4.20}$$

Therefore we see that

$$\mathbf{k}(n) = \mathbf{P}(n)\mathbf{u}(n) = \mathbf{R}^{-1}(n)\mathbf{u}(n) \tag{4.21}$$

Next, we need to develop equations for updating the tap weight vectors. As done in the LMS case, we shall first write, using (4.14) and (4.17)

$$\hat{\mathbf{h}}(n) = \mathbf{R}^{-1}(n)\mathbf{p}(n) = \lambda\mathbf{P}(n)\mathbf{p}(n-1) + \mathbf{P}(n)\mathbf{u}(n)d^*(n) \tag{4.22}$$

Using (4.19) for the first term only on the right hand side, we have on simplification

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) - \mathbf{k}(n)\mathbf{u}^H(n)\hat{\mathbf{h}}(n-1) + \mathbf{P}(n)\mathbf{u}(n)d^*(n) \tag{4.23}$$

Finally, using (4.21), we get the desired recursive equation for updating the tap coefficients.

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mathbf{k}(n)[d^*(n) - \mathbf{u}^H(n)\hat{\mathbf{h}}(n-1)] = \hat{\mathbf{h}}(n-1) + \mathbf{k}(n)\xi^*(n) \tag{4.24}$$

$\xi(n) = d(n) - \hat{\mathbf{h}}^H(n-1)\mathbf{u}(n)$ is the *a priori estimation error*. This, in general differs from the *a posteriori estimation error* $e(n) = d(n) - \hat{\mathbf{h}}^H(n)\mathbf{u}(n)$.

4.2.2 RLS Algorithm applied to the Equalizer case

We assume the same notation as used earlier for the LMS case. We first initialize the tap coefficients $\mathbf{h}$, of dimensions $L \times 1$ to the all-zero vector, and the $L \times L$ vector $\mathbf{P}$ to $\delta^{-1}I$ *i.e.* $\mathbf{P}(0) = \delta^{-1}I$.

For each instant of time $n = 1, 2, \ldots$, we compute the following,

$$\pi(n) = \mathbf{P}(n-1)\mathbf{s}(n) \tag{4.25}$$

where

$$s(n,k) = \sum_{m=n}^{n+SF-1} u(m-k)c^*(m); -L_2 \leq k \leq L_1$$

is a vector of dimensions $L \times 1$, obtained after descrambling and despreading.

$$\mathbf{k}(n) = \frac{\pi(n)}{\lambda + \mathbf{s}^H(n)\pi(n)} \tag{4.26}$$

$$\xi(n) = d(n) - \hat{\mathbf{h}}^H(n-1)\mathbf{s}(n) \tag{4.27}$$

is the a priori estimation error. Finally, we have

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mathbf{k}(n)\xi^*(n) \tag{4.28}$$

and

$$\mathbf{P}(n) = \lambda^{-1}\mathbf{P}(n-1) - \lambda^{-1}\mathbf{k}(n)\mathbf{s}^H(n)\mathbf{P}(n-1) \tag{4.29}$$

(4.28) is the RLS update equation and (4.29) is the equation for updating $\mathbf{P}$ each time. It is usually seen that the RLS algorithm has a greater rate of convergence (typically an order of magnitude faster) compared to the LMS algorithm. However, this is traded-off with the computational complexity involved in this algorithm. This kind of robust behaviour is explained by the fact that, in deriving the update formula for the weight vector of the LMS filter, no assumptions are made on the statistical characterization of the input data.

CHAPTER 5

Simulation Results

5.1 Results for the Linear MMSE Equalizer

We first present results for the linear symbol-level MMSE equalizers, introduced in *Chapter 3*. The performance of the linear MMSE Equalizer is compared against the performance of the RAKE filter (which assumes perfect knowledge of channel coefficients). The results are plotted for different multipath delays and G values. G is defined to be $\frac{I_{or}}{I_{oc}}$. First, we consider a single path. Following this, the multipath is assumed to have 2 paths with equal powers. Finally, we consider the case wherein, the multipath is assumed to be a 4-path equal-powered one. The length of the equalizer is assumed to be 14 for each case. The performance of the analytically computed MMSE equalizer against the RAKE filter is shown in *Figure 5.1*. Clearly, the performance of the linear MMSE equalizer surpasses that of the RAKE filter, over the range of interest.

5.2 Results for the Adaptive MMSE Equalizer

The adaptive MMSE filters (based on RLS and LMS) are first tested on static channels. For the LMS-based equalizer, we test for the step-size parameter μ varying from 0.002 to 0.2. Correspondingly, for the RLS-based equalizer, forgetting factors(λ) varying from 0.7 to 0.99 are used. Two values of G, 1 (0 dB) and 10 (10 dB) are considered. The multipath is assumed to have 4 paths (1 direct path and 3 delayed versions with delays 1, 2 and 3 symbols respectively). Also, two cases of multipath (with equal and unequal powers) are considered. In the first case, the channel is represented as $[\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}]^T$, while in the second case, the channel is $[\frac{2\sqrt{2}}{\sqrt{15}}, \frac{2}{\sqrt{15}}, \frac{\sqrt{2}}{\sqrt{15}}, \frac{1}{\sqrt{15}}]^T$. The plots for the mean squared errors, averaged out after 35 simulations each, (establishing convergence of these algorithms) are shown in *Figures 5.2-5.9*.

Finally, we present the simulation results for the fading channel. Here, we work on only the RLS-based adaptive equalizers, as the LMS-based equalizer was not found to perform significantly better than the RAKE receiver. The channel is taken to be $[\frac{2\sqrt{2}}{\sqrt{15}}, \frac{2}{\sqrt{15}}, \frac{\sqrt{2}}{\sqrt{15}}, \frac{1}{\sqrt{15}}]^T$. After looking at the plots for the static channel case, we choose

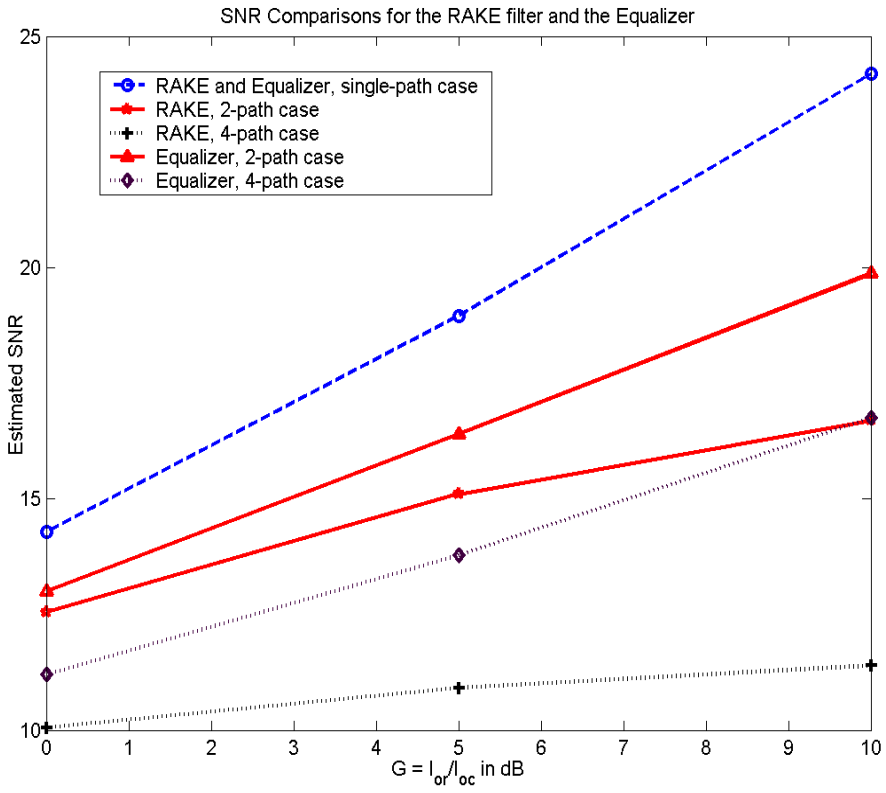

Figure 5.1: Comparison of the performances of the RAKE filter and the Equalizer for different G and multipath parameters.

a suitable value of λ which would enable quick convergence for low to moderate speeds as well. We choose $\lambda = 0.9$. The only parameter that is changed in this case is the vehicle speed. We work at 3 different speeds - 10, 30 and 50 km/hr. We work at $G = 10dB$ for both the cases. The plots for the mean-squared error for the three different cases are shown in *Figures 5.10-5.12*. In order to show the improvement in performance of the adaptive RLS equalizer over the RAKE filter, we plot the histogram of the estimated SNR values for the equalizer and the RAKE filter for $\lambda = 0.99$ and speed = 10 km/hr. The histogram plots are shown in *Figure 5.13*.

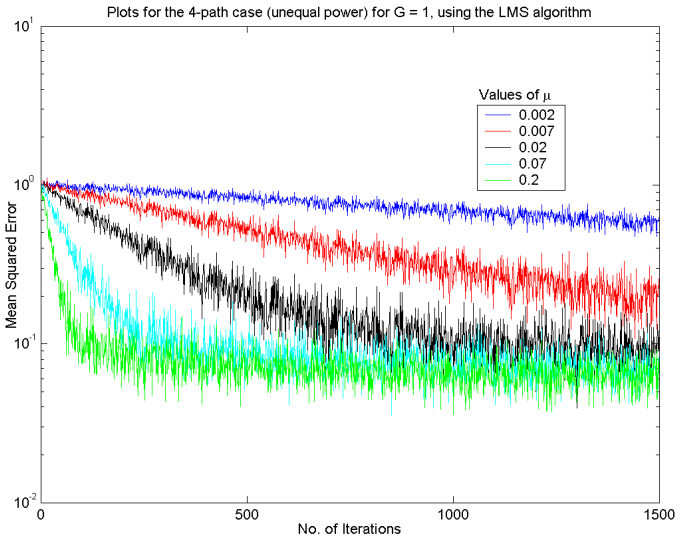

Figure 5.2: Plot of the mean-squared error for the 4-path case (Unequal power), for $G = 1$, using the LMS algorithm.

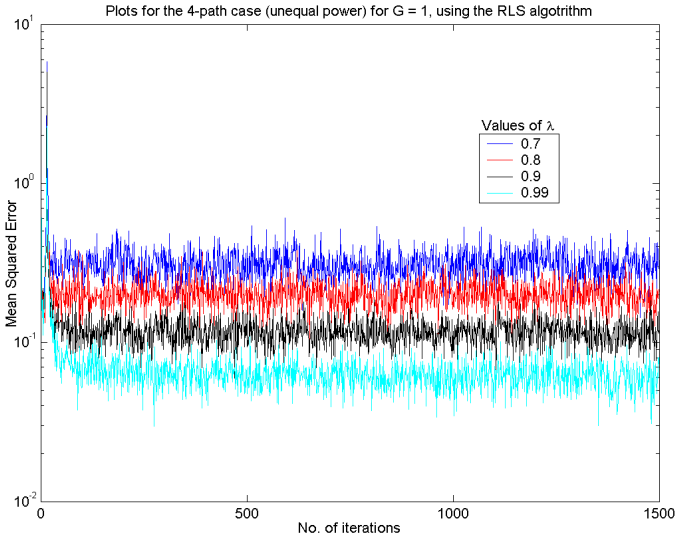

Figure 5.3: Plot of the mean-squared error for the 4-path case (Unequal power), for $G = 1$, using the RLS algorithm.

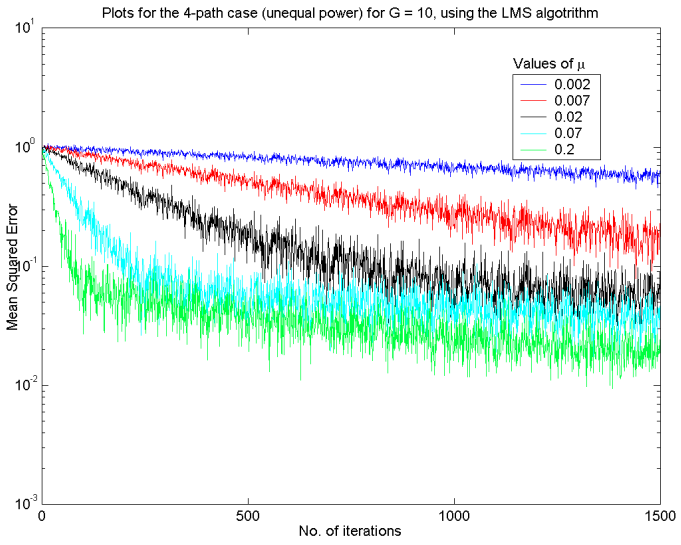

Figure 5.4: Plot of the mean-squared error for the 4-path case (Unequal power), for $G = 10$, using the LMS algorithm.

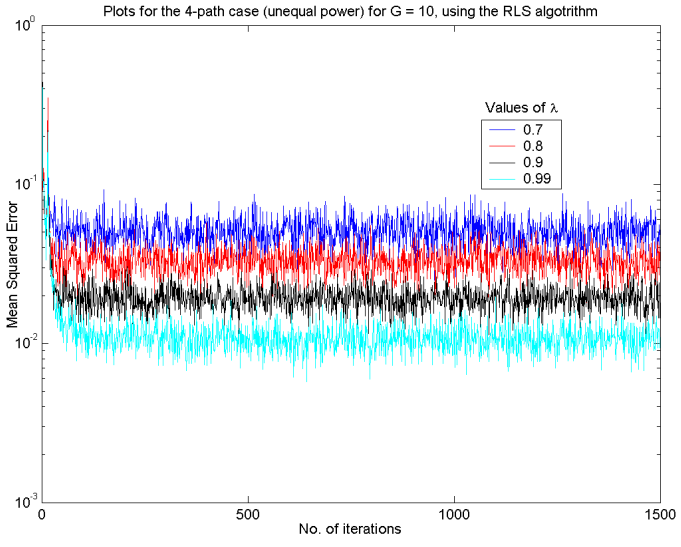

Figure 5.5: Plot of the mean-squared error for the 4-path case (Unequal power), for $G = 10$, using the RLS algorithm.

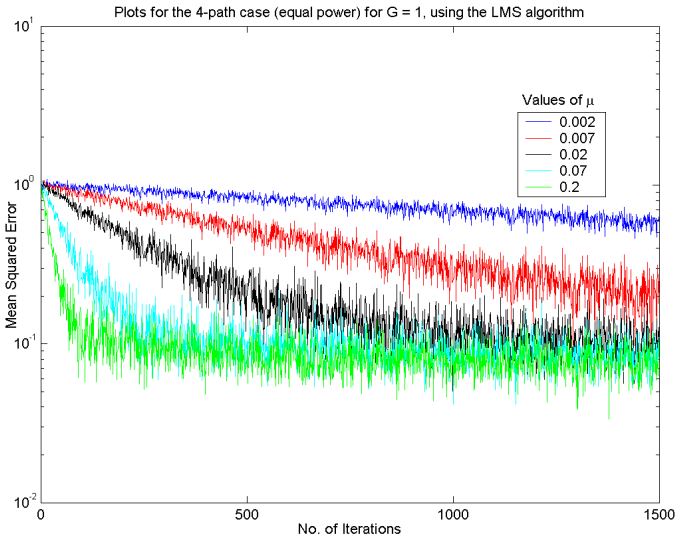

Figure 5.6: Plot of the mean-squared error for the 4-path case (Equal power), for $G = 1$, using the LMS algorithm.

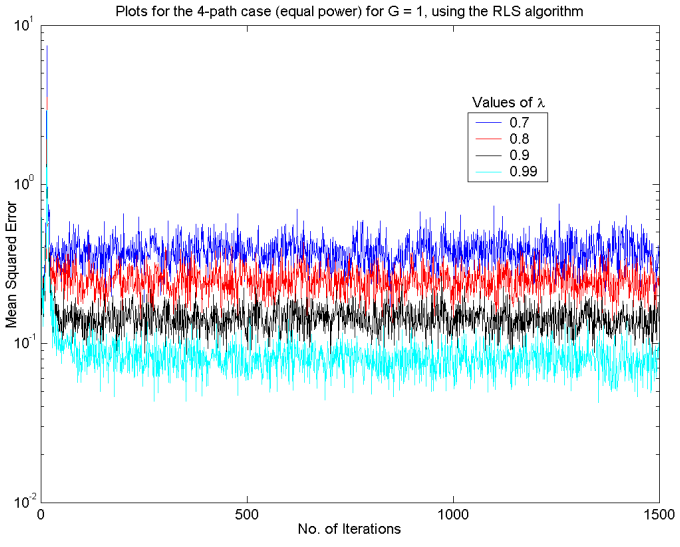

Figure 5.7: Plot of the mean-squared error for the 4-path case (Equal power), for $G = 1$, using the RLS algorithm.

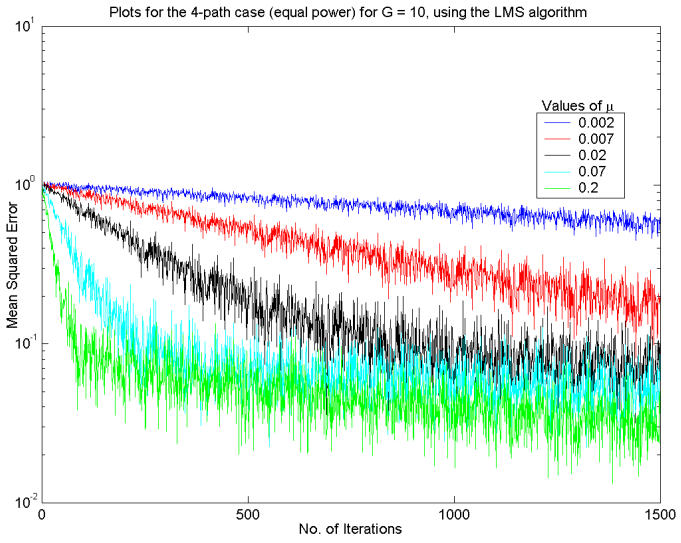

Figure 5.8: Plot of the mean-squared error for the 4-path case (Equal power), for $G = 10$, using the LMS algorithm.

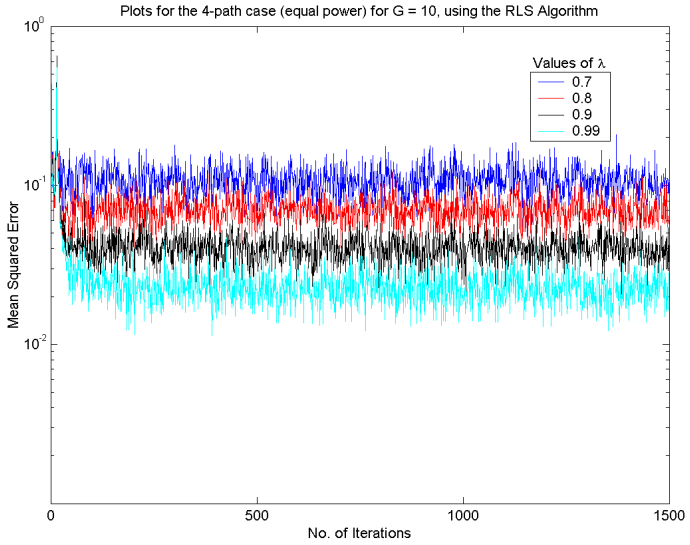

Figure 5.9: Plot of the mean-squared error for the 4-path case (Equal power), for $G = 10$, using the RLS algorithm.

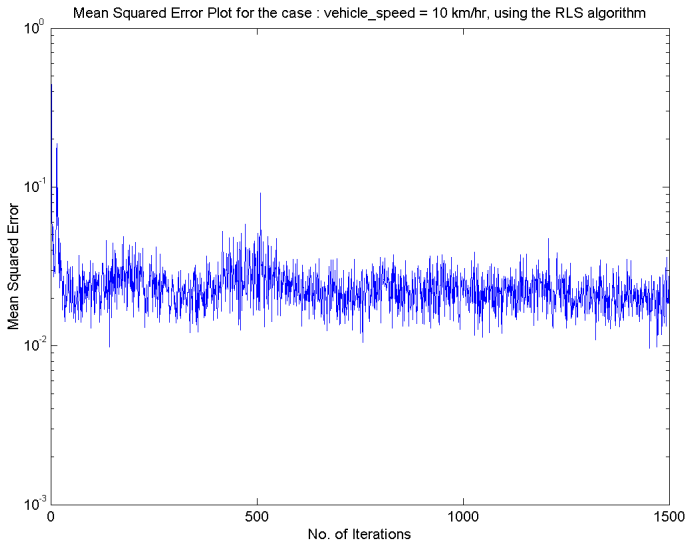

Figure 5.10: Plot of the mean-squared error for vehicle speed of 10 km/hr, using the RLS algorithm.

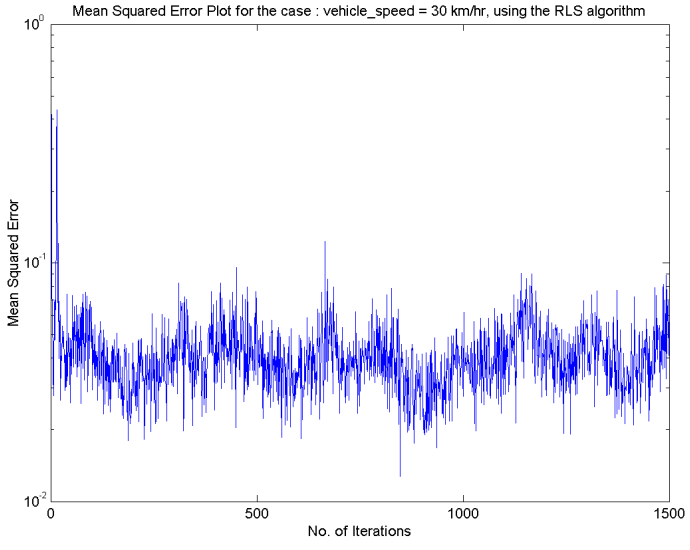

Figure 5.11: Plot of the mean-squared error for vehicle speed of 30 km/hr, using the RLS algorithm.

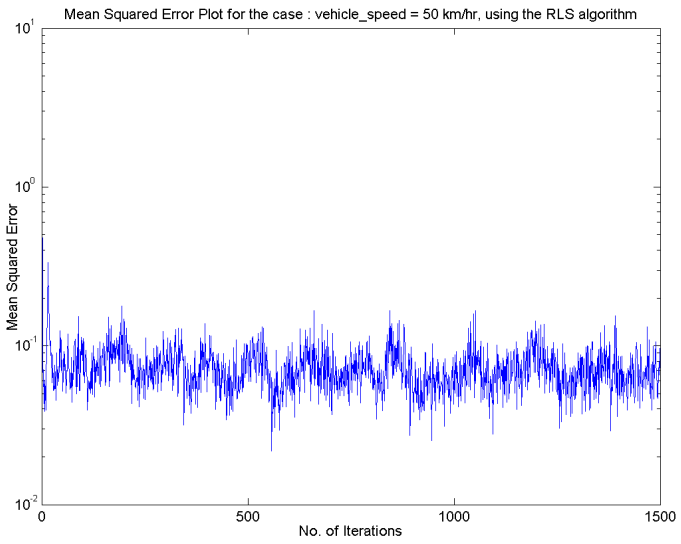

Figure 5.12: Plot of the mean-squared error for vehicle speed of 50 km/hr, using the RLS algorithm.

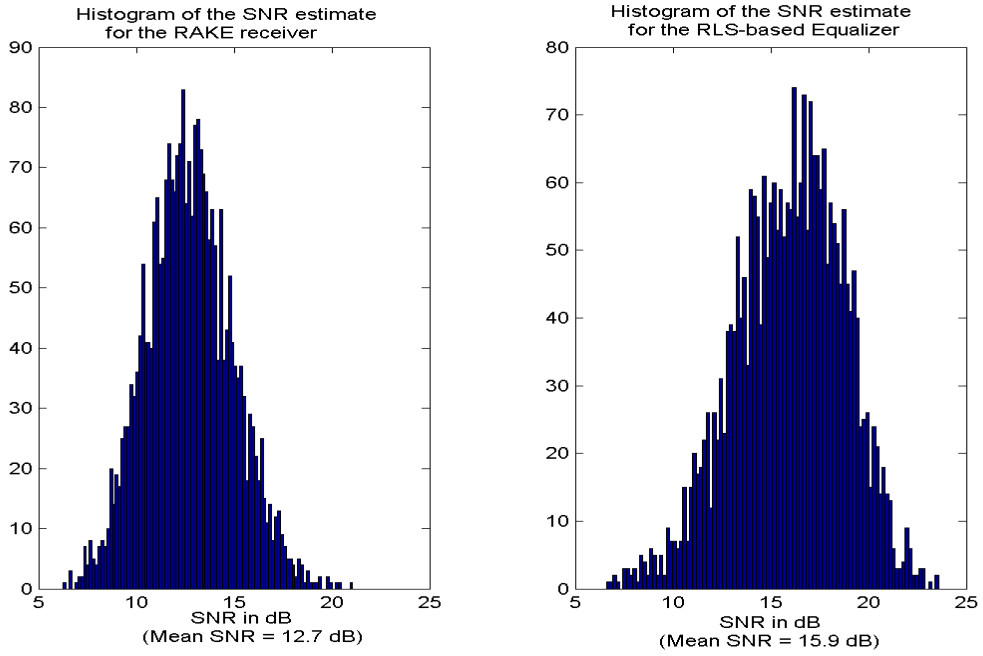

Figure 5.13: Histogram plots of the estimated SNR values for the RAKE receiver and the Adaptive MMSE equalizer at vehicle speed = 10 km/hr and G = 10.

CHAPTER 6

Conclusions and Future Work

6.1 Conclusions

The MMSE symbol-level equalizer model presented in this report can significantly improve the performance of the downlink of 3G W-CDMA systems. Also, it is computationally less complex and would be implemented at a higher SIR. It reduces intra-cell interference, thus automatically restoring orthogonality among the different users. The linear MMSE equalizer can suppress interference from other cells by whitening the interference. The theoretical model of the equalizer clearly exhibits significant improvement in performance over the RAKE receiver, considering the case of the static channel. Even adaptive equalizers, that adapt using the pilot sequence work well when it comes to stationary channels. For fading channels, the LMS adaptive filter does not perform adequately as it requires a long period of time to converge. However, the adaptive filter based on the RLS algorithm provides significant improvement in performance over the RAKE filter even for moderate mobile speeds.

6.2 Future Work

In our project, we worked on the CPICH alone. The equalizer coefficients found thus, can be used to decode even the data on the HS-PDSCH or the DPCH. Also, we restricted ourselves to a single-user case. The theory introduced and the equations derived can be generalized to the multi-user case as well. The equalizer coefficients would still remain the same, although the computation time and complexity are increased due to increase in dimensionality. It is the just the spreading code which changes for different users. We could also work at higher speeds than those used for the simulations, with the fact that a greater amount of averaging to be done, as the algorithms would take sufficiently long time to adapt. Also, we could have worked on other converging algorithms such as normalized LMS or Kalman filtering. Besides, we could introduce Interleaving and Error Control Coding into the channel model, which would automatically improve the performance.

Bibliography and References

[1] T. S. Rappaport, *Wireless Communications - Principles and Practice*, Prentice Hall, 1991.

[2] J. G. Proakis, *Digital Communications*, Mc-Graw Hill, 1995.

[3] S. Haykin, *Adaptive Filter Theory*, Prentice Hall, 1996.

[4] W. C. Jakes, *Microwave Mobile Communications*, Wiley, 1974.

[5] C. D. Frank, U. Visotsky and U. Madhow, "Adaptive Interference Suppression for the Downlink of a DS-CDMA System with Long Spreading Sequences", *Proc. 36^{th} Annual Conference on Communication, Control and Computing*, Monticello, IL, Sep. 1998.

[6] 3GPP TS 25.101 v 5.7.0: UE radio transmission and reception (FDD), Dec. 2003.

[7] 3GPP TS 25.211 v 5.5.0: Physical channels and mapping of transport channels onto physical channels (FDD), Sep. 2003.

[8] 3GPP TS 25.213 v 5.4.0: Spreading and modulation (FDD), Sep. 2003.

[9] P. Dent, G. E. Bottomley, T. Croft, "Jakes model for fading channels", *Proc. IEE*, 1993.

[10] W. A. Sethares, "The Least Mean Square Family", Oct. 1993.

[11] S. U. H. Qureshi, "Adaptive Equalization", *Proc. IEEE*, Vol. 73, No. 9, Sep. 1983.

[12] A. Klein, "Data Detection Algorithms Specially Designed for the Downlink of CDMA Mobile Radio Systems," *Proc. IEEE International Vehicular Technology Conference*, VTC'97, Phoenix, May 1997, pp. 203-207.

[13] C. D. Frank and E. Visotsky, "Adaptive Interference Suppression for Direct-Sequence CDMA Systems with Long Spreading Codes," *Proc. 36^{th} Annual Conference on Communication, Control and Computing*, Monticello, IL, Sep. 1998.

[14] I. Ghauri and D. T. M. Shock, "Linear Receivers for the DS-CDMA Downlink Exploiting Orthogonality of Spreading Sequences," *Proc. 32nd Asilomar*

Conference on Signals, Systems and Computers, Asilomar, CA, Nov. 1998, pp. 650-654.

[15] G. E. Bottomley, "Optimizing the RAKE Receiver for the CDMA Downlink," *Proc. IEEE International Vehicular Technology Conference*, VTC'93, 1993, pp. 742-745.

[16] P. Komulainen, M. J. Heikkila, J. Lilleberg, "Adaptive Channel Equalization for CDMA Downlink," *Proc. IEEE* 6^{th} *International Symposium on Spread-Spectrum techniques and Applications*, ISSSTA 2000, NJ, Sep. 2000, pp. 363-367.