VLSI Implementation of OFDM Channel Estimation

by

Eng. Khaled Mohamed ElWazeer
Electronics and Communications Department
Faculty of Engineering, Cairo University

A Thesis Submitted to the
Faculty of Engineering at Cairo University
in Partial Fulfillment of the
Requirement for the Degree of
MASTER OF SCIENCE
in
ELECTRONICS AND COMMUNICATIONS ENGINEERING

FACULTY OF ENGINEERING, CAIRO UNIVERSITY
GIZA, EGYPT
JULY 2009

VLSI Implementation of OFDM Channel Estimation

by

Eng. Khaled Mohamed ElWazeer
Electronics and Communications Department
Faculty of Engineering, Cairo University

A Thesis Submitted to the
Faculty of Engineering at Cairo University
in Partial Fulfillment of the
Requirement for the Degree of
MASTER OF SCIENCE
in
ELECTRONICS AND COMMUNICATIONS ENGINEERING

Under the supervision of

Prof. Dr. Serag E.D. Habib
Associate Prof. Mohamed M. Khairy
Assistant Prof. Hossam A. Fahmy
Electronics and Communications Dept.

FACULTY OF ENGINEERING, CAIRO UNIVERSITY
GIZA, EGYPT
JULY 2009

VLSI Implementation of OFDM Channel Estimation

by

Eng. Khaled Mohamed ElWazeer
Electronics and Communications Department
Faculty of Engineering, Cairo University

A Thesis Submitted to the
Faculty of Engineering at Cairo University
in Partial Fulfillment of the
Requirement for the Degree of
MASTER OF SCIENCE
in
ELECTRONICS AND COMMUNICATIONS ENGINEERING

Approved by the
Examining Committee

Prof. Dr. Magdy Fikry Ragai, Member

Associate Prof. Mohamed Amin Dessouky, Member

Prof. Dr. Serag. E.D. Habib , Thesis Main Advisor

Associate Prof. Mohamed M. Khairy, Thesis Advisor

FACULTY OF ENGINEERING, CAIRO UNIVERSITY
GIZA, EGYPT
JULY 2009

ACKNOWLEDGMENTS

I would like to thank my supervisors, Prof. Dr. Serag Habib, Dr. Mohamed Khairy and Dr. Hossam Fahmy for their continuous support, advice, and guidance throughout my work.

Many thanks to Eng. Amr Mohamed Hussein, Eng. Mohamed Ismail, and Eng. Abd El-Mohsen Khater who started the work of hardware implementation of mobile WiMAX. Their help and cooperation will not be forgotten.

Many thanks to my parents, my grandmother and my sister for their continuous support and encouragement during all working days.

Many thanks to my wife, for her cooperation, support and patience while I was working on my thesis.

DEDICATION

To my parents, my wife, and to the loving memory of my grandfather.

ABSTRACT

Channel estimation and equalization is an essential part of today's wireless communication systems. Wireless channels are usually much more difficult to track and estimate than the wireline channels. Two main challenges exist in wireless channels which are multipath fading and doppler spread. Multipath fading represents the selectivity of the channel and as the delay spread of the channel increases, the frequency selectivity increases, and the channel changes rapidly between adjacent subcarriers. As the mobility of the user increases, the doppler shift increases, and the channel is less correlated in time. These effects are usually limiting the wireless systems and cause very high bit error rates if not estimated and equalized correctly.

In this thesis, different channel estimation techniques for OFDM systems are examined. OFDM systems are dominating in wireless transmission. Simulations are made for many widely used channel estimation techniques and they are all compared with respect to their bit error rates and mean square errors. These methods are tested in different channel conditions to cover most of the states for wireless systems.

From all of the compared techniques, the best algorithms are tested in WiMAX 802.16e as a real wireless system. A novel estimation technique is proposed for the WiMAX case. This technique is simple, and gives good performance compared to other techniques. This technique is based on Wiener Filtering and simple linear interpolations.

Also, a minor modification in WiMAX PUSC permutation scheme pilot pattern is proposed. This modification is only in the leftmost and the rightmost edges of the symbol. One pilot is added in each symbol to prevent edge effects. This modification reduces the error floors significantly in the traditional system and has a performance gain of 10 dB's in the high SNR's bit error rates.

An efficient hardware architecture is built for the proposed algorithm for WiMAX system. The main advantage of the architecture is its ability to cope with different wireless systems not only the WiMAX system by reconfiguring some registers

and memories in it. The hardware implementation obeys the timing constraints of WiMAX and is capable of estimating all the complex channels within each OFDM symbol in the time duration of the symbol.

Contents

List of Tables

List of Figures

List of Symbols

$h(t, \tau)$	Complex channel impulse response
$\gamma_k(t)$	Average path gain of the k^{th} tab of the channel
τ_k	Average path delay of the k^{th} tab of the channel
$H(t, f)$	Complex channel frequency response at time t - continuous
$H[n, k]$	Complex channel frequency response at subcarrier k, symbol n
$H[m]$	Complex channel frequency response at point m on the time/frequency grid
T_{sym}	OFDM symbol duration
Δf	OFDM subcarrier spacing in Hz
$r_H[n, k]$	Channel correlation function between n symbols and k subcarriers
$r_t[n]$	Channel time correlation function between n symbols
$r_f[k]$	Channel frequency correlation function between k subcarriers
$r_t(\Delta t)$	Channel time correlation function at a delay Δt seconds
$r_f(\Delta f)$	Channel frequency correlation function at a frequency spacing Δf Hz
f_D	Channel doppler frequency
f_c	RF carrier frequency of transmission
C	Speed of light in air
$x[n, k]$	OFDM subcarrier k at the n^{th} symbol
$x[m]$	OFDM subcarrier at point m on the time/frequency grid
$n[n, k]$	AWGN term added at subcarrier k at the n^{th} symbol
$n[m]$	AWGN term added at point m on the time/frequency grid
$\widehat{x}[n, k]$	Estimated OFDM subcarrier k at the n^{th} symbol at the receiver
$\widehat{x}[m]$	Estimated received OFDM subcarrier at point m on the time/frequency grid

$\widehat{H}[n,k]$	Complex estimated channel frequency response at subcarrier k, symbol n
$\widehat{H}[m]$	Complex estimated channel frequency response at point m on the T/F grid
$y[n,k]$	Received OFDM subcarrier k at the n^{th} symbol
$y[m]$	Received OFDM subcarrier at point m on the time/frequency grid
BER	Received Bit Error Rate
MSE	Mean squared error between the estimated and the real channel
LS	Least Square estimate
$\widetilde{H}[n,k]$	Complex initial LS estimate of the channel frequency response
$\breve{x}[n,k]$	Initial remodulated decision at subcarrier k, symbol n at the receiver
N_{FFT}	The FFT size at the receiver
$R_H[m]$	Channel Covariance matrix between m apart points on the time/frequency grid
R_f	Channel frequency cross correlation matrix
ρ	Received signal to noise ratio
T_m	Maximum channel delay spread
B_c	Channel coherence bandwidth

Chapter 1

Introduction

1.1 Introduction to OFDM

OFDM stands for Orthogonal Frequency Division Multiple Access. This technique is very famous nowadays and is showing enormous success especially in digital high data rates communication systems. This is because OFDM is robust against multipath fading channels and intersymbol interference. [1]

In OFDM, the whole bandwidth is divided into number of subcarriers and each subcarrier carries different modulated data. This is the main idea of the conventional muticarrier communication systems. The frequency spacing between different subcarriers is chosen such that the different subcarriers are orthogonal to each other. This makes OFDM spectral efficient than other conventional muticarrier techniques. The main disadvantage of the OFDM systems is the higher complexity. In order to modulate N subcarriers into one OFDM symbol, we will need N modulators in the transmitter an N demodulators at the receiver. This limited the use of OFDM at the beginning, but after the Fast Fourier Transform "'FFT"' method was invented, the OFDM systems became more realistic, and their use started to dominate especially in wireless systems.

To generate an OFDM symbol, the baseband version of the symbol is generated first, and then the whole symbol is raised on an RF carrier frequency for transmis-

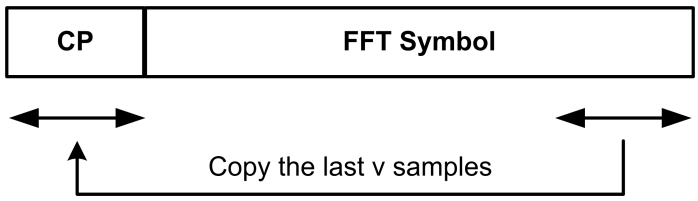

Figure 1.1: Cyclic Prefix Illustration

sion.

The baseband OFDM symbol is generated by simply modulating a serial digitized data stream using common modulation schemes such as Quadrature Phase Shift Keying "QPSK", Quadrature Amplitude Modulation "QAM", or any other modulation scheme. The serial data stream is converted to a parallel stream and sampled with a sampling rate of $\frac{N}{T_s}$ where N is the number of subcarriers, and T_s is the OFDM symbol duration.

The sampled data at the n^{th} subcarrier X_n represents a single subcarrier. All subcarriers are summed together to form the final OFDM symbol "time domain version" as follows:

$$x_m = \sum_{n=1}^{N-1} X_n e^{j\frac{2\pi mn}{N}}, 0 < m < N-1 \tag{1.1}$$

The previous equation can be done easily using an "Inverse Fourier Transform" IFFT engine that gets all the X_n parallel samples and produces all the x_m time domain samples. The IFFT engine is now very popular and many work was done in very efficient, fast, configurable and reliable implementations of it.

After performing the IFFT, the cyclic prefix is added to the OFDM system. This is done by copying the last v samples to the beginning of the OFDM symbol, where v is the number of cyclic prefix "CP" samples. This is shown in figure 1.1

The cyclic prefix is essential and has two advantages:

1. The cyclic prefix gather the delayed versions of the OFDM time domain symbol, so that the delay spread of the wireless channel doesn't cause intersymbol interference. This is always achieved if the cyclic prefix length is larger than

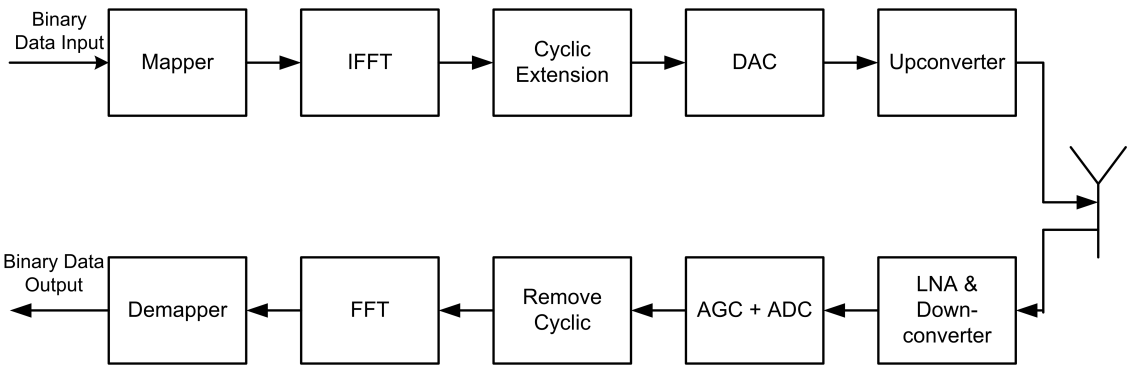

Figure 1.2: Simple OFDM System Illustration

the largest delay spread of the channel.

2. The cyclic prefix makes the channel appears to provide a circular convolution to the OFDM symbol. This makes it possible to realize the IFFT/FFT operations in the transmitter/receiver without having any ISI, i.e. the samples of the resulting filtered data with the channel are only dependent on the symbol itself and doesn't have any information from subsequent symbols.

Another benefit from the OFDM system is the ability to have very simple equalization. The large number of subcarriers makes the spectral bandwidth per subcarrier very small, so it can be assumed that the channel frequency response is constant in this narrow band. This makes the equalization in frequency domain is as simple as dividing the complex received subcarrier by the complex estimated channel at this subcarrier. A simple block diagram of the OFDM system is shown in figure 1.2

1.2 Wireless Systems Overview

Wireless communications is the transfer of information over a distance without using any electrical conductors or wires. Wireless communications is very famous especially after the invention of radios, cellular phones, personal digital assistants "PDAs" and other devices. Examples of the use of wireless communications are mobile phones, GPS systems, wireless computer networks, satellite televisions and

more.

The transmission of data in wireless systems can be via radio frequency communications, microwave communication or infrared. Microwave is adopted for long range line of sight communications such as sattelites, and infrared is adopted for short range communications such as remote controls.

One famous example of the wireless systems nowadays is the WiFi. WiFi is a wireless local area network "LAN" technology that enables many PCs to communicate with each other without the traditional wire-based Ethernet. WiFi access points are fixed in places that should be covered by this type of network in order to transmit and/or receive data from different PCs and laptops simultaneously.

Systems like WiFi are called LAN systems, which -as we stated before- stands for Local Area Network. These type of systems are always covering very limited area including for example a home, office, a small group of buildings, an airport, or any similar areas. Systems that extend this coverage area to cover complete cities for example are called: Metropolitan Area Network "MAN". The most broad network is the "WAN" which covers many countries simultaneously. A popular example of the WAN networks is the Internet.

In this thesis, we are focusing on the MAN networks especially for cellular phone networks. Such networks are more complex because the users are allowed to move with high velocities, and the quality of service should be maintained.

In order to cover the whole area -a city for example-, many base stations are built all around the city to assure proper coverage of the mobile communications service area and offer needed services to available subscribers. This kind of mobile service has started late 70s and early 80s and from that era until now many evolutions have occurred that changed the face of this service from usability, cost and quality and quantity of services it offers, below we will see a snap shot of different generations of mobile telecommunications through the history of this service.

1.2.1 1G Systems

At the every beginning of wireless transmission and cellular phones, the systems were completely analog. The systems were offering voice services mainly. No data transfer services or internet services were available at this time. The main suppliers for the 1G systems were: AMPS in north America, NMT and TACS in Europe, Middle east and Asia. The difference between 1G systems and any other next generation wireless system is that it is analog, all the next generation systems are digital.

1.2.2 2G Systems

In 1980, the 2G systems started to appear in Europe. It was based on the GSM technology which stands for: (Global System for Mobile Communication). It mainly produced voice service, basic messaging services which is called SMS, and very slow data services. The main achievement of the 2G systems is the reduction of the cost per subscriber. By the late 1990, data transfer speeds were raised up to 384 Kbps, with the aid of new introduced technologies such as CDMA.

The starting of the 2G network operations was first in Finland in 1991. One of the main advantages of the 2G over 1G is the digital transmission, which helped in increasing the number of subscribers, and utilizing the available spectrum efficiently. Minor data transfer capabilities started to evolve such as the SMS messaging service.

1.2.3 3G Systems

In the middle of 1990's, research started to go in the direction of increasing the speed and the quality of service for the 2G mobile systems, in order they can support new services such as video calling, video and TV streaming, and fast internet access. Research resulted in the 3^{rd} generation mobile telecommunication

systems (UMTS) which stands for Universal Mobile Telecommunication System. Using UMTS, the data connection speed can theoretically reach up to 14.4 Mbps. The main improvements are the use of adaptive modulation and coding and using modified ARQ -Automatic Repeat Request- standards. The technology is approved by the (ITU), which stands for International Telecommunication Union. The standard for 3G mobile technology is named IMT-2000 which stands for International Mobile Telecommunication 2000.

The new services offered by the 3G technology are achieved while improving the spectral efficiency and the network overall capacity. Services range from voice calls up to broadband wireless internet and data transmission.

1.2.4 4G Systems

The next evolution that is expected to be released soon is the 4th generation which is based on LTE (Long Term Evolution) and WiMAX technologies that are promising an internet speed that reaches 233 Mbit/s for mobile users.

4G, an abbreviation for Fourth-Generation, is a term used to describe the next complete evolution in wireless communications. A 4G system will be a complete replacement for current networks and be able to provide a comprehensive and secure IP solution where voice, data, and streamed multimedia can be given to users on an "Anytime, Anywhere" basis, and at much higher data rates than previous generations. It is promoted to replace the standard DSL -Digital Subscriber Lines- by wireless equivalent systems with much higher data rates.

1.3 Wireless Channels

The communication channel that exists in wireless applications is the fading channel. Fading channels are characterized by their impulse response $h(t, \tau)$, or by their

frequency response $H(t, f)$

Any fading channel model is composed of taps, which are the different paths from the transmitter to the receiver. Each k^{th} path is characterized by its delay τ_k and its complex amplitude γ_k. At the receiver, delayed and attenuated versions of the transmitted signal are received according to the different path powers and delays. Using the previous convention, the channel impulse response can be written as [2]:

$$h(t, \tau) = \sum_k \gamma_k(t)\delta(\tau - \tau_k) \tag{1.2}$$

The frequency response of the channel can be written as:

$$H(t, f) = \sum_k \gamma_k(t)e^{-j2\pi f\tau_k} \tag{1.3}$$

In an OFDM system, OFDM symbols are transmitted every T_{sym} seconds, with N_{FFT} subcarriers that are spaced by Δf Hz, the channel frequency response correlation function of such a system as a function of the OFDM symbol index n and subcarrier spacing index k can be written as:

$$\begin{aligned} r_H[n, k] &= r_t[n]r_f[k] \\ &= r_t(nT_{sym})r_f(k\Delta f) \end{aligned} \tag{1.4}$$

The frequency correlation function $r_f(\Delta f)$ can be written as:

$$r_f(\Delta f) = \sum_k \sigma_k^2 e^{-j2\pi\Delta f\tau_k} \tag{1.5}$$

Here we assumed that total average power of the channel impulse response is unity, σ_k^2 is the average power of the k^{th} path.

The time correlation function depends mainly on the doppler spectrum of the channel. If we assumed flat doppler spectrum, then the time correlation function $r_t(\Delta t)$ can be written as:

$$r_t(\Delta t) = sinc(2\pi f_D\Delta t) \tag{1.6}$$

Where the doppler frequency f_D is dependent on the vehicle speed v and the carrier frequency f_c. It can be written as:

$$f_D = \frac{v f_c}{C} \tag{1.7}$$

1.4 Channel Estimation

The wireless channel described in the previous section destroys dramatically the performance of any wireless system especially if it is varying rapidly in time and is highly selective in frequency. This is mainly because every OFDM subcarrier is now multiplied by a complex channel frequency response which has a magnitude and phase. The magnitude is some sort of noise, and the phase is a rotation in the subcarrier phase. This results in the loss of orthogonality between all the subcarriers and impossible detection of such a signal. Due to this, the channel response and effect should be estimated and the received subcarriers should be corrected by dividing them by the estimated channel frequency response.

Channel estimation is a complex task. The wireless channels, as we discussed, are varying very rapidly and it is not possible to estimate a single subcarrier channel effect, and assume that this is the effect that exists for ever. Estimation should be made on each subcarrier, trying to get the channel estimation at this particular subcarrier only.

Usually, there is no pre-existing information about the channel that could be used in the estimation. This is the challenging part. The channel is a totally random and no information is available about the type of randomness. The estimator then is blind and doesn't know anything about the channel. To aid in the channel estimation, modern wireless systems add some well-known subcarriers to the transmitted symbols. These subcarriers are known to the receiver and the receiver

mainly utilizes them to estimate the channel correctly.

The added well-known subcarriers are divided into two categories, first is the transmission of individual subcarriers -called pilots- frequently in between the data subcarriers. The second method is to send a complete training symbol -called a preamble- in the starting of the frame transmission. Both methods can be used together to get better channel estimates.

In this thesis, we will try to utilize both of these aids, the preamble and the pilots, in order to produce channel estimates which are as exact as possible, to overcome the problem of rapidly changing wireless channels that are corrupting the data transmission.

1.5 Thesis Outline

This thesis is organized as follows. This chapter included a brief introduction about wireless systems and channels. The next chapter -chapter 2- discusses in details the problem of channel estimation in OFDM systems concentrating on the Decision Directed channel estimation methods. Chapter 3 introduces the WiMAX system as a good example to prototype our work on. WiMAX is one of the next generation wireless systems providing higher data rates with the support of much harsh environments and rapid mobile users. Chapter 4 discusses the channel estimation problem in WiMAX. It is mainly talking about Pilot Assisted channel estimation methods. Chapter 5 describes in details the hardware architecture of the channel estimation system. Finally chapter 6 includes the conclusions and the future work in this topic.

Chapter 2

Channel Estimation Techniques in OFDM

2.1 Introduction

In this chapter, various downlink channel estimation algorithms will be stated, derived, and simulated in order to compare their performance with each other.
Channel estimation and equalization is an essential part of the receiver in OFDM systems in case coherent demodulation and detection is used. The multipath channel with its delay spread and doppler spectrum corrupts the subcarriers of the OFDM symbol and channel equalization is necessary to correctly decode and detect the data at the receiver.

The downlink channel estimation is much different than the uplink. In the downlink, the mobile station is responsible for the estimation, and it sees the whole channel as a correlated channel, because it is the channel between it and the base station. For the uplink, this is not the case, since the base station receives from many users, each with his own channel, and this leads to uncorrelated channel seen by the base station. Base station uplink estimation should be done on a tile based grids, and this out of our scope here.

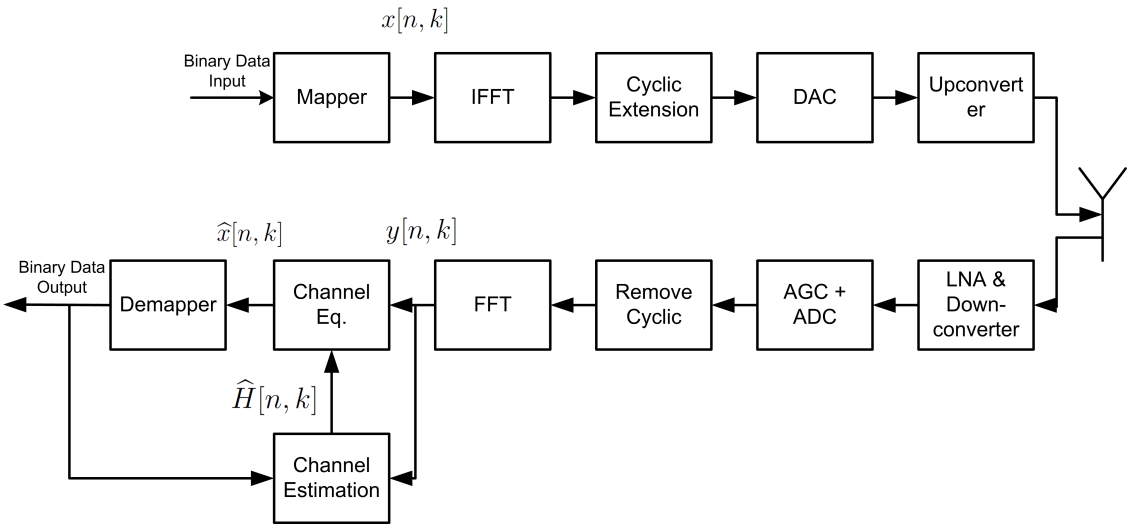

Figure 2.1: OFDM Block Diagram with Channel Estimation

A detailed block diagram of the OFDM system, with channel estimation and equalization blocks is shown in figure 2.1.

If ISI is eliminated, the fading channel is represented simply as a multiplication of the different subcarriers $x[n,k]$ with the corresponding complex channel frequency response $H[n,k]$ plus adding a random AWGN noise $n[n,k]$ to produce the received subcarriers $y[n,k]$. This is clarified by the following equation:

$$y[n,k] = x[n,k]H[n,k] + n[n,k] \tag{2.1}$$

The aim of the channel estimation block is to estimate a value for each $H[n,k]$ so that the channel equalizer 'Channel Eq.' block can cancel the effect of the channel correctly. The channel equalizer block performs the following simple operation:

$$\widehat{x}[n,k] = \frac{y[n,k]}{\widehat{H}[n,k]} \tag{2.2}$$

Different types of channel estimation techniques are available in literature. They can be simply divided into two main categories which are:

1. Decision Directed Channel Estimation

2. Pilot Assisted Channel Estimation

In the Decision Directed case, the current decisions are fed back to the estimator block, so that the estimator can get a better estimates for the following symbol and so on. On the other case, the Pilot Assisted case depends only on the pilots that are usually sent within each OFDM symbol. These pilots are well-known at the receiver and the channel can be estimated with very low error at their locations. These pilots are used to estimate the remaining data symbols. No feedback is required in the pilot assisted case.

In the case of decision directed estimation, pilots are less effective than in the pilot assisted estimation. On the other hand, complete training symbols -well-known as preambles- are required in the decision directed case to perform well. This will be shown later in this chapter.

In our work, we investigated both methods with different conditions. Simulations are carried on for performance evaluation of both methods under different channels. The figure of merit for comparison is the resulting bit error rate curve vs. the signal to noise ratio, and the mean square error curve between the estimated and the correct channel. This mean square error is defined as:

$$MSE = E[|\widehat{H}[n,k] - H[n,k]|^2] \tag{2.3}$$

2.2 Decision Directed Estimation

As we stated before, the decision directed estimation utilizes the decisions of the previous symbol to better estimate the channel frequency response at the current symbol. These decisions are fed back to the estimator, the estimator remodulates

them and produces a new sliced version of it $\breve{x}[n,k]$. This new sliced symbol is used to make an initial estimate $\widetilde{H}[n,k]$ where:

$$\begin{aligned} \widetilde{H}[n,k] &= \frac{y[n,k]}{\breve{x}[n,k]} \\ &= H[n,k]\frac{x[n,k]}{\breve{x}[n,k]} + \frac{n[n,k]}{\breve{x}[n,k]} \end{aligned} \tag{2.4}$$

If the decisions are error-free, i.e., $\breve{x}[n,k] = x[n,k]$ then this is a good initial estimate of the channel. If not, the estimation is not correct and needs further improvement. The second step of the estimation is to filter these initial estimates to produce the final estimates. The filter is in general manipulated as a multiplication of the initial estimates $\widetilde{H}[n,k]$ with some coefficients and summing these multiplications to produce the final estimates $\widehat{H}[n,k]$. The mathematical formulation of this FIR filter is different and depends on whether the filter is one dimensional or two dimensional filter.

In the following sections, we will describe each type of these filters, mathematically formulate the problem, derive the set of coefficients for each case which minimizes the mean squared error between the estimated and original channel.

2.2.1 Ideal 2D MMSE Estimator

The ideal 2D MMSE estimator is an infinite impulse response filter whose coefficients are chosen to minimize the error between the original and the estimated channel frequency response. [3]
To mathematically formulate the problem, let $\widetilde{H}_n$ be a column vector of size $N_{FFT} * 1$, where N_{FFT} refers to the total number of FFT subcarriers in symbol number n. $\widetilde{H}_n$ represents the initial estimates in symbol number n.

$$\widetilde{H}_n = \begin{bmatrix} \widetilde{H}[n,1] \\ \widetilde{H}[n,2] \\ \vdots \\ \widetilde{H}[n,N_{FFT}] \end{bmatrix}$$

Also, let $\widehat{H}_n$ be a column vector of size $N_{FFT} * 1$, where N_{FFT} refers to the total number of FFT subcarriers in symbol number n. $\widehat{H}_n$ represents the final estimates in symbol number n.

$$\widetilde{H}_n = \begin{bmatrix} \widehat{H}[n,1] \\ \widehat{H}[n,2] \\ \vdots \\ \widehat{H}[n,N_{FFT}] \end{bmatrix}$$

Let C_n be an $N_{FFT} * N_{FFT}$ matrix representing the filter coefficients. This matrix is written as:

$$C_n = \begin{bmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,N_{FFT}} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,N_{FFT}} \\ \vdots & \vdots & \ddots & \vdots \\ c_{N_{FFT},1} & c_{N_{FFT},2} & \cdots & c_{N_{FFT},N_{FFT}} \end{bmatrix}$$

Then, the ideal estimator can be written as:

$$\widehat{H}_n = \sum_{m=0}^{\infty} C_m \widetilde{H}_{n-m} \tag{2.5}$$

As shown from the above equation, the ideal MMSE filter is impractical because it requires an infinite number of OFDM symbols to work on. This is impossible. The ideal estimator coefficients are derived in the following section in order to compare the ideal resulting mean square error with other estimation methods that imply finite impulse response filters. Also, the ideal infinite impulse response filter is the only way to deduce a robust filter that doesn't require information about channel statistics. This will be clarified later.

2.2.2 Calculation of the Ideal MMSE Filter Coefficients

To calculate the coefficients in equation (2.5), the orthogonality principle is applied which states [4]:

$$E[(\widehat{H}_n - H_n)\widetilde{H}^H_{n-m}] = 0 \tag{2.6}$$

This leads to:

$$\sum_{m_1=0}^{\infty} C_{m_1} R_H[m - m_1] - R_H[m] + \rho C_m = 0 \tag{2.7}$$

Where $R_H[m]$ is the covariance matrix of the channel frequency response between two OFDM symbols separated by m symbols. This covariance matrix can be separated to two correlation functions [3] multiplied by each other one is the time correlation function and the other is the frequency covariance matrix as below:

$$R_H[m] = r_t[m]R_f = E[H_{n+m}H^H_n] \tag{2.8}$$

The time correlation function depends on the doppler spectrum of the channel and was discussed in chapter 1 (1.6). The frequency covariance matrix can be written using the frequency correlation function of the channel previously mentioned in (1.5) as shown below:

$$R_f = \begin{bmatrix} r_f(0) & r_f(1) & \cdots & r_f(N-1) \\ r_f(-1) & r_f(-2) & \cdots & r_f(N-2) \\ \vdots & \vdots & \ddots & \vdots \\ r_f(-N+1) & r_f(-N+2) & \cdots & r_f(0) \end{bmatrix} \tag{2.9}$$

Then:

$$\sum_{m_1=0}^{\infty} C_{m_1} r_t[m - m_1]R_f - r_t[m]R_f + \rho C_m = 0 \tag{2.10}$$

Let R_f be eigen decomposed as follows:

$$R_f = UDU^H \tag{2.11}$$

Then:

$$\sum_{m_1=0}^{\infty} C_{m_1} r_t[m - m_1] UDU^H - r_t[m] UDU^H + \rho C_m = 0 \tag{2.12}$$

Multiply the last equation from the left by U^H and from the right by U, we get:

$$\sum_{m_1=0}^{\infty} U^H C_{m_1} U[r_t[m - m_1]D + \rho\delta[m - m_1]I] - r_t[m]D = 0 \tag{2.13}$$

Every term from the above equation is a diagonal matrix, Let $U^H C_m U = \Phi_m$ where:

$$\Phi_m = \begin{bmatrix} \phi_m^{(1)} & \cdots & \cdots & \cdots \\ \cdots & \phi_m^{(2)} & \cdots & \cdots \\ \vdots & \vdots & \ddots & \cdots \\ \cdots & \cdots & \cdots & \phi_m^{(N_{FFT})} \end{bmatrix}$$

Also, D is a diagonal matrix where:

$$D = \begin{bmatrix} d_1 & \cdots & \cdots & \cdots \\ \cdots & d_2 & \cdots & \cdots \\ \vdots & \vdots & \ddots & \vdots \\ \cdots & \cdots & \cdots & d_{N_{FFT}} \end{bmatrix}$$

If we take the k^{th} row of each matrix from equation (2.13), we will get:

$$\sum_{m_1=0}^{\infty} \phi_{m_1}^{(k)} [d_k r_t[m - m_1] + \rho\delta[m - m_1]] - d_k r_t[m] = 0 \tag{2.14}$$

Dividing by ρ we get:

$$\sum_{m_1=0}^{\infty} \phi_{m_1}^{(k)}[\frac{d_k}{\rho} r_t[m-m_1] + \delta[m-m_1]] = \frac{d_k}{\rho} r_t[m] \tag{2.15}$$

Let:

$$R_m^{(k)} = \frac{d_k}{\rho} r_t[m] \tag{2.16}$$

$$M_m^{(k)} = R_m^{(k)} + \delta[m] \tag{2.17}$$

then equation (2.15) becomes:

$$R_m^{(k)} = \sum_{m_1=0}^{\infty} M_{m-m_1}^{(k)} \phi_{m_1}^{(k)}, m > 0 \tag{2.18}$$

In the following, we will remove the superscript k. The aim is to solve this equation and calculate the different $\phi_m^{(k)}$ for different ks.

Let:

$$M_{m_1} = \sum_{l=0}^{\infty} M_l^+ M_{m_1-l}^- \tag{2.19}$$

$$R_m = \sum_{l=0}^{\infty} M_l^+ X_{m-l}^- \tag{2.20}$$

Substituting in equation (2.18), we get:

$$\sum_{l=0}^{\infty} M_l^+ [X_{m-l} - \sum_{m_1=0}^{\infty} \phi_{m_1} M_{m-m_1-l}^-] = 0 \tag{2.21}$$

Then:

$$X_m = \sum_{m_1=0}^{\infty} \phi_{m_1} M^-_{m-m_1} \tag{2.22}$$

Let:

$$X(\omega) = \sum_{n=-\infty}^{\infty} X_n e^{-j\omega n} \tag{2.23}$$

From 2.20

$$R(\omega) = X(\omega) M^+(\omega) \tag{2.24}$$

By taking the negative single sided fourier transform of equation (2.22), we get:

$$\begin{aligned} X^-(\omega) &= \sum_{m=-\infty}^{0} X_m e^{-j\omega m} \\ &= \sum_{m=0}^{\infty} \sum_{m_1=-\infty}^{0} \phi_{m_1} M^-_{m-m_1} e^{-j\omega m} \\ &= \sum_{m_1=0}^{\infty} \phi_{m_1} \sum_{m=-\infty}^{0} M^-_{m-m_1} e^{-j\omega m} \end{aligned} \tag{2.25}$$

Let $m - m_1 = k$, then:

$$\begin{aligned} X^-(\omega) &= \sum_{m_1=0}^{\infty} \phi_{m_1} e^{-j\omega m_1} \sum_{k=-\infty}^{0} M^-_k e^{-j\omega k} \\ &= \phi^+(\omega) M^-(\omega) \\ &= \phi(\omega) M^-(\omega) \end{aligned} \tag{2.26}$$

This is because $\phi^+(\omega) = \phi(\omega)$ as $\phi_m = 0 \ , \forall m < 0$

From (2.24):

$$X(\omega) = \frac{R(\omega)}{M^{+}(\omega)} \tag{2.27}$$

From equation (2.26)

$$\phi(\omega) = \frac{X^{-}(\omega)}{M^{-}(\omega)} \tag{2.28}$$

Now we need to calculate $X^{-}(\omega)$ and $M^{-}(\omega)$. To calculate $X^{-}(\omega)$ recall that:

$$X^{-}(\omega) = \sum_{n=-\infty}^{0} X_n e^{(-j\omega n)} \tag{2.29}$$

From equation (2.17):

$$R(\omega) = M(\omega) - 1 \tag{2.30}$$

then we have:

$$X(\omega) = \frac{R(\omega)}{M^{+}(\omega)} = M^{-}(\omega) - \frac{1}{M^{+}(\omega)} \tag{2.31}$$

Let $M^{'}(\omega) = \frac{1}{M^{+}(\omega)}$, then:

$$
\begin{aligned}
X(\omega) &= M^{-}(\omega) - M^{'}(\omega) \\
X_n &= M_n^{-} - \frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{e^{j\omega n}}{M^{+}(\omega)} d\omega \\
&= M_n^{-} - \frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{e^{j\omega n}}{\sum_{m=0}^{\infty} M_m^{+} e^{-j\omega m}} d\omega \\
X^{-}(\omega) &= \sum_{n=-\infty}^{0} X_n e^{-j\omega n} \\
&= \sum_{n=-\infty}^{0} M_n^{-} e^{-j\omega n} - \sum_{n=-\infty}^{0} [\frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{e^{j\omega n}}{\sum_{m=0}^{\infty} M_m^{+} e^{-j\omega m}} d\omega] \\
&= M^{-}(\omega) - \frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{\sum_{n=-\infty}^{0} e^{j\omega n}}{\sum_{m=0}^{\infty} M_m^{+} e^{-j\omega m}}\, d\omega \\
&= M^{-}(\omega) - \frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{d\omega}{(1-e^{-j\omega})(M_o^{+} + M_1^{+} e^{-j\omega} + M_2^{+} e^{-2j\omega} + \cdots)} \\
&= M^{-}(\omega) - \frac{1}{M_o^{+}} \qquad (2.32)
\end{aligned}
$$

Then, from equation (2.26) we get:

$$
\begin{aligned}
\phi(\omega) &= \frac{X^{-}(\omega)}{M^{-}(\omega)} \\
&= 1 - \frac{1}{M_o^{+} M^{-}(\omega)} \qquad (2.33)
\end{aligned}
$$

Now, to get $M^{-}(\omega)$, from equation (2.20):

$$
\begin{aligned}
M(\omega) &= M^{+}(\omega) M^{-}(\omega) \qquad (2.34) \\
ln(M(\omega)) &= ln(M^{+}(\omega)) + ln(M^{-}(\omega)) \\
&= \sum_{-\infty}^{\infty} \gamma_n e^{-j\omega n} \\
\gamma_n &= \frac{1}{2\pi}\int_{-\pi}^{\pi} ln(M(\omega)) e^{j\omega n} d\omega \;, n \neq 0 \qquad (2.35)
\end{aligned}
$$

Also, we can write $M(\omega)$ as follows:

$$
\begin{aligned}
ln(M(\omega)) &= \sum_{n=-\infty}^{0} \gamma_n e^{j\omega n} + \sum_{n=0}^{\infty} \gamma_n e^{j\omega n} && (2.36)\\
so, \quad M^{+}(\omega) &= exp(\sum_{n=0}^{\infty} \gamma_n e^{j\omega n}) && (2.37)\\
and \quad M^{-}(\omega) &= exp(\sum_{n=-\infty}^{0} \gamma_n e^{j\omega n}) && (2.38)\\
Also, \quad \gamma_o &= \frac{1}{4\pi} \int_{-\pi}^{\pi} ln(M(\omega)) d\omega && (2.39)
\end{aligned}
$$

$$
\begin{aligned}
Mo^{+} &= \frac{1}{2\pi} \int_{-\pi}^{\pi} M^{+}(\omega) d\omega \\
&= \frac{1}{2\pi} \int_{-\pi}^{\pi} exp(\sum_{n=0}^{\infty} \gamma_n e^{j\omega n}) d\omega \\
&= \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{\gamma_o} e^{(\gamma_1 e^{j\omega} + \gamma_2 e^{2j\omega} + \cdots)} d\omega \\
&= \frac{1}{2\pi} e^{\gamma_o} \int_{-\pi}^{\pi} e^{(\gamma_1 e^{j\omega} + \gamma_2 e^{2j\omega} + \cdots)} d\omega \\
&= \frac{1}{2\pi} e^{\gamma_o} \int_{-\pi}^{\pi} (1 + \gamma_1 e^{j\omega} + \gamma_2 e^{2j\omega} + \cdots) d\omega \\
&= e^{\gamma_o} && (2.40)
\end{aligned}
$$

So, our final solution will be:

$$
\begin{aligned}
\phi_m^{(k)} &= \frac{1}{2\pi} \int_{-\pi}^{\pi} \phi^{(k)}(\omega) e^{j\omega m} d\omega \\
&= \frac{1}{2\pi} \int_{-\pi}^{\pi} (1 - \frac{1}{M_o^{+} M^{-}(\omega)})^{(k)} e^{j\omega m} d\omega && (2.41)
\end{aligned}
$$

$$M_o^+ = e^{\gamma_o} \tag{2.42}$$

$$\gamma_o = \frac{1}{4\pi}\int_{-\pi}^{\pi} ln(M(\omega))d\omega \tag{2.43}$$

$$\begin{aligned} M^-(\omega) &= exp(\sum_{n=-\infty}^{0} \gamma_n e^{j\omega n}) \\ \gamma_n &= \frac{1}{2\pi}\int_{-\pi}^{\pi} ln(M(\omega))e^{-j\omega n}d\omega, \quad n \neq 0 \end{aligned} \tag{2.44}$$

$$\begin{aligned} M(\omega) &= FT(\frac{d_k}{\rho}r_t(m) + \delta(m)) \\ &= \frac{d_k}{\rho}p_t(\omega) + 1 \end{aligned} \tag{2.45}$$

2.2.3 2D FIR MMSE Estimator

The ideal estimator mentioned above is not practical as it requires infinite number of LS channels to operate on. A small modification on its main equation by limiting the filter to be finite impulse response, this leads to the following estimator:

$$\widehat{H}[n,k] = \sum_{m=0}^{L-1}\sum_{l=k-N}^{k-1} c[m,l,k]\widetilde{H}[n-m,k-l] \tag{2.46}$$

Where $\widetilde{H}(n,k)$ is defined as in equation (2.4), which is repeated below assuming error free decisions:

$$\widetilde{H}[n,k] = H[n,k] + n[n,k] \tag{2.47}$$

To illustrate this filtering equation, it can be written in a clearer matrix form

like this:

$$\widehat{H}[n,k] = \sum_{m=0}^{L-1} C[m]\widetilde{H}[n-m] \tag{2.48}$$

Where:

$$C[m] = \begin{bmatrix} c[m,0,1] & c[m,-1,1] & c[m,-2,1] & \cdots & c[m,-N+1,1] \\ c[m,1,2] & c[m,0,2] & c[m,-1,2] & \cdots & c[m,-N+2,2] \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ c[m,k-1,k] & c[m,k-2,k] & c[m,k-3,k] & \cdots & c[m,k-N,k] \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ c[m,N-1,N] & c[m,N-2,N] & c[m,N-3,N] & \cdots & c[m,0,N] \end{bmatrix}$$

And,

$$\widetilde{H}[n-m] = \begin{bmatrix} \widetilde{H}[n-m,1] \\ \widetilde{H}[n-m,2] \\ \vdots \\ \widetilde{H}[n-m,N] \end{bmatrix}$$

The above filter implies only $N \times L$ multiplications and additions only instead of infinite number in the case of the ideal estimator.

This FIR filter equation is the most important estimation technique in this thesis. By well-defining the coefficient sets, we can simply apply it and use it in our simulations.

2.2.4 Calculation of the 2D FIR Filter Coefficients

Starting from the orthogonality principle as we did before in the ideal MMSE estimator, we get:

$$E((\widehat{H}[n,k] - H[n,k])\widetilde{H}^*[n-m,k-l]) = 0 \tag{2.49}$$

This leads to:

$$E\{[\sum_{m_1=0}^{L-1}\sum_{l_1=k-N}^{k-1} c[m_1,l_1,k][H[n-m_1,k-l_1]+N[n-m_1,k-l_1]] - H[n,k]] * [H^*[n-m,k-l]+N^*[n-m,k-l]]\} = 0 \quad (2.50)$$

$$\sum_{m_1=0}^{L-1}\sum_{l_1=k-N}^{k-1} c[m_1,l_1,k][r_H[m-m_1,l-l_1]+\rho\delta[m-m_1,l-l_1]] - r_H[m,l] = 0 \quad (2.51)$$

Let:

$$c[m,k] = \begin{bmatrix} c[m,k-1,k] \\ c[m,k-2,k] \\ \vdots \\ c[m,k-N,k] \end{bmatrix}$$

and

$$\mathbf{r_f}[k] = \begin{bmatrix} r_f[k-1] \\ r_f[k-2] \\ \vdots \\ r_f[k-N] \end{bmatrix}$$

Where $N = N_{FFT}$ is the FFT size, knowing that R_f is the covariance matrix of the channel frequency response, which was stated before in equation (2.9), and its eigen decomposition which was stated in equation: (2.11), we can write:

$$\sum_{m_1=0}^{L-1} r_t[m-m_1]R_f c[m_1,k] + \rho c[m,k] - r_t[m]\mathbf{r_f}[k] = 0 \quad (2.52)$$

$$\sum_{m_1=0}^{L-1} r_t[m-m_1]UDU^H c[m_1,k] + \rho c[m,k] - r_t[m]\mathbf{r_f}[k] = 0 \quad (2.53)$$

Assume that:

$$D = \begin{bmatrix} d_1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & d_2 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & d_K & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Where K is the number of non-zero diagonal elements of D which is equal to the number of eigen values of the matrix R_f. Then, we can assume that:

$$D^{-1} = \begin{bmatrix} \frac{1}{d_1} & 0 & 0 & 0 & \cdots & 0 \\ 0 & \frac{1}{d_2} & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \frac{1}{d_K} & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Multiply equation (2.53) from the right and the left by: $D^{-1}U^H$, we get:

$$\sum_{m_1=0}^{L-1} r_t[m-m_1]U^H c[m_1,k] + \rho D^{-1}U^H c[m,k] - r_t[m]D^{-1}U^H \mathbf{r_f}[k] = 0 \qquad (2.54)$$

Let:

$$\widetilde{c}[m,k] = U^H c[m,k] \qquad (2.55)$$

$$\widetilde{\mathbf{r}}_\mathbf{f}[k] = U^H \mathbf{r_f}[k] \qquad (2.56)$$

Then:

$$\sum_{m_1=0}^{L-1} r_t[m-m_1]\widetilde{c}[m_1,k] + \rho D^{-1}\widetilde{c}[m,k] - r_t[m]D^{-1}\widetilde{\mathbf{r}}_{\mathbf{f}}[k] = 0 \tag{2.57}$$

For any row l where $d_l \neq 0$ we have:

$$\sum_{m_1=0}^{L-1} r_t[m-m_1]\widetilde{c}[m_1,l,k] + \frac{\rho}{d_l}\widetilde{c}[m,l,k] - r_t[m]\frac{1}{d_l}\widetilde{\mathbf{r}}_{\mathbf{f}}[l,k] = 0 \tag{2.58}$$

Where:

$$\widetilde{c}[m,k] = \begin{bmatrix} \widetilde{c}[m,1,k] \\ \widetilde{c}[m,2,k] \\ \vdots \\ \widetilde{c}[m,N,k] \end{bmatrix} = U^H \begin{bmatrix} c[m,k-1,k] \\ c[m,k-2,k] \\ \vdots \\ c[m,k-N,k] \end{bmatrix}$$

and

$$\widetilde{\mathbf{r}}_{\mathbf{f}}[k] = \begin{bmatrix} \widetilde{\mathbf{r}}_{\mathbf{f}}[1,k] \\ \widetilde{\mathbf{r}}_{\mathbf{f}}[2,k] \\ \vdots \\ \widetilde{\mathbf{r}}_{\mathbf{f}}[N,k] \end{bmatrix} = U^H \begin{bmatrix} r_f[k-1] \\ r_f[k-2] \\ \vdots \\ r_f[k-N] \end{bmatrix}$$

In equation (2.58), we have L coefficient sets that are needed to be calculated, and for each set of coefficients, $K \times N$ coefficients exist for different $l's$ and $k's$.

Equation (2.58) represents a single row of a matrix, and for each m in the range: $0 < m < L-1$, we have L different rows of this matrix, so we can write the L sets of equations that are needed to calculate the L coefficients for constant l and k as follows:

$$\begin{bmatrix} r_t(0) & r_t(-1) & \cdots & r_t(-L+1) \\ r_f(1) & r_t(0) & \cdots & r_t(-L+2) \\ \vdots & \vdots & \ddots & \vdots \\ r_t(L-1) & r_t(L-2) & \cdots & r_t(0) \end{bmatrix} \begin{bmatrix} \widetilde{c}[0,l,k] \\ \widetilde{c}[1,l,k] \\ \vdots \\ \widetilde{c}[L-1,l,k] \end{bmatrix}$$
$$+\frac{\rho}{d_l} \begin{bmatrix} \widetilde{c}[0,l,k] \\ \widetilde{c}[1,l,k] \\ \vdots \\ \widetilde{c}[L-1,l,k] \end{bmatrix} - \frac{\widetilde{\mathbf{r_f}}[l,k]}{d_l} \begin{bmatrix} r_t[0] \\ r_t[1] \\ \vdots \\ r_t[L-1] \end{bmatrix} = 0 \qquad (2.59)$$

Which is simplified as:

$$R_t \widetilde{c_t}[l,k] + \frac{\rho}{d_l} \widetilde{c_t}[l,k] - \frac{\widetilde{\mathbf{r_f}}[l,k]}{d_l} \mathbf{r_t} = 0 \qquad (2.60)$$

Where:

$$\mathbf{r_t} = \begin{bmatrix} r_t[0] \\ r_t[1] \\ \vdots \\ r_t[L-1] \end{bmatrix}, \quad \widetilde{c_t}[l,k] = \begin{bmatrix} \widetilde{c}[0,l,k] \\ \widetilde{c}[1,l,k] \\ \vdots \\ \widetilde{c}[L-1,l,k] \end{bmatrix}$$

Then:

$$\begin{aligned} \widetilde{c_t}[l,k] &= (\frac{\widetilde{\mathbf{r_f}}[l,k]}{d_l})(R_t + \frac{\rho}{d_l} I)^{-1} \mathbf{r_t} \ , \quad d_l \neq 0 \\ &= 0 \ , \qquad\qquad\qquad\qquad\qquad\qquad d_l = 0 \end{aligned} \qquad (2.61)$$

This equation should be calculated $L \times N$ times to calculate all the coefficients needed, the original coefficients are calculated from the obtained coefficients by the following equation:

$$\begin{bmatrix} c[m,k-1,k] \\ c[m,k-2,k] \\ \vdots \\ c[m,k-N,k] \end{bmatrix} = U \begin{bmatrix} \widetilde{c}[m,1,k] \\ \widetilde{c}[m,2,k] \\ \vdots \\ \widetilde{c}[m,N,k] \end{bmatrix} \tag{2.62}$$

2.2.5 1D FIR MMSE Estimator

The above 2D FIR MMSE estimator can be simplified to a 1D estimator. This 1D estimator utilizes only the LS pilots $\widetilde{H}[n,k]$ from the current OFDM symbol n only. In this way, the correlation in time is not used at all, and most probably, the system will perform worse than the equivalent 2D estimator.

In 1D estimators, the estimation obeys the following equation:

$$\widehat{H}[n,k] = \sum_{l=0}^{N-1} c[n,l]\widetilde{H}[n,k-l] \tag{2.63}$$

Which can be written in a matrix form as follows:

$$\widehat{H}[n,k] = C_n^T \widetilde{H}_n \tag{2.64}$$

Where: C_n is a column vector defined as:

$$\begin{bmatrix} c[n,k-1] \\ c[n,k-2] \\ \vdots \\ c[n,k-N] \end{bmatrix}$$

And, $\widetilde{H}_n$ is also a column vector defines as:

$$\begin{bmatrix} \widetilde{H}[n,1] \\ \widetilde{H}[n,2] \\ \vdots \\ \widetilde{H}[n,N] \end{bmatrix}$$

2.2.6 Calculation of the 1D MMSE Filter Coefficients

To calculate the values of the 1D MMSE filter coefficients, one can simply use the same 2D coefficient derivation mentioned above, and remove the subscript m from all the equations, put $r_t[n] = 1$ for any n, and also putting $L = 1$ which means removing all the summations on m and m_1 variables.

Another simple derivation is to start from the orthogonality principle, and complete the derivation using matrix convention, this derivation will be much easier than the 2D estimator derivation deduced in the previous section.

Here, we will briefly deduce the coefficients for the 1D estimator starting from the orthogonality principle. Assume in the subsequent equations that $C_n^T = C$ for simplicity, also $\widetilde{H}_n = \widetilde{H}$. The definition of $\widetilde{H}[n,k]$ didn't change from equation (2.4). We will use the matrix form of this equation which is:

$$\widetilde{H} = H + X^{-1}N \tag{2.65}$$

Where H is the column vector containing all the values of $H[n,k]$ for different $k's$ in the range of $1 \leq k \leq N$. X is an $N \times N$ diagonal matrix, its diagonal elements are the reciprocal values of the decisions, i.e. it can be written as follows:

$$X = \begin{bmatrix} \frac{1}{x[n,1]} & 0 & 0 & 0 \\ 0 & \frac{1}{x[n,2]} & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \frac{1}{x[n,N]} \end{bmatrix}$$

The orthogonality principle can be written as:

$$E[(\widehat{H} - H)\widetilde{H}^H] = 0 \tag{2.66}$$

This leads to:

$$E[C(\widetilde{H} - H)\widetilde{H}^H] = 0 \tag{2.67}$$

Then:

$$E[C\widetilde{H}\widetilde{H}^H - H\widetilde{H}^H] = 0 \tag{2.68}$$

This leads to the following equation for C matrix:

$$C = E[H\widetilde{H}^H](E[\widetilde{H}\widetilde{H}^H])^{-1} \tag{2.69}$$

To get the two terms in the previous equation, we will use the correlation matrix of the channel frequency response R_f, which was previously defined in equation (2.9), we will have the following:

$$\begin{aligned} E[H\widetilde{H}^H] &= E[HH^H + HN^H X^{-1^H}] \\ &= R_f \end{aligned} \tag{2.70}$$

And also:

$$
\begin{aligned}
E[\widetilde{H}\widetilde{H}^H] &= E[(H + X^{-1}N)(H + X^{-1}N)^H] \\
&= E[HH^H + (X^{-1}N)(X^{-1}N)^H] \\
&= R_f + \rho I
\end{aligned} \tag{2.71}
$$

Where:

$$\rho = E[||(X^{-1}N)||^2] \tag{2.72}$$

So, the coefficients are:

$$C = R_f(Rf + \rho I)^{-1} \tag{2.73}$$

2.2.7 System Model

In order to simulate the performance of the decision feedback algorithms, we built a simple OFDM system, which is mainly composed of a preamble followed by complete OFDM symbols to form a frame. No pilots are available in these methods, in order to evaluate the decision directed algorithms clearly. The preamble symbol servers as a training symbol which is sent periodically to improve the estimation. This is the case in the real wireless systems.

The system parameters are shown in table 2.1 below:

Parameter	Value
FFT Size	128
Bandwidth	800 KHz
Guard Subcarriers	8
Symbol Duration	200 μsec
Cyclic Prefix	20% of the Symbol
Modulation	QPSK
Channel Coding	None
Channel Delay Spread	20 μ sec
Channel Paths	2 Taps

Table 2.1: System Parameters

The algorithms were tested on a wireless channel having 2 taps, one at 0 sec, and the other at 20 μ sec. The first tap has 75 % of the power, and the second tap has the remaining 25 % of the power.

Our system uses two antennas in the receiver to utilize space diversity. This requires a diversity combiner in the receiver. We used a maximum likelihood combiner as in [3] , its equation is:

$$\widetilde{x}[m] = \frac{y_1[m] * H_1^*[m] + y_2[m] * H_2^*[m]}{|H_1[m]|^2 + |H_2[m]|^2} \tag{2.74}$$

2.2.8 Obtained Simulation Results

Computer simulations were carried out on MATLAB software to test the performance of the decision directed channel estimation techniques. We first simulated the one dimensional method, then we extended it to two dimensional methods. Results are shown in case of no diversity, i.e., single receiver antenna, vs. the receiver diversity where two antennas are present in the receiver and a maximum likelihood combiner is used in the receiver to combine them.

The results for the one dimensional filtering is shown in figure 2.2 and its associated mean square error is shown in figure 2.3.

As shown in the figures, the decision directed channel estimation suffers from a SNR threshold, after which the bit error rate curve approaches the ideal estimation curve. This is a well-known phenomena in the decision directed channel estimation [5] and its main cause is the error propagation. If SNR value is low, the noise will affect the decisions badly and the error rate is very high. In this case, the decision directed case propagates these errors, and the estimation goes off the way.

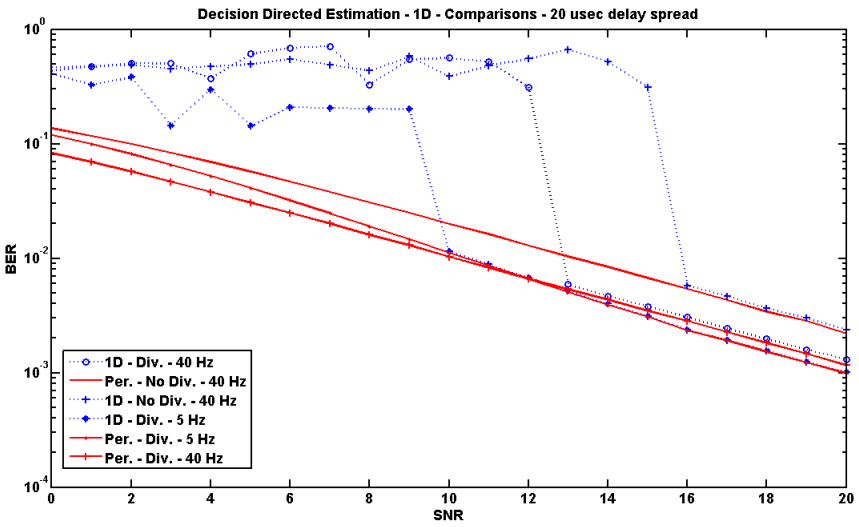

Figure 2.2: 1D Techniques - BER Comparisons

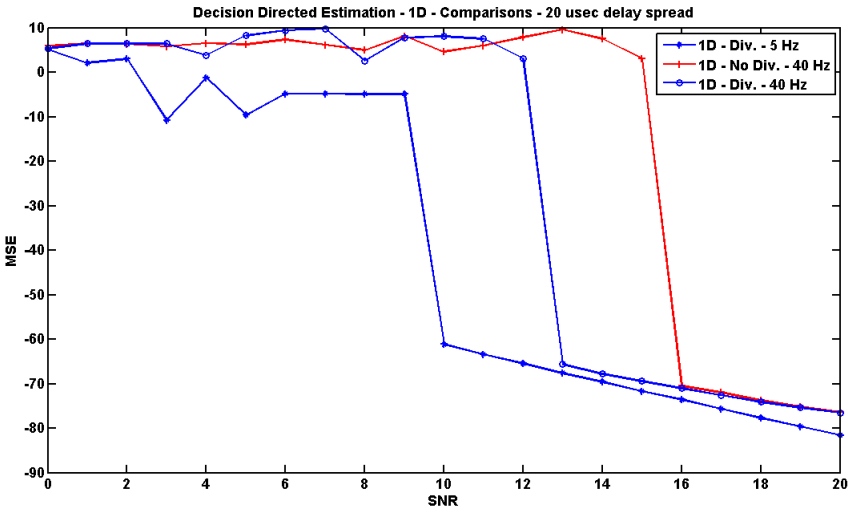

Figure 2.3: 1D Techniques - MSE Comparisons

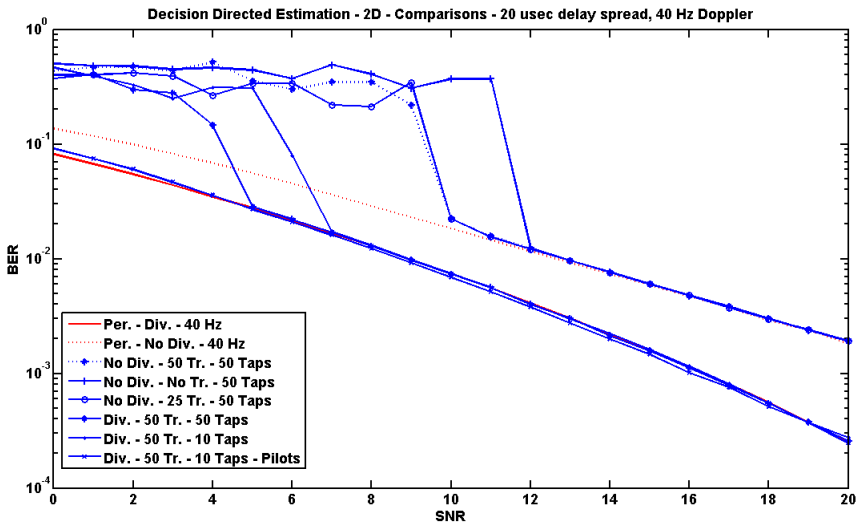

Figure 2.4: 2D Techniques - BER - Comparisons

If space diversity is not used, this SNR threshold is around 16 dB in 40 Hz doppler frequency, which is equivalent to around 17 Km/hr. If the diversity is used, this threshold drops to a value of 13 dB. In case we lowered the doppler frequency, the SNR threshold drops also. An example is shown at which 10 dB threshold appears at 5 Hz doppler frequency. The MSE curves in figure 2.3 are consistent with the bit error rate results.

If the two dimensional estimation is used, the resulting bit error rate curves are shown in figure 2.4, and the corresponding mean square error curves are shown in figure 2.5.

As shown in the figures, the 2D estimation drops the SNR threshold and is performing better than the 1D estimator. In case no diversity is used, 12 dB threshold is obtained if only one preamble is used to serve the whole OFDM frame which in this case is as long as 10000 symbols. This threshold is lowered to 10 dB's if the preamble is sent periodically every 50 symbols. It doesn't matter to increase the rate of sending the preamble above this. As shown, the same 10 dB threshold is obtained if the preamble is sent every 25 symbol. In these cases, 50 filter taps are

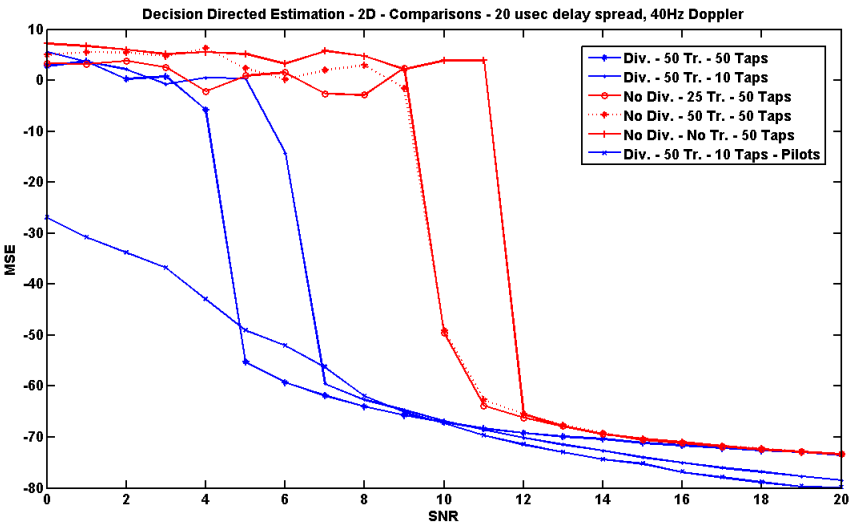

Figure 2.5: 2D Techniques - MSE - Comparisons

used in time which means that the previous 50 symbols are used to estimate the current symbol.

In case of diversity, a better threshold '7 dB' is obtained with less filter taps '10 taps'. If filter taps are increased to 50, the threshold becomes 5 dB.

In a try to remove the threshold, some pilots were added to the symbol. These pilots are well-known to the receiver and they aim at improving the estimation performance. The inserted pilots are separated by 7 subcarriers each. The total number of pilots are 8 per symbol. The pilots as well as the decisions are used in the estimation. In this case, the SNR approximately disappears. This is achieved with only 10 filter taps.

The MSE curves in figure 2.5 are consistent with these results. One comment on the MSE figures is that the more filter taps in time, the more error floor value. This is clearly shown in the figure where the 10 taps filters have less floor values than the 50 filter taps. The main cause of this is the edge effects. The 50 taps filter requires 50 previous symbols, so the first 50 symbols will have higher errors, and this will cause an error floor. The same thing will happen to the 10 taps filters, but 10 symbols only will be affected, so the error floor is less.

2.2.9 Robust Estimation

The main problem in the previous methods is that the exact correlation function should be known in order to calculate the coefficients. The correlation functions of the channel are usually unknown and need to be estimated, which results in more and more complex system. One solution to such a problem is to use a generic robust correlation function that gives acceptable mean square error for all channel correlations. This solution was proposed by Li in [3]. It states that a uniform power delay profile which is longer than the channel delay spread as well as a uniform doppler spectrum profile with doppler frequency higher than that of the channel can be used and theoretically if the total number of Wiener Filtering taps is infinite, the resulting mean square error "MSE" between the estimated and the exact channel will be the same as if the exact correlation functions were used. To calculate the correlation function in this case, we assume that the longest delay spread we have is equal to T_m, where $T_m = n_{max}t_s$, t_s is the sampling time of the system. Also, the maximum doppler frequency is f_D, substituting in equations (1.5), (1.6) with these values and setting $\sigma_k^2 = \frac{1}{n_{max}}$, we can find the robust correlation functions that can be used to get the robust filter coefficients.
In our case here, we can tolerate up to 20μ seconds spread, and $200Hz$ doppler frequency which is equivalent to vehicle speed of $86Km/hr$ at carrier frequency $f_c = 2.5GHz$.

2.2.10 Robust Estimation Results

The robust estimation was applied only to the best cases in the two dimensional filtering techniques specified above. The results are compared with the ideal cases in which the exact correlation functions are known.

Figure 2.6 shows the bit error rate curves of the different studied cases, and the resulting mean square error is shown in figure 2.7.

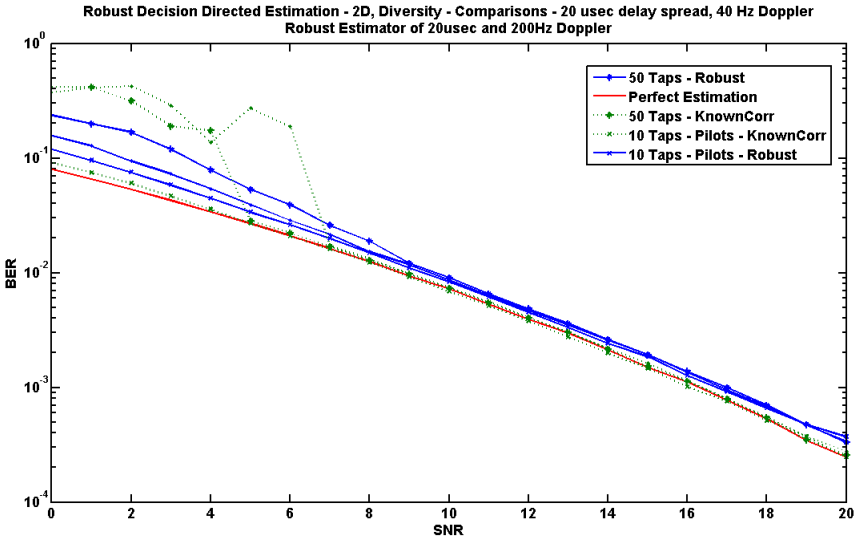

Figure 2.6: Robust Techniques - BER - Comparisons

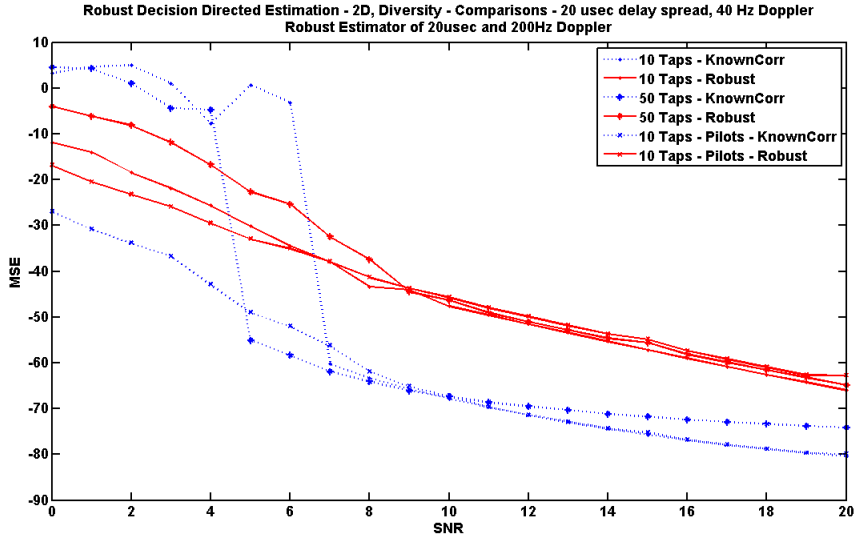

Figure 2.7: Robust Techniques - MSE - Comparisons

We can see that the robust curves are in general worse than the ideal curves by an amount of approximately 1 dB. The interesting thing here is that the SNR threshold disappeared totally. This is because the filter coefficients in the robust case are like a sinc function that vanishes after certain number of coefficients, but the exact coefficients are dependent on the correlation function shape, which is periodic in general.

To clarify this, let's talk about our case here. The two taps channel has a sin wave correlation function. In this case, the coefficients are in a sin wave function shape, which means that to estimate the channel at the n location, the filter will depend on all the subcarriers around n either near or far from n. In the robust case, the nearest locations only are taken, because the coefficients are vanished after sometime. If the SNR value is low, the errors are high and the decisions are very inaccurate. The ideal estimations in this case will give very faulty estimations because error propagation is magnified which is not the case in the robust estimation, which will have somehow better estimations in this case.

2.3 Conclusions

In this chapter, we derived the ideal, 1D and 2D filter coefficients for MMSE technique in OFDM systems. These techniques were applied to an OFDM system, and the results were obtained. The main conclusions that we reached from the simulations are:

1. Decision directed channel estimation methods always suffer from SNR thresholds.

2. SNR thresholds can be lowered significantly using one of the following methods:

Using 2D filters instead of 1D filters.

Using space diversity in the receiver.

Inserting a preamble periodically between every OFDM frame.

Increase the filter taps. But higher error floors may be obtained.

3. SNR thresholds can be removed if pilots were used beside the normal decisions.

4. Robust estimation has a 1 dB performance degradation than the ideal estimation, but it is better in the low SNR region.

Chapter 3

Introduction to WiMAX

3.1 Overview

WiMAX stands for 'Worldwide Interoperability for Microwave Access'. It is a wireless system based on OFDM, OFDMA, and SOFDMA. It is mainly used to transmit very high speed data 'broadband access' in a wireless medium. This is done through a mobile phone, a portable computer, or any other wireless device.

The WiMAX system is based on the IEEE 802.16 standard. Two widely used versions of this standard are available, one for the fixed WiMAX [6], and the other is for mobile users [7]. The WiMAX technology should be able to provide users with very fast internet connections, that can replace today's DSL and cable connections. This system is partially contributing to the next generation 4G systems for cellular phones.

The history of WiMAX starts when the IEEE 802.16 group was formed in 1998. Their aim was to develop a new standard that is based on LOS point to multipoint wireless communication. They aimed to cover the RF frequency range between 10 GHz to 66 GHz. The first standard was completed in 2001 and was based on single

carrier physical layer, with a TDM MAC layer.

After that, the group made an amendment to the standard to support the NLOS applications. In this case, multipath fading channels are severely destroying the performance of the system mainly because the ISI resulting from the multipath channel. Due to this, the PHY layer changed to be based on OFDM. MAC layer is based on OFDMA, which we will talk about in the next section.

Over the years, the current fixed and mobile WiMAX standards, which are the IEEE802.16d-2004 'fixed' and the IEEE802.16e-2005 are used. They support multiple PHY layer mechanisms, which are:

1. **Wireless MAN SC** a single carrier PHY layer, beyond 11GHz, requiring LOS conditions

2. **Wireless MAN OFDM** a 256 point OFDM PHY layer in the range from 2GHz to 11GHz, intended for NLOS conditions, and dedicated to fixed applications.

3. **Wireless MAN OFDMA** a scalable OFDMA PHY layer from 128 up to 2048 point, in the range from 2GHz to 11GHz, intended for NLOS conditions, and dedicated to mobile WiMAX.

The WiMAX system has many interesting features, including -but not limited to- :

1. Very high data rates, up to 74 Mbps at 20 MHz bandwidth.

2. Scalable bandwidth and data rates, to support multiple channel profiles for multiple bandwidths.

3. Adaptive modulation and coding that supports many modulation/coding schemes for different channel conditions.

4. Flexible dynamic user allocation, according to the demands, in a TDM scheme.

5. Support for multiple antenna techniques including MIMO support.

In the following sections, we will describe in details, the inner structure for the physical layer of the WiMAX standard, and how it supports multiple channel profiles, in order to be able to do correct channel estimation on the system, which is the aim of this thesis.

3.2 Introduction to SOFDMA

OFDMA stands for Orthogonal Frequency Division Multiple Access. It is a multiple access version of the well-known OFDM systems, where multiple users are sharing the resources.

Figure 3.1 shows an example of an OFDMA symbol. As shown, the subcarriers are divided among the different users such that each user is allocated a portion of the subcarriers. Usually, a single user is allocated a group of subcarriers that are not contiguous to each other. This to avoid the complete loss of information for this user if the channel is bad in this band, and provide multi-user and frequency diversity.

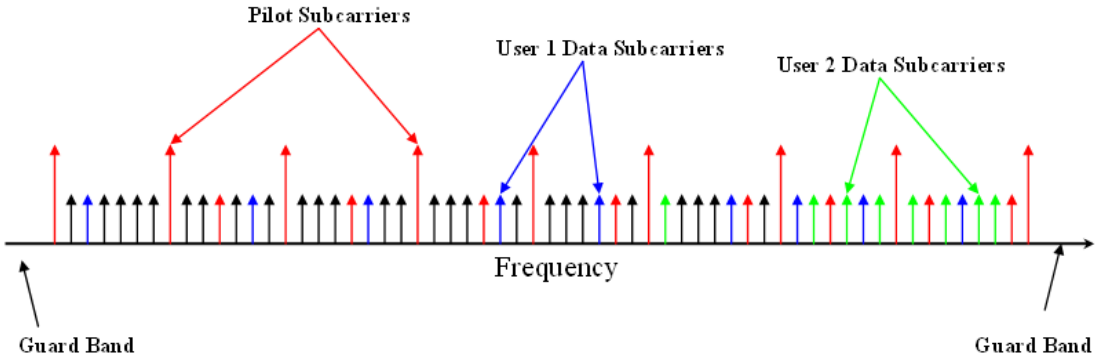

Figure 3.1: OFDMA Illustration

The OFDMA system allows flexible user data rates according to demand. Also, it averages the interference from neighboring cells by using different permutation scheme -which is the scheme for dividing the users between the available subcarriers- for different users in the contiguous cells. It offers frequency diversity by spreading the subcarriers all over the spectrum.

Scalable OFDMA 'SOFDMA' is a modified version of the OFDMA where the FFT size varies according to the bandwidth. If the bandwidth is increased, the FFT size is also increased and vice versa. This makes the FFT symbol duration fixed, which makes the system robust to multipath and varying delay spreads and various doppler spectrum's. This makes the subcarrier spacing constant with a value chosen to trade-off the fast varying channels in time and frequency as well as providing flexibility to users moving between different environments with different bandwidths.

The basic concept for the OFDMA in WiMAX is to keep the subcarrier spacing constant in a value that can tolerate the maximum delay spread occurring in a channel. The maximum delay spread is defined in the ITU-R M.1225 document [8] and is equal to 20 μ sec. By this, we can deduce the coherence bandwidth which is the bandwidth in which the channel is still correlated. This coherence bandwidth is defined in terms of the delay spread T_m by the following equation [9].

$$B_c = \frac{1}{5T_m} = 10KHz \tag{3.1}$$

If the bandwidth is increased for example, and the FFT size didn't change to maintain this subcarrier spacing, then the subcarrier spacing will increase and the subcarriers will be less correlated and suffer from the selective channel. This is translated in time domain in a shorter symbol duration, in which the delay spread affects it badly. This is very problematic if the channel delay spread is large and the channel is highly selective. On the other side, if the bandwidth became smaller, while maintaining the same FFT size, no problem in frequency selectivity/delay spread, but the symbol time will increase and it will be largely affected by the doppler spectrum and the doppler shift of the channel. The value of 10 KHz frequency spacing is the trade off between the high selective and high doppler channel. It is claimed in [10] that this value can tolerate up to 20 μ sec delay spread with a mobile user at a velocity of 120 Km/h.

3.3 OFDMA Symbol Structure

The OFDMA symbol is mainly composed of three types of subcarriers. These are:

1. **Data Subcarriers** are modulated and carrying the different users data symbols.

2. **Pilot Subcarriers** are known subcarriers to the receiver, usually power boosted, to be used in the channel estimation, detection and synchronization issues.

3. **Null Subcarriers** are zeros at both edges of the symbol as well as in the middle subcarrier. The middle subcarrier -called DC subcarrier- is set to zero for not having DC excess power in the receiving amplifiers, and the edges null bands are to avoid interference with adjacent bands.

As shown in figure 3.1, the data subcarriers are distributed among all users, pilots are boosted with respect to data. Guard bands are available at the edges of the symbol. This is the general structure of the OFDMA symbol.

3.4 OFDMA Frame Structure

The frame structure of the WiMAX OFDMA scheme is usually sent in a Time Division Duplex mode 'TDD' and is composed of two main parts, the downlink subframe, and the uplink subframe.
The downlink subframe is the data sent from the base station to the mobile station. It is composed mainly of the following:

1. **Preamble** is the first symbol of the frame, one third of its subcarriers is BPSK modulated and the others are set to zeros. Its main aim is to provide initial channel estimation and synchronization.

2. **Frame Control Header** or FCH, provides the mobile station with some downlink configurations such as the modulation scheme and coding rate used.

3. **DL-MAP** or the downlink map messages. This tells the mobile station which subcarriers are allocated to him within the FFT. This is called the user's data region.

4. **DL-Burst** is the group of symbols which are carrying the data to the user.

The uplink has similar structure to the downlink, and in addition, it has another important group of subcarriers called ranging subcarriers. The ranging subcarriers are sent from the mobile station to the base station, and the base station can deduce from it the channel conditions at the mobile side. If channel conditions are

good, the next burst will have higher modulation schemes, if not, it will have lower modulation schemes, and so on.

3.5 Permutation Schemes

Permutation schemes are the methods by which the base station allocates the subcarriers to different users. This is made in WiMAX by combining some group of subcarriers into one group called a subchannel. The subchannel is composed of either distributed or adjacent subcarriers. Each user is allocated one or more subchannels according to the permutation scheme used. The permutation schemes are discussed below.

3.5.1 Downlink FUSC

In this permutation scheme, each subchannel is composed of 48 data subcarriers which are distributed along the OFDM symbol. The number of subchannels per OFDM symbol differs with respect to the FFT size. This is clarified in detail in the standard.
It is noted here that the subchannels are allocated with subcarriers from only one OFDM symbol. The pilot locations are preset before dividing the symbol into subchannels.

3.5.2 Downlink PUSC

Here, subcarriers are divided into clusters, each of them is distributed along two OFDM symbols. Each cluster consists of 14 adjacent subcarriers, two of them are pilots. The clusters are then divided into groups, and a subchannel is composed of two clusters from each group.

In this permutation scheme, pilots are allocated with respect to the cluster, or by other means, they are allocated with respect to the subchannel. This is different from the FUSC permutation scheme.

3.5.3 Uplink PUSC

In this case, the subcarriers are divided into tiles. Each tile consists of 12 subcarriers spread over 3 symbols. This means that a tile is 4 subcarriers by 3 symbols. Tiles are renumbered and divided into 6 groups. A subchannel is composed of 6 tiles from the same group.

3.5.4 Band Adaptive Modulation and Coding

In this scheme, the subchannel is composed of adjacent subcarriers. Subcarriers are first divided into bins, each bin is composed of 8 data subcarriers plus one pilot subcarrier.
The subchannel may have different shapes. It can be one bin over six symbols, two bins over three symbols, or six adjacent bins over one OFDM symbol.

Chapter 4

Channel Estimation Techniques in WiMAX

In this chapter, we will investigate mainly different pilot assisted channel estimation methods for OFDM, applying them directly to the WiMAX case. We will begin by showing that decision directed methods mentioned in the previous chapter can't be used in the highly mobile schemes. We will be using pilot assisted methods. First, we will be illustrating our system model for WiMAX, then we will review the estimation techniques as well as the simulation environment used, then we will proceed to the simulation results and conclusions.

4.1 Decision Directed vs. Pilot Assisted Estimation

As we mentioned in the previous chapter, the decision directed channel estimation uses the current decisions to produce an initial channel estimates, then these estimates are enhanced using Wiener Filtering techniques. In this chapter, we will focus our work on realistic OFDM systems, targeting mainly mobile WiMAX wireless standard.

Two main issues exist in wireless standards, that makes the channel estimation much more difficult. First, these standards are usually targeting high throughput, which requires less number of pilot subcarriers to be transmitted in the preamble -the start of each frame- as well as the remaining symbols. As the number of pilots decreases, the channel estimation becomes much more difficult. The other challenging issue is that the wireless channels targeted by these new standards are usually harsh, with very high selectivity in the frequency direction, and very rapid changes in the time direction. The high selectivity is a direct result of the long multipath delay spread, and the rapid time changes is a direct result of the very high mobile speeds supported by these new standards.

These challenges make a question mark on whether the decision directed estimation will perform well given these harsh conditions. This question is straightforward, since the decisions in such cases will not be reliable any more, and the high decision errors encountered in these harsh channels, will result in much more bad estimates.

We have two main methods that we must choose from while making our estimations. We have the pilot assisted methods which only relies on pilots and no decisions are used, or using the decisions with the pilots in the preamble as initial channel estimates. To decide this, we made a comparison between the two methods using the ideal Wiener Filters described in the previous chapter. We used the WiMAX system as our simulation environment. Detailed discussion about the system model will be shown later in this chapter. What we are concerning about now is which method we will use in the remaining of our work.

The resulting bit error rates of the comparison are shown in figure 4.1 while varying the signal to noise ratio of the transmitted signal from 0dB to 40dB. Figure 4.2 show the corresponding mean square error curves. In these simulations, we

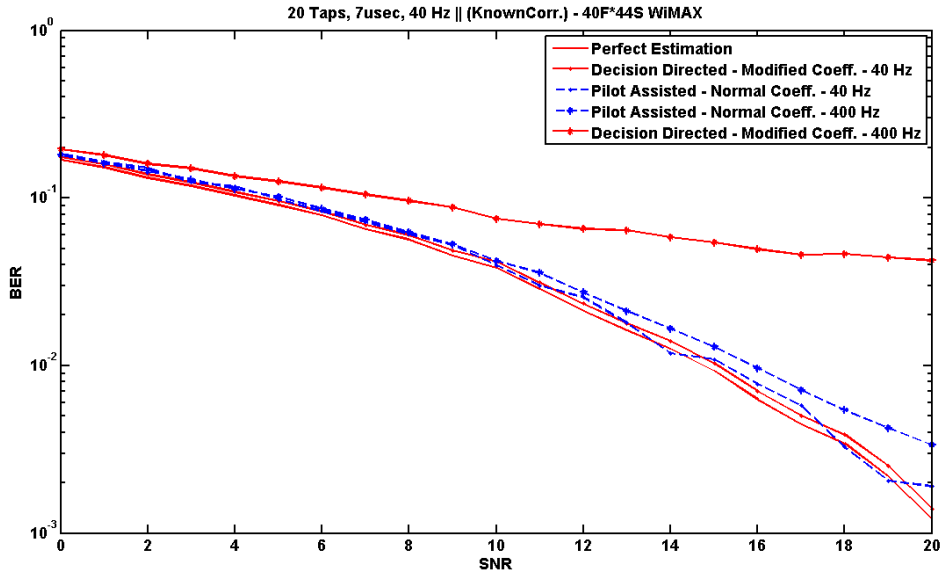

Figure 4.1: BER - Decision Directed vs. Pilot Assisted

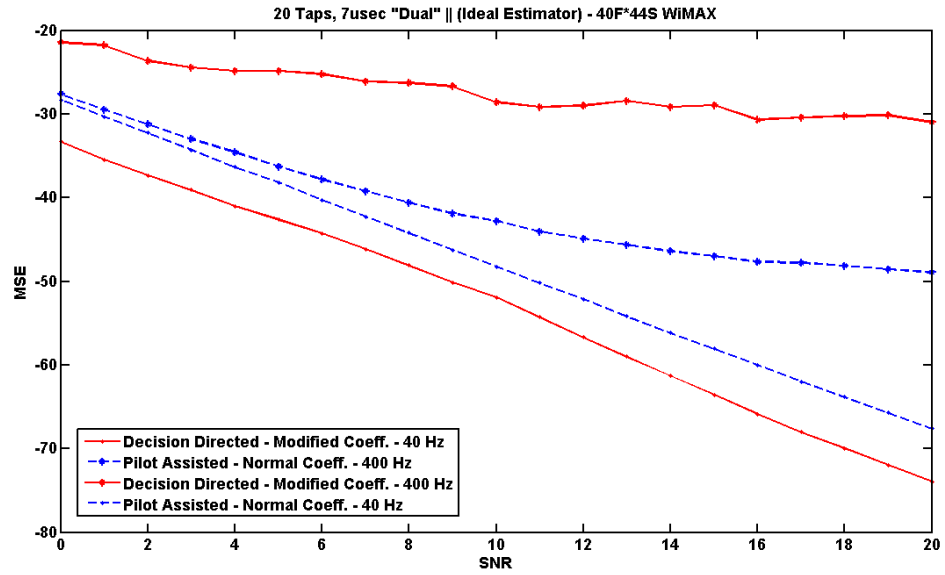

Figure 4.2: MSE - Decision Directed vs. Pilot Assisted

modified the decision directed estimator coefficients previously deduced in chapter 2, in order to cope with the WiMAX preamble structure in which one third of the subcarriers are available, and the remaining subcarriers are set to zeros.

We can see that in the low doppler channels, which correspond to the low mobile speeds, decision directed estimation has better performance than the pilot assisted methods. This is apparent in the mean square error curves, and is not clear in the bit error rate curves because the hard decision device in the receiver can tolerate some errors in the estimation. In cases of high mobile speeds, decision directed estimation shows bad performance compared to the pilot assisted methods. This is because the decisions are not reliable any more.

The conclusion from this section is that the best method to use in WiMAX and other highly mobile wireless standards are the pilot assisted methods as the pilots are constant and don't depend on the channel state or the mobility. The decisions in the decision directed methods can be reliable or not depending on the channel state. This is the same conclusion that Li reached in his work [11], which is that at high doppler frequencies, decision directed estimation has degraded performance compared with the pilot assisted methods. This is why we are focusing in the remaining of this chapter on different pilot assisted methods targeting mobile WiMAX.

4.2 System Model

The system block diagram is shown in Figure 4.3. The transmitter is basically composed of a Mapper to modulate the data, then pilots are added and the whole OFDM symbol is transformed to the time domain through the IFFT block. Let the input to the IFFT block is $x[m]$ where $m \in \{(n_i, k_j)\}$; i=1:NSymbols, j=1:Ndata Here NSymbols refer to the total number of OFDM Symbols in a frame, and Ndata refers to the total number of data subcarriers in each OFDM symbol, n, k are the

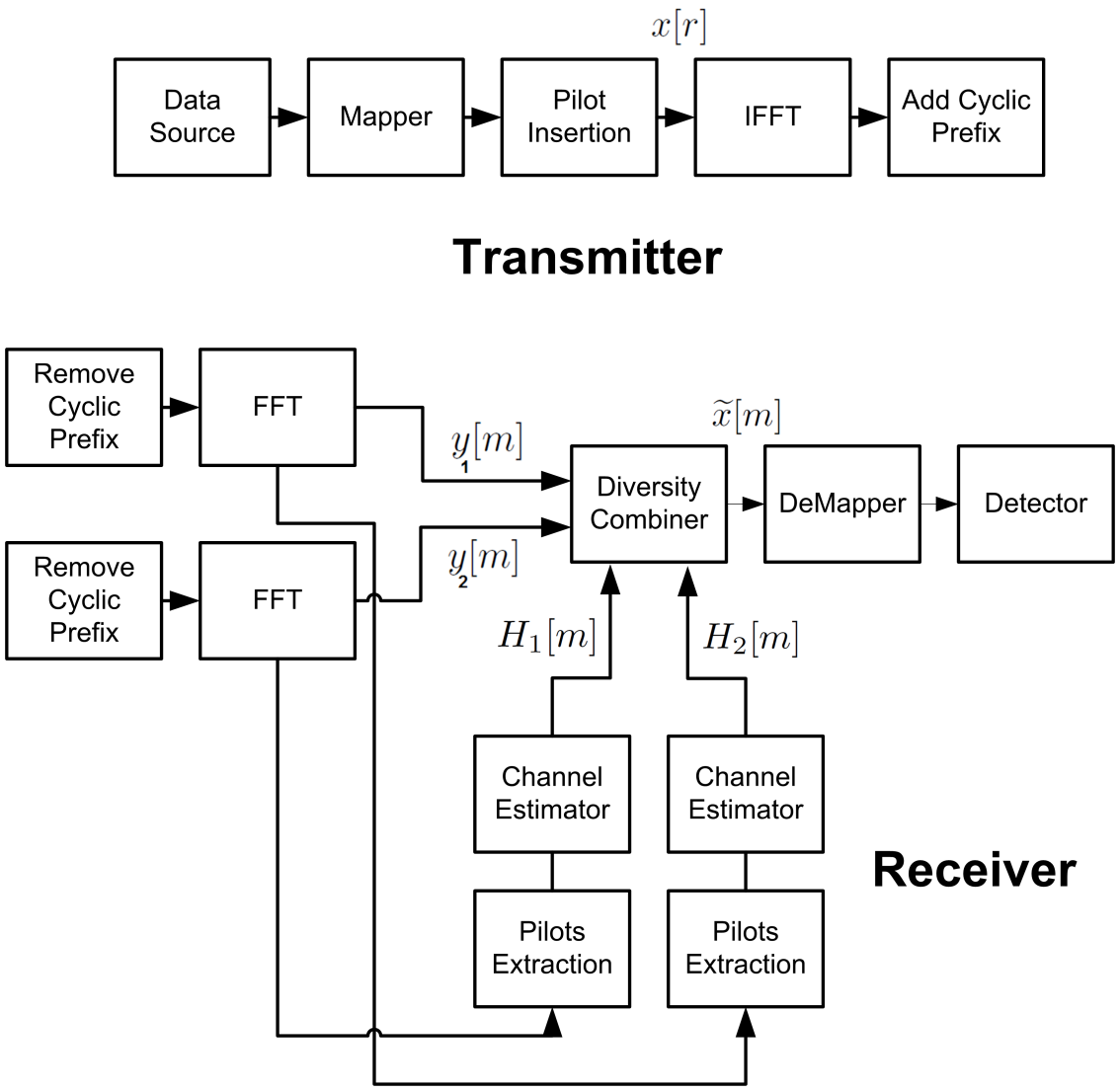

Figure 4.3: System Block Diagram

time and frequency positions on the time/frequency grid.
At the receiver, the output from the FFT block can be written as:

$$y[m] = x[m]H[m] + n[m] \tag{4.1}$$

Where $H_i[m]$ is the complex channel frequency response of the i^th antenna at the point m on the time/frequency grid. The pilot extraction block extracts the pilot subcarriers from the FFT output and produces initial LS estimates $\widetilde{H}[m_p]$ at the pilot locations where:

$$\begin{aligned} \widetilde{H}[m_p] &= x^*[m_p]y[m_p] \\ &= |x[m_p]|^2 H[m_p] + x^*[m_p]n[m_p] \end{aligned} \tag{4.2}$$

Where $m_p \in P$ and P is the set of all pilot locations on the time/frequency grid. Our system uses two antennas in the receiver to utilize space diversity. This requires a diversity combiner in the receiver. We used a maximum likelihood combiner [3] , which is given by:

$$\widetilde{x}[m] = \frac{y_1[m] * H_1^*[m] + y_2[m] * H_2^*[m]}{|H_1[m]|^2 + |H_2[m]|^2} \tag{4.3}$$

In this work, we are targetting the downlink channel estimation. The same time/frequency grid which is used in mobile WiMAX 802.16e PUSC permutation scheme [7] is used. In this permutation scheme, the OFDM symbol is divided into clusters, and each cluster has two pilots in different locations depending on whether it is an even or odd symbol. The cluster structure is shown in figure 4.4

4.3 MMSE Robust Estimator

MMSE algorithm is the optimum algorithm for minimizing the mean square error of the estimation [12]. It is based on Wiener Filtering approaches [5]. In this

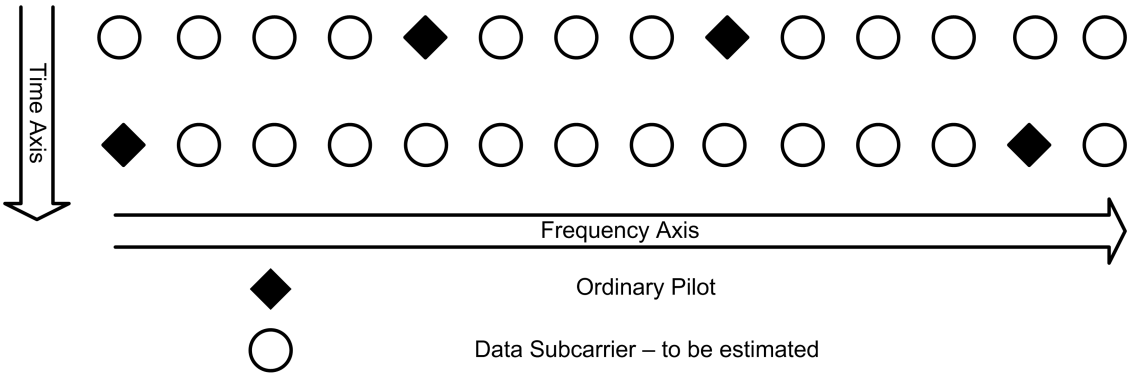

Figure 4.4: Cluster Structure

section, we will review briefly the concepts of Wiener Filtering that we will use in the subsequent sections.

Our problem is as follows: We need to estimate the complex channel frequency response $\widehat{H}[r]$ at the r location on the time/frequency grid, where $r \in \{(n_i, k_j)\}; i = 1 : NSymbols, j = 1 : Ndata$

To make this estimation, we will use the nearest N_{tap} pilots to the location r from the complete set of pilots P, and multiply each LS estimate $\widetilde{H}[m_1]; m_1 \in P_r$ by a complex coefficient $c[m_1; r]$ according to the following equation:

$$\widehat{H}[r] = \sum_{m_1 \in P_r} c[m_1; r]\widetilde{H}[m_1; r] \tag{4.4}$$

Here $[m_1; r]$ represents the indices of the nearest N_{tap} pilots to the required subcarrier r, The set P_r represents the N_{tap} pilot locations near r. $\widetilde{H}[m]$ can be calculated as in (4.2)

In order to calculate $c[m_1; r]$, the orthogonality principle [4] is used which states:

$$E[(\widehat{H}[r] - H[r])\widetilde{H}^*[m]] = 0, \forall m \in P_r \tag{4.5}$$

If the correlation of the channel is known, the above equation leads to:

$$C(r) = \frac{1}{|x[r]|^2}(R^{-1}_{\widetilde{H}}[r]r_H[r]) \tag{4.6}$$

Where $C(r)$ is a column vector containing all the $Ntaps$ coefficients required to calculate $\widehat{H}[r]$, $r_H[r]$ is the correlation vector of the channel frequency response between data location and different pilot locations. $R_{\widetilde{H}}[r]$ is the correlation matrix between different LS pilot estimates within P_r which is equivalent to:

$$R_{\widetilde{H}}[r] = R_H[r] + \rho_p I \tag{4.7}$$

$R_H[r]$ represents the correlation matrix of the channel frequency response between different pilot locations.

The main problem in this equation is that the exact correlation function should be known in order to calculate the coefficients. The correlation functions of the channel are usually unknown and need to be estimated, which results in more and more complex system. One solution to such a problem is to use a generic robust correlation function that gives acceptable mean square error for all channel correlations. This solution was proposed by Li in [3]. It states that a uniform power delay profile which is longer than the channel delay spread as well as a uniform doppler spectrum profile with doppler frequency higher than that of the channel can be used. Theoretically if the total number of Wiener Filter taps is infinite, the resulting mean square error "MSE" between the estimated and the exact channel will be the same as if the exact correlation functions were used. To calculate the correlation function in this case, we assume that the longest delay spread we have is equal to T_m, where $T_m = n_{max}t_s$, t_s is the sampling time of the system. Also, the maximum doppler frequency is f_D, substituting in equations (1.5), (1.6) with these values and setting $\sigma_k^2 = \frac{1}{n_{max}}$, we can find the robust correlation functions that can be used to get the robust filter coefficients.

In our system, we can tolerate up to 20μ seconds spread, and $300Hz$ doppler frequency which is equivalent to vehicle speed of $130Km/hr$ at carrier frequency $f_c = 2.5GHz$. We set the maximum delay spread slightly larger than 20μ seconds $T_m = 20.5\mu$ seconds to account for the leakage if one or more of the tap delays τ_k is not an integer multiple of the sampling time t_s [12].

4.4 Related Work

In this section, we will describe briefly the previous work that was done in channel estimation for OFDM systems,and we will concentrate on WiMAX systems, so as to be able to compare our work and proposals with the related work in the literature.

As we stated in the introduction, the channel estimation for OFDM systems in general can be classified into two main methods [5], the Decision Directed and the Pilot Assisted channel estimation. Decision directed methods depend on the previous decisions in the estimation of the current channel. All the previous decisions are used as pilots to estimate the channel frequency response at their locations using multiple methods ranging from normal LS estimates up to MMSE filtering estimation which is the best estimator for such a case. In [3], the Decision Directed channel estimation method was studied using MMSE filtering technique. It was shown that the Decision Directed method is prone to a well-known error propagation phenomena that affects greatly the performance. One proposed solution to such a problem is to depend on either the reference undecoded or the decoded decisions to benefit from the error correcting codes to have more reliable decisions. It is shown in [3] that the decision directed estimation method introduces an SNR threshold in the bit error rate and mean square error curves. After this threshold, the bit error rate and mean square error reaches its optimum value as if we know the channel.

It is shown in [13] that there is a great performance degradation in the decision directed estimation in cases of high mobility. In this case, the more convenient method is the pilot assisted case which depends only on the pilots that are already sent in each OFDM symbol. As the number of pilots increases, the estimation is better but with the cost of more overhead. The MMSE filtering method using only pilots is described in detail in [13]. We will make a review on this method later on in this paper.

In the WiMAX case, the estimation methods investigated are utilizing the pilots in the clusters in different ways. Most of the work is trying to interpolate between the pilots in different ways including linear, second order, cubic, spline and MMSE interpolations. We will show some examples of the work done below.

In [14], piecewise linear interpolation was tested against cubic, spline and Newton Lagrange polynomial interpolation. The channel used had a 2.5μsec delay spread. The linear interpolation performed better than all other interpolation methods due to the fact that the other interpolation methods introduce unexpected oscillations in the estimations that are not found in the original low selective channel under test. The authors proposed a modified linear interpolation method that utilizes all the pilots in the cluster while estimating each data symbol. They scale each pilot with a factor according to its location. Their method performed better especially in higher mobile speeds.

The same idea of linear interpolations vs. other non linear interpolation was tested in other papers such as: [15], [16]. All of these papers reach the same conclusion as above, utilizing a very low selective channel. For example, in [15], the channel has zero delay spread, which means that it is flat in the whole band.

In [17], the authors proposed another estimation method that makes linear interpolation in time, which is a simple averaging operation, and then makes a 3rd order polynomial fitting in frequency. In their work, they utilized the adjacent clusters in time and frequency and made their interpolation/polynomial fitting based on a sliding window on the adjacent clusters. They tested this against utilizing only one cluster a time, which showed more errors at the edges of the clusters. The channel they used had a 6μsec delay spread, with mobility of 100Km/hr.

In [18], the authors study the initial estimation problem using the preamble. The

preamble in WiMAX has one single pilot every three subcarriers. Two subcarriers have zero value and the third is the pilot. They studied the linear interpolations vs. transform domain, MMSE and robust MMSE methods. Transform domain method is a simple two stages transforms, the first converts the symbol to the transform domain, then the second returns it back to the original frequency domain. MMSE filtering is the optimum solution of the problem, and the robust MMSE is an approximation to the exact MMSE while not knowing the channel statistics. We will describe the robust MMSE later. The authors tested these methods in two channels one with a 2.5μsec delay spread and the other with 20μsec spread. All the methods succeeded in the lower selective channel with the MMSE filtering and the robust MMSE having the superior performance. At the high selective channels, all methods failed except the MMSE and the robust MMSE. The authors adopted the robust MMSE as the most practical and convenient method in the high selective channels.

It is important to realize that the WiMAX system should support channels with a delay spread up to 20μsec. This is clearly stated in [10] where the scalable OFDMA subcarrier spacing was fixed at a value of 10 KHz approximately for WiMAX systems, in order to support delay spreads up to 20μsec.

4.5 Studied Channel Estimation methods

In this section, we will briefly describe each studied channel estimation algorithm. All algorithms discussed here are utilizing the LS estimates in equation (4.2) in different manners to produce the channel frequency response estimate $\widehat{H}[r]$ for all r data locations on the time/frequency grid. It is important to mention that in our work, we assume that the mobile station has a built in FFT engine, which means that all the clusters -whether they are his own clusters or other user's clusters- are available. The estimation methods described in this paper utilize all the clusters in

the symbol.

4.5.1 Cascaded one dimensional time/frequency interpolation

This method is widely used in channel estimation as in [14], [15]. As shown in figure 4.5, one dimensional time interpolation is used in the time axis to get the in between channel frequency response. These channel frequency responses are used as new pseudo pilots, and one dimensional frequency interpolation is made to estimate the channel frequency response in every OFDM symbol.

By using this method, both time and frequency correlations are utilized in a very simple manner, and the frequency interpolation is now based on increased number of pilots. In WiMAX systems, the time interpolation is a simple averaging operation which is very simple. This is the case in our system.

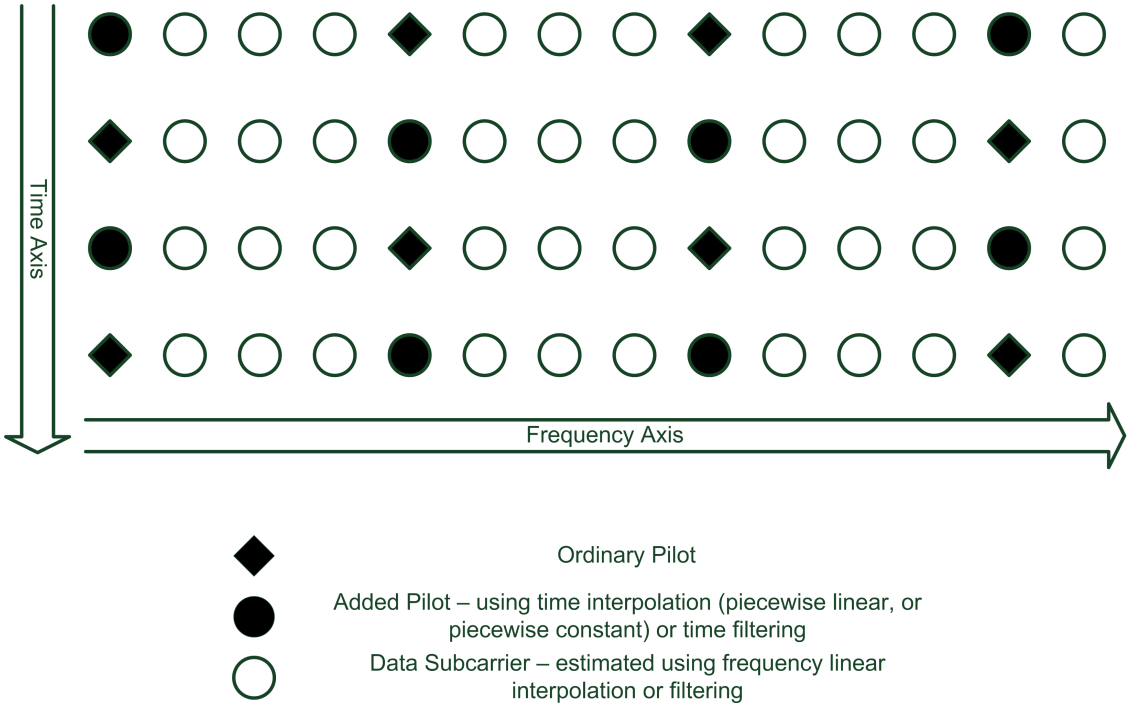

Figure 4.5: Cascaded Filtering/Interpolation Illustration

4.5.2 One dimensional frequency Filtering

This method is described in detail in [5]. In this method, the FIR filter described in (4.4) is used to estimate the channel frequency response. It uses the nearest N_{tap} LS pilots in the same OFDM symbol as the input data to the filter. The filter coefficients are calculated using equation (4.6) using only robust frequency correlation function 'previously mentioned in equation (1.5) and setting the robust time correlation function to unity because we are not using LS pilots from other OFDM symbols. Using the robust correlation functions makes it easy to store these coefficients in a memory, and thus the hardware implementation becomes more simpler.

4.5.3 Two dimensional time/frequency Filtering

In this method, both the robust time and the frequency correlations are used to enhance the performance. This method is well known and described in the literature [5], [13] for OFDM systems. We extend this here by applying it to WiMAX systems and comparing the results with other methods. We are using 2D Filtering in two approaches which are:

2D Filtering on two consecutive OFDM Symbols

Here, the FIR filter described in (4.4) utilizes the nearest N_{tap} LS pilots in the current symbol n and the next OFDM symbol $n+1$. In the last OFDM symbol of the frame, we use the current symbol n as well as the previous symbol $n-1$. By using LS estimates from two successive symbols, time correlation is well-exploited because the correlation between two consecutive symbols is still high. This algorithm is illustrated in figure 4.6

Using equation (4.6) with robust correlation functions in both time and frequency, we can calculate the coefficients that are multiplied by the LS estimates at pilot locations. These coefficients don't change and so, they can be stored in a memory

to simplify the hardware architecture of the system.

It is clear that in this algorithm, the time correlation function used has only two values which are $r_t[0], r_t[1]$. This depends on whether the used LS pilot lies in the current or in the next OFDM symbol.

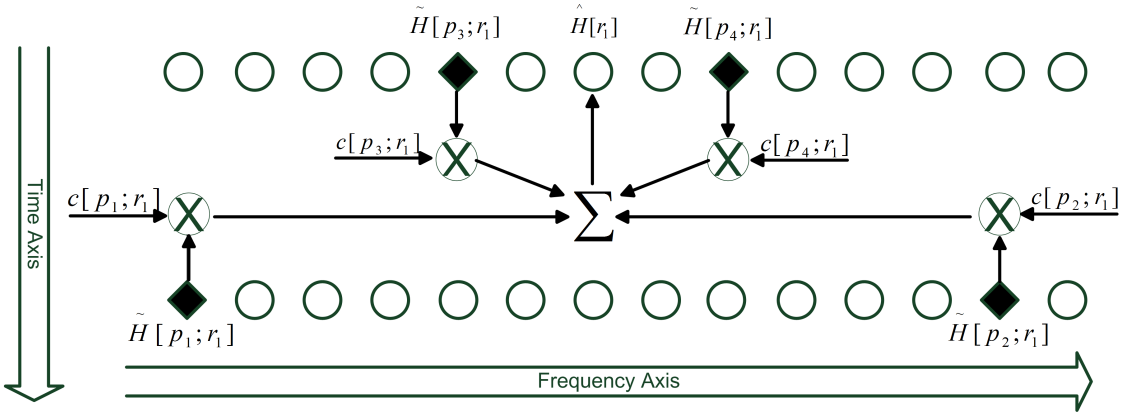

Figure 4.6: 2D Filtering on two symbols illustration

2D Filtering on three consecutive OFDM Symbols

Here, the same previous algorithm is applied with minor modifications. The FIR filter described in (4.4) utilizes the nearest N_{tap} LS pilots in the current symbol n, the previous symbol $n-1$ and the next symbol $n+1$. In the first and last OFDM symbol of the frame, the first and last three symbols in the frame are used instead. By using LS estimates from the nearest three symbols, time and frequency correlation is best-exploited because we are using the highest correlated LS pilots nearby.

4.5.4 Cascaded one dimensional time/frequency filtering

In this method, we use the MMSE filtering technique described in (4.4) twice. First of all, we do one dimensional filtering in the time direction to get the pseudo pilots in figure 4.5. Then, those obtained pseudo pilots are utilized in addition to the normal pilots in each symbol, and a second filtering operation is made to get the

remaining data subcarriers in each symbol.

This method is used to simplify the two dimensional filtering operation described before. The simplified cascaded one dimensional filtering technique described here has better performance than the normal two-dimensional filtering technique for the same complexity [5]. The only problem in this method is that it causes large latency in the hardware implementation because you should wait until certain number of OFDM symbols are received, in order to make the time filtering operations. The more OFDM symbols you wait for, the more latency you get, but the better performance because more filter taps are involved in time.

Another issue in this method is that we are required to store the robust coefficients for two filters. Two sets of coefficients need to be stored, one set for time filtering, and the other for frequency filtering. It is clear from equations (1.6), (4.6) that the time coefficients are real. Frequency filtering coefficients can be made real in the expense of degraded performance [5]. In our work, we keep the robust frequency correlation functions complex.

4.6 Proposed Method

It is clear that the best methods -with respect to performance- from those stated in the previous section are the cascaded filtering and the two dimensional filtering based on three consecutive symbols. The cascaded filtering is very impractical because it introduces large hardware latency. If we have six filter taps in time for example, we should wait complete 12 symbols in order to start our estimation. This is a difficulty in the algorithm. Another difficulty is that we should need much larger memories. Two sets of coefficients should be stored, one for the time filtering operation, and the other for the frequency filtering operation.

We propose a simpler and more practical modification to the cascaded filtering method. The time filtering is replaced by time interpolation or "time averaging".

Time interpolation is used to get the pseudo pilots illustrated in figure 4.5. These pseudo pilots in addition to the normal pilots are passed to the one dimensional filter described in (4.4) using robust frequency correlation functions to get the remaining data subcarriers in each symbol.

The main advantage of this method is that it has very little latency compared to the cascaded filtering operation, and much less complexity than the two dimensional filtering method. The time interpolation is a simple add and shift operation except for the edges. The frequency filter can be made of lower order than that of the two dimensional filter case.

Another important proposal is that we don't need to estimate the SNR value to make the frequency filtering. As we will see in the simulations, we fixed the SNR value in the filter equation (4.4) with a nominal value of 20 dB which is an average value between the used SNRs. Simulations proved that this approximation has no effect on the performance.

As we will see later, the main problem of all frequency filtering methods is that the edges of the OFDM symbol are more prone to errors than the middle of the symbol. This is normal since in the middle of the symbol, the frequency filter is using pilots and pseudo pilots from the two sides of the location to be estimated. On the other hand, this is not the case on the edges. At the leftmost edge of the symbol, only the pilots and the pseudo pilots at the right side of the needed location are used. The same thing applies to the rightmost edge of the symbol.

To overcome this problem, two extra pilots are added on each side of the symbol. The cluster structure within the middle of the symbol is not changed and remains as in figure 4.4. The cluster structure at the leftmost edge is as in figure 4.7, and at the rightmost edge is as in figure 4.8

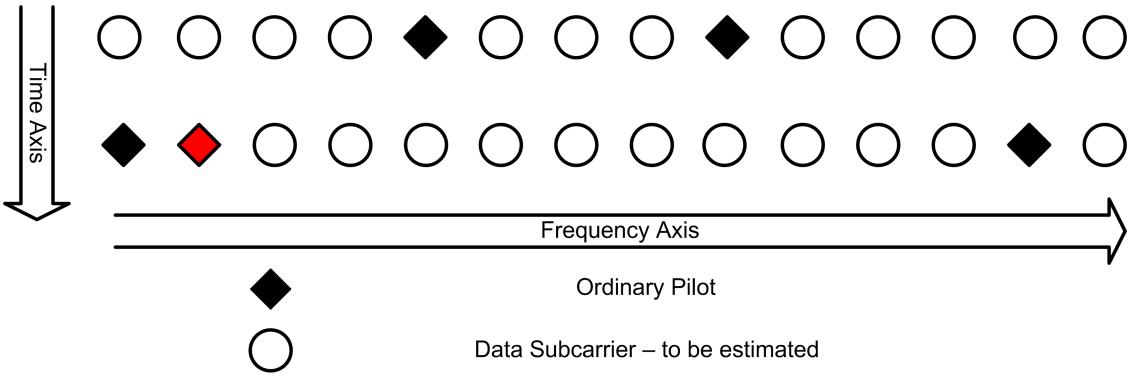

Figure 4.7: Cluster structure at the leftmost of the symbol

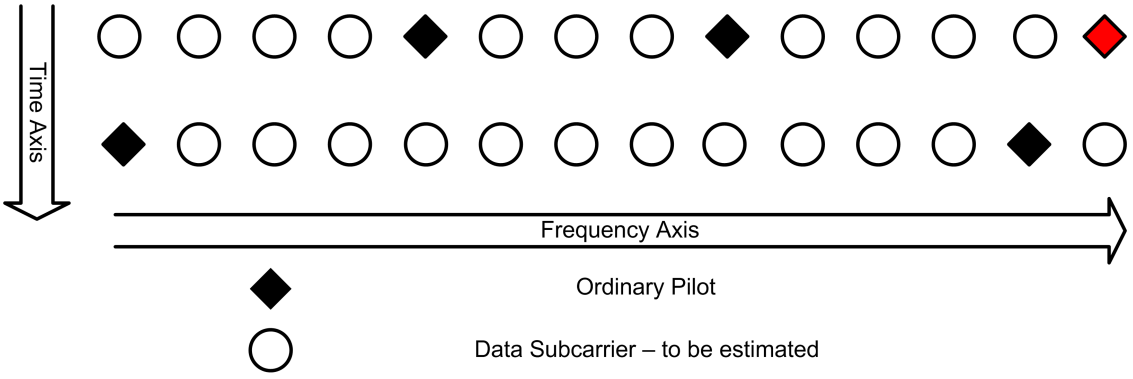

Figure 4.8: Cluster structure at the rightmost of the symbol

By making this minor modification in the pilot structure, we totally eliminate the edge problems, since after the time interpolation, there will be two pilots/pseudo pilots at each edge of the symbol. These two pilots will act as a guard to the edge regions, and the edge problem will be eliminated. This will be verified in the simulations.

4.7 Simulation Environment

In order to simulate the performance of all these algorithms, we built a simple OFDM system, which has the same pilot structure as the WiMAX PUSC permutation scheme. The system parameters are shown in table 4.1 below: We simulate

Parameter	Value
FFT Size	1024
Bandwidth	10 MHz
Left Guard	92 Subcarriers
Right Guard	91 Subcarriers
Cyclic Prefix	25% of the Symbol
Modulation	16 QAM
Number of Clusters	60
Number of Pilot subcarriers	120
Number of Data subcarriers	720
Channel Coding	None

Table 4.1: System Parameters

the mentioned algorithms in section 4.5 as well as the proposed algorithm in section 4.6 using four models of wireless channels which cover all possible wireless channels with respect to delay spread "selectivity", and doppler spread "mobility". We have four channels, one with low selectivity and low doppler, another one with low selectivity and high doppler. The third one is for high selectivity and low doppler, and the forth is a high selective and high doppler channel.

The less selective channel is the suburban-alternative channel model found in [19], which has a delay spread of 7 usec and consists of 20 taps. The delay power profile of this channel is shown in table 4.2:

The other highly selective channel is the vehicular test environment with high

Path #	Delay [ns]	Amplitude [dB]
1	0	0
2	110	-1.8
3	290	-4.2
4	490	-4.6
5	700	-7.5
6	870	-7.6
7	1100	-8.4
8	1320	-10
9	1470	-8.6
10	1690	-8.3
11	1910	-6.4
12	2100	-11.8
13	2310	-12.8
14	2510	-12.1
15	2700	-16.3
16	2850	-15.3
17	3120	-18.4
18	3280	-19.6
19	4880	-20.2
20	7170	-21.6

Table 4.2: Low selective channel power delay profile

antenna channel of the ITU-R [8] , which has a delay spread of 20 usec and consists of 6 taps. The delay power profile of this channel is shown in table 4.3:

Path #	Delay [ns]	Amplitude [dB]
1	0	-2.5
2	300	0
3	8900	-12.8
4	12900	-10
5	17100	-25.2
6	20000	-16

Table 4.3: High selective channel power delay profile

Regarding the doppler frequency, two doppler frequencies are used, the first one is 40 Hz, which is equivalent to a velocity of 17Km/hr at RF frequency of 2.5 GHz. The other is 300 Hz which is equivalent to around 130Km/hr at the same RF fre-

quency. These are the low and high doppler in our convention.

We applied all the algorithms discussed earlier to this system, and in order to fairly compare the results, we fixed the number of multiplications per subcarrier in each system to 16 except for the one dimensional and cascaded interpolation. In the one dimensional frequency filtering method, the number of filter taps is 16. This means that in order to get the channel frequency response at each subcarrier, 16 pilots are needed with 16 coefficients.

For the two dimensional filtering, based on two successive symbols, 8 pilots are used from each symbol, which makes the total filter taps is also 16. For the two dimensional filtering based on three consecutive symbols, 6 pilots are needed from the current symbol, and 5 pilots are needed from each of the preceeding and the following symbols.

For the cascaded one dimensional filters, 6 pilots in time are needed for each time filtering operation, then 15 pilots "from the original and the added ones" are needed in the frequency filtering operation, knowing that the pilots to data ratio in our system is 1 to 6, the total number of multipliers needed in this case is 16 multiplications per data subcarrier. For the combined time interpolation, frequency filtering, the number of pilots we used for the frquency filtering operation is 15. We approximated the interpolation operation as a one multiplier operation, although it is much simpler, the total number of multipliers per subcarrier is slightly less than 16. Approximately, it has the same complexity.

4.8 Simulation Results

The resulting bit error rates of the low selectivity channels are shown in figure 4.9 while varying the signal to noise ratio of the transmitted signal from 0dB to 40dB. Figure 4.10 show the corresponding mean square error curves. We can see that in the low selectivity channel, interpolation methods have the best MSE in the low SNRs, but they floor very fast in the high SNRs, on the other hand, the filtering techniques don't have this floor. The robust filters can get better estimates in the

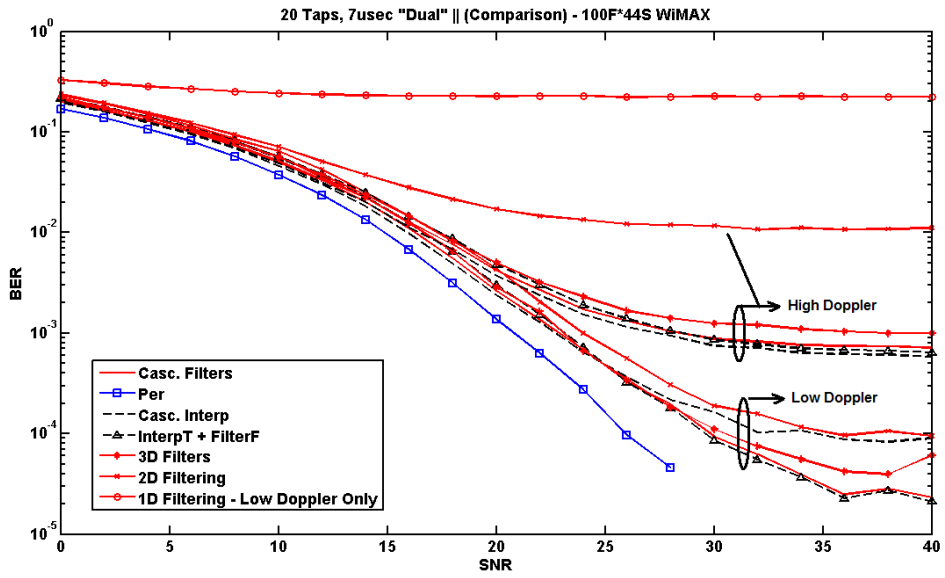

Figure 4.9: BER for Low Selectivity channel

higher SNRs, but their performance degrades in the low SNRs.

The cascaded filters, and the time interpolation/frequency filtering methods are the superior in the low selectivity channel, provided that the SNR is high. For low SNRs, the cascaded interpolation methods are the superior. The 1D methods either based on interpolations or filtering are the worst. We didn't simulate these methods in further more selective channels because of this. The 2D filtering based on two symbols, as well as the simple 2D interpolation methods give bad performance in all SNRs. This is because the correlation is not well-exploited in both methods. The 2D filtering based on 3 consecutive symbols "which we name 3D filtering in the graphs" is much better, because the correlations are best exploited. As the case in all filtering techniques, the 3D filtering technique is not that good in the low SNRs.

The differences in MSE in low SNRs is not clearly shown in the BER curve, because the hard decision device at the end of the receiver can tolerate some error, also, the BER is already bad in this area, so it is not making big differences.

If the doppler frequency becomes high, the floor of the MSE jumps from around -77dB in the best case, to -50dB. This is because the correlation now is very week

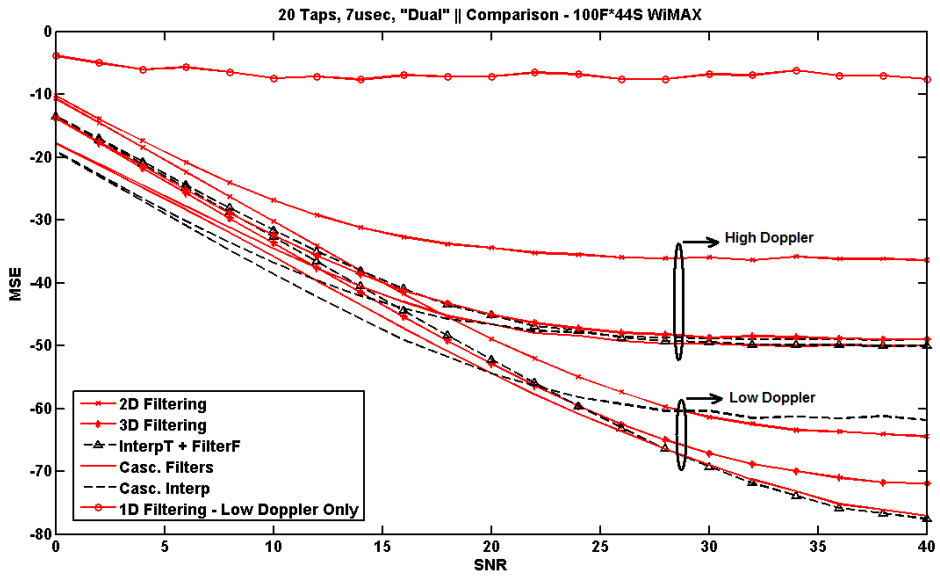

Figure 4.10: MSE for Low Selectivity channel

even within the same symbol. The rapid velocity makes the channel varies very fast within each symbol as well as between symbols. In this case, the results are the same as the low doppler case, except that they have higher error floor.

In the case of high selective channel, the resulting bit error rates and mean square errors are shown in figures 4.11, 4.12. The filtering techniques are superior in all cases because the frequency correlation is now very small, so the interpolation makes large performance degradation. The cascaded filtering technique is the best of all, then the time interpolation/frequency filtering technique. The 2D filtering is worse than the 3D filtering as before.

If the doppler frequency becomes higher, the MSE floor jumps to -50dB as in the low selectivity case. the 2D MSE floors at around -35dB which is far from the 3D and cascaded filters by 15dB. This is due to the low correlation and high varying channel in both time and frequency directions.

The filtering equation (4.6) requires the SNR in order to calculate the coefficients. This means that the coefficients vary with the SNR. One solution is to estimate the SNR, and store sets of coefficients for each SNR portion. The SNR

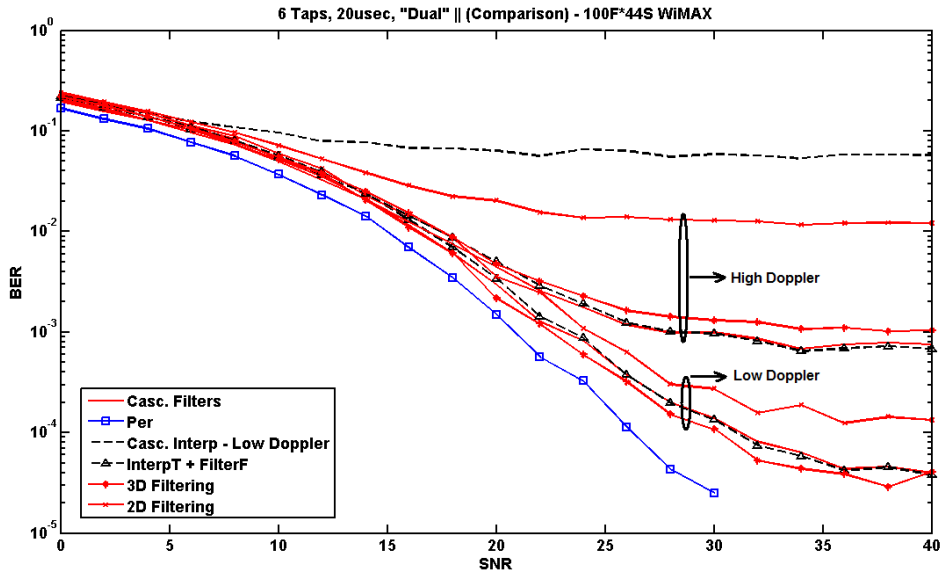

Figure 4.11: BER for High Selectivity

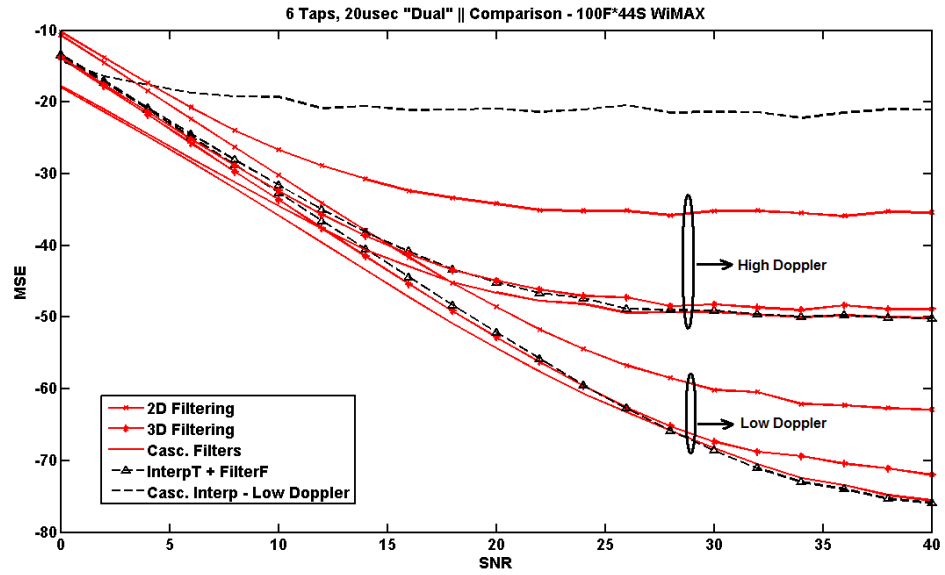

Figure 4.12: MSE for High Selectivity

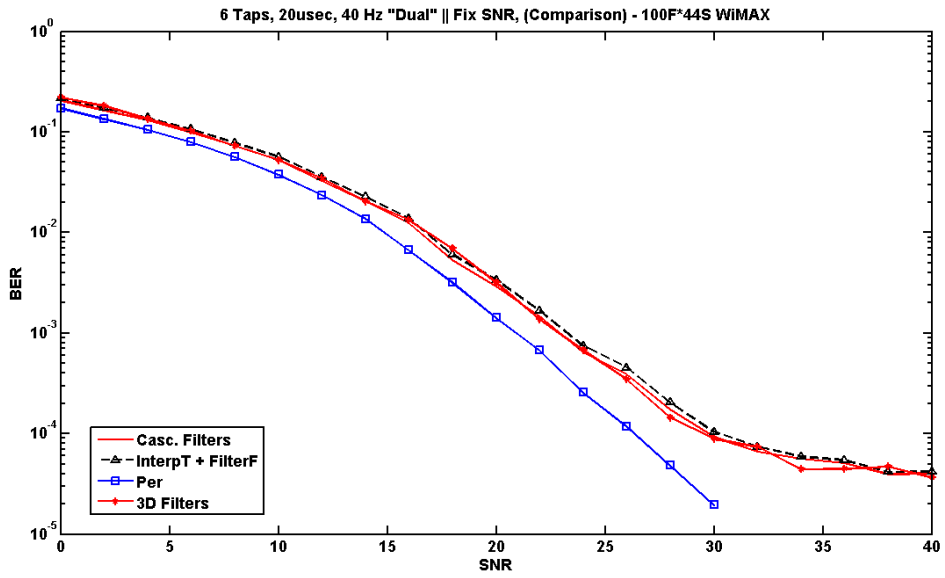

Figure 4.13: BER for High Selectivity, Low doppler channel, Fixed SNR Coefficients

estimator chooses the appropriate set of coefficients from all the sets. This was the case in the previous simulations.

Another solution to this problem is to use a nominal SNR value to calculate the coefficients, and fixate the coefficients. Only one set of coefficients will be needed in this case and no SNR estimator is needed at least for the channel estimator block. This is the solution we chose. We fixed the SNR at 20dB, and calculated the coefficients for this SNR value. We simulated the system for only the high selective low doppler case only. The results are shown in figures 4.13, 4.14. The performance didn't change with this new modification compared to figures 4.11 and 4.12.

In all filtering methods, there is a floor in the MSE as well as the BER at high SNRs. This floor is mainly due to the edge effects. Most of the errors found in high SNRs are found in the edges of the symbol. At the edges, the filters utilize the next N_{tap} pilots in only one direction. These pilots become far away from the required data location at the edge and become less correlated.

As we discussed before in section 4.6, we solve this problem by proposing a minor modification to the cluster pilot structure, in which a single pilot is added

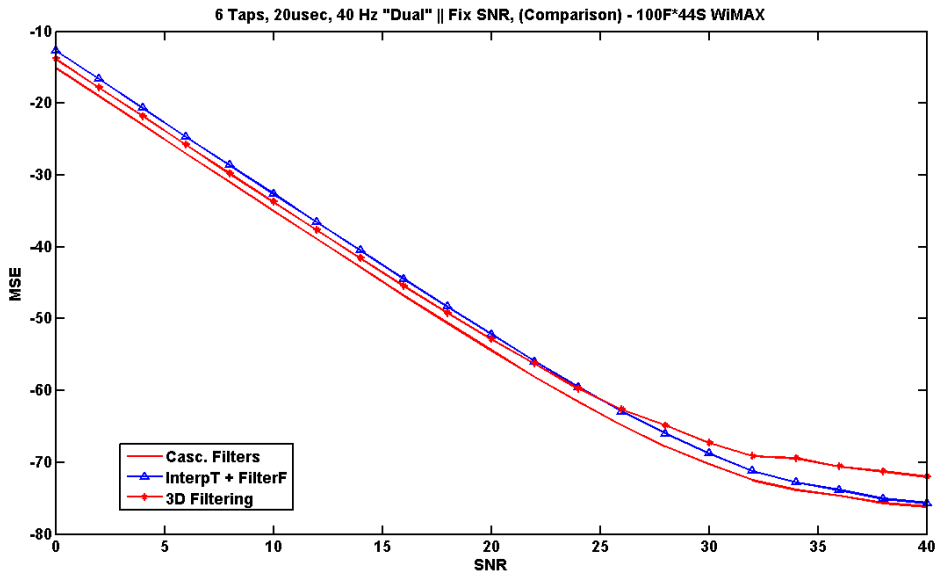

Figure 4.14: MSE for High Selectivity, Low doppler channel, Fixed SNR Coefficients

at the edge of each symbol as in figures 4.7 and 4.8. Simulations were carried out for this modified pilot structure, and the resulting bit error rates and mean square errors are shown in figures 4.15, 4.16.

We can see an improvement of more than 10dB in the BER floor at the high SNRs due to this modification. The MSE of the cascaded filters as well as the time interpolation frequency filtering method improved by about 5 dBs.

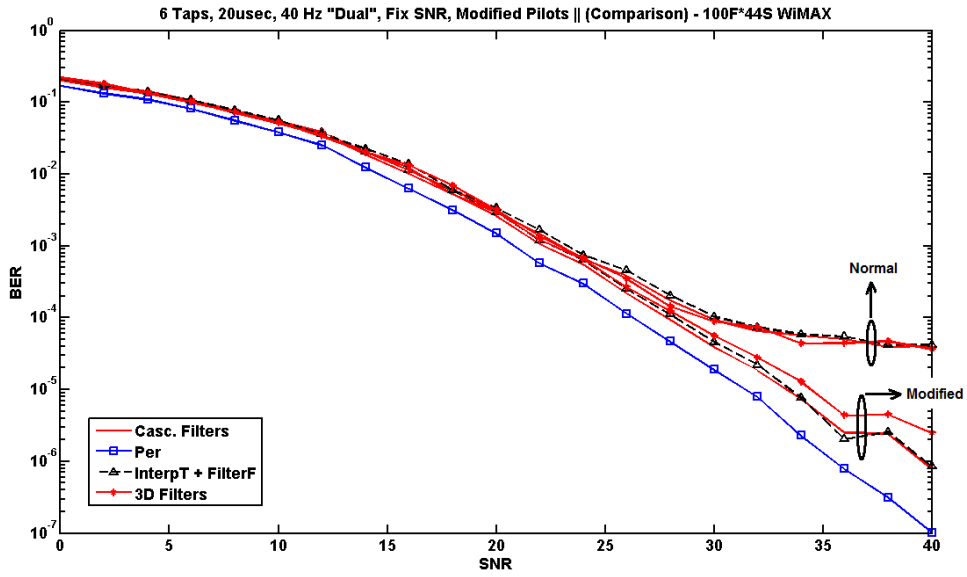

Figure 4.15: BER for High Selectivity, Low doppler channel, Fixed SNR Coefficients, Modified Pilots

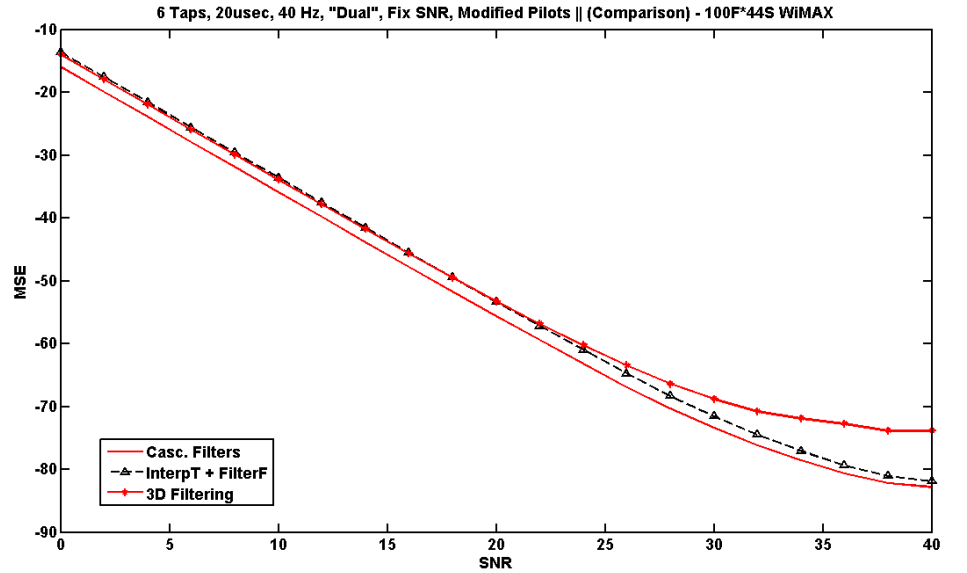

Figure 4.16: MSE for High Selectivity, Low doppler channel, Fixed SNR Coefficients, Modified Pilots

Chapter 5

Hardware Implementation

In this chapter, the hardware implementation of the proposed channel estimation technique is presented. The cascaded interpolation and filtering algorithm is used as it is simple and gives acceptable performance compared to other algorithms as described in chapter 4 before. The target environment for this hardware implementation is the STRATIX II FPGA. The design is using the WiMAX environment and timing constraints as the prototype that will be implemented. This doesn't mean that the hardware architecture is only designed for WiMAX. The hardware architecture -as we will demonstrate- is capable of dealing with different OFDM systems by only minor modifications to some registers and memory contents.

This chapter is organized as follows, first, we describe the implementation environment, which is the STRATIX II FPGA. Then, we proceed to a general block diagram of the system. After that, we describe in details the timing restrictions for the design in order to cope with the WiMAX environment, and deduce the throughput requirements of each block in our system, and its required minimum clock frequency. Then, detailed discussion of each block is presented including both the data path and the control path of the block. Finally, the resources used in the overall design are described as well as the timing achievements.

5.1 Introduction to STRATIX III FPGA

The desired platform for our design is the STRATIX III EP3SL150F1152 from Altera. This FPGA is chosen mainly because it has a development kit associated with it, that makes the testing and prototyping much easier. Also, it has many features and built in IP's that makes the design easier and more flexible to implement. This includes -but is not limited to- the following [20]:

1. Total of 5,630,976 bits of memory, divided between 9K and 144K blocks.

2. 384 DSP blocks, each includes four 18× 18 bits multipliers, and adders.

3. Total of 8 PLLs built on chip.

4. Maximum clock frequency of 400 MHz for the RAM blocks on chip.

5. Maximum clock frequency of 400 MHz for the multipliers and DSP blocks on chip.

The previous features and IPs makes this family very suitable to our design.

5.2 Implemented Channel Estimation Algorithm

The implemented channel estimation algorithm is the cascaded time interpolation / frequency filtering method described in chapter 4. In this section, we review briefly this algorithm to make it easy for the reader to understand the associated hardware architecture.

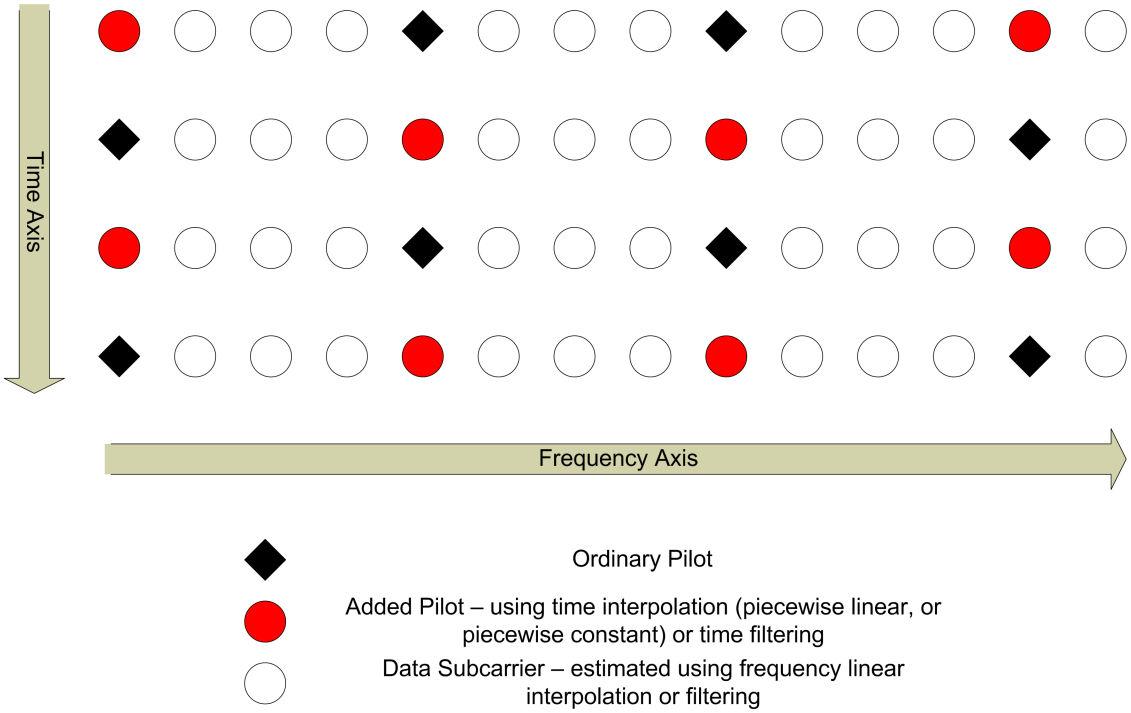

Figure 5.1: Cascaded Interpolation/Filtering Method Illustration

Figure 5.1 shows the pilot structure for each cluster in WiMAX. Each row represents a portion of the OFDM symbol called a cluster, and this unit is repeated along the whole OFDM symbol.

The estimation method is basically done in two steps. First of all, the pilots are interpolated in time to obtain accurate estimations at the in-between subcarriers. In our case, the interpolation is simply an averaging operation that is very simple. Extrapolation is done on the edges to get the estimations at the first and the last OFDM symbols of the frame.

These accurate estimations are used as extra pilots added to the original pilots set available in each symbol. All these pilots and added pilots are available at the input to an FIR filter that estimates the remaining channels in between along each symbol.

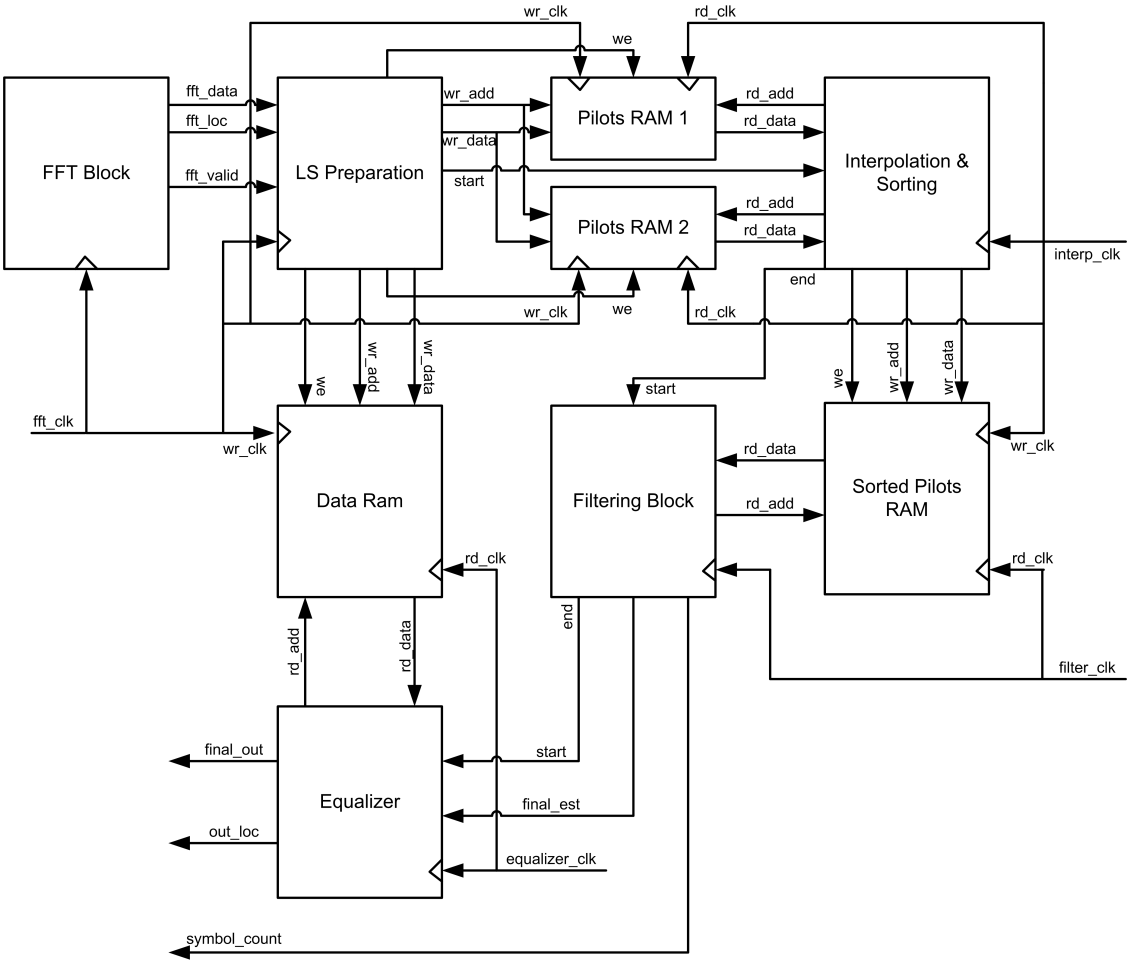

Figure 5.2: Complete System Block Diagram

This method is described in detail in chapter 4, and detailed simulation results were obtained for different channel conditions. The overall performance is very good, taking into consideration the simplicity of the algorithm compared to other algorithms described in chapter 4. This is why we chose this method to implement.

5.3 System Block Diagram

Figure 5.2 shows a general block diagram of the whole system. The system should interface with the FFT block which is available in the receiver, extracts the pilots and separates them from the data,

The data path is as follows. The FFT engine starts producing its output, and its valid signal is raised high. This is the starting ignition of our blocks. The LS Preparation block is responsible for separating the pilots and the data subcarriers into separate memories. The pilots are stored in two memories successively, i.e. the pilots of the first two symbols are stored in order in the first memory, then the pilots in the third and fourth symbols are stored in the second memory and so on. The data subcarriers are stored in order in a different memory. Here, we are assuming that the FFT output is produced in natural order. This assumption is justified since many commercially available IPs have this ability [21] [22] [23] [24].

Before the LS preparation block stores the pilots, it normalizes them by multiplying them by their well-known magnitude, in order that the subsequent filtering operation is correctly performed. This is what we call 'Least Squares' or 'LS'. The data subcarriers are stored as they are without any normalization.

The interpolation and sorting block is responsible for interpolating the subcarriers in time to obtain the new subcarriers in between which will be used as extra pilots in the filtering operation. Interpolation is a simple averaging operation. At the symbol edges, interpolation is replaced by extrapolation. The interpolation is made simply by accessing the pilots RAM1 and RAM2 with the same address, and interpolating the output data from the two memories together. This is clarified later in the detailed architecture of the block as well as in the timeline in the following section.

In addition to interpolation, the Interpolation and Sorting block is responsible for sorting the pilots all together in a single RAM, so that the filtering block can perform its operation easily. The original pilots are copied from RAM1 and RAM2 and are stored in their correct location in a new RAM. The interpolated data is also stored in the appropriate location. This sorting operation is clarified later.

The filtering block is responsible for filtering the sorted pilots to obtain the final channel estimator. It is a complex multiply accumulate operation that is performed many times depending on the number of taps for the filter. In our case, we chose 15 taps as its performance is good, as was discussed in chapter 4. The coefficients are stored in a RAM inside the block, and are loaded when needed.

Finally, the equalizer gets the filtering output as well as the original received data from the FFT which was stored before by the LS preparation block. The main function of this block is to divide these two complex quantities by each other. This is to eliminate the channel effect from the received data, so subsequent demodulation, decoding, and detection can work properly.

As shown in the block diagram, the main inputs to our system are the FFT output signals, and 4 different clock signals, one for each block. The main outputs are the final equalized data, its location, and its symbol number. In the following section, we calculate the minimum frequencies for each clock signal such that our timing constraints are met and not violated.

5.4 Timing Constraints

In this section, we will describe the hardware time budget in detail, and deduce the required minimum frequency for each block to operate correctly.

Figure 5.3 shows the timeline for our architecture. The basics in our system are the time for a single OFDM symbol which is divided into the complete FFT size time plus the cyclic prefix time. The FFT time is 91.4 μ sec and the cyclic prefix is 22.9 μ sec. This is corresponding to the system model described in chapter 4 for

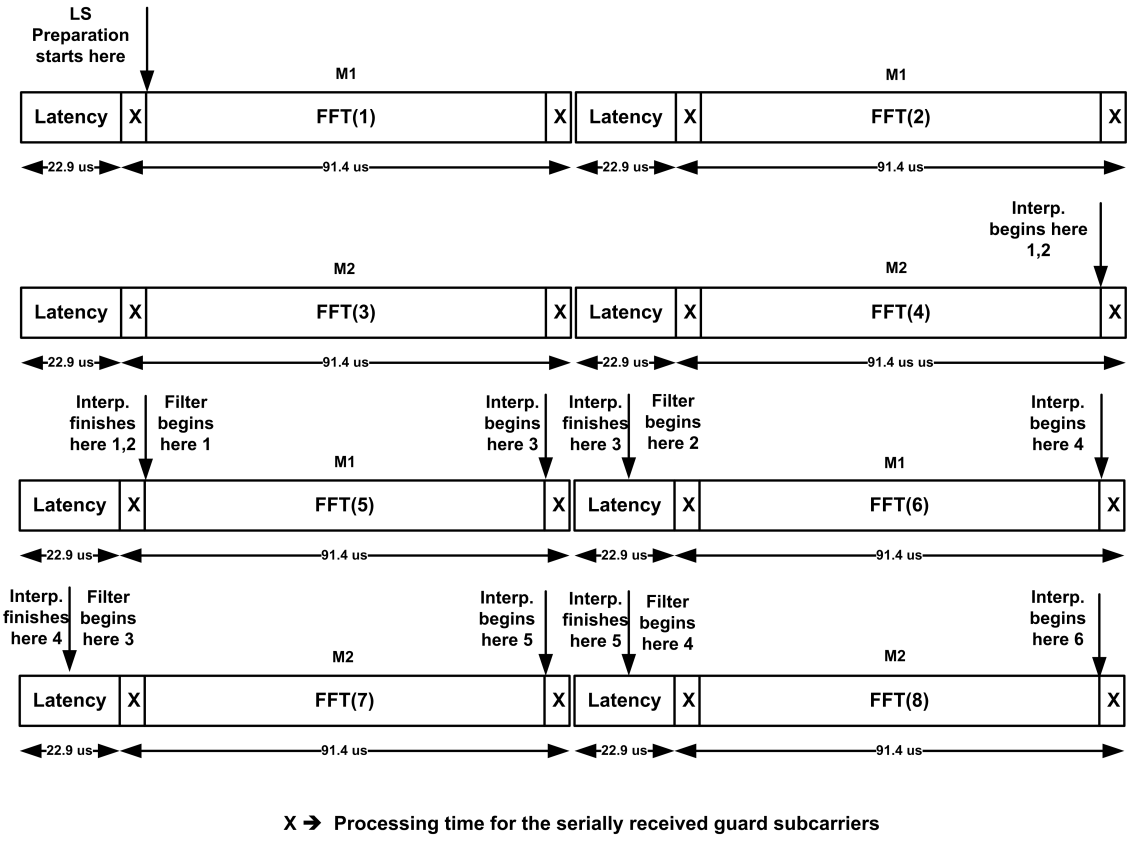

Figure 5.3: Hardware Timeline

the PUSC WiMAX - 1024 FFT size. Next, we describe the requirements for each block separately.

5.4.1 LS Preparation Time Budget

The LS preparation block should start whenever the FFT starts to send valid data. The FFT valid signal is raised when the FFT block starts producing its output, but the first 92 subcarriers are guard subcarriers and are not important. When the guard subcarriers are finished, the block begins to operate and separates the data and pilots. It is assumed here in this work that the FFT block has a latency which is equal to the cyclic prefix between successive OFDM symbols.

By the start of the next frame, the LS preparation should finish four complete

OFDM symbols in order that the subsequent interpolation begins. To make it easy for the interpolation block, the LS preparation fills two memories in sequence such that every two successive symbols are buffered to one RAM.

It is clear that the LS preparation block has no option except that keeping in pace with the FFT block. This means that the clock frequency used in this block should be synchronized with the FFT block clock frequency. For the 1024 FFT size, this clock frequency is simply calculated as follows:

$$F_{fft} = \frac{1024}{91.4\ \mu sec} = 11.2\ MHz \tag{5.1}$$

$$T_{fft} = \frac{1}{11.2\ MHz} = 89.3\ nsec \tag{5.2}$$

5.4.2 Interpolation Time Budget

The interpolation block should start its work after the first four symbols were prepared. At the beginning of the guard interval for the fourth symbol, the interpolation starts. The main job for it is to interpolate and copy the different prepared pilots and store them in order as they should appear to the filtering block. The pilots should be in the same order as in figure 5.1.

To operate correctly, the interpolation is made between the pilot in RAM1, and the corresponding pilot at the same address in RAM2. Both RAM1 and RAM2 should have the correct data for the interpolation to operate correctly. The problem is that as the fifth symbol is received, RAM1 will be refilled again and its old content will be overwritten. The interpolation should have finished its work before RAM1 is being filled again by the LS preparation block.

As an example to clarify things, by looking at figure 5.1, for the first symbol,

the interpolation should first extrapolate from symbol 2 and symbol 4 to get the first added pilot. This will be done through accessing the second half of RAM1 and RAM2 with the same address. After that, two pilots will be copied from RAM1, then the extrapolation is repeated again and so on.

From the previous discussion, it is clear that the maximum time duration allowed to the interpolation, is that between symbol 4 and symbol 5, it should have interpolated and sorted pilots in symbols one and two. This is the duration of two guard intervals plus one cyclic prefix interval. We can calculate the maximum time interval for this block as follows:

$$T_{interp} = 92T_{fft} + 22.9\ \mu sec + 91T_{fft} = 39.2\ \mu sec \tag{5.3}$$

To get the minimum clock frequency, we should know how many cycles will be spent during this time. In this time duration, 120 pilots should be interpolated as well as other 120 pilots should be copied. This means we have 4*120 cycles 'approximately', then the clock frequency is:

$$F_{interp} = \frac{120 * 4}{39.2\ \mu sec} = 12.2\ MHz \tag{5.4}$$

5.4.3 Filtering Time Budget

The filtering operation is the most time consuming operation in our work. It starts after the interpolation blocks finishes interpolation and sorting for each symbol. As we discussed in the previous section, the interpolation for symbols 1 and 2 finishes before the beginning of preparation of the fifth FFT symbol. At this time instance, the filter has enough data to begin its work. The worst case that it takes as much time as the complete FFT symbol. In this case, before the next symbol being

prepared, the filtering should have finished. In our case, the filtering of symbol number one should have been finished before the starting of preparation of symbol number six. In this case the time duration allowed for filtering operation is the complete symbol time which is:

$$T_{filter} = 22.9\ \mu sec + 91.4\ \mu sec = 114.3\ \mu sec \tag{5.5}$$

In this case, we will notice an overlap between the filtering operation and the interpolation operation. If we assumed that the ordered pilots memory is enough to buffer two symbols only, then there will be a problem. While the filtering operation is performed on symbol 1, which is stored in the upper half of this memory, this upper half is being updated by the interpolation of symbol 3, so the ordered pilots memory should be wide enough to buffer 3 complete symbols. We can relax the memory requirements by tightening the timing requirements. If we can guarantee that there is no overlap between interpolation and filtering block, then we can use a memory that will buffer only two symbols. In this case, the time duration for the filter to complete its work is:

$$T_{filter} = 91.4 - 92 * T_{fft} - 91 * T_{fft} = 75.1\ \mu sec \tag{5.6}$$

Another issue, if we assumed the FFT data memory that the LS preparation fills is enough for only four OFDM symbols, then we will have a problem. In this case, at the beginning of the fifth OFDM symbol, LS preparation will fill the first half of the FFT data memory, but equalizer still needs this data because filtering is still performed on symbol 1. This is why the FFT data RAM should be wide enough to buffer 5 OFDM symbols not only 4.

In order to calculate the minimum clock frequency for this block, we should

estimate how many cycles we need. Approximately, we will need 15*600 multiplier cycles + 120 copy cycles for the interpolated pilots + 720*2 cycles to clear the accumulator between each subcarrier. This adds to 10560 cycles. In this case the minimum frequency should be:

$$F_{filter} = \frac{10560}{114.3\ \mu sec} = 92.4\ MHz \tag{5.7}$$

$$F_{filter-tight} = \frac{10560}{75.1\ \mu sec} = 140.6\ MHz \tag{5.8}$$

5.4.4 Equalizer Time Budget

The equalizer should finish its work in pace with the filtering block. Once the filtering block finishes its work, the divider should take the estimated channel, and divide the data from the FFT memory by it.

During the working period of the filter, the divider will perform 720 divisions. Assuming a pipelined divider, then the minimum frequency will be:

$$F_{filter} = \frac{720}{114.3\ \mu sec} = 6.3\ MHz \tag{5.9}$$

$$F_{filter-tight} = \frac{720}{75.1\ \mu sec} = 9.6\ MHz \tag{5.10}$$

The problem here is that some subcarriers are output from the filtering block faster than others. For example, at the LS pseudo pilots locations, no filtering is needed, and the output will be ready after a single clock cycle. This means that we are forced to use the same clock frequency of the filtering block in the equalizer block, or we should buffer the output of the filtering block to a memory, then the divider can work with the above calculated frequency. We choose the first solution and the equalizer is made to work with the same clock frequency of the filter.

5.5 Detailed Hardware Architecture

In this section, we describe each main block of the channel estimator blocks in figure 5.2. The architecture is illustrated and described in detail. The communication between each block and the other blocks is discussed. Finally, the synthesis result is summarized.

5.5.1 LS Preparation

The inner structure of the LS preparation block is shown in figure 5.4. As we mentioned is section 5.3, this block is responsible for separating the data subcarriers from the pilot subcarriers. The pilot subcarriers are multiplied by their well-known amplitude value then they are stores in two memories, each one has two consecutive symbols. The data subcarriers are buffered directly to the FFT RAM.

The main challenge in this block is how it will differentiate between pilot locations and data locations. Pilot locations are well-know for WiMAX, but we need to make the system more generic such that it can support other wireless systems other than the WiMAX system. The key to solve this problem is that it is logical that the pilots are usually periodic. The pilot locations are usually fixed during a portion of the symbol then they are repeated again. Taking this into consideration, we put a shift register which can be pre-programmed by the user by entering the systems pilot locations as a series of zero's and one's. A 'one' represents a pilot, and a 'zero' represents the data. By circularly shifting this register, we produce a serial output which can either enable the pilots memory or the data memory.

To choose whether pilots will be stored in RAM1 or RAM2, a symbol counter is available in the LS preparation controller. Its second least bit will indicate whether

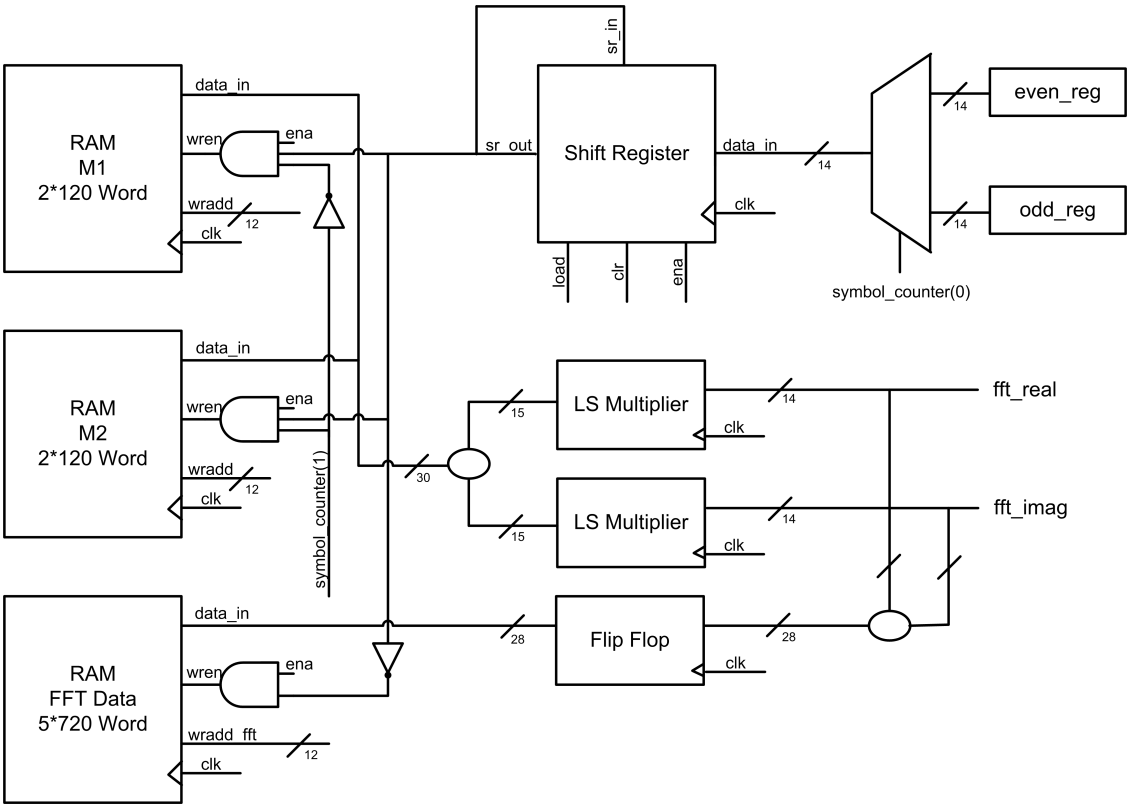

Figure 5.4: LS Preparation Block Data Path

the RAM1 or RAM2 will be used.

In WiMAX case, the pilot locations are varying between odd and even symbols. This is solved by using a multiplexer at the input of the shift register that feeds it with either the even pilot locations or odd pilot locations. Two multipliers are used to normalize the output of the FFT at the pilot locations in order that the subsequent filtering component works properly.

In order to have equal path delays at the inputs of all the memories used, a flip flop is built at the input of the FFT RAM, which makes the data available to the RAM with a one cycle delay to have the same delay that the LS multiplier produces.

The component controller should deliver the correct control signals to each block in the correct time. It is mainly composed of a finite state machine that waits the valid signal from the FFT block, then waits until the guard subcarriers are finished, then loads the shift register and start shifting. At this time, the memories enable signals are raised high, and the system works. At the DC subcarrier location, the system should suspend itself for one clock cycle, because the DC subcarrier -which lies in the middle of the symbol- is neither a pilot nor a data. After the used portion of the symbol is finished, and the right guard subcarriers start getting out from the FFT, the controller should stop writing into memories and return to its idle state waiting for another FFT valid. In this case, a finish signal is raised.

5.5.2 Interpolation

The architecture of the interpolation and sorting block is shown in figure 5.5. This block is responsible for producing a sorted RAM including the normal pilots as well as the pseudo pilots resulting from the interpolation. They should be in their

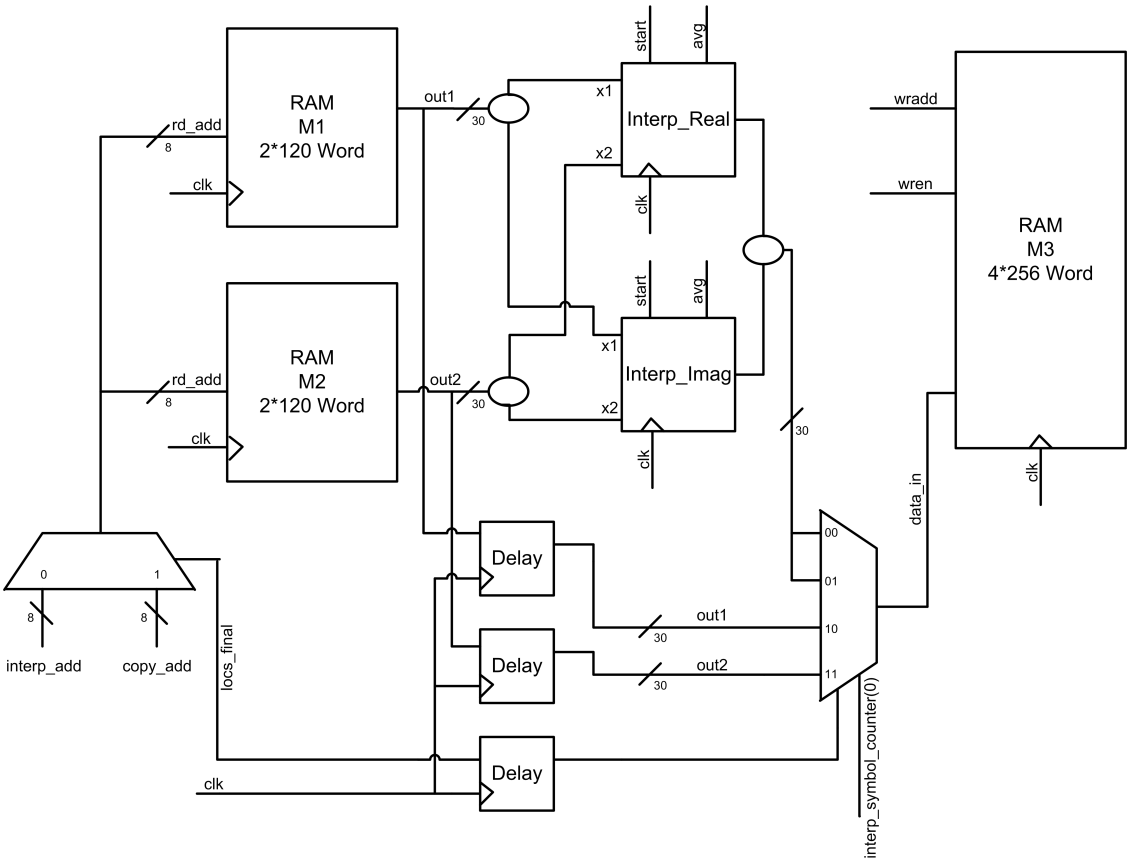

Figure 5.5: Interpolation Block

correct order.

This block simply reads from the RAM1 and RAM2 previously prepared by the LS preparation block. The address supplied to each RAM is the same, which is resulting from a multiplexer that chooses from a copy address, or an interpolation address, depending on whether the current operation is an interpolation or copying operation.

The interpolation is done through the block in figure 5.6. The interpolation is either one of three simple linear operations. These operations are:

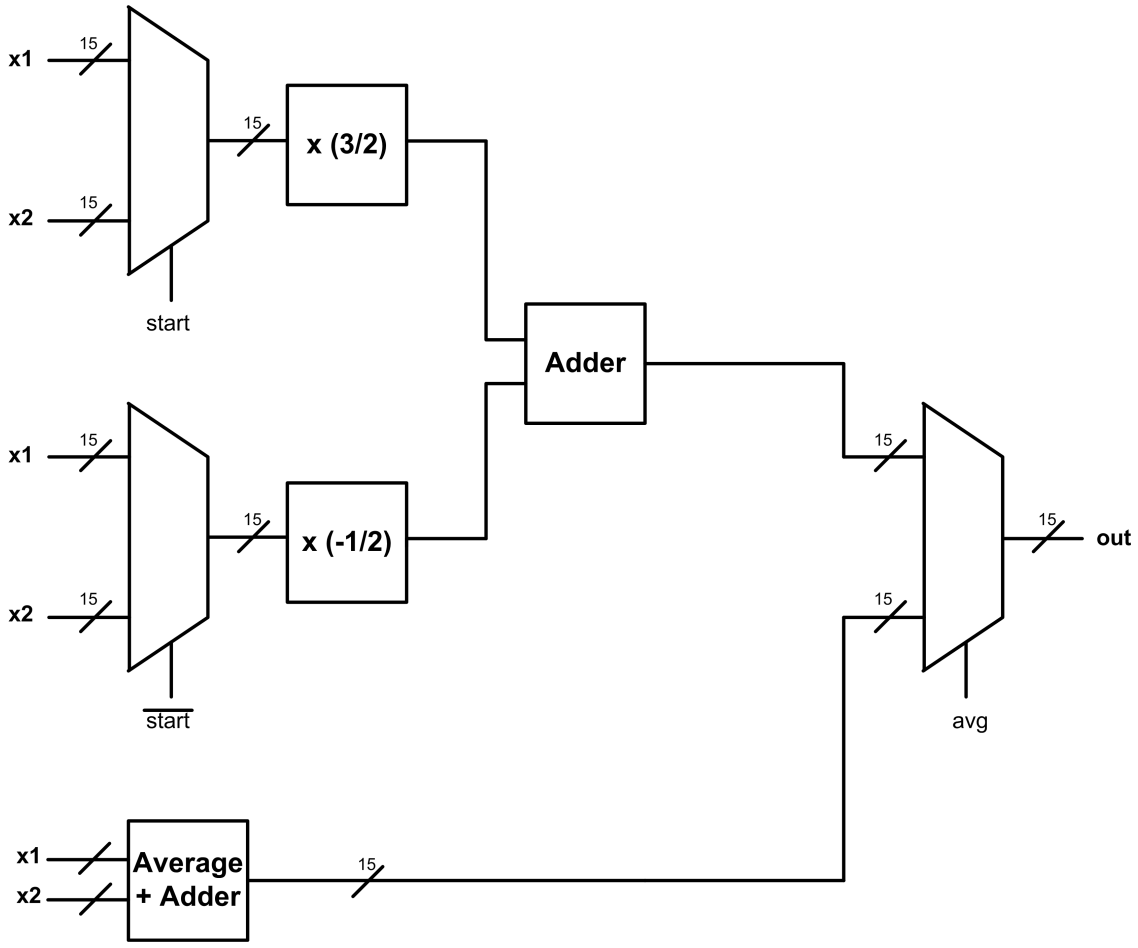

Figure 5.6: Real Interpolation Component

In the first OFDM symbol, extrapolation equation should apply which is:

$$\widetilde{H}[1,k] = \frac{3}{2}\widetilde{H}[2,k] - \frac{1}{2}\widetilde{H}[4,k] \tag{5.11}$$

In the middle symbols, simple averaging equation is applied which is:

$$\widetilde{H}[n,k] = \frac{1}{2}\widetilde{H}[n-1,k] + \frac{1}{2}\widetilde{H}[n+1,k] \tag{5.12}$$

In the last OFDM symbol, extrapolation equation should apply which is:

$$\widetilde{H}[N,k] = \frac{-1}{2}\widetilde{H}[N-3,k] + \frac{3}{2}\widetilde{H}[N-1,k] \tag{5.13}$$

It is clear that we have only three multiplication operations, one is by half, the other is by (-1/2) and the last is a multiplication by (3/2). That is what we are doing in the interpolation engine in figure 5.6. The start and avg signals are output from the controller. The set of multiplexers verify that the correct operation will be made.

The controller of this block should output the control signals in the correct time. It should first wait until it reads 4 successive finish signals from the LS preparation block before beginning. A shift register which is loaded by the locations of the copying and/or interpolation is available inside the controller. It stores one if the current pilot should be interpolated, and zero if the current pilot should be copied. The controller is working on RAM1 and RAM2 successively. For example, if we looked at the pilot structure in figure 5.1, we see that the shift register contents are 1001. The first subcarrier should be interpolated, the second and third should be copied directly, and the fourth should be interpolated. The controller should correctly chooses whether it copies from RAM1 or RAM2. This is controlled by a counter inside the controller, where the least significant bit indicates which memory to copy from. The controller have three stages, one for the first symbol, one for the

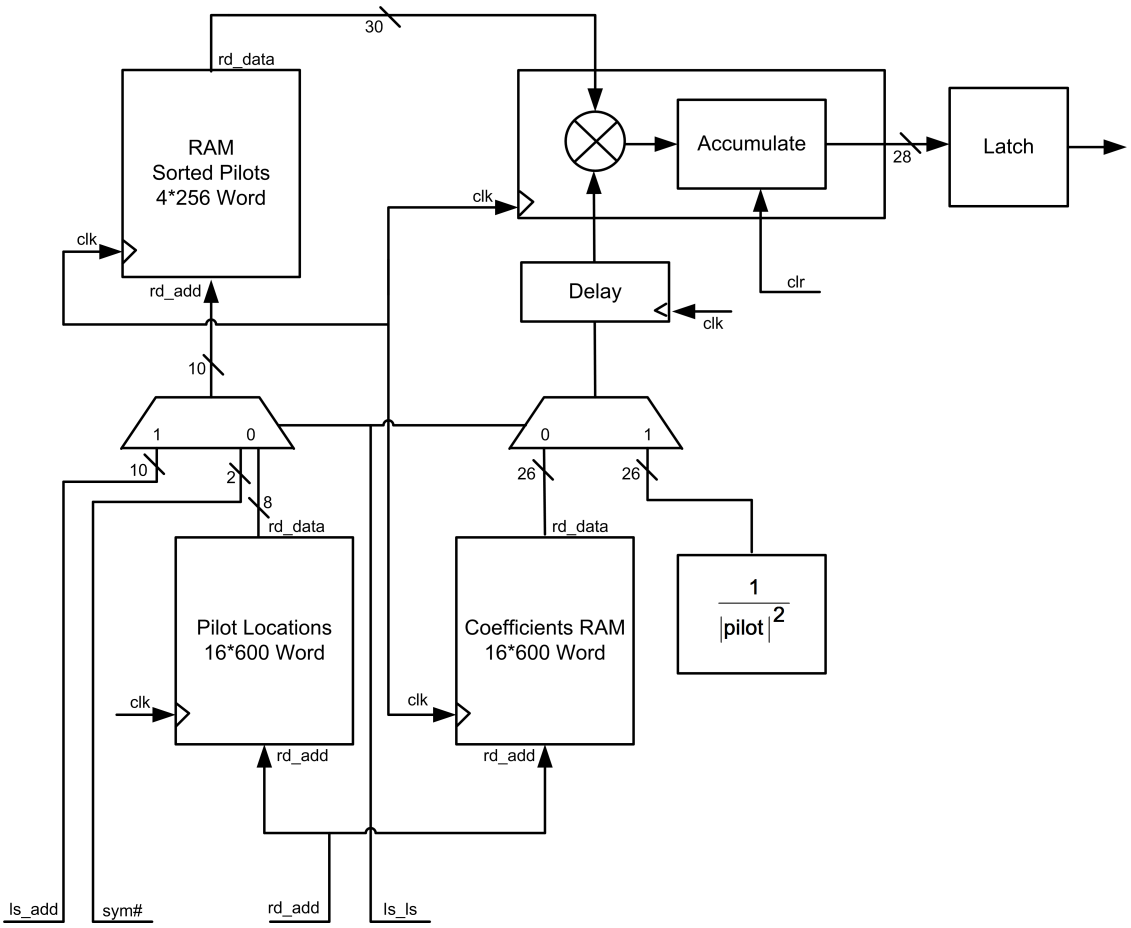

Figure 5.7: Filtering Block Data Path

mid-symbols, and one for the last symbol. During each stage, the correct start/avg signal is raised, in order that the inner interpolation engine works correctly.

5.5.3 Filtering

The filtering block is the bottleneck of our design. It is shown in figure 5.7. As shown, it is mainly composed of a complex multiplier and an accumulator. The complex multiplier is composed of three integer multipliers that are directly used from the pre-built DSP blocks on chip.

The filtering operation is done by first accessing the correct coefficient and the corresponding pilot location from two pre-programmed memories. the pilot loca-

tion is introduced as the address to the RAM containing the sorted pilots from the previous interpolation and sorting block. At the same time, the complex multiplier gets the coefficient and the pilot, it multiplies them and adds the result to the accumulator. During multiplication, an additional pilot with the corresponding coefficient is available to the multiplier. The filtering block continues working in a pipelined fashion until the 15 multiplications are finished, and the final output is available. The accumulator is cleared after single cycle to continue the next filtering operation and so on.

One should take care that some estimated data locations don't require all these number of multiplications. These are the locations that were interpolated successfully. All what should be made to them is to multiply them by the inverse of the pilot amplitude. This is done by the controller. The controller should be able to discriminate these locations by using a built in shift register similar to that found in the LS preparation block before. This shift register contains the pilot patterns in the symbol, so that the controller can identify the locations that doesn't need the 15 multiplications operation. A finite state machines in the controller controls the timing of loading the shift register, enabling it, making either 15 multiplications or a simple one multiplication, and finally clearing the accumulator and producing the handshaking signal to the subsequent divider.

5.5.4 Equalization

The equalizer is the last block in our design. It is shown in figure 5.8. It is mainly composed of two complex multipliers, to produce the pilot multiplied by the channel's complex conjugate, and the other produces the magnitude square of the complex channel output from the filter.

Two real dividers are available to divide the received data multiplied by the channel's conjugate, by the magnitude square of the complex channel. This totally removes the channel effect. This is illustrated in the following equation:

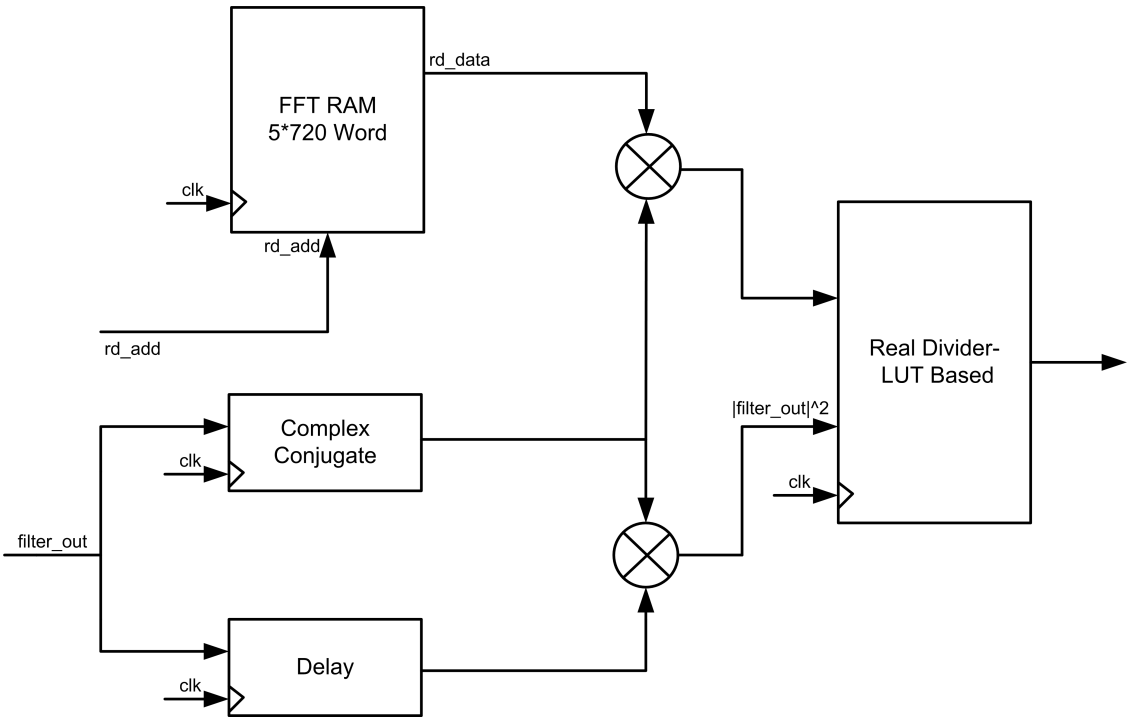

Figure 5.8: Equalizer Block Data Path

$$\widehat{x}[n,k] = \frac{y[n,k]\widehat{H}^*[n,k]}{|\widehat{H}[n,k]|^2} \tag{5.14}$$

The dividers used in the design are built simply by look-up tables. The inverse squared of the absolute value of the channel response is pre-loaded in a ROM, that the divider is able to access, and a simple on-chip real multiplier is used instead of the driver operation. This gives the divider more speed at the cost of increased area.

5.6 Fixed Point Analysis

To correctly build the above architecture, fixed point simulations were carried out to correctly choose the minimum number of bits for each block in order to minimize the quantization errors and achieve a near optimal system.

Here, we concentrate on the system without any diversity gains, and our analysis requires that the bit error rate curve of the floating point system be the nearest to

the corresponding curve for the fixed point case.

After many trials, we found out that in order to achieve nearly the same performance as the floating point, we will need 11 bits to store our fraction, and variable bit widths for the integer portion according to the location of the block within the system. All our blocks utilize the two's complement form to represent negative numbers. Table 5.1 summarizes the number of bits for each important signal in our design.

Signal Name	Total Bits	Integer Bits
FFT Inputs	14	3
M1&M2 Contents	15	4
M3 Contents	15	4
Filter Coefficients	13	2
Filter Outputs	14	3
Divider LUT	23	12
Divider Output	15	4

Table 5.1: Fixed Point Representation

The LUT ROM is the most wide block in out system. This is normal since the inverse squared of the channel frequency response can change from zero to 2048 which is the inverse of the least word in the design -which has the least significant bit = 1-. The smallest word in our design is $2^{-11} = 0.00048828$ which has an inverse of 2048.

In our simulations on Matlab, we simulated a rounding method to deal with any overflows or errors in quantization. The quantized floating point value is approximated to the nearest point in the fixed point representation. This shows exactly the same performance as floating point case. This will be shown in the next section.

5.7 Design Verification & Simulations

In order to verify our hardware, a complete test bench is implemented to compare the results of the hardware architecture with those of Matlab. In order to accurately simulate the hardware performance, another version of the Matlab simulations was built using only truncation and no rounding is used. This means that any multiplication or addition in the simulated program has a truncated result to cope with the fixed point representation specified in table 5.1. The Matlab simulation was done and a test bench verified its exact equivalence to Matlab results.

A small hardware truncator is built to deal with overflows, such that the result is saturated when it exceeds the allowed range specified by the number of bits. By this way, truncation can't cause errors in the sign of the truncated value as well as its approximate value.

The resulting bit error rates of the low selectivity channels are shown in figure 5.9 while varying the signal to noise ratio of the transmitted signal from 0dB to 20dB. Figure 5.10 show the corresponding mean square error curves.

As shown in the figures, the fixed point rounding and the floating point analysis are identical. The hardware truncation introduces a floor in the mean square error results, such that it is far from the floating point results by around 2dBs. This is not affecting the BER curves because the demodulator can tolerate some error.

5.8 FPGA Resources and Timing Achievements

After compiling each block on the STRATIX III EP3SL150F1152 FPGA using Altera Quartus II software [25], synthesizing, placing and routing, the synthesis report showed the results in table 5.2 for the area utilization and speed.

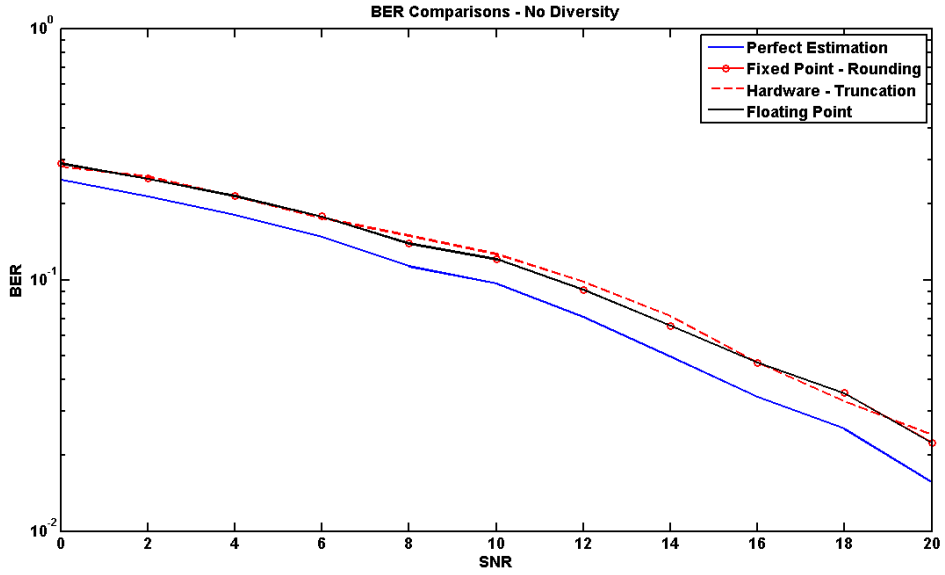

Figure 5.9: BER Comparisons for Hardware Verification

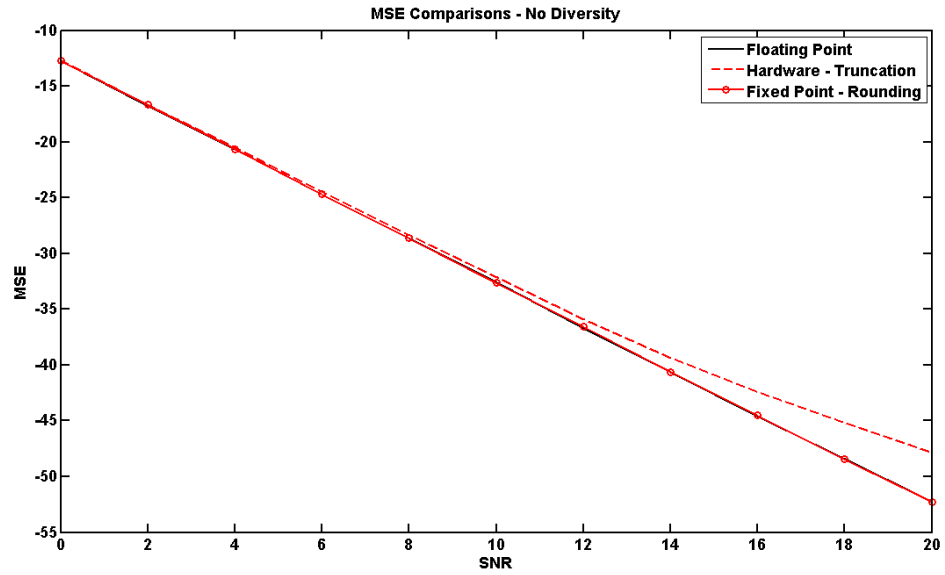

Figure 5.10: MSE Comparisons for Hardware Verification

Block Name	Required Frequency	Obtained Max. Freq.	LUT - Registers	DSP Blocks
LS Preparation	11.2 MHz	374.4 MHz	212 - 138	0
Interpolation & Sorting	12.2 MHz	377 MHz	311 - 217	0
Filtering	140.6 MHz	374.4 MHz	247 - 439	4
Equalizer	140.6 MHz	325.8 MHz	43 - 66	14
Total System			798 (1%) 1407 (1%) RAM Bits: 1,225,984 (22%)	18 (5%)

Table 5.2: Synthesis Report

As shown in the synthesis report, all the blocks are satisfying the timing requirements, and the area utilization is less than 2% of the whole FPGA area. The memory utilization is: 1,225,984 bits -around 22% of the total memory block in the design- The logic utilization is very low compared to memory utilization because the system is made to support multiple standards. This slightly large amount of memory is required to mantain the reconfigurability. It can be relaxed if the system is only supporting WiMAX or other standards.

Chapter 6

Conclusions and Future Work

In this thesis, we present a simulation model as well as a hardware architecture of a channel estimation and equalization technique for WiMAX as a special case to the general OFDM systems case.

In our work, we investigate different channel estimation techniques currently used, and propose a new estimation technique based on Wiener Filtering and simple interpolations. The new technique proved to be an efficient technique and provides good performance. The performance is much better than the currently used interpolation techniques which fail in the highly selective channels.

We reached a conclusion that the well-known decision directed channel estimation is not a good choice in rapidly changing wireless channels, but it is very efficient in wireline channels where approximately no pilot subcarriers are used. We introduce a new method to improve the SNR thresholds problem in the decision directed estimation which is simply adding the pilots to the decisions, to aid in better estimations.

Also, we proposed a new technique in Wiener Filtering estimation which is totally removing the SNR detectors in the estimation block. Simulations proved that

the if there is a mismatch in the SNR used while calculating the filter coefficients, the resulting mean square error is not changed. The mismatch simulated ranged from 1 dB to 20 dB. We propose a channel estimation system that uses only one robust set of coefficients independent on the channel statistics and independent on the received SNR. The coefficient set will only depend on the pilot locations and the maximum tolerable selectivity and mobile velocity of the user.

In our work, we propose a minor modification to the pilot structure of WiMAX PUSC permutation scheme. Simulations proved that if we added a single pilot at the edge of every symbol, the edge effects resulting from the filtering operations will improve significantly. The edge effects are simply high error floors for high SNR's, resulting from bad estimation at the edges. By adding the proposed pilots, the error floor improved significantly and almost vanished.

The hardware architecture was built for a prototype WiMAX system using a PUSC permutation scheme. The hardware can support other permutation schemes as well as other OFDM systems. In the architecture, many registers and memories can be programmed to support other systems, and the architecture can be interfaced directly to the FFT block available in every OFDM receiver. Hardware simulations proved that the architecture copes with the timing constraints of WiMAX on the targeted STRATIX II Altera FPGA.

Future work includes applying this estimation technique to MIMO OFDM systems, integrating this estimation block with the synchronization block in the receiver and investigating the performance while interchanging the locations of the blocks, i.e. putting the estimation block before and after the synchronization block.

On the hardware level, future work includes attaching the estimation block to a real time WiMAX system, and evaluating the performance of the whole system.

Trying to put a mechanism to automatically adapt the estimation block to the targeted system without manually programming the registers and memories in it.

References

[1] R. M. Jeffrey G. Andrews, Arunabha Ghosh, *Fundamentals of WiMAX: Understanding Broadband Wireless Networking.* Prentice Hall Communications Engineering and Emerging Technologies Series, 2007. 1

[2] M. C. Jeruchim, P. Balaban, and K. S. Shanmugan, Eds., *Simulation of Communication Systems: Modeling, Methodology and Techniques.* Norwell, MA, USA: Kluwer Academic Publishers, 2000. 7

[3] Y. Li, J. Cimini, L.J., and N. Sollenberger, "Robust channel estimation for ofdm systems with rapid dispersive fading channels," *Communications, IEEE Transactions on*, vol. 46, no. 7, pp. 902–915, Jul 1998. 13, 15, 32, 36, 53, 55, 56

[4] A. Papoulis and S. U. Pillai, *Probability, Random Variables, and Stochastic Processes.* McGraw-Hill, 2002. 15, 54

[5] L. Hanzo, T. Keller, M. Muenster, and B.-J. Choi, *OFDM and MC-CDMA for Broadband Multi-User Communications, WLANs and Broadcasting.* New York, NY, USA: John Wiley & Sons, Inc., 2003. 32, 53, 56, 60, 62

[6] "IEEE Std 802.16d-2004:IEEE standard for local and metropolitan area networks," 2004. 40

[7] *IEEE Std 802.16e-2005:IEEE Standard for Local and metropolitan area networks*, IEEE Std. 802.16e, 2005. 40, 53

[8] I. T. U. Recommendation ITU-R M.1225, "Guidelines for evaluation of radio transmission technologies for imt-2000," 1997. 43, 66

[9] T. S. Rappaport, *Wireless Communications: Principles and Practice*. Institute of Electrical and Electronics Engineers, 1996. 43

[10] H. Yaghoobi, "Scalable OFDMA Physical Layer in IEEE 802.16 WirelessMAN," *Intel Technology Journal*, vol. 8, no. 3, pp. 210–212, Aug 2004. 44, 58

[11] Y. Li, "Pilot-symbol-aided channel estimation for ofdm in wireless systems," in *IEEE 49th Vehicular Technology Conference*, vol. 2, Jul 1999, pp. 1131–1135 vol.2. 51

[12] J.-J. van de Beek, O. Edfors, M. Sandell, S. Wilson, and P. Borjesson, "On channel estimation in ofdm systems," *in Proc. of the IEEE Vehicular Technology Conference, VTC 95*, vol. 2, pp. 815–819 vol.2, Jul 1995. 53, 55

[13] P. Hoeher, S. Kaiser, and P. Robertson, "Pilot-symbol-aided channel estimation in time and frequency," in *In Proc. IEEE Global Telecommunications Conference (GLOBECOM 97), Communication Theory Mini-Conference*, 1997, pp. 90–96. 56, 60

[14] D. H. Lee, S. C. Kim, D. C. Park, and Y. il Kim, "A comparative study of channel estimation for mobile wimax system in high mobility," in *10th International Conference on Advanced Communication Technology*, vol. 1, Feb. 2008, pp. 781–785. 57, 59

[15] M. Mohamad, M. Saeed, and A. Priantoro, "Downlink channel estimation and tracking in mobile wimax systems," in *ICCCE 2008. International Conference on Computer and Communication Engineering*, May 2008, pp. 1340–1343. 57, 59

[16] X. Zhou, Z. Zhang, and P. Cheng, "A practical cluster-based channel estimation method for ieee 802.16e," in *8th International Conference on Signal Processing*, vol. 3, Nov. 2006, pp. –. 57

[17] W.-R. Wu, W.-Y. Lin, and S.-L. Kao, "A low-complexity channel-estimation method for ieee 802.16e systems," in *Asia-Pacific Conference on Communications*, Oct. 2007, pp. 85–88. 57

[18] T. Yucek, M. Ozdemir, H. Arslan, and F. Retnasothie, "A comparative study of initial downlink channel estimation algorithms for mobile wimax," in *IEEE Mobile WiMAX Symposium*, March 2007, pp. 32–37. 57

[19] K. Jeong, S. Kim, K. Chung, J. Kim, J. Yu, J. Lee, and S. Seo, "Multipath channel models for wireless local and metropolitan area networks," in *Third International Conference on Information Technology and Applications*, vol. 2, July 2005, pp. 295–298. 65

[20] "stratix ii device family overview". Altera. [Online]. Available: "http://www.altera.com/products/devices/stratix-fpgas/stratix-ii/stratix-ii/overview/st2-overview.html" 75

[21] Dillon Engineerin FFT Cores. Dillon Engineerin. [Online]. Available: "http://www.dilloneng.com/fft_ip" 78

[22] The Xilinx LogiCORE IP FFT. Xilinx. [Online]. Available: http://www.xilinx.com/support/documentation/ip_documentation\/xfft_ds260.pdf 78

[23] Seasolve Mobile WiMAX FFT/IFFT IP Core. Seasolve. [Online]. Available: "http://www.seasolve.com/wimax-fft-ifft-ip-core.html" 78

[24] Altera FFT MegaCore Function. Altera. [Online]. Available: "http://www.altera.com/products/ip/dsp/transforms/m-ham-fft.html" 78

[25] Quartus II Software, Altera. Altera. [Online]. Available: http://www.altera.com/products/software/quartus-ii/subscription-edition/qts-se-index.html 95