

WIMAX TESTBED

By

KOH MINGHAO

FINAL YEAR PROJECT REPORT

**Submitted to the Electrical & Electronics Engineering Programme
in Partial Fulfillment of the Requirements
for the Degree
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)**

**Universiti Teknologi Petronas
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan**

© Copyright 2008

by

Koh Minghao, 2008

CERTIFICATION OF APPROVAL

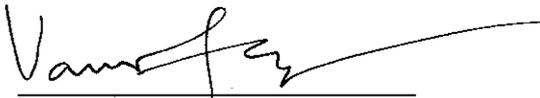
WIMAX TESTBED

by

Koh Minghao

A project dissertation submitted to the
Electrical & Electronics Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Approved:



Assoc. Prof. Dr. Varun Jeoti
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK

June 2008

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

Koh Minghao.

KOH MINGHAO

ABSTRACT

WiMAX, the Worldwide Interoperability for Microwave Access, is a telecommunications technology aimed at providing wireless data over long distances in a variety of ways, from point-to-point links to full mobile cellular type access. It is based on the IEEE 802.16 standard, which is also called WirelessMAN. The name WiMAX was created by the WiMAX Forum, which was formed in June 2001 to promote conformance and interoperability of the standard. The forum describes WiMAX as a standards-based technology enabling the delivery of last mile wireless broadband access as an alternative to cable and DSL. This Final Year Project attempts to simulate via Simulink, the working mechanism of a WiMAX testbed that includes a transmitter, channel and receiver. This undertaking will involve the baseband physical radio link. Rayleigh channel model together with frequency and timing offsets are introduced to the system and a blind receiver will attempt to correct these offsets and provide channel equalization. The testbed will use the Double Sliding Window for timing offset synchronization and the Schmidl & Cox algorithm for Fractional Frequency Offset estimation. The Integer Frequency Offset synchronization is achieved via correlation of the incoming preamble with its local copy whereas Residual Carrier Frequency Offset is estimated using the L^{th} extension method. A linear Channel Estimator is added and combined with all the other blocks to form the testbed. From the results, this testbed matches the standard requirements for the BER when SNR is 18dB or higher. At these SNRs, the receiver side of the testbed is successful in performing the required synchronization and obtaining the same data sent. Sending data with SNR lower than 18dB compromises its performance as the channel equalizer is non-linear. This project also takes the first few steps of hardware implementation by using Real Time Workshop to convert the Simulink model into C codes which run outside MATLAB. In addition, the Double Sliding Window and Schmidl & Cox blocks are converted to Xilinx blocks and proven to be working like their Simulink counterparts.

ACKNOWLEDGEMENTS

My deepest appreciation goes out to my supervisor, Dr. Varun Jeoti for his guidance and assistance throughout the entire duration my final year project. Without him, I would certainly not be able to progress in my tasks.

I would also like to thank Mr. Yew Kuan Min for taking time off his busy schedule to give me lessons on the MATLAB simulations and to discuss certain issues with me in order to improve my project. Last but not least, I would also like to thank my parents for their support and encouragement all through my life.

TABLE OF CONTENTS

LIST OF TABLES	1
LIST OF FIGURES.....	2
LIST OF ABBREVIATIONS	5
CHAPTER 1 INTRODUCTION	6
1.1 WiMAX – An Answer To Broadband Wireless Access	7
1.2 Problem Statement	9
1.3 Objective	9
1.4 Scope of Work.....	10
1.5 Organization of the Final Report.....	10
CHAPTER 2 LITERATURE REVIEW	12
2.1 OFDM Basics	12
2.2 Guard Interval and Cyclic Prefix.....	16
2.3 OFDM System impairments.....	17
2.3.1 Time synchronization Error	18
2.3.2 Carrier Synchronization Error	18
2.3.3 Common Carrier and Timing offset.....	20
2.4 Description of fading channel	20
2.5 Rayleigh Channel	22
2.6 Overview of WiMAX IEEE 802.16 standard	24
CHAPTER 3 METHODOLOGY	26
3.1 Implementation of WiMAX testbed.....	27
3.1.1 WiMAX testbed in Simulink.....	29
3.1.1.1 WiMAX transmitter	30
3.1.1.2 Channel.....	36
3.1.1.3 WiMAX Receiver.....	37
3.1.2 WiMAX testbed in C code.....	45
3.1.3 Double Sliding Window and Schmidl & Cox using Xilinx blocks.....	48
3.2 Performance Analysis Approach of WiMAX Testbed.....	51
3.2.1 Performance analysis for frame timing synchronization	52
3.2.2 Performance analysis for fractional frequency offset synchronization.....	52

3.2.3 Performance analysis for integer frequency offset synchronization.....	52
3.2.4 Performance analysis for residual carrier frequency offset synchronization.....	52
3.2.5 Performance analysis for channel estimation	52
3.2.6 Overall Performance Analysis for the Receiver	53
3.2.7 Validity of WiMAX testbed in Simulink model.....	53
3.2.8 Validity of WiMAX testbed in C code	53
3.2.9 Validity of DSW and S&C in Xilinx blocks	53
CHAPTER 4 RESULTS AND DISCUSSION	54
4.1 Frame timing synchronization.....	54
4.2 Fractional frequency offset synchronization	57
4.3 Integer Frequency Offset synchronization	60
4.4 Residual Carrier Frequency Offset Synchronization.....	62
4.5 Channel Estimation & Correction	64
4.6 Overall performance of Receiver	67
4.7 Validity test for WiMAX testbed in Simulink	68
4.8 Validity test for WiMAX testbed in C codes	73
4.9 Validity test for DSW and S&C in Xilinx blocks.....	76
CHAPTER 5 CONCLUSION AND RECOMMENDATION.....	77
5.1 Conclusion.....	77
5.2 Recommendation.....	78
REFERENCES.....	79
APPENDICES.....	82
Appendix A CYCLIC PREFIX – FROM LINEAR CONVOLUTION TO CYCLIC CONVOLUTION.....	83
Appendix B M-FILE CONTAINING PARAMETERS FOR WIMAX TESTBED	84

LIST OF TABLES

Table 1 Broadband fading parameters	21
Table 2 WiMAX physical layer parameters.....	24
Table 3 WiMAX physical layer parameters for testbed.....	28
Table 4 Parameters for SUI channel models.....	36
Table 5 SUI-3 Parameters	36

LIST OF FIGURES

Figure 1 Applications of WiMAX	8
Figure 2 WiMAX network topology	8
Figure 3 Spectrum of an OFDM signal.....	13
Figure 4 Channel induces ISI without cyclic prefix.....	15
Figure 5 OFDM symbol with cyclic prefix prevents ISI	15
Figure 6 Addition of CP to the OFDM symbol.....	16
Figure 7 Elimination of ISI and ICI with cyclic prefix.....	17
Figure 8 Synchronization errors in OFDM systems	17
Figure 9 ICI arises when there is carrier synchronization error.....	19
Figure 10 One second Rayleigh fading with max Doppler shift of 10Hz.....	23
Figure 11 One second Rayleigh fading with max Doppler shift of 100Hz.....	23
Figure 12 Process flow chart for FYP.....	26
Figure 13 WiMAX transceiver structure	27
Figure 14 WiMAX testbed structure.....	29
Figure 15 Data block in Simulink	30
Figure 16 Randomizer in logic diagram.....	31
Figure 17 Randomizer in Simulink	31
Figure 18 Block Encoder in Simulink.....	32
Figure 19 Convolutional Encoder in Simulink	32
Figure 20 Interleaver in Simulink	34
Figure 21 Structure of the training symbol. Top: For original Schmidl and Cox scheme Bottom: New proposed method	35
Figure 22 Stem diagram of the Training Symbol.....	35
Figure 23 Block creating OFDM symbols in Simulink. The training symbol is added in front of the data.....	35
Figure 24 Channel Block in Simulink.....	36
Figure 25 Inner Receiver Design in Simulink.....	37
Figure 26 Double Sliding Window Packet Detection.....	38
Figure 27 Implementation of Double Sliding Window in Simulink.....	39
Figure 28 Implementation of Schmidl & Cox technique in Simulink	40
Figure 29 FFO Correction Implementation in Simulink.....	41
Figure 30 Outer Receiver Design in Simulink.....	42

Figure 31 IFO Estimation implementation in Simulink.....	43
Figure 32 RCFO Estimation Implementation in Simulink	44
Figure 33 Channel Estimation implementation in Simulink.....	45
Figure 34 Process of generating source code from Simulink models using Real-Time Workshop.....	46
Figure 35 WiMAX testbed model for C code generation.....	46
Figure 36 Configuration Parameters Window for RTW usage.....	47
Figure 37 Real-Time Workshop Report.....	47
Figure 38 Model showing System Generator block.....	49
Figure 39 Double Sliding Window Implementation using Xilinx Blockset.....	50
Figure 40 Schmidl & Cox Implementation using Xilinx Blockset (A)	50
Figure 41 Schmidl & Cox Implementation using Xilinx Blockset (B).....	51
Figure 42 m_n metric at SNR = 30dB	55
Figure 43 m_n metric at SNR = 15dB	55
Figure 44 Time domain plot before and after timing synchronization	56
Figure 45 Performance of timing offset estimator vs SNR.....	57
Figure 46 $M(d)$ and $-<P(d)$ plots from Schmidl & Cox method at SNR = 30dB.....	58
Figure 47 Enlarged $-<P(d)$ and resulting FFO plots for SNR = 30dB	58
Figure 48 $M(d)$ and $-<P(d)$ plots for Schmidl & Cox at SNR = 15dB.....	59
Figure 49 Enlarged $-<P(d)$ and resulting FFO plots for SNR = 15dB	59
Figure 50 Estimation performance of FFO estimator versus SNR	60
Figure 51 Decision metric for IFO at SNR = 30dB	61
Figure 52 Decision metric for IFO at SNR = 15dB	61
Figure 53 Performance of IFO estimator versus SNR	62
Figure 54 Graph for phase of pilot subcarriers at SNR = 30dB.....	63
Figure 55 Graph for phase of pilot subcarriers at SNR = 15dB.....	63
Figure 56 Performance of RCFO estimator versus SNR	64
Figure 57 Real and imaginary plot for data before Channel Correction and the Channel Estimate at SNR = 30dB.....	65
Figure 58 Real and imaginary plot for data sent at transmitter and after Channel Correction at SNR = 30dB.....	65
Figure 59 Real and imaginary plot for data before Channel Correction and the Channel Estimate at SNR = 15dB.....	66
Figure 60 Real and imaginary plot for data sent at transmitter and after Channel Correction at SNR = 15dB.....	66

Figure 61 Performance of Channel Estimator vs SNR	67
Figure 62 BER vs SNR at the receiver of WiMAX testbed.....	68
Figure 63 Introduction of timing and frequency offsets (Test Case 1).....	68
Figure 64 Results of simulation in Array Editor (Test Case 1).....	69
Figure 65 Scatter plot at the transmitter.....	69
Figure 66 Scatter plot at the receiver with all corrections performed at SNR = 30dB	70
Figure 67 Introduction of timing and frequency offsets (Test Case 2).....	70
Figure 68 Results of simulation in Array Editor (Test Case 2).....	71
Figure 69 Scatter plot at the receiver with all corrections performed at SNR = 20dB	71
Figure 70 Results in Array Editor showing input data matches output data (A)	72
Figure 71 Results in Array Editor showing input data matches output data (B)	73
Figure 72 Screenshot of wimax_rtw.exe running in Command Prompt.....	73
Figure 73 Results in Array Editor from file wimax_rtw.mat.....	74
Figure 74 Input data matching output data from file wimax_rtw.mat (A).....	75
Figure 75 Input data matching output data from file wimax_rtw.mat (B).....	75
Figure 76 Introduction of timing and frequency offsets (Test Case 3).....	76
Figure 77 Results of simulation in Array Editor (Test Case 3).....	76
Figure 78 Introduction of timing and frequency offsets (Test Case 4).....	76
Figure 79 Results of simulation in Array Editor (Test Case 4).....	76

LIST OF ABBREVIATIONS

AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BWA	Broadband Wireless Access
DSL	Digital Subscriber Line
DSW	Double Sliding Window
FFT	Fast Fourier Transform
FFO	Fractional Frequency Offset
FTTH	Fiber to the home
HDTV	High Definition Television
ICI	Inter Carrier Interference
IEEE	Institute of Electrical & Electronic Engineers
IFO	Integer Frequency Offset
ISI	Inter Symbol Interference
LAN	Local Area Network
LOS	Line of Sight
NLOS	Non Line of Sight
OFDM	Orthogonal Frequency Division Multiplexing
PN	Pseudo-Noise
QPSK	Quaternary Phase Shift Keying
QoS	Quality of Service
RCFO	Residual Carrier Frequency Offset
RTW	Real-Time Workshop
SC	Subcarrier
S&C	Schmidl & Cox
SNR	Signal to Noise Ratio
SUI	Stanford University Interim
TS	Training Symbol
VDSL	Very high data rate digital subscriber loop
VoD	Voice on Demand
VoIP	Voice over Internet Protocol
WiMAX	Worldwide Interoperability for Microwave Access

CHAPTER 1

INTRODUCTION

The predominant mass-market broadband access technologies of today include DSL, which delivers broadband over twisted-pair telephone wires and cable modem, which delivers broadband over coaxial cable TV plant. Both of these technologies typically provide up to few megabits per second of data to each user and continue to make advances of up to several tens of megabits possible. Worldwide, the number of broadband users is more than 200 million and is projected to grow beyond 400 million by 2010 [1].

The overwhelming response in using broadband access comes because it not only provides faster Web surfing and quicker video downloads; it also enables several multimedia applications such as real-time audio and video streaming, multimedia conferencing and interactive gaming. Broadband are also being used for video telephony using VoIP technology and advanced systems such as FTTH and VDSL that enables applications such as HDTV and VoD [1].

The availability of a wireless solution will definitely accelerate the growth of broadband usage [15]. Broadband wireless brings the broadband experience to the wireless context and there are basically two different types of broadband wireless service. The first attempts to provide a set of services similar to that of the traditional fixed-line broadband but using wireless as a medium of transmission. This is known as fixed wireless broadband. The second type is mobile broadband and offers the additional functionality of portability, nomadicity and mobility. WiMAX is designed to accommodate for both fixed and mobile applications [4].

1.1 WiMAX – An Answer To Broadband Wireless Access

WiMAX is a standards-based technology that promises broadband connectivity without the need for direct LOS between subscriber terminals and a single BTS will suffice to serve hundreds of users [17]. It will help accelerate the introduction of wireless broadband equipment into the marketplace, speeding up last-mile broadband deployment worldwide by enabling service providers to increase system performance and reliability while reducing their equipment costs and investment risks.

Due to its wireless nature, the most outstanding advantage of WiMAX compared to traditional wireline such as DSL and cable modem is its low cost of installation and maintenance, enabling it to serve as a launch pad for companies to build networks at areas left out due to geographical or physical obstacles [15, 17]. Therefore, broadband wireless access can extend networks to areas which are too remote or difficult to reach where it has been economically unfeasible before [10]. Furthermore, networks can be created in a short time by a small number of BTS to create high-capacity wireless access systems.

WiMAX can provide BWA up to 30 miles (50 km) for fixed stations and 3 – 10 miles (5 – 15km) for mobile stations. In contrast, the WiFi wireless local area network is limited to only 100 – 300 feet (30 – 100m). Besides that, it can be expected to deliver a capacity of up to 40Mbps per channel which is enough bandwidth to simultaneously support hundreds of businesses with T1 speed connectivity and thousands of residences with DSL speed connectivity [16].

WiMAX has the potential to impact all forms of telecommunication as it offers competitive internet access and can bypass the telephone company's copper wire network. As the industry moves along quadruple-play service bundles (voice, data, video, mobility), service providers are likely to find WiMAX attractive. Flexible channel bandwidths and multiple level of QoS support also allows WiMAX to be used for differentiated bandwidth and low-latency entertainment applications. For example, streaming audio services delivered to MP3 players and video services to portable media players [1].

With the roll out of 3G still in progress, WiMAX is envisioned as the 4th Generation system and will extend its applications to the following:

- Broadband for residential, SOHO and SME markets
- T1 or fractional T1-like services to businesses
- Wireless backhaul for Wi-Fi hotspots
- VoIP and IPTV services

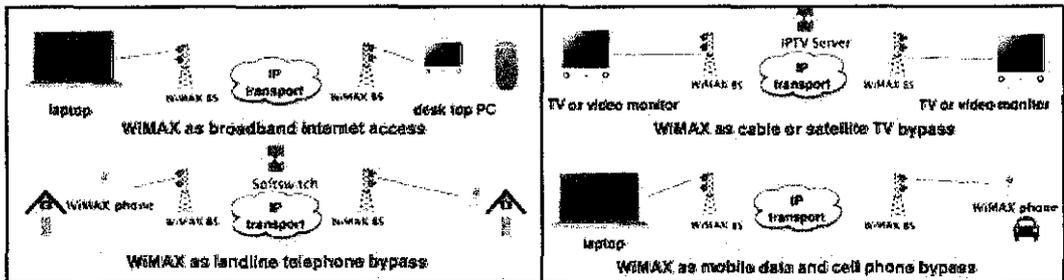


Figure 1 Applications of WiMAX

A WiMAX base station is connected to public networks using optical fiber, cable, microwave link or any other high speed point-to-point connectivity referred to as backhaul. In few cases such as mesh networks, point-to-multi point connectivity is also uses a backhaul. A base station serves CPEs using NLOS or LOS point-to-multipoint connectivity and this connection is referred to as the last mile [1].

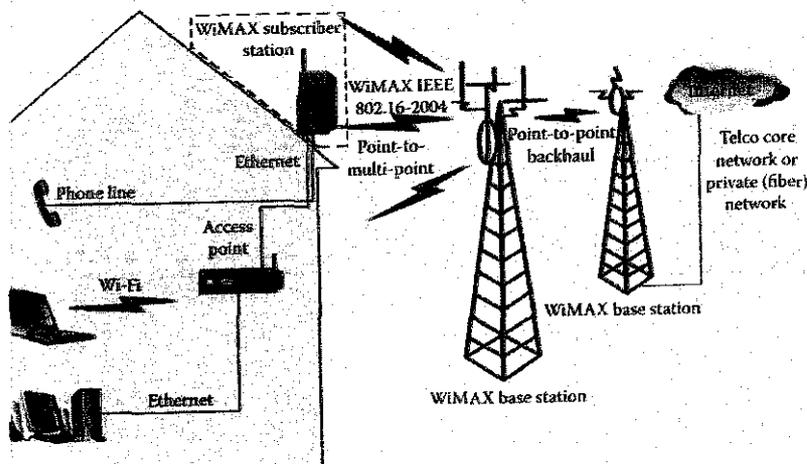


Figure 2 WiMAX network topology

1.2 Problem Statement

The first and most fundamental challenge for broadband wireless comes from the transmission medium itself. In wired communications channels, a physical connection such as a copper wire, guides the signal from transmitter to receiver but wireless communication systems rely on complex radio wave propagation mechanisms for traversing the intervening space. The requirements of most broadband wireless services are that the signal must travel through challenging NLOS conditions.

Therefore, synchronization and channel correction, which will be performed by the receiver, is a critical step that must be performed to ensure that the transmitted data symbols can be properly recovered. Hostile wireless channel has always been proven to be the bottleneck for high speed wireless systems and the need for synchronization as well as channel estimation is the key to attaining accurate data [2, 6, 7, 11].

In addition, before hardware implementation and evaluation can take place, a testbed (transmitter, receiver and channel) is necessary to simulate and identify design issues. So far, there hasn't been many technical papers on the end-to-end implementation of a WiMAX system. Most research only focuses on one particular section such as the timing offset estimation. This project aims to have a end-to-end realization of a WiMAX testbed with all the necessary blocks for synchronization.

1.3 Objective

This final year project attempts to design the WiMAX testbed consisting of a transmitter, channel and receiver in Simulink according to the IEEE 802.16 standard. Combining all the blocks to obtain a working testbed, is not a trivial task as many theory and simulation limitations must be looked into.

The receiver must be able to perform synchronization and channel correction to obtain the original data sent by the WiMAX transmitter. Performance analysis for the system will also be performed.

The next goal is to move towards hardware implementation by two mainstream methods. By using RTW to convert the Simulink model into C codes and by translating blocks that perform the Double Sliding Window and Schmidl & Cox algorithm into Xilinx blocks.

1.4 Scope of Work

The following areas are my scope of work:

- Design a WiMAX testbed in Simulink consisting of a transmitter, receiver and channel according to the specifications in the IEEE 802.16 standard.
- Implement and modify the Timing and FFO estimation and correction blocks designed by the previous FYP student
- Design the blocks for IFO, Channel and RCFO estimation and correction. Perform performance evaluation on these synchronization methods
- Using Real-time workshop, convert the Simulink model into C codes showing hardware co-simulation is possible.
- Translate the Double Sliding Window and Schmidl & Cox blocks from Simulink into Xilinx Blocks as a step towards hardware implementation.
- Work is focused on the physical baseband level only

1.5 Organization of the Final Report

This final report describes the work done in the WiMAX testbed design project.

Chapter 1 highlights about broadband wireless access and WiMAX as one of the standards for BWA as well as the importance and advantages of WiMAX. The significance of synchronization at the receiver and issues relating to it are stated in the *Problem Statement*. The *Objective* and *Scope of Work* shows the goals of this Final Year Project.

Chapter 2 Literature Review will give an insight into the OFDM theory that is the underlying technology behind WiMAX. Next, an overview of the 802.16 standard is shown, highlighting key parameters. After that, synchronization errors involving timing and frequency offset are explained. It also includes the discussion about the

Rayleigh channel.

Chapter 3 Methodology defines how this project will be implemented. The receiver will be modeled and simulated in Simulink with offsets and impairments introduced into the system. Furthermore, This system will run on Rayleigh and AWGN channels. It also demonstrates how the testbed is evaluated.

Chapter 4 Results and Discussion will show the various decision metrics used for synchronization. The scatter plots before and after demodulation, in addition to the capabilities of the testbed, are also shown in this section

Chapter 5 Conclusion will give a summation of this project and recommendations for future work.

CHAPTER 2

LITERATURE REVIEW

The WiMAX standard utilizes OFDM which is a form of multicarrier modulation. Multicarrier modulation escapes the penalty of heavy equalization as compared to single carrier systems. This chapter will mention about OFDM theory and the cyclic prefix which is used to combat ISI. Synchronization errors in OFDM systems and the key parameters of WiMAX systems will also be presented. Included in this chapter is also a discussion about the Rayleigh channel.

2.1 OFDM Basics

The basic principle of orthogonal frequency division multiplexing (OFDM) is to split a high-rate datastream into a number of lower rate streams that are transmitted simultaneously over a number of subcarriers. Since the symbol duration increases for the lower rate parallel subcarriers, the relative amount of dispersion time caused by multipath delay spread is decreased. Intersymbol interference (ISI) is eliminated almost completely by introducing a guard time in every OFDM symbol [2]. In the guard time, the OFDM symbol is cyclically extended to avoid intercarrier interference (ICI).

In OFDM, each sub-band will experience flat fading. The flat fading condition is favored in communication systems because it reduces the complex multi-tap equalizer in the frequency selective fading case compared to a simple one-tap equalizer [3]. A high data rate stream is divided into several interleaved bit streams of lower data rate. These lower rate streams are then used to modulate several carriers simultaneously and are sent these over the many different sub-bands.

The spectrum of a single symbol is a convolution of a group of Dirac pulses located at subcarrier frequencies with the spectrum of a square pulse that is one for a T -second period and zero otherwise. The amplitude spectrum of the square pulse is equal to $\text{sinc}(\pi fT)$ which has zeroes for all frequencies f that are an integer multiple of $1/T$ [6]. At maximum of each subcarrier spectrum, all other subcarrier spectra are zero. The effect is shown in the Figure 3, which shows the overlapping sinc spectra of individual subcarriers.

Basically, Figure 3 also shows that the OFDM spectrum fulfils Nyquist criterion for an ISI free pulse shape. By having the maximum of one subcarrier correspond to the zero crossing of all the others, we also eliminate ICI [6].

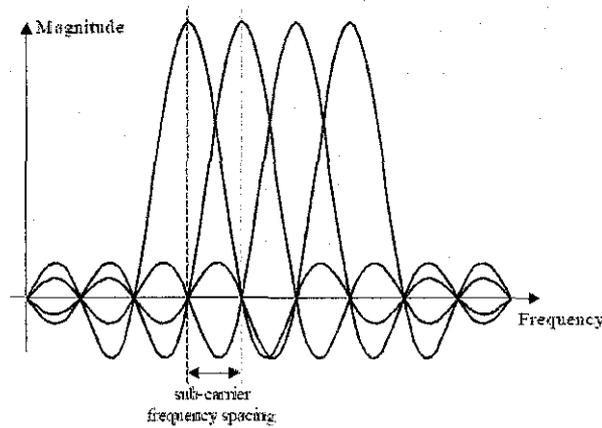


Figure 3 Spectrum of an OFDM signal

Mathematically, the complex envelop of the OFDM signal is written in the formula below

$$\begin{aligned}
 & s_k(t - KT) \\
 &= w(t - kT) \sum_{i=-N/2}^{N/2-1} x_{i,k} e^{j2\pi \left[\frac{i}{T_{FFT}} \right] (t-KT)} \quad kT - T_{win} - T_{guard} \leq t \leq kT + T_{FFT} + T_{win}
 \end{aligned} \tag{2-1}$$

- T Symbol length; time between two consecutive OFDM symbols
- T_{FFT} FFT-time; effective part of the OFDM symbol
- T_{guard} Guard-interval; duration of the cyclic prefix
- T_{win} Duration of windowed prefix/postfix for spectral shaping

f_c	Center frequency of the occupied frequency spectrum
$F = 1/T_{FFT}$	Frequency spacing between adjacent sub-carriers
N_{FFT}	FFT-length; number of FFT points
k	Index on transmitted symbol
I	Index on sub-carrier; $i \{-N/2, -N/2+1, \dots, -1, 0, 1, \dots, N/2-1\}$
$x_{i,k}$	Signal constellation point; complex {data, pilot, null} symbol modulated on the i -th subcarrier of the k -th OFDM symbol

The window in the equation is a raised cosine function. It has a value of 1 within T_{FFT} so it is not very significant in the modulation and the demodulation process. The window is incorporated in order to reduce the level of side lobes and therefore reduce the signal power transmitted out of band.

The effect of a time variant multipath channel can be modeled by a convolution with a channel impulse response of $h(t,\tau)$ and AWGN, $n(t)$.

$$\begin{aligned}
r(t) &= h(\tau, t) * s(t) + n(t) \\
&= \int_0^{\tau_{max}} h(\tau, t) s(t - \tau) d\tau + n(t)
\end{aligned} \tag{2-2}$$

Since a cyclic prefix is added to each OFDM symbol such that $T_{guard} \gg \tau_{max}$, no ISI occurs. This is illustrated in Figure 5 and Figure 6. With this, the channel impulse response is simplified to $h(\tau)$. Consequently, the OFDM demodulation can be shown as follows

$$\begin{aligned}
y_{i,k} &= \frac{1}{T_{FFT}} \int_{i=kT}^{kT+T_{FFT}} r(t) e^{-j2\pi(t-kT)/T_{FFT}} dt \\
&= \frac{1}{T_{FFT}} \int_{i=kT}^{kT+T_{FFT}} \left[\int_{\tau=0}^{\tau_{max}} h_k(\tau) s(t - \tau) d\tau + n(t) \right] e^{-j2\pi(t-kT)/T_{FFT}} dt \\
&= \frac{1}{T_{FFT}} \int_{i=kT}^{kT+T_{FFT}} \left[\int_{\tau=0}^{\tau_{max}} h_k(\tau) \sum x_{i,k} e^{-j2\pi\left(\frac{t}{T_{FFT}}\right)(t-kT-\tau)} d\tau \right] e^{-j2\pi(t-kT)/T_{FFT}} dt + \\
&\frac{1}{T_{FFT}} \int_{i=kT}^{kT+T_{FFT}} n(t) e^{-j2\pi(t-kT)/T_{FFT}} dt
\end{aligned} \tag{2-3}$$

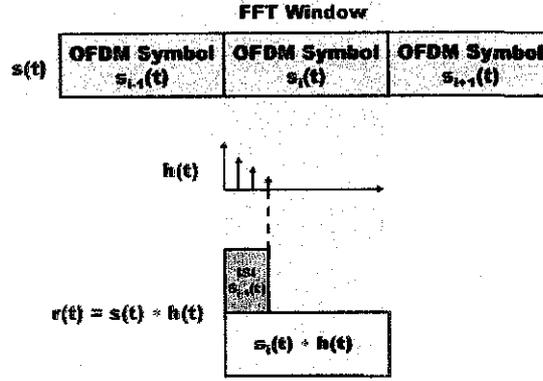


Figure 4 Channel induces ISI without cyclic prefix

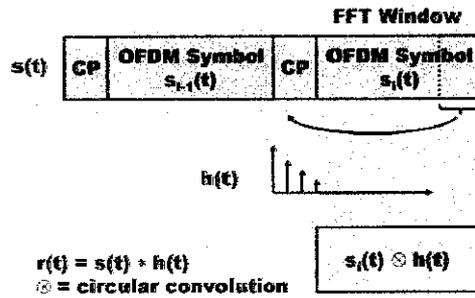


Figure 5 OFDM symbol with cyclic prefix prevents ISI

The second term represents AWGN and the integration of the $n(t)$ term yields $n_{i,k}$. Substituting $u = t-kT$, and changing the order of summation and integration yields:

$$y_{i,k} = \sum_{i'=-N/2}^{N/2-1} x_{i',k} \frac{1}{T_{FFT}} \int_{u=0}^{kT+T_{FFT}} \left[\int_{\tau=0}^{\tau_{max}} h_k(\tau) \sum x_{i',k} e^{-j2\pi\tau/T_{FFT}} d\tau \right] e^{-j2\pi(i-i')u/T_{FFT}} du + n_{i,k} \quad (2-4)$$

The integration of the $h(\tau)$ term is the Fourier transform of the impulse response, which yields $h_{i,k}$. Noting this, the OFDM demodulation can then be resolved to:

$$y_{i,k} = \sum_{i'=-N/2}^{N/2-1} x_{i',k} h_{i',k} \frac{1}{T_{FFT}} \int_{u=0}^{T_{FFT}} e^{-j2\pi(i-i')u/T_{FFT}} du + n_{i,k}$$

$$y_{i,k} = x_{i,k} + n_{i,k} \quad (2-5)$$

The value of integration will be 1 if and only if $i=i'$ and 0 otherwise. This proves the orthogonal property. For $i \neq i'$, there is an integer number of sinusoids within T_{FFT} and the integration will become 0.

2.2 Guard Interval and Cyclic Prefix

Passing the signal through a time-dispersive channel causes ISI. In an OFDM system, a loss of orthogonality appears due to ISI resulting in ICI. For a given system bandwidth, the symbol rate is much lower than a single transmission scheme. This is because the OFDM system bandwidth is broken into N_c subcarriers resulting in a symbol rate that is N_c times lower. This low symbol rate makes OFDM naturally resistant to effects of ISI caused by multipath propagation [2].

The multiple signals that appear due to multipath propagation arrive at the receiver at different times, spreading the symbol boundaries and causing energy leakage between the OFDM symbols. In an OFDM signal, the amplitude and phase must remain constant over a period of the symbol in order to maintain the orthogonality of the subcarriers. If they are not constant, the spectral shape will not have null at the correct frequencies, resulting in ICI [9].

To combat this issue, a guard period to the start of each symbol is added. This guard period, which is called cyclic prefix (CP), is a copy of the last part of the OFDM symbol [9]. Figure 6 shows the structure of the OFDM symbol. The total length of the symbol can be written as $T_{\text{sym}} = T_g + T_b$, where T_{sym} is the total length of the symbols in samples, T_g is the length of the guard period in samples and T_b is the size of the IFFT used to generate the OFDM symbol, representing useful OFDM symbol time.

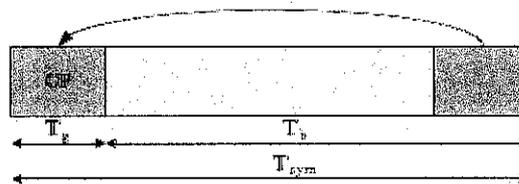


Figure 6 Addition of CP to the OFDM symbol

Figure 7 shows that the delayed replicas of the OFDM symbol which has been added with CP always have an integer number of cycles within the FFT interval, as long as the delay spread is smaller than the guard time. The effective part of the

received signal can be seen as the cyclic convolution of the transmitted OFDM symbol by the channel impulse response [2].

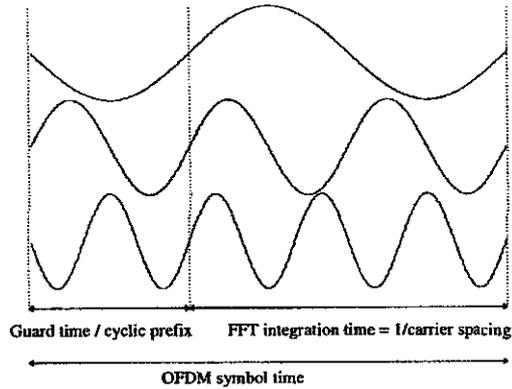


Figure 7 Elimination of ISI and ICI with cyclic prefix

2.3 OFDM System impairments

Before an OFDM receiver can demodulate the subcarriers, it has to perform at least two synchronization tasks. Firstly, it has to find out where the symbol boundaries and optimal timing instants are to minimize ICI as well as ISI. Secondly, it has to estimate and correct for the carrier frequency offset of the received signal. OFDM is highly sensitive to time and frequency synchronization errors, especially when demodulation of an OFDM signal with an offset in the frequency can lead to high bit error rate [1]. The following diagram shows synchronization errors and its sources.

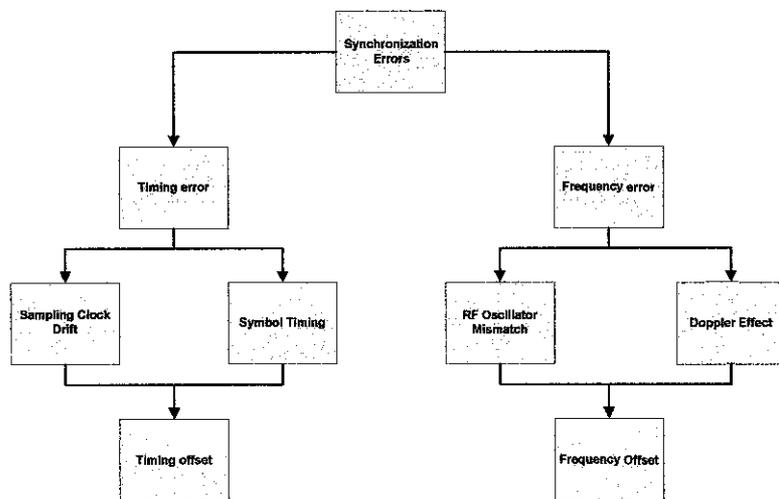


Figure 8 Synchronization errors in OFDM systems

2.3.1 Time synchronization Error

FFT time synchronization, frame detection and timing offset detection will be used interchangeably from here on and will carry the same meaning. Frame detection is used to determine the symbol boundary so that correct samples for a symbol frame can be taken. Any deviation from this boundary means that the sensitivity to delay spread increases, so the system can handle less delay spread than what it was designed to accommodate. With this, any OFDM system should be designed so that the timing error is small compared to guard interval in order to maintain robustness.

The FFT time synchronization error can be modeled as a shift in the interval of integration of the matched filter [6]. For a timing error of δt , the interval $t \in [kT + \delta t, kT + T_{FFT} + \delta t]$.

$$y_{i,k} = \frac{1}{T_{FFT}} \int_{t=kT+\delta t}^{kT+T_{FFT}+\delta t} r(t) e^{-j2\pi(t-kT+\delta t)/T_{FFT}} dt \quad (2-6)$$

The OFDM transceiver should be designed such that $\delta t < T_{guard}$, in which no ISI occurs as a result of FFT time synchronization error. From (2-3), the following is obtained.

$$y_{i,k} = \sum_{i'=-N/2}^{N/2-1} x_{i',k} \frac{1}{T_{FFT}} \int_{u=0}^{kT+T_{FFT}} \left[\int_{\tau=0}^{\tau_{max}} h_k(\tau) \sum x_{i',k} e^{-j2\pi\tau/T_{FFT}} d\tau \right] e^{-j2\pi[(t-i')u+t\delta t]/T_{FFT}} du + n_{i,k} \quad (2-7)$$

With this, the expression for the demodulated signal constellations in case of timing error becomes

$$\begin{aligned} y_{i,k} &= x_{i,k} h_{i,k} e^{-j2\pi\delta t/T_{FFT}} + n_{i,k} \\ &= x_{i,k} h_{i,k} e^{-j2\pi\delta t/N} + n_{i,k} \end{aligned} \quad (2-8)$$

2.3.2 Carrier Synchronization Error

Carrier synchronization error occurs because of a small deviation of frequency between local oscillator in the transmitter and receiver. It also can be due to Doppler effect when mobility is considered. This can be visualized as an error in the frequency instant, where the signal is sampled during demodulation via FFT. The carrier offset can be modeled as a frequency shift δf and phase shift θ in the received signal.

Putting all this into (2-3), gives

$$\begin{aligned}
 y_{i,k} &= \frac{1}{T_{FFT}} \int_{i=kT}^{kT+T_{FFT}} r(t) e^{-j(2\pi\delta f t + \theta)} e^{-j2\pi(t-kT-\delta)/T_{FFT}} dt \\
 &= e^{-j(2\pi\theta)} \frac{1}{T_{FFT}} \int_{i=kT}^{kT+T_{FFT}} \left[\int_{\tau=0}^{\tau_{max}} h_k(\tau) s(t-\tau) d\tau + n(t) \right] e^{-j(2\pi\delta f t)} e^{-j2\pi(t-kT-\delta)/T_{FFT}} dt \\
 &= e^{-j(\theta+2\pi\delta kT)} \sum_{i'=-N/2}^{N/2-1} x_{i,k} h_{i,k} \frac{1}{T_{FFT}} \int_{u=0}^{T_{FFT}} e^{-j2\pi\left(\frac{i-i'}{T_{FFT}} - \delta f\right)u} du + n_{i,k}
 \end{aligned} \tag{2-9}$$

From (2-9), it can be seen that the desired amplitude of the constellation points will be increased and there will be contributions from adjacent subcarriers, leading to ICI. The integration will not be 1 for $i=i'$ and neither is it 0 for $i \neq i'$, hence there will be a loss of orthogonality between subcarriers.

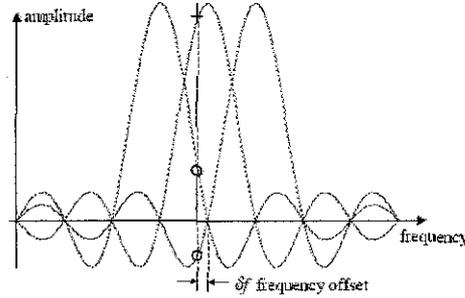


Figure 9 ICI arises when there is carrier synchronization error

The ICI term can be seen as an additional noise term and can be represented as a degradation of SNR [6]. Evaluation of the phase rotation and attenuation due to frequency error yields

$$y_{i,k} = x_{i,k} h_{i,k} \text{sinc}(\delta f T_{FFT}) e^{j(\theta+2\pi\delta f(kT+T_{FFT}/2))} e^{j(i-i')u/T_{FFT}} + n'_{i,k} \tag{2-10}$$

Since

$$\begin{aligned}
 \frac{1}{T_{FFT}} \int_0^{T_{FFT}} e^{j2\pi\delta f t} dt &= \frac{1}{j2\pi\delta f T_{FFT}} [e^{j2\pi\delta f T_{FFT}} - 1] \\
 &= e^{j2\pi\delta f T_{FFT}} \frac{\sin \pi\delta f T_{FFT}}{\pi\delta f T_{FFT}} \\
 &= e^{j2\pi\delta f T_{FFT}} \text{sinc} \pi\delta f T_{FFT}
 \end{aligned} \tag{2-11}$$

2.3.3 Common Carrier and Timing offset

Synchronization errors involving both FFT time synchronization and carrier synchronization error. The synchronization errors in the system transfer function can be generalized as follows.

$$y_{i,\delta f_i,k} = x_{i,k} h_{i,k} \sin c[(\delta f - \delta f_i F)T_{FFT}] e^{j\psi_{i,k}} + n'_{i,k} \quad (2-12)$$

where

$$\psi_{i,k} = \theta + 2\pi\delta f \left(kT + \frac{T_{FFT}}{2} + \delta t \right) + 2\pi\delta\alpha \frac{i}{T_{FFT}} \quad (2-13)$$

Expressing (2-13) in terms of time samples

$$\psi_{i,k} = \theta + 2\pi\delta f' \left(\frac{1}{2} + k \frac{N + N_{guard} + N_{win}}{N} + \frac{\delta\alpha'}{N} \right) + 2\pi\delta\alpha' \quad (2-14)$$

2.4 Description of fading channel

In a realistic wireless radio environment, a single received signal is composed of a number of scattered waves, caused by the reflection and diffraction of the original transmitted signal by objects in the surrounding geographical area [3].

These multipath waves are combined at the receiver to give a resultant signal that can widely vary in amplitude and phase. Physical factors influencing the characteristics of the fading experienced by the transmitter are multipath propagation, mobility of the reflecting objects and scatterers, and the relative motion between the transmitter and the receiver [9].

The presence of reflecting objects and scatterers in the wireless channel causes a constant change in the propagation environment. This changing environment alters the signal energy in amplitude, phase, and time, and as a result, multipath propagation occurs causing signal fading [9]. The transmitted signal arrives at the receiver via multiple propagation paths, each of which has an associated time delay. Because the received signal is spread in time due to the multipath scatterers at different delays, the channel is said to be time dispersive.

The difference between the largest and the smallest among these delays defines the maximum delay spread. τ_{RMS} gives a measure of width of the channel response in time. A large τ_{RMS} implies that a highly dispersive channel in time and a long channel impulse response [1].

On the other hand, when the receiver and the transmitter are in relative motion, the received signal is subject to a constant frequency shift, called the Doppler shift [1]. Therefore, as it occurs in the time domain, the Doppler spread is defined as the difference between the largest and the smallest among these frequency shifts,

$$f_d = f_M \cos \varphi \quad (2-15)$$

- $f_M = f_c v/c$ is the maximum Doppler shift,
- v is the vehicle speed,
- f_c is the carrier frequency,
- c is the speed of light, and
- φ is the arrival angle of the received signal component.

Channel coherence time and bandwidth gives a period of time over which the channel is significantly correlated. Channel coherence time is

$$T_c = \frac{1}{f_d} \quad (2-16)$$

Channel coherence bandwidth is given as

$$B_c = \frac{1}{5\tau_{RMS}} \quad (2-17)$$

Table 1 Broadband fading parameters

Quantity	If "Large"	If "Small"
Delay spread, τ	If $\tau \gg T$, frequency selective	If $\tau \ll T$, frequency flat
Coherence bandwidth, B_c	If $1/B_c \ll T$, frequency flat	If $1/B_c \gg T$, frequency selective
Doppler spread, f_d	If $f_d \gg c$, fast fading	If $f_d \leq c$, slow fading
Coherence time, T_c	If $T_c \gg T$, slow fading	If $T_c \leq T$, fast fading

For flat fading channels, the spectral characteristics of the transmitted signal are preserved when it propagates through the channel, and only the received signal power fluctuates due to the multipath effects. For this reason, flat fading channels are also known as amplitude varying channels [9].

Frequency-selective fading is due to time dispersion of the transmitted symbols within the channel, thus inducing to inter-symbol interference (ISI). However, when communicating with OFDM techniques, the effects of frequency-selective channel conditions can be decreased [9]. Since the signal is split into many narrowband subchannels, the channel can be considered as constant (flat) over each OFDM subchannel, provided that the respective conditions for flat fading channels are accomplished.

With fast fading, the transmitter and receiver move fast relative to each other and a large burden is placed on the channel estimation algorithms [1].

2.5 Rayleigh Channel

Rayleigh fading is a statistical model for the effect of a propagation environment on a radio signal, such as that used by wireless devices. It assumes that the magnitude of a signal that has passed through such a transmission medium will vary randomly, or fade, according to a Rayleigh distribution. Rayleigh fading is most applicable when there is no dominant propagation along a line of sight between the transmitter and receiver [9].

Wireless channels can be characterized with tap coefficients that are complex valued Gaussian random variables. A channel model where there are only non line-of-sight communications is characterized by a Rayleigh distribution [3]. On the contrary, if dominating paths are present, the channel coefficients are modeled by a Ricean distribution.

As already mentioned, a Rayleigh distribution is normally used to model NLOS communications. It is statistically characterized by a fading amplitude, $\alpha(t)$, modeled with a Rayleigh probability distribution, which has zero-mean Gaussian components.

Furthermore, the phase, $\phi(t)$, is uniformly distributed over the interval $(0, 2\pi)$. The fading amplitude is described by the probability density function (pdf):

$$\begin{aligned}
 & f_{RAY}(a) \\
 & = \left\{ \begin{array}{ll} \frac{a}{\sigma^2} \exp\left(-\frac{a^2}{2\sigma^2}\right) & \text{if } a \geq 0 \\ 0 & \text{if } a < 0 \end{array} \right. \quad (2-18)
 \end{aligned}$$

The figures show the power variation over 1 second of a constant signal after passing through a single-path Rayleigh fading channel with a maximum Doppler shift of 10 Hz and 100 Hz.

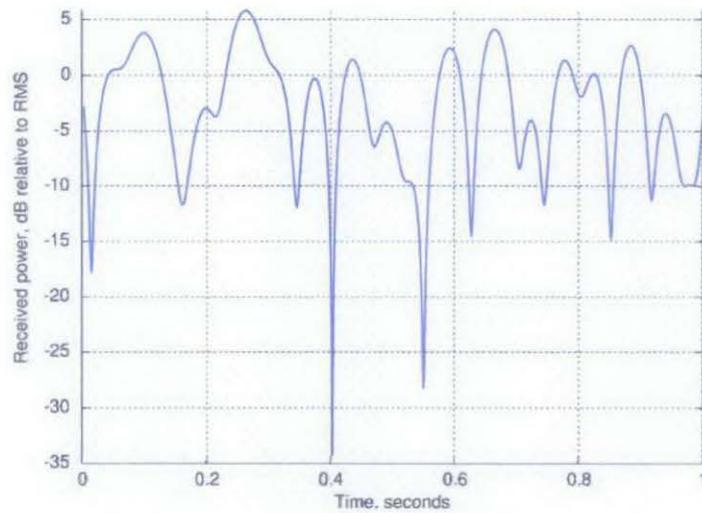


Figure 10 One second Rayleigh fading with max Doppler shift of 10Hz

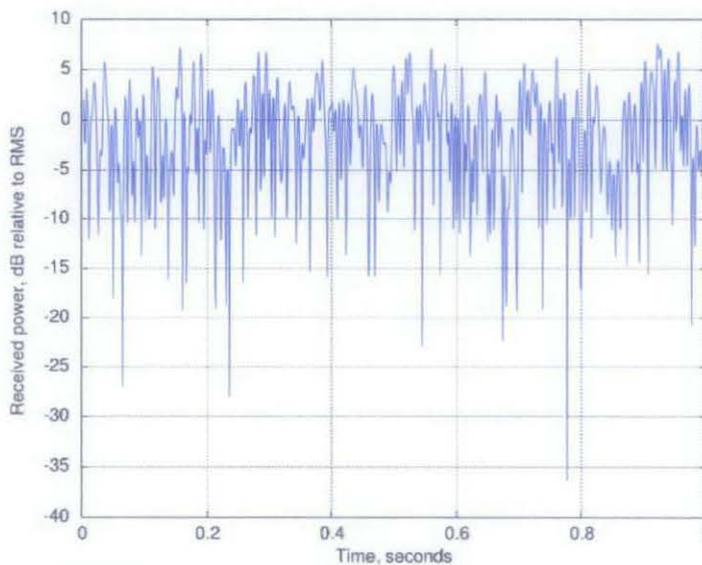


Figure 11 One second Rayleigh fading with max Doppler shift of 100Hz

2.6 Overview of WiMAX IEEE 802.16 standard

The IEEE 802.16 standard specifies three air interfaces which are single-carrier modulation, 256-point transform OFDM, and 2048-point transform OFDMA [4]. Nevertheless, mainstream implementations involve the 256 point FFT OFDM and will be standard used for the testbed in this Final Year Project. The following table lists basic data of the IEEE 802.16 Standards [1].

Table 2 WiMAX physical layer parameters

	802.16	802.16-2004	802.16e-2004
Status	Completed December 2001	Completed June 2004	Completed December 2005
Frequency band	10GHz – 66GHz	2GHz – 11GHz	2GHz – 11GHz for fixed; 2GHz-6GHz for mobile applications
Application	Fixed LOS	Fixed NLOS	Fixed and mobile NLOS
MAC architecture	Point-to-multipoint, mesh	Point-to-multipoint, mesh	Point-to-multipoint, mesh
Transmission scheme	Single carrier only	Single carrier, 256 OFDM or 2048 OFDMA	Single carrier, 256 OFDM or scalable OFDM with 128,512,1024,2048 subcarriers
Modulation	QPSK, 16QAM, 64 QAM	QPSK, 16QAM, 64QAM	QPSK, 16QAM, 64QAM
Gross data rate	32Mbps – 134.4Mbps	1Mbps – 75Mbps	1Mbps – 75Mbps
Multiplexing	Burst TDM/TDMA	Burst TDM/TDMA/OFDMA	Burst TDM/TDMA/OFDMA
Duplexing	TDD & FDD	TDD & FDD	TDD & FDD
Channel BW	20MHz, 25MHz, 28MHz	1.75MHz, 3.5MHz, 7MHz, 14Mhz, 1.25MHz, 5MHz, 10MHz, 15MHz,	1.75MHz, 3.5MHz, 7MHz, 14Mhz, 1.25MHz, 5MHz, 10MHz, 15MHz,

		8.75Mhz	8.75Mhz
Air-interface designation	WirelessMAN-SC	WirelessMAN-SC WirelessMAN-OFDM WirelessMAN-OFDMA WirelessHUMAN	WirelessMAN-SC WirelessMAN-OFDM WirelessMAN-OFDMA WirelessHUMAN
WiMAX implementation	None	256-OFDM as Fixed WiMAX	Scalable OFDMA as Mobile WiMAX

CHAPTER 3 METHODOLOGY

The flow chart below shows the steps taken to complete the WiMAX testbed.

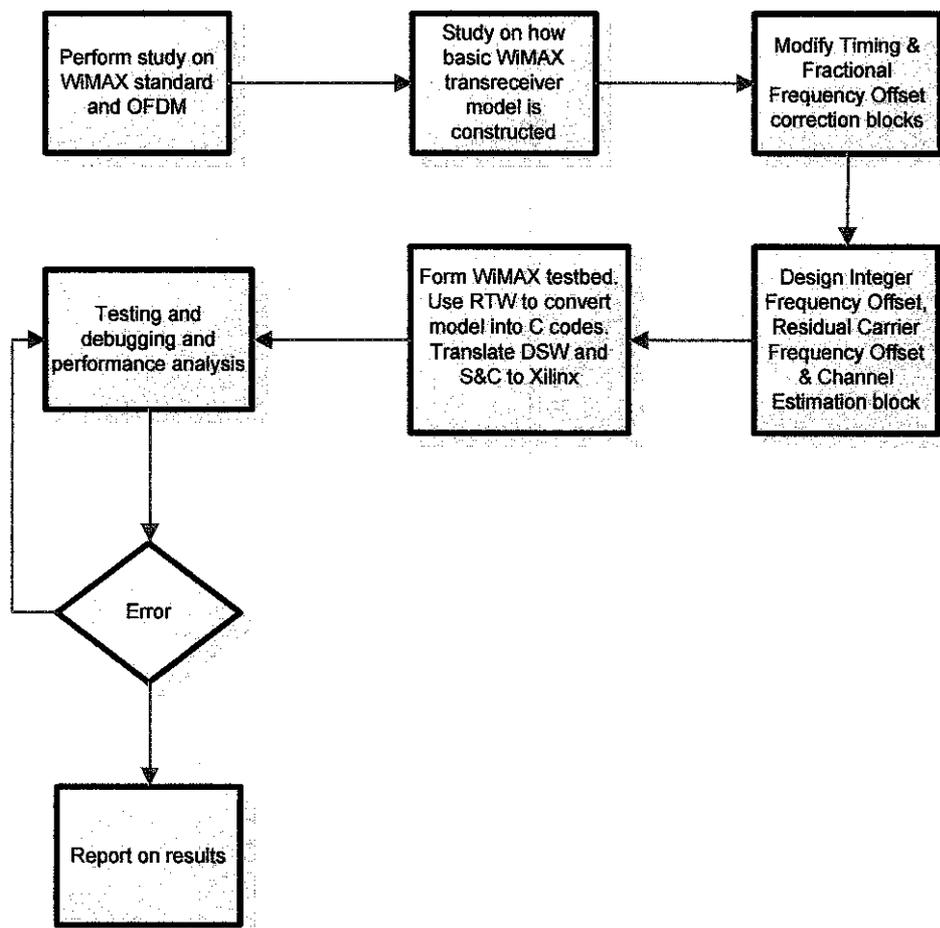


Figure 12 Process flow chart for FYP

Under this chapter, it will be divided into two main sections. Firstly, it will discuss about how the testbed is designed and secondly, it will mention about the steps taken for performance evaluation of the testbed.

3.1 Implementation of WiMAX testbed

Theoretically a WiMAX transceiver must have the following blocks as shown in Figure 13.

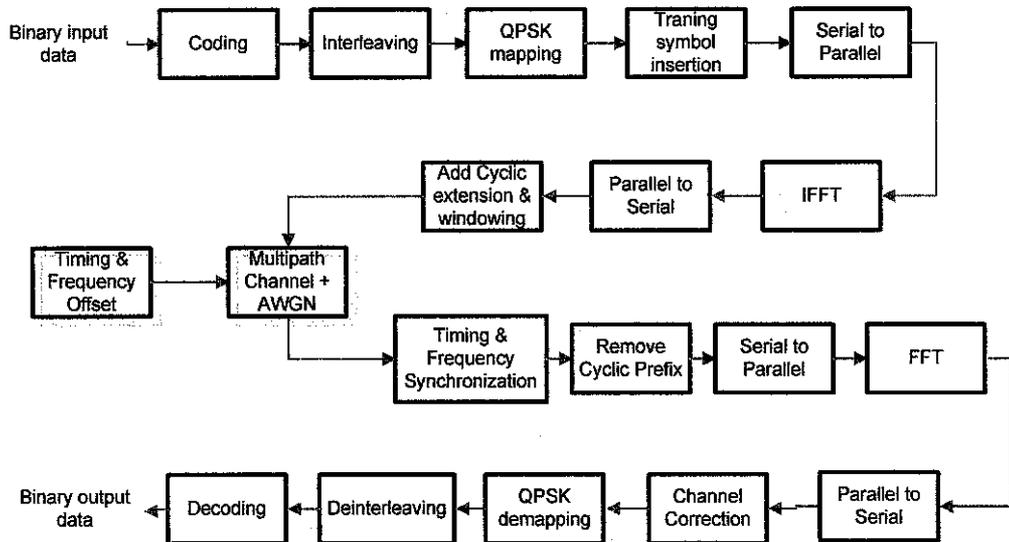


Figure 13 WiMAX transceiver structure

For this FYP, all the blocks shown in the figure above will be implemented via Simulink blocks in order to simulate a working testbed. Then, using RTW, the Simulink model will be converted into C codes and run using the command prompt. Finally, the blocks performing the Double Sliding Window and Schmidl & Cox algorithm will be converted into Xilinx blocks to show that hardware co-simulation is possible.

The table below list out all the parameters for the WiMAX testbed based on the IEEE802.16 standard. This testbed is based on the WirelessMAN-OFDM which uses OFDM modulation and is designed for NLOS operation at frequency bands below 11GHz. The following table indicates the parameters for the testbed design.

Table 3 WiMAX physical layer parameters for testbed

Parameters	Value
Carrier Frequency	Below 11GHz licensed bands
NFFT : Number of FFT/IFFT points	256
Nused : Number of used sub carriers	200
BW : Channel Bandwidth	14 MHz
n : Sampling factor	8/7 for channel bandwidths that are a multiple of 1.75Mhz
G : Cyclic prefix length	1/4 of FFT points
Number of lower frequency guard carriers	28
Number of higher frequency guard carriers	27
Frequency offset indices of guard carriers	-128,-127,...,-101 +101,+102,...,+127
Frequency offset indices of pilot carriers	-88, -63, -38, 13, 38, 63, 88
F _s : Sampling Frequency	Floor(n.BW/8000)*8000 = 16 Mhz
Δf: Subcarrier spacing	F _s /NFFT = 62500 Hz
T _b : Useful symbol time	1/Δf = 16μs
T _g : CP Time	G.T _b = 4μs
T _s : OFDM Symbol Time	T _s = T _b + T _g
Sampling time	T _b /NFFT = 6.25*10 ⁻⁸ s

The tools necessary to implement this FYP are:

- MATLAB
- Simulink
- Real-Time Workshop
- Xilinx System Generator

Installation version for Xilinx System Generator must match the version of MATLAB running on the computer. For MATLAB 2006a, Sysgen 8.1.01 need to be used.

3.1.1 WiMAX testbed in Simulink

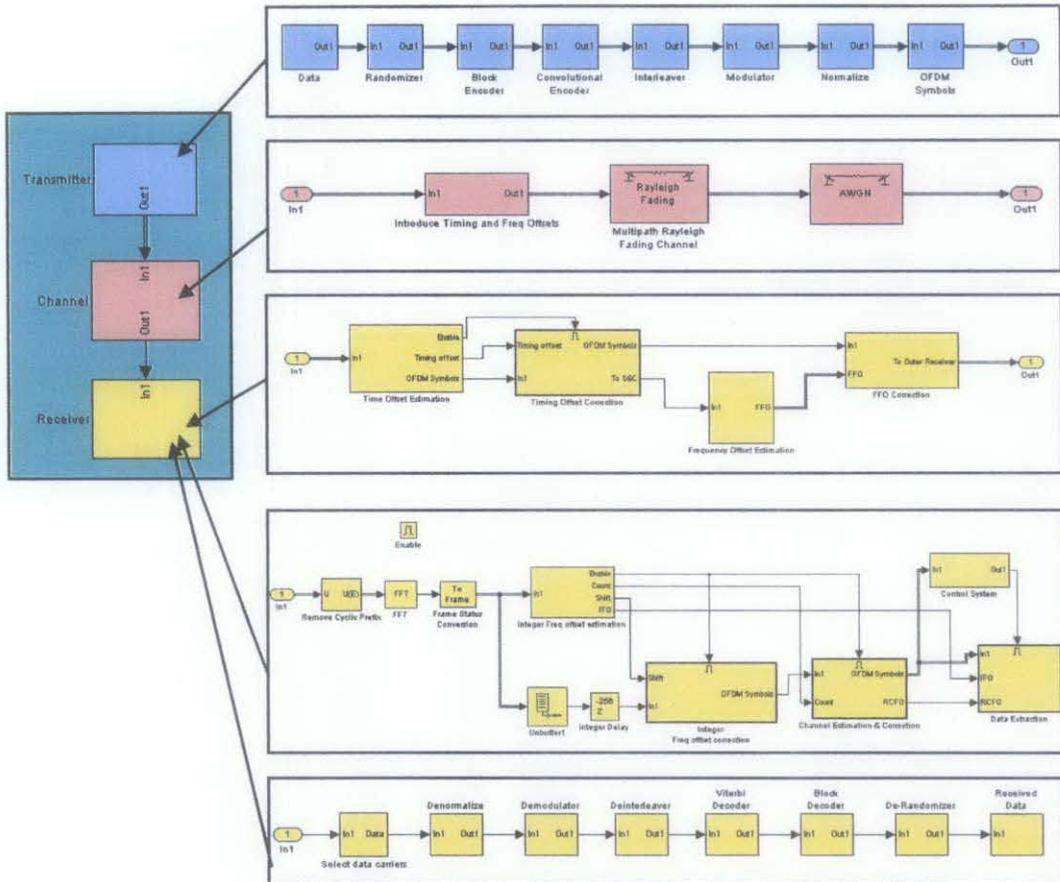


Figure 14 WiMAX testbed structure

Figure 14 indicates the overall WiMAX testbed design in Simulink. The WiMAX testbed is divided into the transmitter, channel and receiver. Explanation about each section will be done in detail. As shown the testbed will implement all the blocks necessary for the WiMAX transceiver.

The basic blocks implementing an executable specification for the WiMAX standard (that perform randomizing, block encoding, convolutional encoding, interleaving, modulating, normalizing and creating OFDM symbols at the transmitter and de-randomizing, block decoding, viterbi decoder, deinterleaver, demodulator and denormalizing at the receiver) are obtained from [14]. They have been designed to follow the specifications in [4]. I have modified some of the blocks to suit this testbed

such as using a random data source via the Bernoulli Binary Generator and adding the Training Symbol to the front of the frame.

The Double Sliding Window (for Timing Offset Estimation) and Schmidl & Cox (for FFO Estimation) blocks are obtained from [2]. In this FYP, the blocks that have been designed are:

- Timing Offset Correction
- FFO Correction
- IFO Estimation & Correction
- RCFO Estimation & Correction
- Channel Estimation & Correction

Furthermore, this FYP involves combining all these blocks using special control blocks to form a working testbed.

3.1.1.1 WiMAX transmitter

At the transmitter, random data is sent using a Bernoulli Random Data Generator. The data then passes through a randomizer, block encoder, convolutional encoder, block interleaver before being modulated using QPSK. The OFDM symbols block will send the data on the appropriate subcarriers and add pilots to each symbol. A preamble (training symbol) is added in front of the frame.

3.1.1.1.1 Data Source

The source of data is random from a Bernoulli Random Data Generator. The sampling time is in accordance to the WiMAX standard and shown in Table 3.

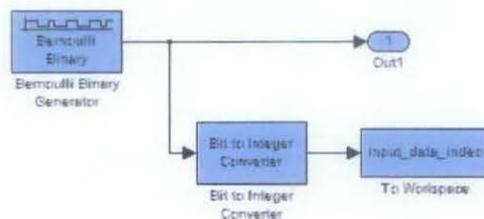


Figure 15 Data block in Simulink

3.1.1.1.2 Randomizer

Based on the IEEE 802.16 standard, channel coding is composed of three steps: randomizer, FEC and interleaving. They shall be applied in this order at transmission. The complementary operations shall be applied in reverse order at reception [4].

Data randomization is performed on each burst of data on the downlink and uplink. For Reed-Solomon encoded, data, padding will be added to the end of the transmission block, up to the allocated amount, minus one byte which is reserved for the 0x00 tail byte by the FEC.

The PRBS generator is $1+X^{14}+X^{15}$. Each byte is entered sequentially into the randomizer, MSB first. Preambles are not randomized.

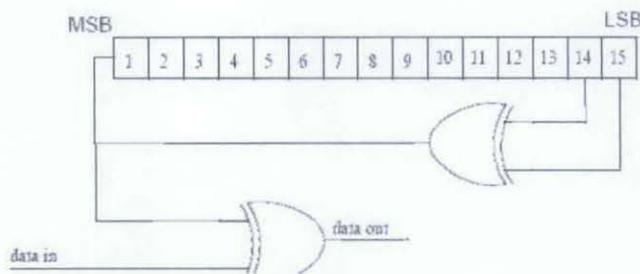


Figure 16 Randomizer in logic diagram

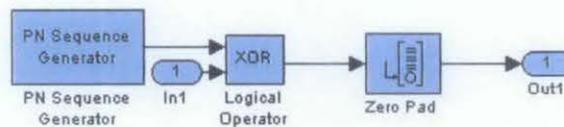


Figure 17 Randomizer in Simulink

3.1.1.1.3 Forward Error Correction

An FEC, consisting of the concatenation of a Reed-Solomon outer code and a rate-compatible convolutional inner code, shall be supported on both uplink and downlink. The encoding is performed by first passing the data in block format through the RS encoder and then passing it through a zero-terminating convolutional encoder [4].

The Reed-Solomon encoding shall be derived from a systematic RS (N=255, K=239, T=8) code using GF(2⁸)

where

N = number of overall bytes after encoding

K = number of data bytes before encoding

T = number of data bytes which can be corrected.

The following polynomials are used for the systematic code:

Code Generator Polynomial: $g(x) = (x+\lambda^0)(x+\lambda^1)\dots(x+\lambda^{2T-1})$, $\lambda=02_{\text{HEX}}$

Generator Polynomial: $p(x) = x^8 + x^4 + x^3 + x^2 + 1$

This code is shortened and punctured to enable variable block sizes and variable error-correction capability. When a block is shortened to K' data bytes, add 239-K' zero bytes as a prefix. After encoding discard these 239-K' zero bytes. When a codeword is punctured to permit T' bytes to be corrected, only the first 2T of the total 16 parity bytes shall be employed [4]. Each RS block is encoded by the binary convolutional encoder, which shall have native rate of 1/2, a constraint length of 7 and shall use the generator polynomial codes G₁ = 171_{OCT} for X and G₂ = 133_{OCT} for Y to derive its two code bits.

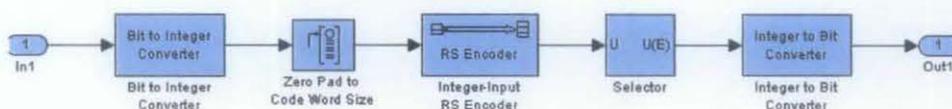


Figure 18 Block Encoder in Simulink

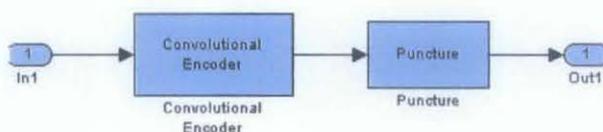


Figure 19 Convolutional Encoder in Simulink

3.1.1.1.4 Interleaver

All encoded data bits shall be interleaved by a block interleaver with a block size corresponding to the number of coded bits per the allocated subchannels per OFDM symbol, N_{cbps} . The interleaver is defined by a two step permutation.

$N_{cpc} = 2$ for QPSK

$s = \text{ceil}(N_{cpc}/2)$

k = index of the coded bit before first permutation.

m_k = index of the coded bit after the first and before the second permutation

j_k = index after the second permutation just prior to modulation mapping.

The first permutation is defined by

$$m_k = (N_{cbps}/12) \cdot k_{\text{mod}12} + \text{floor}(k/12) \quad k = 0, 1, \dots, N_{cbps}-1 \quad (3-1)$$

The second permutation is defined by the equation

$$j_k = s \cdot \text{floor}(m_k/s) + (m_k + N_{cbps} - \text{floor}(12 \cdot m_k / N_{cbps}))_{\text{mod}(s)} \quad k = 0, 1, \dots, N_{cbps}-1 \quad (3-2)$$

The de-interleaver, which performs the inverse operation, is also defined by two permutations.

j = index of the coded bit before first permutation.

m_j = index of the coded bit after the first and before the second permutation

k_j = index after the second permutation just prior to modulation mapping.

The first permutation is defined by the equation:

$$m_j = s \cdot \text{floor}(j/s) + (j + \text{floor}(12 \cdot j / N_{cbps}))_{\text{mod}(s)} \quad k = 0, 1, \dots, N_{cbps}-1 \quad (3-3)$$

The second permutation is defined by the equation:

$$k_j = 12 \cdot m_j - (N_{cbps}-1) \cdot \text{floor}(12 \cdot m_j / N_{cbps}) \quad j = 0, 1, \dots, N_{cbps}-1 \quad (3-4)$$

The first permutation in the de-interleaver is the inverse of the second permutation in the interleaver and conversely [4].

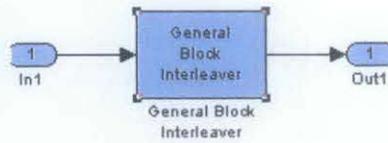


Figure 20 Interleaver in Simulink

3.1.1.1.5 Training symbol

The design of the training-symbol is base on the work of Schmidl and Cox [7] on frequency and time synchronization for OFDM. The first of the two training symbols (TS) used in [7] is a unique OFDM symbol because every alternate subcarrier (SC) is zero. This implies that the symbol has identical halves in the time-domain, due to properties of the Fourier transform.

In the original scheme [7], the odd-numbered subcarriers are zero, while the even subcarriers contain a known, binary pseudo noise (PN)-sequence. Many of the even-numbered sub-carriers must be zero, because they are used to separate sub-bands and to avoid problems with DC offsets and carrier feed-through. In order to apply the synchronization scheme to this OFDM system, odd-numbered sub-carriers contain the PN-sequence. This is a novel modification of the well-known technique. Nevertheless, the corresponding time-domain symbol has identical halves, but the samples of its second half have opposite signs, which are easily taken into consideration in the estimation steps [7].

In [7], two OFDM symbols comprise the training sequence. Known data is modulated there differentially between the two OFDM symbols, for the purpose of integer frequency- synchronization. The modification used here therefore reduces the overhead introduced by one OFDM symbol per frame [12].

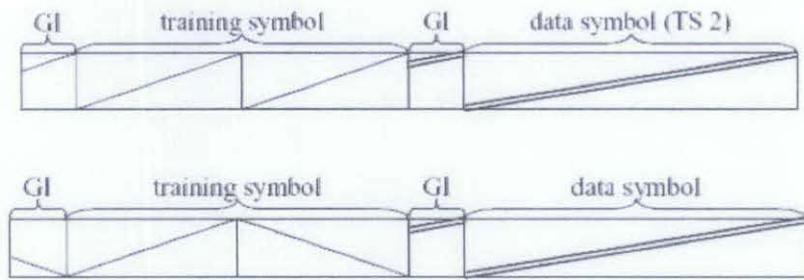


Figure 21 Structure of the training symbol. Top: For original Schmidl and Cox scheme Bottom: New proposed method

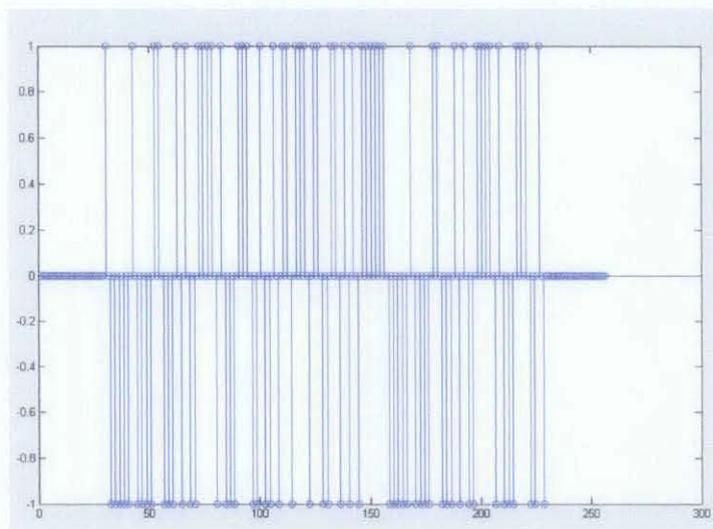


Figure 22 Stem diagram of the Training Symbol

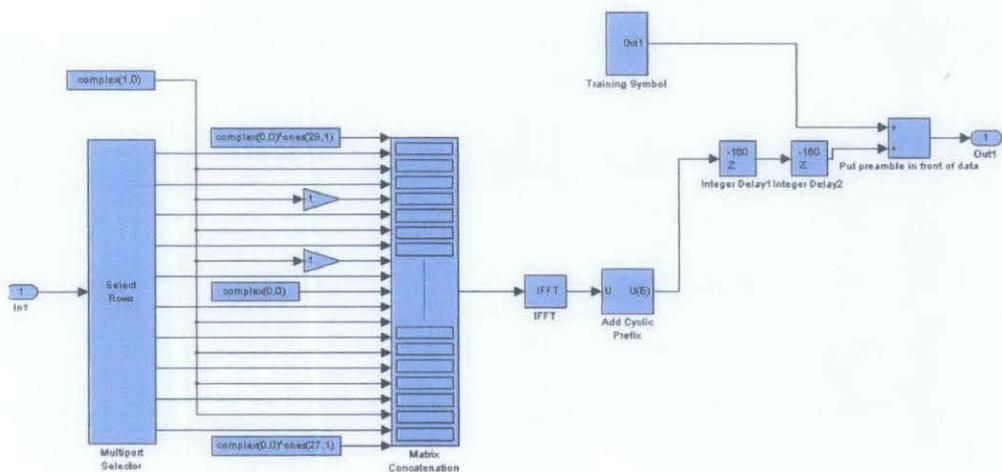


Figure 23 Block creating OFDM symbols in Simulink. The training symbol is added in front of the data

3.1.1.2 Channel

At the channel, timing and frequency offset are deliberately introduced into the simulation model. The OFDM symbols then pass through the Multipath Rayleigh and AWGN channel. SUI channel models are an extension of the earlier work by AT&T Wireless and Erceg et al. [13]. The key parameters for SUI channels are stated in the table below. In this testbed, SUI-3 channel model is used.

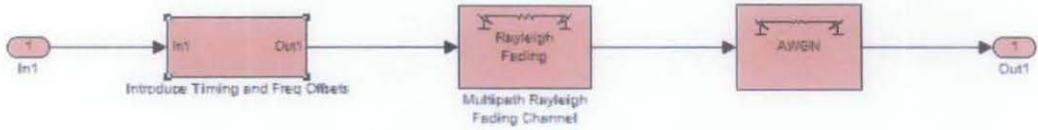


Figure 24 Channel Block in Simulink

Table 4 Parameters for SUI channel models

Terrain type	SUI Channel
C (Mostly flat terrain with light tree densities)	SUI-1, SUI-2
B (Hilly terrain with light tree density or flat terrain with moderate to heavy tree density)	SUI-3, SUI-4
A (Hilly terrain with moderate to heavy tree density)	SUI-5, SUI-6

Table 5 SUI-3 Parameters

Channel Model	Tap 1	Tap 2	Tap 3	RMS delay spread
Delay	0 μ s	0.4 μ s	0.9 μ s	0.264 μ s
Power	0dB	-5dB	-10dB	-

3.1.1.3 WiMAX Receiver

At the receiver, synchronization is achieved using the following steps:

- Timing Synchronization using Double Sliding Window Technique
- Fractional Frequency Offset Synchronization using Schmidl & Cox method
- Application of FFT
- Integer Frequency Synchronization using local copy of Training Symbol
- Residual Carrier Frequency Offset Synchronization based on rotation of pilots
- Linear Channel Estimation & Correction using Training Symbol

After synchronization has been performed, the data streams is passed through the de-randomizer, block decoder, viterbi decoder, deinterleaver and finally demodulated based on QPSK to recover the transmitted data.

3.1.1.3.1 Frame timing synchronization

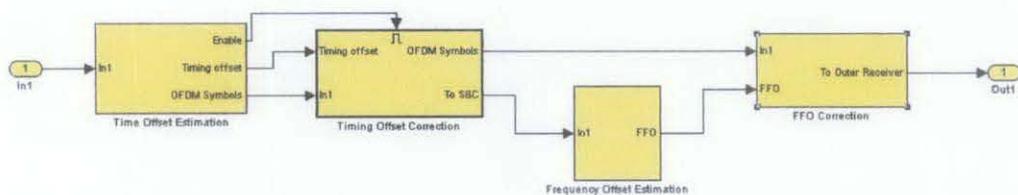


Figure 25 Inner Receiver Design in Simulink

The incoming OFDM symbols will first pass through the Inner Receiver which contains the timing offset estimation and correction as well as the fractional frequency offset estimation and correction. The OFDM symbols will then be sent to the Outer Receiver

Two methods can be adopted to achieve frame timing synchronization in the inner receiver. The first is Schmidl and Cox technique [7]. In fact, this method is the common method in most conventional OFDM receivers. The second method is double sliding window packet detection. This is the method used in this FYP.

The underlying principle of the double sliding window packet detection algorithm is to form the decision variable m_n as a ration of the total energy contained inside the two windows [18]. Therefore, the receiver does not need to know the absolute value of the received energy.

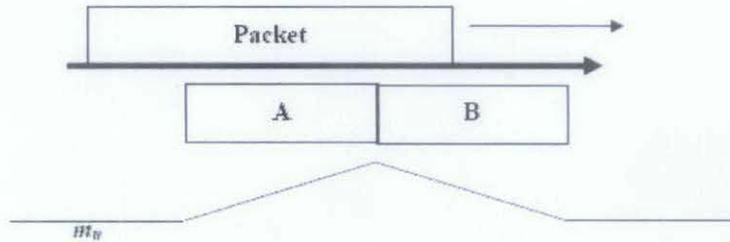


Figure 26 Double Sliding Window Packet Detection

Let the energy inside A and B be

$$\begin{aligned}
 a_n &= \sum_{m=0}^{M-1} r_{n-m} r_{n-m}^* = \sum_{m=0}^{M-1} |r_{n-m}|^2 \\
 b_n &= \sum_{l=0}^{L-1} r_{n-l} r_{n-l}^* = \sum_{l=0}^{L-1} |r_{n-l}|^2
 \end{aligned}
 \tag{3-5}$$

M and L are the window sizes of a_n and b_n respectively and are usually made the same. With this, the decision variable is given by

$$m_n = \frac{a_n}{b_n}
 \tag{3-6}$$

The occurrence of peak $E[m_n]$ will give rise to where the packet has started (i.e. a_n would consist of signal and noise and b_n would consist purely of noise) [18].

$$E[m_n]_{\text{peak}} = \text{SNR} + 1$$

Thus, locating the peak of m_n would provide information regarding the position of the start for the packet. Figure 27 shows the implementation in Simulink. It also incorporates the circuitry to prevent divide by 0 problems.

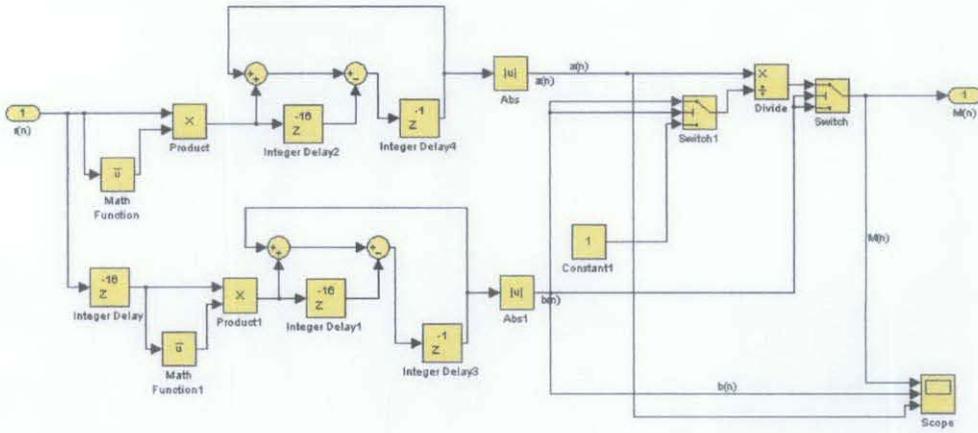


Figure 27 Implementation of Double Sliding Window in Simulink

3.1.1.3.2 Fractional Frequency Synchronization

The Schmidl and Cox technique incorporates a sliding window correlator in addition to an energy detector used to normalize the decision statistic and hence guard against fluctuations of the received signal power [7]. A sliding window P computes the cross-correlation between the received signal and a version of the delayed (by one short preamble) received signal. Another sliding window R is used to compute the received signal energy within the cross-correlation interval.

Let there be $L-1$ ($L = N/2$) complex samples in one-half of the first training symbol (excluding the cyclic prefix), and let the sum of the pairs of products be

$$P(d) = \sum_{i=0}^{L-1} (r_{d+i} * r_{d+i+L}) \quad (3-7)$$

which can be implemented with the iterative formula

$$P(d) = P(d-1) + r_{d+L-1} * r_{d+2L-1} - r_{d-1} * r_{d+L-1} \quad (3-8)$$

Note that d is a time index corresponding to the first sample in a window of $2L$ samples. This window slides along in time as the receiver searches for the first training symbol. The received energy for the second half-symbol is defined by

$$P(d) = \sum_{i=0}^{L-1} |r_{d+i+L}|^2 \quad (3-9)$$

which also can be implemented iteratively as

$$P(d) = R(d-1) + |r_{d+2L-1}|^2 - |r_{d+L-1}|^2 \quad (3-10)$$

With this, the decision statistic, $M(d)$, is given by:

$$M(d) = \frac{|P(d)|^2}{(R(d))^2} \quad (3-11)$$

The decision statistic reaches a plateau which has a length equal to the length of the guard interval minus the length of the channel impulse response. The start of the frame can be taken to be anywhere within this window without a loss in the received SNR but the optimum timing, d_{opt} , is obtained at the end of the plateau [7].

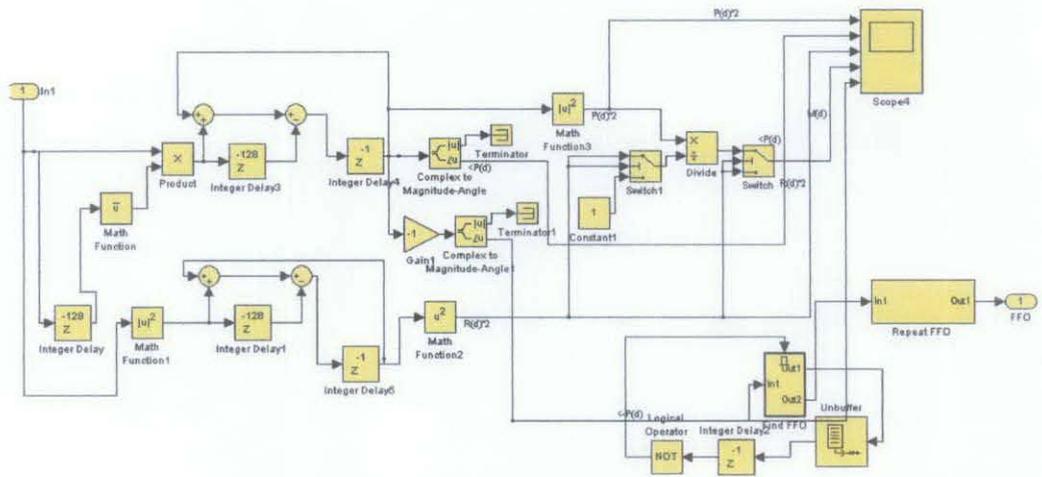


Figure 28 Implementation of Schmidl & Cox technique in Simulink

Figure 28 shows how the above equations are realized in Simulink. $P(d)$ and $R(d)$ are implemented in a hardware efficient manner. With this, the computations require just one complex multiplication and two additions/ subtractions per input sample respectively.

The algorithm for frequency-synchronization is based on the fact that the phase-angle of the above-defined correlation sum at its optimum position, $P(d_{opt})$, is

proportional to the carrier frequency-offset, as seen from the following derivation. The received signal with frequency offset can be modelled as

$$r_i = s_i e^{j(2\pi\delta' i / N + \theta)} + n_i \quad (3-12)$$

where $\delta f'$ is the frequency-offset normalized to the subcarrier spacing F , and θ is the carrier phase offset. Considering that during the TS, $s_i = -s_{i+L}$, we obtain in the absence of noise

$$P(d_{opt}) = -\sum_{i=0}^{L-1} |s_{i+d_{opt}}|^2 e^{j(2\pi\delta' L / N)} \quad (3-13)$$

The estimate of the frequency offset is obtained from the phase-angle of this expression

$$\hat{\delta f'} = \frac{1}{\pi} \angle [-P(d_{opt})] \quad (3-14)$$

After obtaining the frequency offset estimates, correction is written as

$$\hat{r}_{d_{opt+i}} = r_{d_{opt+i}} e^{-j2\pi\hat{\delta f'} i / N}$$

where $i = \{0, 1, \dots, N-1\}$ (3-15)

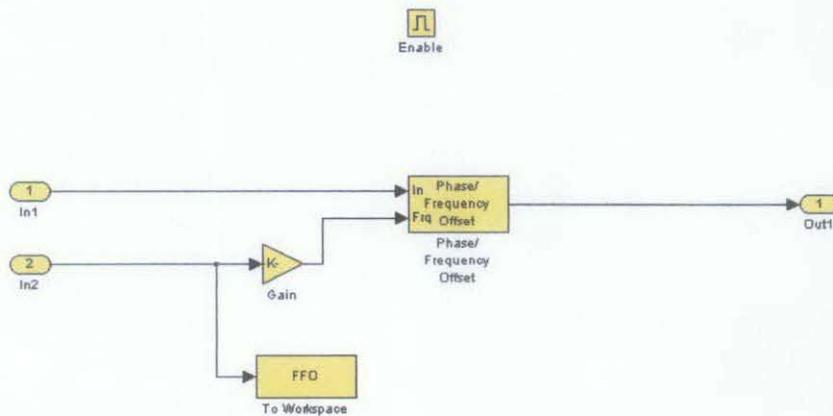


Figure 29 FFO Correction Implementation in Simulink

3.1.1.3.3 Integer Frequency Synchronization

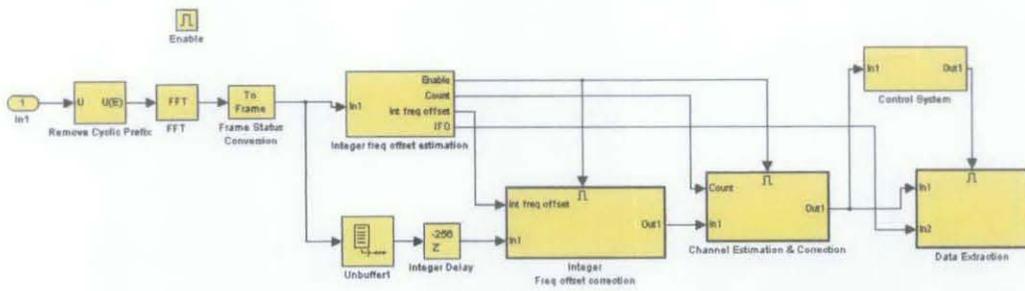


Figure 30 Outer Receiver Design in Simulink

The outer receiver consists of FFT, Integer Frequency Offset Estimation & Correction, Channel Offset as well as the Data Extraction blocksets. Input comes from the Inner receiver where timing and fractional frequency offset has been corrected. The cyclic prefix is then removed and the Fast Fourier Transform is performed by the FFT block. Next comes the Integer Frequency Offset Estimation block where the integer frequency offset will be detected and corrected.

The integer frequency-synchronization is based on the known data carried in the TS [6]. The even-indexed SCs are removed from this sequence because they don't carry data and correlation between the incoming signal and a local copy of the Training Symbol is performed. An autocorrelation of the preamble is performed and a peak occurs when the incoming symbol passes exactly on top the local copy.

The integer frequency-offset is found by searching for the magnitude-maximum of the metric and calculating the shift from the center point. This shift indicates the integer frequency offset. This is defined in the function below

$$\max_m y_{gg} = \sum_n g(n)g_l^*(n+m) = \begin{cases} c & \dots m = l \\ 0 & \dots m \neq l \end{cases} \quad (3-16)$$

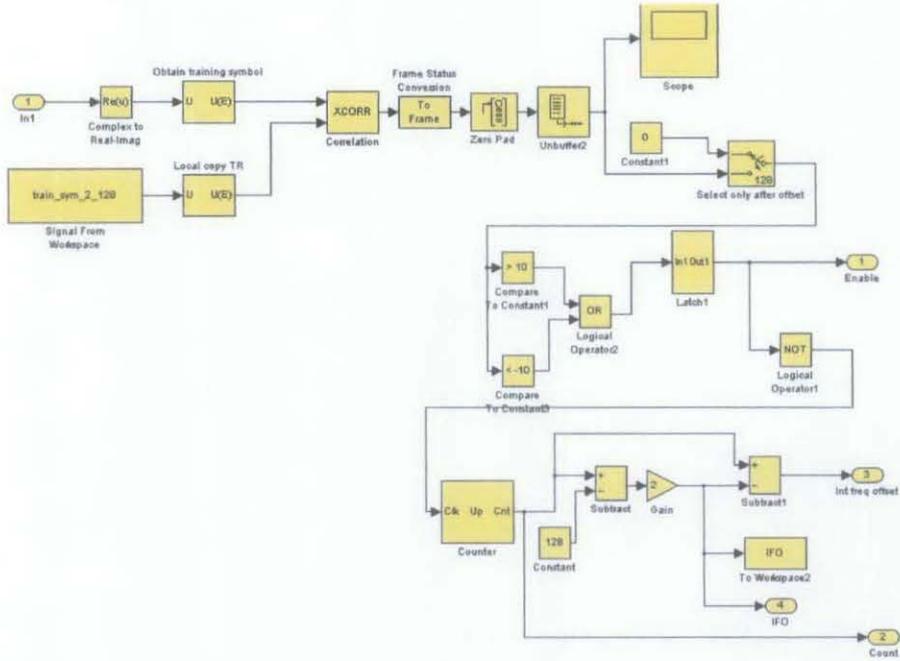


Figure 31 IFO Estimation implementation in Simulink

3.1.1.3.4 Residual Carrier Frequency Offset Synchronization

The Schmid & Cox method will only give an estimate of the FFO. Any remaining offset will cause a progressive phase rotation of signal constellation points. Many methods can be implemented to perform RCFO correction and the one that has been chosen for this FYP is the L^{th} extension method [8].

This method estimates the RCFO by using the current symbol and the subsequent L^{th} symbol which in this testbed is 8. The difference in phase of the reference carriers (pilots) is given by

$$\begin{aligned}
 \phi_L &= \theta_{l+L}(k') - \theta_l(k') \\
 &= 2\pi\eta L N_s / N + \theta_l^H(k') - \theta_l^W(k')
 \end{aligned} \tag{3-17}$$

where a slow fading channel has $\theta_l^H(k') \approx \theta_{l+L}^H(k')$

Each OFDM symbol consists of $L+1$ ($L+1 < N$) data samples. Data samples are $X_{l,k}$ where l ($0 \leq l < \infty$) and k ($-L/2 \leq k \leq L/2$) stand for the indexes of an OFDM symbol and sub-carrier respectively. An OFDM symbol has $N_s = N + N_g$ where N_g is the number of

samples in the Guard Interval.

The RCFO can be obtained by using (3-17) and expressed as

$$\Delta \hat{f} = \frac{\sum_{g=1}^p \phi_L(m_g)}{2\pi T_p L N_s} \quad (3-18)$$

By plotting the difference in reference carriers, we can obtain the RCFO from the slope and using (3-18).

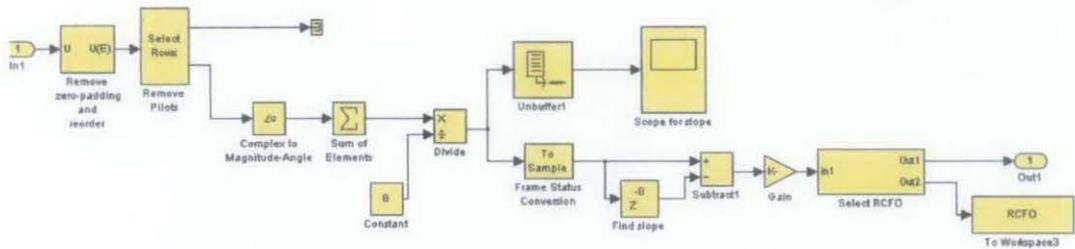


Figure 32 RCFO Estimation Implementation in Simulink

3.1.1.3.5 Channel estimation

A linear equalizer simply runs the received signal through a filter that roughly models the inverse of the channel. If the transmitted signal in the frequency domain is denoted by $X(k)$ and received signal $Y(k)$ with k for subcarrier index [1]. Their relation can be expressed as

$$Y(k) = H(k)X(k) + W(k), \text{ for } k = 0, 1, \dots, N-1 \quad (3-19)$$

where $H(k)$ and $W(k)$ denote channel frequency response and additive white Gaussian Noise at the k -th subcarrier.

Thus the data symbols can be recovered by a one-tap equalizer. Channel gain is defined by

$$H_k = \frac{Y_k}{X_k} \quad (3-20)$$

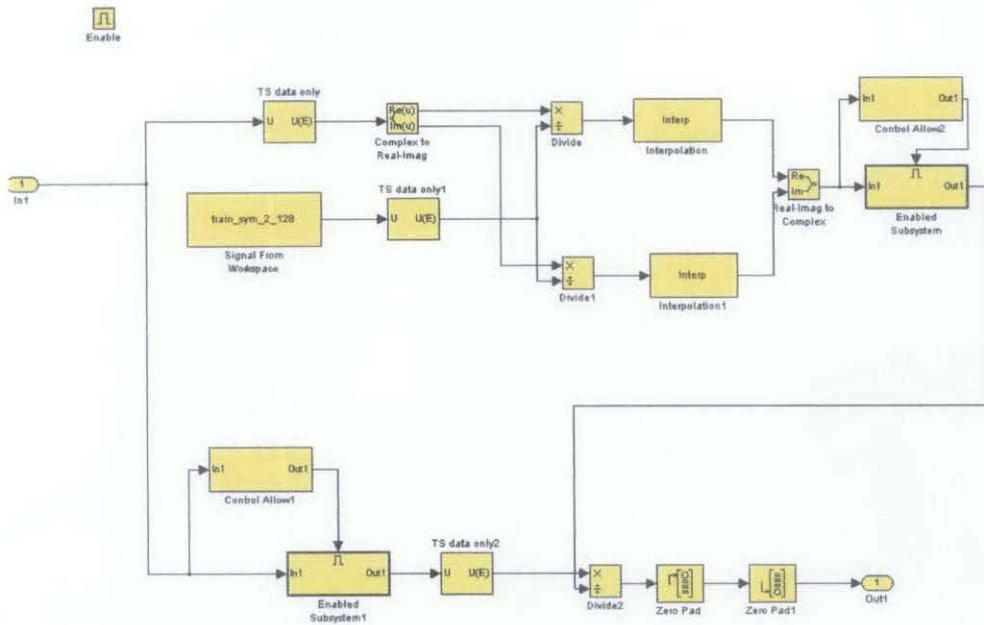


Figure 33 Channel Estimation implementation in Simulink

For Channel Estimation, we isolate the Training Symbol and use a local copy to perform division with it. That way, the channel gain for each sub-carrier will be known. Linear interpolation is performed to get the estimates for even sub carriers since they carry no data. This estimate will be used to correct all incoming data symbols in the frame. However, this technique assumes that the channel remains constant for one frame. In other words, the coherence time must be longer than one frame.

3.1.2 WiMAX testbed in C code

Using Real-Time Workshop, the Simulink model can be converted into C codes. This is one of the methods for implementing the system in hardware. Another mainstream method is to convert the Simulink blocks into Xilinx blocks and run them on the FPGA [19].

Real-Time Workshop generates optimized, portable, and customizable ANSI C or C++ code from Simulink models that operate in real-time and non-real-time. Generated code can run on PC hardware, DSPs, microcontrollers on bare-board environments, and with commercial or proprietary real-time operating systems (RTOS). Real-Time Workshop speeds up simulations, build in intellectual property

protection, and operate across a wide variety of real-time rapid prototyping targets.

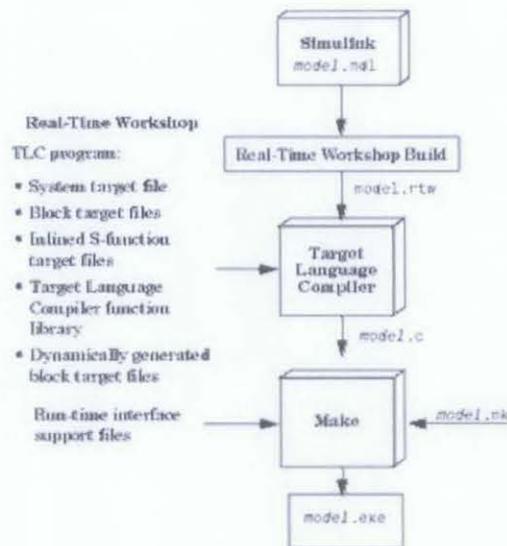


Figure 34 Process of generating source code from Simulink models using Real-Time Workshop

After creating a model, the Real-Time Workshop rapid simulation (RSim) target can be used to characterize the model's behavior [19]. The RSim target is able to generate an executable that runs fast, standalone simulations. You can repeat simulations with varying data sets, interactively or programmatically with scripts, without rebuilding the model. This can accelerate the characterization and tuning of model behavior and code generation testing.

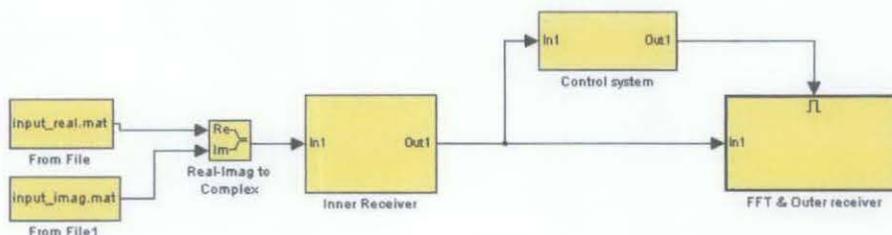


Figure 35 WiMAX testbed model for C code generation

In this FYP, only the WiMAX receiver shall be converted into C codes. In order to simulate the process of data coming into the receiver, From File blocks are used. The input data that goes to the receiver are real and imaginary components of

actual OFDM symbols that have gone through all the necessary coding and modulation at the transmitter. They must be of MAT file format and will simulate a real-case scenario where input comes from the antenna and the Analog-To-Digital converter. Using the model shown in Figure 35, I select Simulation->Configuration Parameters->Real-Time Workshop. The rsim.tlc is selected for the system target file. Additional parameters are entered by following the tutorial stated in MATLAB help folders. Then the build button is pressed.

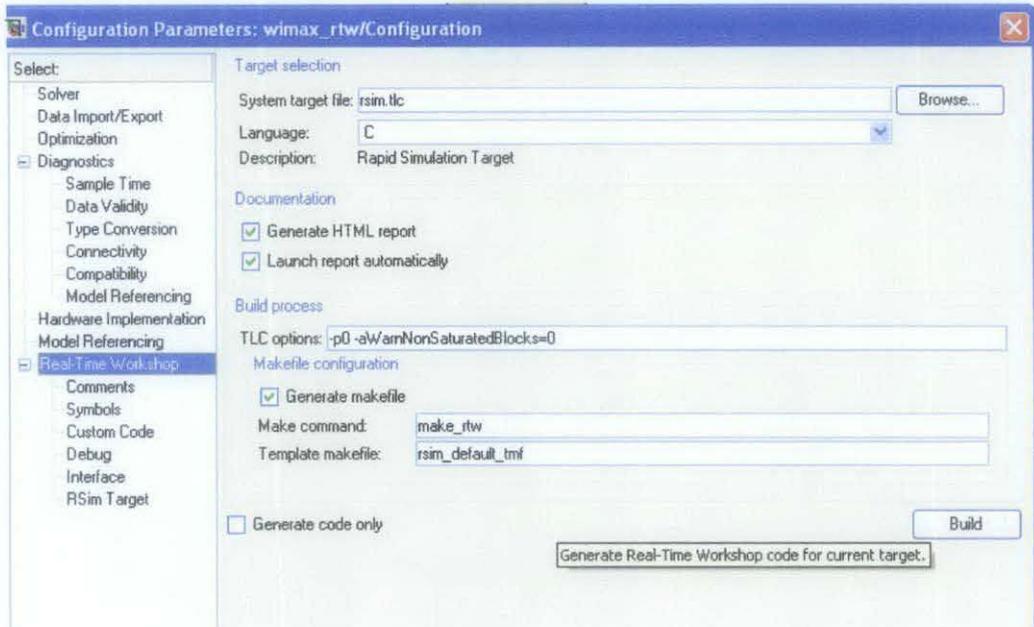


Figure 36 Configuration Parameters Window for RTW usage

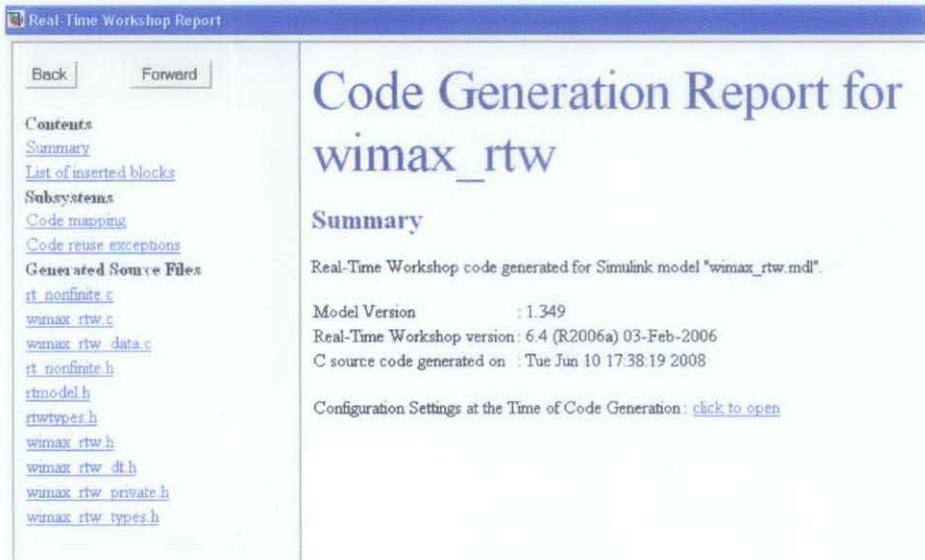


Figure 37 Real-Time Workshop Report

MATLAB will then convert the model in C codes and produce the report as shown in Figure 37. An executable file will be generated as well (In this FYP, the file name is `wimax_rtw.exe`).

3.1.3 Double Sliding Window and Schmidl & Cox using Xilinx blocks

As the Double Sliding Window and Schmidl & Cox blocks in Simulink have been designed and obtained from [2], this FYP takes it one step further by translating those blocks into Xilinx blocks.

In recent years, field-programmable gate arrays (FPGAs) have become key components in implementing high performance digital signal processing (DSP) systems, especially in the areas of digital communications, networking, video, and imaging. Coupled with a capability for implementing highly parallel arithmetic architectures, this makes the FPGA ideally suited for creating high-performance custom data path processors for tasks such as digital filtering, fast Fourier transforms, and forward error correction. For example, all major telecommunication providers have adopted FPGAs for high performance DSP out of necessity. A third-generation (3G) wireless base station typically contains FPGAs and ASICs in addition to microprocessors and digital signal processors (DSPs) [20].

System Generator is a tool that extends Simulink with software and blocks for bit and cycle-accurate modeling of digital logic and DSP functions. It also a translator that converts a Simulink model into hardware for Xilinx FPGAs [20]. Models constructed from Xilinx blocks behave in exactly the same way in Simulink as they do in hardware. In Simulink these models can readily be combined with non-Xilinx blocks to model parts of a system not bound to the FPGA.

In order to provide bit-accurate simulation of hardware, System Generator blocks operate on Boolean and arbitrary precision fixed-point values. The fundamental scalar signal type in Simulink is double precision floating point. The Gateway In and Gateway Out blocks form interfaces between data representation within System Generator and data types that can be examined and manipulated in the standard Simulink environment.

To translate a Simulink model into Xilinx blocks, the System Generator block is required, which can be found in the Xilinx blockset->Basic Elements library. This block controls hardware generation. Then, it is very much like in the Simulink environment where we drag and drop blocks from the library and create interconnections between blocks in the model.

It is important to note that since Xilinx blocks cannot handle complex numbers, the input must be split into real and imaginary components. Xilinx blocks are also very basic (such as addition and subtraction) and its library does not contain functions such as finding the absolute or conjugate values. Therefore, the Xilinx blocks have to be combined to bring about the same results as the Simulink blocks.

For example, in the DSW, we multiply the incoming signal with its complex conjugate. Representing in terms of formula

$$\begin{aligned} z_1 z_2 \\ &= (a + jb)(a - jb) \\ &= a^2 + b^2 \end{aligned}$$

This means the equation can be simplified to a^2+b^2 and implemented using Xilinx blocks. This is one of the examples of how the Simulink blocks can be translated into Xilinx blocks.

As for the conversion of real and imaginary values to magnitude and angle, the CORDIC ATAN block is used. It is important to note that the magnitude scale factor $K = 1.646760$ is not compensated in the processor and so the magnitude output should be scaled by this factor.

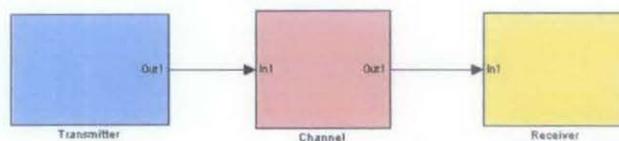


Figure 38 Model showing System Generator block

The switch to prevent division by zero is implemented in Xilinx by using an MCode block that uses a self-written code named xlswitch.

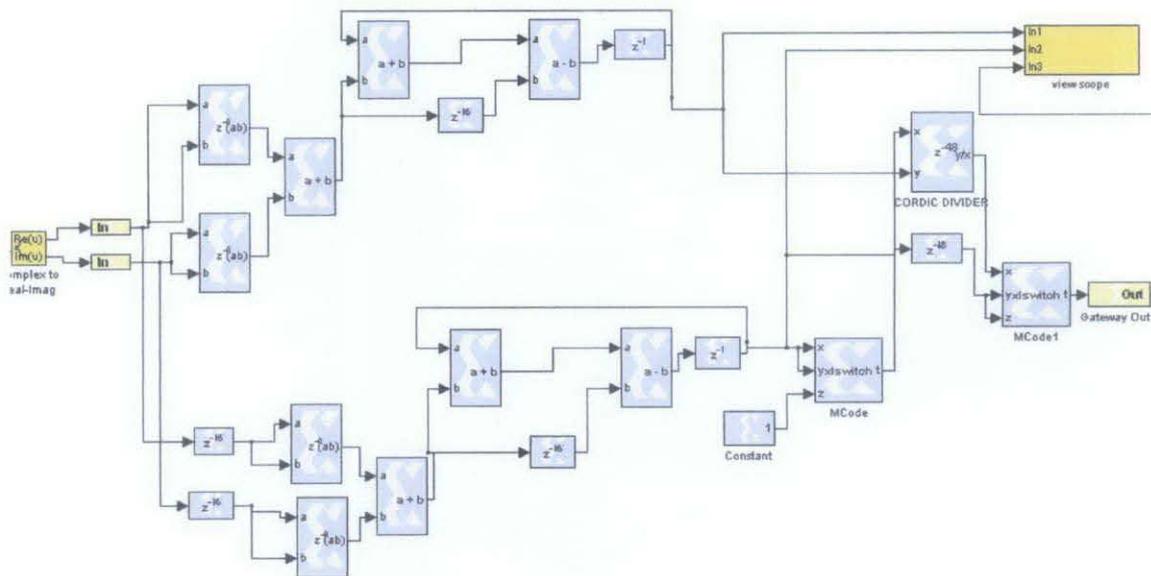


Figure 39 Double Sliding Window Implementation using Xilinx Blockset

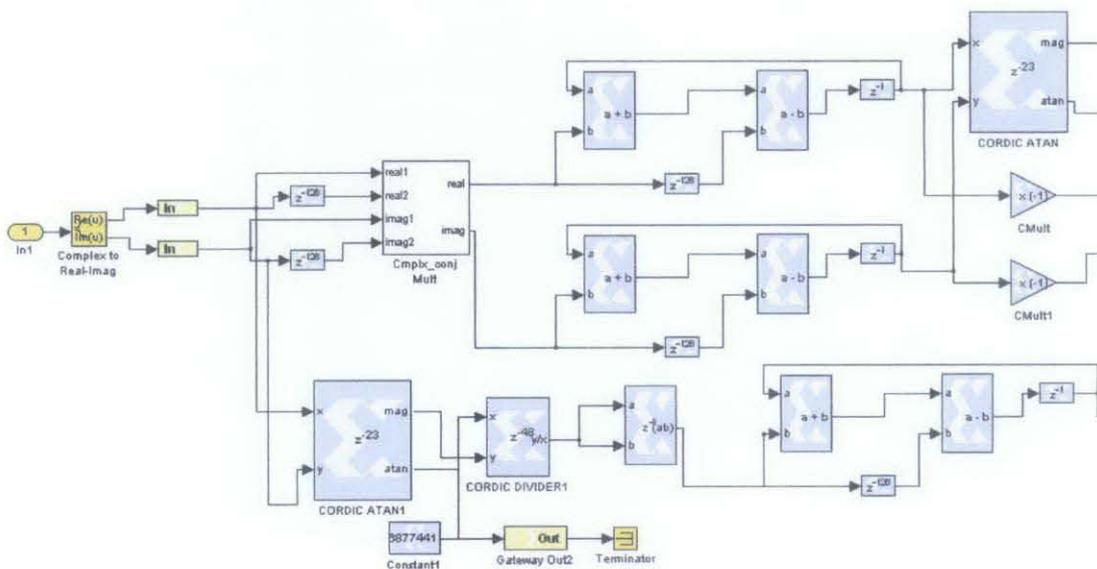


Figure 40 Schmidl & Cox Implementation using Xilinx Blockset (A)

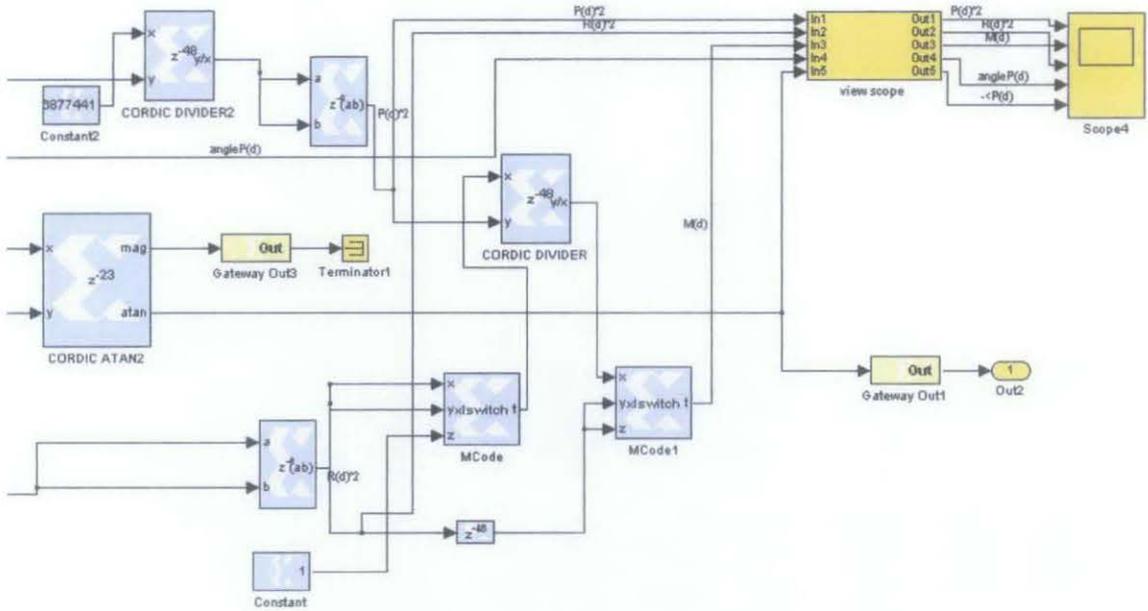


Figure 41 Schmidl & Cox Implementation using Xilinx Blockset (B)

3.2 Performance Analysis Approach of WiMAX Testbed

The performance analysis for the testbed shall be performed for:

- Frame timing synchronization
- FFO synchronization
- IFO synchronization
- RCFO synchronization
- Channel Estimation
- Overall performance of Receiver

Tests will also be performed on the following to prove its validity:

- WiMAX testbed in Simulink model
- WiMAX testbed in C code (generated from RTW)
- DSW and S&C in Xilinx blocks

Unless specifically mentioned, the SUI-3 channel model is used for Rayleigh multipath fading. 100 iterations are performed for each test case and the mean of the results are obtained.

3.2.1 Performance analysis for frame timing synchronization

The testbed is evaluated by looking at the probability of missing the peak for optimum timing vs SNR. For frame timing synchronization, the metric from the Double Sliding Window is obtained at SNR = 15dB and 30dB. A time-domain plot showing the frame before and after timing synchronization is captured through the scope in Simulink.

3.2.2 Performance analysis for fractional frequency offset synchronization

For FFO estimator performance, the SNR is varied from 0dB to 30dB and the MSE between the estimated FFO and the actual FFO is obtained for AWGN, Rayleigh at 10Hz maximum Doppler Frequency and 100Hz maximum Doppler Frequency. The $M(d)$ and corresponding $-P(d)$ plots are obtained for SNR = 15dB and at 30dB.

3.2.3 Performance analysis for integer frequency offset synchronization

For IFO estimation, the estimator's performance is also calculated by varying the SNR from 0dB to 30dB. The probability of missing the peak in the decision metric is counted for AWGN, Rayleigh at 10Hz Doppler Frequency and 100Hz Doppler Frequency. The decision metric plots are also generated for SNR = 15dB and 30dB.

3.2.4 Performance analysis for residual carrier frequency offset synchronization

For RCFO estimation, the estimator's performance is also calculated by varying the SNR from 0dB to 30dB. The MSE between the estimated RCFO and the actual RCFO is calculated for AWGN, Rayleigh at 10Hz Doppler Frequency and 100Hz Doppler Frequency. The MSE plots are generated for SNR = 15dB and 30dB.

3.2.5 Performance analysis for channel estimation

The SNR is varied from 0dB to 30dB and the MSE between the actual channel gain and the estimated channel gain is plotted. For the channel estimator, the real and imaginary plot for data at each subcarrier before Channel Correction and the Channel Estimate for each subcarrier that are in one frame is generated. Then, the real and imaginary plot for data at each subcarrier sent at transmitter and after Channel

Correction is also obtained. This is done for SNR = 15dB and 30dB.

3.2.6 Overall Performance Analysis for the Receiver

A BER vs SNR plot for SNR varied from 15dB to 30dB is shown. This step is performed with 50000 data points received and at least 100 iterations. The plot includes AWGN, Rayleigh at 10Hz and 100Hz Maximum Doppler Frequency.

3.2.7 Validity of WiMAX testbed in Simulink model

In order to ensure the model is working correctly, different values of timing and frequency offset shall be input into the system and the receiver must be able to correctly detect and synchronize the incoming data symbols. One of the iterations shall be performed with timing offset of 55 and frequency offset of 2.35 at 30dB SNR. Another iteration shall be shown with timing offset of 82 and frequency offset of 3.77 at 20dB SNR. The results of the estimation shall be shown with screenshots of the array editor, showing the values that have been written to the workspace. The Scatter plots for the signal constellation at the receiver are shown.

3.2.8 Validity of WiMAX testbed in C code

The C code shall run at the command prompt. Begin by typing: “cd MATLAB\work\FYP_KMH\rtw\rsim” (by default), bringing us to the folder where the wimax_rtw.exe is located. The executable file can be run with different input files to show that it can accept various input signals. To run, type “wimax_rtw.exe -f input_real.mat=input_reall.mat -f input_imag.mat=input_imagl.mat”. The -f function replaces the original input file with another specified file.

3.2.9 Validity of DSW and S&C in Xilinx blocks

As usual, different values of timing offset and frequency offset shall be used to prove that the DSW and S&C blocks in Xilinx produce the correct results. One of the iterations shall be performed with timing offset of 37 and frequency offset of 2.75 at 30dB SNR. Another iteration shall be shown with timing offset of 64 and frequency offset of 3.8 at 20dB SNR. These values are chosen at random.

CHAPTER 4

RESULTS AND DISCUSSION

Having elaborated the synchronization algorithms and how they are implemented, this chapter will present all the results from the simulations. Chapter 3 has mentioned the performance analysis approach that has been taken. For all time-domain plots, the y-axis represents the amplitude whereas the x-axis represents sample index. The results show that estimation of the timing, FFO, IFO and RCFO can be obtained even at low SNR. However, since the Channel Estimator is linear, it places a limit on the overall performance of the receiver. The receiver's performance closely resembles the theoretical performance only at SNR 18dB or higher. Test results also prove that the implementation of the testbed in Simulink, C code and Xilinx give the accurate and expected results.

4.1 Frame timing synchronization

Figure 42 shows the decision metric for the double sliding window packet detection at 30dB (high SNR). m_n will go up to a large value when the channel has low noise. When OFDM symbol slides into window A, the metric starts to rise and later will decline as it starts to fill up window B. By observation, it is easy to identify the start of the OFDM frame. In this case, it is at sample no. 72. The peak of the metric is chosen to find the optimum timing. Since the window for the DSW is 16, therefore, the start of the frame occurs 17 samples before the peak which is at sample no. 55. This matches the timing offset that has been set and thus estimation is correct.

Figure 43 shows that even at low SNR (15dB), there is no ambiguity of discerning the start of the frame. We can see that the peak occurs at the same sample no. as in Figure 43 only that the amplitude is lower. Double sliding window always produces a sharp peak even in cases where SNR is low. As proven in [2], it is a better algorithm compared to the S&C in determining timing offset. This is because the

S&C has a large ambiguity where any portion of the plateau can be used for determining timing offset.

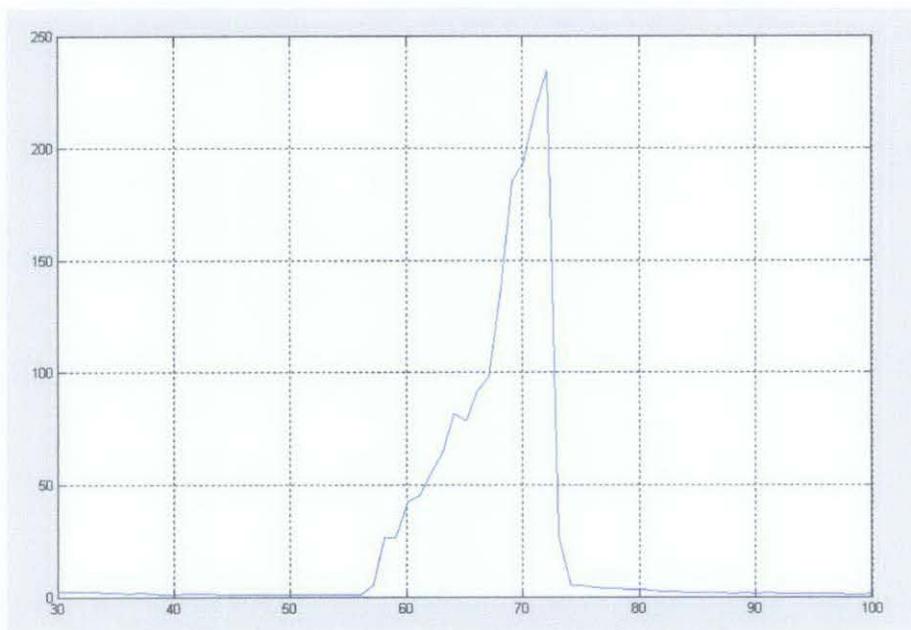


Figure 42 m_n metric at SNR = 30dB



Figure 43 m_n metric at SNR = 15dB

Figure 44 shows the time domain plots implementing timing synchronization. The first plot shows the training symbol and data that has been deliberately delayed.

This can be seen from the AWGN noise in front of the frame. The signal no longer starts from the beginning of the simulation. The second plot shows the frame after timing synchronization. Observe that the AWGN noise preceding the frame is no longer there. Thus, the receiver has identified the start of the frame.

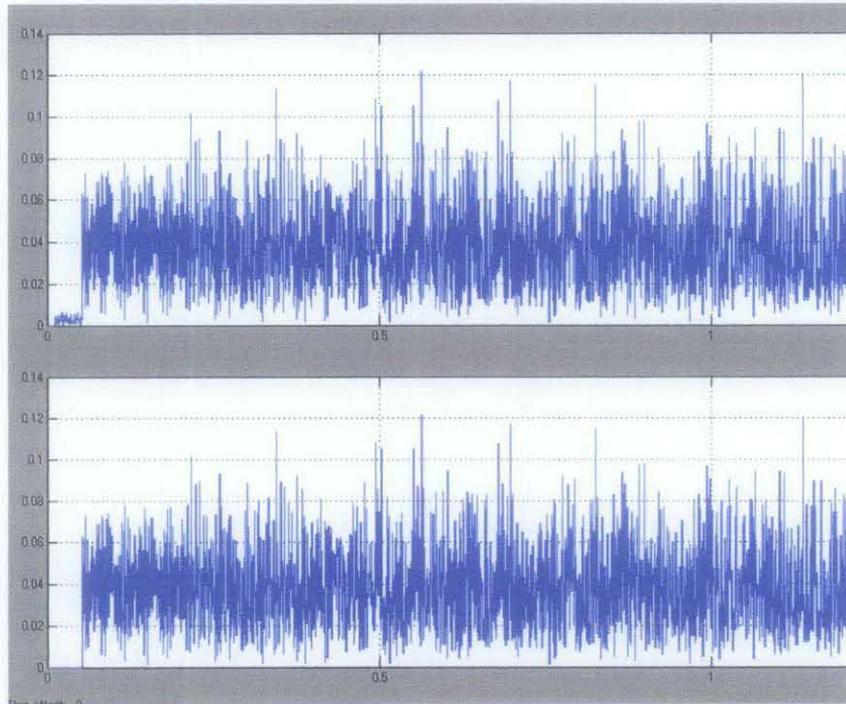


Figure 44 Time domain plot before and after timing synchronization

Figure 45 shows the performance of the DSW in determining the timing offset. The probability of false detection increases with lower SNR. Data traveling in an AWGN channel will have a lower probability of missing the peak compared to Rayleigh channels. In the model used in this FYP, the timing offset in Rayleigh channels can be determined accurately when SNR is 14dB or higher.

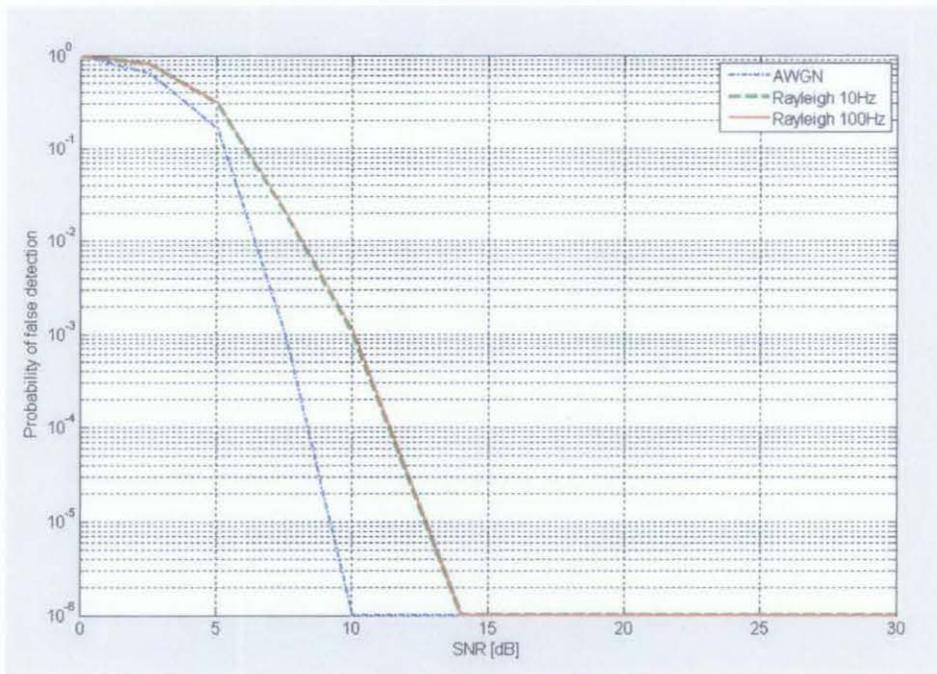


Figure 45 Performance of timing offset estimator vs SNR

4.2 Fractional frequency offset synchronization

The Schmidl and Cox method is used to handle the fractional frequency offset. From the plot of $-\langle P(d) \rangle$ at 30dB SNR (Figure 46 bottom), it is seen that this remain constant over where the plateau exists for the $M(d)$ metric. Any point therefore can be chosen as the fractional frequency offset. Nevertheless, the optimum point would be to the angle of $-\langle P(d) \rangle$ at the optimum timing which is at the end of the plateau. Figure 47 shows the enlarge view of the $-\langle P(d) \rangle$ angle and selecting the optimum point yields the fractional frequency offset (which is 0.35 in this simulation). Figure 48 and 49 depict the same thing with SNR = 15dB. This show that even at low SNR, the Schmidl & Cox method can obtain a close estimation of FFO. As expected, with lower SNR, the error of the estimation increases.

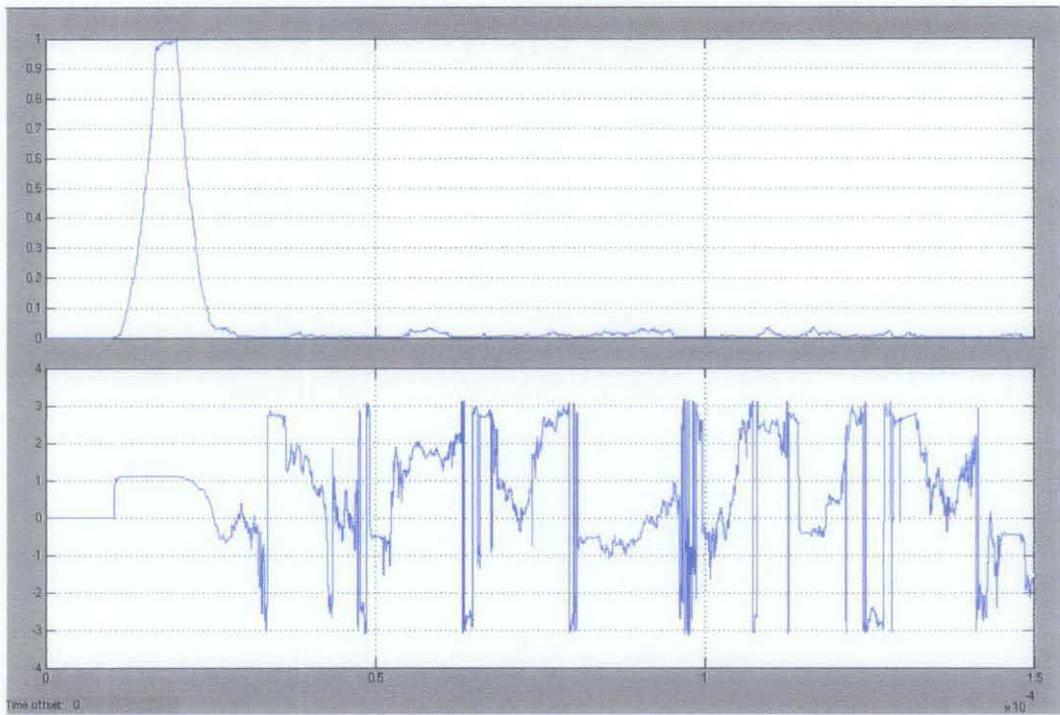


Figure 46 $M(d)$ and $-\langle P(d) \rangle$ plots from Schmidl & Cox method at SNR = 30dB

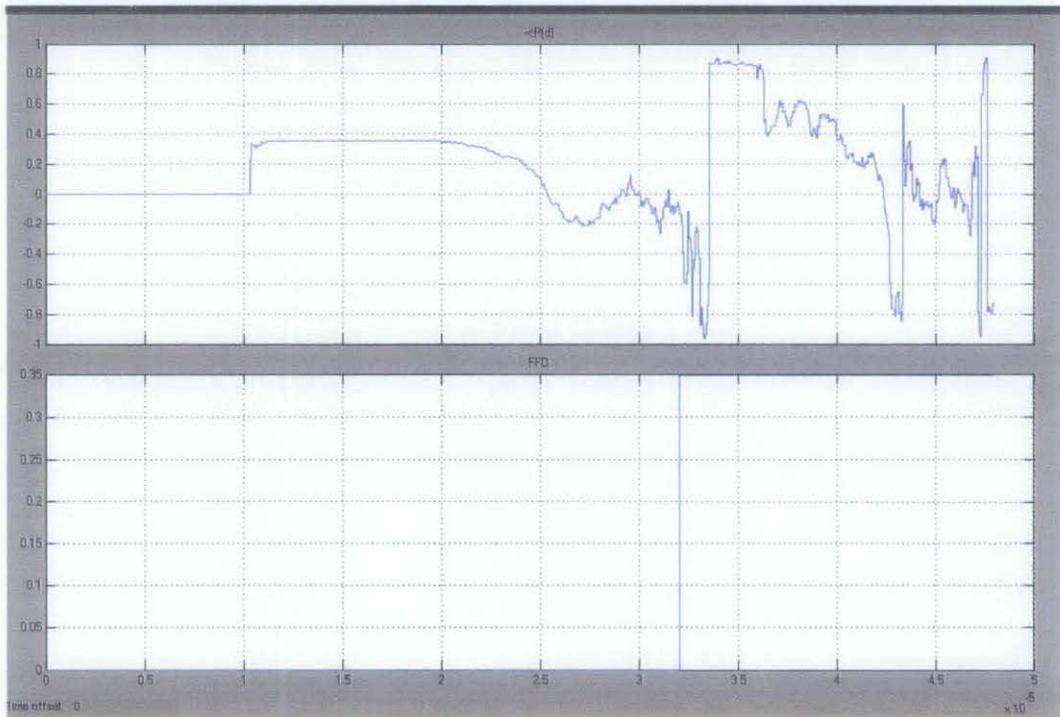


Figure 47 Enlarged $-\langle P(d) \rangle$ and resulting FFO plots for SNR = 30dB

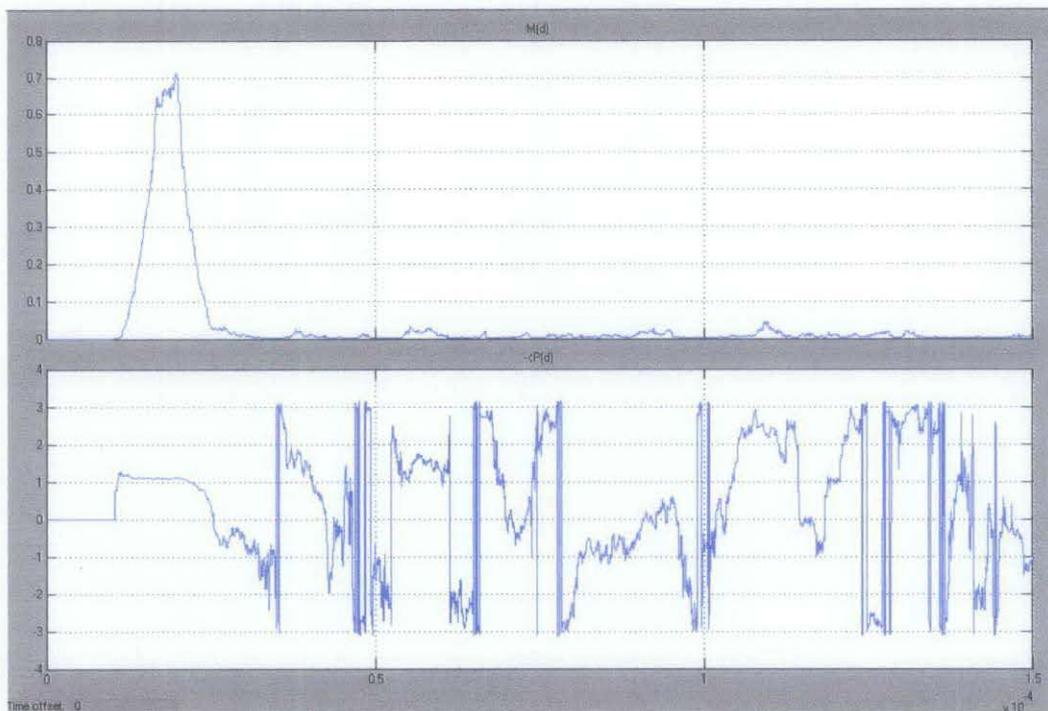


Figure 48 $M(d)$ and $\langle P(d) \rangle$ plots for Schmid & Cox at SNR = 15dB

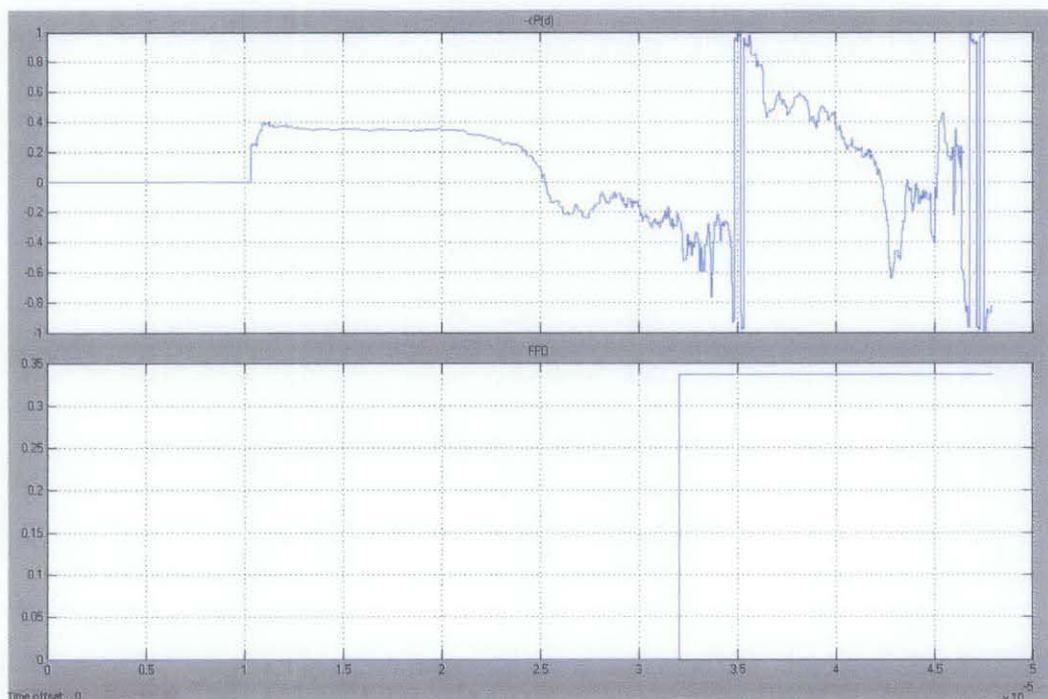


Figure 49 Enlarged $\langle P(d) \rangle$ and resulting FFO plots for SNR = 15dB

Figure 50 indicates that the estimation becomes closer to the correct value at higher SNR. Therefore, getting the Mean Squared Error for AWGN, Rayleigh at 10Hz and 100Hz maximum doppler frequency of the estimation yields the following

performance results. As expected, AWGN will give a lower MSE because it is assumed that the channel does not go through multipath fading. A higher Doppler frequency will affect the performance of the FFO estimator as it causes the channel to change more rapidly.

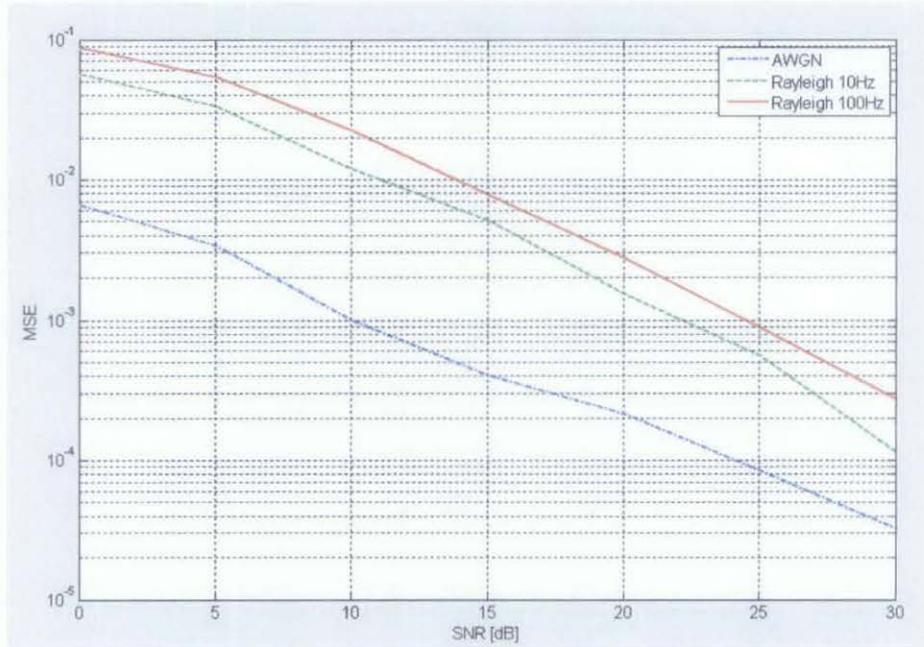


Figure 50 Estimation performance of FFO estimator versus SNR

4.3 Integer Frequency Offset synchronization

Using a local copy of the TS, the incoming signal is correlated with it to discover the start of the frame in terms of integer frequency offset. This correction must be done because the Schmidl and Cox technique only estimates the frequency offset up to one subcarrier spacing. In other words, only the fractional part of the frequency offset is determined. Since we know that the TS precedes the data, a correlation with itself will yield a peak at the start of the frame. As seen, there is a peak which occurs when the preamble is directly over the local copy. Calculating the shift will result in the integer frequency offset which will be used to correct the symbols. This is done by just shifting the symbols.

The peak is relatively easy to be found at high SNR (30dB) and at low SNR (15dB). This peak always crosses a threshold value and can be identified as the peak.

In fact, when timing and fractional frequency offset are corrected accurately, finding the IFO always gives the right value even for very low SNR.

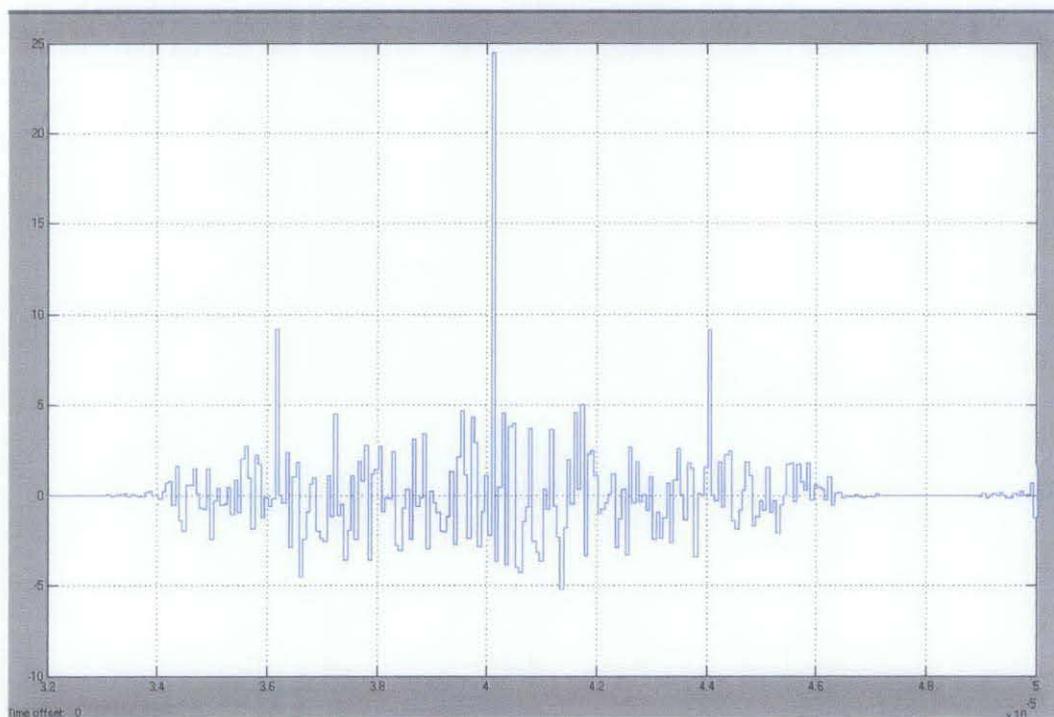


Figure 51 Decision metric for IFO at SNR = 30dB

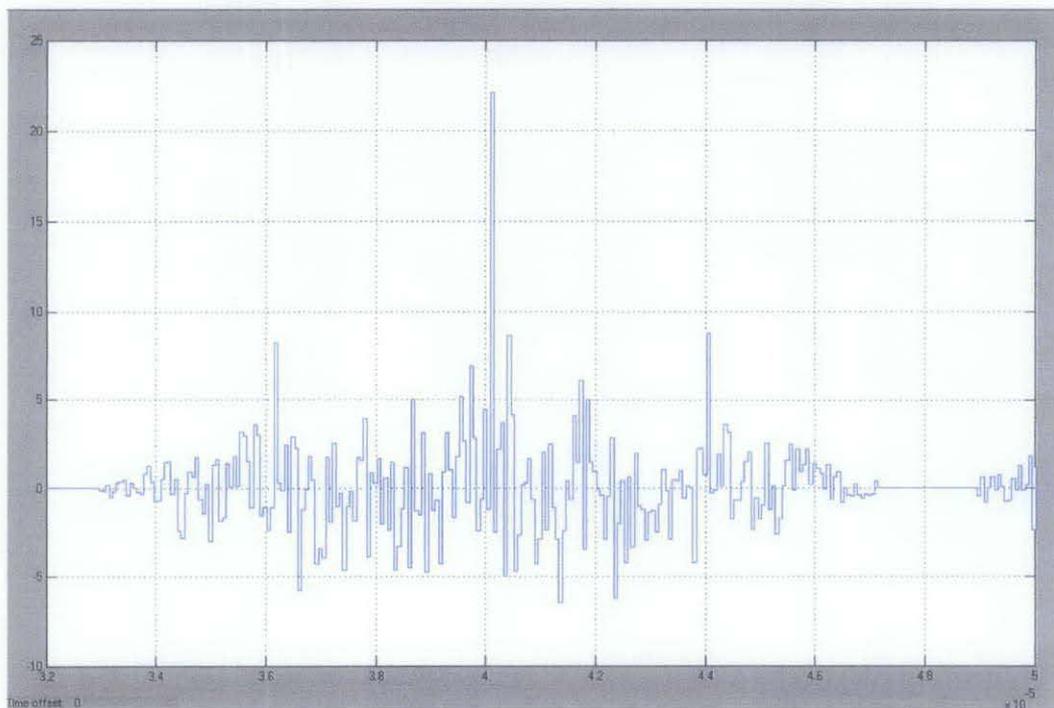


Figure 52 Decision metric for IFO at SNR = 15dB

Assuming that the correct time and FFO estimates have already been done, the IFO estimator accurately detects the IFO and has a probability of false detection of more than zero only at SNR lower than 5dB. For AWGN, the probability is lower compared to testing involving Rayleigh channels. This is shown in Figure 53.

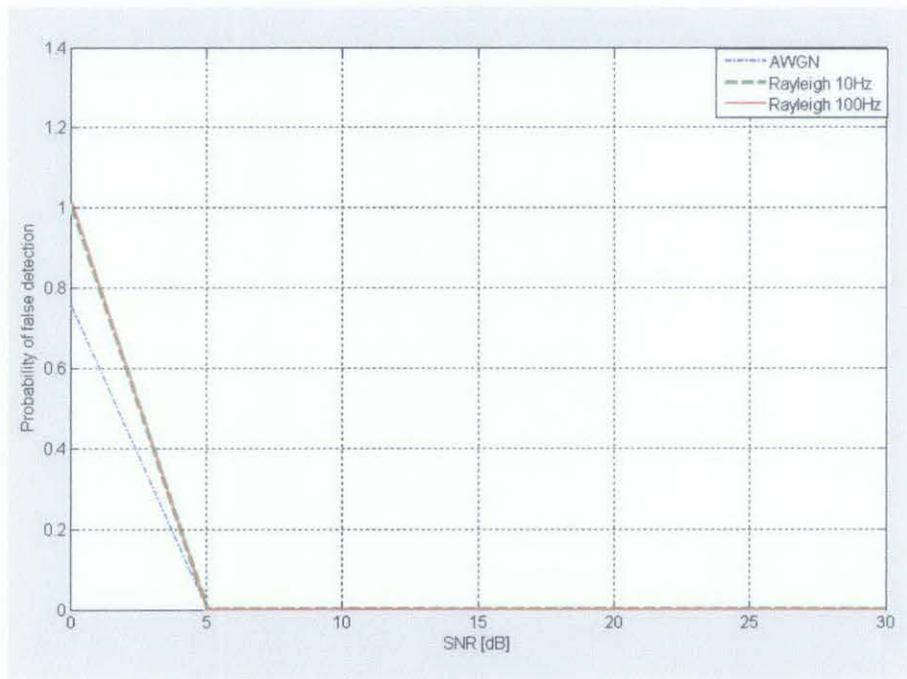


Figure 53 Performance of IFO estimator versus SNR

4.4 Residual Carrier Frequency Offset Synchronization

Figure 54 and 55 are graphs indicating the phase of pilot subcarriers. The RCFO can be obtained from the slope of the graph. This is from the L^{th} extension method which uses the difference in phase of pilot subcarriers for a current symbol and a symbol L distance away. Comparing the two figures, it is clear that at higher SNR, the accuracy of the slope that can be obtained is better.

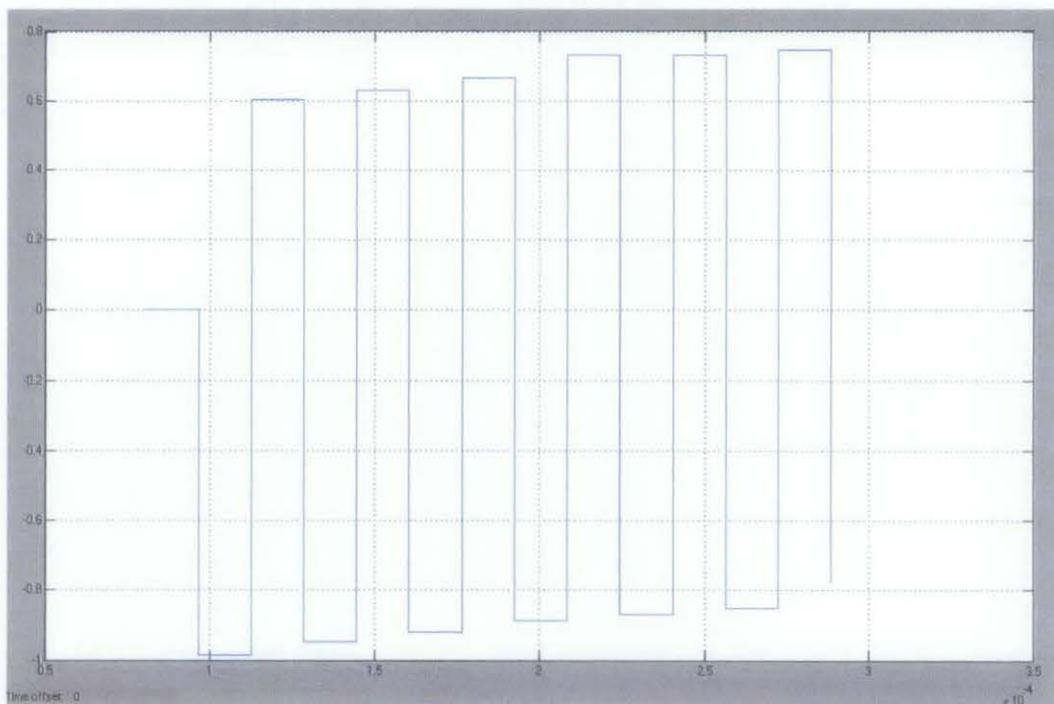


Figure 54 Graph for phase of pilot subcarriers at SNR = 30dB

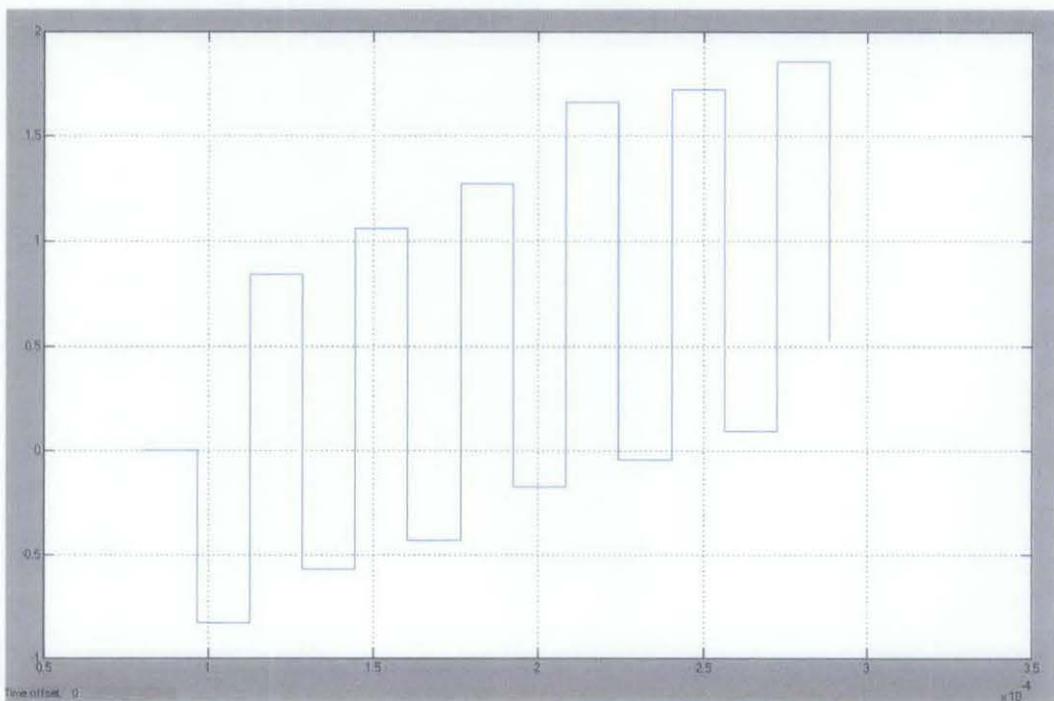


Figure 55 Graph for phase of pilot subcarriers at SNR = 15dB

The MSE for the RCFO estimator will decrease as SNR increases. A higher Doppler frequency in Rayleigh channels will lead to a slightly higher MSE since higher Doppler means that the channel is changing faster. Though at 10Hz and 100Hz

the channel remains coherent for the entire frame, the channel is only quasi-static and not fully static. Thus, at 100Hz, the channel gains are changing faster and will affect the accuracy of the slope obtained from the graph.

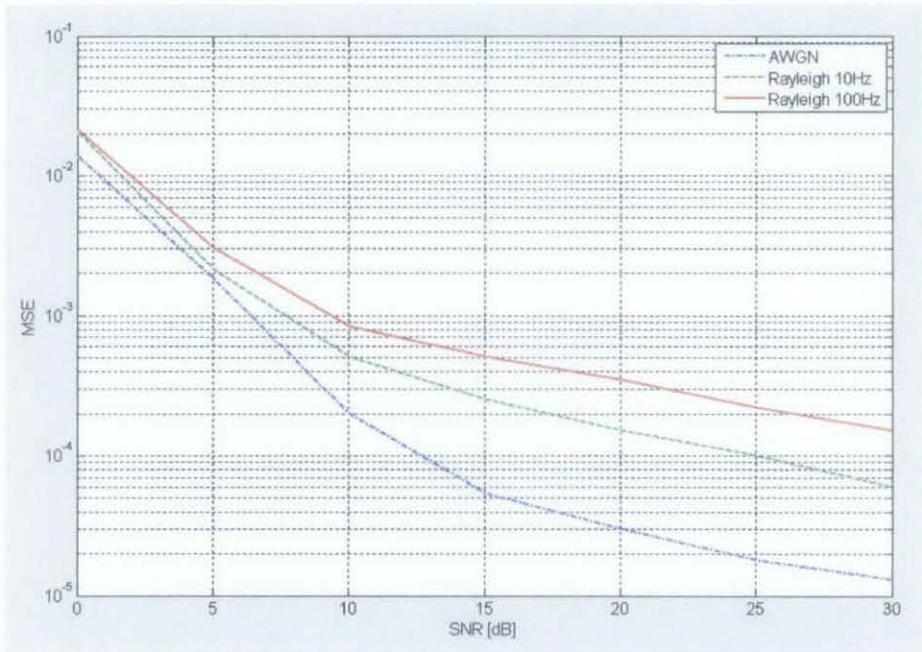


Figure 56 Performance of RCFO estimator versus SNR

4.5 Channel Estimation & Correction

Linear channel estimation is performed by dividing the incoming TS with a local copy to obtain the channel estimates. This is used to correct the rest of the incoming data symbols.

Figure 57 (left) shows the data at each subcarrier for one OFDM symbol before Channel Correction. Only the 200 subcarriers are shown in the diagram because only 200 subcarriers constitute the data subcarriers. It can be seen that this values are not the same as the ones transmitted (shown in Figure 58 (left)). Figure 57 (right) also shows the Complex Channel Estimate for each subcarrier.

Figure 58 (left) shows the data at each subcarrier for one OFDM symbol sent at the transmitter. Figure 58 (right) shows the data for the subcarriers after Channel Correction. It can be seen that this closely resembles the data sent at the transmitter

and can give correct demodulation of the data.

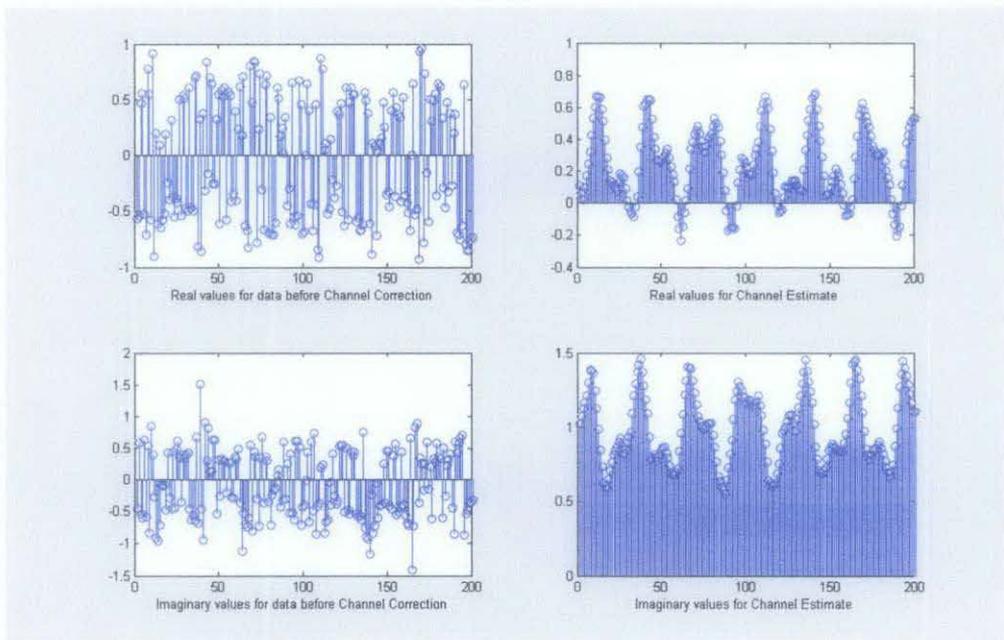


Figure 57 Real and imaginary plot for data before Channel Correction and the Channel Estimate at SNR = 30dB

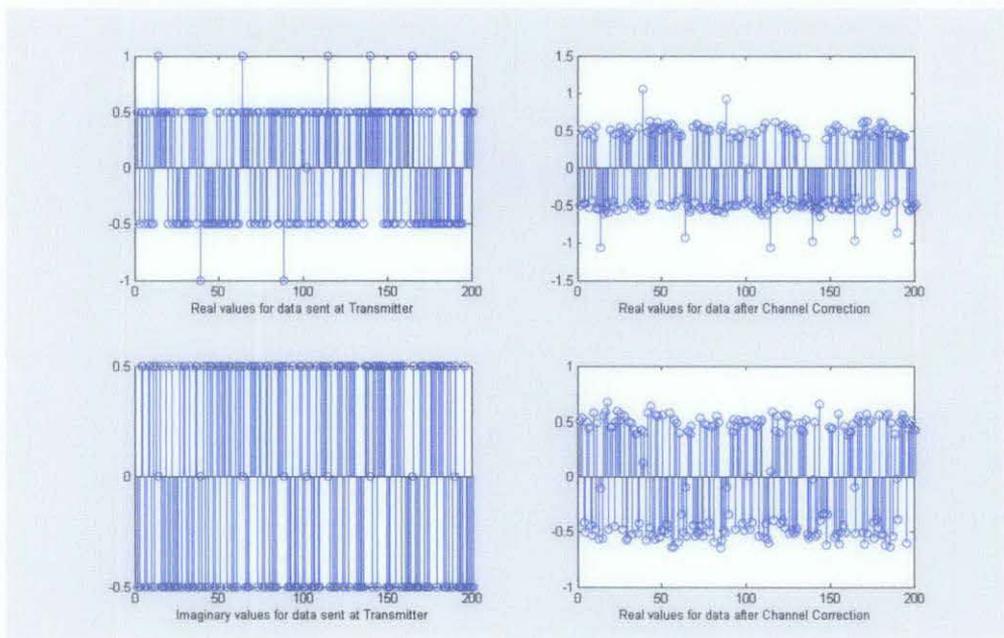


Figure 58 Real and imaginary plot for data sent at transmitter and after Channel Correction at SNR = 30dB

At low SNR (15dB), the channel estimate becomes inaccurate since it applies

linear channel estimation and correction. This is shown plot of data after channel correction. Figure 60 (left) shows the data at each subcarrier for one OFDM symbol sent at the transmitter at 15dB and it does not resemble the data for the subcarriers after Channel Correction shown in Figure 60 (right).

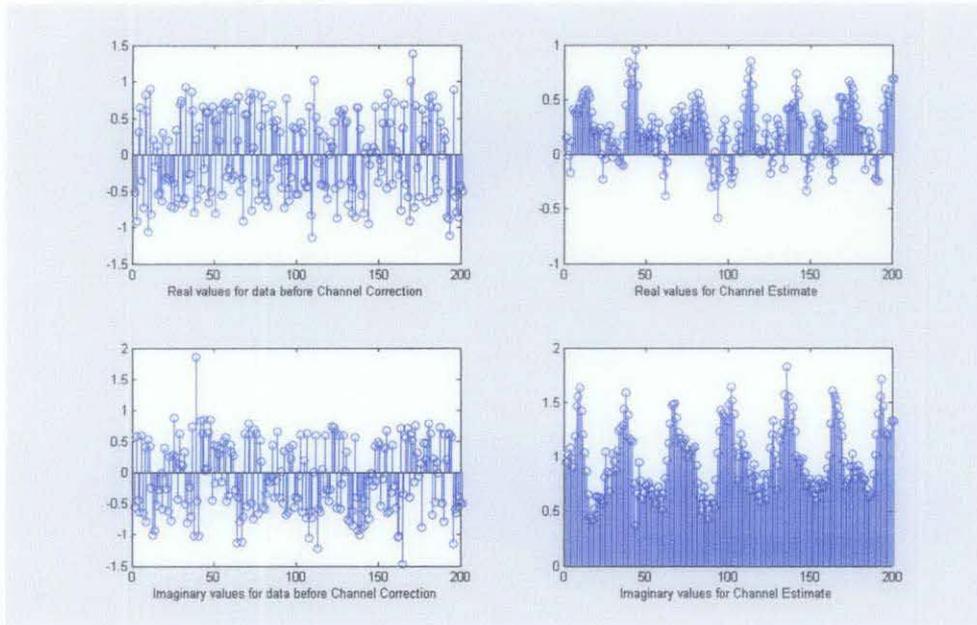


Figure 59 Real and imaginary plot for data before Channel Correction and the Channel Estimate at SNR = 15dB

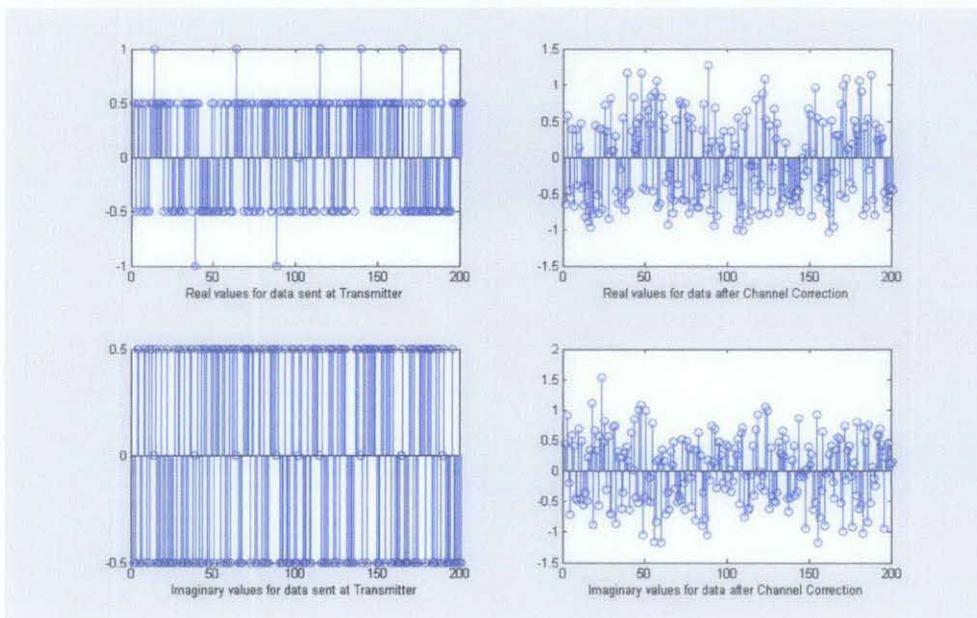


Figure 60 Real and imaginary plot for data sent at transmitter and after Channel Correction at SNR = 15dB

For the performance of the channel estimator, the plot shows that at 18dB and above it gives fairly accurate estimates with low margin of error. However, for SNR lower than 18dB, the error increases exponentially.



Figure 61 Performance of Channel Estimator vs SNR

4.6 Overall performance of Receiver

Figure 62 shows the BER vs SNR plot when SNR is varied from 15dB to 30dB. From this plot, the receiver's performance closely resembles the theoretical performance at SNR 18dB or higher. The theoretical performance of a WiMAX receiver is obtained from [1]. The Bit Error Rate is not satisfactory for lower SNR because a linear channel estimator is used. Although timing and frequency synchronization can be estimated to a fairly accurate results as shown earlier, the channel estimator is linear and the problem with this approach is that it not only inverses the channel but also the received noise. This noise enhancement can severely degrade the receiver's performance at low SNR [1]. For Rayleigh multipath fading simulations, the BER is slightly higher than the theoretical values and gets higher with increasing maximum Doppler frequency.

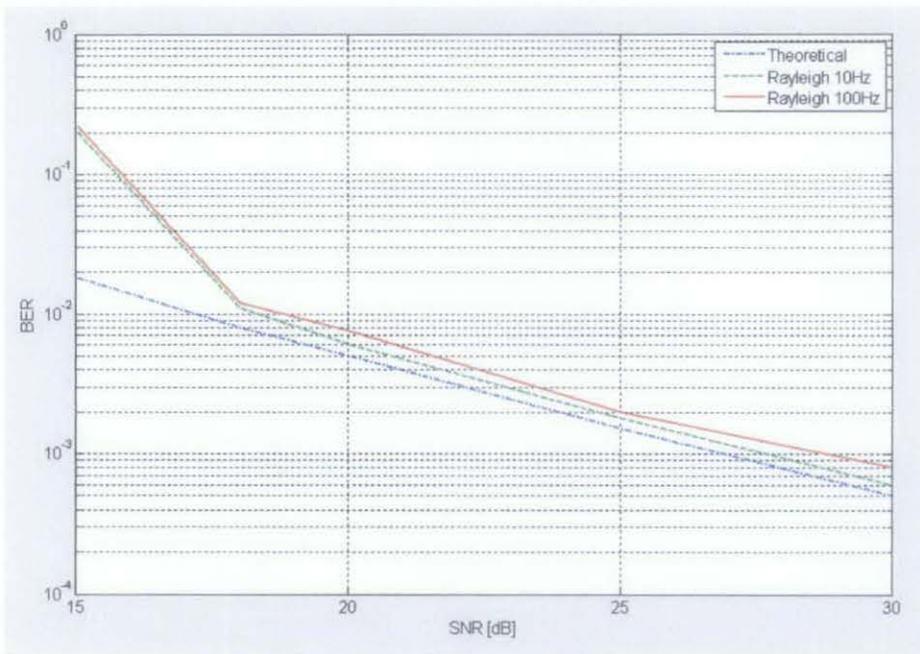


Figure 62 BER vs SNR at the receiver of WiMAX testbed

4.7 Validity test for WiMAX testbed in Simulink

This testbed has been tested for the SUI-3 channel model with slow fading for various SNR and offsets. Two test cases have been chosen to demonstrate the results, though the model can correct any timing and frequency offset introduced:

- Timing offset = 55 and Frequency offset = 2.35 at SNR = 30dB
- Timing offset = 82 and Frequency offset = 3.77 at SNR = 15dB

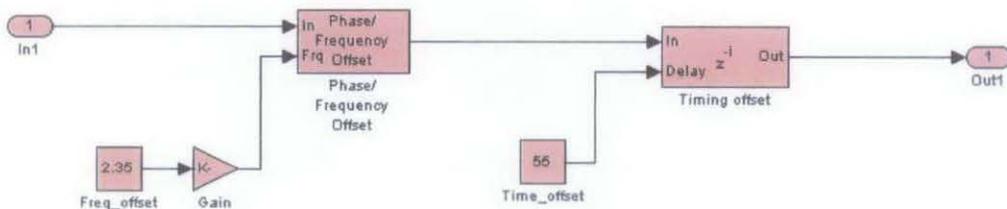


Figure 63 Introduction of timing and frequency offsets (Test Case 1)

Using the To Workspace blocks, we can transfer simulation results to the Array Editor. From Figure 64, it is clear that the estimation for timing, FFO, RCFO and IFO are close to the values that have been introduced. Thus, synchronization has been achieved. The S&C method will only give an estimate of the FFO and the RCFO estimator is needed to correct any remaining offset.

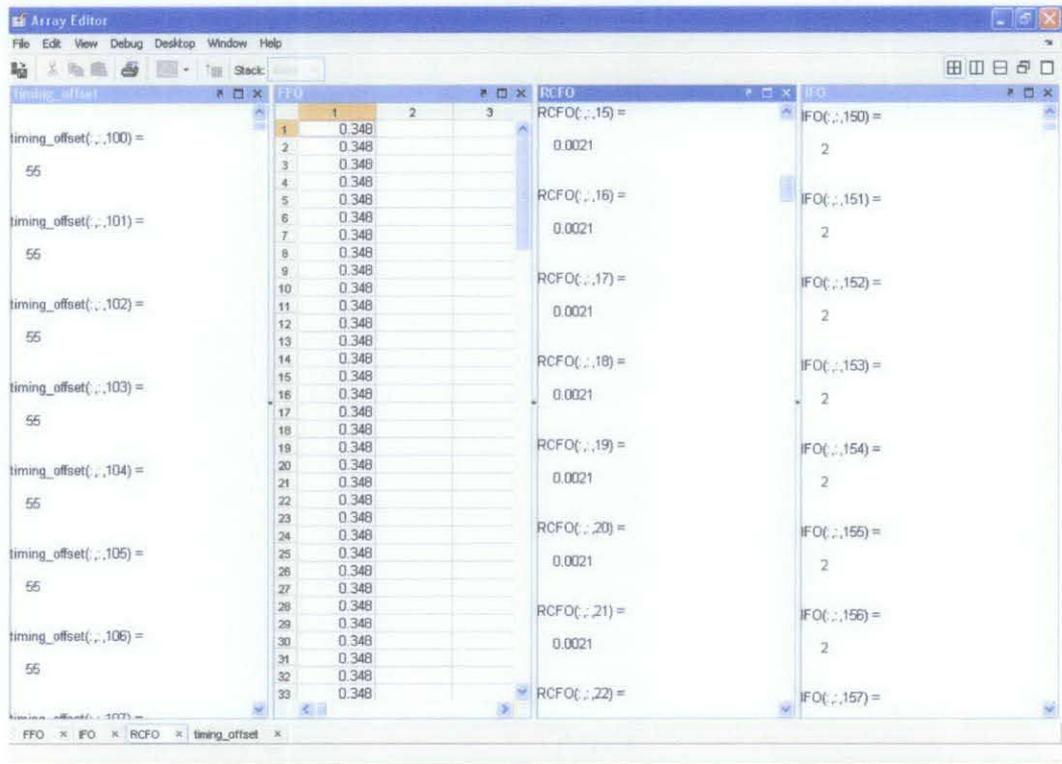


Figure 64 Results of simulation in Array Editor (Test Case 1)

Figure 65 shows the scatter plot for the transmitter whereas Figure 66 shows the scatter plot at the receiver with all the corrections performed at 30dB. From Figure 66, we can see that this plot resembles the QPSK plot. At the transmitter, QPSK modulation was performed. No rotation occurs at the scatter plot of the receiver, therefore synchronization has been achieved.

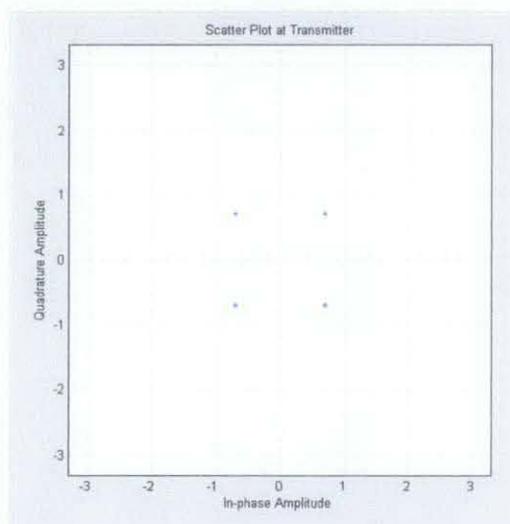


Figure 65 Scatter plot at the transmitter

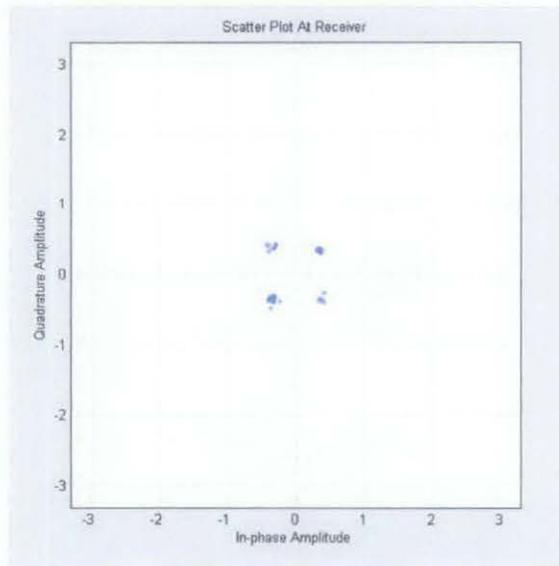


Figure 66 Scatter plot at the receiver with all corrections performed at SNR = 30dB

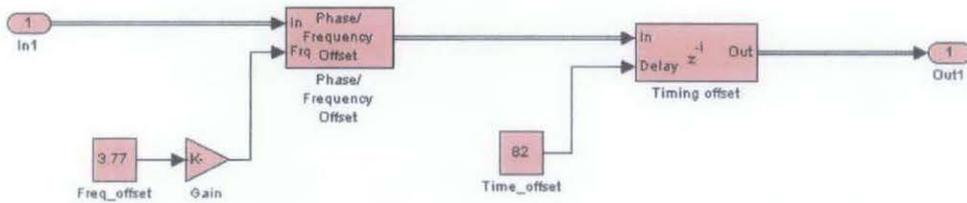


Figure 67 Introduction of timing and frequency offsets (Test Case 2)

From Figure 68, the results show that the estimation for the offset are accurate and close to those introduced. Note that the FFO is negative because if the IFO is an odd number, the FFO will be Actual FFO-1. However, this will not impact the results as the IFO estimated is 4. Summing this with the FFO will give the same value as the frequency offset introduced.

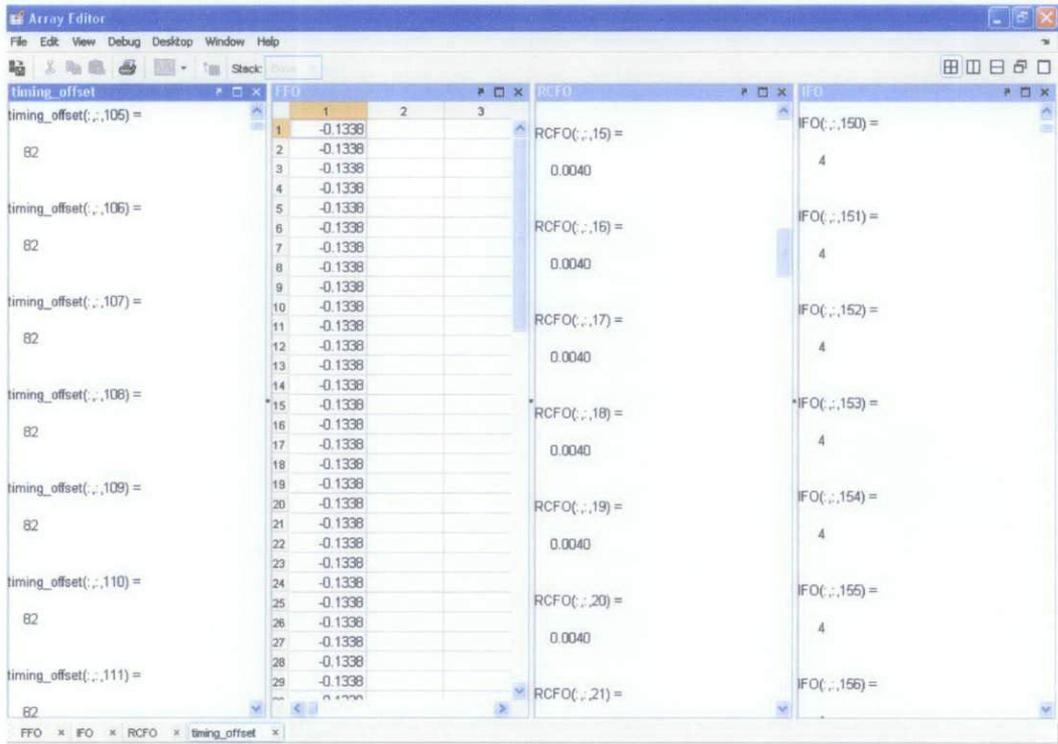


Figure 68 Results of simulation in Array Editor (Test Case 2)

At 20dB, the points have already started to drift far apart but are still within the decision boundary so it can still be demodulated to obtain the right data. The scatter plot proves that the estimation and correction technique was successful.

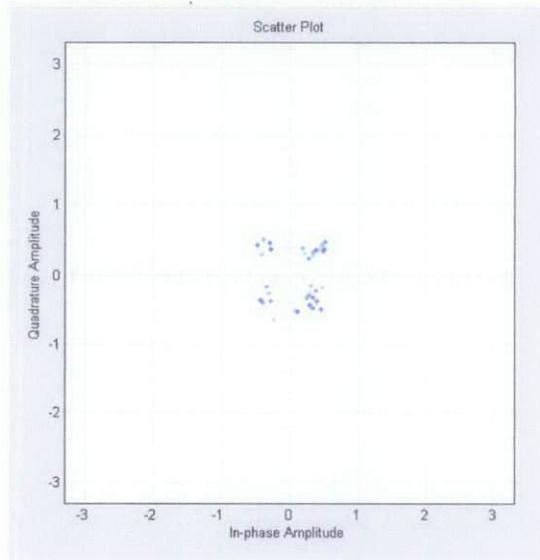


Figure 69 Scatter plot at the receiver with all corrections performed at SNR = 20dB

In Figure 70 and 71, by checking the Array Editor in MATLAB, it shows that the input data matches the output data. Thus the testbed is successful in sending and receiving the correct data. However, this testbed has its limitations as it cannot decipher the right data for SNR lower than 18dB due to the channel estimator which is linear.

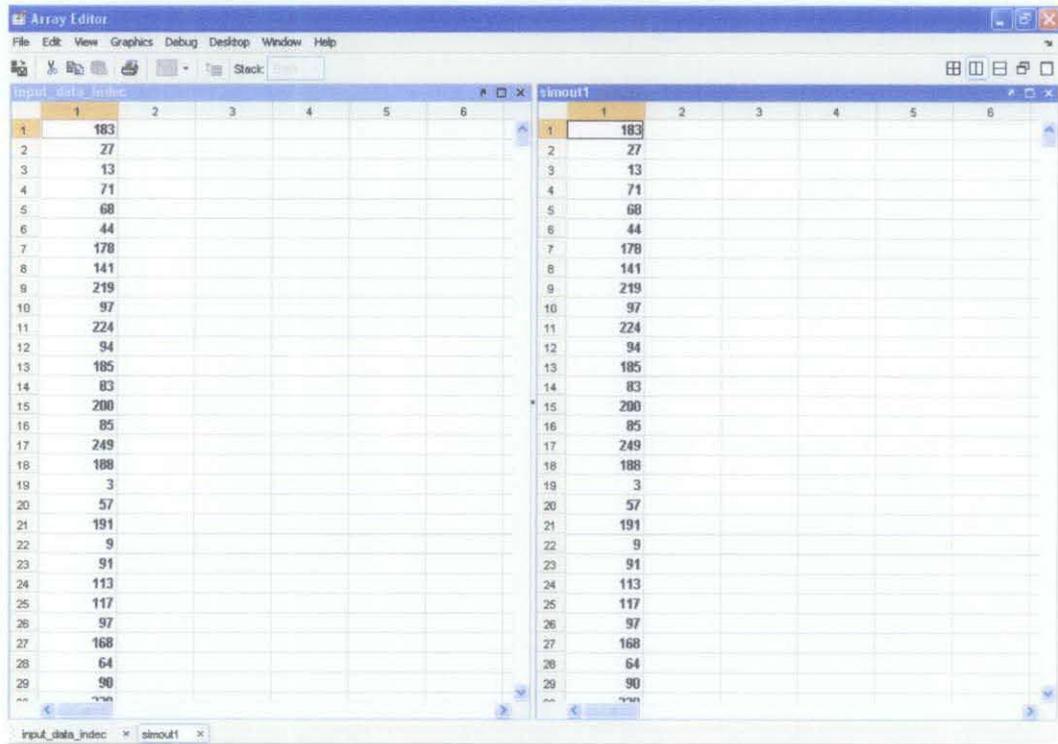


Figure 70 Results in Array Editor showing input data matches output data (A)

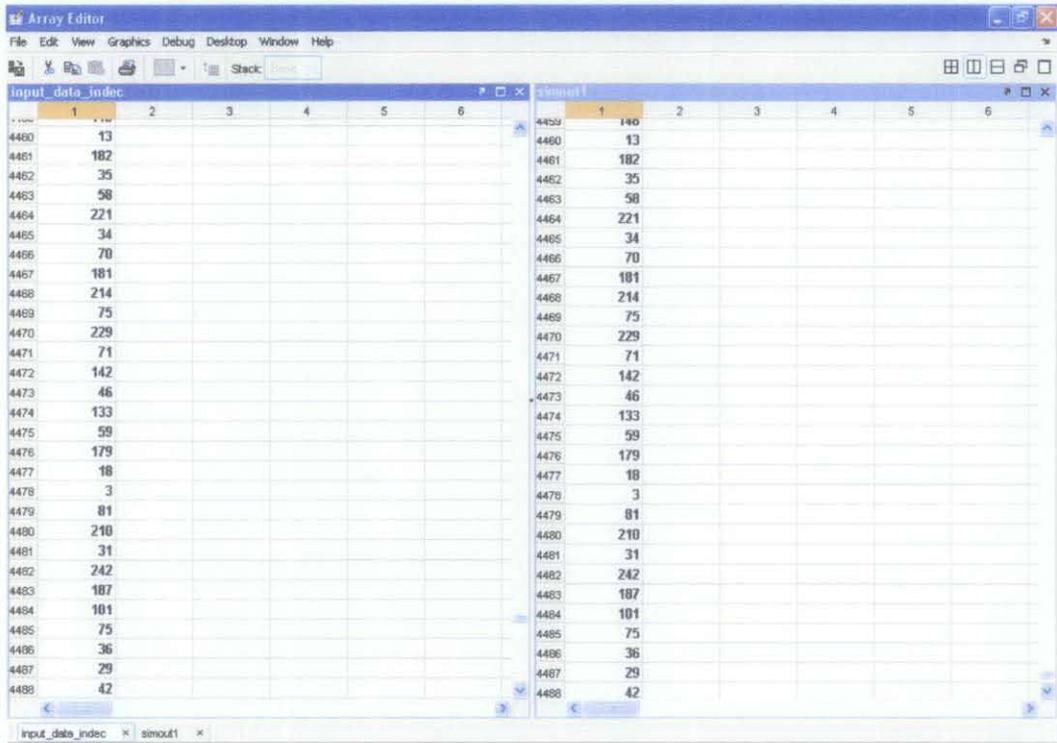


Figure 71 Results in Array Editor showing input data matches output data (B)

4.8 Validity test for WiMAX testbed in C codes

The executable file generated using RTW will run from the command prompt outside Simulink environment. It attempts to show that the receiver can be running in the processor as a C code and accepts input from the antenna. In this case, the input will be MAT files that have been generated with the same parameters as Test Case 1 in the previous section.

```

C:\>cd MATLAB\work\FYP_RMH\RTW\rsin
C:\MATLAB\work\FYP_RMH\RTW\rsin>wimax_rtw.exe -f input_real.mat=input_real.mat
-f input_imag.mat=input_imag1.mat
** Starting model 'wimax_rtw' @ Wed Jun 11 08:17:55 2008
** created wimax_rtw.mat **
C:\MATLAB\work\FYP_RMH\RTW\rsin>_

```

Figure 72 Screenshot of wimax_rtw.exe running in Command Prompt

After the simulation has ended, a MAT file will be generated containing all the results. In this FYP, the file is named `wimax_rtw.mat`. By using MATLAB, we can view the results of the MAT file. This is shown in Figure 73 and as expected, the results are similar to Test Case 1 in the previous section.

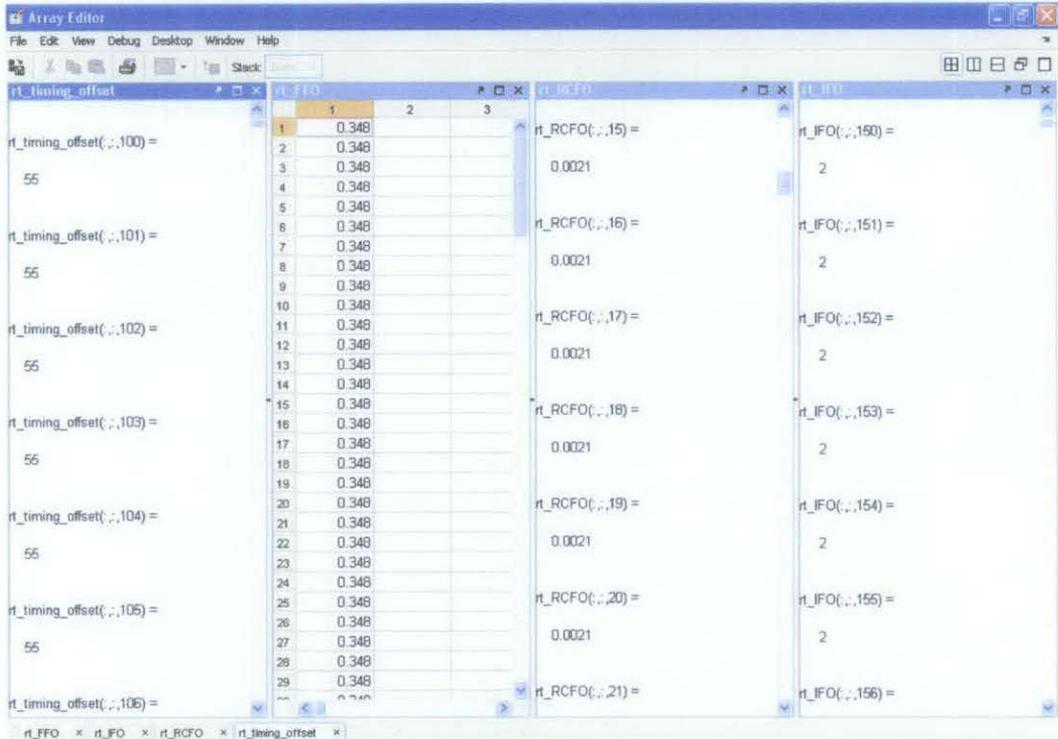


Figure 73 Results in Array Editor from file `wimax_rtw.mat`

Results shown in Figure 74 and 75 are displayed to show that they match Figure 70 and 71. This means that the C code running in the Command Prompt is successful in synchronizing and obtaining the same data as the ones being sent. Note that variables that are simulated via RTW have a prefix of “`rt_`” in front.

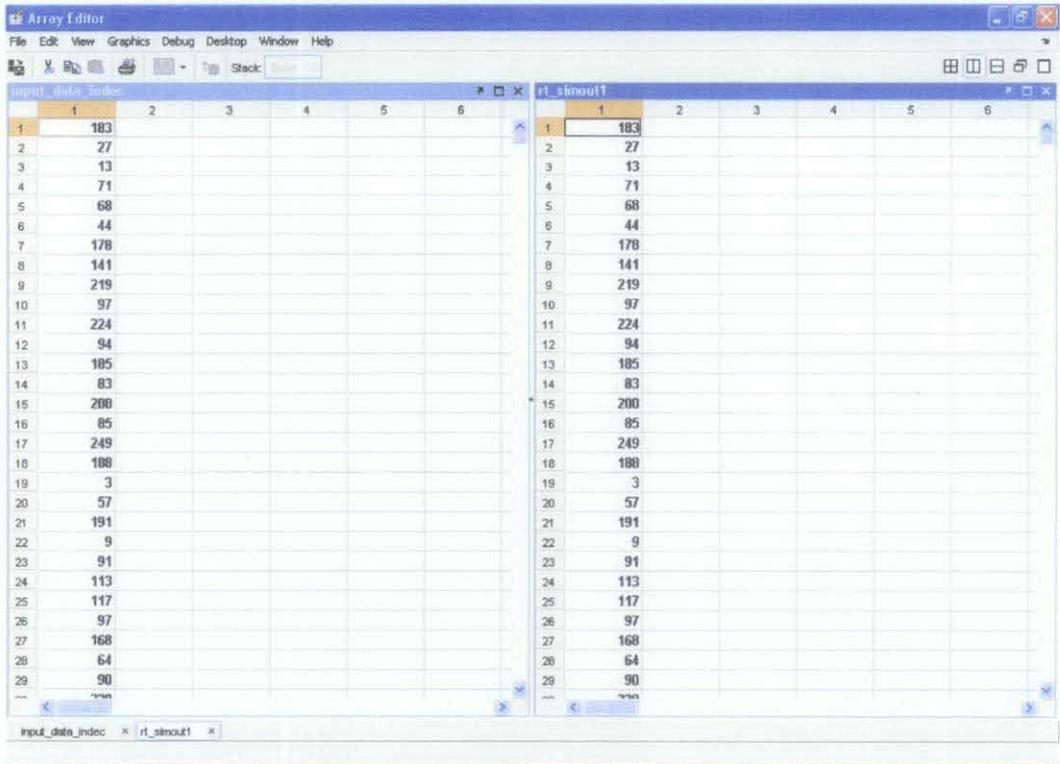


Figure 74 Input data matching output data from file wimax_rtw.mat (A)

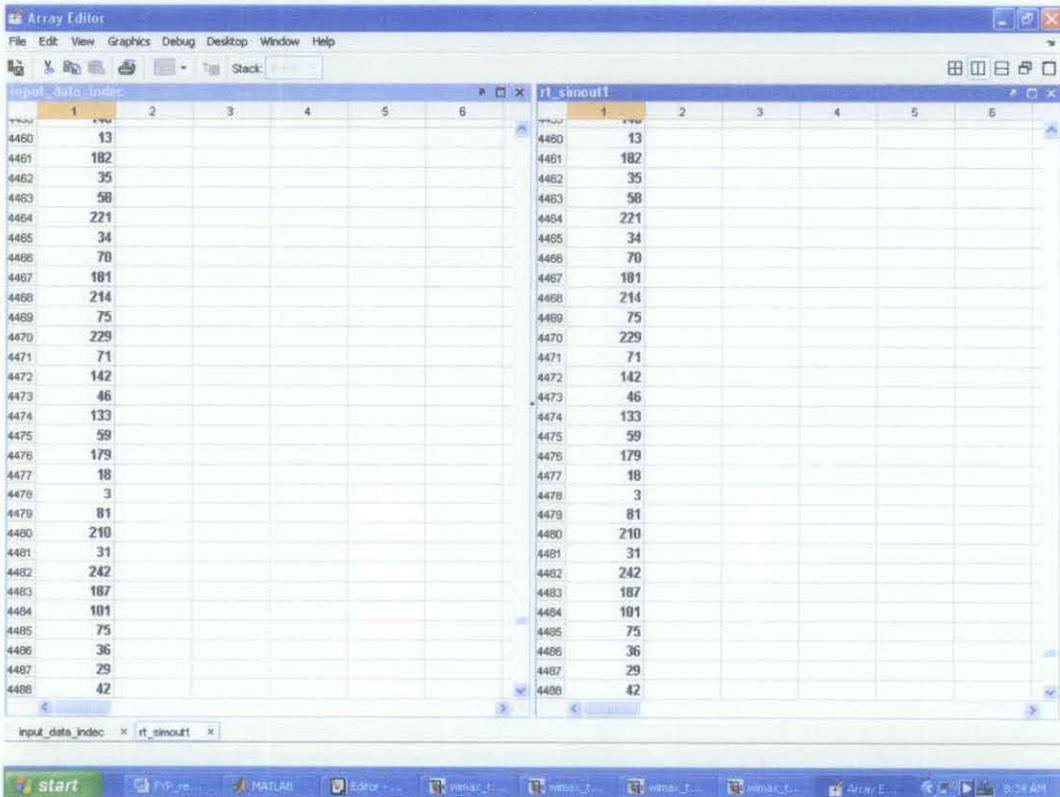


Figure 75 Input data matching output data from file wimax_rtw.mat (B)

4.9 Validity test for DSW and S&C in Xilinx blocks

To show that the Double Sliding Window and Schmidl & Cox that are running with Xilinx blocks are functioning correctly, two more test cases are shown:

- Timing offset = 37 and Frequency offset = 2.75 at SNR = 30dB
- Timing offset = 64 and Frequency offset = 3.8 at SNR = 15dB

The results are satisfactory for both cases as the estimation for timing and FFO are close to the values introduced. This means that the blocks are functioning as it should. Note that for IFO that an odd number, the FFO will be Actual FFO-1.

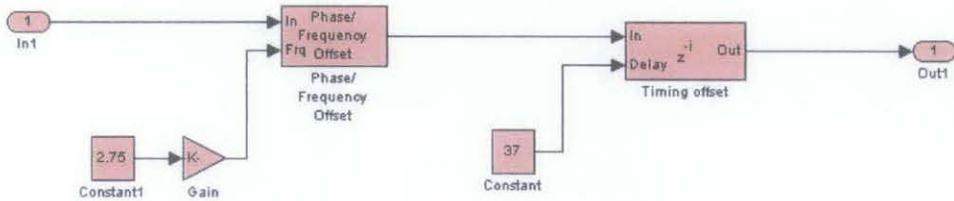


Figure 76 Introduction of timing and frequency offsets (Test Case 3)

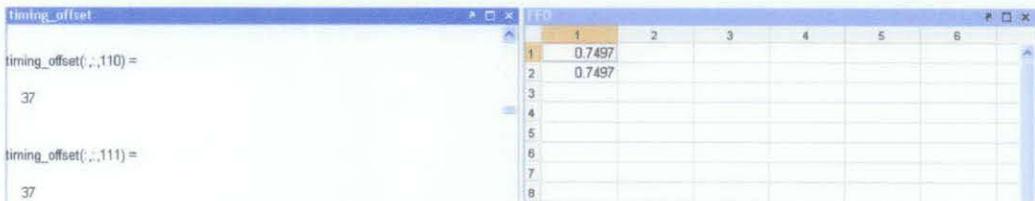


Figure 77 Results of simulation in Array Editor (Test Case 3)

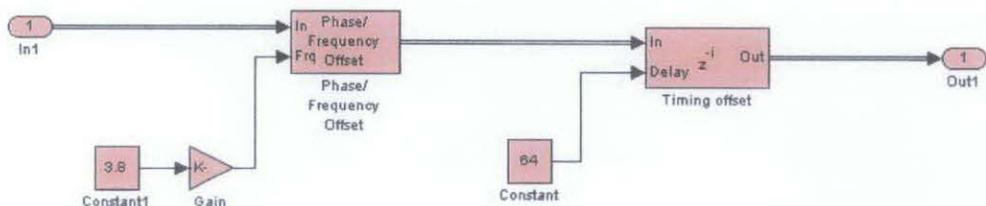


Figure 78 Introduction of timing and frequency offsets (Test Case 4)

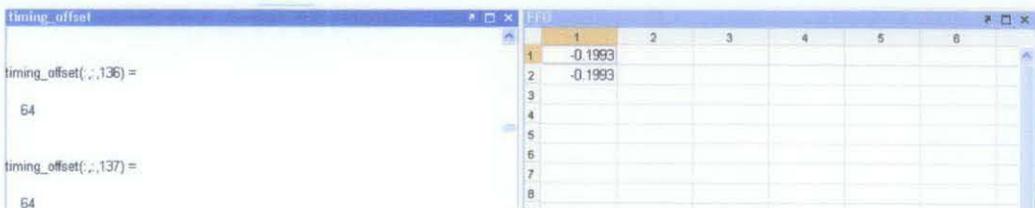


Figure 79 Results of simulation in Array Editor (Test Case 4)

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

This FYP has managed to reach all of its targeted objectives. I have designed the WiMAX testbed in Simulink according to the IEEE 802.16 standard consisting of a transmitter, channel and receiver. In other words, there is an end-to end implementation of the WiMAX system in Simulink. The base of the Simulink model is obtained from [14] and the Double Sliding Window and Schmidl & Cox blocks are obtained from [2]. My addition to the model is the design of the blocks for Timing Offset Correction, FFO Correction, IFO Estimation & Correction, RCFO Estimation & Correction as well as Channel Estimation & Correction. Furthermore, this FYP involves combining all these blocks using special control blocks to form a working testbed. With the software section of the project being completed, this FYP also has taken the first steps towards hardware implementation using two mainstream methods. Firstly, by converting the Simulink model into C codes via Real-Time Workshop. Secondly, is the translation of the DSW and the S&C in the Simulink model into Xilinx System Generator blocks. Using Xilinx blocks, the design can be converted into VHDL and bitstream and implemented in the FPGA.

Performance analysis is also done for every section of the receiver. Results of validity tests are also shown to prove that the testbed using Simulink, C codes and Xilinx blocks are functioning correctly. The results show that this testbed is working well as it can obtain the right values for synchronization and correct the incoming data symbols with the output data matches the input data. This subjected to an SNR of 18dB or higher. As shown in the results, although the estimation blocks give the correct estimates for timing, FFO IFO and RCFO at much lower SNR, since the channel estimation is linear, it uses the inverse of the noise as well to correct the

incoming data. This means that if the noise level is too high, the estimates will be wrong and thus causing the data to be wrong. Therefore, channel estimation using a linear estimator can only work with SNR of 18dB or more.

This testbed utilizes the Double Sliding Window for Timing estimation, the Schmidl & Cox method for FFO estimation. Correlation with the local copy of the TS is performed after FFT to obtain the IFO estimation and RCFO estimation uses the L^{th} extension method. Finally, linear estimation is performed to obtain knowledge about the channel.

5.2 Recommendation

First of all, my recommendation is to replace the channel estimation block with a non-linear estimator which is more robust and superior such as the LMS algorithm to give better performance in a real world system. Secondly, further work can be done on this project to continue converting all the Simulink blocks into Xilinx blocks and changing them into a bitstream. This can then be loaded into the FPGA for hardware-in-the-loop implementation. If rapid prototyping using RTW is to be implemented, conversion from analog data of the input into a format recognizable by the C program is necessary fully implement it in the hardware.

REFERENCES

- [1] Jeffrey G. Andrews, Arunabha Ghosh, Rias Muhamed, "*Fundamentals of WiMAX*," Prentice-Hall, 2007
- [2] R. van Nee and R. Prasad, "*OFDM for Wireless Multimedia Communications*," Boston: Artech House, 2000.
- [3] Simon Haykin, Michael Moher, "*Modern Wireless Communications*," Prentice-Hall, Inc., 2005
- [4] IEEE, "*Part 16: Air Interface for Fixed Broadband Wireless Access Systems*," IEEE Standard 802.16, 2004.
- [5] Ang Ken Li, "*WiMAX Inner Receiver Design*," Final Year Project Dissertation, Universiti Teknologi PETRONAS, 2006
- [6] K. Witrisal, "*OFDM Air-Interface for Multimedia Communications*," Ph.D Thesis, Technische Universiteit Delft, Netherlands, 2002.
- [7] T.M. Schmidl and D.C. Cox, "*Robust frequency and timing synchronization for OFDM*," IEEE Trans Commun., vol.45, no.12, pp. 1613-1621, Dec 1997
- [8] Yu-Cheng Lai, "*Research on Residual Carrier Frequency Offset Tracking in OFDM Wireless LAN Systems*," Master Thesis, National Chung Hua University, Taiwan, 2002

- [9] Amalia Roca , “Implementation of a WiMAX simulator in Simulink,” Diploma Thesis, Institut für Nachrichtentechnik und Hochfrequenztechnik, 2007
- [10] Zhang Lili, “*A study of IEEE 802.16a OFDM-PHY Baseband,*” M.S. Thesis, Linköping University, Sweden, February 16, 2005.
- [11] Nicola Laurenti, “*Implementation Issues in OFDM systems,*” Ph.D Thesis, Università degli Studi di Padova, 1998
- [12] Y.H. Kim, Y.K. Hahm, H.J. Jung and I.Song, “*An efficient frequency offset estimator for timing and frequency synchronization in OFDM systems,*” in Proc. IEEE 1999 Pacific Rim Conf. on Commun., Computers and Signal Proc., pp. 580-583
- [13] V.Erceg, et.al, “*An empirically based path loss model for wireless channels in suburban environments,*” IEEE JSAC, vol. 17, no.7, July 1999, pp. 1205-1211.
- [14] Mike Mulligan, Creating an Executable Specification for the WiMAX Standard. [Online]. Available at:
http://www.mathworks.com/company/newsletters/news_notes/oct06/wimax.html, January 2006.
- [15] WiMAX Forum, “*IEEE 802.16a Standard and WiMAX – Igniting Broadband Wireless Access White Paper,*” WiMAX Forum 2004.
- [16] What is WiMAX? [Online]. Available at <http://www.wimax.com/education>, May 5, 2008.
- [17] Wireless architectures – WiMAX [Online]. Available at
http://www.wimax.com/education/wimax/wireless_architectures, May 5, 2008

[18] Wei- Wen Hu, “*Carrier frequency synchronization for 802.11a*,” WITS Lab, Institute of Communication Engineering, April 2004

[19] Real-Time Workshop Development Team, “*Real-Time Workshop User Guide Version 3*,” The MathWorks Inc., 1999

[20] Xilinx Development Team, “*Xilinx System Generator for DSP User Guide Version 3.1*,” Xilinx Inc., 2004

APPENDIX A

CYCLIC PREFIX – FROM LINEAR CONVOLUTION TO CYCLIC CONVOLUTION

Consider a channel input sequence $x[n] = x[0], \dots, x[N-1]$ of length N and a discrete-time channel with finite impulse response (FIR) $h[n] = h[0], \dots, h[\mu]$ of length $\mu + 1 = T_m/T_s$, where T_m is the channel delay spread and T_s the sampling time associated with the discrete time sequence. The cyclic prefix for $x[n]$ is defined as $\{x[N-\mu], \dots, x[N-1]\}$: it consists of the last μ values of the $x[n]$ sequence. For each input sequence of length N , these last μ samples are appended to the beginning of the sequence. This yields a new sequence $\tilde{x}[n]$, $-\mu \leq n \leq N-1$, of length $N+\mu$, where $\tilde{x}[-\mu], \dots, \tilde{x}[N-1] = x[N-\mu], \dots, x[N-1]$, $\tilde{x}[0], \dots, \tilde{x}[N-1]$, as shown in Figure 12.5. Note that with this definition, $\tilde{x}[n] = x[n]_N$ for $-\mu \leq n \leq N-1$, which implies that $\tilde{x}[n-k] = x[n-k]_N$ for $-\mu \leq n-k \leq N-1$.

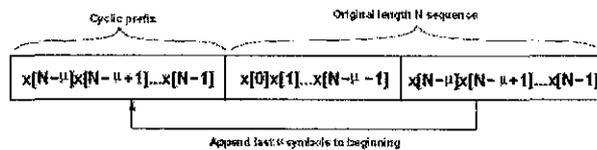


Figure 12.5: Cyclic Prefix of Length μ .

Suppose $\tilde{x}[n]$ is input to a discrete-time channel with impulse response $h[n]$. The channel output $y[n]$, $0 \leq n \leq N-1$ is then

$$\begin{aligned}
 y[n] &= \tilde{x}[n] * h[n] \\
 &= \sum_{k=0}^{\mu-1} h[k] \tilde{x}[n-k] \\
 &= \sum_{k=0}^{\mu-1} h[k] x[n-k]_N \\
 &= x[n] \odot h[n].
 \end{aligned} \tag{12.18}$$

