PEROMANCE ANALYSIS OF OFDM SYSTEM CONSIDERING QUANTIZATION EFFECTS

By

Imran Pervez

2007-NUST-MS PhD-ComE-05

Submitted to the Department of Computer Engineering
in fulfillment of the requirements for the Degree of

Master of Science
in
Computer Engineering

Thesis Advisor

Lt Col. Dr. Shoab A Khan

College of Electrical and Mechanical Engineering

National University of Sciences and Technology, Pakistan

2009

In the name of Allah, the most Merciful and the most Beneficent

ACKNOWLEDGMENTS

All praises to Allah almighty, the merciful, the magnificent. It was with His blessing only which helped me undertake this thesis and complete it.

This thesis is a culmination of the effort exerted not only on my part but with the help and encouragement of many people, whom I would like to acknowledge. I would like to extend my appreciation to my parents who always remained as a source of inspiration for me in all the ups and downs which I went through. Their sincere prayers helped me achieve my goals.

I am especially thankful to my advisor Dr Shoab A Khan for his valuable guidance and advice that resulted in successful completion of this work. His vast experience and knowledge in the fields of Digital Design and Signal Possessing was a valuable asset to me and helped me develop further interest in these subjects.

I thank my HOD Brig. Dr. Muhammad Younus Javed for his expertise and commitment in providing me and all other students of Computer Engineering department a chance to undertake this thesis and motivation for further research. I am thankful to Mr. Mudissir Cheema who provided us insight into synchronization techniques about OFDM. Finally I would like to acknowledge the guidance I received from all my teachers, in particular my guidance committee. The knowledge I gained from them proved helpful in undertaking this task.

Dedicated to my parents, for their unconditional love, support and encouragement.

Abstract

This thesis presents the effects of fixed point implementation on the performance of a digital communication system. In recent times orthogonal frequency division multiplexing (OFDM) has emerged as a promising technology for wireless and mobile communication systems, since it eliminates the use of complex equalizers and utilizes the bandwidth efficiently. OFDM is a multi-carrier modulation scheme which can support high data rates without degradation induced by interference by exploiting orthogonality amongst the sub-carriers. Most DSP and Digital Communication algorithms are implemented using fixed-point hardware having finite precision. The reason for this is that the use of fixed-point hardware is cost effective in terns of silicon area and power consumption. However this reduction in precision introduces quantization noise which degrades the performance, thus giving rise to the classical trade-off of signal quality vs. cost effectiveness in DSP. A generic OFDM transceiver with frequency and timing synchronization was implemented and was converted to fixed point. This thesis shows that the performance of OFDM communication system measured in terms of the Bit Error Ratio (BER) remains within tolerable bounds even when the word-lengths are reduced to a certain limit provided the communication channel is clean enough. This analysis can be used to design a reconfigurable architecture, where we can adapt our OFDM transceiver according to our need in terms of the number of errors or the power consumption.

Table of contents

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

1. Introduction:

In recent times orthogonal frequency division multiplexing (OFDM) has gained popularity in high data rate wireless applications. Simplified equalizer structures and resistance to fading and jamming are some of the reasons for OFDM becoming the physical layer of choice in upcoming new air interface standards. In fact a well engineered OFDM system with an efficient forward error correction (FEC) scheme might eliminate the need of an equalizer.

The development of large number of wireless air interfaces have resulted due to the ever increasing demand of high data rates in a limited RF spectrum. Second generation systems like GSM was developed for accommodating large number of voice applications in a limited bandwidth.

Apart form voice applications there has been tremendous growth in internet applications primarily due to the development of the wired Internet Protocol (IP). With the growing demand for wireless data and multimedia applications, cellular telephony and the Internet have become convergent technologies. This led to the development of 3rd generation air interface of WCDMA.

With the growth of wired internet there is a surging demand for wireless internet access as well. This has resulted in the development of a number of wireless local area network (WLAN) standards, which allow mobile connectivity to the internet. These are developed under the IEEE 802.11 working group. The IEEE 802.11a standard uses OFDM.

More recently the development is focused towards providing broadband fixed wireless access (BFWA) and mobile broadband wireless access (MBWA) comparable to ADSL, coaxial cable and satellite [2]. These technologies are referred to as 4th generation systems. The notable standards for BFWA and MBWA are IEEE 802.16 and IEEE 802.20 respectively. Both of these 4th generation systems use OFDM.

Similarly OFDM is considered for use along with multiple input multiple output (MIMO) in IEEE 802.11n WLAN standard.

OFDM has also seen application in broadcasting standards. These include the *Digital Audio Broadcast* (DAB) and *Digital Terrestrial Video Broadcast* (DVBT) standards in Europe and Japan.

Form the above description the importance of OFDM in the current and future wireless technologies is clearly highlighted.

This exponential growth of wireless communication would not have been possible without the advances in digital signal processing hardware. Many of the complex algorithms involved require dedicated hardware resources and are not realizable in software. However if we implement everything in hardware then it would be very costly. The trend in digital system design for wireless communication is to provide high performance at a low cost in terms of silicon area and power consumption.

In the development of a digital processing system the first step is the validation of the algorithm. For this purpose high level programming languages like C/C++ and other DSP friendly software like MATLAB are used. This helps in better understanding of the algorithm and helps in the HW and SW partitioning of the digital system. Once this decision is made than the level of precision, dictates the selection of the embedded processor to be used in the digital system.

The floating point variables in the high level simulation are converted to fixed point variables. By fixed point we mean that we set the number of significant figures on either side of the decimal point and then keep track of it. The Qn.m format is a fixed positional number system for representing floating-point numbers. Qn.m simply means that N-bit binary number has n bits to the left and M-bits to the right of the binary point. In case of signed numbers the MSB is used for sign and has negative weight. A fixed-point number in Qn.m format is equivalent to $\boldsymbol{b}=b_{n-1}b_{n-2}\ldots b_{-1}b_0.b_{-1}b_{-2}\ldots b_{-m}$ with floating point value

$$b_{n-1}2^{n-1}+b_{n-2}2^{n-2}+\ldots+b_1 2^1+b_0+b_{-1}2^{-1}+b_{-2}2^{-2}+\ldots+b_{-m}2^{-m} \rightarrow (1)$$

Floating point processors usually have a word length of 32 bits while fixed point processors have a word length of 16 bits. Thus the memory requirements for a floating point processor are higher than that of a fixed point processor. Hence a floating point processor is more expensive than that of a fixed point processor.

Another reason for the floating-point processor to be more expensive than its fixed point counter part is that in a floating-point HW, the radix point floats around and is recalculated with each operation and it automatically detects whether overflow has occurred and adjusts the decimal place by changing the exponent. This eliminates overflow errors and reduces inaccuracies caused by unnecessary rounding.

The design cycle for the development of a digital communication system involves validation of the algorithms involved at a high level. This is usually done using a high level programming language like C/C++ or DSP friendly MATLAB, which use floating point representation. However actual implementation is done using fixed point format. The fixed point format has less precision due to limited wordlengths.

With the world going wireless the major hurdle is the limited power of the hand-held device. The life time of the battery sets the upper limit on the time one can stay mobile.

1.1 Motivation:

Wireless communication systems and handheld devices have now become a necessity of the present modern life style. Due to active R&D in recent times these wireless communication systems are offering comparable performance to their wire-line counter parts and are on the verge of replacing them. However the major hurdle in this becoming a reality is that handheld wireless devices run on battery power, and require recharging. Hence power conservation which implies extended battery life is a hot area of research in these times. Apart form investing the chemistry of the battery itself, efficient signal processing approaches are also under study for reducing the power consumption by the communication system.

OFDM is used as the physical layer standard of choice in recent high data rate wireless systems. Some well known examples are WiFi, WiMax which are now becoming house hold names all have an OFDM based physical layer.

Digital signal processing (DSP) algorithms particularly digital communication algorithms are implemented using fixed point hardware like DSP processors or FPGAs (Field Programmable Gate Arrays). These fixed point devices have a finite word-length and hence finite precision. The greater the word-lengths used in implementation, higher will be the precision and greater will be the signal to noise ratio (SNR). However greater the word-length more will be the area and hence greater will be power required to drive the circuit. On the contrary if the word-lengths are reduced then the SNR decreases due to increase in the quantization noise. Thus there exists a trade off between the word-length and the power consumption. The problem then is to find the optimal word-lengths.

1.2 Problem Statement:

In this thesis we have investigated the effects of quantization due to fixed point implementation on the performance of OFDM communication system. It is shown through simulations that the performance of the system in terms of error rates remains with in tolerable limits even if we reduce the word-lengths; provide the channel is good enough. The aim is thus to find the wordlength which will give us the required BER for a given baseband modulation scheme, specified number of subcarriers and FFT length.

Also synchronization issues for OFDM are studied, and maximum likelihood (ML) estimation of frequency and timing offsets as reported by [1] is carried out, in order to counter the degrading effects of the communication channel.

1.3 Thesis Organization:

This thesis is arranged in six chapters. In chapter 2 gives the background information about OFDM and fixed point format (Q - Format). In chapter 3, the analysis of a generic OFDM modem for different Q formats is carried out and the BER curves for these different formats are presented. Chapter 4 describes the fixed point implementation of the OFDM system in Simulink. Chapter 5 presents the synchronization in OFDM, along with the implementation and performance of joint ML frequency and timing offset estimator in the system used. Chapter 6 presents the conclusion and further recommendations.

CHAPTER 2

2. Background:

This chapter presents an overview of OFDM and the mathematical representation of an OFDM signal. Also the Qn.m (Fixed point format) number representation, arithmetic and issues related to the Q format are described.

2.1 Overview of OFDM;

OFDM can be thought of as a hybrid of multi-carrier modulation (MCM) and frequency shift keying (FSK) modulation [3]. MCM is the principle of transmitting data by dividing the stream into several parallel bit streams and modulating each of these data streams onto individual carriers or sub-carriers; FSK modulation is a technique whereby data is transmitted on one carrier from a set of orthogonal carriers in each symbol duration. OFDM can also be viewed as a combination of modulation and multiplexing [4], where the multiplexing is done between the sub-channels coming form a data stream of a single user.

Orthogonality amongst the carriers is achieved by separating the carriers by an integer multiples of the inverse of symbol duration of the parallel bit streams. With OFDM, all the orthogonal carriers are transmitted simultaneously. In other words, the entire allocated channel is occupied through the aggregated sum of the narrow orthogonal sub-bands. By transmitting several symbols in parallel, the symbol duration is increased proportionally, which reduces the effects of ISI caused by the dispersive Rayleigh-fading environment.

2.2 OFDM as a special case of FDM;

In a classical parallel data system, the total signal frequency band is divided into N non overlapping frequency sub-channels. Each sub-channel is modulated with a separate symbol and then the N sub-channels are frequency-multiplexed. It seems good to avoid spectral overlap of channels to eliminate inter-channel interference. However, this leads to inefficient use of the available spectrum. To cope with the inefficiency, the ideas proposed from the mid-1960s were to use parallel data and FDM with overlapping sub-channels, in which, each carrying a signaling rate b is spaced b apart in frequency to avoid the use of high-speed equalization and to combat impulsive noise and multi-path distortion, as well as to fully use the available bandwidth.

The figure 1 illustrates the difference between the conventional non-overlapping multi-carrier technique and the overlapping multi-carrier modulation technique. As shown in figure 1, by using the overlapping multi-carrier modulation technique, we save almost 50% of bandwidth.

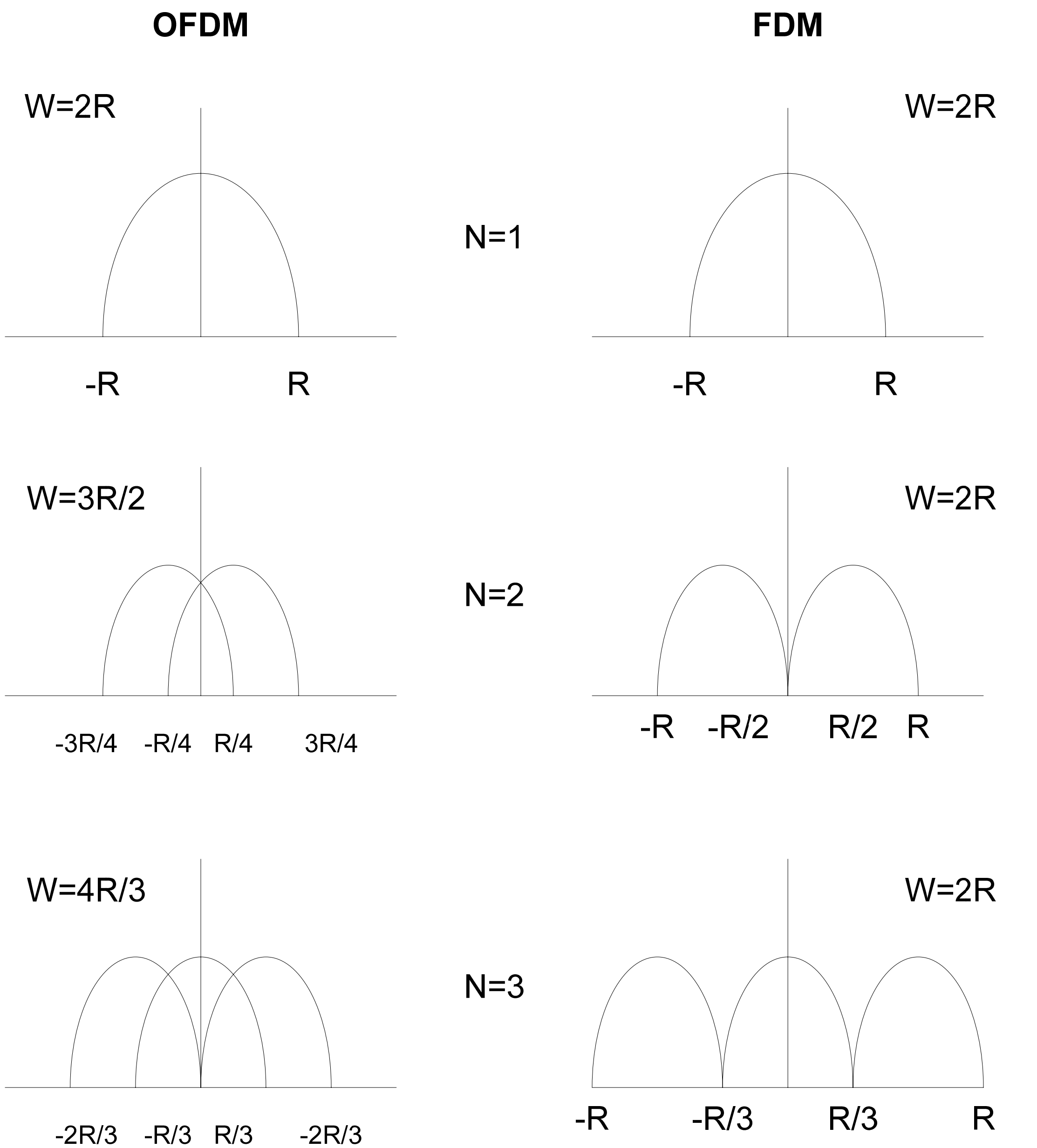

Figure 1: Comparison between orthogonal multi-carrier and conventional multi-carrier technique

To realize the overlapping multi-carrier technique, however, we need to reduce crosstalk between sub-carriers, which means that we want orthogonality between the different modulated carriers.

The word orthogonal indicates that there is a precise mathematical relationship between the frequencies of the carriers in the system. In a normal frequency-division multiplex system, many carriers are spaced apart in such a way that the signals can be received using conventional filters and demodulators. In such receivers, guard bands are introduced between the different carriers and in the frequency domain, which results in a lowering of spectrum efficiency.

It is possible, however, to arrange the carriers in an OFDM signal so that the sidebands of the individual carriers overlap and the signals are still received without adjacent carrier interference. To do this, the carriers must be mathematically orthogonal. The receiver acts as a bank of demodulators, translating each carrier down to DC, with the resulting signal integrated over a symbol period to recover the raw data. If the other carriers all beat down the frequencies that, in the time domain, have a whole number of cycles in the symbol period T, then the integration process results in zero contribution from all these other carriers. Thus, the carriers are linearly independent (i.e., orthogonal) if the carrier spacing is a multiple of 1/T.

In 1971, Weinstein and Ebert applied the Discrete Fourier Transform (DFT) to parallel data transmission systems as part of the modulation and demodulation process [6]. The results of this experiment were similar to those shown in figure 1, i-e there is no crosstalk from other channels. Therefore, if we use DFT at the receiver and calculate correlation values with the center of frequency of each sub-carrier, we recover the transmitted data with no crosstalk. In addition, using the DFT-based multi-carrier technique, frequency-division multiplex is achieved not by band-pass filtering but by base-band processing.

2.3 Mathematical treatment of OFDM signal;

Let $\left\{s_k\right\}_{k=0}^{N-1}$ complex symbols to be transmitted by OFDM modulation; the OFDM (modulated) signal can be expressed as

$$s(t)=\sum_{k=0}^{N-1} s_k e^{j2\pi f_k t}=\sum_{k=0}^{N-1} s_k \varphi_k(t),\ for\ \ 0\le t\le T_s, \rightarrow (2)$$

$$where\ \ f_k = f_o + k\Delta f\ \ and$$

$$\varphi_k(t)=\begin{cases} e^{j2\pi f_k t}\ if\ 0\le t\le Ts, \\ 0\ otherwise, \end{cases} \rightarrow (3)$$

For k=0, 1, 2, 3…, N-1. T_s and Δf called the symbol duration and sub-channel space of OFDM, respectively. In order for the receiver to demodulate OFDM signal, the symbol duration must be long enough such that $T_s\ \Delta f$ =1, which is also called orthogonality condition, defined as

$$\frac{1}{T_s}\int_0^{Ts} \varphi_k(t)\varphi_l{}^*(t)dt$$
$$=\frac{1}{T_s}\int_0^{Ts} e^{j2\pi(f_k-f_l)t}dt$$
$$=\frac{1}{T_s}\int_0^{Ts} e^{j2\pi(k-l)\Delta f t}dt$$
$$=\delta[k-l],$$

2.4 FFT Implementation:

In the previous section it was seen that to demodulate an OFDM signal an integral is needed. This leads us to develop a relationship between OFDM and Discrete Fourier Transform (DFT), which is economically, implemented using the Fast Fourier Transform (FFT). The FFT algorithm reduces the number of complex multiplications from N^2 as realized by the DFT algorithm to $\frac{N}{2}\log_2 N$ [2]. In the previous section in equation 1, an OFDM symbol was mathematically expressed as

$$s(t)=\sum_{k=0}^{N-1} s_k e^{j2\pi f_k t}$$

If s(t) is sampled at an interval of $T_{sa}=\frac{T_s}{N}$, then

$$S_n = s(n\Delta_s) = \sum_{k=0}^{N-1} s_k e^{j2\pi f_k t \frac{nT_s}{N}} \rightarrow (4)$$

Setting $f_0 = 0$, $f_k T_s = k$, we can write the above equation as

$$S_n = \sum_{k=0}^{N-1} s_k e^{\frac{j2\pi kn}{N}} = IDFT\{s_k\} \rightarrow (5)$$

Thus from the equation (4) we can see that the OFDM modulation can be achieved by taking the inverse Fourier transform of the message symbols. Typically the DFT is achieved by using the fast Fourier transform algorithm.

2.5 Cyclic Extension:

The major impairment which a wireless channel causes is known as 'Fading'. Fading is caused when multiple copies of a transmitted signal are received at the receiver following different paths. These copies are produced due to the reflection from the different obstacles. These copies arrive with different path delays. The maximum path delay is known as the 'Delay Spread' of the channel.

To overcome the delay spread of the wireless channel, a cyclic extension is usually done to the OFDM symbol. This cyclic extension is necessary because we want the starting of the OFDM symbol not to be corrupted due the effects of the delay spread. One way is to just let the symbol run longer in accordance to the amount of the delay spread, but this will not be useful since the symbol starting point will not be out of the danger zone. The method used is to append the last few samples within the OFDM symbol at the beginning of the symbol, and is known as cyclic prefixing. Typically the length of the cyclic prefix is ¼ of the OFDM symbol [4]. This is shown in the figure 2.

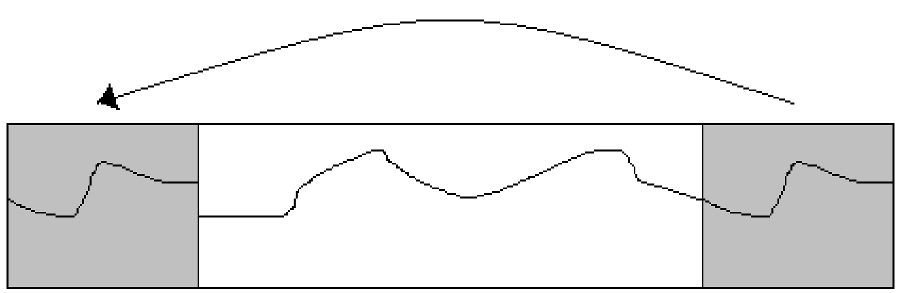

Figure 2: Cyclic Prefixing

The figure 2 shows that last ¼ of the OFDM symbols are appended to the beginning of the OFDM symbol as indicated by the arrow head.

2.6 Advantages of OFDM transmission;

1. Makes efficient use of the spectrum by allowing overlap
2. By dividing the channel into narrowband flat fading sub-channels, OFDM is more resistant to frequency selective fading than single carrier systems are.
3. Eliminates ISI and inter-carrier interference (ICI) through use of a cyclic prefix.
4. Using adequate channel coding and interleaving, one can recover symbols lost due to the frequency selectivity of the channel.
5. Channel equalization becomes simpler than by using adaptive equalization techniques with single carrier systems.
6. It is possible to use maximum likelihood decoding with reasonable complexity, as discussed in OFDM is computationally efficient by using FFT techniques to implement the modulation and demodulation functions.
7. In conjunction with differential modulation there is no need to implement a channel estimator.
8. Is less sensitive to sample timing offsets than single carrier systems are.
9. Provides good protection against co-channel interference and impulsive parasitic noise.

2.7 Disadvantages of OFDM transmission;

1. The OFDM signal has a noise like amplitude with a very large dynamic range; therefore it requires RF power amplifiers with a high peak to average power ratio.
2. It is more sensitive to carrier frequency offset and drift than single carrier systems are due to leakage of the DFT.

2.8 Realization of a Digital Communication System (DCS);

As mentioned in chapter 1, a DCS is realized using fixed point hardware having limited precision. In the following the effects of these limitations and fixed point arithmetic are discussed.

2.9 Qn.m Arithmetic;

Like the floating point numbers the same arithmetic functions are present for fixed point numbers. Since there is a limitation in the fixed point numbering system imposed by the position of the decimal point, the fixed point arithmetic has some pre-defined guidelines while performing arithmetic which need to be understood.

Since in digital hardware, addition and multiplication are the main arithmetical operations, while the other two operations of subtraction and division are achieved by addition and multiplication, we will look at fixed point addition and multiplication only.

2.9.1 Addition;

The addition of two Qn.m numbers with different values of n and m will result in a Q format number having the larger values of n and m among the two numbers.

Suppose there are two numbers a and b of Qn1.m1 and Qn2.m2 formats respectively. When added they result in a Qn.m format number, where n is larger of n1 and n2 and m is larger of m1and m2. In actual reality there is no decimal in hardware. Therefore designer needs to align the location of the decimal of both numbers and perform appropriate sign extension with less number of integer bits. This sign extension is only necessary for the integer part, the fractional bits are stored in least significant bits, require no extension.

2.9.2 Multiplication;

The multiplication of the Q format numbers ‘a’ and ‘b’ having a format of $Qn_1 .m_1$ and $Qn_2 .m_2$ results in $Qn_1 + n_2 .m_1 + m_2$ format product. If both numbers are signed 2′ s complement numbers, we get a redundant sign bit at MSB position. This redundancy is removed by left shifting the sign bit of the product by 1. Now the Q format of the product is changed to $Q(n_1 + n_2 - 1. m_1 + m_2 + 1)$.

2.10 Issues with Qn.m Arithmetic;

The implication of the Q format arithmetic and the limited range of numbers which a Q format number can accommodate results in some serious short comings in the fixed point number system. However the designer can easily overcome these shortcomings if he is aware of them. In the following we look at these limitations of the Q format and the remedies for them.

2.10.1 Bit Growth;

Multiplication of an N-bit number with an M bit number results in an N+M bit number. The problem aggravates in recursive computation. For example while implementing the recursive equation

$$y[n]= ay[n-1] + x[n] \quad \rightarrow (6)$$

requires multiplication of constant a with previous value of the output. If the first output is an N bit number in Qn1.m1 format and the constant a is an M bit number in Qn2.m2format, in the first iteration the product ay[n-1] will be an N+M bit number in Qn1+n2-1.m1+m2+1 format. In the second iteration the previous value of the output is now an N+M bit, and once multiplied by constant a will become N+2M number. The size will keep growing after every iteration of the algorithm. It is therefore important to curtail the size of the output. In many applications the size of out is same as the size of input. This requires truncating the output after every iteration. Truncation is achieved by ignoring least significant bits.

2.10.2 Rounding and Truncation;

In multiplication of two Q format numbers the number of bits in the product will increase. We sacrifice precision by throwing away some low precision bits of the product Qn1.m1 is truncated to Qn1.m2, where m2 < m1. Simply truncation of numbers will biased the results and in many applications it is preferred to round before trimming the number to desired size. For this 1 is added to the bit right of point of truncation and then the resultant number is truncated to desired number of bits.

2.10.3 Overflow and Saturation;

Overflow occurs if two positive or negative numbers are added and the sum can not be represented in a given number of bits. For example in a 4-bit representation if 1 is added in 7 the sum is 8, which can not be represented in 4-bits and will appear as -8. This will cause an error in all subsequent computations. This error is very grave and is equal to the full dynamic range of the number and adversely affects the results. It is therefore imperative to check overflow condition after performing arithmetic operations that can cause a possible overflow. If an overflow results, the design should use and set an overflow flag. In many circumstances, it is better to curtail the result to the maximum positive or negative value that number can take.

CHAPTER 3

Analysis of OFDM system for Different Q formats;

In this chapter we present the effects of quantization noise arising due to the fixed point implementation of the OFDM system. The Qn.m format discussed in the previous chapter is applied to a generic OFDM system and the word-lengths are kept on reducing from 16 bit precision right down to 2 bits. The chapter starts by giving a literature review of similar research followed by the results in the form of bit error rate (BER) curves generated via MATLAB simulation.

3.1 Related Work:

The implementation of a digital system in fixed point results in the introduction of quantization noise. This affects the signal quality and lowers the SNR of the system. The quantization effects should therefore be mitigated. A number of studies have been made to investigate these effects on the performance of different digital systems. In the following a brief description of the related work is presented.

On of the earliest work regarding the effects of quantization errors was presented in [8] where the author categorized the quantization effects on digital filters into four categories, namely quantization of system coefficients, errors due to A/D conversion, errors due to round-offs in the arithmetic, saturation of the signal value in case of overflow. In [9] a comparison of uniform vs. non-uniform quantization was done using a direct digital control system. It was shown that the spectrum of errors resulting due to non-uniform quantization was essentially white. In [10] the author showed that to make an active filter programmable, an appropriate selection of a quantization scheme was essential, since this selection strongly affects the characteristics of the filter response.

The effects of quantization on digital filters are also addressed in [11] where its effects are reported that quantization of the filter transfer functions results in filter poles and zeros existing only at specific locations in the z plane, the input being quantized, and noise being introduced from the DSP operations of multiplication and division. This paper gives the machinery to calculate finite word length effects and then shows how to minimize their significance through improved topology, differing word length, and improved DSP numerical properties.

The quantization effects are also responsible for degradations in compressed images using MPEG encoding, utilizing an 8×8 kernel. In [12] a post processing method is presented to overcome these quantization effects adaptively, utilizing both spatial frequency and temporal information extracted from the compressed data. The impact of quantization effects on the performance of a digital communication system has also been investigated. In [13] the quantization effects on the BER performance of a QAM receiver are studied. In [14] and [15] the clipping and quantization effect for WLAN and OFDM are studied and models describing the spectral characteristics of the distortion noise for each are developed. In [16] an analytical approach for analyzing the impact of the numerical accuracy on the DFT computations for a multi-carrier (MC) system has been presented.

A broadband 16 bit fixed point OFDM model using ANSI C is presented in [17]. The model the developed for applications requiring low power consumption such as the Joint tactical radio system (JTRS) and the tactical operations centre (TOC) also requiring small form factors.

3.2 OFDM Modem Description;

A typical OFDM modem has the following main components as shown in the schematic

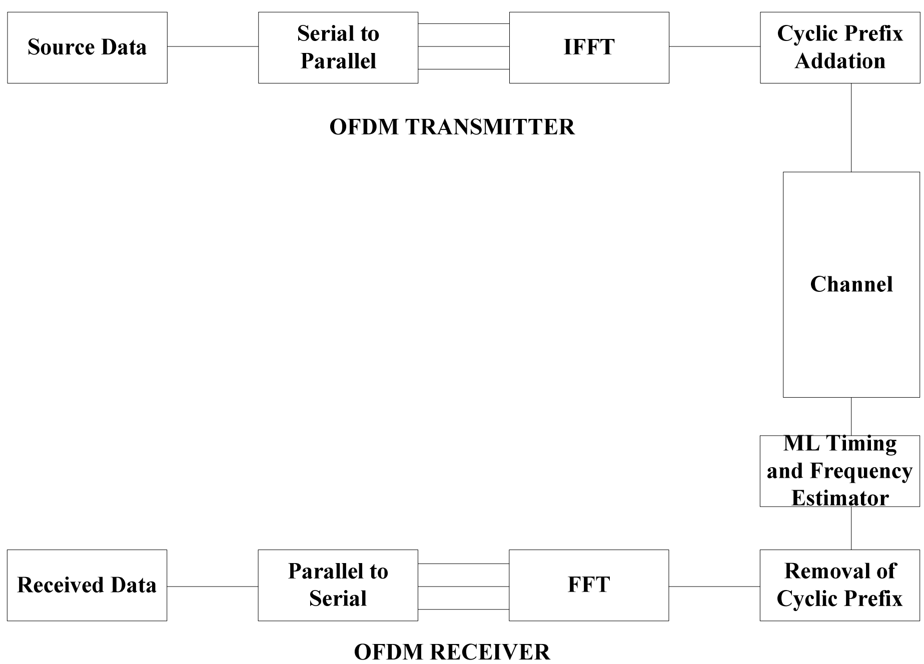

Figure 3: OFDM Transceiver

The figure 3 shows the basic blocks in a generic OFDM transceiver. The source data block consists of a data source followed by a base band modulator. The baseband modulator can be of any type like BPSK, QAM etc. In the thesis we have used QPSK modulator. The next block is the serial to parallel block, where the serial stream of baseband symbols corresponding to a single carrier frequency is converted to a parallel stream corresponding to a number of subcarriers. The number of outputs of this block must be an integer power of 2. The next block is the IFFT block which will introduce orthogonality among the sub-carriers. This is followed by the addition of cyclic prefix to counter the delay spread of the channel as mentioned in the section 2.6.

In the current thesis an un-coded version of the system is used. The system used for the simulation is resembles closely to the 802.11a WLAN standard.

3.3 Design Parameters

In this section the parameters used for the simulation of the OFDM system are discussed.

3.3.1 OFDM Transmitter;

The transmitter performs the following functions

1. *Random data generation*

Data is generated assuming a bit rate of 8192 bps

2. *Base-band modulation*

Grey encoded QPSK modulation is used, resulting in a symbol rate of 4096 symbols per second

3. *Serial to parallel conversion*

The generated serial data is converted to parallel and assigned to 64 orthogonal sub-carriers.

4. *Performing OFDM modulation*

In ofdm sub-carriers are orthogonal, where by they do not interfere with each other. The orthogonal carriers are generated with the help of Inverse DFT which is economically done by using the Inverse Fast Fourier Transform (IFFT). For the simulation 64 point IFFT is taken. Thus each OFDM block consists of 64 QPSK symbols.

5. *Addition of redundancy to combat the channel delay*

Cyclic prefix is added to combat the delay spread of the channel. Typical length of the cyclic prefix is 25% of the FFT symbol duration. In our case where the OFDM symbol is of 64 samples, the cyclic prefix is of length equal to 16 samples.

6. *Parallel to serial conversion for transmission*

The data coming out of the IFFT block is in parallel form and is converted to serial. Practically this means that the IFFT data is super imposed over a RF carrier for transmission.

3.3.2 The Channel;

The received signal is mathematically described as

$$Y = H*X + N \rightarrow (7)$$

Where X is the transmitted signal, Y is the received signal, N is the noise which is added and H is the channel impulse response. For the analysis both Additive White Gaussian Noise (AWGN) channel and flat fading rayleigh channel with diversity order 1 is used. For AWGN H=1.

3.3.3 OFDM Receiver;

The receiver performs the following steps which are the converse of those performed at the transmitter

1. *Synchronization of frequency and timing*

Synchronization is required to find the starting of each OFDM symbol. For OFDM the frequency synchronization is of particular importance and an OFDM signal is particularly sensitive to frequency offsets. Due to the presence of the cyclic prefix the timing offsets are easily accounted for and the OFDM signal is robust to these. Both frequency and timing offsets are jointly estimated using maximum likelihood algorithm which is discussed in detail in chapter 5.

2. *Serial to parallel conversion*

The serial data coming from the offset estimator is converted to virtual sub-carriers by converting it to parallel.

3. *Removal of redundancy*

The cyclic prefix is removed.

4. *Performing OFDM demodulation*

The data is then sent into the FFT block and the OFDM demodulation is carried out. 64 point FFT is done.

5. *Parallel to serial conversion*

The FFT data is then converted to serial.

6. *Base-band demodulation*

The serial data is then demodulated using QPSK demodulation function.

7. *Reception of message*

The demodulated signal is then converted to symbols and compared against the transmitted message and the errors are counted using the "*biterr*" function of the communication toolbox of MATLAB.

3.4 Bit Error Rate (BER) Curves for OFDM;

The performance of any digital communication system is conveniently expressed in terms of BER curves. Bit error rate is the ratio of the number of bits received in error to the total number of bits transmitted by a DCS. Smaller the value of the BER, better is the performance of the system.

A BER curve is a plane where the y-axis represents the BER values and the x-axis represents the value of SNR (in dB) or for a digital communication system EbNo. Thus the BER curves tell us the number of bit errors for different values of SNR. The steeper the curves i-e the more closely it is towards the origin the better is the system performance. For un-coded systems the theoretical limit is 1.6 dB.

3.4.1 BER curves for OFDM over AWGN channel in double precision format;

OFDM can use any base band modulation and the BER performance OFDM signal is similar to that of the base band modulation used [4].

In the system that was used in the research, 4096 QPSK symbols were generated per second. The BER curve for the floating point was generated using *Monte Carlo* simulation with 1000 trials. The theoretical curve was obtained using 'bertool' of communication toolbox in MATLAB. The BER curve for OFDM is similar to that of QPSK and is shown below in figure 4.

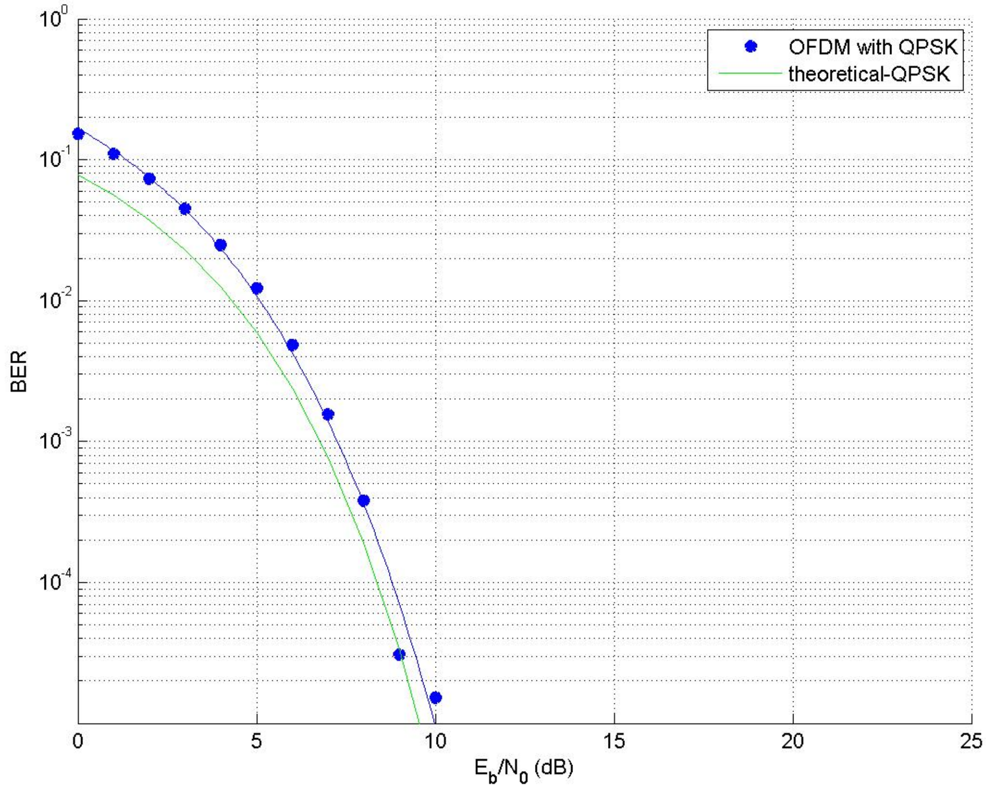

Figure 4: BER curve for OFDM with QPSK in AWGN Channel

The blue curve of the figure 4 is the one obtained via simulation of the studied OFDM system. The offset is due to the high peak to average power ratio (PAPR) of the OFDM signal. The high value of PAPR is one the major problems with OFDM. Various techniques to reduce it have been employed which are beyond the scope of the current thesis. In the current research a normalized value of the PAPR is used.

3.4.2 BER curves for OFDM over Fading (Raleigh) channel in double precision format;

The true potential of OFDM is exploited in high data rate wireless transmission. Therefore BER curve for OFDM was also generated for flat fading Raleigh channel. The analysis for the wireless channel was performed using floating point format. The diversity order was chosen to be 1; i-e a single wireless path was modeled. The wireless channel was modeled as Rayleigh fading channel using the ***rayleighchan(TS, FD)***object in MATLAB where Ts is the sampling time and Fd is the Doppler spread of the channel. The values of these two variables used in the simulation are 1/8192 and 1/4096 respectively.

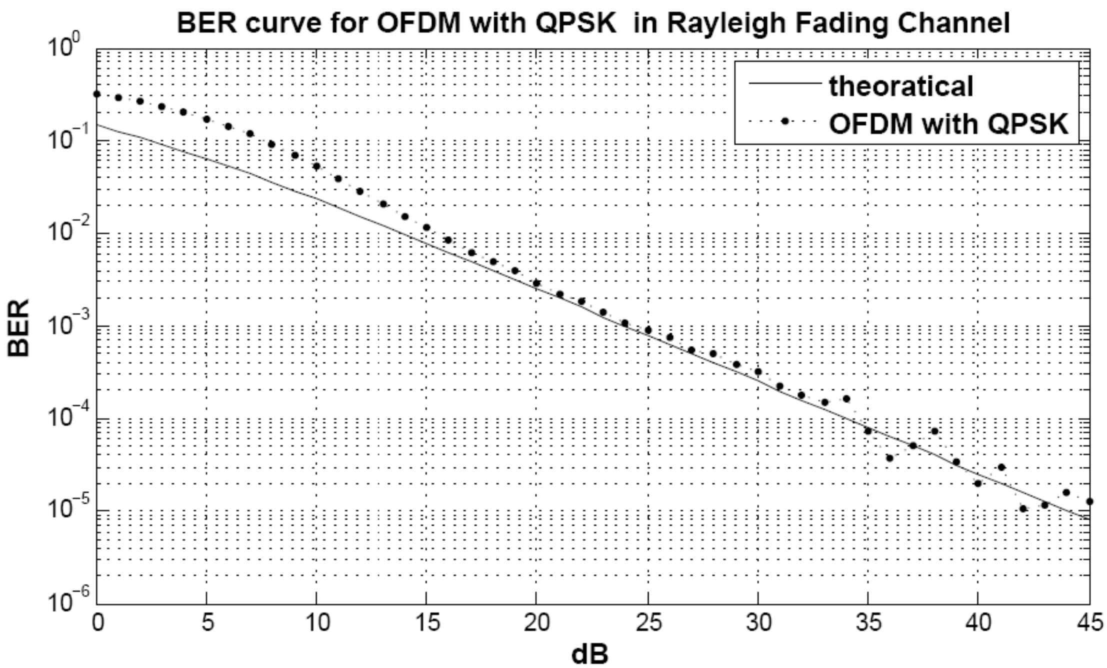

Figure 5: BER curve for OFDM with QPSK in Fading Channel

Again the results closely follow the theoretical values as shown in figure 14. Due to the fact that an OFDM signal uses multiple orthogonal frequencies hence the BER performance remains within converging limits with the theoretical curve.

3.4.3 BER curves for OFDM over AWGN channel in fixed point format;

Till now all the analysis and simulations were done using the double precision format. In double precision format the word lengths are large enough to support high precision. For example the double word used by MATLAB is 64-bit wide. Hence the dynamic range which can be accommodated within a MATLAB double word would be

$$-2^{64-1} - 1 \leq X \leq 2^{64-1} \quad \rightarrow (8)$$

In actual practice the DSP algorithms are implemented using digital hardware. The digital communication algorithms are mostly implemented using fixed point hardware. Fixed point hardware has a limited precision depending upon the word-lengths. Typical word lengths are 16, 24, 32 bits for most of the available COTS DSPs. However if the requirements are to use only 7 bits and we have a DSP with 16-bit word-length then the unused 9 bits will be consuming power and doing nothing. An alternative would be to use a field programmable gate array (FPGA), where we can tailor made the data path according to our requirements. Recent advancement in reconfigurable computing allows us to change the widths of the data path according to our requirements.

For this we investigate how many bits are sufficient to attain a desired level of BER for our OFDM system. A similar research was done in [17] mentioned in section 3.1, where quantization effects due to different analog to digital (ADC) resolutions were examined.

The OFDM modem is now converted to fixed-point format and the bit widths are changed and a BER curve is generated for that particular Q format. The main computationally expensive process involved in OFDM is the FFT/IFFT block. Only these are converted to fixed point using the '*fi*' object of the fixed point block set of MATLAB. Shown below are

3.4.4 Simulation and Results;

The figures below from figure 5 to figure 12 are the BER curves resulting from changing the bit widths from 16 bits to 2 bits respectively. The value of BER at an SNR value of 10 dB is also mentioned for each curve.

Q1.15 Format: BER at 10 dB = 0.0003147

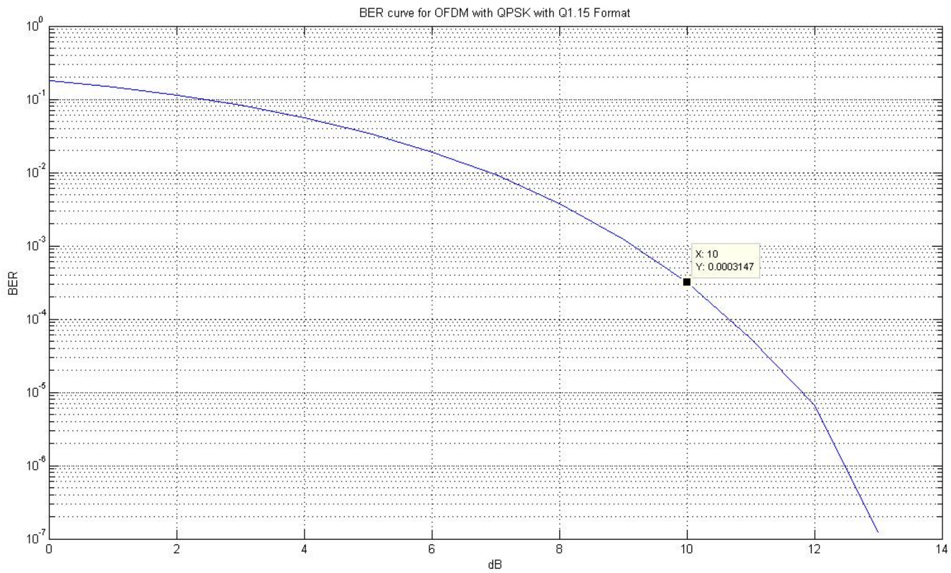

Figure 6: BER curve for OFDM with QPSK in AWGN Channel for Q1.15 Format

Q1.7 Format: BER at 10 dB = 0.0003677

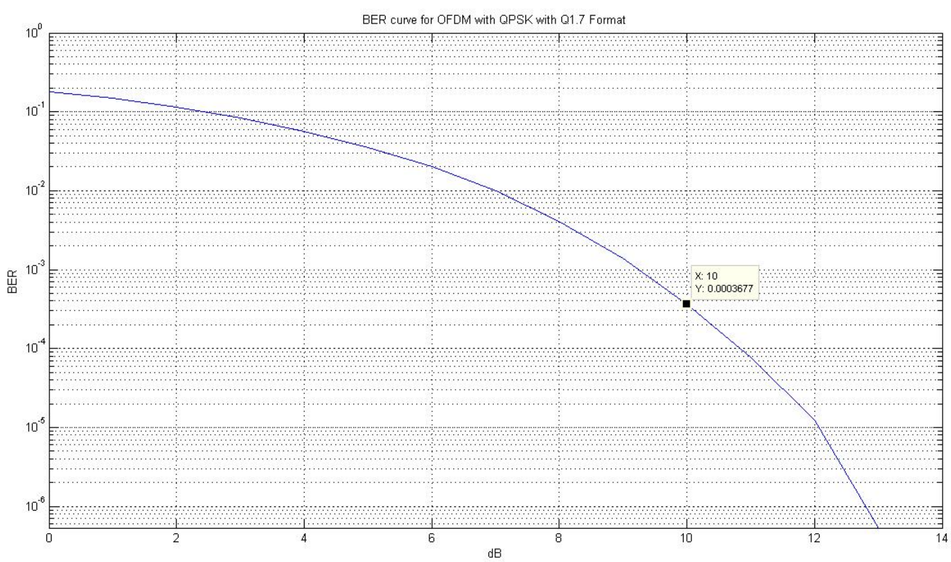

Figure 7: BER curve for OFDM with QPSK in AWGN Channel for Q1.7 Format

Q1.6 Format: BER at 10 dB = 0.0003912

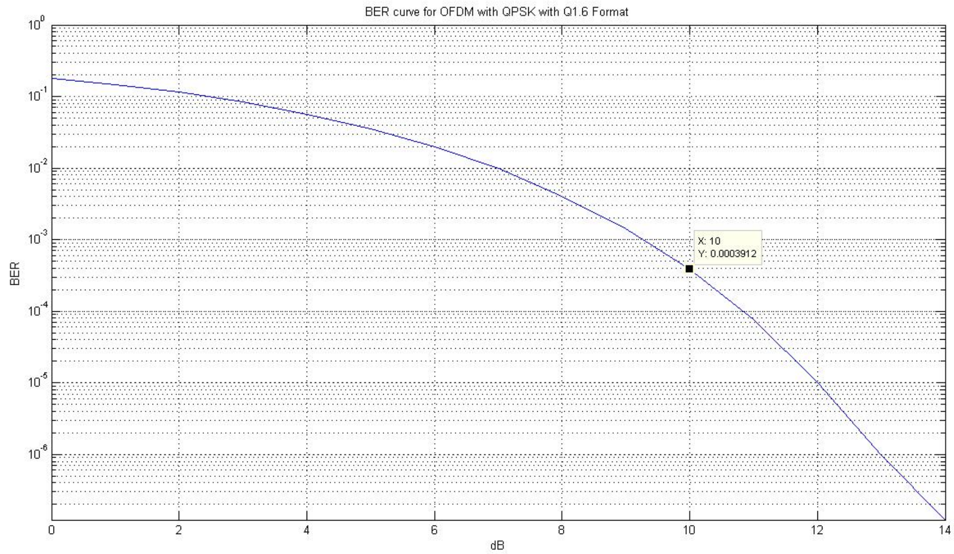

Figure 8: BER curve for OFDM with QPSK in AWGN Channel for Q1.6 Format

Q1.5 Format: BER at 10 dB = 0.003211

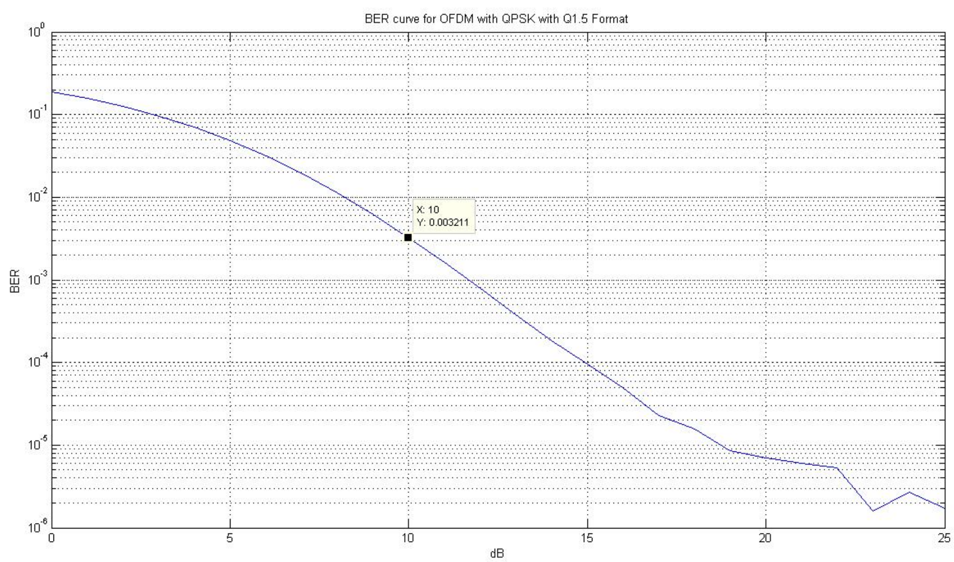

Figure 9: BER curve for OFDM with QPSK in AWGN Channel for Q1.5 Format

Q1.4 Format: BER at 10 dB = 0.03282

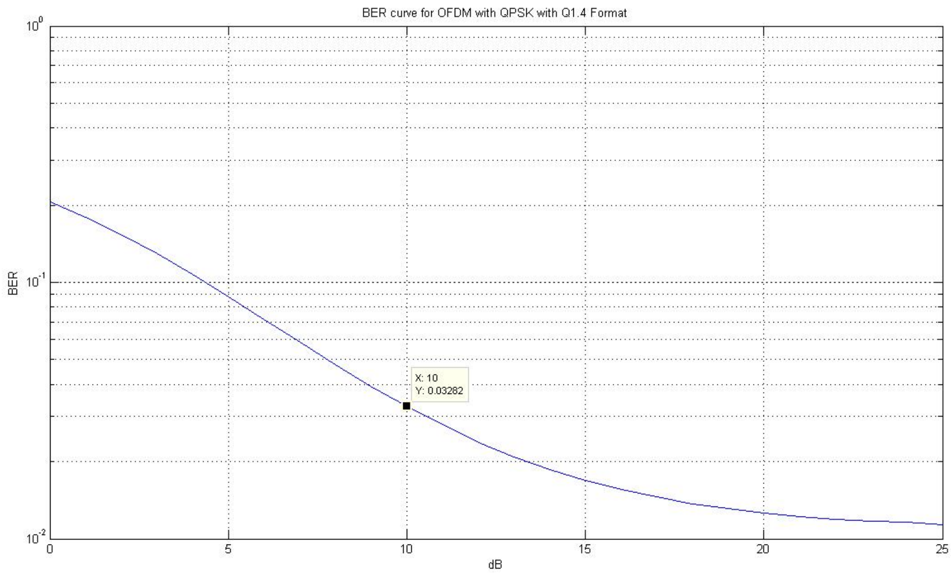

Figure 10: BER curve for OFDM with QPSK in AWGN Channel for Q1.4 Format

Q1.3 Format: BER at 10 dB = 0.2094

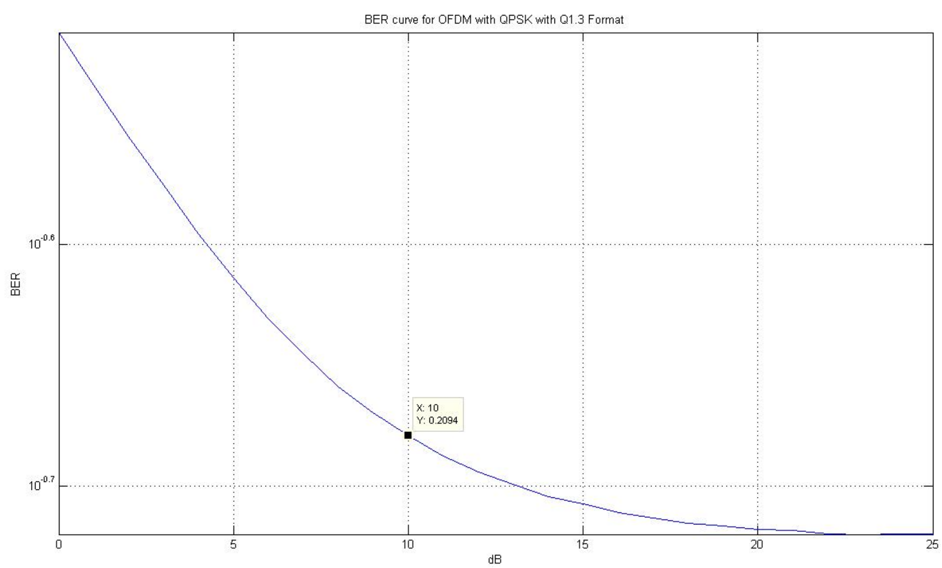

Figure 11: BER curve for OFDM with QPSK in AWGN Channel for Q1.3 Format

Q1.2 Format: BER at 10 dB = 0.2094

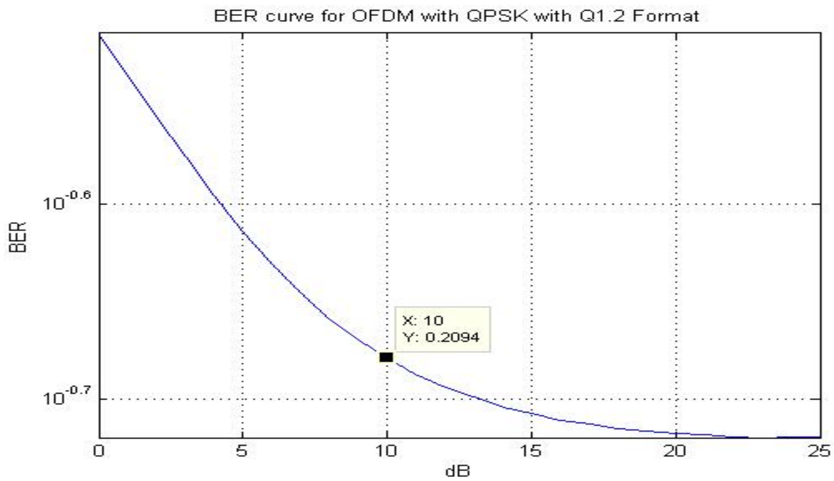

Figure 12: BER curve for OFDM with QPSK in AWGN Channel for Q1.2 Format

Q1.1 Format: BER at 10 dB = 0.4614

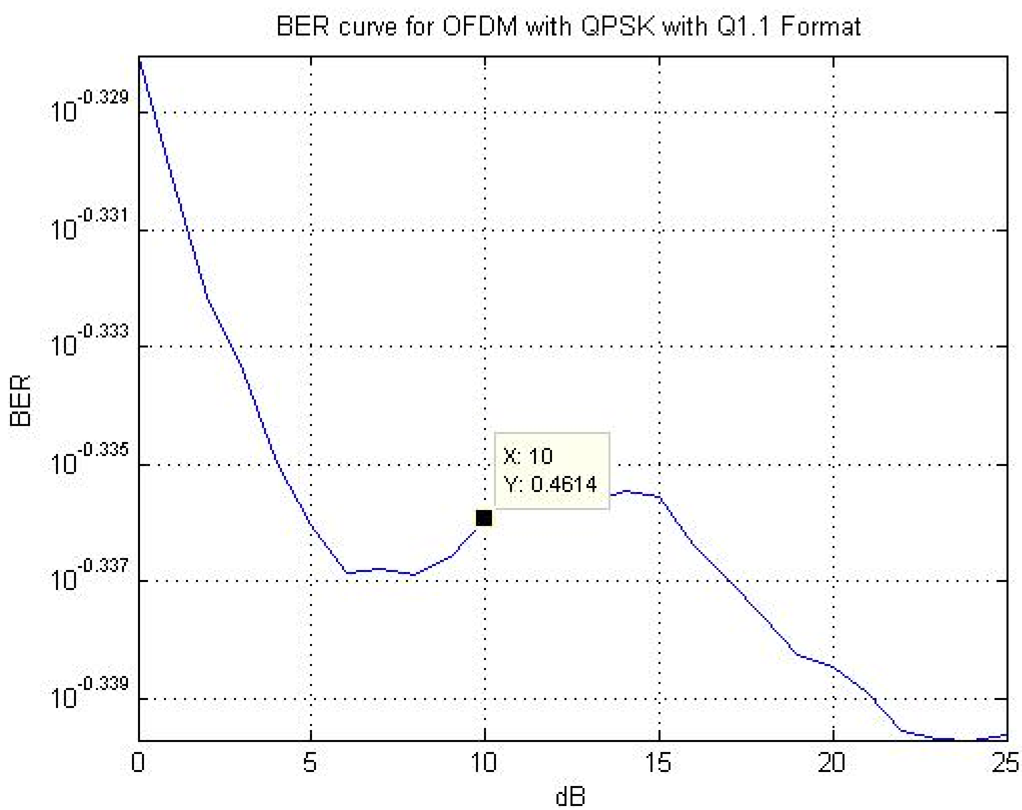

Figure 13: BER curve for OFDM with QPSK in AWGN Channel for Q1.1 Format

3.4.5 Combined BER curves for OFDM over AWGN for different Q -format;

For a better understanding of the effects of the quantization noise in terms of its effects on the BER, the above individual BER plots are combined in a single plot shown below in figure 13.

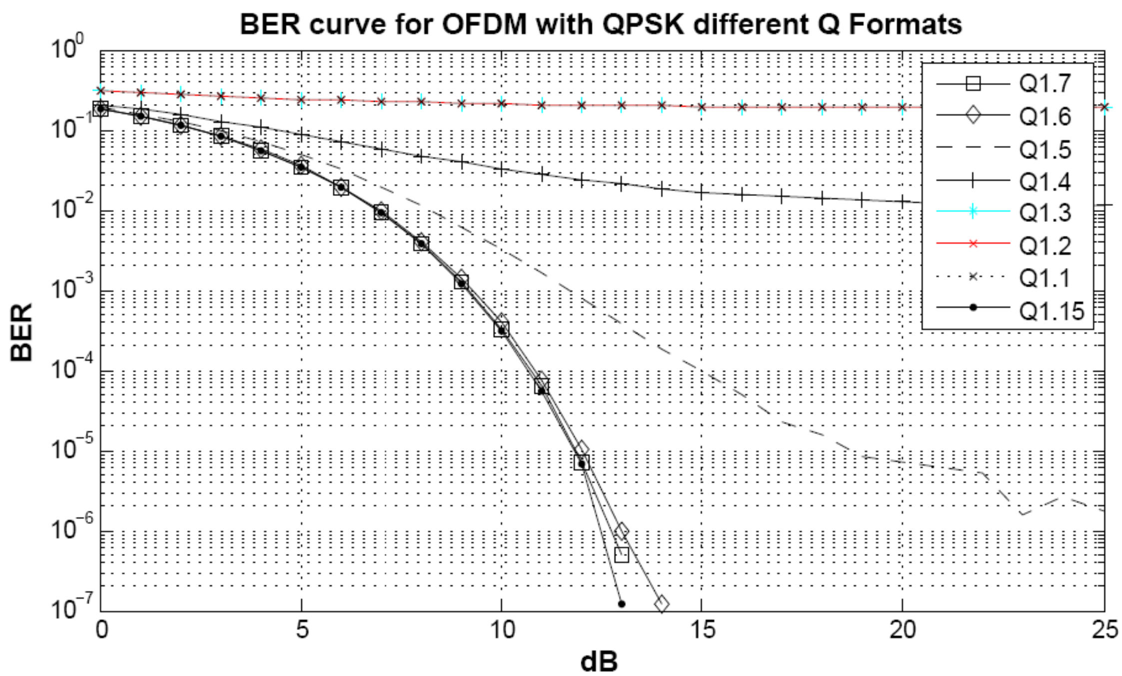

Figure 14: BER curve for OFDM with QPSK in AWGN Channel for Different Q Format

The BER increases inversely with the reduction in the number of bit used in the data path. This becomes evident when we see the BER curve for Q1.5 format which is quite high and for the test value of 10 dB the BER for Q1.5 is 0.003211 and for Q1.7 it is 0.0003677. The ratio of these values comes out be nearly 8. This means that BER for Q1.5 format turns out to be almost 8 times higher than that of Q1.7 format. The other formats follow the same trend and the BER gets on worsening with reduction in the number of bits.

The interesting results obtained from the analysis are for the formats with more number of bits. From the analysis figure 13 we can clearly see that the BER for Q1.15, Q1.7, and Q1.6 is almost identical. This means 50% saving of the bits form going from 16 bits of Q1.15 to 8 bits of Q1.7. This in turn implies a dramatic saving in power as well.

CHAPTER 4

4 Implementation of Optimum Fixed Point OFDM System in Simulink;

In this section we will present the fixed-point implementation of OFDM system for which the analysis was done in chapter 3.

Form the results obtained and the BER curves generated via the MATLAB simulations, we can easily say that the best possible word length i-e the one with least number of bits, maintaining good value of BER is 7-bits. Thus we can design an OFDM system with Q1.6 format and achieve the BER similar to the double precision format.

In [18] the author has presented search method employing genetic algorithms to obtain the optimum wordlengths for base-band modulators. The author also presents other search method and their short comings. This study helped us to do a simplified search for our case. Since the main computationally expensive task for OFDM is the FFT, we concentrated on its fixed point transformation and used a systematic approach where we have changed the wordlengths one bit at a time.

Simulink® is a powerful simulation tool which can be used to carry out fixed point conversion of an algorithm verified in double precision. The Simulink model can also be used to generate fixed point C code or hardware description language (HDL) like of the double precision MATLAB code. Since we have achieved the optimum word-length of 7-bits, now we make a Simulink model of the OFDM system and convert it to fixed point with Q1.6 format. The dialog boxes for the Simulink blocks help us to specify the word-lengths and the data types. The optimum OFDM model generated in Simulink in figure 15.

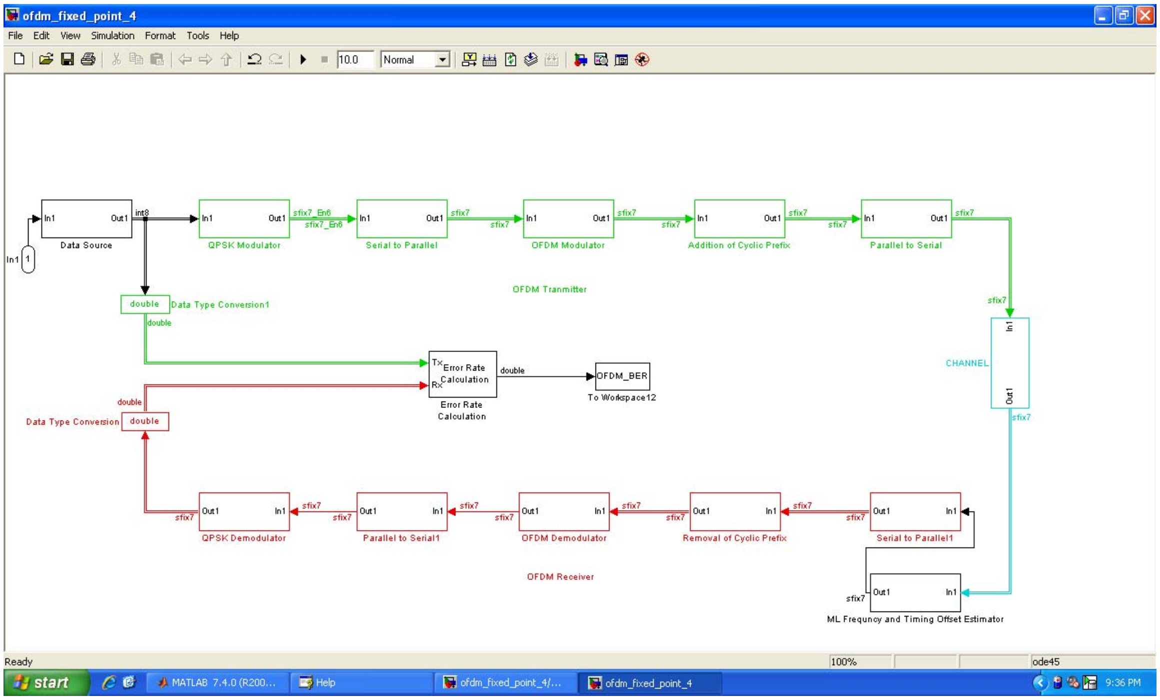

Figure 15: Optimized OFDM modem with Q1.6 format (Top Level)

The figure 15 gives a top level model of the optimized OFDM system. Now we discuss the each of the modules in this the top level model. The **"Error Rate Calculation"** block is utility block of the communication block set used to find the number of bit errors for the entire model. The BER data for the model is stored in the work space variable named **"OFDM_BER".** For running the simulation first load the file named *"initial_parameters_ofdm"*, containing the necessary variables for the initial state of the system.

Now we look at the blocks individually.

4.1 Data Source;

The print screen for this module is given below in figure 16. The dialog box for the **"random integer generator"** shows the values of the parameters. The M-ary size is selected as 4 since we will be doing QPSK modulation of the generated data for which M=4. The next field is the number of samples per frame whose value is defined in MATLAB work space by the variable named **"no_symbols"**. The next field is the output data type which is selected as **int8**.

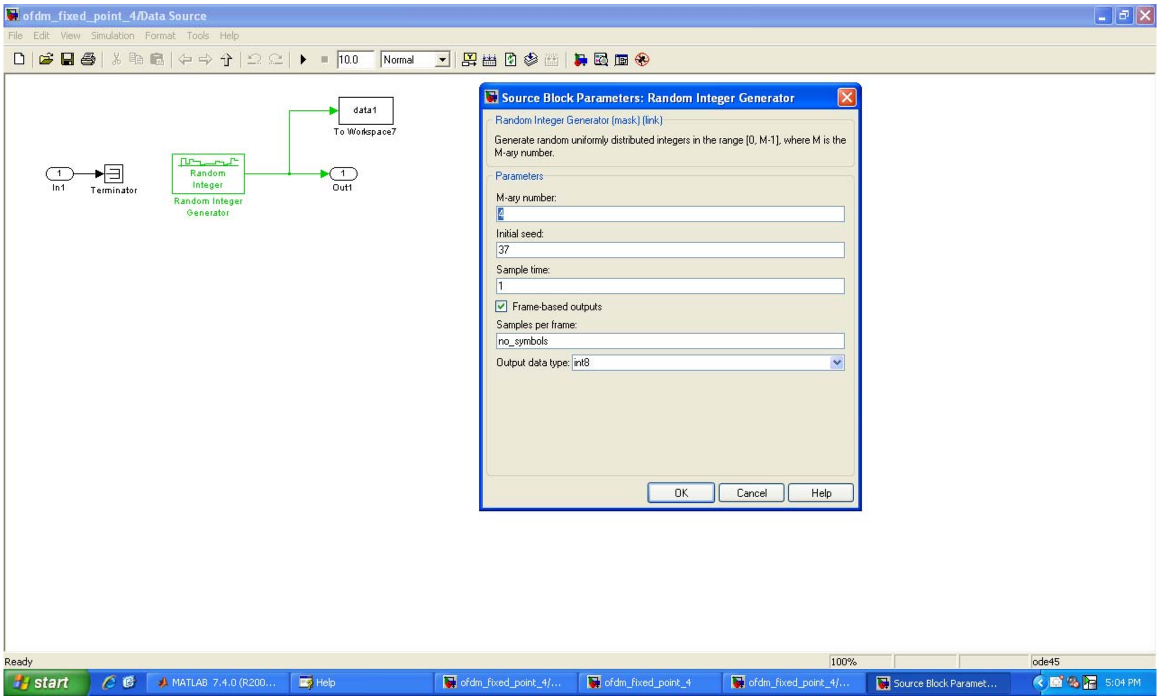

Figure 16: Data Generator Block.

4.2 QPSK Modulator;

The next block is the "QPSK modulator Baseband" shown in figure 17.

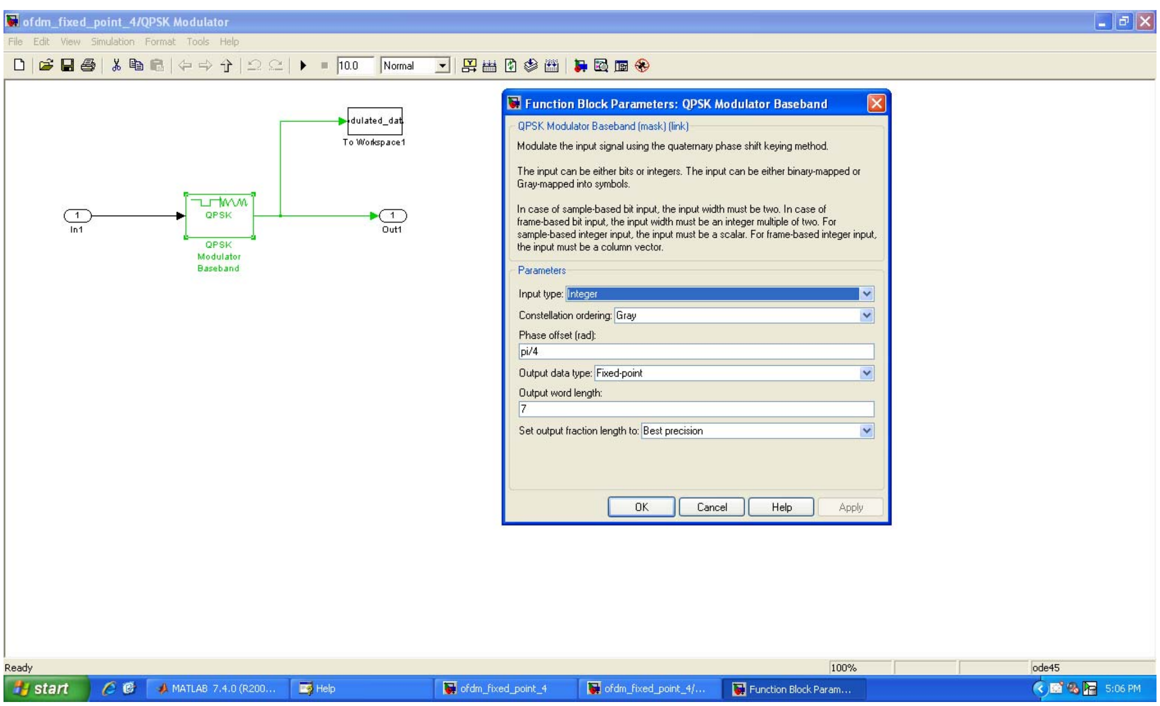

Figure 17: QPSK Modulator Block.

The dialog box has the following fields with the values. These include Input Type = Integer. Constellation ordering = Grey. Phase Offset = 45 degrees. Output data type = Fixed point. Output word length = 7 implying Q1.6 format.

4.3 Serial to Parallel Converter 1;

The **"Resahpe"** function block is used in this module. The block is shown in figure 18.

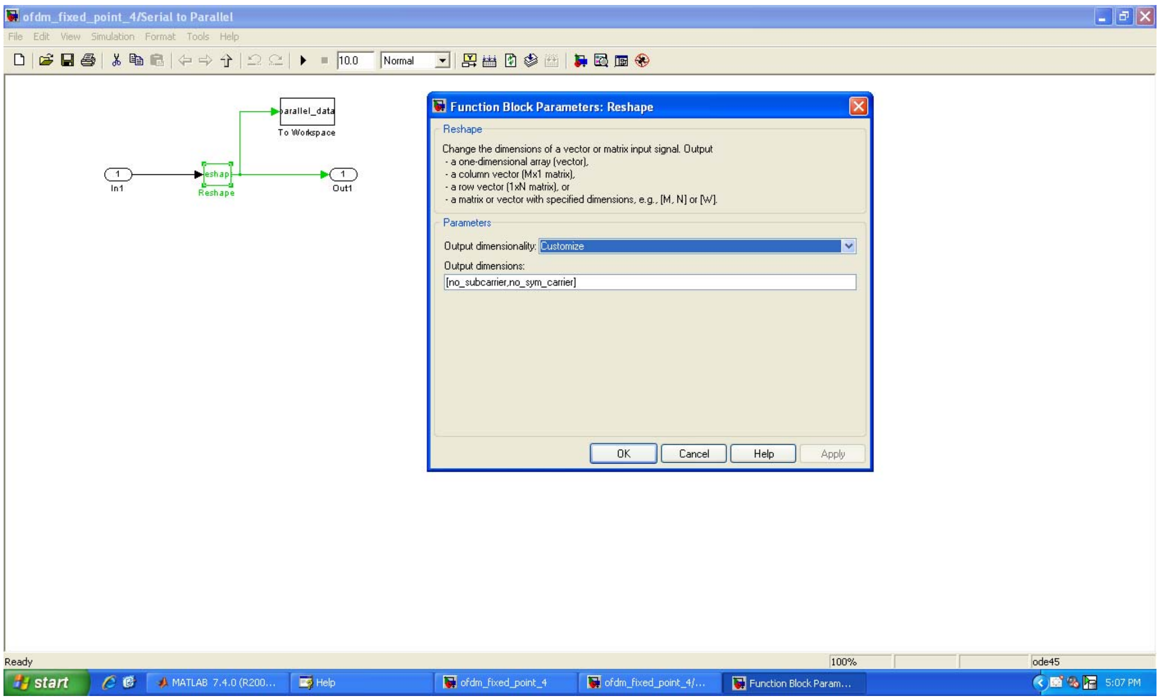

Figure 18: Serial to Parallel Block (Transmitter).

This block is used to reshape the serial stream of the base band symbols to a matrix whose dimensions are defined as [no_subcarriers,no_sym_carriers], where no_sym_carrier = FFT length selected for the system.

4.4 OFDM Modulator;

The parallel stream of QPSK symbols coming from the serial to parallel block go to the transpose block so that each symbol is given to a different sub carrier, shown in figure 19. After taking the transpose the parallel stream goes to the IFFT block. The number of inputs to the IFFT block should be integer power of 2. To overcome the PAPR issue, a normalization is done by multiplying the IFFTd data with a coefficient having value of no_sym_carrier/sqrt(no_subcarriers). A two input multiplyer is used to perform the scaling. The fixed point option in the dialog box of the IFFT block is used to implement Q1.6 format.

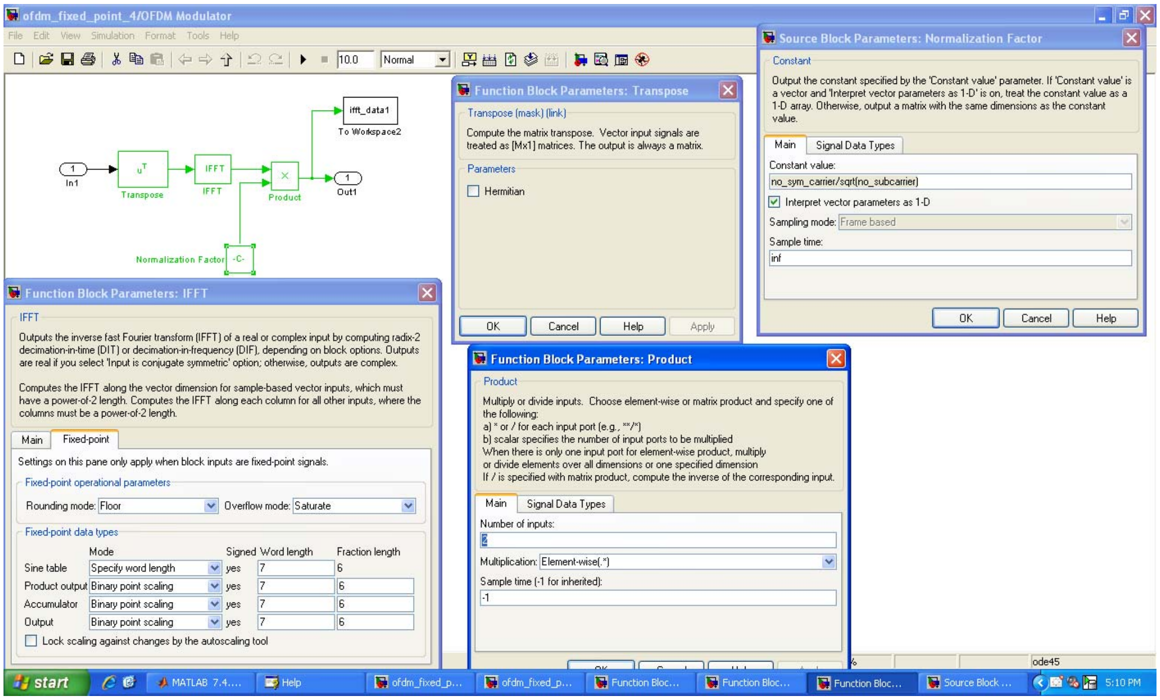

Figure 19: OFDM Modulator Block.

4.5 Cyclic Prefix Addition;

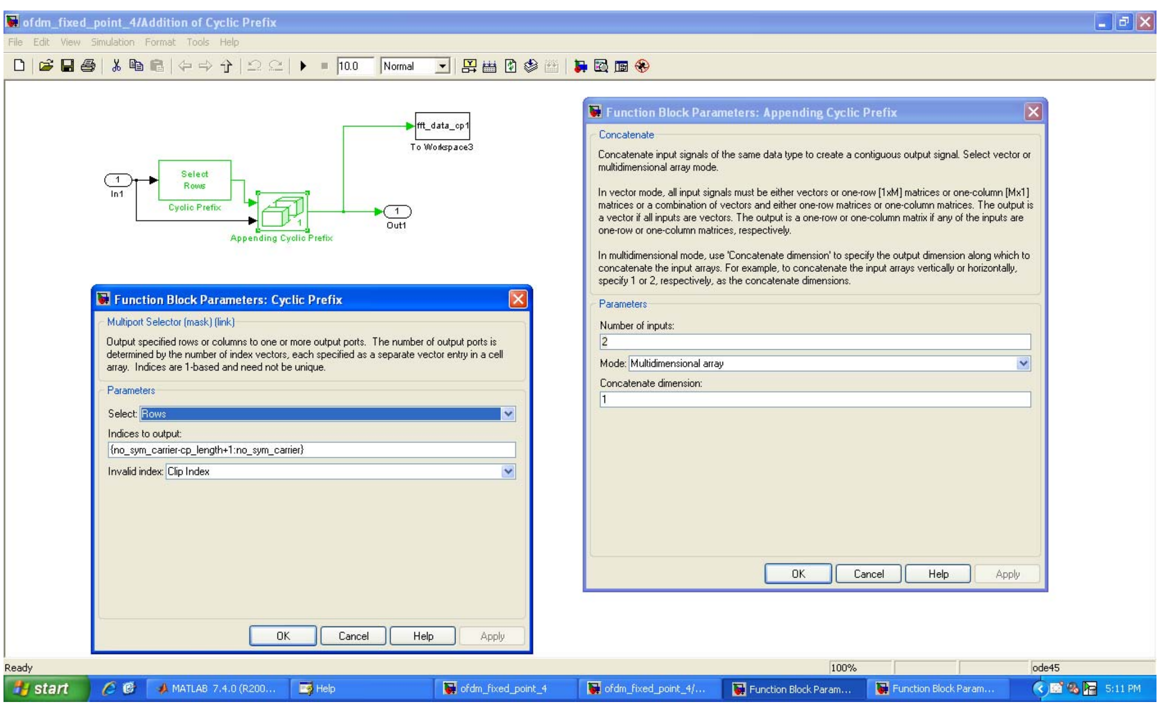

Figure 20: Cyclic prefix Addition Block.

Figure 20 shows the block for appending the cyclic prefix. A cyclic prefix of length 16 values of the IFFT ‘d data is appended. To select the last 16 values form each OFDM symbol **“row select”** block is used. The appending is done using the **“matrix concatenation”** block.

4.6 Parallel to Serial Converter 1;

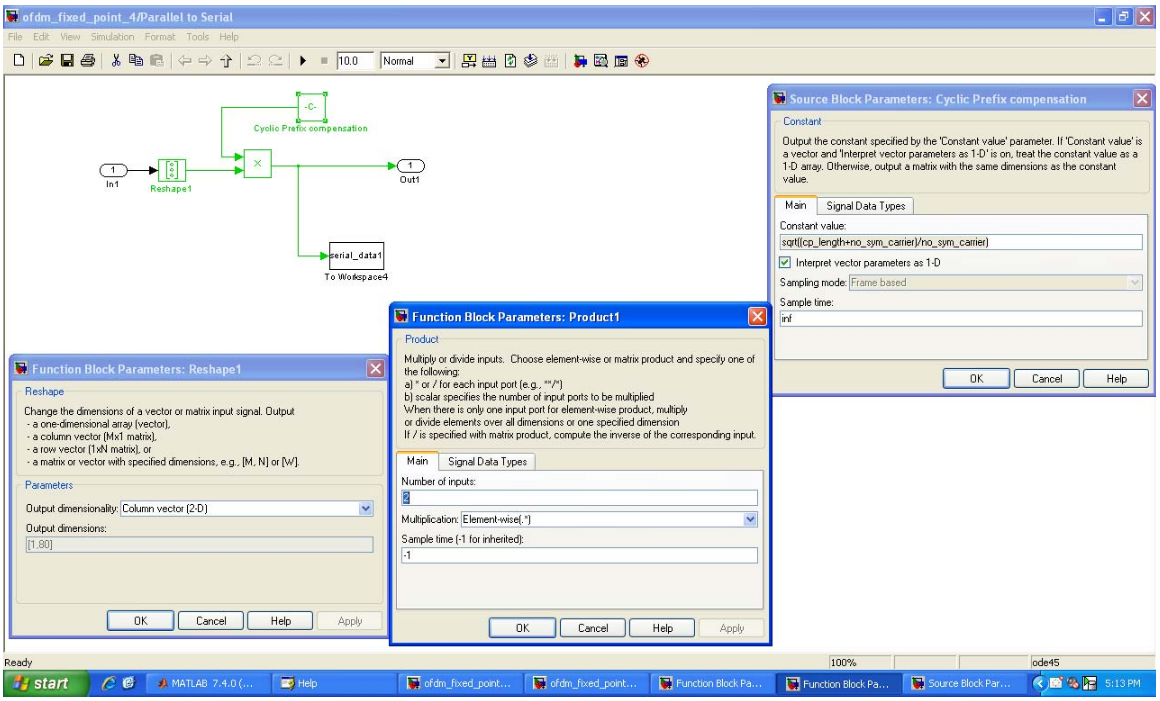

Figure 21: Parallel to Serial Converter Block (Transmitter).

Figure 21 shows the final stage of the transmitter. The parallel IFFT data is converted to serial and the excess power due to the addition of cyclic prefix is compensated by multiplying the serial data by a factor of sqrt((cp_length+no_sym_carrier)/no_sym_carrier).

4.7 Channel;

The **“AWGN channel”** communication tool box is used to model the channel conditions used in the simulations in chapter 3. Since this block accepts inputs in data types other than fixed point, data type conversion block is used. The figure 22 shows the channel model. EbNo is the SNR variable getting value of the noise power form the workspace.

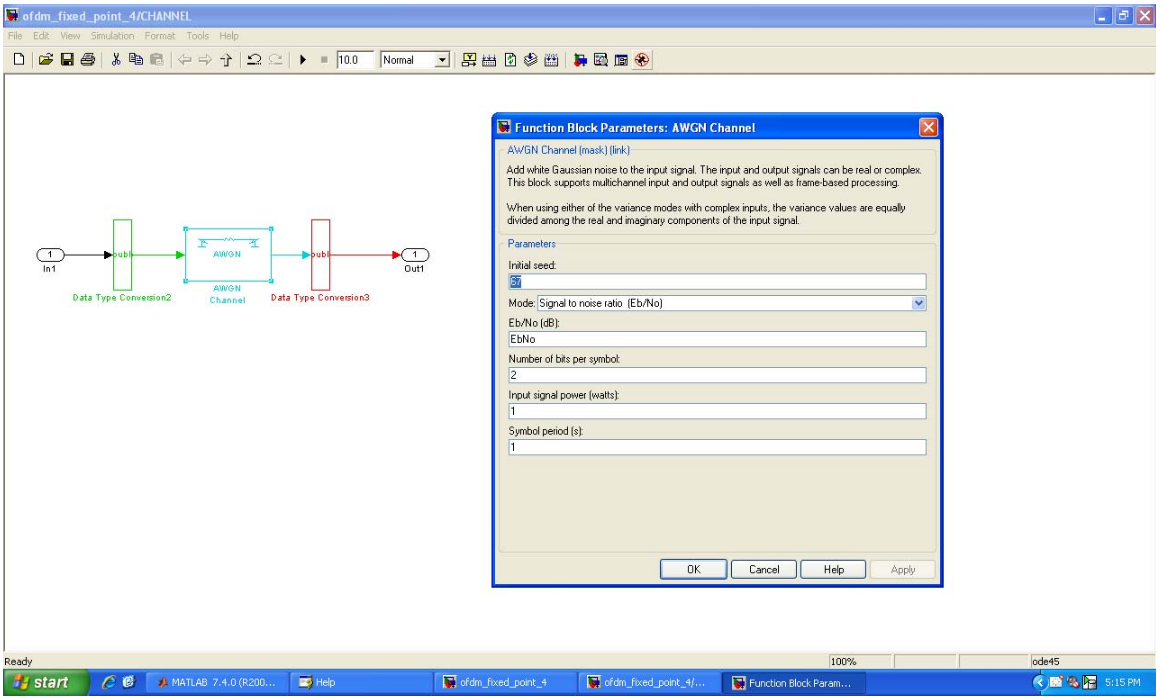

Figure 22: Channel Block

4.8 Serial to Parallel Converter 2;

For performing the demodulation on the different subcarriers the serial data is converted to parallel as shown in figure 23 using the reshape block.

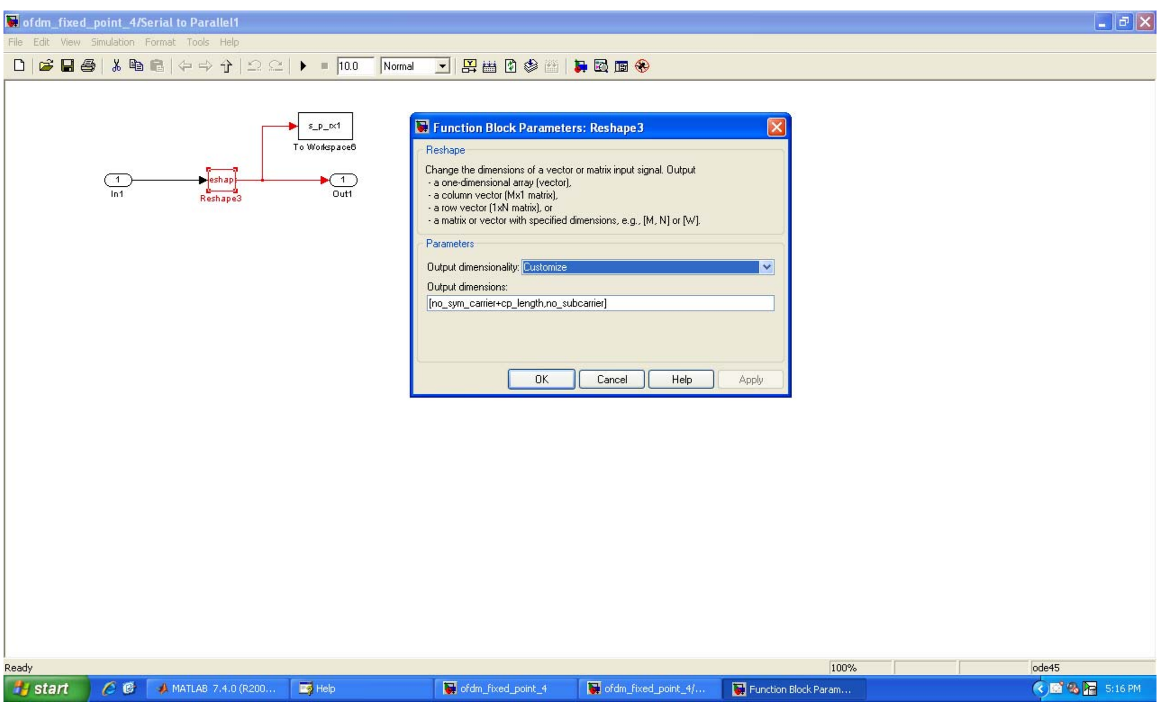

Figure 23: Serial to Parallel Conversion Block (Receiver)

4.9 ML Frequency and Timing Offset Estimator;

The ML estimator Simulink model is shown below in figure 24. The estimator is explained in detail in chapter 5.

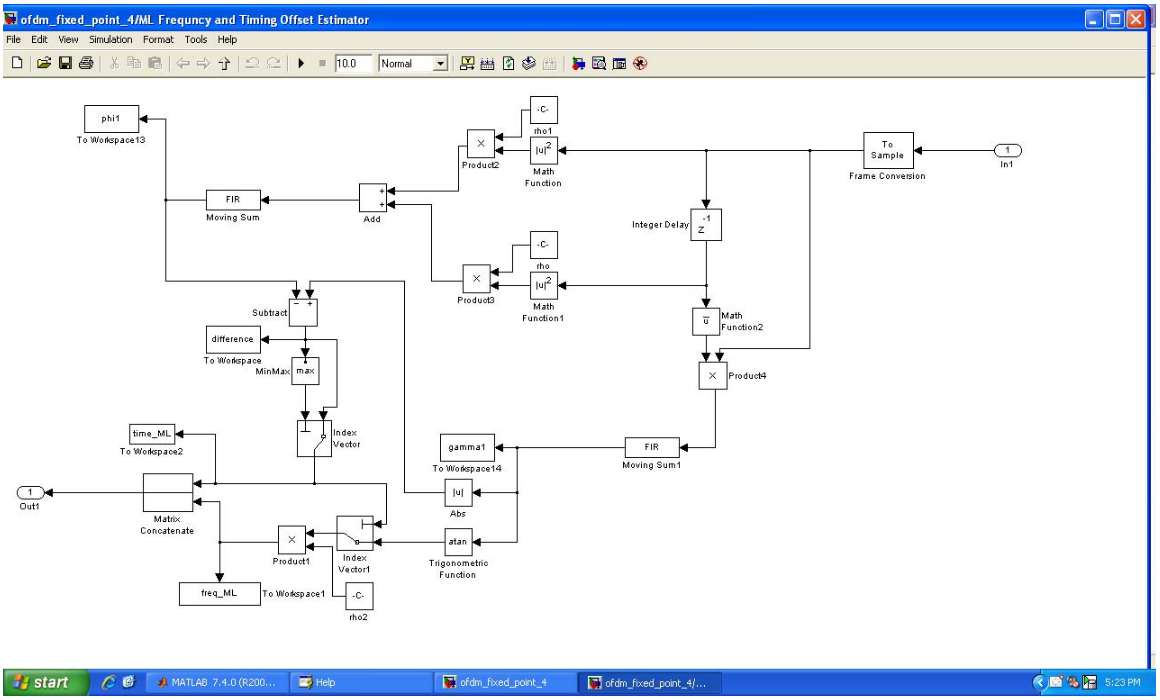

Figure 24: ML Offset Estimator.

4.10 Removal of Cyclic Prefix;

Figure 25 shows the cyclic prefix removal block. For removing the cyclic prefix, "row select" block is used where all the rows except the ones containing the cyclic prefix are selected.

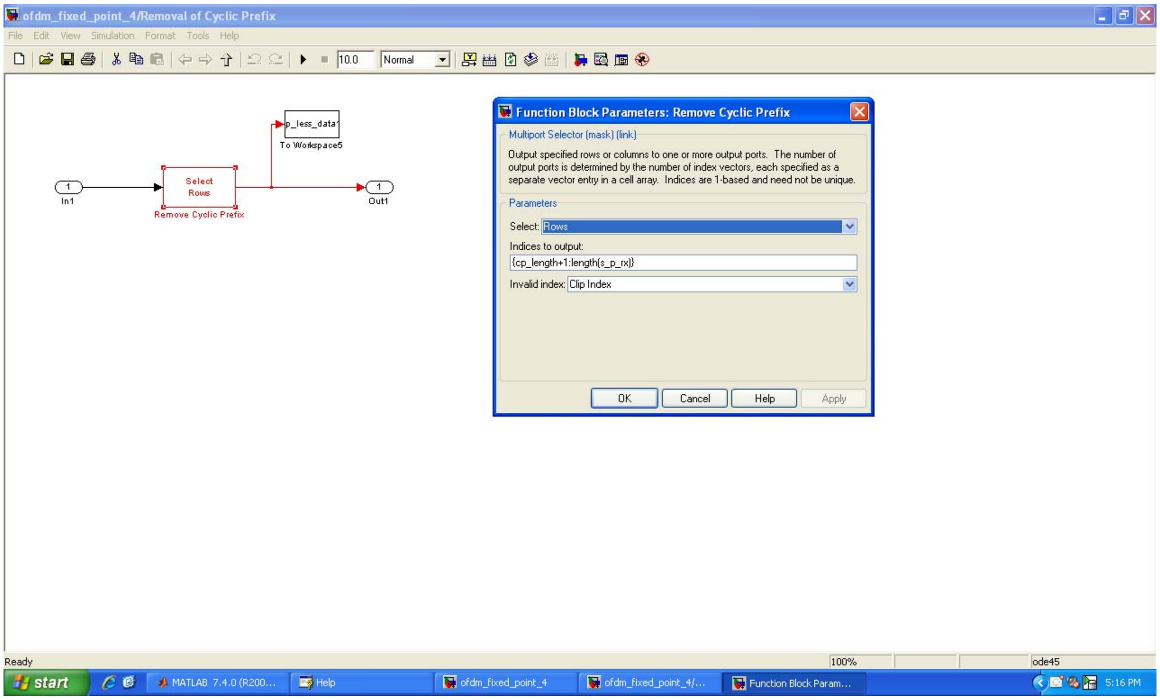

Figure 25: Removal of Cyclic Prefix Block.

4.11 OFDM Demodulator;

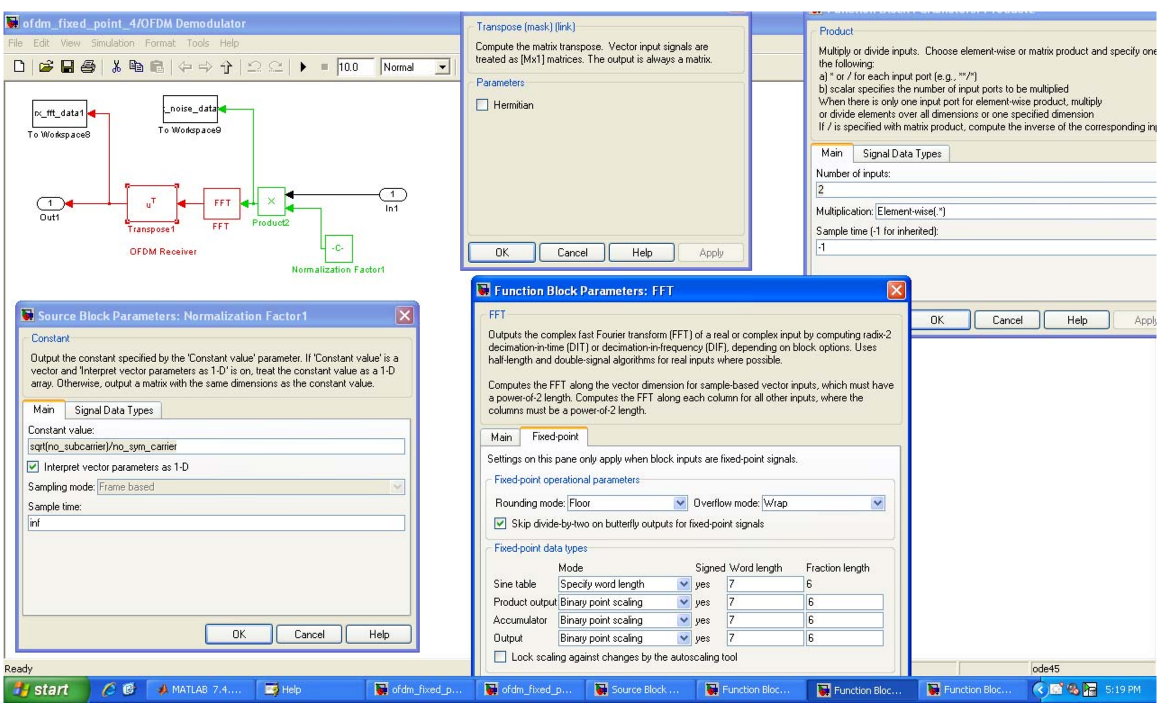

Figure 26: OFDM Demodulator Block.

In the OFDM demodulator shown in the figure 26, all the steps performed for the OFDM modulation are reversed.

4.12 Parallel to Serial Converter 2;

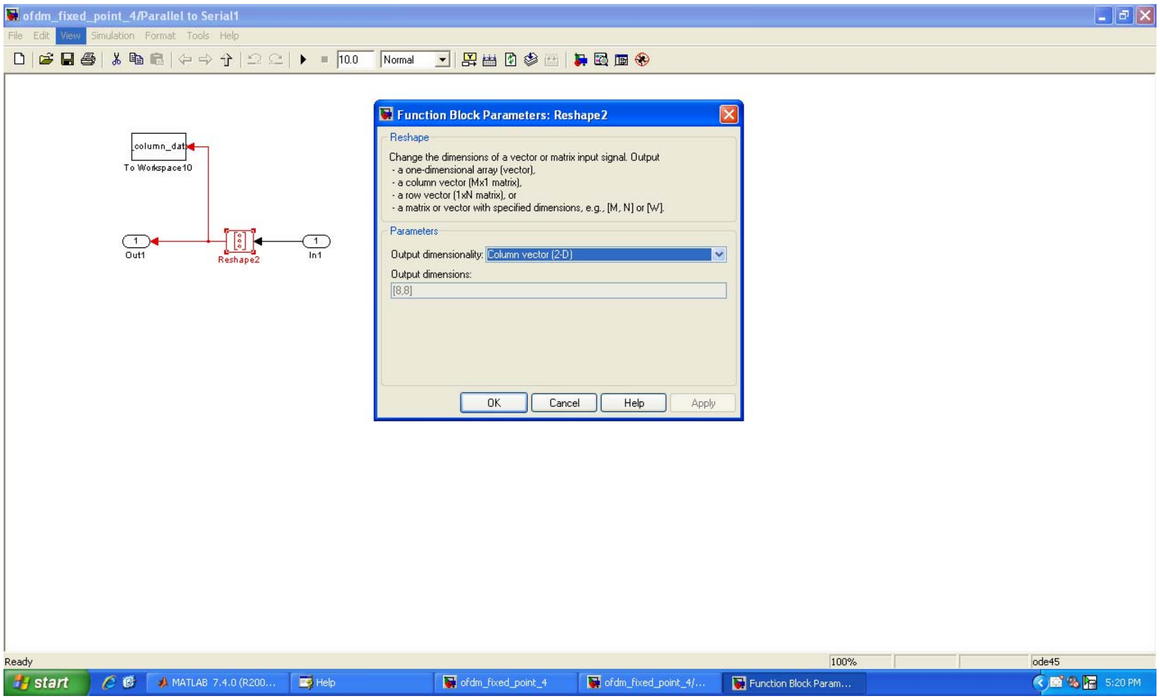

Figure 27: Parallel to Serial Conversion (Receiver).

The FFT ‘d data is then converted to serial for performing base band demodulation, shown in figure 27.

4.13 QPSK Demodulator;

The final step is shown in figure 28. The “QPSK demodulator baseband” is used to perform the demodulation. The output of the demodulator goes to the data type conversion block for calculating the errors.

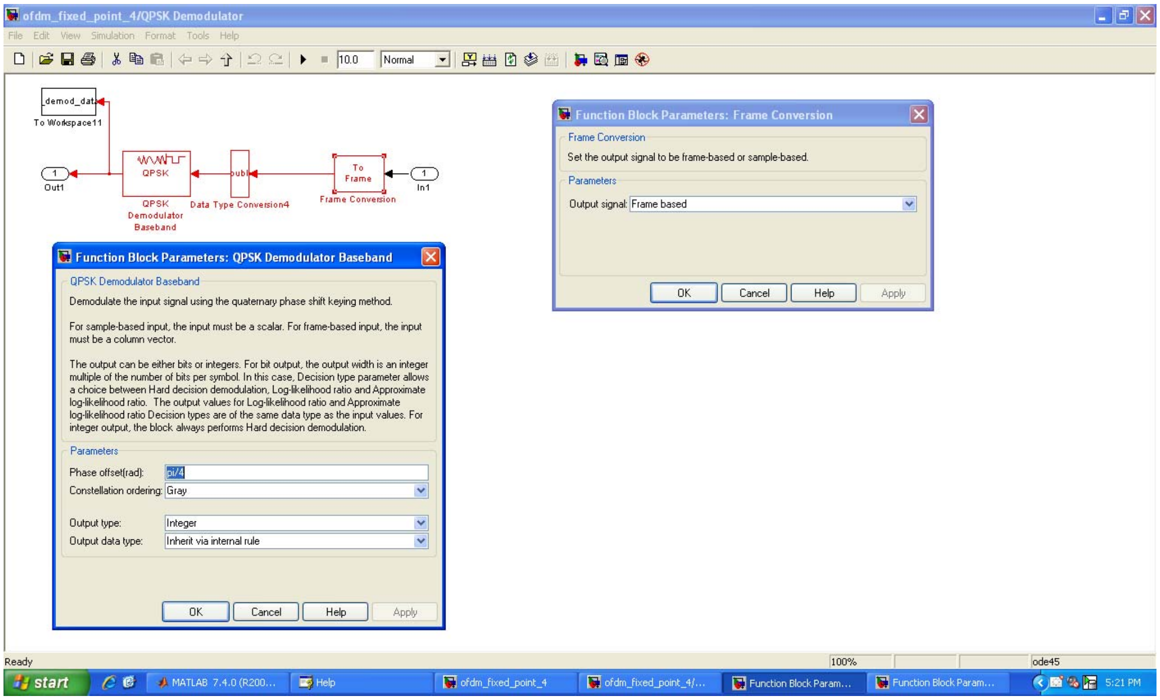

Figure 28: QPSK Demodulator Block.

CHAPTER 5

5. Timing and Frequency offset Estimation;

This chapter is dedicated to the synchronizer mentioned briefly in section 3.3.3 and the simulink model given in section 4.9. Proper synchronization helps reduce the errors induced by the channel impairments.

5.1 Introduction;

Synchronization is on of the most important element of any digital communication system. In a digital communication system one is not concerned with the exact reproduction of the transmitted signal, rather interested to know whether a 1 or a 0 was sent. So the thing which is of concern for us is the sampling time instant T_s and the carrier frequency f_c. If the sampling time instant is not determined correctly then the bit/ symbol duration is incorrectly determined and leads to ISI among other symbols. Thus the sampling instant (sampling time), which helps in determining the starting of a message symbol needs to be determined accurately.

A base band signal is modulated by a carrier signal with a centre frequency of f_c pass band signal. The carrier frequency f_c also needs to be accurately determined since the received signal is heterodyned with a sinusoid of this frequency at the receiver to convert it to base band at the receiver and do further processing. If the carrier frequency is not estimated correctly then the demodulation will be effected and bit errors will increase.

5.2 Causes for Timing and Frequency Offsets;

Timing and frequency offsets arise due to the non ideal channel conditions and the non-linearity of the electronic devices in the transmitter and receiver. For example, carrier frequency offsets, which are caused by the inherent instabilities of the transmitter and receiver carrier frequency oscillators [19]. For instance, if the sampling clock specification is 10 parts per million (ppm) and the sampling frequency is 5 MHz, and then the symbol timing has a drift of about 50 samples per second. Thus, sampling clock synchronization is also an important issue [20]. These offsets inhibit the correct detection of the transmitted signals and result in an increase in the BER. Hence these offsets must be estimated and should be countered.

5.3 Synchronization Requirements for OFDM;

OFDM, like any other digital communication system, requires synchronization. However, OFDM as a multi-carrier system has a different structure than a single-carrier system and so have different requirements and different resources. For example in OFDM, one can tolerate larger errors in estimating the start of a symbol than in a single-carrier system. This is due to OFDM's longer symbol period and its cyclic prefix. On the other hand, frequency synchronization in OFDM must be tighter than that in single-carrier systems, due to the narrowness of the OFDM sub-carriers. In terms of resources, OFDM has a structure that is not available in single-carrier systems that is useful for synchronization. For example, most OFDM systems have a cyclic prefix that, as we will see, can be used for synchronization. The cyclic prefix can act as pilot data [21]. Often, an OFDM symbol itself is used as pilot data. In this case, the structure of the OFDM symbol can be exploited for time and frequency offset estimation. The choice of pilots versus no-pilots depends on many parameters: the operating SNR, the size of the cyclic prefix, coherent versus differential modulation. Whether to insert pilot data or use OFDM symbols as pilot data often depends on how much overhead the system can tolerate.

5.3.1 Timing Offset Estimation;

The goal in timing offset estimation is to find a place to start the N-point FFT for demodulating an OFDM symbol [22].

5.3.2 Frequency Offset Estimation;

The purpose of frequency offset estimation is

- fine frequency offset estimation: estimating the center frequencies of each sub-channel, ε;
- Coarse frequency offset estimation: estimating the tone numbering index, k.

5.4 Synchronization Techniques for OFDM;

There are a number of synchronization techniques for estimating the timing and frequency offsets is OFDM. Some techniques make use of pilot tones for estimating the offsets. Some make use of the OFDM symbol itself as the pilot. Some techniques estimate the timing offset only while others only the frequency offsets.

The presence of the cyclic prefix in an OFDM symbol gives us the opportunity to exploit its redundancy. This redundancy can be used to estimate the timing and frequency offsets. In this thesis a joint maximum likelihood approach using the cyclic prefix as given in [1] is used to estimate the timing and fine frequency offset. Since the objective of the current research is the reduction of resources and the power hence the ML approach is used which requires no additional pilot symbols which add to the overhead involved.

5.5 Joint Maximum Likelihood Estimation of Timing and Frequency Offsets for OFDM;

In this section the details of the ML estimation algorithm are presented. The algorithm is simulated in MATLAB and applied to the OFDM transceiver used in the study. The simulink model presented in section 4.9 is developed using the algorithm discussed below.

There are two problems in OFDM reception. First we need to know the unknown symbol arrival time for the OFDM symbol. The second is mismatch of the oscillators in the transmitter and the receiver, which introduces a frequency offset. The symbol arrival time and the frequency offset can be estimated by maximizing the average log likelihood function. For this purpose pilot symbols can be used as explained by [23] and [24]. The algorithm being used in our study exploits the redundancy offered by the cyclic prefix, thus reducing the need for using pilots.

5.5.1 Modeling of Estimator Parameters;

An OFDM system having an 'N' sample long OFDM symbol is considered. In order to avoid ISI, last 'L' samples of the OFDM symbol are copied and appended to the N sample long OFDM symbol as a preamble. This L sample long preamble is called the cyclic prefix. Thus the effective length of the transmitted OFDM symbols is 'N+L' samples. Let $s(t)$ be the transmitted OFDM symbol, then the received signal in AWGN is expressed as

$$r(k) = s(k-\theta)e^{j2\pi\varepsilon k/N} + n(k) \rightarrow (9)$$

Where the timing and frequency offsets are modeled as θ and ε respectively.

5.5.2 *Description of the ML Estimator;*

For the ML estimation 2N+L consecutive samples of the received signal r(k) are observed. The observation interval is shown in figure 29. There is one complete N+L OFDM symbol in this observation interval, the starting of which is unknown due to the timing offset θ.

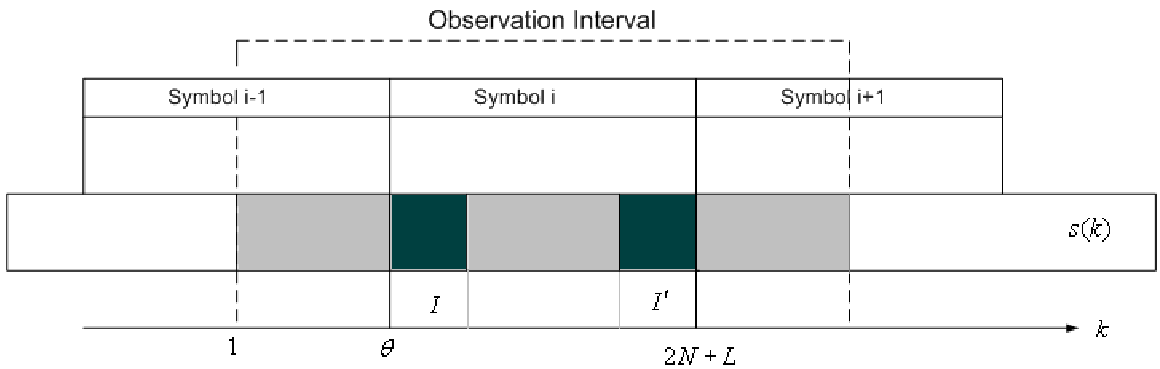

Figure 29: Observation interval for the Estimator.

Two sets of index, I and I` are defined, such that

$$I \triangleq \{\theta, \ldots, \theta + L - 1\} \rightarrow (10)$$
$$I' \triangleq \{\theta + N, \ldots, \theta + N + L - 1\} \rightarrow (11)$$

Set I` contains the indices of the data which is copied as the cyclic prefix, while I contains the indices of the elements of the cyclic prefix.

The elements of the these two sets are pair wise correlated for the observed 2N+L samples of the received signal r(k). This is mathematically expressed as

$$\forall k \in I : E\{r(k)r^*(k+m)\} = \begin{cases} \sigma_s{}^2 + \sigma_n{}^2, m = 0 \\ \sigma_s{}^2 e^{-j2\pi\varepsilon}, m = N \rightarrow (12) \\ 0, otherwise \end{cases}$$

The log-likelihood function of θ and ε is the logarithm of the pdf of the 2N+L samples of the received signal r(k), i-e

$$\Lambda(\theta, \varepsilon) = \log f(r \mid \theta, \varepsilon) \rightarrow (13)$$

The ML estimation requires us to maximize the argument of the log-likelihood function. Assuming that the received signal r(k) is jointly Gaussian, we can express the likelihood function as follows

$$\Lambda(\theta,\varepsilon) = |\gamma(\theta)|\cos(2\pi\varepsilon + \angle\gamma(\theta)) - \rho\Phi(\theta) \rightarrow (14)$$

Where $\angle$ denotes the argument of a complex number

The term $\gamma(\theta)$ is a weighting factor whose value depends upon the frequency offset. The weighting factor is a sum of L consecutive correlations between pair of samples spaced N samples apart, and is expressed as

$$\gamma(m) \triangleq \sum_{k=m}^{m+L-1} r(k)r^*(k+N) \rightarrow (15)$$

The term $\Phi(\theta)$ is an energy term and is independent of the frequency offset and is written as

$$\Phi(m) \triangleq \frac{1}{2}\sum_{k=m}^{m+L-1} |r(k)|^2 + |r(k+N)|^2 \rightarrow (16)$$

And ρ is the magnitude of the correlation coefficient between r(k) and r(k+N) and is defined as

$$\rho = \frac{SNR}{SNR+1} \rightarrow (17)$$

The maximization process can be done in two steps, first we maximize the log-likelihood function with respect to the frequency offset ε and then with respect to the timing offset θ.

In order to maximize with respect to the frequency offset the cosine term in the likelihood function needs to be 1 which results in the frequency offset estimate to be equal to

$$\hat{\varepsilon}_{ML}(\theta) = -\frac{1}{2\pi}\angle\gamma(\theta) + n \rightarrow (18)$$

For the ML estimate of frequency the log-likelihood function becomes

$$\Lambda(\theta, \hat{\varepsilon}_{ML}(\theta)) = |\gamma(\theta)| - \rho\Phi(\theta) \rightarrow (19)$$

Now we can write the joint ML estimates of θ and ε as

$$\hat{\theta}_{ML} = \arg\max_{\theta}\{|\gamma(\theta)| - \rho\Phi(\theta)\} \rightarrow (20)$$

$$\hat{\varepsilon}_{ML}(\theta) = -\frac{1}{2\pi}\angle\gamma(\hat{\theta}_{ML}) \rightarrow (21)$$

Only two quantities affect the log-likelihood function (and thus the performance of the estimator): the number of samples in the cyclic prefix and the correlation coefficient given by the SNR. The former is known at the receiver, and the latter can be fixed. Basically, the quantity provides the estimates of θ and ε. Its magnitude, which is compensated by an energy term, peaks at time instant $\hat{\theta}_{ML}$,

5.5.3 Structure of the ML Estimator;

The structure of the ML estimator is given below in figure 30. The estimator has an open loop or feed forward structure. The quantity of $\gamma(\theta)$ is calculated online. The block diagram is used to implement the estimator in Simulink in section 4.9.

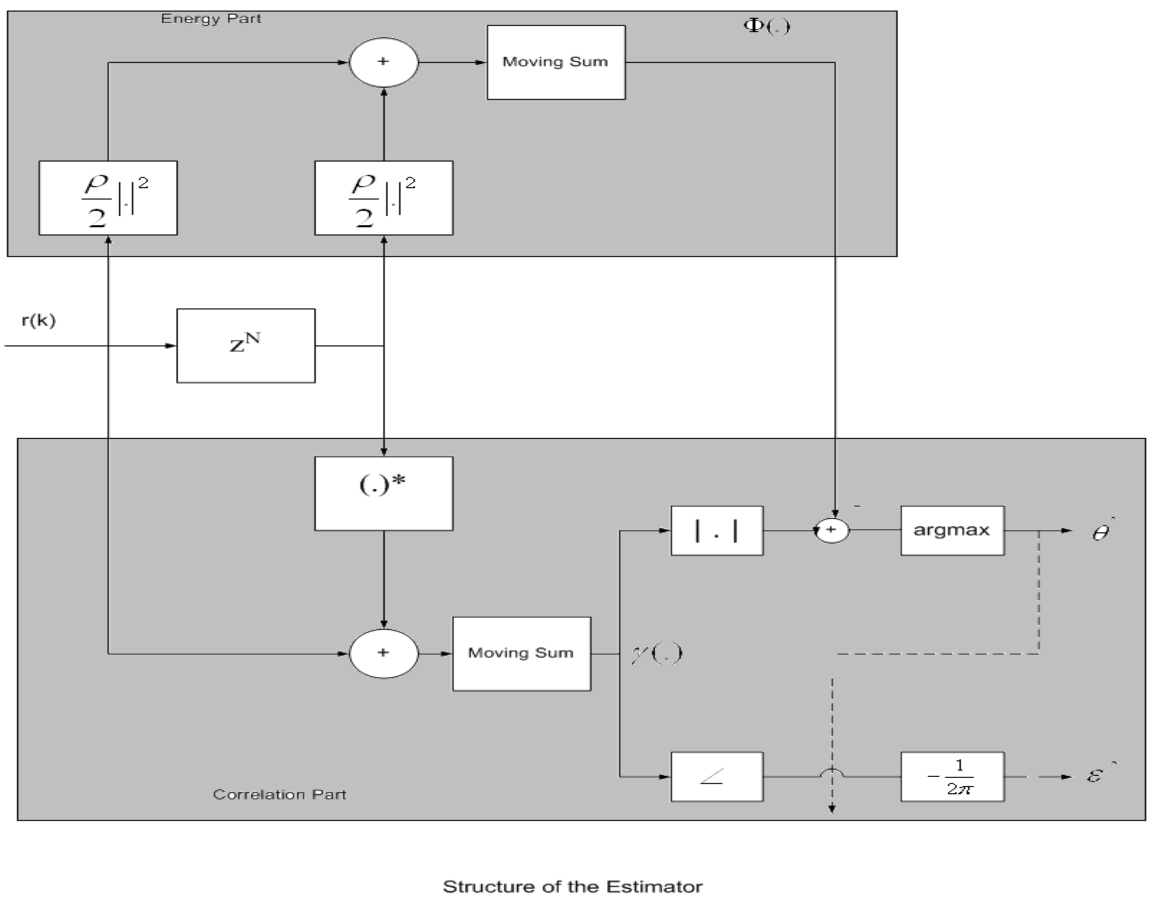

Figure 30: Block diagram for the ML Estimator

5.5.4 Advantages of the ML Estimator;

1. The estimator has low complexity.
2. The estimator is open loop or feed forward hence remains stable.
3. Exploits the redundancy of the cyclic prefix and thus does not need to employ pilots.

5.5.5 Simulation and Results;

The ML estimator was applied to the OFDM receiver simulation. The following parameters for the OFDM transmitter used were

Bit Rate	8192 bps
Symbol Rate	4096 symps
Base Band Modulation	Grey encoded QPSK,
FFT symbol size	64 symbols
No of sub-carriers	64
Cyclic Prefix Length	¼ of FFT Length=16 symblos
Timing Offset	4 samples(symbols)
Frequency Offset	0.25 rad
Offset Estimator used	Joint ML Timing and Frequency Estimation

Table 1: Simulation Parameters for ML Estimation

A timing offset of 4 samples was introduced by appending 4 zeros to the received OFDM vector. Note that the timing offset should be less than or equal to the length of the cyclic prefix.

A frequency offset of 0.25 as given in the paper used for implementation was used. This result in a rotation of the scatter plot of the OFDM received signal, as shown in the figure 31.

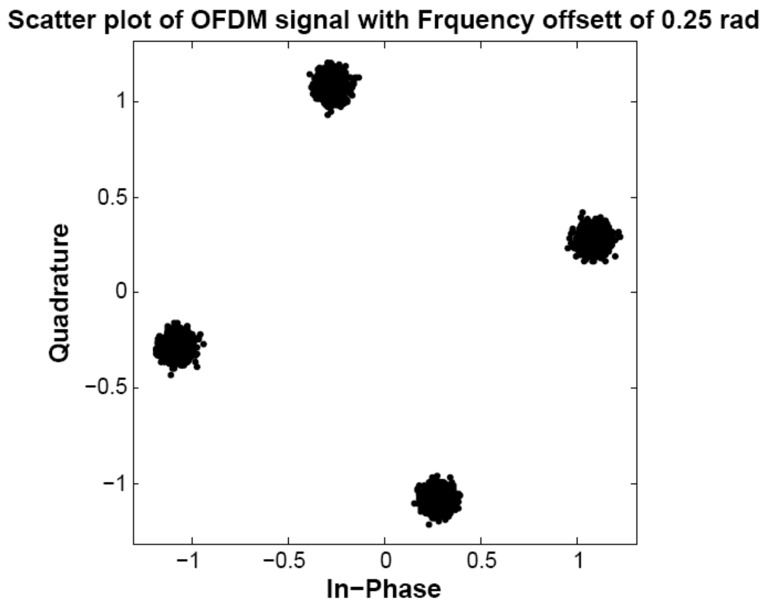

Figure 31: Scatter Plot of Received OFDM Signal with Frequency Offset

The plots below in figure 32 give the timing offset estimate (above) and the frequency offset estimate (below). The peeks for the plot 1 give the estimate of the timing offset, which is the value at the x-axis for the peek in plot 1 describes the starting instant of the OFDM symbol. While the values on the y-axis of plot 2 corresponding to the peak in plot 1 gives the estimate of the frequency estimate.

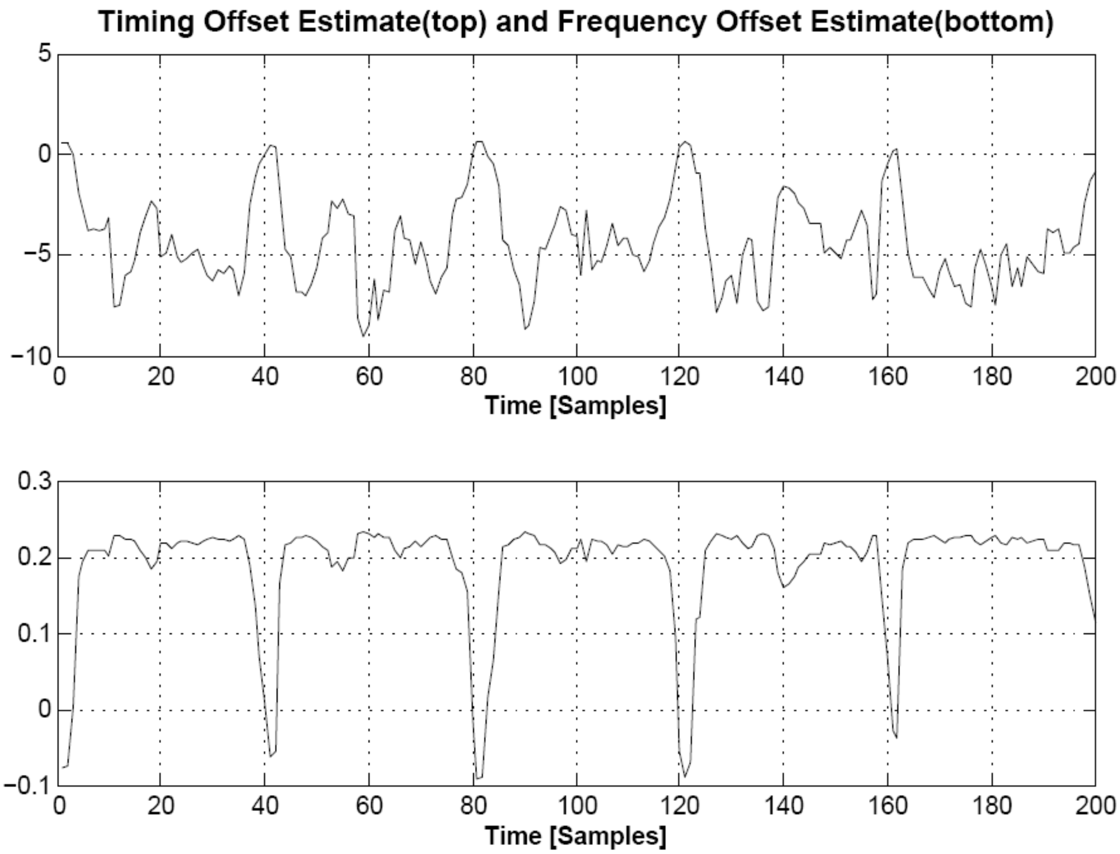

Figure 32: Timing (top) and frequency (bottom) offset estimate plots.

CHAPTER 6

This chapter presents the conclusions derived form the thesis and give an introduction to reconfigurable computing. Some limitations encountered during the research are also mentioned. The chapter is concluded with recommendations for further research directions study.

6.1 Conclusion:

The results of this experimental study of an OFDM system employing QPSK modulation show that the bit error rate (BER) for Q1.15, Q1.7 and Q1.6 are almost identical. This implies that we can use Q1.7 format instead of Q1.15 for a given value of BER. The results obtained via MATLAB simulation can be used to design an optimum OFDM system where the bit-widths can be adjusted according to SNR to achieve a desired level of BER. This also helps in reduction of power and silicon area. Such a system can be realized via reconfigurable computing hardware. Modern FPGAs have the ability of bit level manipulation at runtime and therefore can be used to construct such systems. A brief introduction to reconfigurable computing is presented and it application to wireless communication is also explained with an example.

6.2 Further Research:

The results obtained from the thesis can be used to develop a feasible implementation which can adapt in real-time according to the users need for quality or power. If the user is interested in quality i.e less number of errors, the system operates at a higher Q format (with more number of bits). If the user concern is the power then the system adapts to operate at a lower Q format (with lesser number of bits). Thus the desired system requires to reconfigure it self at runtime, which gives us the definition of reconfigurable hardware, i-e a hardware device in which the functionality of the logic gates is customized at run-time [25]. The most common example of such a device is the Field Programmable Gate Array (FPGA).

From the implementation point of view we find the following types of processing devices [26].

1. **General Purpose Processor**
2. **Dedicated/Application Specific Processor**
3. **Reconfigurable Processor**

General purpose processors offer high flexibility to run any type of algorithm since they are fully programmable, but the overheads caused by this high degree of freedom make them very inefficient.

The application specific processors on the other hand are extremely efficient as they offer dedicated hardware for the computations however they offer no flexibility as they are usually not programmable.

A reconfigurable architecture finds a middle ground between flexibility and efficiency by limiting its flexibility to particular domain of algorithms. Reconfigurable architectures have been classified into two categories based on what granularity it offers this flexibility.

- **Fine Grained Reconfigurable Hardware**

  The functionality of the hardware is specified at the bit level and the programmable interconnects are treated as individual wires. Such type of devices are relatively slow as there is additional overhead for manipulating the interconnects. Examples of fine grained architectures can be found in FPGAs like Altera Cyclone [27] and Xilinx Virtex-E [28]

- **Coarse Grained Reconfigurable Hardware**

  The functionality of the hardware is specified at the word-level and the architecture contains word level functional units, such as multipliers and Arithmetic Logic Units (ALUs). The programmable interconnect in coarse grained architectures is manipulated as buses. Coarse-grained architectures require less configuration data than fine-grained architectures and therefore their configuration time is shorter. An example of the coarse grained reconfigurable processor is the Montium Tile Processor [29].

Apart from the selection of the reconfigurable platform, a few things can be investigated further and will be helpful in the implementation of the reconfigurable OFDM systems are listed below.

1. Effects of higher order constellations like 16 QAM, 64 QAM on the BER when implemented in fixed point.
2. Effects of source and channel coding on the BER when implemented in fixed point.

3. Investing different implementations of IFFT/FFT for performing the OFDM modulation in terms of area and power.
4. Investigating low-complexity synchronizer architectures and their effects on BER.

REFERENCES:

[1] Jan-Jaap van de Beek, Magnus Sandell, Per Ola B¨orjesson, “ML Estimation of Time and Frequency Offset in OFDM Systems”, IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 45, NO. 7, JULY 1997

[2] Ye (Geoffrey) Li, Gordon Stuber “ORTHOGONAL FREQUENCY DIVISION MULTIPLEXING FOR WIRELESS COMMUNICATIONS”, 2006 Springer Science+Business Media, Inc.

[3] Juha Heiskala, John Terry, “OFDM Wireless LANS: A Theoretical and Practical Guide”, 2002 - Sams Publishing

[4] Charan Langton, “Orthogonal Frequency Division Multiplexing (OFDM) Tutorial”, Intuitive Guide to Principles of Communications, 2004

[5] Anibal Luis Intini, “Orthogonal Frequency Division Multiplexing for Wireless Networks, Standard IEEE 802.11a”, UNIVERSITY OF CALIFORNIA SANTA BARBARA, Dec 2000

[6] S. B. Weinstein and P. M. Ebert, "Data transmission by frequency division Multiplexing using the discrete Fourier transform," IEEE Trans. Commun., vol. 19, no. 10, pp. 628-634, Oct. 1971

[7] Shoab A Khan, “Digital Design of Signal Processing Systems”

[8] Weinstein and Clifford J, “QUANTIZATION EFFECTS IN DIGITAL FILTERS”, Technical rept, MASSACHUSETTS INST OF TECH LEXINGTON LINCOLN LAB, 21 NOV 1969

[9] R Edwards, “Application of non-uniform quantization to closed-loop digital systems”, J. Phys. E: Sci. Instrum. 2 321-326, 1969

[10] Jaime Ramirez-Angulo, Randall L. Geiger and Edgar San, “Component Quantization Effects on Contineous Time Filters”, Proceedings of ISCAS 1985.

[11] Evert Cooper, Ph.D. Member of the Technical Staff, Texas Instruments, “Minimizing Quantization Effects Using the TMS320 Digital Signal Processor Family”, 1994

[12] Hyun Wook Park and Yung Lyul Lee, “A Post-processing Method for Reducing Quantization Effects in Low Bit-Rate Moving Picture Coding”, IEEE

TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL. 9, NO. 1, FEBRUARY 1999

[13] Bo Shen and Qian-liang Zang, "A New Method for Analyzing the Quantization Effects of ADC in Broad Band QAM Receiver", IEEE 2002

[14] Davide Dardari, "Exact Analysis of Joint Clipping and Quantization Effects in High Speed WLAN Receivers", IEEE 2003

[15] Davide Dardari, "Joint Clip and Quantization Effects Characterization in OFDM Receivers", IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS, VOL. 53, NO. 8, AUGUST 2006

[16] Teresa Araújo and Rui Dinis, "Performance Evaluation of Quantization Effects on Multicarrier Modulated Signals", IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, VOL. 56, NO. 5, SEPTEMBER 2007

[17] Steve Gifford, John E. Kleider and Scott Chuprun, "BROADBAND OFDM USING 16-BIT PRECISION ON A SDR PLATFORM", IEEE Military Communications Conference, MILCOM 2001.

[18] Kyungtae Han, "Automating transformations from floating-point to fixed-point for implementing digital signal processing algorithms", PhD dissertation, Graduate School of Electrical and Computer Engineering, The University of Texas at Austin, 2006

[19] P. H. Moose, "A technique for orthogonal frequency division multiplexing frequency offset correction," *IEEE Trans. Commun.*, vol. 42, pp. 2908–2914, Oct. 1994.

[20] Baoguo Yang, Khaled Ben Letaief, Roger S. Cheng and Zhigang Cao, "Timing Recovery for OFDM Transmission", IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, VOL. 18, NO. 11, NOVEMBER 2000

[21] M. Speth, D. Daecke, and H. Meyr, "Minimum overhead burst synchronization for OFDM based broadband transmission," in *Proc. Global Telecommun. Conf.*, Sydney, Australia, Nov. 1998, pp. 2777–2782.

[22] Hlaing Minn, Vijay K. Bhargava and Khaled Ben Letaief, "A Robust Timing and Frequency Synchronization for OFDM Systems", IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, VOL. 2, NO. 4, JULY 2003

[23] W. D. Warner and C. Leung, “OFDM/FM frame synchronization for mobile radio data communication,” *IEEE Trans. Veh. Technol.*, vol. 42, pp. 302–313, Aug. 1993

[24] P. J. Tourtier, R. Monnier, and P. Lopez, “Multicarrier modem for digital HDTV terrestrial broadcasting,” *Signal Processing: Image Commun.*, vol. 5, no. 5–6, pp. 379–403, Dec. 1993.

[25] http://www.cs.cmu.edu/~phoenix/reconfigurable.html, Phoenix CMU Reconfigurable Nanotechnology Project

[26] Paul M. Heysters, Recore Systems, “Coarse-Grained Reconfigurable Computing for Power Aware Applications”, 2004

[27] http://www.altera.com/

[28] http://www.xilinx.com/

[29] http://www.recoresystems.com/

Appendix

MATLAB Code for OFDM BER in AWGN

```
clc
clear all
close all

% Definations
N = 64;           % fft/ifft size(no of symbols per OFDM symbol)
no_samples = 64;  % no of symbols
M = 4;            % modulation size
nsc = 64;         % no of subcarriers
L = 0.25*N;
EbNo = 0:25;
realizations=1000;% no of trials in Monte Carlo simulation
error_vector=[];
ratio_vector=[];
ensamble_error=zeros(1,length(EbNo));
ensamble_ratio=zeros(1,length(EbNo));
error_vector=[];
ratio_vector=[];
ensamble_error=zeros(1,length(EbNo));
ensamble_ratio=zeros(1,length(EbNo));
for iteration=1:realizations
    noise_index=0;
    error_vector=[];
    ratio_vector=[];
for start=EbNo(1):1:EbNo(length(EbNo))
    noise_index=noise_index+1;

% % TRANSMITTER
data = randint(1,no_samples*N,M);    % random data symbols
modulated_data = pskmod(data,M);     % modulated data

% Serial to Parallel Conversion for IFFT
parallel_data = reshape(modulated_data,no_samples,N).';
```

```
% Assigning data to Subcarriers
sc_data = parallel_data(1:N,1:nsc);

% Taking IFFT for OFDM modulation
ifft_data = (N/sqrt(nsc))*fi(ifft(sc_data),1,16,15);

% Computing of Cyclic Prefix
c_prefix = ifft_data(N-L+1:N,:);

% Appending Cyclic Prefix
c_prefixed_data = vertcat(c_prefix , ifft_data);
[rows,cols]=size(c_prefixed_data);

% Parallel to serial Coversion
serial_data = reshape(c_prefixed_data,1,rows*cols);

% %CHANNEL
rx =
awgn(sqrt(length(c_prefixed_data)/N)*serial_data,EbNo(noise_index));

% offset values
t_offset=[0 0 0 0];
f_offset=exp(i*0.25);
rx = f_offset*[t_offset rx];

% %RECEIVER
% %-------------------------------------------------------------------%%
%Maximum Liklihood Estimation of Frequencey and Timing offset

% Offset corrected data
rx_estimated = ML_estimation(rx, t_offset, f_offset, N, L, EbNo);

% Serial to Parallel Conversion for tranmission
rx_s_p = reshape(rx_estimated,rows,cols);

% Removing Cyclic Prefix
```

```
rx_prefix_less_data = rx_s_p(L+1:rows,:);

% FFT of Received data
rx_fft_data = (sqrt(nsc)/N)*fft(double(rx_prefix_less_data));
rx_fft_data = rx_fft_data.';

% Parallel to Serial
rx_p_s = reshape(rx_fft_data,1,N*nsc);

% QPSK Demodulation
rx_demodulate = pskdemod(rx_p_s,M);

% BER Calculations
[err,ratio]=biterr(data,rx_demodulate,2);
error_vector = [error_vector err];
ratio_vector = [ratio_vector ratio];
end
ensamble_error = ensamble_error+error_vector;
ensamble_ratio = ensamble_ratio+ratio_vector;
end

BER_1_15=ensamble_ratio/realizations;
save BER_data_Q1_15 BER_1_15

% Ploting BER Curves
semilogy(EbNo,BER_1_15),grid on,xlabel('dB'),ylabel('BER'),title('BER
curve for OFDM with QPSK with Q1.15 Format')
```

MATLAB Code for OFDM BER in Flat Fading Raleigh Channel

```
clc
clear all
close all

% Definations
N = 64;            % fft/ifft size(no of symbols per OFDM symbol)
no_samples = 64;  % no of symbols
M = 4;             % modulation size
nsc = 64;          % no of subcarriers
L = 0.25*N;
EbNo = 0:45;
realizations=1000;% no of trials in Monte Carlo simulation
error_vector=[];
ratio_vector=[];
ensamble_error=zeros(1,length(EbNo));
ensamble_ratio=zeros(1,length(EbNo));
error_vector=[];
ratio_vector=[];
ensamble_error=zeros(1,length(EbNo));
ensamble_ratio=zeros(1,length(EbNo));
for iteration=1:realizations
    noise_index=0;
    error_vector=[];
    ratio_vector=[];
for start=EbNo(1):1:EbNo(length(EbNo))
    noise_index=noise_index+1;



data = randint(1,no_samples*N,M); % random data symbols

modulated_data = pskmod(data,M);  % modulated data

% Serial to Parallel Conversion for IFFT
parallel_data = reshape(modulated_data,no_samples,N).';
```

```
% Assigning data to Subcarriers
sc_data = parallel_data(1:N,1:nsc);

% Taking IFFT for OFDM modulation
ifft_data = (N/sqrt(nsc))*fi(ifft(sc_data),1,8,7);

% Computing of Cyclic Prefix
c_prefix = ifft_data(N-L+1:N,:);

% Appending Cyclic Prefix
c_prefixed_data = vertcat(c_prefix , ifft_data);
[rows,cols]=size(c_prefixed_data);

% Parallel to serial Coversion
serial_data = reshape(c_prefixed_data,1,rows*cols);



% %CHANNEL
% multipath channel
channel = rayleighchan(1/8192,1/4096);
tx = filter(channel,serial_data);
rx = awgn(sqrt(length(c_prefixed_data)/N)*tx,EbNo(noise_index));
rx = serial_data;

h = [randn(1,(N+L)*no_samples) + j*randn(1,(N+L)*no_samples)];
tx = h.*serial_data;
rx = awgn(sqrt(length(c_prefixed_data)/N)*tx,EbNo(noise_index));

% offset values
t_offset=[0 0 0 0];
f_offset=0.25;
rx = f_offset*[t_offset rx];



% %RECEIVER
% %-----------------------------------------------------------------%%
```

```
% %Maximum Liklihood Estimation of Frequencey and Timing offset

% Offset corrected data
rx_estimated = ML_estimation(rx, t_offset, f_offset, N, L, EbNo);

% Equlization
rx_eq = double(rx_estimated)./h;

% Serial to Parallel Conversion for tranmission
rx_s_p = reshape(rx_eq,rows,cols);

% Removing Cyclic Prefix
rx_prefix_less_data = rx_s_p(L+1:rows,:);

% FFT of Received data
rx_fft_data = (sqrt(nsc)/N)*fft(double(rx_prefix_less_data));
rx_fft_data = rx_fft_data.';

% Parallel to Serial
rx_p_s = reshape(rx_fft_data,1,N*nsc);

% QPSK Demodulation
rx_demodulate = pskdemod(rx_p_s,M);

% BER Calculations
[err,ratio]=biterr(data,rx_demodulate,2);
error_vector = [error_vector err];
ratio_vector = [ratio_vector ratio];
end
ensamble_error = ensamble_error+error_vector;
ensamble_ratio = ensamble_ratio+ratio_vector;
end
BER_1_7_fading=ensamble_ratio/realizations;
save BER_data_Q1_7_fading BER_1_7_fading
ber_fading =  berfading(EbNo, 'psk', M, 1);
```

```
% Ploting BER Curves
semilogy(EbNo,ber_fading),grid on,xlabel('dB'),ylabel('BER'),title('BER
curve for OFDM with QPSK with Q1.7 Format in Rayleigh Fading Channel')
hold on
semilogy(EbNo,BER_1_7_fading,'-dr'),grid
on,xlabel('dB'),ylabel('BER'),title('BER curve for OFDM with QPSK with
Q1.7 Format in Rayleigh Fading Channel')
```

MATLAB Code for Maximum Likelihood (ML) Timing and Frequency Offset Estimator for OFDM signals

```
function rx_estimated = ML_estimation(rx, t_offset, f_offset, N, L,
EbNo)

rho = EbNo/(EbNo+1);                % normalization factor

rx_padded=[rx zeros(1,2*N+L) ];     % observation interval

gamma=[];

phi=[];

buffer=zeros(1,2*N+L); % observation interval

for i=1:(length(rx_padded)-(2*N+L+1))

        buffer=rx_padded(i:length(buffer)-1+i);
        temp_gamma=0;
        temp_phi=0;

    for m=1:L-1

        temp_gamma = temp_gamma + buffer(m) * conj(buffer(m+N)) ;
        temp_phi =temp_phi+( 0.5*( (abs(buffer(m))^2)
+(abs(buffer(m+N))^2) ));
```

```
    end

        gamma=[gamma temp_gamma];
        phi=[phi temp_phi];
end

difference=abs(gamma)-rho*(phi);
e_freq_offset=-(1/(2*pi))*atan(difference);
subplot(2,1,1),plot(difference(1:200)),xlabel('Time [Samples]'),grid on
subplot(2,1,2),plot(e_freq_offset(1:200)),xlabel('Time [Samples]'),grid
on
[temp theta_ml]= max(difference(1:2*N+L));
theta_ml-2 % compensate for the one-based index
epsilon_ml=-1/(2*pi)*angle( gamma( theta_ml))

% Offset corrected data
rx_estimated = 1/(f_offset)*(rx(length(t_offset)+1:length(rx)));
```