MODULATION TECHNIQUE FOR INTEGRATED BROADBAND WIRELESS APPLICATIONS

(UPP vote: 71261)

This project report is submitted in part fulfillment
of the requirements of RMC, UTM

UNIVERSITY TECHNOLOGY OF MALAYSIA

DECEMBER, 2006

ACKNOWLEDGEMENT

I would like to thank my former head of department and colleagues in the Radio Communication Engineering Department for their support, encouragement, and comments during the duration of the projects. I also wish to thank my PSM students for their help and not forgetting RMC for funding this project.

ABSTRACT

The increasing demand on high bit rate and reliable wireless system has led to many new emerging modulation techniques. One of the techniques will be Orthogonal Frequency Division Multiplexing (OFDM), which offers reliable high bit rate wireless system with reasonable complexity. The primary reasons OFDM is preferred in most high bandwidth efficiency transmission systems are because it effectively resist Intersymbol Interference (ISI) and is robust towards multipath fading.

Nevertheless, several of the main factors affecting the performance of an OFDM system are high Peak to Average Power Ratio (PAPR), phase noise and frequency offset.

This project developed a communication system based on OFDM modulation technique. The system employs Block Coding scheme in reducing the PAPR of the OFDM system. The capability of Block Coding scheme to reduce the Bit Error Rate (BER) in an OFDM system was also measured. The performance of OFDM was accessed by using computer simulations performed using two of the Matlab tools which are Simulink and GUIDE.

ABSTRAK

Perkembangan dan pertambahan permintaan terhadap sistem wireles yang menawarkan kadar bit yang tinggi telah menyebabkan banyak teknik modulasi yang baru muncul. Salah satu daripada teknik itu ialah Pemultipleksan Frekuensi Terbahagi Orthogon (OFDM) yang menawarkan sistem wireles berkadar bit tinggi dengan menggunakan kompleksiti yang munasabah. Sebab utama OFDM diberi keutamaan untuk kebanyakan sistem penghantaran yang menawarkan kecekapan lebar jalur yang tinggi ialah OFDM dapat menangani masalah Gangguan Antara Simbol (ISI) dengan berkesan dan ketahanannya terhadap pemudaran berbilang laluan.

Walaubagaimanapun, beberapa faktor yang menurunkan prestasi sistem OFDM ialah kuasa maksima isyarat OFDM yang tinggi berbanding dengan kuasa puratanya (PAPR), gangguan fasa dan ofset frekuensi.

Projek ini membangunkan sistem perhubungan yang berlandaskan pemodulatan OFDM. Sistem ini menggunakan teknik mengekod secara blok dalam mengurangkan PAPR untuk sistem OFDM. Kebolehan teknik Pengekodan Blok ini dalam mengurangkan Kadar Kesalahan Bit (BER) juga dikaji. Prestasi OFDM dikaji dengan menggunakan simulasi komputer iaitu Simulink dan GUIDE dari perisian Matlab.

TABLE OF CONTENTS

LIST OF TABLE

LIST OF FIGURES

LIST OF ABBREVIATIONS

4G	-	Fourth Generation
ADSL	-	Asymmetric Digital Subscriber Line
BER	-	Bit Error Rate
BPSK	-	Binary Phase Shift Keying
DAB	-	Digital Audio Broadcasting
DVB-T	-	Terrestrial Digital Video Broadcasting
FDM	-	Frequency Division Multiplexing
FFT	-	Fast Fourier Transform
GUIDE	-	Graphical User Interface Development Environment
IFFT	-	Inverse Fast Fourier Transform
ISI	-	Intersymbol Interference
LAN	-	Local Area Network
MC	-	Multi-carrier Communications
OFDM	-	Orthogonal Frequency Division Multiplexing
PA	-	Power Amplifier
PAPR	-	Peak to Average Power Ratio
PEP	-	Peak Envelope Power
PSK	-	Phase Shift Keying
PTS	-	Partial Transmit Sequencing
QAM	-	Quadrature Amplitude Modulation
QPSK	-	Quaternary Phase Shift Keying
SNR	-	Signal to Noise Ratio
VCO	-	Voltage-controlled Oscillator
VLSI	-	Very Large Scale Integration

CHAPTER I

INTRODUCTION

1.0 Introduction

Multimedia communications over radio channel requires wireless transmission system to offer high bandwidth efficiency. One of the best modulation techniques will definitely be Orthogonal Frequency Division Multiplexing (OFDM) that meets such requirements with reasonable complexity [6]. The primary reasons OFDM is preferred in most high bandwidth efficiency transmission systems are because it effectively resist Intersymbol Interference (ISI) and is robust towards multipath fading. Conventional modulation methods suffer from multipath in both the frequency domain and the time domain. In the frequency domain, multipath causes groups of frequencies to be attenuated and shifted in phase relative to each other which severely distorts the symbol. In the time domain, multipath basically smears adjacent symbols into each other. Many typical systems overcome these problems with expensive adaptive filters. OFDM, on the hand, uses groups of narrowband signals to carry the stream of data in order to counter this problem and employs a guard interval between symbols to prevent the ISI from causing errors to the data. This is because the guard interval does not carry any important information and it is just some cyclic redundancy bits that serve to let the ISI occur at the period of the guard interval. In that way the receiver can just discard the stream of bits that overlap with each other as it will not cause any error to the information bits.

1.1 History of OFDM

Orthogonal Frequency Division Multiplexing (OFDM) has grown to be the most popular communication system in high-speed wireless communications in the last decade. In fact, it has been claimed by many industry leaders that OFDM technology is the future of wireless communications, as we know it.

The roots of OFDM could be tracked back to late 1950’s. It was then developed in the 1960’s and originally found use in military radios. However, these ideas could not be realized efficiently, since powerful semiconductor devices were not available at that time. Today, even a relatively complex OFDM transmission system with high data rate can be realized with the advancement of Very Large Scale Integration (VLSI) and effective signals processing method.

Commercial use came later when the first OFDM-based standards were established. The first were the ETSI Digital Audio Broadcasting (DAB) system, standardized in 1995, and the Terrestrial Digital Video Broadcasting (DVB-T) system standardized in 1997. OFDM is also the modulation method for Asymmetric Digital Subscriber Line (ADSL) services.

OFDM slowly gaining popularity after it was set as a standard DAB. Then, using 16 independent channels where each carrying 256 kbps data rate, OFDM enabled transmissions to be sent and received simultaneously. DVB-T in Europe was also an early OFDM application. However, these broadcasting systems did not offer much promise for two-way communication.

In 1997, Lucent and NTT independently submitted proposals to The Institute of Electrical and Electronics Engineers (IEEE) for a high speed wireless standard for Local Area Network (LAN). Eventually, they combined their proposals and was accepted as a draft standard in 1998 and as a standard, now known as IEEE 802.11a standard in 1999. In the wireless arena, OFDM is the physical layer basis for the IEEE 802.11a standard, and has recently become standardized as the IEEE fixed broadband wireless access standard 802.16. Due to the suitability for high data rate services, OFDM has been identified as the leading contender for the physical layer of Fourth Generation (4G) wireless services. Table 1 shows the brief history of the OFDM application development.

Table 1: Brief History of OFDM

OFDM History

1957: Kineplex multi-carrier HF modem
1966: Chang, Bell Labs: OFDM paper+patent
1971: Weinstein & Ebert propose use of FFT and guard interval
1985: Cimini describes use of OFDM for mobile communications
1987: Alard & Lasalle: OFDM for digital broadcasting
1995: ETSI DAB standard: first OFDM-based standard
1997: DVB-T standard
1998: Magic WAND project demonstrates OFDM modems for wireless LAN
1999: IEEE 802.11a and HIPERLAN/2 standards for wireless LAN
2000: V-OFDM for Fixed Wireless Access
2001: OFDM considered for new IEEE 802.11 and 802.16 standards

Richard van Nee – Woodside Networks

1.2 Evolution of OFDM

Frequency Division Multiplexing (FDM) has been used for a long time to carry more than one signal over telephone line. FDM is the concept of using different frequency channels to carry information of different users. Each channel is identified by the center frequency of its transmission. Guard-band or gap is left between the different channels so that the signal of one channel will not overlap with the signal from the adjacent channel. Therefore the guard-band will lead to inefficient which were exaggerated in the early days since the lack of digital filtering made it difficult to filter closely packed adjacent channels.

Multi-carrier Communications (MC) was introduced to increase the overall capacity of communications, thereby increasing the overall throughput. MC is actually the concept of splitting a signal into a number of signals and modulating each of these new signals over its own frequency channel. These different frequency channels will then be multiplexed together in FDM manner.

Orthogonal Frequency Division Multiplexing (OFDM) is a form of MC where the different carriers are orthogonal to each other. Orthogonal in this aspect means that the signals are totally independent. In OFDM, the data on each carrier overlap the data in the adjacent carriers. This overlap creates a source of extra spectral efficiency in OFDM. Another source of spectral efficiency is the fact that the drop off of the signal at the band is primarily due to a single carrier that is carrying a low data rate. OFDM allows for sharp edges that correspond closer to the desired rectangular shape of the spectral power density of the signal. Figure 1.1 shows the orthogonal overlapping of the OFDM subcarriers.

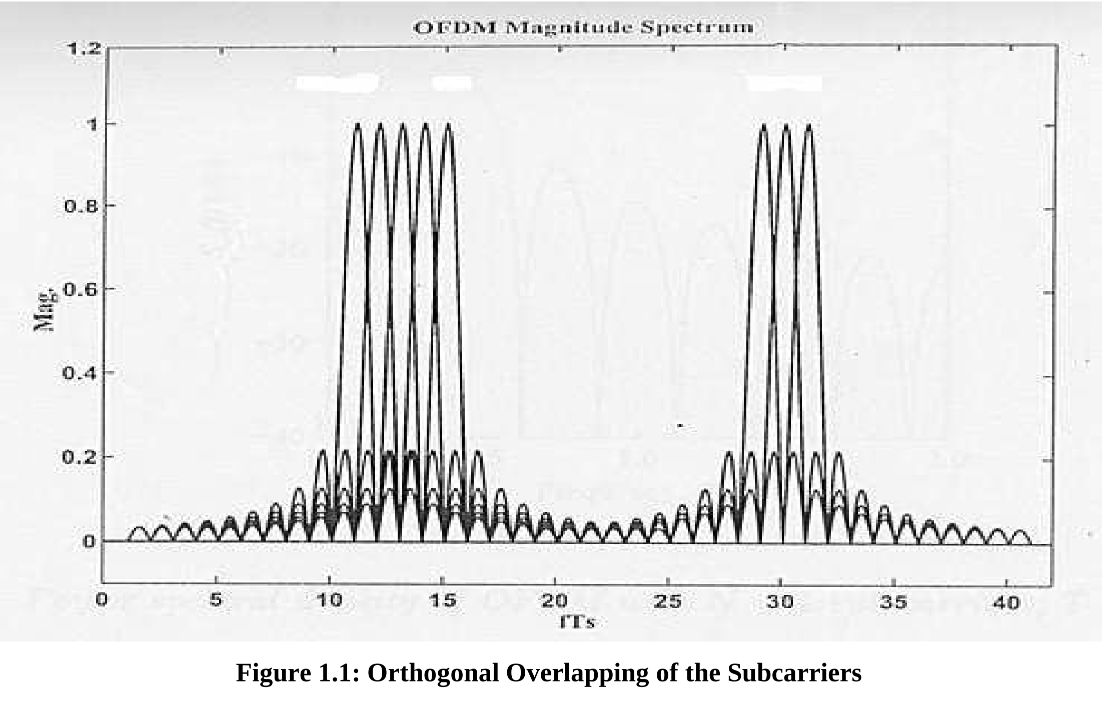

Figure 1.1: Orthogonal Overlapping of the Subcarriers

1.3 OFDM Concept and Architecture

An OFDM signal is basically a bundle of narrowband carriers transmitted in parallel at different frequencies from the same source. In fact, this modulation scheme is often termed "multicarrier" as opposed to conventional "single carrier" schemes. Each individual carrier, commonly called a subcarrier, transmits information by modulating the phase and possible the amplitude of the subcarrier over the symbol duration. That is, each subcarrier uses either phase-shift-keying (PSK) or quadrature-amplitude-modulation (QAM) to convey information just as conventional single carrier systems. However, OFDM or multi-carrier systems use a large number of low symbol rate subcarriers. The spacing between these subcarriers is selected to be the inverse of the symbol duration so that each subcarrier is orthogonal or non-interfering. This is the smallest frequency spacing that can be used without creating interference. At first glance it might appear that OFDM systems must modulate and demodulate each subcarrier individually. Fortunately, the well-known Fast Fourier transform (FFT) provides designers with a highly efficient method for modulating and demodulating these parallel subcarriers as a group rather than individually.

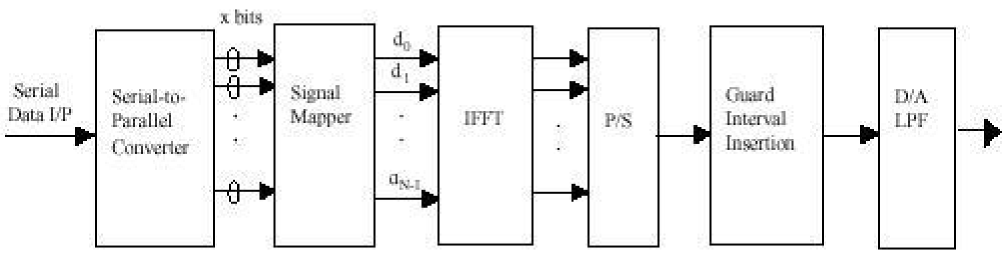

Figure 1.2: General OFDM Transmitter

Figure 1.2 illustrates the process of a typical OFDM transmitter. The incoming data is first converted from serial to parallel and grouped into x bits each to be modulated by Quadrature Amplitude Modulation (QAM), Quaternary Phase Shift Keying (QPSK), or Binary Phase Shift Keying (BPSK). The required spectrum is then converted back to its time domain signal using an Inverse Fast Fourier Transform (IFFT), commonly used in most applications. The IFFT performs the transformation very efficiently, and provides a simple way of ensuring the carrier signals produced are orthogonal. The signals are then converted back to serial for transmission. A guard interval is inserted between symbols to avoid Intersymbol Interference (ISI) caused by multipath distortion. The discrete signals are converted back to analogue. Although it would seem that combining the inverse FFT outputs at the transmitter would create interference between subcarriers, the orthogonal spacing allows the receiver to perfectly separate out each subcarrier. Figure 1.3 illustrates the general OFDM receiver. The receiver performs the inverse process of the transmitter.

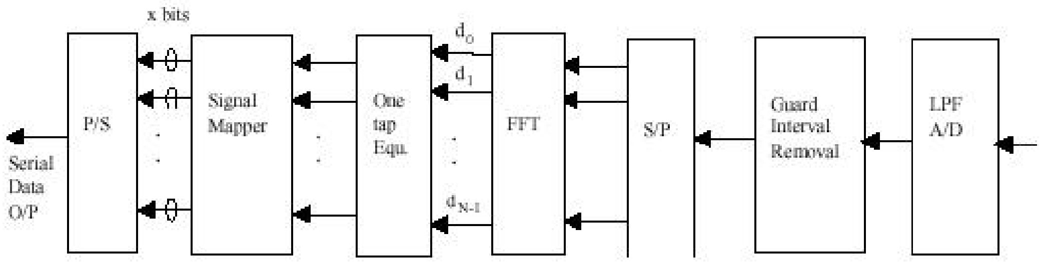

Figure 1.3: General OFDM Receiver

1.4 Fast Fourier Transform (FFT)

The Fast Fourier Transform (FFT) transforms a cyclic time domain signal into its equivalent frequency spectrum. This is done by finding the equivalent waveform, generated by a sum of orthogonal sinusoidal components. The frequency representation of the time domain signal will be the amplitude and phase of the sinusoidal components. The IFFT performs the reverse process, transforming the frequency domain signals into time domain signals. An IFFT converts a number of complex data points, of length, which is a power of 2, into the time domain signal of the same number of points. Each data point in frequency spectrum used is called a bin. Setting the amplitude and phase of each bin, then performing the IFFT can easily generate the orthogonal carriers.

1.5 Benefits of OFDM

The main benefits of OFDM compared to other multi-carrier techniques are:-

1. Spectral Efficiency
2. Resiliency to RF Interference
3. Lower Multi-Path Distortion
4. High Bit Rate Wireless Communications
5. Reducing ISI Probability

1.5.1 Multi-path Challenges

In OFDM-based WLAN architecture, as well as many other wireless systems, multi-path distortion is a key challenge. This distortion occurs at a receiver when objects in the environment reflect a part of the transmitted signal energy. Figure 1.4 illustrates one such multi-path scenario from a WLAN environment.

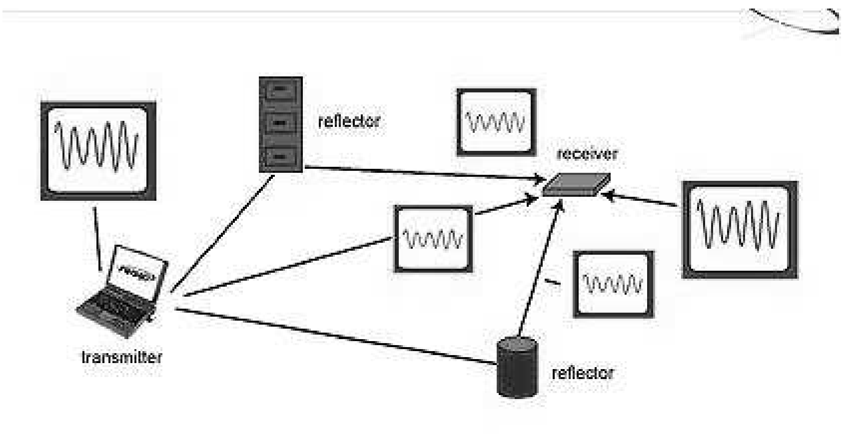

Figure 1.4: Multi-path Reflections in a Common WLAN Environment

Multipath reflected signals arrive at the receiver with different amplitudes, different phases, and different time delays. Depending on the relative phase change between reflected paths, individual frequency components will add constructively and destructively. Consequently, a filter representing the multi-path channel shapes the frequency domain of the received signal. In other words, the receiver may see some frequencies in the transmitted signal that are attenuated and others that have a relative gain.

In the time domain, the receiver sees multiple copies of the signal with different time delays. The time difference between two paths often means that different symbols will overlap or smear into each other and create Intersymbol Interference (ISI). Thus, designers building WLAN architectures must deal with distortion in the demodulator.

Recall that OFDM relies on multiple narrowband subcarriers. In multi-path environments, the subcarriers located at frequencies attenuated by multi-path will be received with lower signal strength. The lower signal strength leads to an increased error rate for the bits transmitted on these weakened subcarriers.

Fortunately for most multi-path environments, this only affects a small number of subcarriers and therefore only increases the error rate on a portion of the transmitted data stream. Furthermore, the robustness of OFDM in multi-path can be dramatically improved with interleaving and error correction coding. Let's look at error correction and interleaving in more detail.

1.5.2 Handling ISI

The time-domain counter part of the multipath is the ISI or smearing of one symbol into the next. OFDM gracefully handles this type of multipath distortion by adding a “guard interval” to each symbol. This guard interval is typically a cyclic or periodic extension of the basic OFDM symbol. In other words, it looks like the rest of the symbol, but conveys no 'new' information.

Since no new information is conveyed, the receiver can ignore the guard interval and still be able to separate and decode the subcarriers. When the guard interval is designed to be longer than any smearing due to the multipath channel, the receiver is able to eliminate ISI distortion by discarding the unneeded guard interval. Hence, ISI is removed with virtually no added receiver complexity.

It is important to note that discarding the guard interval does have an impact on the noise performance since it reduces the amount of energy available at the receiver for channel symbol decoding. In addition, it reduces the data rate since no new information is contained in the added guard interval. Thus a good system design will make the guard interval as short as possible while maintaining sufficient multipath protection.

Why don't single carrier systems also use a guard interval? Single carrier systems could remove ISI by adding a guard interval between each symbol. However, this has a much more severe impact on the data rate for single carrier systems than it does for OFDM. Since OFDM uses a bundle of narrowband subcarriers, it obtains high data rates with a relatively long symbol period because the frequency width of the subcarrier is inversely proportional to the symbol duration. Consequently, adding a short guard interval has little impact on the data rate.

Single carrier systems with bandwidths equivalent to OFDM must use much shorter duration symbols. Hence adding a guard interval equal to the channel smearing has a much greater impact on data rate.

1.5.3 Spectral Efficiency and High Bit Rate Data Transfer

The data is distributed over a large number of subcarriers. These subcarriers are spaced apart at precise frequencies so that the different subcarriers will be orthogonal in time, but overlap in frequency. Thus, higher data rate can be sent over a specific bandwidth, increasing the spectral efficiency. The frequency spacing is selected at a minimum distance that is able to provide orthogonal to the subcarriers. This is done through the use of IFFT computations to optimize the spectral efficiency.

The two main obstacles that limit the maximum bit rate of the wireless system are multi-path fading and ISI. These can be overcome without much complexity in OFDM system and thus propel it to be the leading candidate for high data rate wireless system.

1.6 OFDM Setbacks

OFDM signals have a high Peak to Average Power Ratio (PAPR) causing RF devices to operate at a lower efficiency to avoid working in their non-linear region. OFDM systems also require a guard interval that causes a loss in power and bandwidth efficiency. Besides that, OFDM is sensitive to frequency offsets and phase noise.

Frequency offsets occur when the voltage-controlled oscillator (VCO) at the receiver is not oscillating at exactly the same carrier frequency as the VCO in the transmitter. For the receiver, this offset between the two VCOs is seen as frequency translation in the signal and can lead to an increase in the error rate. While this is generally true for all modulations, OFDM is particularly sensitive to frequency offsets.

In addition to the constant frequency offset discussed above, the frequency generated by a practical VCO tends to jitter, or vary, over time. To the receiver, this frequency variation looks like noise in the phase of the received signal and as a result this impairment is referred to as phase noise.

1.7 Report Outline

The purpose of this project is to develop an OFDM system through MATLAB simulation model. As an enhancement to the OFDM system, a PAPR reduction technique is employed with a GUI presentation. Block coding technique has been chosen in this work as it provides simplicity in the implementation. This report is written in a format according to the university standard. It is divided into 5 chapters. Chapter I is a brief introduction and explanation to this project; Chapter II is the introduction about the Peak to Average Power Ratio (PAPR) problem and solutions; Chapter III is the description of the software implementation of this project; Chapter IV is the simulation results and analysis; Chapter V is the conclusions and recommendations.

CHAPTER II

LITERATURE REVIEW

2.0 Introduction

The major drawback of OFDM is the high Peak to Average Power Ratio. This can be defined as the large variation or ratio between the average signal power and the maximum or minimum signal power as illustrated by Figure 2.1

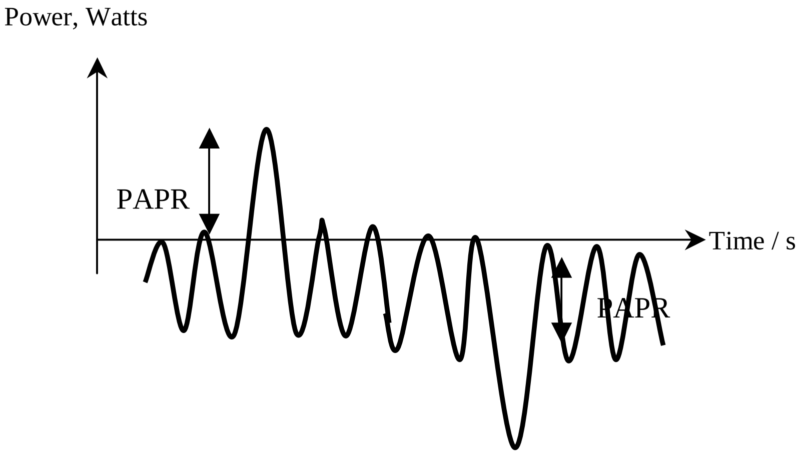

Figure 2.1 Peak to Average Power Ratio

To understand the cause of PAPR, it must be recalled that the OFDM signal generated by IFFT is the summation of all the subcarriers. These subcarriers are independent of each other. They tend to add up in phase and out of phase and this create a large variation [5] in the signal power as illustrated in Figure 2.2

2.1 Effects of PAPR

The large variation in signal power affects both the transmitter and receiver design. This is because a very linear power amplifier with a large dynamic range is required at both the transmitter and the receiver. Any amplifier nonlinearity will result in a significant signal distortion.

OFDM must keep its average power below the nonlinear region to accommodate the infrequent signal power peaks. But this result in a lower output power and also affects the efficiency and the range of the signal. Thus a careful design will compensate the distortion and the output power. In order to select an average input level, it must be assured that the average input level is possible of generating sufficient output power with no interference. Higher number of subcarriers used also will increase the value of PAPR. This is because PAPR is defined as

$$\text{PAPR} = 10 \log N,$$

where N is the number of subcarriers

Thus PAPR depends on the number of subcarriers and the level of SNR that must be maintained at the receiver.

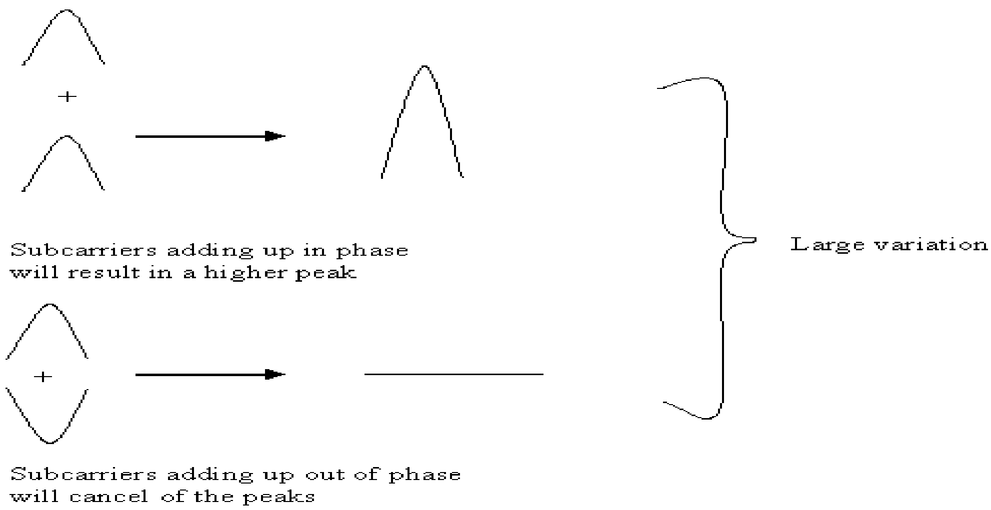

Figure 2.2 Large variations in OFDM power signal

2.2 Methods to Reduce PAPR

2.2.1 Clipping and Filtering

Deliberate clipping is a simple approach and, since the large peaks occur with a very low probability, clipping could be on effective technique for the reduction of the PAPR. However, clipping is a nonlinear process and may cause significant in- band distortion, which degrades the bit error rate performance, and out of band noise, which reduces the spectral efficiency. Filtering after clipping can reduce the spectral splatter but may also cause some peak regrowth. If digital signals are clipped directly, the resulting clipping noise will all fall into in-band and can not be reduced by filtering. To avoid this aliasing problem, oversample each OFDM block by padding the original input with zeros and taking a longer IFFT. Filtering after clipping is required to reduce the out of band clipping noise. The other approach to clipping is to use Forward Error Correcting codes and bandpass filtering with clipping. This method improves the bit error rate performance and spectral efficiency.

2.2.2 Partial Transmit Sequencing (PTS)

In the Partial Transmit Sequencing (PTS) scheme, subcarriers are partitioned into blocks and each block is multiplied by a constant phase factor. These phase factors are optimized to minimize the PAPR. These sub-blocks are then combined to minimize the peak. Optimal phase factors are sent to the receiver as side information.

One of the method to reduce the PAPR is block coding. This technique, proposed by Jones, Wilkinson and Barton (1994), uses a special block codes for subcarrier modulation rather than allowing the data to modulate the subcarriers directly. This technique limits the set of possible signals that can be transmitted as it only allows codes that generate lower envelopes to be transmitted. Code words with minimum PAPR will be determined from a given set of codes words. The reduction of PAPR is at the expense of a decrease in coding rate. This technique is also not suitable for higher order bit rates or large number of subcarriers.

Block Coding scheme will not cause distortion to the data during encoding. Unlike clipping scheme where in-band distortion and phase noise is introduced during the clipping process, Block Coding technique can even offer bit error correction capability through the use of the added redundancy bits. In probability scheme, delay will be caused by the looping search to find the most optimized phase factor so that the sub-blocks upon combining will give a low PEP. This will create a long delay if the number of subcarriers and data is large. Fortunately, for Block Coding scheme, there is no delay during the process of encoding. While both technique will add redundancy bits to the data, the Block Coding technique proven to be the far better choice due to its non-delay encoding characteristic.

2.2.3 Concept of Block Coding to Reduce PAPR

Block Coding is one of the leading methods used to reduce PAPR in OFDM system due to its simple algorithm, implementation and distortion less properties. A block coding scheme has been proposed by Jones, Wilkinson and Barton in 1994 to combat the high PAPR exhibits by OFDM system that severely limits its usage in Wireless Local Area Network environment. This is because if the amplifiers are to be operated with a high input back-off (IBO) to avoid the severe nonlinear distortion cause by high PAPR, the power efficiency of the amplifier will be very low. However, power efficiency is of critical importance for indoor wireless systems with portable terminals.

Moreover the cost of the amplifier will be very high and will be not economical in the corporate point of view as there are cheaper alternatives. Therefore block coding scheme is widely preferred as it reduces PAPR with simple complexity.

The basic concept of block coding is to find code words with minimum PAPR from a given set of code words and to map the input data blocks to these selected code words. Thus, it avoids transmitting the code words which generates high Peak Envelope Power. But, this reduction of PAPR is at the expense of a decrease in coding rate.

It can be seen from this table that four words result in the maximum PEP of 16.00 W and another four words result in PEP of 9.45 W. It is clear that we could reduce the PAPR of this multicarrier signal by avoiding transmitting these words. This can be done by block coding the data such that a 3 bit data word is mapped on to a 4 bit code word that gives a low PEP. For example, the data: 0 1 0 is mapped to a code word of 0 1 0 0 that gives the PEP value of 7.07 W. Note that whenever a data sequence of 0 1 0 is sent, it will first be encoded to 0 1 0 0 before being sent. This way, it has prevent the sequence of 0 1 0 1 (16 W) being sent.

Table 2: PEP during Symbol Period for all Possible Data Words

d_{dec}	d_1	d_2	d_3	d_4	PEP, W
0	0	0	0	0	16
1	0	0	0	1	7.07
2	0	0	1	0	7.07
3	0	0	1	1	9.45
4	0	1	0	0	7.07
5	0	1	0	1	16
6	0	1	1	0	9.45
7	0	1	1	1	7.07
8	1	0	0	0	7.07
9	1	0	0	1	9.45
10	1	0	1	0	16
11	1	0	1	1	7.07
12	1	1	0	0	9.45
13	1	1	0	1	7.07
14	1	1	1	0	7.07
15	1	1	1	1	16

Figure 2.1 shows the OFDM System with Block Encoder. The application of the block codes in OFDM system is straightforward. The encoded data before being mapped through QAM is given by:

$$x = uG_N + b_N \qquad \text{(mod M)}$$

where

G_N k x N constant matrix, $N = 2^{k-1}$

b_N constant sequence of length N

u input data

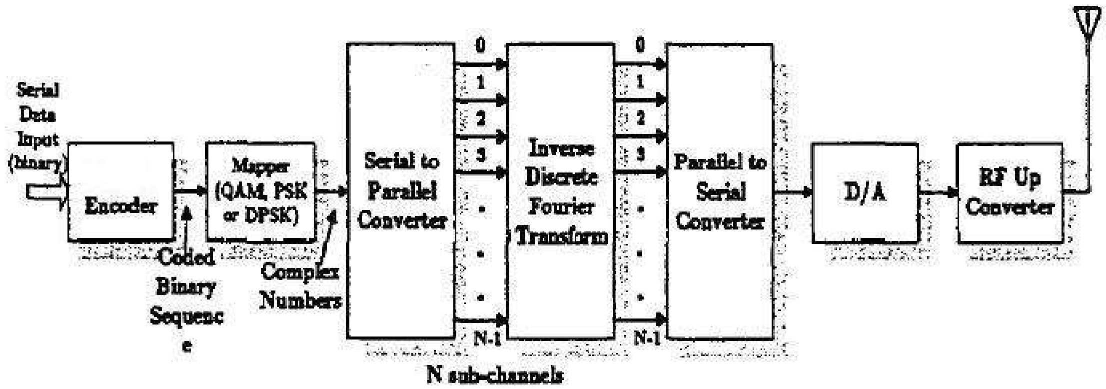

Figure 2.3: Block Diagram of the OFDM System with Block Encoder

Therefore, we can summarize that the idea of this block-coding scheme is, in general, to avoid the use of code words which yield a high PAPR and exploit the resultant redundancy for error correction property.

2.2.4 Generating Block Codes

The method chosen as the PAPR reduction scheme is Linear Block Coding. Linear block codes are a class of parity-check codes that can be characterized by the (n, k) notation. The encoder transforms a block of k message digits (a message vector) into a longer block of n codeword digits (a code vector) constructed from a given alphabet of elements. When the alphabet consists of two elements (0, 1), the code is a binary code comprising binary digits (bits).

The k-bit messages form 2^k distinct message sequences, referred to as k-tuples (sequences of k digits). The n-bit blocks can form as many as 2^n distinct sequences referred to as n-tuples. The encoding procedure assigns to each of the 2^k message k-tuples to one of the 2^n n-tuples. A block code represents a one-to-one assignment, whereby the 2^k message k-tuples are uniquely mapped into a new set of 2^k codeword n-tuples; the mapping can be accomplished via a look-up table. For linear codes, the mapping transformation is, of course linear.

The linear block coding can be used to reduce PAPR by assigning the n-bit block combination that yields the lowest PAPR after the Inverse Fast Fourier Transform. The n-bit block combination is selected from the n-tuples. The performance of the reduction of BER will depends on the code rate used for encoding the data. For a code word of (8, 4), the maximum error can be corrected is one because the limitation of d_{min} is three. As for a lower code rate where the code word is given by (8, 3), the maximum error that can be corrected is two. Further discussions will be seen later in Chapter 5.

CHAPTER III

METHODOLOGY

3.0 Introduction

OFDM system with various Block Coding techniques and coding rates were modeled using Matlab tools such as Simulink and Graphical User Interface Development Environment (GUIDE). This was to allow various parameters in the system to be varied and tested. The aim of doing the simulations was to measure the performance of the PAPR reduction in the OFDM system under different Block Coding techniques. Three main types of Block Coding techniques were developed which were using Generator Matrix, Look-up Table Mapping and Odd Bit Encoding. The effects of different coding rates on the BER performance of the OFDM system under an AWGN channel were also analyzed.

3.1 Development of Simulation System

The main and basic simulation system was based on the general OFDM system that uses Block Coding as its PAPR reduction scheme. The data was generated by Bernoulli Random Binary Generator and will be encoded before being mapped by Rectangular QAM Modulator Baseband. The subcarriers will then be processed using the IFFT. The OFDM signals will go through the AWGN channel where the Signal to Noise Ratio (SNR) and input power can be changed to suit the analysis need. The receiver will perform the reverse process of the transmitter. PAPR will be calculated based on the transmitted OFDM signals. BER will also be calculated by comparing the transmitted data with the received data. The data generated was sent in frame form to represent the parallel data in real world application. This will ease the simulation process as the serial to parallel converter was omitted.

3.2 Basic OFDM Simulation Model

The basic OFDM model was a general OFDM model used in real-time application. The data was not coded with any of the Block Coding technique. Therefore no reduction of the PAPR will be obtained. This model was developed in order to give a comparison analysis of the performance with the block coded model.

This OFDM system consisted of the OFDM transmitter, OFDM receiver, AWGN Channel, PAPR Calculation System, and also BER Calculation system. Figure 3.1 clearly shows the flow of the simulation model process.

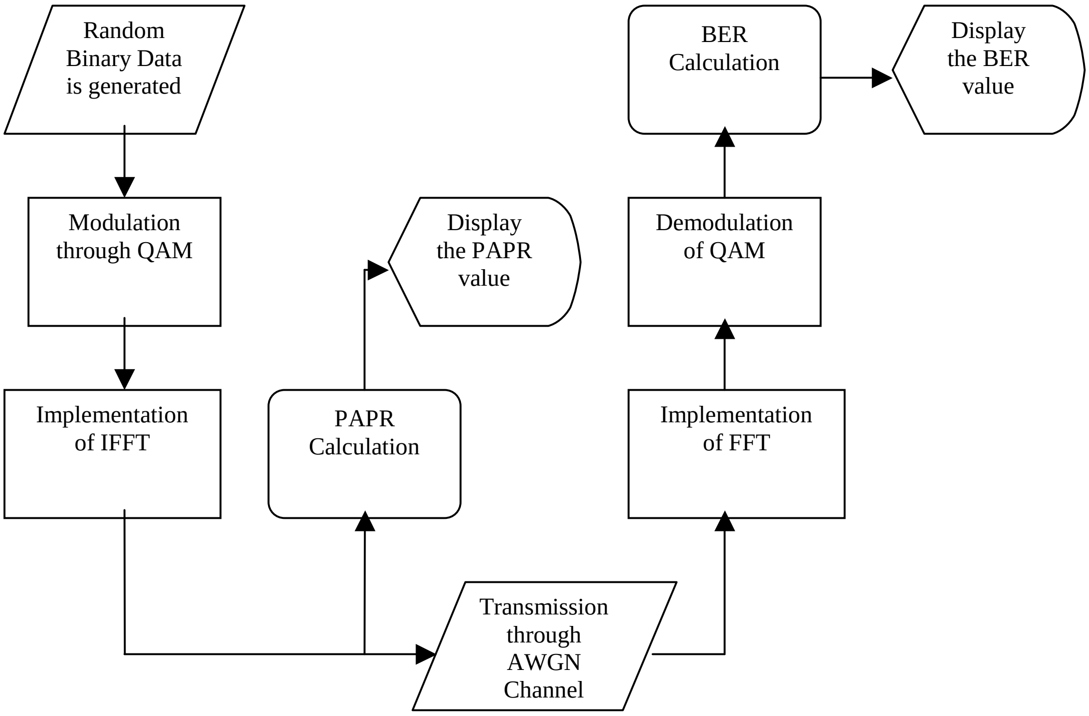

Figure 3.1: Basic OFDM Simulation Model Flowchart

Simulink Model

Figure 3.2 illustrates the simulation model of the Uncoded Basic OFDM Model developed using Simulink.

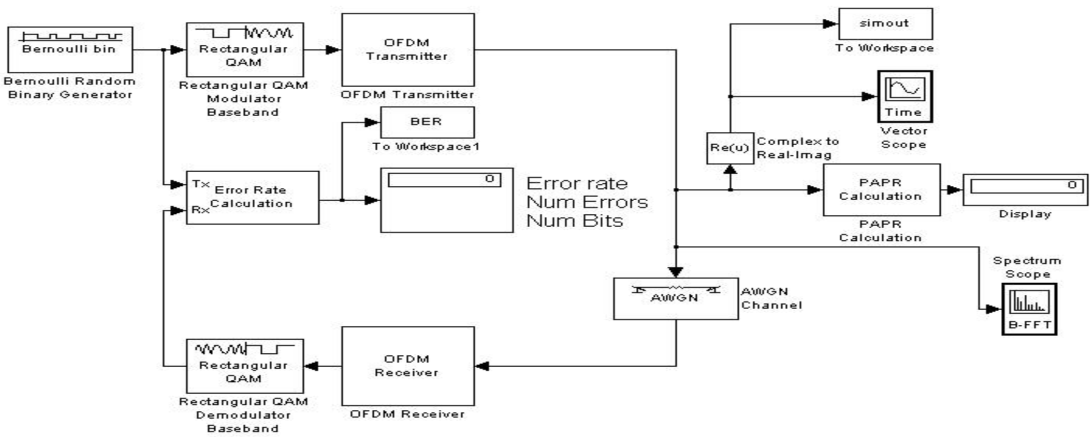

Figure 3.2: Uncoded Basic OFDM Model

3.2.2 Bernoulli Random Binary Generator

Figure 3.3 shows the dialog box to setup the parameters for Bernoulli Random Binary Generator. Few parameters were set to generate the random binary data in the simulation model.

1. Probability of a zero:
   - The probability with which a zero output occurs.
   - This parameter is set to 0.2 based on the maximum likelihood the PAPR will occur nearest to the theory value.
2. Initial seed:
   - The initial seed value for the random number generator.
   - The seed can be either a vector of the same length as the Probability of a zero parameter, or a scalar.
3. Sample time:
   - The period of each sample-based vector or each row of a frame-based matrix.
   - The sample time of 2.8579×10^{-7} was chosen based on the standard DVB-T sample time.
4. Frame-based outputs:
   - Determines whether the outputs are frame-based or sample-based.

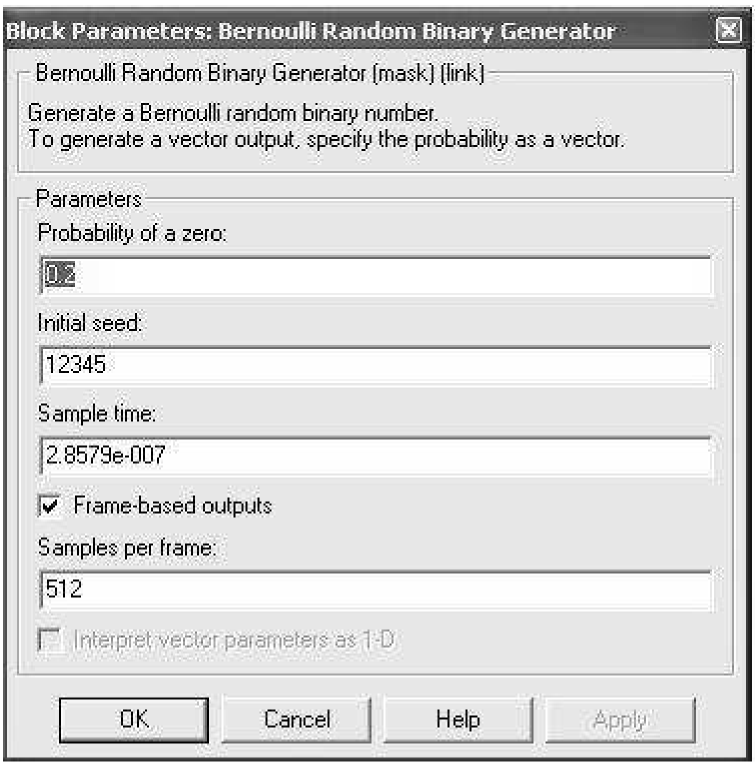

Figure 3.3: The Bernoulli Random Binary Generator Parameter Setup Dialog Box

5. Samples per frame:

- The number of samples in each column of a frame-based output signal.
- This parameter determined the number of parallel data stream in the OFDM system.

3.2.3 Rectangular QAM Modulator Baseband

Figure 3.4 shows the dialog box to setup the parameters for Rectangular QAM Modulator Baseband.

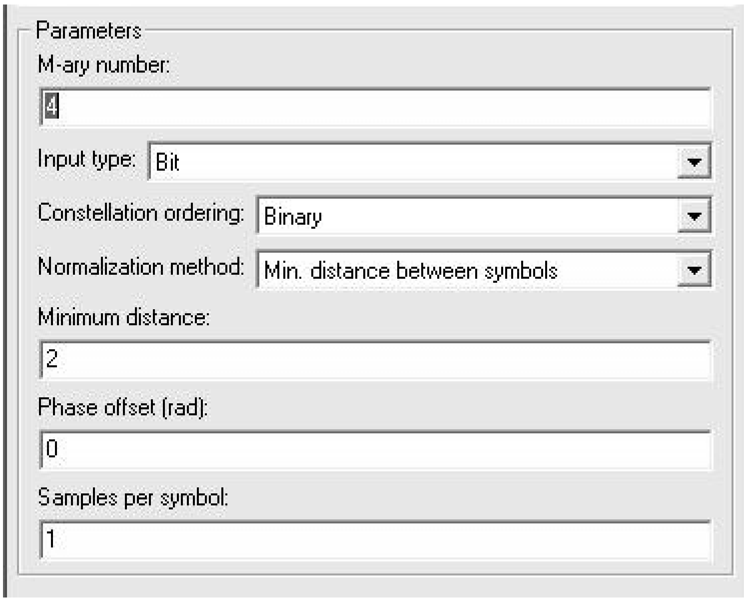

Figure 3.4: Rectangular QAM Modulator Baseband Parameter Setup Dialog Box

The parameters of the QAM Modulator:

1. M-ary number:
    - The number of points in the signal constellation.
    - It must have the form 2^K for some positive integer K.
2. Input type:
    - Indicates whether the input consists of integer or groups of bits.
3. Constellation ordering:
    - Determines how the block maps each group of input bits to a corresponding integer.
4. Normalization method:
    - Determines how the block scales the signal constellation.
    - Min. distance between symbols was chosen for OFDM system.
5. Minimum distance:

- The distance between two constellation points.

6. Phase offset (rad):
   - The rotation of the signal constellation, in radians.
7. Samples per symbol:
   - The number of output samples produced for each integer or binary word in the input.

3.2.4 OFDM Transmitter

The OFDM transmitter in Figure 3.5 consists of the IFFT Simulink blockset that process the data symbols according to the IFFT computations. The number of symbols per column in the frame based input will determine the number of subcarriers. Based on Figure 3.5, the number of subcarriers is 256.

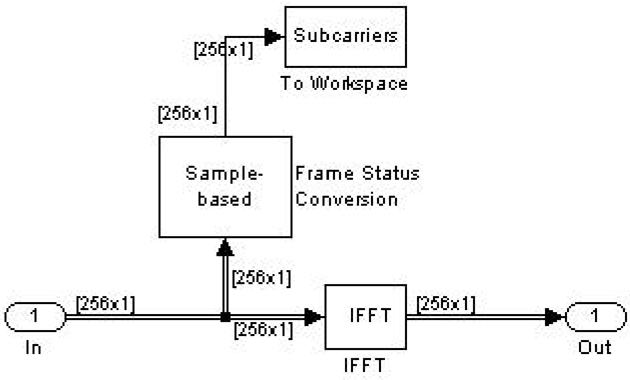

Figure 3.5: OFDM Transmitter

3.2.5 PAPR Calculation Subsystem

The PAPR calculation was based on the theoretical equation of:

$$\text{PAPR} = \frac{\max|S(t)|^2}{\text{average}|S(t)|^2}$$

S(t) = Modulated OFDM signals

Figure 3.6 shows the PAPR calculation of the OFDM signals. The OFDM signals will be converted from complex to real form. Then the real part of the OFDM signals will be squared. This will give the PEP value. The maximum value will then be divided by the mean value of the PEP. The answer is converted to dB value.

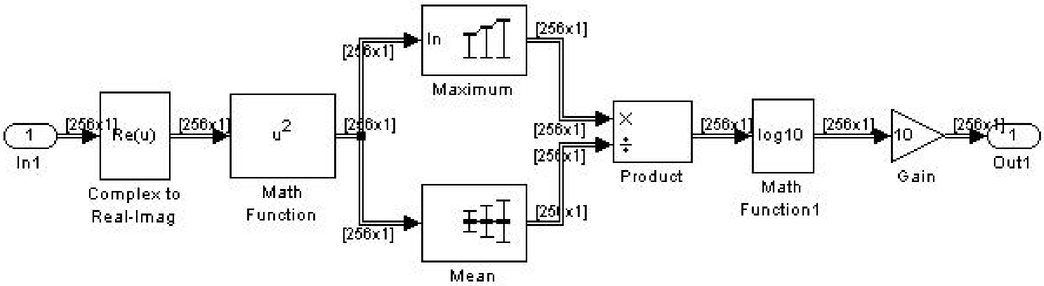

Figure 3.6: PAPR Calculation Subsystem

3.2.6 AWGN Channel

Figure 3.7 shows the setup dialog box of the AWGN Channel. The following parameters were set:

1. Initial seed:
   - The seed for the Gaussian noise generator.
2. Mode:
   - The mode by which you specify the noise variance.
   - The variance chosen for the OFDM model was SNR.
3. SNR (dB):
   - The SNR value can be varied according to the analysis.
4. Input signal power (watts):
   - The root mean square power of the input symbols.

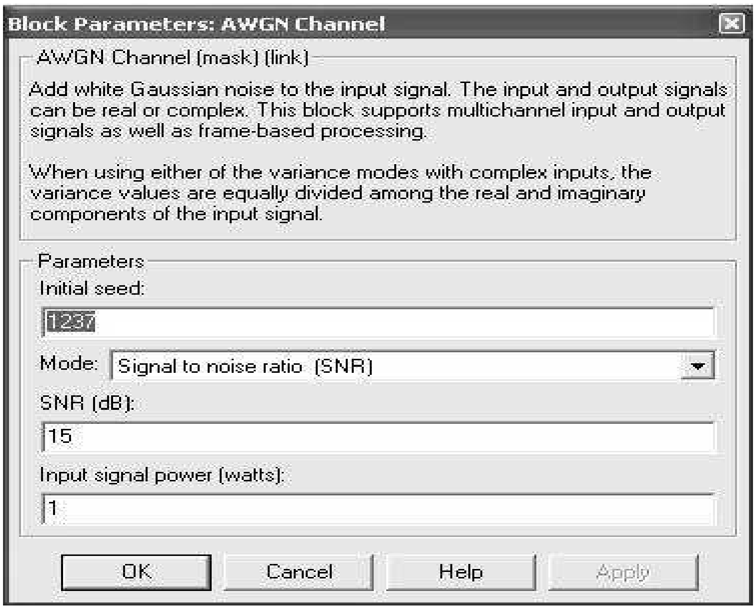

Figure 3.7: AWGN Channel Parameters Setup Dialog Box

3.2.7 OFDM Receiver

Figure 3.8 illustrates the OFDM Receiver Subsystem. The OFDM Receiver consisted of a FFT blockset and a Frame-based status Converter. This was used to convert the output symbol of the FFT blockset to frame-based.

3.2.8 BER Calculation Subsystem

The BER Calculation Subsystem calculates the BER by comparing the binary data generated by the Bernoulli Random Generator with the binary data retrieved from the QAM demodulator.

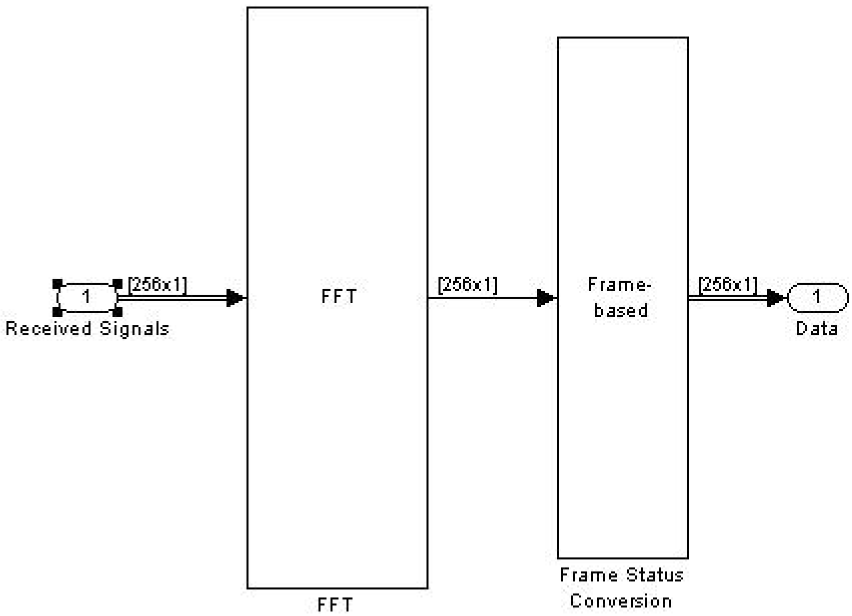

Figure 3.8: OFDM Receiver

3.3 Generator Matrix Technique

The Generator Matrix technique is basically to encode the data to be the code words that give a low PAPR. It is done by generating the required code words as needed through multiplying the data with the Generator Matrix. For example, if a sequence of four bits data 0 0 1 0 is to be transmitted, the encoder will first encode the data by multiplying it with its Generator Matrix.

$$[0 \;\; 0 \;\; 1 \;\; 0] \times \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} = [0 \;\; 1 \;\; 1 \;\; 0 \;\; 0 \;\; 0 \;\; 1 \;\; 0]$$

By sending this sequence instead of 0 0 1 0, we have controlled the PEP to a desired level. The full codeword representations of data are illustrated in Table 3 below.

Table 3: The Data and its Equivalent Codeword after the Generator Matrix Encoder

Data	Code Words
0 0 0 0	0 0 0 0 0 0 0 0
0 0 0 1	0 1 1 1 0 0 0 1
0 0 1 0	0 1 1 0 0 0 1 0
0 0 1 1	0 0 0 1 0 0 1 1
0 1 0 0	0 1 0 1 0 1 0 0
0 1 0 1	0 0 1 0 0 1 0 1
0 1 1 0	0 0 1 1 0 1 1 0
0 1 1 1	0 1 0 0 0 1 1 1
1 0 0 0	0 0 1 1 1 0 0 0
1 0 0 1	0 1 0 0 1 0 0 1
1 0 1 0	0 1 0 1 1 0 1 0
1 0 1 1	0 0 1 0 1 0 1 1
1 1 0 0	0 1 1 0 1 1 0 0
1 1 0 1	0 0 0 1 1 1 0 1
1 1 1 0	0 0 0 0 1 1 1 0
1 1 1 1	0 1 1 1 1 1 1 1

3.3.1 The Generator Matrix Simulation Model

The simulation model of the OFDM transmitter that uses Generator Matrix technique is shown in Figure 3.9 below.

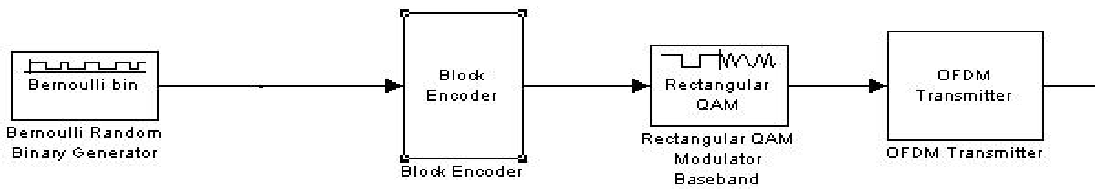

Figure 3.9: Transmitter of the Generator Matrix Model

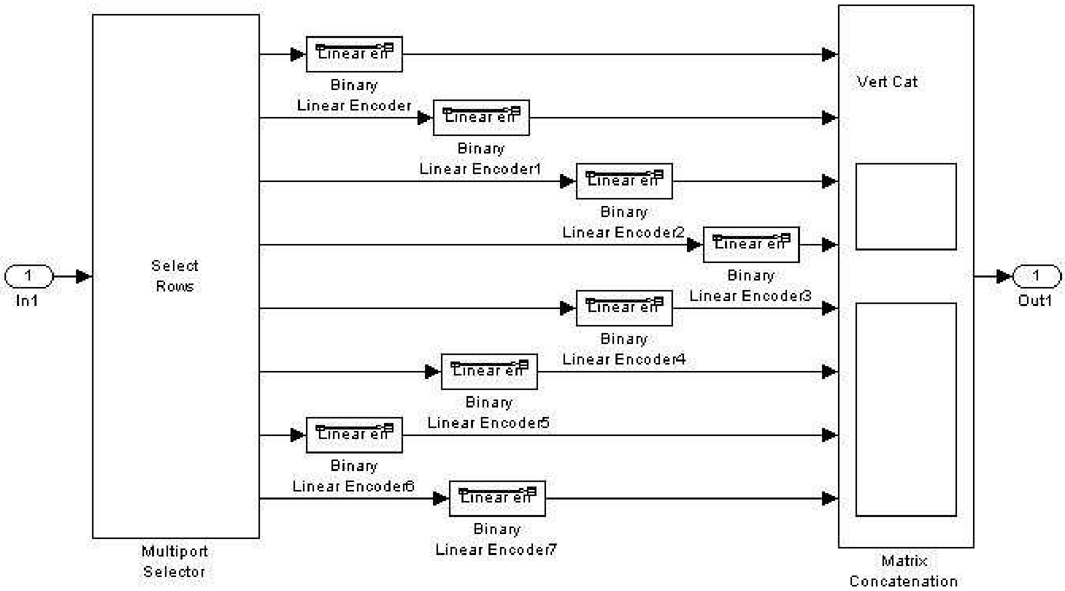

Figure 3.10: The Block Encoder

In the Block Encoder, the data was split into multiple four bits sequence. Each four bits sequence was encoded into an eight bits code word. Then, the code words were combined to be modulated via QAM. The Binary Linear Encoder blockset was used to contain the selected Generator Matrix algorithm. The Block Encoder and Decoder are shown in Figure 3.10 and 3.11 respectively.

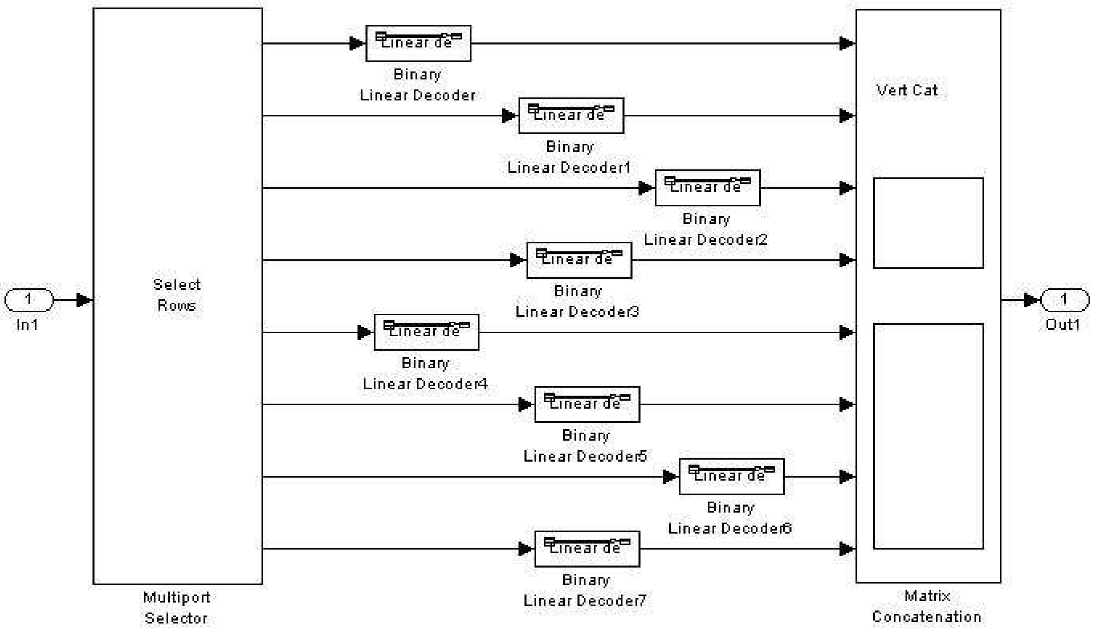

Figure 3.11: The Block Decoder

3.3.2 Methods to Find the Optimum Generator Matrix

These were the steps that have been used to find the Generator Matrix Codewords.

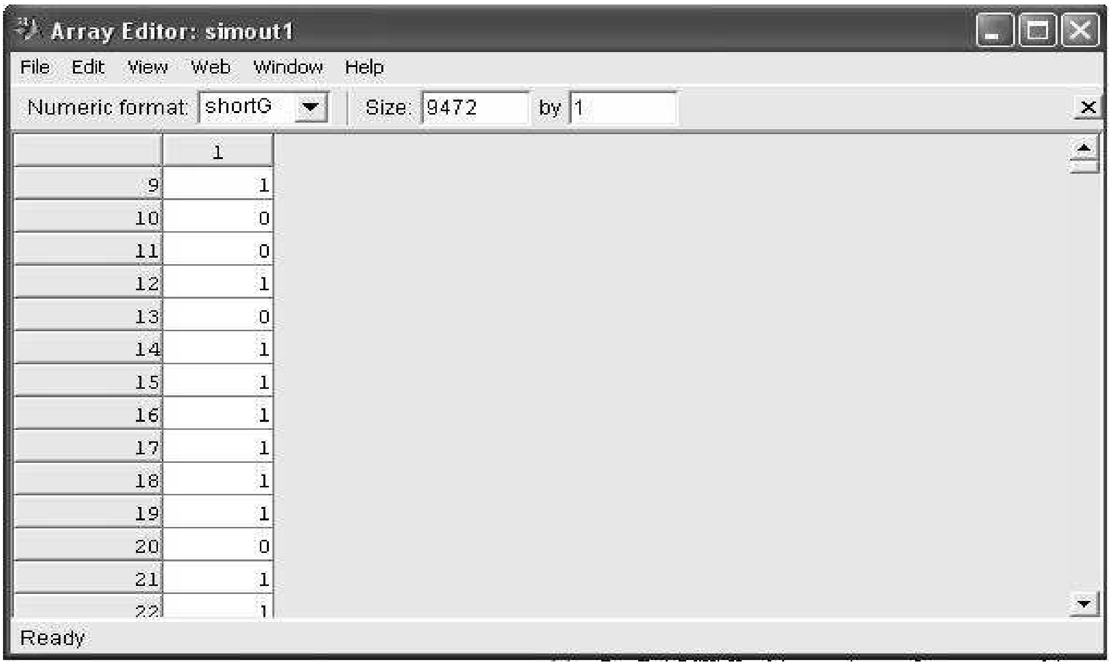

Figure 3.12: The Generated Data from the Uncoded Basic OFDM Model

1) Identify the data sequence that gives a high PEP.
2) This can be done by sending the generated data from the Simulink environment to the Matlab workspace as shown in Figure 3.12.

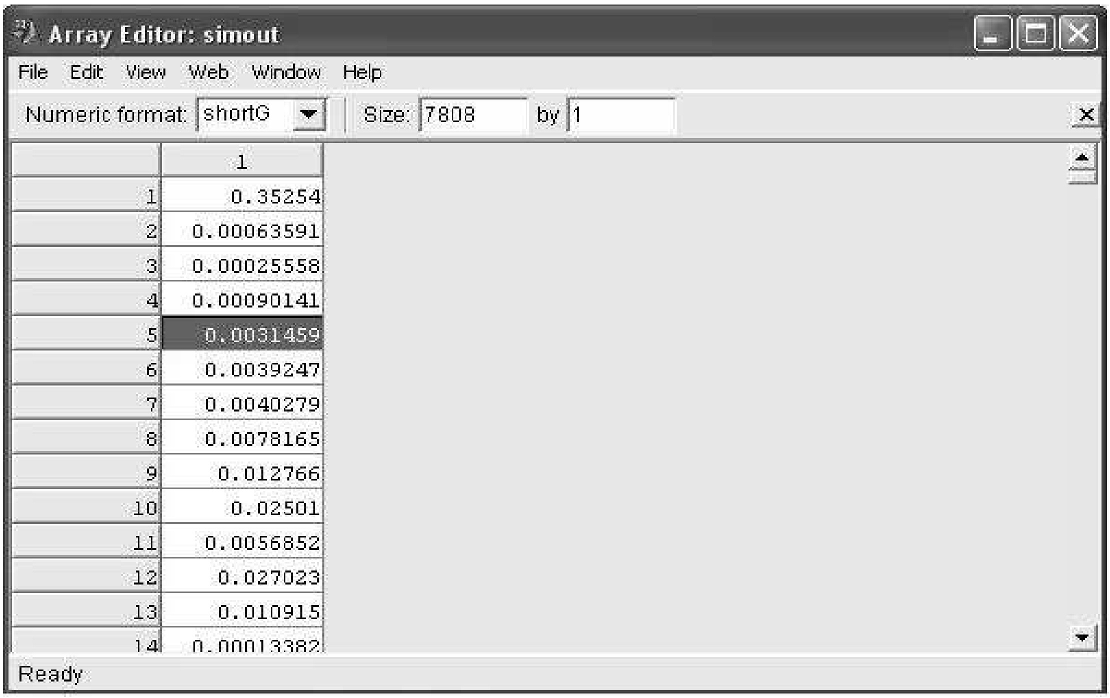

Figure 3.13: The PEP Value of the Data from Figure 3.12

3) Figure 3.13 shows the PEP value of the OFDM symbols.
4) By tracking the equivalent data sequence, the ones that give a high PEP can be identified.
5) Then, to create the optimum Generator Matrix, the d_{min} must first be set. If it is required to offer a one bit error correction capability, the d_{min} must equals to three.
6) Through exhaustive search that use the limitations of not generating the code word sequence that can give high PEP and using d_{min} equals to three, the optimum Generator Matrix can be created.

3.3.3 Generator Matrix (8, 3)

The Generator Matrix (8, 4) used a coding rate of 0.5. To analyze the effects of coding rate on the reduction of PAPR and the BER performance through the AWGN channel, Generator Matrix (8, 3) with coding rate of 0.375 was created. The Generator Matrix was chosen so that it could also offer a one bit error correction capability.

The Generator Matrix (8, 3) $\longrightarrow \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$

Table 4 shows the equivalent codewords for all the three bits data.

Table 4: Generator Matrix (8, 3) Equivalent Codewords

Data	Code words
0 0 0	0 0 0 0 0 0 0 0
0 0 1	1 0 0 0 1 0 0 1
0 1 0	0 1 0 0 1 0 1 0
0 1 1	1 1 0 0 0 0 1 1
1 0 0	1 0 1 1 1 1 0 0
1 0 1	0 0 1 1 0 1 0 1
1 1 0	1 1 1 1 0 1 1 0
1 1 1	0 1 1 1 1 1 1 1

The model used was the same as the Generator Matrix (8, 4) except the Generator Matrix algorithm in the Linear Block Encoder blockset was changed to Generator Matrix (8, 3) algorithm.

3.4 Look-up Table Technique

Figure 3.14 illustrates The Look-up Table Transmission Model. The binary data was converted to a four bit integer. Each integer was mapped to an eight bit integer code word. Then each integer was converted back to its equivalent eight bit binary value. The general process of OFDM can now proceed.

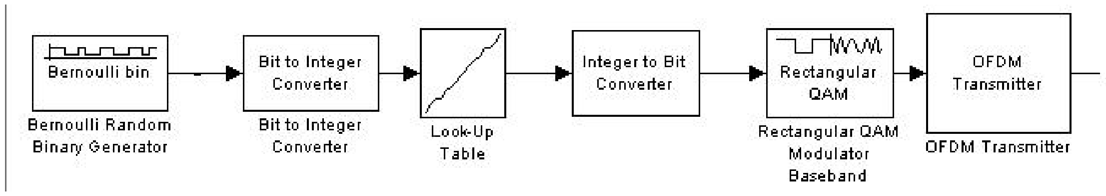

Figure 3.14: The Look-up Table Transmission Model

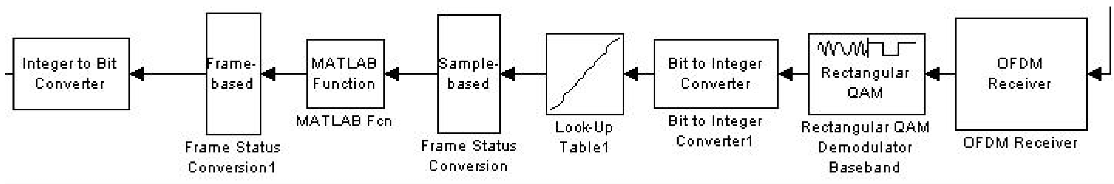

Figure 3.15: The Look-up Table Receiver Model

At the receiver part as shown in Figure 3.15, the demodulated data will be converted to eight bit integers. Through the look-up table, the equivalent mapping of the data will be done. The data will then be processed by the Matlab function of rounding any integer number that exceeds 2^4-1 to 15. This is because the AWGN channel might cause bit error to the data and resulted the four bit data to exceed the limit of 2^4 - 1. The data will then be converted back to binary representation for BER calculation. Table 5 shows the Look-up Table codewords. Coding rate of this Look-up Table technique is 0.5.

3.5 The Odd-bit Encoding Technique

Figure 3.16 illustrates the Odd-bit Encoding Transmission System. In this model, three bits will be encoded into four bits code word with the redundancy bit chosen to give the odd-bit parity encoding.

Table 5: The Look-up Table Codewords

Data (Binary)	Data (Integer)	Code words (Integer)	Code words (Binary)
0 0 0 0	0	3	0 0 0 0 0 0 1 1
0 0 0 1	1	29	0 0 0 1 1 1 0 1
0 0 1 0	2	46	0 0 1 0 1 1 1 0
0 0 1 1	3	48	0 0 1 1 0 0 0 0
0 1 0 0	4	72	0 1 0 0 1 0 0 0
0 1 0 1	5	86	0 1 0 1 0 1 1 0
0 1 1 0	6	101	0 1 1 0 0 1 0 1
0 1 1 1	7	123	0 1 1 1 1 0 1 1
1 0 0 0	8	132	1 0 0 0 0 1 0 0
1 0 0 1	9	154	1 0 0 1 1 0 1 0
1 0 1 0	10	169	1 0 1 0 1 0 0 1
1 0 1 1	11	183	1 0 1 1 0 1 1 1
1 1 0 0	12	207	1 1 0 0 1 1 1 1
1 1 0 1	13	209	1 1 0 1 0 0 0 1
1 1 1 0	14	226	1 1 1 0 0 0 1 0
1 1 1 1	15	252	1 1 1 1 1 1 0 0

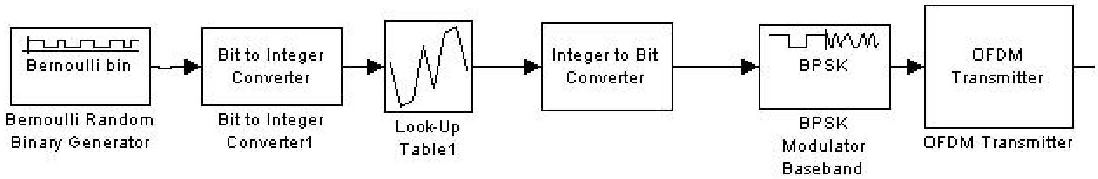

Figure 3.16: Odd-bit Encoding Transmission System

Table 6: The Look-up Table for the Odd-Bit Encoding Model

Data (Binary)	Data (Integer)	Code word (Integer)	Code word (Binary)
0 0 0	0	8	1 0 0 0
0 0 1	1	1	0 0 0 1
0 1 0	2	2	0 0 1 0
0 1 1	3	11	1 0 1 1
1 0 0	4	4	0 1 0 0
1 0 1	5	13	1 1 0 1
1 1 0	6	14	1 1 1 0
1 1 1	7	7	0 1 1 1

3.6 Graphical User Interface for the Simulation Model

Figure 3.17 shows the Graphical User Interface that has been developed for results analysis. While Figure 3.18 shows the flow to use the GUI.

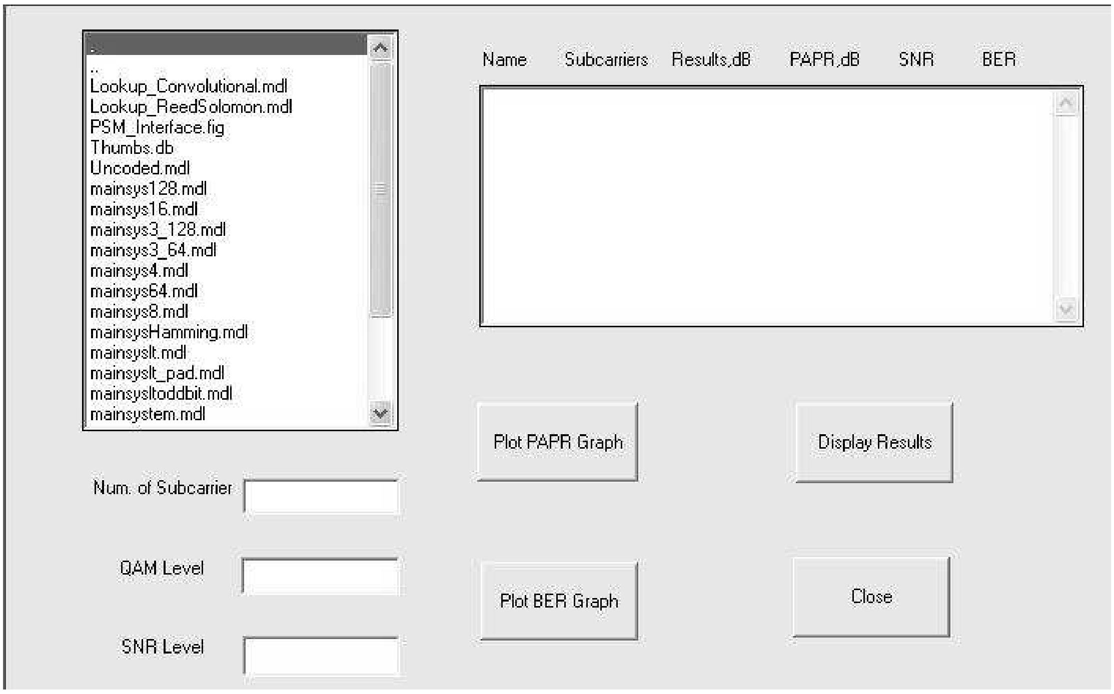

Figure 3.17: Graphical User Interface for Results Analysis

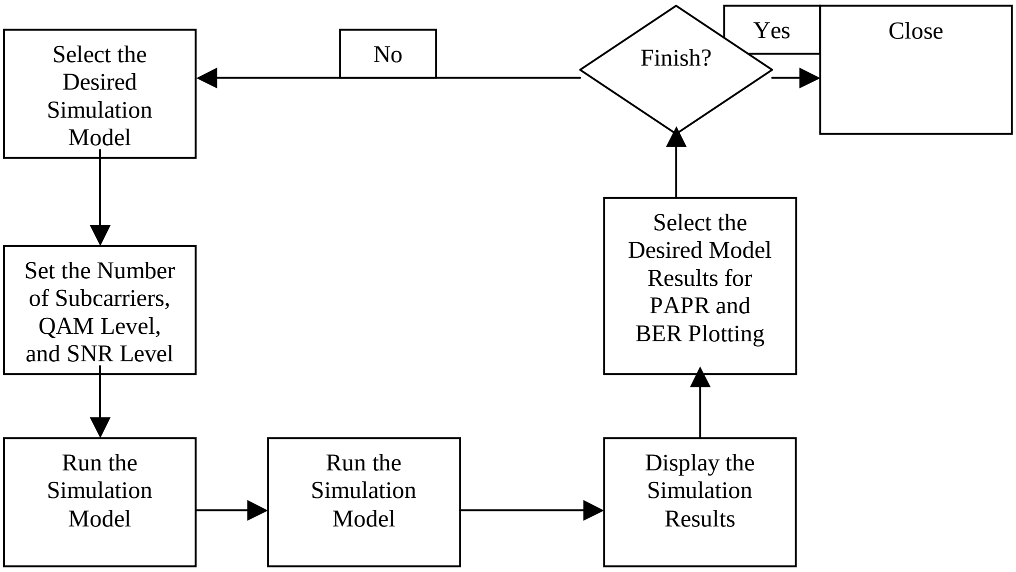

Figure 3.18: Flowchart of the GUI

3.6.1 Step by Step Guide to Use the GUI

These were the steps to use the GUI for running the Simulation Model and results analysis.

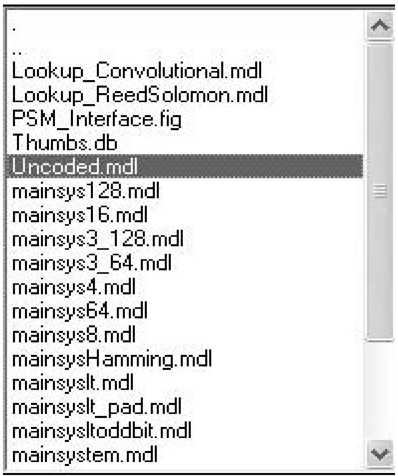

Figure 3.19: The Simulation Model Listbox

1. Figure 3.19 shows the Uncoded Basic OFDM Model being selected. Double click on the selected model; the GUI will open the .mdl file of the selected model.

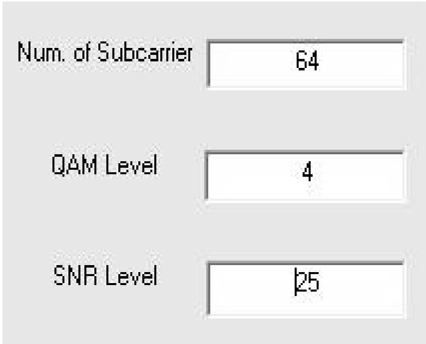

Figure 3.20: Simulation Parameters Setting

2. Next, set the following simulation parameters; Number of Subcarriers, QAM Level, and SNR Level of the AWGN channel. Press ‘Enter’ after setting each parameter as shown in Figure 3.20.
3. Run the simulation in the Simulink environment.
4. Click the ‘Display Result’ button; the simulation result will be displayed as shown in Figure 3.21.

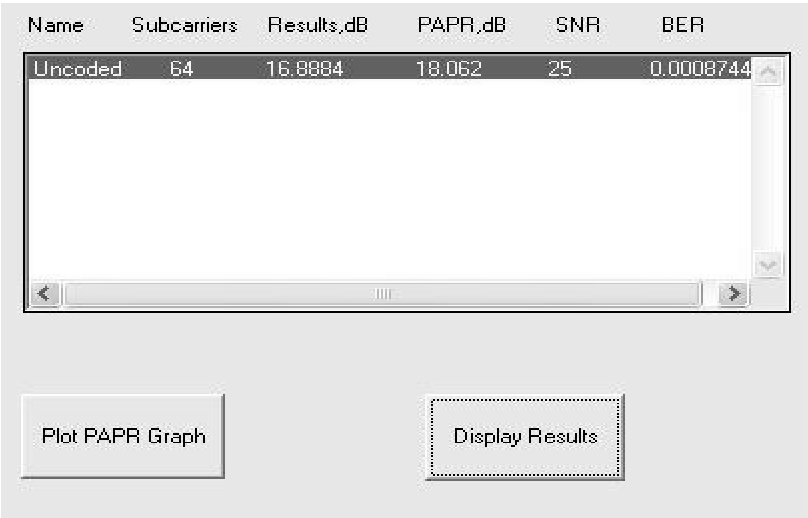

Figure 3.21: The Simulation Results

5. Select the desired simulation results and plot the PAPR or BER graph as shown in Figure 3.22 and Figure 3.23.

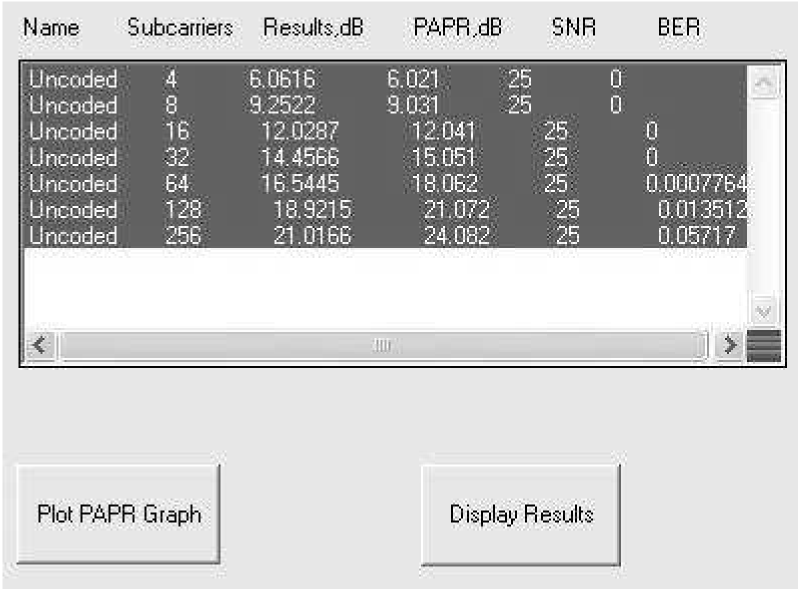

Figure 3.22: Selected Simulation Results

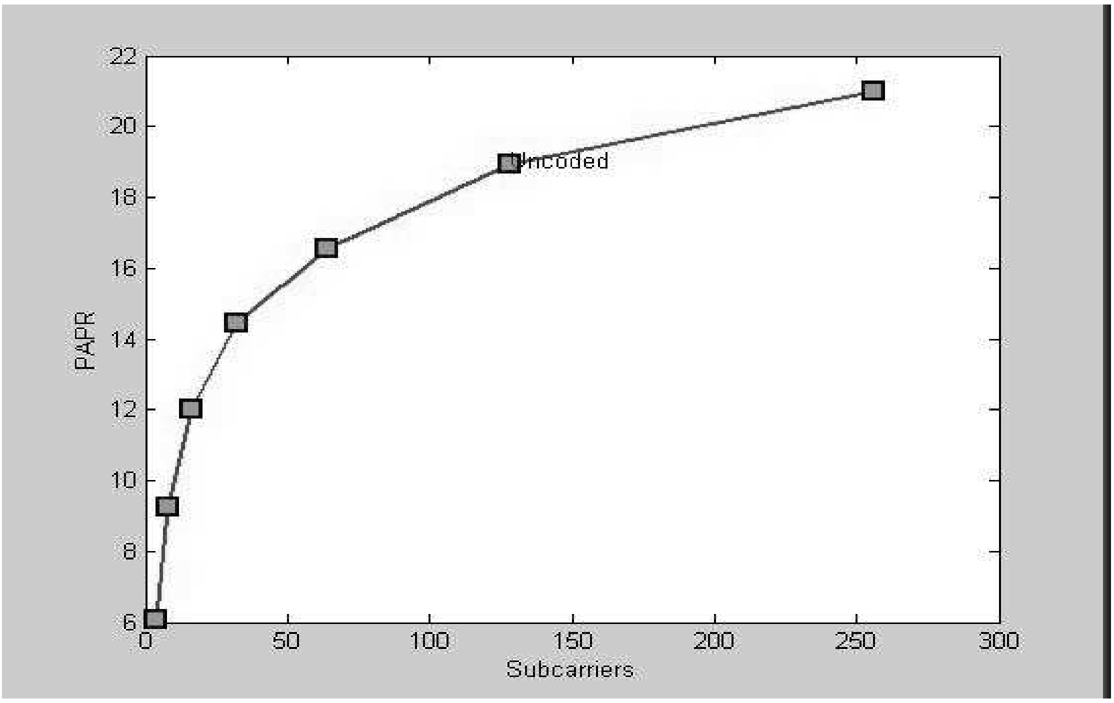

Figure 3.23: The Resultant PAPR Graph

6. Click on the ‘Close’ button if don’t intend to continue.

CHAPTER IV

RESULTS AND DISCUSSIONS

4.0 Peak Envelope Power (PEP) of the OFDM Simulation Model

Figure 4.1 illustrates the Peak Envelope Power (PEP) of the Uncoded OFDM signals. As shown, the OFDM signals give a peak that can yield up to one (non-normalized value) while the average peak is below 0.5. The concept of block coding is to control this peak so that the ratio with the average value is small as will be illustrated in Figure 4.2.

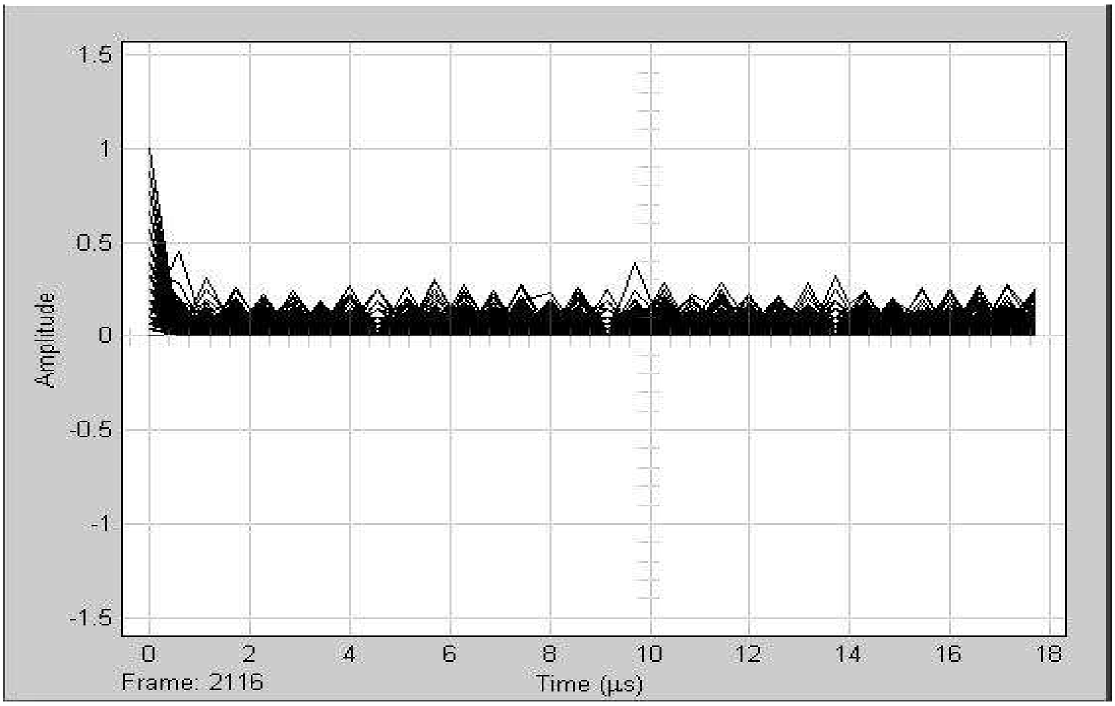

Figure 4.1: PEP of the Uncoded OFDM Signals

Figure 4.2 shows the PEP of the coded OFDM signals using Generator Matrix. It is shown that the maximum peak of the OFDM signals did not reach the value of one. Therefore the PAPR has been reduced. Figure 4.3 illustrates the PEP of the coded OFDM signals using Look-up Table technique. The maximum peak of the signal didn't even reach 0.5 showing that the Look-up Table technique more efficiently reduces the PAPR compared to the Generator Matrix technique. Figure 4.4 shows the PEP of the Odd-bit Encoded OFDM signals. As the figure shows, the PEP has been reduced to less than 0.5.

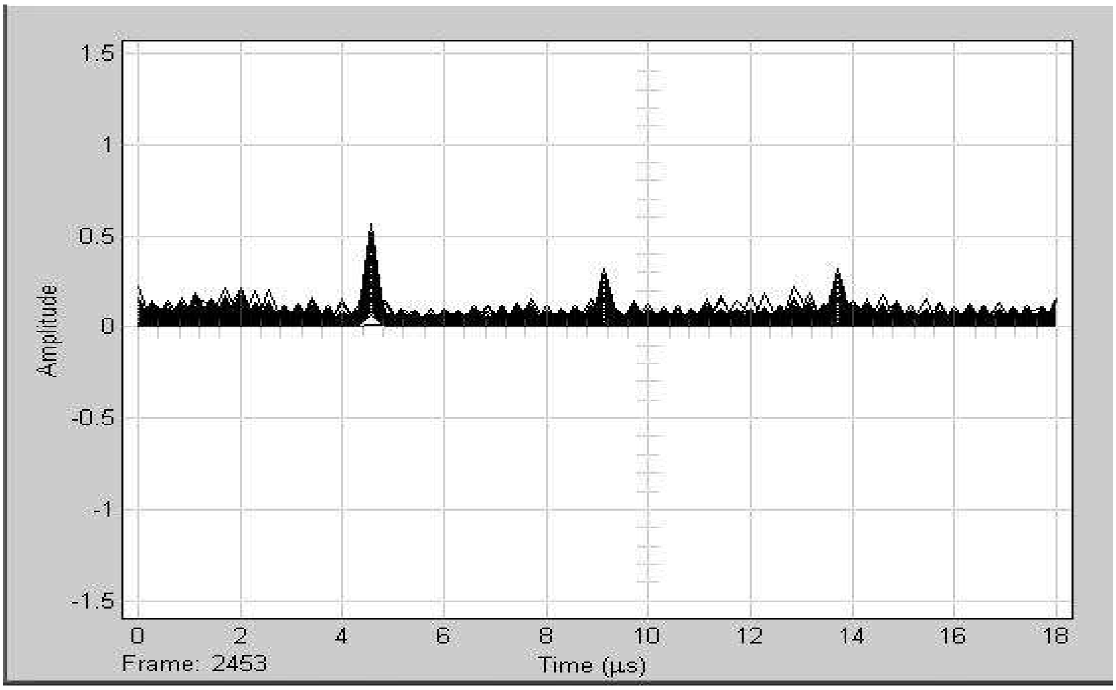

Figure 4.2: PEP of the Generator Matrix Coded OFDM Signal

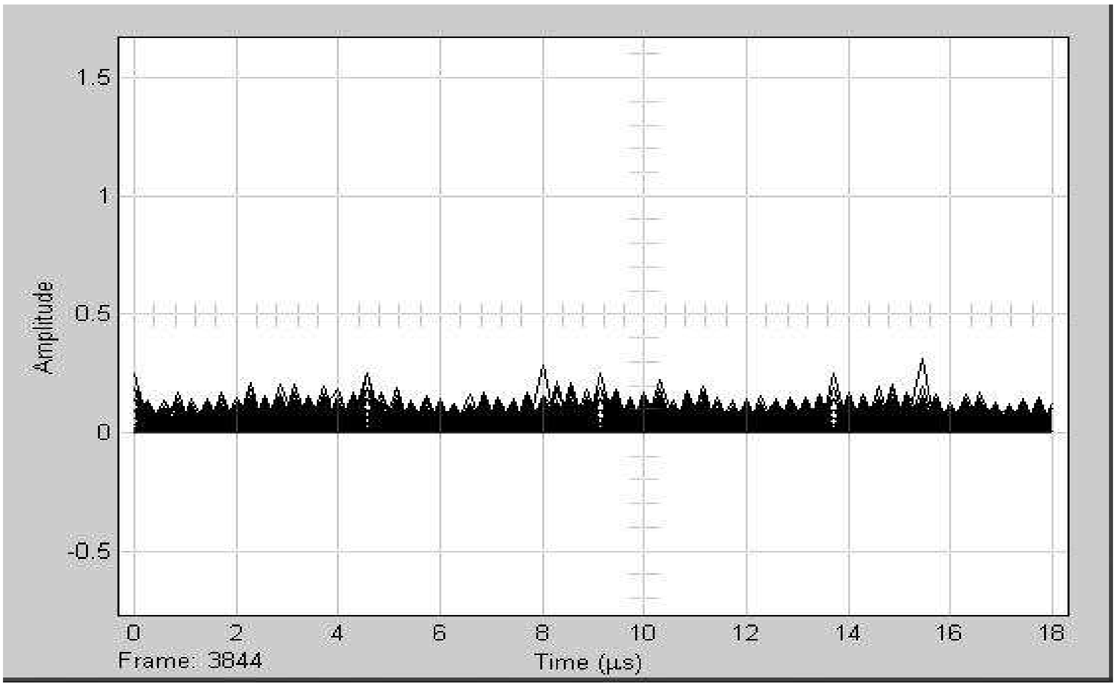

Figure 4.3: PEP of the Look-up Table Coded OFDM Signals

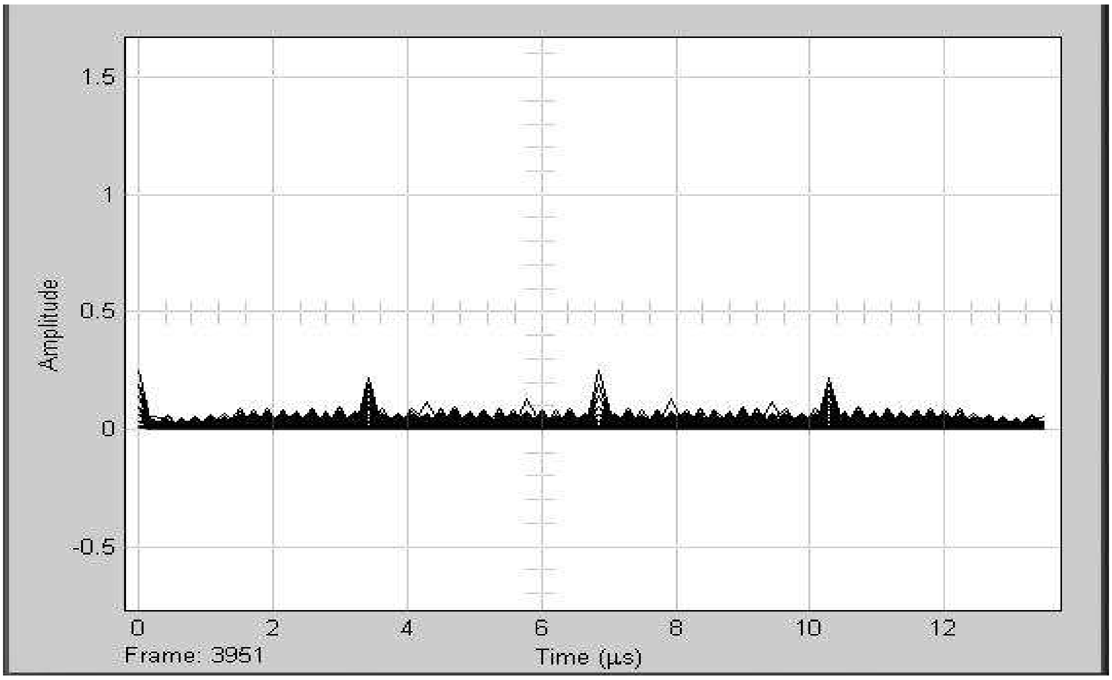

Figure 4.4: PEP of the Odd-bit Encoded OFDM Signals

4.1 Complex Baseband Analysis of the OFDM Signals

The complex baseband waveform is plotted according to the complex signal values of the real and imaginary parts. The generated waveform is based on the OFDM system that has the following parameters set:

1. Subcarriers = 32
2. Symbols observed = 17504
3. Code rate = 0.5

Figure 4.5 illustrates the waveform of the Uncoded OFDM signals. As shown, the OFDM signal values were not controlled and not concentrated. This means that the PEP of the signals is not controlled and can yield up to a very high value.

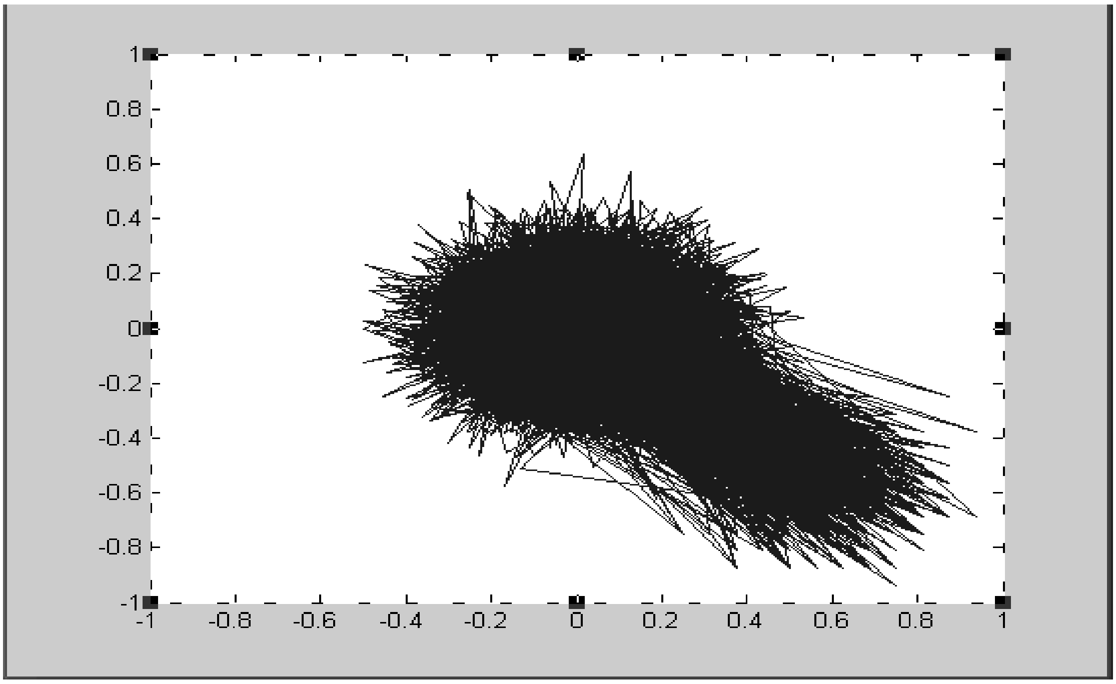

Figure 4.5: Complex Baseband Waveform of the Uncoded OFDM Signals

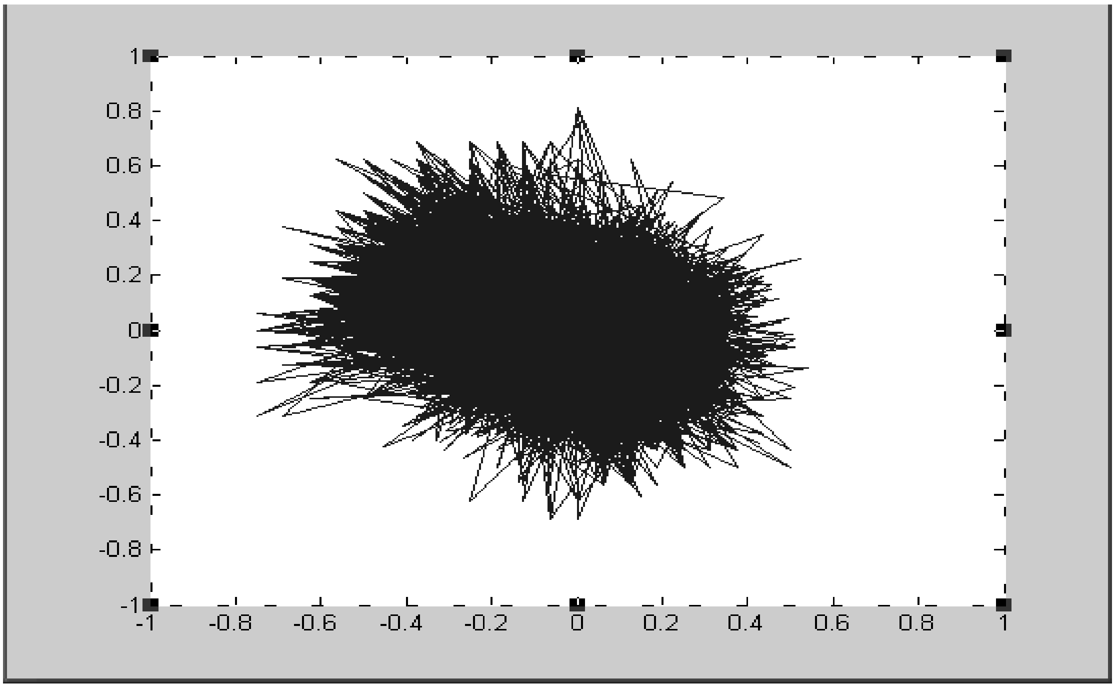

Figure 4.6: Complex Baseband Waveform of the Generator Matrix Encoded OFDM Signals

Figure 4.6 shows the complex baseband waveform of the ODFM signals that has been pre-encoded using Generator Matrix technique. Clearly, the complex OFDM signals have been controlled and are more concentrated. This will yield a better and lower PAPR.

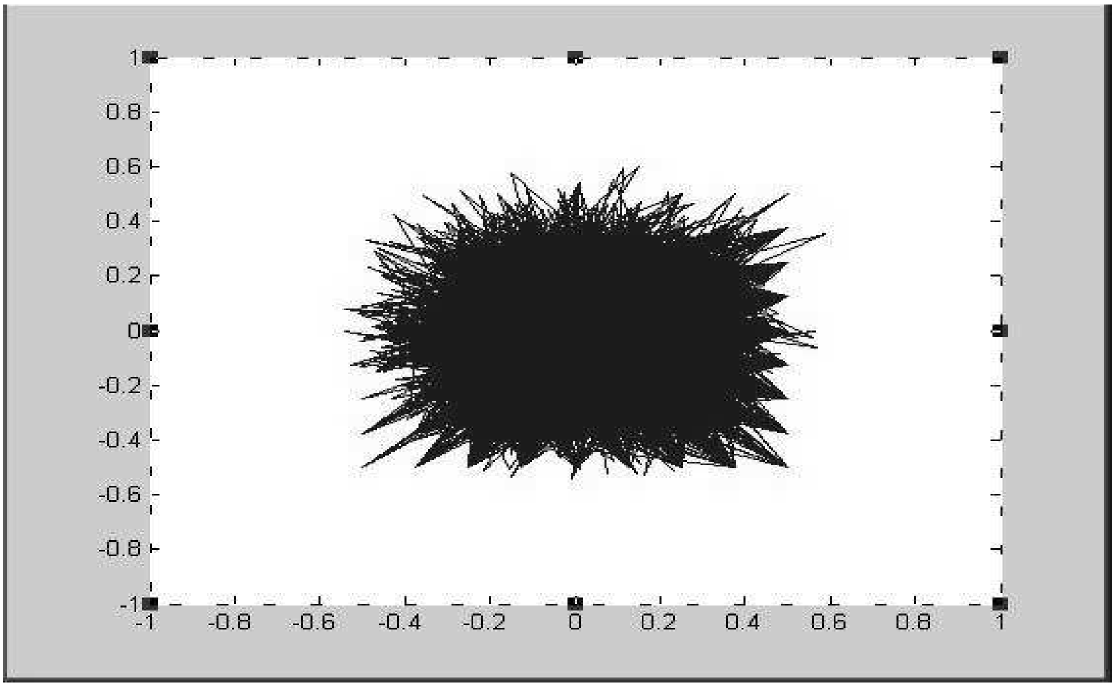

Figure 4.7: Complex Baseband Waveform of the Look-up Table Encoded OFDM Signals

The figure above obviously shows a better signal performance in terms of its controlled complex values. Therefore, these waveform analyses can summarize that the Look-up Table technique can give a better PAPR reduction performance compared to the Generator Matrix technique.

4.2 Simulation Results

The following are the simulation results by using different block coding techniques to reduce the PAPR. The BER reduction performance is also shown in Table 7. It shows the PAPR values for varying subcarriers from 4 to 256 using different Linear Block Coding techniques. From the table, the Look-up Table technique gives a better PAPR reduction performance compared to other techniques. The Generator Matrix (8, 3) that gives the coding rate of 0.375 exhibits better PAPR reduction compared to the Generator Matrix (8, 4) that uses the coding rate of 0.5.

Table 8 shows the Bit Error Rate given by the Generator Matrix that uses different coding rate. The SNR of the AWGN channel varies from 0 to 26 dB. Figure 4.8 shows the CDF graph for a 64 subcarriers simulation model. From the graph, the Look-up table limits the PAPR value to only 13 dB while the Generator Matrix technique limits the PAPR value to 15 dB compared to the Uncoded General OFDM System that can gives a PAPR as high as 17 dB for a 64 subcarriers system. This is due to the fact that, the Look-up table technique uses codewords mapping that can map any data to any codewords that gives a low PEP. As for Generator Matrix technique, this aspect cannot be met as the codewords that has been generated is bound to the constrain of the Generator Matrix sequence itself which is selected based on the d_{min} chosen and the rules of creating the Generator Matrix.

Table 7: PAPR Values on Different Number of Subcarriers

PAPR(dB) (no. of subcarriers)	Look-up Table (8,4)	Generator Matrix (8,4), dB	Odd-bit BPSK (4,3), dB	Generator Matrix (8,3), dB
6.021 (4)	0	6.023	0.7621	6.004
9.031(8)	7.66	9.028	5.841	9.465
12.041 (16)	10.66	11.79	9.208	12.28
15.051 (32)	11.84	12.88	10.77	12.83
18.062 (64)	13.03	15.13	13.88	13.16
21.072 (128)	15.05	17.36	16.94	15.35
24.082 (256)	17.21		19.97	

Table 8: Bit Error Rate on Different SNR Values

Code Rate = 0.5 Code Rate = 0.375

SNR, dB	Generator Matrix(8,4)	Generator Matrix(8,3)	Uncoded
0	0.4214	0.3886	0.401
2	0.3993	0.3601	0.3768
4	0.3702	0.3238	0.3456
6	0.3303	0.2789	0.3088
8	0.2796	0.2242	0.2648
10	0.2151	0.1636	0.2147
12	0.1416	0.101	0.1602
14	0.07224	0.0493	0.1051
16	0.02458	0.01564	0.05696
18	0.004701	0.002792	0.02365
20	0.0004293	0.000212	0.006132
22	0	0	0.0008528
24	0	0	0.00004534
26	0	0	0

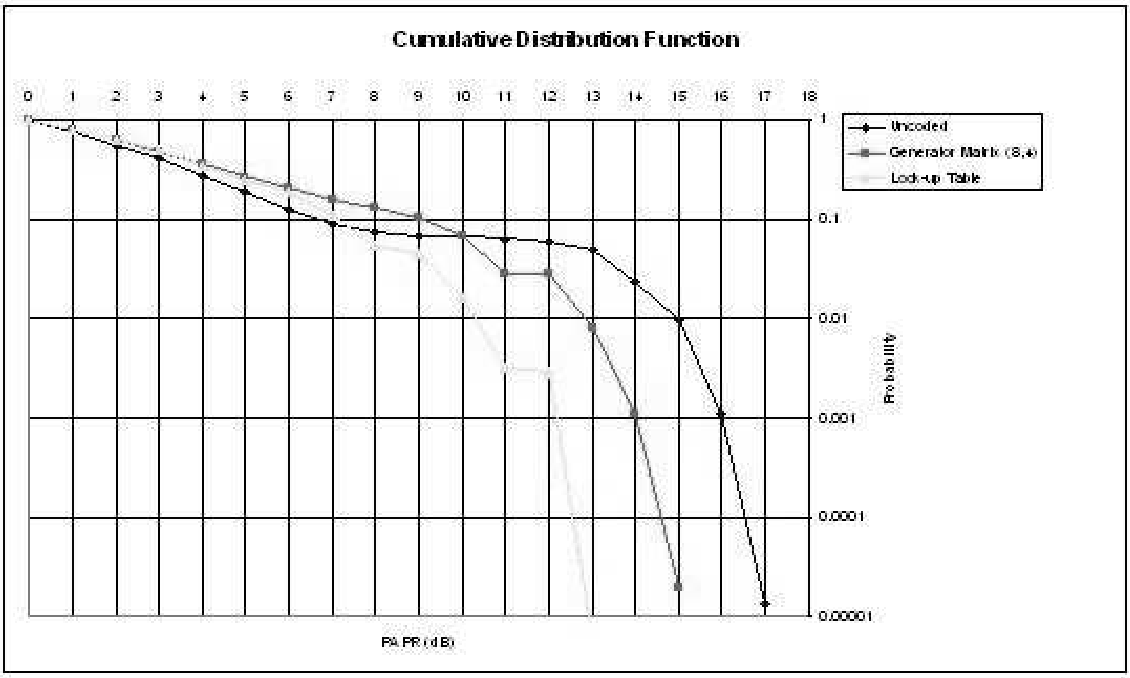

Figure 4.8: PAPR Cumulative Distribution Function for 64 Subcarriers

Figure 4.9 shows the lower the coding rate the performance in terms of PAPR reduction is better. The PAPR doesn't exceed 13 dB for Generator Matrix (8, 3) with coding rate of 0.375. This is because by increasing the number of redundancy bits, the combination sequence that can be generated to give a low PEP has increased. Therefore, lower PAPR can be achieved using a lower coding rate. Figure 4.10 shows the BER performance comparison for different coding rate using Generator Matrix technique. The Generator Matrix that uses 0.375 code rate shows a better BER performance.

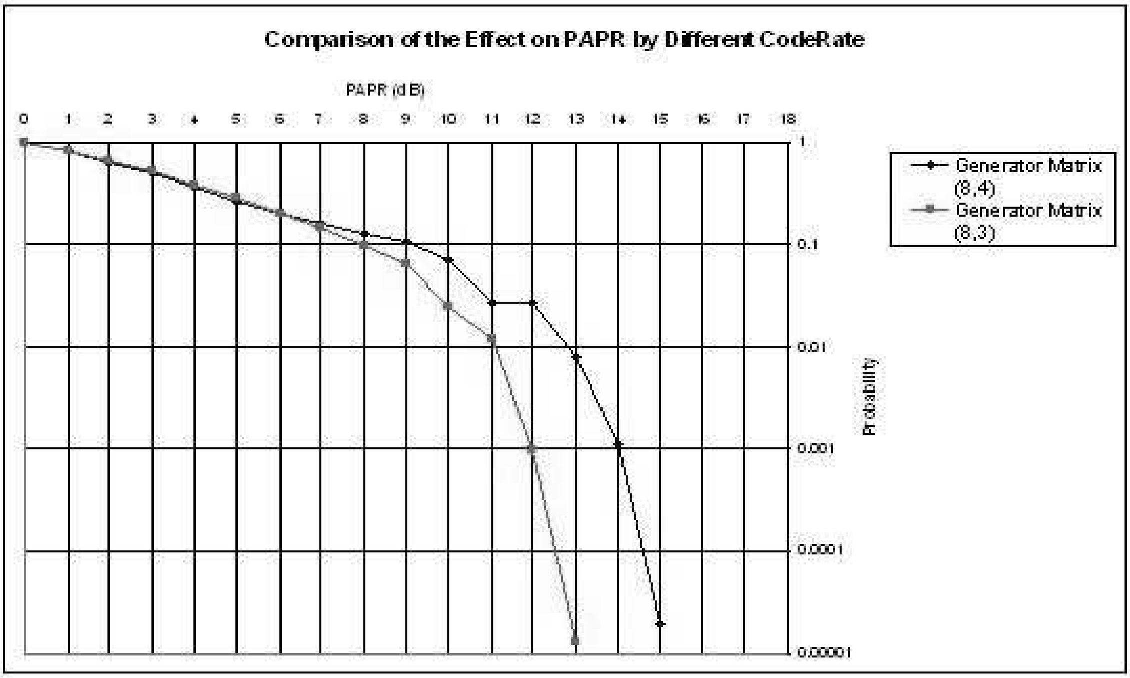

Figure 4.9: Effects on PAPR by using Different Code Rate

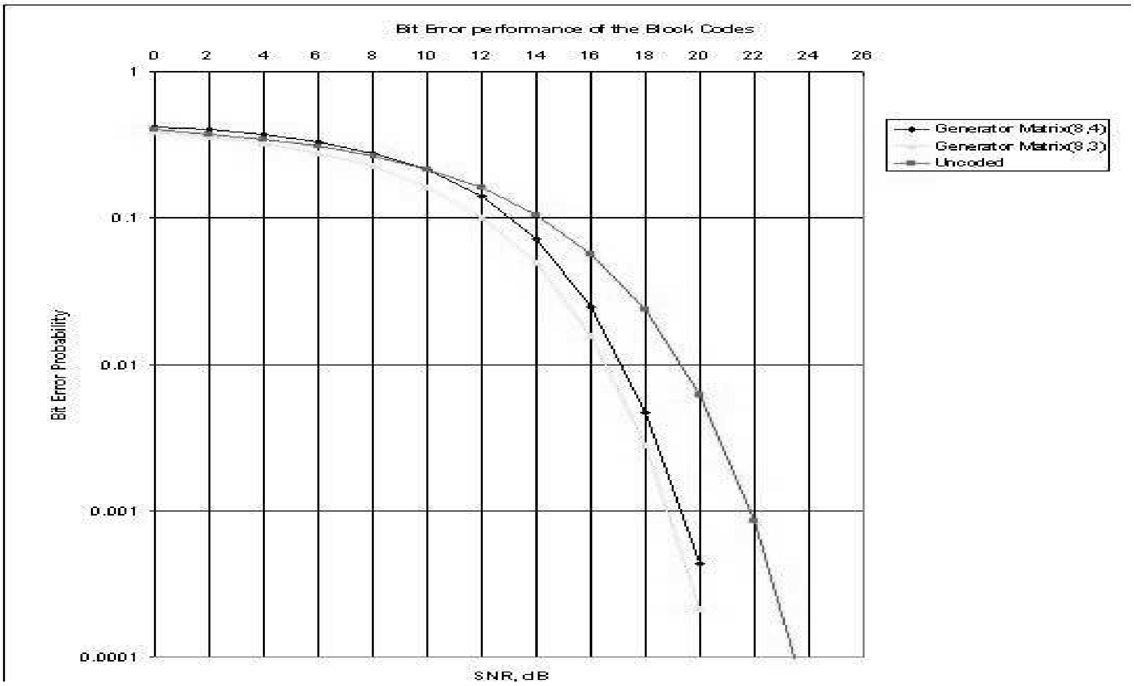

Figure 4.10: BER Distribution for the 32 Subcarriers 4-QAM-OFDM with (8, 4) and (8, 3) Block Codes

CHAPTER V

CONCLUSIONS AND RECOMMENDATIONS

5.0 Conclusions

The current status of the research is that OFDM appears to be a leading candidate for high performance wireless telecommunications. Although due to the fact that it reduces ISI and multipath distortion while giving a very high spectral efficiency, it also exhibits high PAPR that severely limits its popularity.

In this report, it has proven that Block Coding can reduce PAPR without much complexity and has the ability for error correction using certain Block Coding technique. There are several Block Coding techniques which are Generator Matrix Encoding, Look-up Table Mapping, and Odd-bit Parity Encoding. Each technique has its own trade offs and its suitability is based on the application requirements.

Generator Matrix gives a higher PAPR value compared to the Look-up Table technique but it can provides bit error correction capability. Therefore, a lower BER can be achieved compared to Look-up Table technique. Results show that the Look-up Table technique gives the lowest PAPR value in any number of subcarriers. The only trade-off will be that the Look-up Table technique cannot correct the bit error that occurs during transmission with its encoding code. The system needs other error correction techniques to be added into it to give a lower BER. This adds to the complexity and cost of the system. The Odd-bit Parity Encoding is another alternative to reduce PAPR with its advantages of simple encoding algorithm while still provide an average PAPR reduction performance and error detection capability.

Coding rate also gives impact on the reduction of PAPR. The lower the coding rate, the PAPR reduction is shown more effective but the spectral efficiency is dropped due to the code rate.

The pros and cons of each technique provide a suitable reference point in using Block Coding as the PAPR reduction scheme for the OFDM system. It shall be chose according to application needs in order to set the cost to the minimum and still provide the optimum PAPR reduction. With that, OFDM promises to be suitable modulation technique for high capacity wireless transmission system and the choice of the future when the wireless networks become the essential part of telecommunications.

5.1 Future Works

Although this report shows that the Block Coding technique can be used to reduce the PAPR in the OFDM system with ease, there are still certain areas for analysis that must be done before being implemented in the real world application design. Here are some suggestions for future work.

- Design simulation models by implementing the Block Coding scheme into the real world application standard for example Hiperlan2 standard
- The Graphical User Interface developed in this report can still be modified to provide enhance analysis function to the users.
- Uses other techniques to find the most optimum codewords that give lower PAPR values.
- Set the d_{min} of the Generator Matrix sequence to five and find the suitable codewords so that it can provide two error correction capabilities while still reducing the PAPR values.

REFERENCES

1. Richard D.J. van Nee, ‘OFDM codes for Peak-to- Average Power Reduction and Error Correction’, Global Telecommunications Conference, 1996. GLOBECOM’96.
2. Simon Shepherd, John Orriss, and Stephen Barton, ‘Asymptotic Limits in Peak Envelope Power Reduction by Redundant Coding in OFDM’, IEEE Transactions on Communications, Vol. 46, No. 1, January 1998.
3. Pingyi Fan and Xiang-Gen Xia, ‘Block Coded Modulation for the Reduction of the Peak to Average Power Ratio in OFDM Systems’, Wireless Communications and Networking Conference, 1999, WCNC.
4. Hideki Ochiai and Hideki Imai, ‘Block Codes for Frequency Diversity and Peak Power Reduction in Multicarrier Systems’, 1998 IEEE International Symposium, August 1998.
5. Hideki Ochiai and Hideki Imai,’Performance of Block Codes with Peak Power Reduction for Indoor Multicarrier Systems’, Proc. IEEE VTC ’98, May 1998.
6. Hideki Ochiai and Hideki Imai,’MDPSK-OFDM with Highly Power-Efficient Block Codes for Frequency-Selective Fading Channels’, IEEETransactions on Vehicular Technology, Vol.49, No.1, January 2000.
7. A.E. Jones, T.A. Wilkinson and S.K. Barton, ‘Block Coding Scheme for Reduction of PAPR of Multicarrier Transmission Scheme’, Electronics Letters, Volume: 30, Issue: 25, 8 Dec. 1994.
8. Steve Halford, Intersil, Karen Halford, Doc H^2 Consulting, ‘OFDM Uncovered Part 2: Design Challenges’, www.Commsdesign.com.

APPENDIX A

GRAPHICAL USER INTERFACE SOURCE CODE

```
function varargout = PSM_Interface(varargin)
% PSM_INTERFACE Application M-file for PSM_Interface.fig
%    FIG = PSM_INTERFACE launch PSM_Interface GUI.
%    PSM_INTERFACE('callback_name', ...) invoke the named callback.

% Last Modified by GUIDE v2.0 01-Dec-2002 13:55:19
if nargin <= 1   % LAUNCH GUI
        if nargin == 0
                initial_dir = pwd;
        elseif nargin == 1 & exist(varargin{1},'dir')
                initial_dir = varargin{1};
        else
                errordlg('Input argument must be a valid directory','Input Argument Error!')
                return
        end
        % Open FIG-file
        fig = openfig(mfilename,'reuse');     % Generate a structure of handles to pass to callbacks,
and store it.
        handles = guihandles(fig);
        guidata(fig, handles);
        % Populate the listbox
        load_listbox(initial_dir,handles)
        % Return figure handle as first output argument
        if nargout > 0
                varargout{1} = fig;
        end

elseif ischar(varargin{1}) % INVOKE NAMED SUBFUNCTION OR CALLBACK

        try
                if (nargout)
                        [varargout{1:nargout}] = feval(varargin{:}); % FEVAL switchyard
                else
                        feval(varargin{:}); % FEVAL switchyard
                end
        catch
                disp(lasterr);
        end

end
```

```
%| ABOUT CALLBACKS:
%| GUIDE automatically appends subfunction prototypes to this file, and
%| sets objects' callback properties to call them through the FEVAL
%| switchyard above. This comment describes that mechanism.
%|
%| Each callback subfunction declaration has the following form:
%| <SUBFUNCTION_NAME>(H, EVENTDATA, HANDLES, VARARGIN)
%|
%| The subfunction name is composed using the object's Tag and the
%| callback type separated by '_', e.g. 'slider2_Callback',
%| 'figure1_CloseRequestFcn', 'axis1_ButtondownFcn'.
%|
%| H is the callback object's handle (obtained using GCBO).
%|
%| EVENTDATA is empty, but reserved for future use.
%|
%| HANDLES is a structure containing handles of components in GUI using
%| tags as fieldnames, e.g. handles.figure1, handles.slider2. This
%| structure is created at GUI startup using GUIHANDLES and stored in
%| the figure's application data using GUIDATA. A copy of the structure
%| is passed to each callback.  You can store additional information in
%| this structure at GUI startup, and you can change the structure
%| during callbacks.  Call guidata(h, handles) after changing your
%| copy to replace the stored original so that subsequent callbacks see
%| the updates. Type "help guihandles" and "help guidata" for more
%| information.
%|
%| VARARGIN contains any extra arguments you have passed to the
%| callback. Specify the extra arguments by editing the callback
%| property in the inspector. By default, GUIDE sets the property to:
%| <MFILENAME>('<SUBFUNCTION_NAME>', gcbo, [], guidata(gcbo))
%| Add any extra arguments after the last argument, before the final
%| closing parenthesis.



% --------------------------------------------------------------------
function varargout = listbox_Models_Callback(h, eventdata, handles, varargin)
set(handles.Num_Subcarrier,'Visible','on')
set(handles.QAM_Level,'Visible','on')
if strcmp(get(handles.figure1,'SelectionType'),'open')
        index_selected = get(handles.listbox_Models,'Value');
        file_list = get(handles.listbox_Models,'String');
        filename = file_list{index_selected};
        if  handles.is_dir(handles.sorted_index(index_selected))
                cd (filename)
                load_listbox(pwd,handles)
        else
          [path,name,ext,ver] = fileparts(filename);
          switch ext
          case '.fig'
                  guide (filename)
          otherwise
                try
                open(filename)
                catch
                        errordlg(lasterr,'File Type Error','modal')
                end
          end
```

```
  end
   if isequal(filename,'mainsystem.mdl')
      set(handles.Num_Subcarrier,'Visible','off')
   end
   if isequal(filename,'mainsys4.mdl')
      set(handles.Num_Subcarrier,'Visible','off')
   end
   if isequal(filename,'mainsys8.mdl')
      set(handles.Num_Subcarrier,'Visible','off')
   end
   if isequal(filename,'mainsys16.mdl')
      set(handles.Num_Subcarrier,'Visible','off')
   end
   if isequal(filename,'mainsys64.mdl')
      set(handles.Num_Subcarrier,'Visible','off')
   end
   if isequal(filename,'mainsys128.mdl')
      set(handles.Num_Subcarrier,'Visible','off')
   end
   if isequal(filename,'mainsysHamming.mdl')
      set(handles.Num_Subcarrier,'Visible','off')
   end
   if isequal(filename,'mainsyslt_pad.mdl')
      set(handles.Num_Subcarrier,'Visible','off')
      set(handles.QAM_Level,'Visible','off')
   end
   if isequal(filename,'mainsysltoddbit.mdl')
      set(handles.QAM_Level,'Visible','off')
   end
   if isequal(filename,'mainsysltoddbit1.mdl')
      set(handles.QAM_Level,'Visible','off')
   end
   if isequal(filename,'mainsysltoddbit2.mdl')
      set(handles.QAM_Level,'Visible','off')
   end
end


% --------------------------------------------------------------------
function varargout = Num_Subcarrier_Callback(h, eventdata, handles, varargin)
   index_selected = get(handles.listbox_Models,'Value');
          file_list = get(handles.listbox_Models,'String');
          filename = file_list{index_selected};
   NewStrVal = get(h,'String');
   NewVal = str2double(NewStrVal);
   [path,name,ext,ver] = fileparts(filename);
   if isequal(name,'Uncoded')
      if ~isequal(NewStrVal,'4') & ~isequal(NewStrVal,'8') & ~isequal(NewStrVal,'16') &
~isequal(NewStrVal,'32') & ...
      ~isequal(NewStrVal,'64') & ~isequal(NewStrVal,'128') & ~isequal(NewStrVal,'256')
         errordlg('Number of subcarriers must be a value power of 2s')
      end
   end
   if isequal(name,'mainsyslt')
      if ~isequal(NewStrVal,'4') & ~isequal(NewStrVal,'8') & ~isequal(NewStrVal,'16') &
~isequal(NewStrVal,'32') & ...
      ~isequal(NewStrVal,'64') & ~isequal(NewStrVal,'128') & ~isequal(NewStrVal,'256')
         errordlg('Number of subcarriers must be a value power of 2s')
      end
   end
```

```
    if isequal(name,'mainsysltoddbit')
        if ~isequal(NewStrVal,'4') & ~isequal(NewStrVal,'8') & ~isequal(NewStrVal,'16') &
~isequal(NewStrVal,'32') & ...
        ~isequal(NewStrVal,'64') & ~isequal(NewStrVal,'128') & ~isequal(NewStrVal,'256')
            errordlg('Number of subcarriers must be a value power of 2s')
        end
    end
    if isequal(name,'mainsysltoddbit1')
        if ~isequal(NewStrVal,'4') & ~isequal(NewStrVal,'8') & ~isequal(NewStrVal,'16') &
~isequal(NewStrVal,'32') & ...
        ~isequal(NewStrVal,'64') & ~isequal(NewStrVal,'128') & ~isequal(NewStrVal,'256')
            errordlg('Number of subcarriers must be a value power of 2s')
        end
    end
    if isequal(name,'mainsysltoddbit2')
        if ~isequal(NewStrVal,'4') & ~isequal(NewStrVal,'8') & ~isequal(NewStrVal,'16') &
~isequal(NewStrVal,'32') & ...
        ~isequal(NewStrVal,'64') & ~isequal(NewStrVal,'128') & ~isequal(NewStrVal,'256')
            errordlg('Number of subcarriers must be a value power of 2s')
        end
    end
    if isequal(name,'mainsysltoddbit')
        NewStrVal = NewVal/4
        NewStrVal = NewStrVal*3
    end
    if isequal(name,'mainsysltoddbit2')
        NewStrVal = NewVal/4
        NewStrVal = NewStrVal*3
    end
    if isequal(name,'mainsysltoddbit1')
        NewStrVal = NewVal/8
        NewStrVal = NewStrVal*7
    end
    if isequal(name,'Uncoded')
        NewStrVal = NewVal*2
    end
    dest=([name,'/Bernoulli Random Binary Generator'])
    Num = get_param(dest,'MaskValues');
    NewStrVal = num2str(NewStrVal)
    Num(5) = {NewStrVal}
    set_param(dest,'MaskValues',Num)


% --------------------------------------------------------------------
function varargout = QAM_Level_Callback(h, eventdata, handles, varargin)
    index_selected = get(handles.listbox_Models,'Value');
            file_list = get(handles.listbox_Models,'String');
            filename = file_list{index_selected};
    NewStrVal = get(h,'String');
    NewVal = str2double(NewStrVal);
    [path,name,ext,ver] = fileparts(filename);
    if isequal(name,'mainsystem')
        if ~isequal(NewStrVal,'4') & ~isequal(NewStrVal,'16')
            errordlg('QAM value is only valid for 4-ary and 16-ary QAM')
            return
        end
    end
    if isequal(name,'mainsys4')
        if ~isequal(NewStrVal,'4') & ~isequal(NewStrVal,'16')
            errordlg('QAM value is only valid for 4-ary and 16-ary QAM')
```

```
        return
      end
    end
    if isequal(name,'mainsys8')
      if ~isequal(NewStrVal,'4') & ~isequal(NewStrVal,'16')
        errordlg('QAM value is only valid for 4-ary and 16-ary QAM')
        return
      end
    end
    if isequal(name,'mainsys16')
      if ~isequal(NewStrVal,'4') & ~isequal(NewStrVal,'16')
        errordlg('QAM value is only valid for 4-ary and 16-ary QAM')
        return
      end
    end
    if isequal(name,'mainsys64')
      if ~isequal(NewStrVal,'4') & ~isequal(NewStrVal,'16')
        errordlg('QAM value is only valid for 4-ary and 16-ary QAM')
        return
      end
    end
    if isequal(name,'mainsys128')
      if ~isequal(NewStrVal,'4') & ~isequal(NewStrVal,'16')
        errordlg('QAM value is only valid for 4-ary and 16-ary QAM')
        return
      end
    end
    if isequal(name,'mainsysHamming')
      if ~isequal(NewStrVal,'4') & ~isequal(NewStrVal,'16')
        errordlg('QAM value is only valid for 4-ary and 16-ary QAM')
        return
      end
    end
    if isequal(name,'mainsyslt')
      if ~isequal(NewStrVal,'4') & ~isequal(NewStrVal,'16')
        errordlg('QAM value is only valid for 4-ary and 16-ary QAM')
        return
      end
    end
    dest=([name,'/Rectangular QAM Modulator Baseband'])
    Num = get_param(dest,'MaskValues');
    Num(1) = {NewStrVal}
    set_param(dest,'MaskValues',Num)



% ------------------------------------------------------------------
function varargout = SNR_Level_Callback(h, eventdata, handles, varargin)
    index_selected = get(handles.listbox_Models,'Value');
            file_list = get(handles.listbox_Models,'String');
            filename = file_list{index_selected};
    NewStrVal = get(h,'String');
    NewVal = str2double(NewStrVal);
    [path,name,ext,ver] = fileparts(filename);
    dest=([name,'/AWGN Channel'])
    Num = get_param(dest,'MaskValues');
    Num(4) = {NewStrVal}
    set_param(dest,'MaskValues',Num)
```

```
% -------------------------------------------------------------------
function varargout = pushbutton_Results_Callback(h, eventdata, handles, varargin)
   if isfield(handles,'ResultsData') & ~isempty(handles.ResultsData)
              ResultsData = handles.ResultsData;
              % Determine the maximum run number currently used.
              maxNum = ResultsData(length(ResultsData)).RunNumber;
              ResultNum = maxNum+1;
   else, % Set up the results data structure
              ResultsData = struct('RunName',[],'RunNumber',[],...
                     'NumSubcarriers',[],'Coded',[],'Uncoded',[],'SNRLevel',[],'BERLevel',[]);
              ResultNum = 1;
   end

   index_selected = get(handles.listbox_Models,'Value');
          file_list = get(handles.listbox_Models,'String');
          filename = file_list{index_selected};
   NewStrVal = get(h,'String');
   NewVal = str2double(NewStrVal);
   [path,name,ext,ver] = fileparts(filename);
   dest=([name,'/AWGN Channel'])
   Value = get_param(dest,'MaskValues');
   SNR = char(Value(4));
   Simout = evalin('base','simout');
   Temp = Simout.^2;
   a = mean(Temp);
   b = max(Temp);
   c = b/a;
   Val = 10*log10(c);
   Subcarriers = evalin('base','Subcarriers');
   [Num,k,j] = size(Subcarriers);
   if isequal(Num,4)
      PAPR = '6.021'
   end
   if isequal(Num,8)
      PAPR = '9.031'
   end
   if isequal(Num,16)
      PAPR = '12.041'
   end
   if isequal(Num,32)
      PAPR = '15.051'
   end
   if isequal(Num,64)
      PAPR = '18.062'
   end
   if isequal(Num,128)
      PAPR = '21.072'
   end
   if isequal(Num,256)
      PAPR = '24.082'
   end
   BER = evalin('base','BER')
   [k,l,m] = size(BER)
   Error_rate = BER(k)
   ResultsData(ResultNum).RunName = name;
   ResultsData(ResultNum).RunNumber = ResultNum;
   ResultsData(ResultNum).NumSubcarriers = Num;
   ResultsData(ResultNum).Coded = Val;
   ResultsData(ResultNum).Uncoded = PAPR;
```

```
   ResultsData(ResultNum).SNRLevel = SNR;
   ResultsData(ResultNum).BERLevel = Error_rate;
   ResultsStr = get(handles.listbox_Results,'String');
if isequal(ResultNum,1)
        ResultsStr = {[name,'        ',num2str(Num),'            ',num2str(Val),'            ',num2str(PAPR),'
',SNR,'             ',num2str(Error_rate)]};
else
        ResultsStr = [ResultsStr; {[name,'        ',num2str(Num),'            ',num2str(Val),'
',PAPR,'           ',SNR,'             ',num2str(Error_rate)]}];
end
set(handles.listbox_Results,'String',ResultsStr);
% Store the new ResultsData
handles.ResultsData = ResultsData;
guidata(h,handles)
handles.ResultNum = ResultNum;
guidata(h,handles)
flag = 0;
flag1 = 0;
handles.flag = flag;
guidata(h,handles)
handles.flag1 = flag1;
guidata(h,handles)
% --------------------------------------------------------------------
function varargout = pushbutton_Close_Callback(h, eventdata, handles, varargin)
close(handles.figure1);
% --------------------------------------------------------------------
function varargout = pushbutton_PAPR_Callback(h, eventdata, handles, varargin)

maxNum = handles.ResultNum
list_entries = get(handles.listbox_Results,'String');
index_selected = get(handles.listbox_Results,'Value');
for i = 1:length(index_selected);
   x(i) = handles.ResultsData(index_selected(i)).NumSubcarriers;
        y(i) = handles.ResultsData(index_selected(i)).Coded;
end
p = sort(x)
o = sort(y)
   % Plot data
if ~isequal(handles.flag,1)
figure('Name','PAPR Graph')
end
plot(p,o,'-rs','LineWidth',2,...
         'MarkerEdgeColor','k',...
         'MarkerFaceColor','g',...
         'MarkerSize',10)
xlabel('Subcarriers')
ylabel('PAPR')
k = max(x)/2
j = max(y)-2
me = handles.ResultsData(index_selected(1)).RunName
text(k,j,me)
flag = 1;
handles.flag = flag;
guidata(h,handles)
hold on;

% --------------------------------------------------------------------
function varargout = pushbutton_BER_Callback(h, eventdata, handles, varargin)
index_selected = get(handles.listbox_Results,'Value');
for i = 1:length(index_selected);
```

```
   x(i) = str2num(handles.ResultsData(index_selected(i)).SNRLevel);
          y(i) = handles.ResultsData(index_selected(i)).BERLevel;
end
   % Plot data
if ~isequal(handles.flag1,1)
figure('Name','BER Graph')
end
plot(x,y,'-rs','LineWidth',2,...
           'MarkerEdgeColor','k',...
           'MarkerFaceColor','g',...
           'MarkerSize',10)
xlabel('SNR')
ylabel('BER')
k = max(x)-2
j = max(y)-0.05
me = handles.ResultsData(index_selected(1)).RunName
text(k,j,me)
flag = 1;
handles.flag = flag;
guidata(h,handles)
hold on;




% ------------------------------------------------------------------
function load_listbox(dir_path,handles)
cd (dir_path)
dir_struct = dir(dir_path);
[sorted_names,sorted_index] = sortrows({dir_struct.name}');
handles.file_names = sorted_names;
handles.is_dir = [dir_struct.isdir];
handles.sorted_index = [sorted_index];
guidata(handles.figure1,handles)
set(handles.listbox_Models,'String',handles.file_names,...
          'Value',1)
```