

**Deanship of graduate Studies
Al-Quds University**



**Quality of Service Improvement for
WiMAX Physical Layer**

Bashar Farouq Khalil Musa

M.Sc. Thesis

Jerusalem - Palestine

July-2011

**Quality of Service Improvement for
WiMAX Physical Layer**

**Prepared by:
Bashar Farouq Khalil Musa**

B.Sc.: Electrical Engineering from Birzeit University

Supervisor: Dr. Naser Hamad

**Thesis submitted in partial fulfillment of requirement
for the degree of Master of Computer and Electronic
Engineering**

**Faculty of Engineering
Al-Quds University**

**Jerusalem-Palestine
July 2011**

**Al-Quds University
Deanship of Graduate Studies
Electronic and Computer Engineering**

Thesis Approval

Quality of Service Improvement for WiMAX Physical Layer

**Prepared by: Bashar Musa
Registration No: 20510267**

Supervisor: Dr. Naser Hamad

**Master thesis submitted and accepted, date: 2/7/2011
The names and signatures of the examining committee members
as follows:**

**Head of committee: Dr. Naser Hamad
Internal examiner: Dr. Ali Jamoos
External examiner: Dr. Ghandi Manasra**

signature

signature

signature



**Jerusalem-Palestine
July 2011**

Dedication

I dedicate this thesis to my parents. Without their patients, understanding, supports and most of all love, the completion of this work would not have been possible.

Bashar Musa

Declaration

I certify that this thesis, submitted for the degree of Master, is the result of my own research, except where otherwise acknowledged, and that this thesis (or any part of the same) has not been submitted for a higher degree to any other university or institution.

Signed:

Bashar Farouq Khalil Musa

Date: 2/7/2011

Acknowledgments

All the praises is due to Almighty ALLAH whose blessing have always been prevail on us. I'm thankful to my parents, whose unconditional support and encouragement was always there throughout my work.

I'm greatly honoured and thankful to my Supervisor Dr. Nasser Hamad for his guidance and support throughout the thesis period. My sincere thanks all for people who supported and helped me to finish my thesis specially Mrs Ola Sbihat.

Finally, I'm thankful to Al-Quds University which gives me the opportunity to learn and spread the light of education

Abstract

With the rapid growth of digital communication in recent years, the need for high speed data transmission has increased very rapidly. Promising future wireless systems are expected to support a wide range of multimedia services such as video, data and voice. All of these services require different levels of transmission data rates, and Quality of Service (QoS) which imposes a variety of different performance requirements on the newly developed Worldwide Interoperability for Microwave Access (WiMAX) Network. Unlike many conventional communication networks, the WiMAX composed of only two layers, namely, Physical (PHY) and Medium Access Control (MAC) layers.

In WiMAX physical layer, Orthogonal Frequency Division Multiplexing (OFDM) technique is employed to increase the transmission data rate by converting the serial input data to a parallel data output. The immunity of channel imperfectness made OFDM a popular scheme for wideband digital communication. It is used in applications such as digital television and audio broadcasting and the multimedia transmission that is considered as a real time services. Forward Error Coding (FEC) techniques are introduced for system performance enhancement, since Viterbi Decoder of high speed decoding capabilities is a vital challenge in wide band transmission systems like WiMAX; Substituting the conventional Viterbi decoder by the so-called Parallel Viterbi decoder in order to overcome the aforementioned challenge. Promising approach to achieve high throughput have been proposed in this project , look-ahead, is studied for extracting vectorized output bits without taking into consideration the hardware cost involved. It came out that even with small steps of looking ahead the processing gains are very high.

In WiMAX system, the conventional method used to implement the OFDM system is by using the Inverse Fast Fourier Transform (IFFT) in the transmitter side and FFT in the receiver side. Due to the drawbacks of OFDM-FFT based system which are the high peak-to-average ratio (PAR) and the synchronization, an efficient technique for the OFDM system using wavelet transform is proposed. This system shows a superior performance when compared with traditional FFT-OFDM systems through an Additive White Gaussian Noise (AWGN) channel. The system performance is described in Bit Error Rate (BER) as a function of Signal to Noise Ratio (SNR). Furthermore, the proposed system gives nearly a perfect reconstruction for the input signal in the presence of Gaussian noise.

WiMAX

(OFDM) (WiMAX)

(IFFT/FFT)

(IDWT/DWT)

:

WiMAX

▪

. IDWT/DWT IFFT/FFT

(Viterbi Algorithm)

▪

WiMAX

Algorithm)

,WiMAX

(Viterbi

(Parallel Lock ahead)

IFFT/FFT

IDWT/DWT

Definitions and Abbreviations

| | |
|------|--------------------------------------|
| 3G | Third Generations |
| 4G | Fourth Generations |
| ACS | Add-Compare-Select |
| ADSL | Asynchronous Digital Subscriber Line |
| AWGN | Additive White Gaussian Noise |
| BER | Bit Error Rate |
| BWA | Broadband Wireless Access |
| CWT | Continuous Wavelet Transform |
| CFO | Carrier Frequency Offset |
| CMF | Conjugate Mirror Filters |
| CIR | Channel Impulse Response |
| CP | Cyclic Prefix |
| DAB | Digital Audio Broadcasting |
| DHT | Discrete Hartley Transform |
| DFT | Discreet Fourier Transform |
| DWT | Discrete Wavelet Transform |
| DVB | Digital Video Broadcasting |
| FDMA | Frequency Division Multiple Access |
| FEC | Forward Error Correction |
| FFT | Fast Fourier Transform |
| GPRS | General Packet Radio Services |
| ICI | Inter Carrier Interference |
| IDFT | Inverse Discreet Fourier Transform |
| IDWT | Inverse Discrete Wavelet Transform |

| | |
|-------|---|
| IFFT | Inverse Fast Fourier Transform |
| IMD | Inter- Modulation Distortion |
| ISI | Inter Symbol Interference |
| LDPC | Low-Density Parity-Check |
| LTE | Long Term Evolution |
| MAC | Medium Access Control |
| Mbps | Megabits per seconds. |
| MCM | Multi-Carrier Modulation |
| MIMO | Multi Input Multi Output |
| OFDMA | Orthogonal Frequency Division Multiple Access |
| PV | Parallel Viterbi |
| QAM | Quadrature Amplitude Modulation |
| QoS | Quality of Service |
| QPSK | Quadrature Phase Shift Keying |
| TACS | Total Access Communication System |
| TDD | Time Division Duplex |
| TDMA | Time Division Multiple Access |
| UMTS | Universal Mobile Telecommunication System |
| VoIP | Voice over IP |
| WCDMA | Wide-band Code Division Multiple Access |
| WiMAX | World Interoperability for Microwave Access |
| WLANs | Wireless Local Area Networks |
| RS | Reed-Solomon |
| SNR | Signal to Noise Ratio |

Table of Contents

| | |
|--|------|
| Dedication | i |
| Declaration | ii |
| Acknowledgments..... | iii |
| Abstract..... | iv |
| | vi |
| Definitions and Abbreviations | viii |
| Chapter 1 Introduction | 1 |
| 1.1 Mobility Enhancement and Evolution..... | 2 |
| 1.1.1 Evolution in Cellular Technologies..... | 2 |
| 1.1.2 Evolution of WiMAX Technology | 4 |
| 1.2 Radio Specifications of WiMAX Technology | 6 |
| 1.3 Comparison between WiMAX and Long Term Evolution (LTE)... | 7 |
| 1.4 Basic Concepts of OFDM Technology | 8 |
| 1.5 Error Control Coding Techniques | 9 |
| 1.6 Synopsis of Thesis | 10 |
| Chapter 2 WiMAX Physical Layer | 12 |
| 2.1 OFDM Literature Background, and Historical Review | 12 |
| 2.2 OFDM Technology Basic Concepts | 13 |
| 2.3 OFDM System Model | 16 |
| 2.3.1 OFDM Transmitter..... | 16 |
| 2.3.2 OFDM Receiver | 19 |
| 2.3.3 Additive White Gaussian Noise (AWGN) Channel | 23 |
| 2.4 Features of OFDM System..... | 23 |

| | | |
|-----------|--|----|
| Chapter 3 | Channel Coding and Viterbi Algorithm..... | 26 |
| 3.1 | FEC Coding Techniques..... | 26 |
| 3.2 | Convolutional Codes..... | 29 |
| 3.3 | Viterbi Algorithm..... | 31 |
| 3.4 | Performance Issues..... | 34 |
| Chapter 4 | FFT and DWT Based OFDM..... | 36 |
| 4.1 | Comparison between Fourier Transform and Wavelet Transform | 36 |
| 4.2 | Wavelet Transform Technology..... | 38 |
| 4.2.1 | Continuous Wavelet Transform (CWT)..... | 39 |
| 4.2.2 | The Discrete Wavelet Transform (DWT)..... | 40 |
| 4.3 | DWT and Filter Banks..... | 40 |
| 4.3.1 | Multi-Resolution Analysis using Filter Banks..... | 40 |
| 4.3.2 | Conditions for Perfect Reconstruction..... | 42 |
| 4.4 | Classification of wavelets..... | 43 |
| 4.4.1 | Orthogonal wavelet filter banks..... | 43 |
| 4.4.2 | Biorthogonal wavelet filter banks..... | 44 |
| 4.5 | Wavelet Families..... | 44 |
| Chapter 5 | Simulation Model, and Results..... | 48 |
| 5.1 | Simulated OFDM System Model..... | 48 |
| 5.1.1 | FFT-based OFDM..... | 48 |
| 5.1.2 | DWT-based OFDM..... | 50 |
| 5.1.3 | Constellation Mapper..... | 52 |
| 5.1.4 | Convolutional Encoder..... | 52 |
| 5.1.5 | Viterbi Algorithm..... | 53 |

| | | |
|-----------------------|------------------------------------|----|
| 5.1.6 | Channel Model | 55 |
| 5.2 | Simulation Results | 55 |
| 5.2.1 | Viterbi Decoder Results | 56 |
| 5.2.2 | IDWT Instead of IFFT..... | 64 |
| Chapter 6 | Conclusions and Future Works | 70 |
| REFERENCES | | 72 |
| APPENDIX MATLAB CODES | | 76 |

List of Figures

| <i>Number</i> | <i>Page</i> |
|--|-------------|
| Figure 1-1: Major System Developed for Second Generation. | 3 |
| Figure 1-2: Evolution of Cellular Mobile Systems from 2G to 4G. | 4 |
| Figure 2-1: Conventional vs. OFDM Transmission. | 13 |
| Figure 2-2: OFDM Subcarriers in Frequency Domain. | 15 |
| Figure 2-3: OFDM Signal (a) Conventional Multi-carrier (b) Orthogonal Multi-carrier. | 16 |
| Figure 2-4: OFDM Transmitter Block Diagram. | 17 |
| Figure 2-5: OFDM Receiver Block Diagram. | 19 |
| Figure 2-6 : Guard time and Cyclic Extension-Effect of Multipath. | 21 |
| Figure 2-7: Frequency Synchronization Error. | 22 |
| Figure 2-8: Time Synchronization Error. | 23 |
| Figure 3-1: Reed-Solomon Codeword. | 28 |
| Figure 3-2: State Diagram and Convolutional Encoder. | 30 |
| Figure 3-3: Example of Trellis Diagram. | 30 |
| Figure 3-4: Viterbi Decoder Block Diagram. | 32 |
| Figure 3-5: Survivor Path Graph Example. | 34 |
| Figure 4-1: Time-Frequency (a) Time-Domain (b) Frequency-Domain (c) FFT (d) DWT | 38 |
| Figure 4-2: Three-Level Wavelet Decomposition Tree. | 41 |
| Figure 4-3: Three-Level Wavelet Reconstruction Tree. | 42 |
| Figure 4-4: Wavelet Families (a) Haar (b) Daubechies4 (c) Coiflet1 (d) Symlet2 (e) Meyer (f) Morlet (g) Mexican Hat. | 45 |
| Figure 5-1: FFT-WiMAX Block Diagram. | 49 |

| | |
|---|----|
| Figure 5-2: DWT-WiMAX Block Diagram..... | 51 |
| Figure 5-3: 1/2 Coding Rate Convolutional Encoder | 53 |
| Figure 5-4: Time Needed to Decode Bits Using Serial and Parallel Viterbi Decoder (M=2).57 | |
| Figure 5-5: Throughput Enhancement for Parallel Viterbi Decoder with M=2..... | 58 |
| Figure 5-6: BER Performance for Serial and Parallel Viterbi with M=2. | 59 |
| Figure 5-7: The Decoding Time for Serial, Parallel with M=2, 3, 4..... | 60 |
| Figure 5-8: Number of Pathes for M=2 and M=3..... | 60 |
| Figure 5-9: Number of branches Need to be calculated. | 61 |
| Figure 5-10: The Number of Adders, Compare Select Units and Latency with Number of Viterbi Steps M. | 62 |
| Figure 5-11: Throughput Enhancement for Parallel Viterbi Decoder. | 62 |
| Figure 5-12: BER Performance for Serial, Parallel Viterbi (M=2, 3, 4) with QPSK Modulation Scheme..... | 63 |
| Figure 5-13: BER Performance for Serial and Parallel Viterbi with Different Modulation Techniques..... | 64 |
| Figure 5-14: BER for OFDM Based FFT vs. OFDM Based DWT (Haar)..... | 65 |
| Figure 5-15: FFT with CP vs DWT (Haar). | 65 |
| Figure 5-16: FFT and DWT (Haar) size Effect on BER. | 66 |
| Figure 5-17: DWT Families BER Performance. | 66 |
| Figure 5-18: BER for OFDM Based FFT vs. OFDM Based DWT Biorthogonal Family ... | 67 |
| Figure 5-19: BER Performance Comparison between Coded and Uncoded FFT and DWT (Haar). | 68 |

List of Tables

| | |
|--|----|
| Table 1-1: WiMAX vs. Long Term Evolution | 7 |
| Table 5-1: Simulation Parameters..... | 56 |

CHAPTER ONE

Chapter 1 Introduction

With the rapid growth of digital communication in recent years the need for high speed data transmission has increased very rapidly. Promising future wireless systems are expected to support a wide range of multimedia services such as video, data and voice. All of these services require different levels of transmission data rates, and QoS which imposes a variety of different performance requirements on the newly developed WiMAX Network.

The WiMAX technology is considered as an attractive one for many researchers overall the world, as such technology promises to provide high data rates in a mobile environment which cannot be achieved, at least for the time being, by the cellular mobile systems. In WiMAX physical layer, however, various parameters should be addressed and considered when studying the QoS of the WiMAX network, some of these parameters are, number of Sub-carriers, pilots, guard bands, FEC, modulation schemes, etc.¹. Studying and analysing the effect of these parameters on system performance help designers to achieve the required QoS. Delivering high quality of services over the physical layer is considered as the major challenge of the WiMAX research area. Several factors and components that affect the performance of QoS will be studied in this research, e. g., a proper combination of the transmission rate and modulation schemes in the physical layer yields a noticeable improvement of the Bit Error Rate (BER) performance in an Additive White Gaussian Noise Channel (AWGN).

¹ Details of these parameters definition to be introduced in the following chapters.

Based on the above introduction, our main objective in this research is to study and propose several QoS parameters that decreases the BER and hence and improving the Signal to Noise Ratio in a WiMAX network. The overall objectives are:

1. To gain in-depth knowledge about the WiMAX system and how it works.
2. To understand the problems related to WiMAX QoS and to determine the controlling parameters.
3. To increase WiMAX throughput² by employing a high speed Viterbi decoder, that decodes a couple of bits simultaneously instead of one.
4. Improving WiMAX BER performance by utilizing a new method of modulation in OFDM.

Brief details of how the advancement of increasing transmission data rate and the mobility issues are achieved by WiMAX technology is presented in the following subsections. Toward this end, we introduced a quick revision of the evolution of several mobile systems.

1.1 Mobility Enhancement and Evolution

The controlling of mobility issues is a major challenge in mobile system design. Cellular mobile systems and WiMAX system are now the main technologies that provide large scale mobility. In the following subsections, both systems will be introduced.

1.1.1 Evolution in Cellular Technologies

The First Generation (1G) wireless technology evolved in mid-1980's. It is based on analog cellular systems and support only voice transmission. Frequency Division Multiple Access (FDMA) was employed as an air interface protocol to control the frequency channels distribution. Different systems were developed for the first generation technology in different

² The throughput is defined as the average successfully received data bits in unit time.

regions overall the world, e.g., Advanced Mobile Phone System (AMPS), and Total Access Communication System (TACS), were developed in USA, and United Kingdom, respectively [1].

Second Generation (2G) began to evolve in 1990's [2], in which digital transmission of data in wireless cellular networks were introduced for the first time. Examples of the major systems developed for the second generation are shown in Figure 1-1.

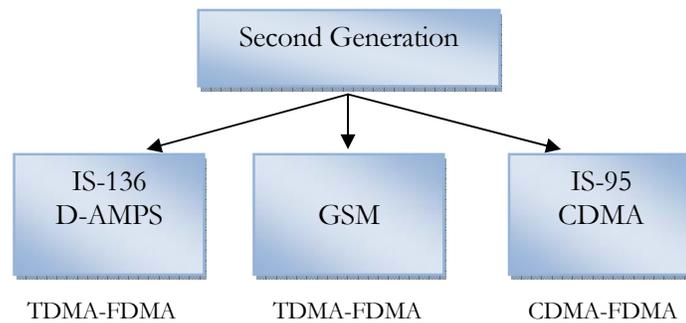


Figure 1-1: Major System Developed for Second Generation.

Third Generation (3G) families are introduced mainly through Universal Mobile Telecommunication System (UMTS), CDMA-2000. These both technologies have high data rates and are much more bandwidth efficient than the second generation systems [1]. Now it is possible to achieve high data rates in Broadband Wireless Access (BWA). By integrating both UMTS and WiMAX technologies, it becomes possible to provide several types of services such as video conferencing and wireless broadband services.

Fourth Generation (4G) cellular mobile system is currently under development stage, it is also known as IMT-Advanced [1]. Such a technology is expected to be a bandwidth efficient and also to provide data rates up to 100 Mbps in high mobility users' environment and up to

one Gbps for fixed or portable data terminals. Main technologies, commercial examples of 4G technologies are the Long Term Evolution (LTE) and Ultra Mobile Broadband (UMB). Figure 1-2 shows the evolution towards 4G in terms of data rates, starting from 2G. It also provides information about the standard and technologies according to their respective generation [3].

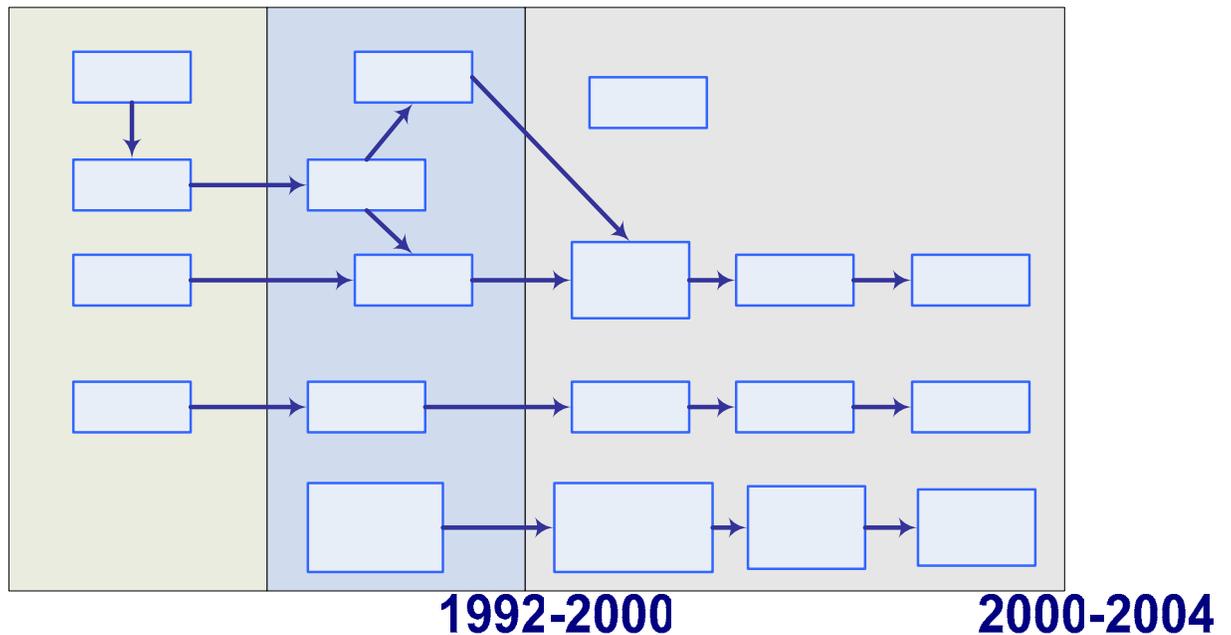


Figure 1-2: Evolution of Cellular Mobile Systems from 2G to 4G.

TDMA

**EGPRS
384 Kbps**

1.1.2 Evolution of WiMAX Technology

WiMAX is a standard-based technology enabling the delivery of last mile wireless broadband access and it is considered as an alternative wireless technology to the existing wire line Digital Subscriber Line (DSL) technology with mobility capabilities. Moreover, it

GSM

**GPRS
150 Kbps**

is standardized as IEEE 802.16 and IEEE Wireless-MAN. Various standards of 802.16 were developed aiming at improving the standard. For enhancement, the first 802.16 standard was

CDMA

**WCDMA
2 Mbps**

CDMA 2000

released in 2000. It provides a standard point-to-multipoint broadcast in 10 to 66 GHz frequency range for Line of Sight (LOS) environment. The second version of WiMAX standard 802.16a was an amendment of 802.16 standards and has the capability to broadcast point-to-multipoint in the frequency range 2 to 11 GHz. It was established in 2003 and assigned both licensed and unlicensed frequency bands. Unlicensed bands cover maximum distance from 31 to 50 miles. It improves QoS features with supporting protocols, for instance, Ethernet, ATM or IP [4].

The third version of WiMAX standard is 802.16c which is an enhancement of 802.16 standards which mostly dealt with frequency range from 10 to 66 GHz. This standard addressed various issues, such as, performance evaluation, testing and detailed system profiling. The system profile is developed to specify the mandatory features to ensure interoperability and the optional features that differentiate products by pricing and functionality.

In 2003, a project known as 802.16d began [5] which were deduced in 2004 with the Release of 802.16d. Such a standard supports mandatory and optional elements along with Time Division Duplex (TDD) and Frequency Division Duplex (FDD) technologies [3]. Theoretically, the data rate reaches 70 Mbps, although practically, the performance is near about 40 Mbps. However, the real evolution of WiMAX technology started from the standard 802.16e which was an amendment of 802.16d standard and completed in 2005, sometimes it is referred to as 802.16e-2005. Its main aim is to provide mobility including large range of coverage. That's why people sometimes called it mobile WiMAX. This standard is a technical updates of fixed WiMAX which has robust support of mobile broadband.

The following section covers the basic radio specification of the WiMAX network to in depth understanding of the radio frequencies that are applied in such a technology. Moreover, a comparison between WiMAX technique and an example of the 4G cellular mobile system is also introduced.

1.2 Radio Specifications of WiMAX Technology

In this section, we introduce most important radio specifications of the WiMAX technology. WiMAX supports multiple physical specifications due to its modular nature. The first version of the standard only supported single carrier modulation. Since that time, OFDM and OFDMA have been included to operate in (Non-Line of Sight) NLOS environments and to provide mobility [4]. The standard has also been extended for use in below 11 GHz frequency bands along with initially supported 10-66 GHz bands. The IEEE 802.16 licensed and unlicensed bands are as follows:

1. *From 10 to 66 GHz licensed band:* In this frequency band, LOS is required due to shorter wave length wherein the effect of multipath propagation is neglected. The standard is established to provide data rates up to 120 Mb/s within the abundant availability of bandwidth that is also another reason to operate in this frequency range. Channels within these bands are typically 25 or 28 MHz range [6].
2. *From 2 to 11 GHz licensed and licensed bands:* In this frequency band, both licensed and licensed exempt bands are addressed. Additional physical functionality supports have been introduced to operate in Near LOS and NLOS environment and to mitigate the effect of multipath propagation [6]. Expected data rates are up to 70Mb/s in available 14 MHz channel bandwidth.

1.3 Comparison between WiMAX and Long Term Evolution (LTE)

From the extensive studies of the LTE and other broadband systems such as WiMAX, we conclude that both technologies are similar to each other in several different aspects, such as throughput, latency, and mobility, such similarity can be seen in Table 1-1 below. However, the obvious difference is that LTE is standardized by 3GPP group to extend the services supported by the 3GPP legacy systems, such as UMTS, GSM, EDGE and GPRS whereas the WiMAX is standardized by the IEEE Association [7-10].

Table 1-1: WiMAX vs. Long Term Evolution

| Aspect | LTE | WiMAX |
|---|---|--|
| Standardization Body | 3GPP | WiMAX Forum |
| Access Technique | OFDMA/SC-FDMA/MIMO | OFDMA/MIMO |
| Throughput Downlink (DL) Uplink (UL) | 100 Mb/s at 20 MHz 50 Mb/s at 20 MHz | 802.16m 128 Mb/s at 20 MHz 56 Mb/s at 20 MHz |
| Band Width | 1.4, 3, 5, 10, 15, 20 MHz | 5, 8.75, 10 MHz |
| Supporters | NSN, Ericsson, Alcatel | Intel |
| Coding Techniques | QPSK, 16QAM, 64QAM | BPSK, QPSK, 16QAM, 64QAM |
| Coverage Range | 5 Km, after 30 Km slightly degraded. | LOS 30 to 50 km NLOS 4 to 9 km |
| Mobility: Speed Handover | Up to 250 Km/h Inter cell soft handover supported | Up to 120 Km/h Optimized hard handover supported |
| Cell Capacity | >200 users per cell at 5 MHz BW >400 users per cell for larger BW | 100-200 users per cell. |
| Legacy | GSM/GPRS/EDGE/UMTS/HSPA | IEEE802.16a through 16e |
| Cost to the end user | Expected to be higher in cost especially for the supporting devices since there is patent and property percentage per device. | Supposed to be cheaper since there are no intellectual propriety rights. |

To complete the picture of our proposed system, an OFDM as a multiplexing technique used in WiMAX is presented. Moreover, our proposed system includes convolutional coding/Viterbi decoding techniques to improve the system performance; hence, forward error correction code is given in Sec. 1.5.

1.4 Basic Concepts of OFDM Technology

The Multi-Carrier Modulation (MCM) transmission technique was the initiator of the idea of OFDM, where the principle concept of MCM is the division of input bit stream into several parallel bit streams used to modulate several subcarriers. Each subcarrier is separated by a guard band to ensure non overlapping bands and hence the problem of Inter-Carrier Interference does not exist. In the receiver side, band pass filters are utilized to gather the spectrum of each subcarrier individually.

A spectral efficient OFDM is a special case of MCM technique, which employs densely spaced orthogonal subcarriers and allowing overlapping spectrums. Unlike MCM, the use of band pass filters is not required in OFDM because of the orthogonal nature of the employed subcarriers. Hence, the available bandwidth is utilized very efficiently without causing Inter-Carrier Interference (ICI) problem. FFT is used to achieve orthogonality between subcarriers. However, very high speed data transmission causes the problem of Inter-Symbol Interference (ISI); OFDM provides a high data rate without ISI by transmitting Low data rate on each subcarrier, that is, the high data rate transmission is obtained by aggregating of multiple low data rate subcarriers. In addition, Cyclic Prefix (CP) [11] also used to reduce the effect of ISI in an OFDM system.

1.5 Error Control Coding Techniques

Since we employ the Viterbi decoder in the receiver side of our WiMAX system, the last part of our introduction is to briefly revise basic ideas of forward error correction codes, techniques employed in digital communication systems to enhance and improve the system performance.

FEC coding technique is a technique usually used in digital communication systems to enhance the system performance by correcting the received bits in error, hence, decreasing the required signal-to-noise ratio to achieve certain quality. The FEC is carried out using several codes, among these codes, the remarkable convolutional code that is sometimes concatenated with other code. In OFDM system, the convolutional code and the Reed-Solomon code are used together in concatenated fashion to correct the bits received in error. The outer code is Reed-Solomon (RS) and inner code is Convolutional code (CC). The RS outer code corrects burst error at the byte level. It is particularly useful for OFDM links in the presence of multipath propagation whereas the CC corrects the errors independently [11].

Convolutional coding with Viterbi decoding is a popular and powerful method for FEC in communication systems. The decoding process of a convolutional encoded message is based on the maximum likelihood estimation principle. This could be achieved by selecting the most likely path (survival path) in the trellis structure that describes all possible states [12]. However, the implementation of the Viterbi decoder required management of the relevant memory contents which is a major design problem for both hardware and software realizations [13].

1.6 Synopsis of Thesis

In Chapter Two, we introduce an overview, background, and literature review of the IEEE 802.16 physical layer specifications and study the OFDM system model, transmitter, receiver and channel, OFDM as a multiplexing technique is also introduced. Since our results show a remarkable enhancement in system performance when using parallel Viterbi decoder, FEC and Viterbi Decoder implementation is given Chapter Three. Chapter Four deals with DWT Algorithm. The overall system model is presented in Chapter Five, where we conclude our results and analysing the numerical figures in this chapter. Finally, future works and other problems are opened in Chapter Six.

CHAPTER TWO

Chapter 2 WiMAX Physical Layer

2.1 OFDM Literature Background, and Historical Review

The concept of OFDM was introduced in the mid of 60s as a parallel transmission scheme. At first, the aim was to use available bandwidth more efficiently by removing guard bands and setting spacing carriers as close as possible. This aim can be achieved by making all carriers orthogonal to each other, which prevent the so-called Inter Carrier Interference (ICI) between them.

Under the limitations of advancement of digital signal processing technology, the OFDM system was too expensive and complex to be realized since using a bank of parallel modulators to produce multiple carriers. Moreover, the receiver needs tight synchronization and precise phasing of the demodulation carriers and sampling times in order to keep interference between subchannels acceptable where this requirement cannot be realized by the analog modulator. The complexity and other limitations can be solved with the available DSP hardware and software. Weinstein and Ebert proposed in their seminal paper [15] the usage of the Discrete Fourier Transform (DFT) to parallelize data transmission system to produce multiple stable carriers.

In the 1980s, OFDM had been studied for high speed modems, digital mobile communications, and high density recording. In the 1990s, OFDM has been exploited for wideband data communications over mobile radio FM channels, high bit rate digital subscribers lines (HDSL), asymmetric digital subscriber lines (ADSL), very high speed digital subscriber lines (VHDSL), digital audio broadcasting (DAB), digital television and HDTV terrestrial broadcasting. During the 1990s, several research documents are published wherein an OFDM for mobile communication based on multicarrier FDMA is proposed, in

which each user has a set of randomly selected subchannels. Recently, OFDM has been successfully implemented in audio broadcasting such as Digital Video Broadcasting (DVB) and Digital Audio Broadcasting (DAB). Also, OFDM has been implemented in wireless LAN application as well [16].

2.2 OFDM Technology Basic Concepts

OFDM is a suitable technique for wireless communications because of its ability to provide large data rates with sufficient robustness to radio channel impairments. Many research Institutes in the world have expertise teams working on the optimization of OFDM for countless applications improvements.

Because of its ISI resistivity, OFDM is considered as a promising modulation scheme for high speed communications. In high data rate communication systems; the time for each transmitted symbol have to be shorter. Moreover, multipath cause time delay and hence, spreading in frequency domain. ISI is a limitation in a high-data rate communication [1], and this limitation is solved in OFDM by sending data of low data rate in parallel simultaneously.

Figure 2-1 shows two ways to transmit the same four binary data.

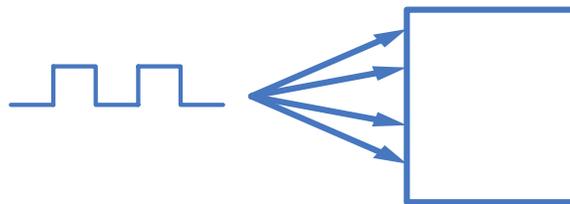


Figure 2-1: Conventional vs. OFDM Transmission.

To clarify the difference between conventional serial and OFDM parallel transmission, suppose the bit duration is T_b , then, the time for transmitting four bits is $4T_b$ as shown in Figure 2-1. If a serial transmission is employed, each bit has duration of one T_b , however, in OFDM these four bits will be transmitted simultaneously, and each bit has duration of $4T_b$. This longer duration leads to reduce the ISI effect for the same transmission data rate.

In an OFDM scheme, a large number of orthogonal, overlapping, narrow band sub-channels or subcarriers, transmitted in parallel, divide the available transmission bandwidth. The separation of the subcarriers is theoretically minimal such that there is a very compact spectral utilization. The "flatness" perceived by a narrow-band channel overcomes the selective fading problem³. Using powerful error correcting codes together with time and frequency interleaving yields even more robustness against frequency selective fading and the insertion of an extra guard interval between consecutive OFDM symbols can reduce the effects of ISI. As a result, the conventional solution of equalization in the receiver side is not necessary [14].

Although, the subcarriers in OFDM are overlapped, but they are orthogonal because the peak amplitude value of one subcarrier occurs when other subcarriers amplitudes are zeros, such concepts of overlapping and orthogonality are shown in Figure 2-2. MATLAB numerical illustrations of 7 overlapped subcarriers are shown in the figure. It is worthy to assist that realizing orthogonality of all subcarriers is achieved mathematically using Inverse Fast Fourier Transform (IFFT).

³ When the bandwidth of the channel is smaller than the signal bandwidth, a selective fading is occurred on a fraction bandwidth of the signal when the channel is subjected to fading.

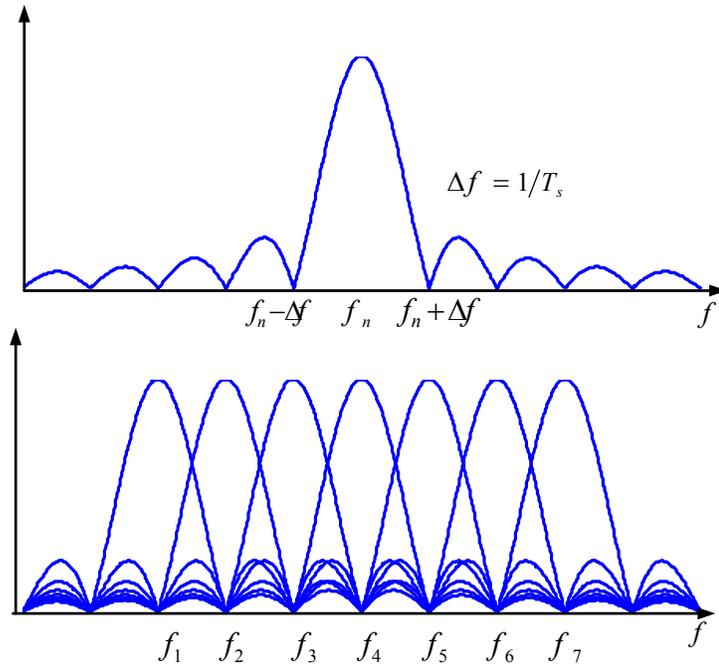


Figure 2-2: OFDM Subcarriers in Frequency Domain

Due to OFDM's immunity to many channel imperfections, it is considered to be the ideal modulation scheme for many applications in which signals are transmitted through practical environments [14].

The following discussion includes mathematical proof of the OFDM concepts. It is meant to provide an intuitive understanding of the OFDM system working.

In a classical parallel-data transmission system, e.g., FDM, the total signal frequency band is divided into N non-overlapping frequency subchannels. Each subchannel is modulated with separate symbol, and then the N subchannels are frequency multiplexed. Figure 2-3 illustrates the difference between the conventional non overlapping multicarrier technique (FDM) and the overlapping multicarrier modulation technique (OFDM). As shown in the figure, high percentage of the bandwidth is kept. The OFDM system comprises of modulator

at the transmitter side, and demodulator at the receiver side. In the following subsections, we introduce an OFDM system implementation.

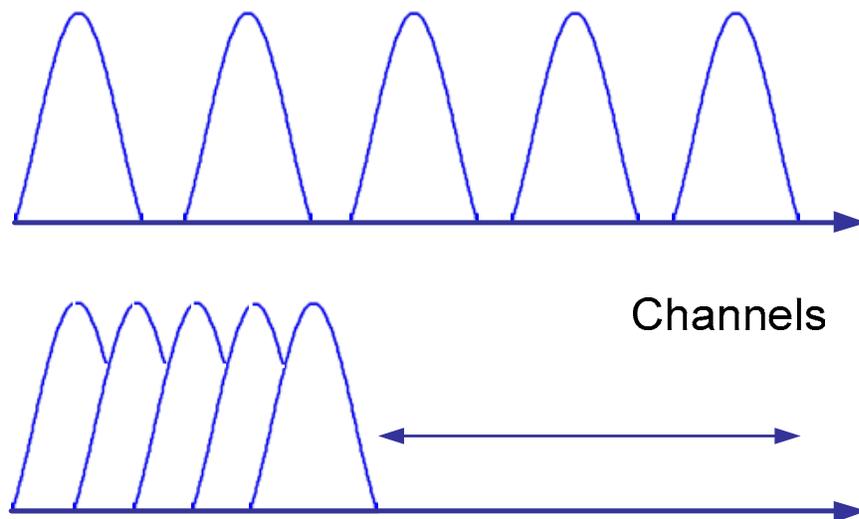


Figure 2-3: OFDM Signal (a) Conventional Multi-carrier (b) Orthogonal Multi-carrier.

2.3 OFDM System Model

In this section, the transmitter, receiver and channel environment of an OFDM system is presented.

Channels

2.3.1 OFDM Transmitter

The DFT basic idea is to transform a cyclic time domain signal into its equivalent frequency spectrum. This is done by generating a sum of orthogonal sinusoidal components. The amplitude and phase of the sinusoidal components represent the frequency spectrum of the time domain signal. Figure 2.4 shows the basic OFDM transmitter, the input bits are line coded mapped in the mapping block, before entering the IFFT, the mapped bits are converted into parallel bit streams. The IFFT block performs the reverse process of the FFT, i.e., transforming a spectrum (amplitude and phase of each component) into a corresponding time domain signal. After insertion of cyclic prefix symbols used to eliminate the ISI, the

data is reversely converted into serial output waveforms, finally, the output waveforms, after band pass filtering, modulates a transmission radio carrier.

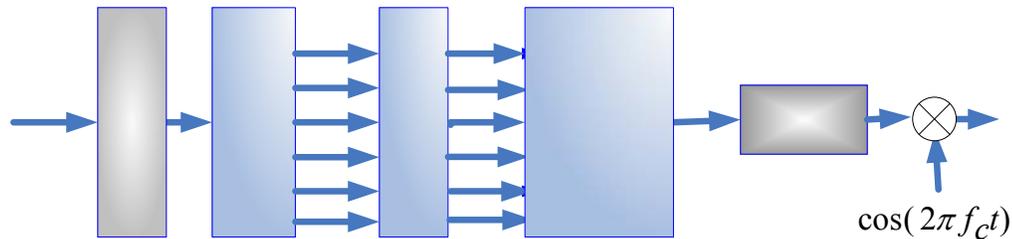


Figure 2-4: OFDM Transmitter Block Diagram

Specifically, consider the case of binary digits transmitted at a rate of R bps. The bandwidth B required to transmit these bits is $R(1+r)$, with r the Nyquist roll off factor, or in the ideal case (rectangular pulse) the order of R Hz. Now consider a sequence of N bits stored for an interval $T_S = N/R$, which is called the OFDM symbol interval. Serial-to parallel conversion is then carried out, with each of the N bits stored used to separately modulate a subcarrier. All N modulated-carrier signals are then transmitted simultaneously over the T_S -long interval. To make the various carrier frequencies orthogonal to each other, it suffices to have spacing Δf between carriers equal to $1/T_S$. Accordingly, we have $B = k\Delta f$, Note that Δf can be viewed as an effective bandwidth of each of the N parallel frequency channels as well.

The implementation of OFDM using multiple carriers transmitted in parallel can, however, produce problem, which is clarified in the following analysis, and such a problem can be avoided by the use of the FFT technique.

Assume N parallel output signals transmitted over a T_S -long symbol interval, and the total signal which is the sum of these signals is $v(t)$. Define the k^{th} carrier frequency f_k as

$$f_k = f_c + k\Delta f, \quad 0 \leq k \leq N-1. \quad (2.1)$$

Data Mapping Serial To Parallel IDFT

Simply, define the carrier nomenclature as the lowest of the N parallel subcarriers, f_c , and the rest of all spaced intervals of Δf . If we write the k^{th} carrier term as $\cos(2\pi(f_c + k\Delta f)t)$, then, $v(t)$ may be written as

$$\begin{aligned}
 v(t) &= \text{Re} \left[\sum_{k=0}^{N-1} a_k e^{j2\pi(f_c + k\Delta f)t} \right] \\
 &= \text{Re} \left[e^{j2\pi f_c t} \sum_{k=0}^{N-1} a_k e^{j2\pi k\Delta f t} \right] \\
 &= \text{Re} \left[e^{j2\pi f_c t} a(t) \right]
 \end{aligned} \tag{2.2}$$

Where

$$a(t) = \sum_{k=0}^{N-1} a_k e^{j2\pi k\Delta f t} \tag{2.3}$$

If the signal $a(t)$ given in (2.3) is sampled at intervals $\frac{T_s}{N}$ apart, i.e., at rate R samples per second, then, it is convenient to replace $a(t)$ by the sampled function $a[n]$, with t replaced by $\frac{nT_s}{N}$, for $n = 0 \dots N-1$. Recall that $\Delta f T_s = 1$. Therefore, $a[n]$ can be written as

$$a[n] = \sum_{k=0}^{N-1} a_k e^{j2\pi kn / N}, \quad n = 0 \dots N-1. \tag{2.4}$$

It is worthy now to note that the expression in (2.4) is exactly in the form of the IDFT. In place of the parallel transmission of N orthogonal carriers which are expressed mathematically by (2.2). Accordingly, the process of OFDM data transmission has been replaced by or can be viewed as IDFT calculations.

2.3.2 OFDM Receiver

Figure 2-5 shows the OFDM receiver where reverse processes to that of the transmitter are carried out on each symbol interval T_S after demodulation of the received modulated carrier signal. The Discrete Fourier Transform is mathematically processed, from which the N coefficients, $a_k, k = 0 \dots N-1$ are recovered and parallel-to-serial conversion is employed to generate the desired output bit stream. It is to be noted that N is selected as multiples of 2 in order to simplify carrying out the DFT calculations, the proof can be found in reference [16].

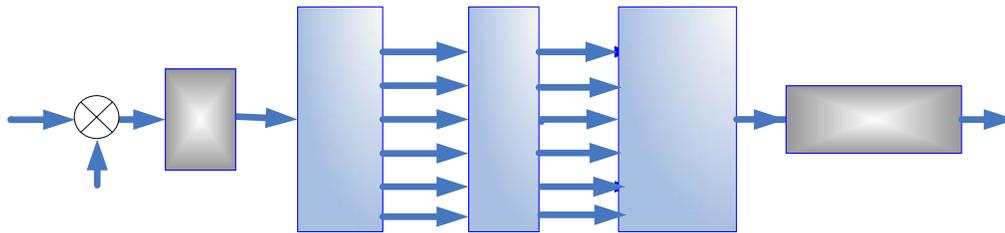


Figure 2-5: OFDM Receiver Block Diagram.

This process of generating the OFDM signal from binary input samples using the IDFT may be generalized to an operation on BPSK signal samples, allowing higher bit rate data signals to be transmitted over a specified bandwidth.

The form of subcarrier pulse is a rectangular shape. It's selected as a rectangular shape because task of pulse forming and modulation can occur by a simple IDFT which can be employed very effectively as an IFFT. On the other hand, at the receiver side, FFT is employed to get the opposite of the IFFT operation.

Fourier Transform theorem states that the well-known function $\text{sinc}(x)$ type of spectrum of the subcarrier from the rectangular pulse shape. Spectra will still be detached from one to another even through subcarriers are overlapped because of their orthogonality.

Antenna

LPF

Serial To Parallel

FFT

occurs at the transmitter for modulation of the transmitted symbols. IFFT selects the gap of the subcarriers in a specific way, that is, at frequency where we calculate the received signal all of the other signals are zero. To satisfy the orthogonality between all subcarriers, there are some rules should be carefully addressed [16]:

1. The receiver and transmitter have to be entirely synchronized. In order to satisfy this requirement, it is necessary to guess the same modulation frequency and the same time scale for transmission which is not really applicable.
2. It is also necessary to have the best quality of the analog transmitter and receiver part. There should not be any multipath channel among the subcarriers to hold the orthogonality. Multipath phenomenon is occurred due to reflection of the transmitted signal by different obstacles like walls, mountains, buildings, etc., because the reflected signals reach the receiver in various paths and hence different times. Energy leakage can occur due to this signal scattering. In order to reduce or optimistically to prevent the multipath severe effects, a technique called Cyclic Prefix is employed, as we will see in the following discussion.

Cyclic prefix insertion is commonly used in OFDM systems as a way to mitigate the effects of ISI. It copies the end section of an IFFT packet to the beginning of an OFDM symbol. Usually the length of the cyclic prefix is longer than the length of the dispersive channel⁴ to completely remove ISI. OFDM modulation, therefore, mostly revolves around cyclic prefix, OFDM modulation includes IFFT operation and cyclic prefix insertion, and OFDM demodulation includes cyclic prefix removal and FFT operation.

Passing the signal through a time-dispersive channel causes ISI. In an OFDM system, a loss of the orthogonality appears due to ISI, resulting in ICI. For a given system bandwidth the symbol rate for an OFDM signal is much lower than a single carrier transmission scheme. It is because the OFDM system bandwidth is broken up into N_c subcarriers resulting in a

⁴ Dispersive channel is due to multipath fading effects.

symbol rate that is N_c times lower. This low symbol rate makes OFDM naturally resistant to effects of ISI caused by multipath propagation. The multiple signals that appear due to the multipath propagation arrive at the receiver side at different times, spreading, this way, the symbol boundaries, hence, causing energy leakage between the OFDM symbols. Furthermore, in an OFDM signal the amplitude and phase of the subcarrier must remain constant over a period of the symbol duration in order to maintain the orthogonality of the subcarriers. If they are not constant, the spectral shape will not have nulls at the correct frequencies, resulting in ICI.

To completely eliminate even the very small ISI that results, Figure 2-6 shows a guard time introduced for each OFDM symbol. The guard time must be chosen to be larger than the expected delay spread, such that multi-path components from one symbol cannot interfere with the next symbol. If the guard time is left empty, this may lead ICI, since the carriers are no longer orthogonal to each other. To avoid such a cross talk between subcarriers, the OFDM symbol is cyclically extended in the guard time. This ensures that the delayed replicas of the OFDM symbols always have an integer number of cycles within the FFT interval as long as the multipath delay spread is less than the guard time.

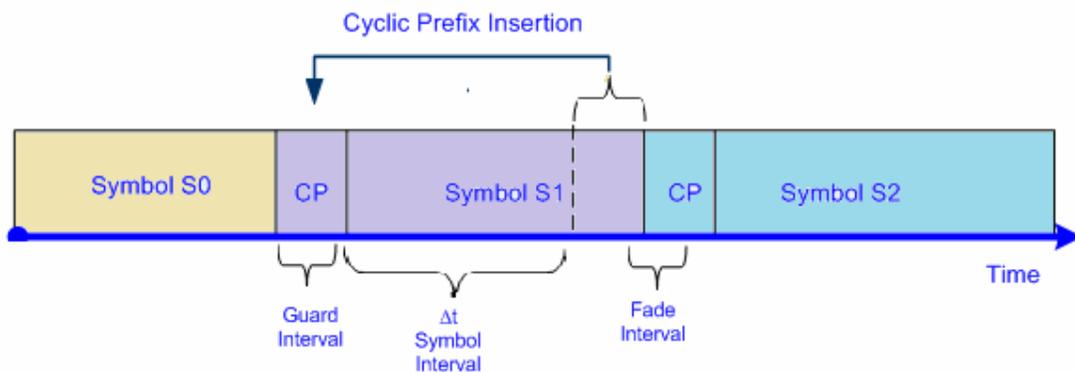


Figure 2-6 : Guard time and Cyclic Extension-Effect of Multipath

As can be seen from the previous discussion, OFDM requires very accurate frequency synchronization between the receiver and the transmitter. However, the less sensitivity of the OFDM to timing synchronization is considered one of the main features of OFDM. Pilot signals and training symbols may also be used for time synchronization (to avoid ISI) [17] and frequency synchronization (to avoid ICI, caused by Doppler shift). In order to clarify these points, frequency and time synchronization are to be discussed below.

OFDM is very sensitive to frequency synchronization results low of performance channel communication. Frequency error occurs from main two resources that local oscillator error and Doppler spread. Figure 2-7 illustrates the frequency synchronization results in a carrier-frequency offset $\Delta f = f_r - f_t$ due to the mismatch between the oscillators of the transmitter and the receiver [15].

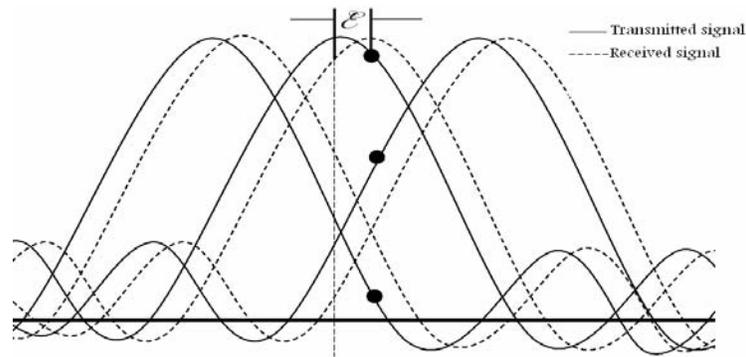


Figure 2-7: Frequency Synchronization Error.

The error in timing synchronization, on the other hand, causes a timing offset τ due to misplacement of the DFT window. Timing synchronization error is roughly divided into the two groups that towards to CP and away from CP can be found out by name. A figure 2.8 show synchronization error occurs in the transmitting symbol belongs to first group.

However, the second group uses a training sequence or pilot symbols for synchronization time estimation. It has high accuracy, but low bandwidth and hence, transmission data rate.

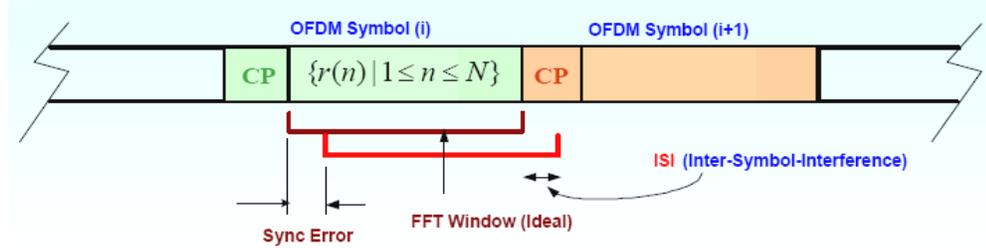


Figure 2-8: Time Synchronization Error.

2.3.3 Additive White Gaussian Noise (AWGN) Channel

In the study of communication systems, the classical AWGN channel, with statistically independent Gaussian noise samples corrupting data samples free of ISI, is the usual starting point for understanding basic performance relationships. An AWGN channel adds white Gaussian noise to the signal that passes through it. The signal subjected to an AWGN and received at the receiver is given by

$$r(t) = s(t) + n(t) . \quad (2.5)$$

Where $s(t)$, $n(t)$ and $r(t)$ representing the transmitted signal, the noise process, and the received signal, respectively.

2.4 Features of OFDM System

Sum of looked out OFDM parameters below can be generally separated into pros and cons aspects. Brief mentions of the major features of OFDM are given below:

- Can easily adapt to severe channel conditions without complex equalization
- Robust against narrow-band co-channel interference
- Robust against ISI and fading caused by multipath propagation

- High spectral efficiency⁵
- Efficient implementation using FFT
- Unlike conventional FDM, tuned sub-channel receiver filters are not required.

On the other hand, however, there are some disadvantages of the OFDM system need to be mentioned, such as

- Sensitive to Doppler shift
- Sensitive to frequency synchronization problems[17]
- High peak-to-average-power ratio (PAPR), requiring linear transmitter circuitry, which suffers from poor power efficiency.

⁵ The spectral efficiency is defined as the number of bits per second per hertz.

CHAPTER THREE

Chapter 3 Channel Coding and Viterbi Algorithm

In typical digital communications system, bits are transmitted through a communication channel and are therefore susceptible to several kinds of noises. When employing a Forward Error Correction (FEC) technique in a digital communication system, the transmitted data is encoded in such a way so that the receiver can correct, as well as detect, errors caused by channel noise. The following section describes briefly several types of coding techniques used broadly in digital communication systems.

3.1 FEC Coding Techniques

FEC is widely used in digital telecommunications systems. Although, a number of redundant bits are to be added to the information message, coding allows reducing of received BER, and hence, improves the system performance. There are several types of FEC employed in digital systems, such as, Repeated Codes, Hamming Codes, Extended Codes, Reed-Solomon (RS) Codes, Convolutional Codes, etc. In general, the FEC technique adds redundant bits to the transmitted information that are used, in the receiver, to either detect or correct the error. Accordingly, redundant bits hold no information (data), and hence, they are used to correct the error and enhance the performance of the system on the expenses of the available bandwidth.

Simply, in the repeated codes, each bit is repeated a certain n times and then use the majority as the correct bit. If less than $n/2$ errors occur in each group then the message will be decoded correctly. As the number of repetitions (n) increases, the probability of error decreases. But that's means an increase in the amount of data being sent.

This is a very simple method of error correction. It's suitable when there is unlimited bandwidth, but it is cost much on a practical scale. Moreover, when there is a need to encrypt

the data for security; knowing that the information is repeated allows for significantly easier cracks [18].

To overcome the problem occurred in the repeated codes, it would be more efficient utilization of the available bandwidth to add only one redundant bit to detect whether the received codeword is corrupted by error or not. Such a technique is called a parity check code, where if it is required to even the number of 1's in a single codeword, we add bit "1" to the tail of the codeword if the codeword was of odd number of 1's. In this case, we have an even parity check. Similar analogy is applied to odd parity check.

The parity-check adds "1" when the number of 1's is odd and adds "0" when the number of 1's is even to the end of the chunk of these data. For example, if we were attempt to send the string "00111", and then the encoded data are "001111". Therefore if one bit is flipped, the receiver knows that there is an error occurred. However, it would not be able to determine which bit was flipped. Moreover, when two bits were flipped the receiver can't detect that there is an error occurs.

Therefore a single parity-check bit could only check the accuracy of a message and would be unable to fix it. More complicated methods used multiple parity bits to gain corrective capabilities [19].

Hamming Code is considered as the first leap towards error correcting codes which was firstly introduced by R. W. Hamming in 1950. The method is still under improvement until these days. Hamming Codes send d information bits padded with a specific k parity-check bits. They have the ability to correct any single mistake. The positions of the k bits in a codeword is distributed as $2^0, 2^1, 2^2, \dots, 2^{k-1}$.

This code is significantly more efficient than pure repetition but it only has the capacity to correct one error per block of code. If two bits are flipped it will incorrectly decode to a different string. Hamming added a further adjustment to his code to allow for the detection of

up to two errors. By adding an additional parity-checking bit to the end of the string which checks the entire string. Hamming codes can detect any two bit errors even though they can still only correct a single error [21].

Reed-Solomon codes are systematic linear block codes. It is considered as a block code because the original message is split into fixed length blocks and each block is split into m -bits symbols, linear because each m -bits symbol is a valid symbol, and systematic because the transmitted information contains the original data with extra Cyclic Redundancy Check (CRC) or 'parity' bits appended.

The Reed-Solomon encoder reads k data symbols computes the r parity symbols, and appends the parity symbols to the k data symbols for a total of n symbols. The encoder is essentially a $2t$ tap shift register where each register is of m bits wide. The multiplier coefficients are the coefficients of the RS generator polynomial. The general idea is the construction of a polynomial where the produced coefficients will be symbols such that the generator polynomial exactly divides the data/parity polynomial.

A Reed-Solomon decoder can correct up to t symbols that contain errors in a codeword, where $2t = r = n-k$ [20]. Figure 3-1 shows a typical Reed-Solomon codeword.

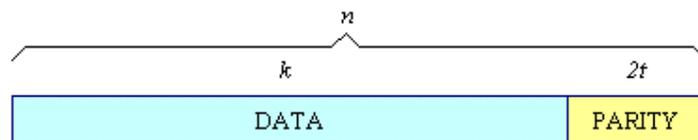


Figure 3-1: Reed-Solomon Codeword

3.2 Convolutional Codes

In the aforementioned FEC techniques, each block of data is encoded as an independent entity. Convolutional encoders encode bits based upon a state which is determined by summing a fixed set of previous bits. Each input bit produces several output bits and state key passed from encoder to decoder. Because of the integral part of this key used for decoding, these codes are often used for cryptography [18].

The encoding algorithms depend on two variables; the constraint length, which is the length or the number of states in the shift register flow, and the coding rate that is defined as the ratio of output bits to the input bits. In general, convolutional codes are significantly better at approaching the theoretical Shannon limit than prior error correcting codes. They are fast, efficient and accurate. Unfortunately, their accuracy varies significantly depending on the input. In specific, convolution codes have specific codewords where their accuracy decreased, some codewords are only separated by a Hamming distance of one.

The Convolutional encoder can be implemented by shift registers. The outputs of the encoder are the logical exclusive sum of the values in the certain register's cells. These generated outputs are presented by mathematical polynomials. Let m be the maximum degree of the polynomials generating a code, then $K = m + 1$ is a constraint length of the code. For example, in Figure 3-2, the polynomials functions are given by

$$G_1(z) = 1 + z^2$$

$$G_2(z) = 1 + z^1 + z^2$$

The generator polynomials are usually denoted in the octal notation. For the above example, these designations are “101” = 5 and “111” = 7, respectively.

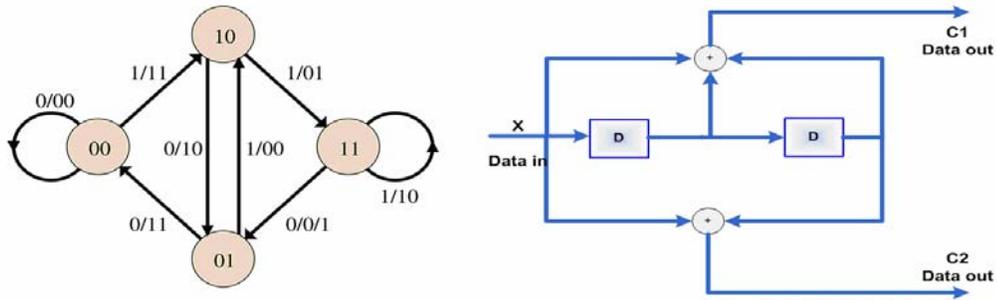


Figure 3-2: State Diagram and Convolutional Encoder

A convolutional encoder is often seen as a finite state machine. Each state corresponds to some value of the encoder's register. Given the input bit value, from a certain state the encoder can move to two other states. These state transitions constitute a diagram which is called a state diagram that is shown in Figure 3-2.

A trellis diagram derived from the above state diagram is shown in Figure 3-3 which is related to the convolutional encoder shown in the Figure 3-2. A solid line corresponds to input 0 and dotted line to input 1. Sequences from the encoder's output can be represented as a path on the trellis diagram. One of the possible paths for a specific data is illustrated by the bold line.

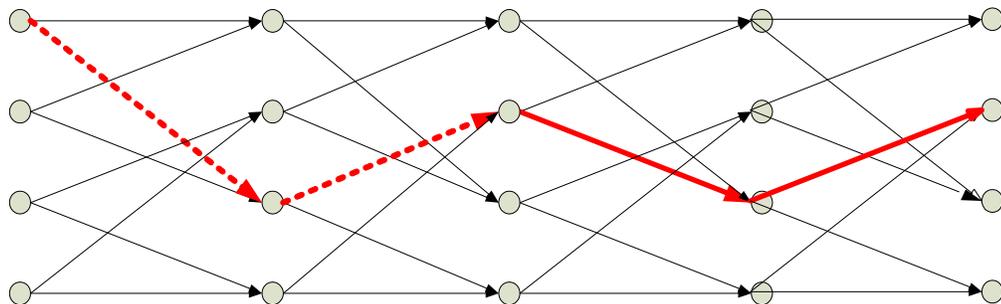


Figure 3-3: Example of Trellis Diagram.

For a convolutional encoder of coding rate equal to $\frac{1}{2}$, as shown in the previous example, each state transition on the diagram corresponds to a pair of output bits. There are only two allowed transitions for every state, so there are two allowed pairs of output bits, and the two other pairs are not defined. If an error occurs, it is very likely that the receiver will get a set of undefined pairs, which do not constitute a path on the trellis diagram. So, the task of the decoder is to find a path (hereinafter, we call it the survival path) on the trellis diagram which is the closest match to the received sequence [18].

3.3 Viterbi Algorithm

We will use the code generated by the encoder in Figure 3-2 as an example. Ideally, Viterbi algorithm reconstructs the maximum-likelihood path (survival path) for a given input sequence. Bits are receiving from the channel with some kind of reliability estimate. There are two types of decisions taken on the decoded data, Hard Decision and Soft Decision.

The distance between the received pair of bits and one of the outputs (“00”, “01”, “10”, “11”) is called branch metric and the sum of metrics of all branches in the path called path metric.

Let's suppose that for every possible encoder state we know a path with minimum metric ending in this state. For any given encoder state, there are two states from which the encoder can move to that state, and for both of these transitions we know branch metrics. So, there are only two paths ending in any given state on the next step. One of them has smaller metric, it is a survivor path. The other path is dropped. Thus we know a path with minimum metric on the next step, and the above procedure can be repeated, for more details on the decoding process, please consult as examples, the references [23] [24].

In the following, the implementation of the Viterbi decoder is given in brief. The Viterbi algorithm consists of three major parts shown in Figure 3-4.

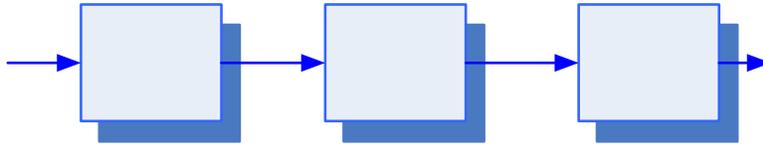


Figure 3-4: Viterbi Decoder Block Diagram.

Encoded Stream

Branch Metric Calculation: As mentioned before, methods of branch metric calculation are different for hard decision and soft decision decoders. For hard decision decoders, a branch metric is measured using the Hamming distance. Therefore, a branch metric can take values of 0, 1 and 2. Thus for every input pair we have 2 branch metrics.

Branch Metric Calculations

For a soft decision decoder, a branch metric is measured using the Euclidean distance. To explain this let R_x be the first received bit in the pair and R_y is the second, x_0 and y_0 are the ideal values. Then branch metric is [24]:

$$BM = (R_x - x_0)^2 + (R_y - y_0)^2. \quad (3.1)$$

Furthermore, when we calculate 2 branch metrics for a soft decision decoder, we don't actually need to know absolute metric values, only the difference between them makes sense. So, nothing will change if we subtract one value from the two branch metrics,

$$BM = (R_x^2 - 2R_x x_0 + x_0^2) + (R_y^2 - 2R_y y_0 + y_0^2)$$

$$BM^* = BM - (R_x^2 + R_y^2) = (x_0^2 - 2R_x x_0) + (y_0^2 - 2R_y y_0) \quad (3.2)$$

Path Metric Calculation: Path metrics are calculated using the Add-Compare-Select (ACS).

This procedure is repeated for every encoder state [25].

1. **Add** – for a given state, there are only two previous states that yield to this state, and the output bit pairs that correspond to these transitions. To calculate new path metrics, we add the old path metrics with the new branch metrics.

2. *Compare, select* – we now have two paths, ending in a given state. The one with minimum metric, i.e., minimum hamming distance, is considered.

While there are 2^{K-1} encoder states, we have 2^{K-1} survivor paths at any given time [26]. The problem with path metrics is that they tend to grow constantly and will eventually overflow. But, since the absolute values of path metric don't actually matter we can at any time subtract an identical value from the metric of every path. It is usually done when all path metrics exceed a chosen threshold. This method is simple, but not very efficient when implemented in hardware. Another way allows overflow, but uses a sufficient number of bits to be able to detect whether the overflow took place or not. The compare procedure must be modified in this case [27].

Trace back: The general approach to trace back is to accumulate path metrics for up to five times the constraint length, find the node with the minimal accumulated cost, and begin trace back from this node. However, computing the node which has accumulated the minimal cost smallest integral path metric involves finding the minima of four states in our example.

Figure 3-5 show a trills diagram with all survivor paths. The decoding depth D is an important parameter and should be large for quality of decoding, no less than $5K$. Increasing D decreases the probability of decoding error. On the other hand, increases latency.

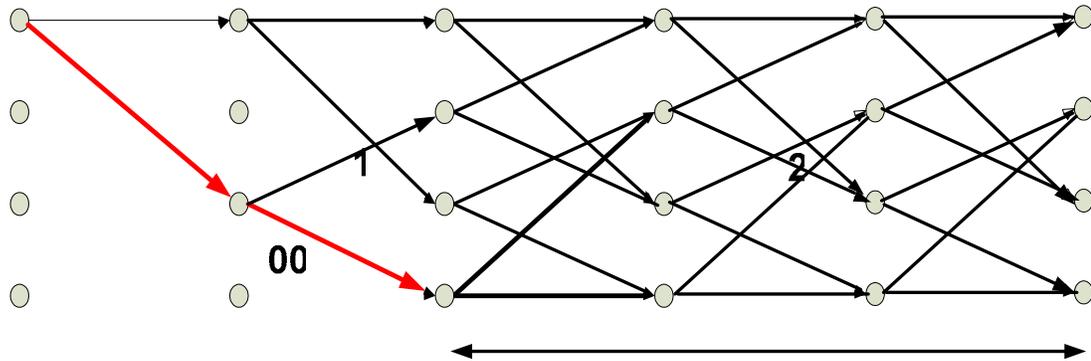


Figure 3-5: Survivor Path Graph Example

Another important parameter is the number of bits we are sending to the output after each trace back. For example, $N = 1$, the latency is minimal, but the decoder needs to trace the whole tree every step. It is computationally ineffective. In hardware implementations N optimally equals D [22].

3.4 Performance Issues

The amount of space is linear in K , the constraint length, and the encoder is much easier to implement than the Viterbi decoder. The decoding time depends mainly on K , it is represented by the exponential function $O(2^K)$, so the time complexity is exponential in K . Moreover, as described, we can decode the first bits of the message only at the end. Although a little future knowledge is useful, it is unlikely that what happens at bit time 1000 will change our decoding decision for bit 1, if the constraint length is 5, as an example. Practically, the decoder starts to decode bits once it has reached a time step that is a small multiple of the constraint length [28]. In simulation, we considered the trace back and start decoding bits when all the bits are received.

CHAPTER FOUR

Chapter 4 FFT and DWT Based OFDM

In general, signals in their nature representation forms, are time-amplitude representations, called time-domain signals which are often needed to be transformed into other different domains for analysis and processing, like frequency domain, time-frequency domain, etc. Transformation of signals helps in identifying distinct information which might otherwise be hidden in the original signal. Depending on the application, the transformation technique is chosen, and each technique has its advantages and disadvantages [30].

In most Digital Signal Processing (DSP) applications, the frequency content of the signal is very important. The Fourier transform is probably the most popular transform used to obtain the frequency spectrum of a signal. But the Fourier transform is only suitable for stationary signals, i.e., signals whose frequency content does not change with time. The Fourier transform tells how much of each frequency exists in the signal, but does not tell at which time these frequency components occur.

The major drawback of the FFT is that it uses a fixed window width. On the other hand, the Wavelet transform, which was developed in the last two decades, provides a better time-frequency representation of the signal than any other existing transforms.

4.1 Comparison between Fourier Transform and Wavelet Transform

The FFT is a modified version of the Fourier transform. The Fourier transform separates the waveform into a sum of sinusoids of different frequencies and identifies their respective amplitudes. Thus it gives us a frequency-amplitude representation of the signal. In FFT, the non-stationary signal is divided into small portions, which are assumed to be stationary. This is done using a window function of a chosen width, which is shifted and multiplied with the

signal to obtain the small stationary signals. The Fourier transform is then applied to each of these portions to obtain the Short Time Fourier transform of the signal.

The problem with FFT goes back to the Heisenberg uncertainty principle [32] which states that it is impossible for one to obtain which frequencies exist at which time instance, but, one can obtain the frequency bands existing in a time interval. This gives rise to the resolution issue where there is a trade-off between the time resolution and frequency resolution. To assume stationary, the window is supposed to be narrow, which results in a poor frequency resolution, i.e., it is difficult to know the exact frequency components that exist in the signal; only the band of frequencies that exist is obtained. If the width of the window is increased, frequency resolution improves but time resolution becomes poor, i.e., it is difficult to know what frequencies occur at which time intervals. Also choosing a wide window may violate the condition of stationary. Consequently, depending on the application, a compromise on the window size has to be made. Once the window function is decided, the frequency and time resolutions are fixed for all frequencies and all times.

The Wavelet transform solves the above problem to a certain extent. In contrast to FFT, which uses a single analysis window, the Wavelet transform uses short windows at high frequencies and long windows at low frequencies. This results in multi-resolution analysis by which the signal is analyzed with different resolutions at different frequencies, i.e., both frequency resolution and time resolution vary in the time-frequency plane without violating the Heisenberg inequality.

In Wavelet transform, as frequency increases, the time resolution increases; likewise, as frequency decreases, the frequency resolution increases. Thus, a certain high frequency component can be located more accurately in time than a low frequency component and a low frequency component can be located more accurately in frequency compared to a high frequency component.

Figure 4-1(a) shows the time-frequency tiling in the time-domain plane and Figure 4-1 (b) shows the frequency-domain plane. It is seen that Figure 4-1 (a) does not give any frequency information and Figure 4-1 (b) does not give any time information. Similarly Figure 4-1 (c) shows the tiling in FFT and Figure 4-1 (d) shows the tiling in Wavelet transform. It is seen that FFT gives a fixed resolution at all times, whereas Wavelet transform gives a variable resolution.

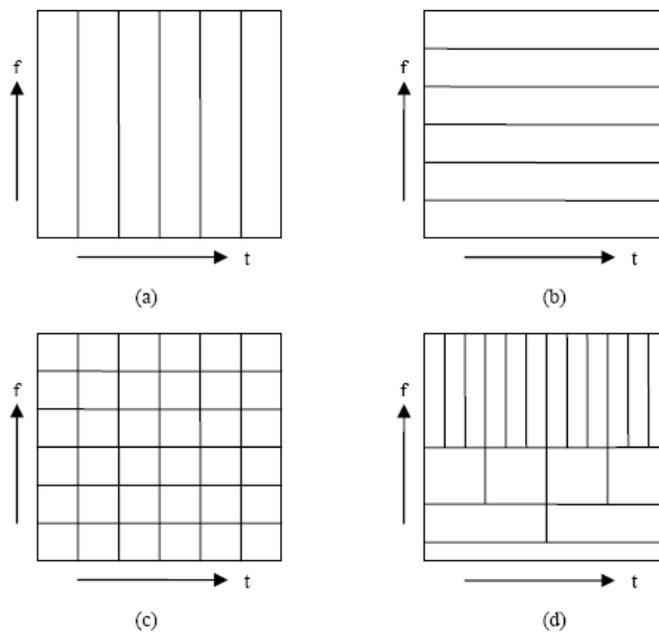


Figure 4-1: Time-Frequency (a) Time-Domain (b) Frequency-Domain (c) FFT (d) DWT

4.2 Wavelet Transform Technology

The wavelet idea can be analyzed in a similar manner to that of FFT. The signal to be analyzed is multiplied with a wavelet function just as it is multiplied with a window function in FFT, and then the transform is computed for each segment generated. However, unlike

FFT, in Wavelet Transform, the width of the wavelet function changes with each spectral component. The Wavelet Transform, at high frequencies, gives good time resolution and poor frequency resolution, while at low frequencies; the Wavelet Transform gives good frequency resolution and poor time resolution [34].

4.2.1 Continuous Wavelet Transform (CWT)

The Continuous Wavelet Transform (CWT) is given in (4-1), where we assume that $x(t)$ is the signal under consideration, i.e., the signal to be analyzed. $\Psi(t)$ is the “mother wavelet” or the basis function. All the wavelet functions used in the transformation are derived from the mother wavelet through shifting and scaling

$$X_{WT}(\tau, s) = \frac{1}{\sqrt{|s|}} \int x(t) \cdot \Psi^* \left(\frac{t - \tau}{s} \right) dt \quad (4-1)$$

The mother wavelet used to generate all the bases functions is designed according to some desired characteristics associated with that function. The translation parameter τ relates to the location of the wavelet function as it is shifted through the signal. Thus, it corresponds to the time information in the Wavelet Transform. The scale parameter s is defined as $|1/\text{frequency}|$ and corresponds to frequency information. Scaling either expands or compresses a signal. Low frequencies expand the signal and provide detailed information hidden in the signal, while high frequencies compress the signal and provide global information about the signal. Notice that the Wavelet Transform simply performs the convolution operation of the signal and the basis function. The above analysis becomes very useful as in most practical applications, high frequencies (low scales) do not last for a long duration, but instead, appear as short bursts, while low frequencies (high scales) usually last for entire duration of the signal.

The Wavelet Series is obtained by forming discretized CWT. This aids in computation of CWT using computers and is obtained by sampling the time-scale plane. The sampling rate can be changed accordingly with scale change without violating the Nyquist criterion. Nyquist criterion states that, the minimum sampling rate that allows reconstruction of the original signal is 2ω radians, where ω is the highest frequency in the signal. Therefore, as the scale goes higher (lower frequencies), the sampling rate can be decreased thus reducing the number of computations.

4.2.2 The Discrete Wavelet Transform (DWT)

The Wavelet Series is just a sampled version of CWT and its computation may consume significant amount of time and resources, depending on the resolution required. The DWT which is based on sub-band coding is found to yield a fast computation of Wavelet Transform. It is easy to implement and reduces the computation time and resources required. In CWT, the signals are analyzed using a set of bases functions which relate to each other by simple scaling and translation. In the case of DWT, a time-scale representation of the digital signal is obtained using digital filtering techniques. The signal to be analyzed is passed through filters with different cutoff frequencies at different scales.

4.3 DWT and Filter Banks

4.3.1 Multi-Resolution Analysis using Filter Banks

Filters are one of the most widely used signal processing functions. Wavelets can be realized by iteration of filters with rescaling. The resolution of the signal, which is a measure of the amount of detail information in the signal, is determined by the filtering operations, and the scale is determined by up-sampling and down-sampling (sub-sampling) operations [35].

The DWT is computed by successive lowpass and highpass filtering of the discrete time-domain signal as shown in Figure 4-2. This is called the Mallat algorithm or Mallat-tree decomposition. Its significance is in the manner it connects the continuous-time multi-resolution to discrete-time filters. In the figure, the signal is denoted by the sequence $x[n]$, where n is an integer. The lowpass filter is denoted by G_0 while the highpass filter is denoted by H_0 . At each level, the highpass filter produces detail information $d[n]$, while the lowpass filter associated with scaling function produces rough approximations, $a[n]$.

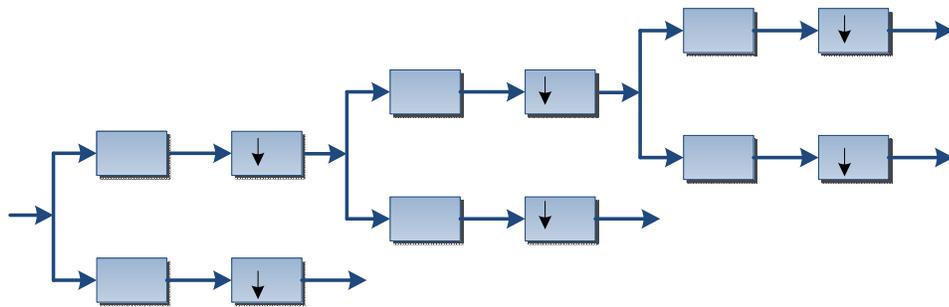


Figure 4-2: Three-Level Wavelet Decomposition Tree.

At each decomposition level, the half band filters produce signals spanning only half the frequency band. This doubles the frequency resolution as the uncertainty in frequency is reduced by half. In accordance with Nyquist rule if the original signal has a highest frequency of ω_0 , which requires a sampling frequency of $2\omega_0$ radians, then it has a highest frequency of $\omega_0/2$ radians. It can now be sampled at a frequency of ω_0 radians thus discarding half the samples with no loss of information. This decimation by 2, halves the time resolution as the entire signal is now represented by only half the number of samples. Thus, while the half band lowpass filtering removes half of the frequencies and thus halves the resolution, the decimation by 2 doubles the scale.

$X[n]$

41

G_0

2

$a_1[n]$

G_0

H_0

2

$d_1[n]$

H_0

With this approach, the time resolution becomes arbitrarily good at high frequencies, while the frequency resolution becomes arbitrarily good at low frequencies. The time-frequency plane is thus resolved as shown in Figure 4-1(d). The filtering and decimation process is continued until the desired level is reached. The maximum number of levels depends on the length of the signal. The DWT of the original signal is then obtained by concatenating all the coefficients, $a[n]$ and $d[n]$, starting from the last level of decomposition,

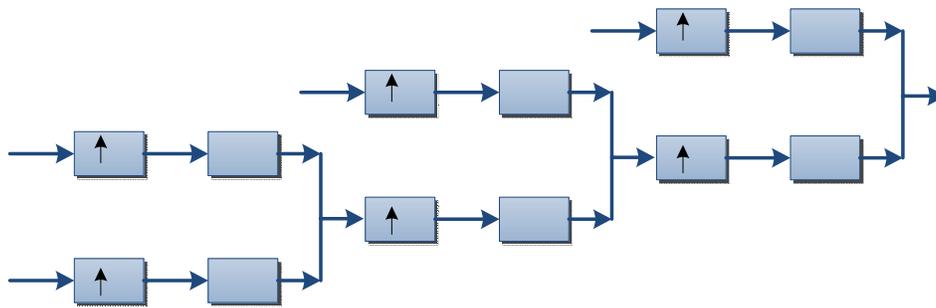


Figure 4-3: Three-Level Wavelet Reconstruction Tree.

Figure 4-3 shows the reconstruction of the original signal from the wavelet coefficients. Basically, the reconstruction is the reverse process of decomposition. The approximation and detail coefficients at every level are up-sampled by two, passed through the lowpass and highpass synthesis filters and then added. This process is continued through the same number of levels as in the decomposition process to obtain the original signal. The Mallat algorithm works equally well, if the analysis filters, G_0 and H_0 , are exchanged with the synthesis filters, G_1 and H_1 .

4.3.2 Conditions for Perfect Reconstruction

In most Wavelet Transform applications, it is required that the original signal be synthesized from the wavelet coefficients. To achieve perfect reconstruction the analysis and synthesis

$d_3[n]$

2

H_1

2

$a_2[n]$

2

$a_3[n]$

42

2

G_1

filters have to satisfy certain conditions. Let $G_0(z)$ and $G_1(z)$ be the lowpass analysis and synthesis filters, similarly, let $H_0(z)$ and $H_1(z)$ be the highpass analysis and synthesis filters respectively. Then the filters have to satisfy the following two conditions as given in

$$G_0(-z)G_1(z) + H_0(-z)H_1(z) = 0 \quad (4.2)$$

$$G_0(z)G_1(z) + H_0(z)H_1(z) = 2z^{-d} \quad (4.3)$$

The first condition implies that the reconstruction is aliasing-free and the second condition implies that the amplitude distortion has amplitude of one. It can be observed that the perfect reconstruction condition does not change if we switch the analysis and synthesis filters.

There are a number of filters which satisfy these conditions. But not all of them give accurate Wavelet Transforms, especially when the filter coefficients are quantized. The accuracy of the Wavelet Transform can be determined after reconstruction by calculating the SNR of the signal. Some applications like pattern recognition do not need reconstruction, and in such applications, the above conditions need not apply. But in OFDM implementation we need both of them.

4.4 Classification of wavelets

We can classify wavelets into two classes, orthogonal wavelet and biorthogonal wavelet. Based on the application, either of them can be used.

4.4.1 Orthogonal wavelet filter banks

The coefficients of orthogonal filters are real numbers. The filters are of the same length and are not symmetric. The lowpass filter G_0 and the highpass filter H_0 are related to each other by

$$H_0(z) = z^{-N} G_0(-z^{-1}) \quad (4.4)$$

The two filters are alternated flip of each other. The alternating flip automatically gives double-shift orthogonality between the lowpass and highpass filters, i.e., the scalar product of the filters, for a shift by two is zero. i.e., $\sum G[k]H[k-2l]=0$, where $k, l \in Z$ [30]. Filters that satisfy (4.4) are known as Conjugate Mirror Filters (CMF). Perfect reconstruction is possible with alternating flip.

Furthermore, for perfect reconstruction, the synthesis filters are identical to the analysis filters except for a time reversal. Orthogonal filters offer a high number of vanishing moments. This property is useful in many signal and image processing applications. They have regular structure which leads to easy implementation and scalable architecture.

4.4.2 Biorthogonal wavelet filter banks

In the case of the biorthogonal wavelet filters, the lowpass and the highpass filters do not have the same length. The lowpass filter is always symmetric, while the highpass filter could be either symmetric or asymmetric. The coefficients of the filters are either real numbers or integers.

For perfect reconstruction, biorthogonal filter bank has all odd length or all even length filters. The two analysis filters can be symmetric with odd length or one symmetric and the other asymmetric with even length. Also, the two sets of analysis and synthesis filters must be dual. The linear phase biorthogonal filters are the most popular filters for data compression applications.

4.5 Wavelet Families

There are a number of bases functions that can be used as the mother wavelet for Wavelet Transformation. Since the mother wavelet produces all wavelet functions used in the transformation through translation and scaling, it determines the characteristics of the

resulting Wavelet Transform. Therefore, the details of the particular application should be taken into account and the appropriate mother wavelet should be chosen in order to use the Wavelet Transform effectively.

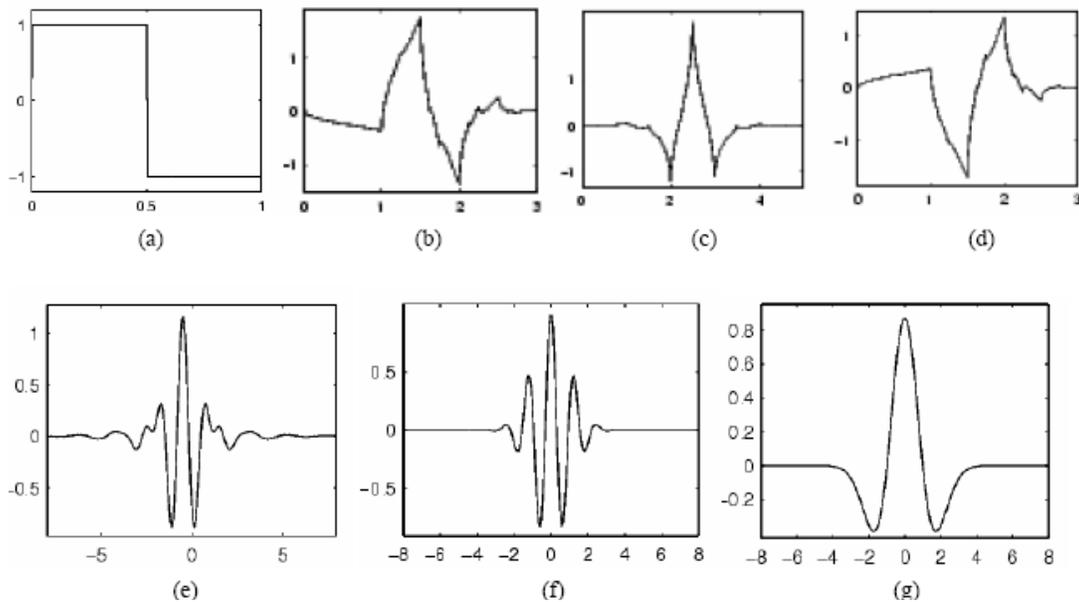


Figure 4-4: Wavelet Families (a) Haar (b) Daubechies4 (c) Coiflet1 (d) Symlet2 (e) Meyer (f) Morlet (g) Mexican Hat.

Figure 4-4 illustrates some of the commonly used wavelet functions. Haar wavelet is one of the oldest and simplest wavelet. Therefore, any discussion of wavelets starts with the Haar wavelet. Daubechies wavelets are the most popular wavelets. They represent the foundations of wavelet signal processing and are used in numerous applications. These are also called Maxflat wavelets as their frequency responses have maximum flatness at frequencies 0 and π . This is a very desirable property in some applications. The Haar, Daubechies, Symlets and Coiflets are compactly supported orthogonal wavelets. These wavelets along with Meyer wavelets are capable of perfect reconstruction. The Meyer, Morlet and Mexican Hat

wavelets are symmetric in shape. The wavelets are chosen based on their shape and their ability to analyze the signal in a particular application [33].

CHAPTER FIVE

Chapter 5 Simulation Model, and Results

In this chapter, we discuss the simulation model and results of a system employed in our research. As we have mentioned before, the research goal is to improve QoS in WiMAX physical layer. This task involves modeling of the physical layer as well as the propagation environment. Simulation was chosen to be the primary tool for our study. We employed MATLAB software to develop the simulator. Let us first define the OFDM symbol parameters that are used in our simulation.

5.1 Simulated OFDM System Model

OFDM is a multi-carrier technique based on a very simple modulation scheme that uses IFFT / FFT at its core for transmission and reception respectively. The performance of the OFDM receiver is significantly affected in the presence of noise in the wireless channel. In this work a receiver that has built-in capacity with additional computation as proposed. It is based on a DWT or FFT algorithm. Experiments using MATLAB are conducted to verify the feasibility of this algorithm on OFDM signals.

This structure corresponds to the physical layer of the IEEE 802.16-2004 air interface. In this setup, I have just implemented the mandatory features of the specification. Channel coding part is very important one. Through the rest of the sections, the individual block of the system will be discussed with implementation technique.

5.1.1 FFT-based OFDM

For the current FFT-based OFDM compute a long-length IFFT at the modulator, and the demodulator needs to compute a long-length FFT. For such computations, a great number of complex multiplications are required. Clearly, the complexity of a FFT-OFDM would be

reduced if the corresponding modulator and demodulator could be replaced by DWT transforms [33]. Figure 5-1 shows the diagram of FFT-based OFDM:

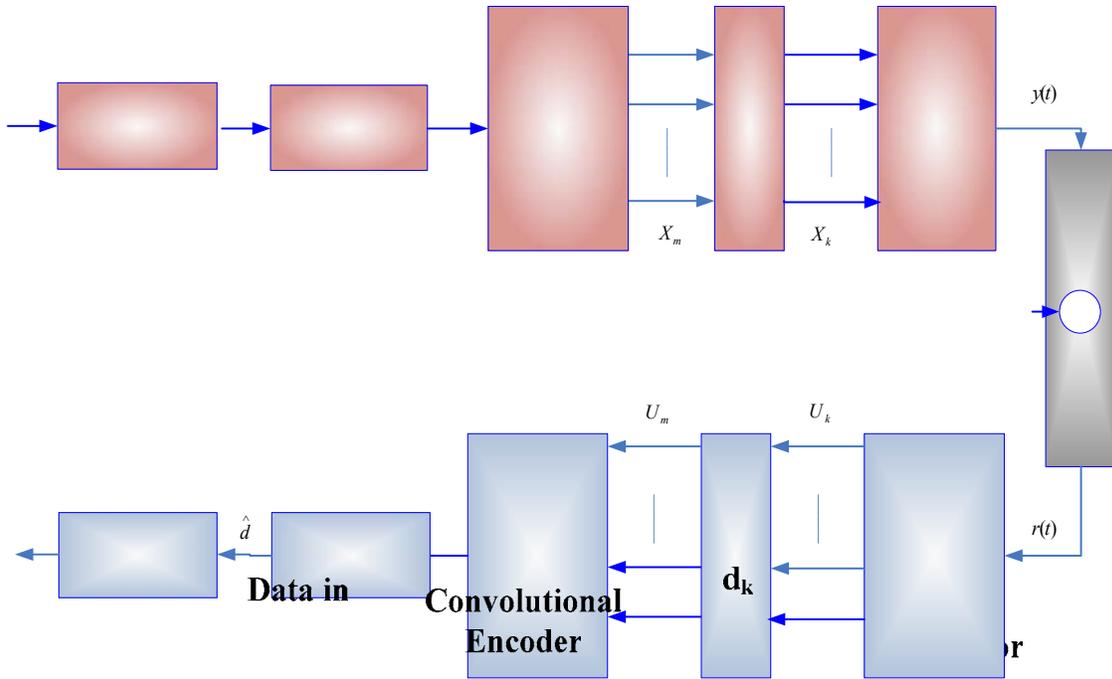


Figure 5-1: FFT-WiMAX Block Diagram.

The system model for FFT-based OFDM will not be discussed in detail as it is well known in the literature. Thus, a brief description will be presented about it. The data generator produces in random binary form. It is first being processed by a constellation mapping. BPSK modulator is used for this work to map the raw binary data to appropriate BPSK symbols. These symbols are then input into the IFFT block. This involves taking N parallel streams of BPSK symbols and performing an IFFT operation on this parallel stream. The output in discrete time domain is

$$X_k(n) = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} X_m(i) \exp(j2\pi \frac{n}{N} i) \quad (5.1)$$

Serial/P
Conv

Parallel/Se
Convert

Where x is a sequence in the discrete time domain and U_k are complex numbers in the discrete frequency domain. Then the CP is lastly added before transmission and the signal will be transmitted.

At the receiver, the process is reversed to obtain the decoded data. The CP is removed to obtain the data in the discrete time domain and then processed to FFT for data recovery. The output of the FFT in the frequency domain is as follows:

$$U_m(i) = \sum_{n=0}^{N-1} U_k(n) \exp(-j2\pi \frac{n}{N}i) \quad (5.2)$$

The MATLAB functions “ifft”, “fft” are used to incorporate the IFFT/FFT functions.

5.1.2 DWT-based OFDM

The DWT of a signal x is calculated by passing it through a series of filters. First the samples are passed through a low pass filter with impulse response “ g ” resulting in a convolution of the two:

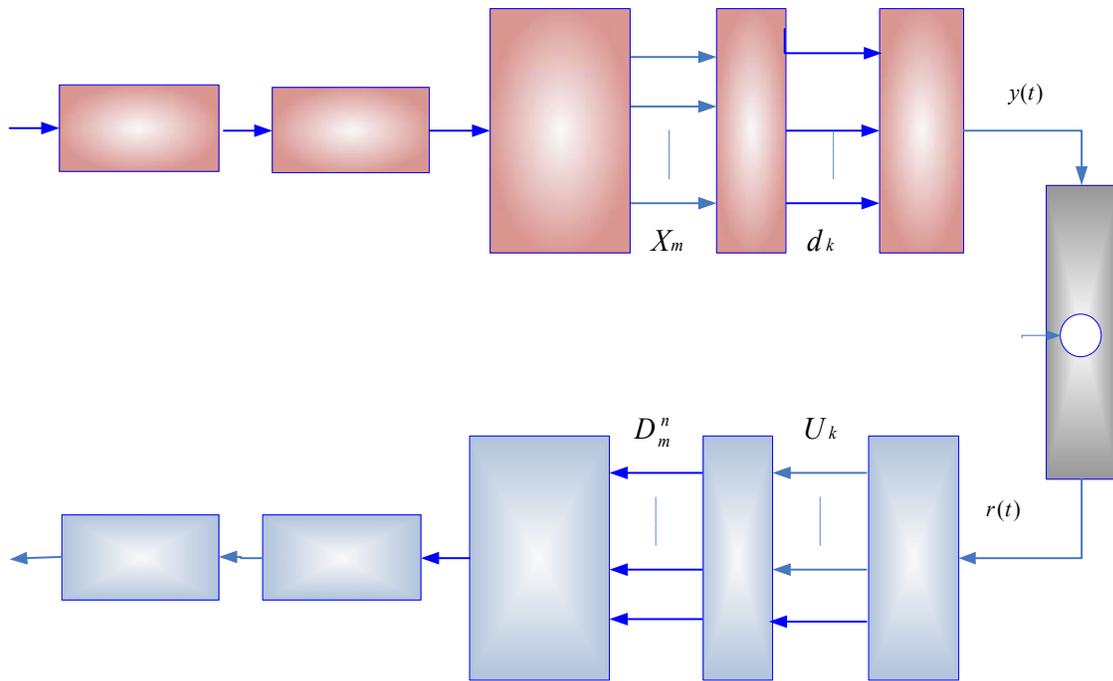
$$y[n] = (x * g) = \sum_{k=-\infty}^{\infty} x[k]g[n - k] \quad (5.3)$$

The signal is also decomposed simultaneously using a high-pass filter “ h ”. The output gives the detail coefficients (from the high-pass filter) and approximation coefficients (from the low-pass). It is important that the two filters are related to each other and they are known as a quadrature mirror filters. However, since half the frequencies of the signal have now been removed, half the samples can be discarded according to Nyquist’s rule. The filter outputs are then sub sampled by two. The outputs of the low-pass filter and the high-pass filter are shown below and are the convolutions of the input data with the respective filter responses:

$$y_{low}[n] = (x * g)[n] = \sum_{k=-\infty}^{\infty} x[k]g[2n - k] \quad (5.4)$$

$$y_{high}[n] = (x * h) = \sum_{k=-\infty}^{\infty} x[k]h[2n - k] \quad (5.5)$$

The decomposition has halved the time resolution since only half of each filter output characterizes the signal. However, each output has half the frequency band of the input so the frequency resolution has been doubled.



Data in Figure 5.2: DWT-based OFDM Block Diagram. **Encoder**

BPSK Modulator

Serial/Parallel Converter

In a DWT based system, the wavelet transform blocks, inverse discrete wavelet transform (IDWT) and discrete wavelet transform (DWT) replace the IFFT and FFT of FFT-OFDM system in modulation and demodulation processes.

One of the advantages of using wavelet transform is that due to the overlapping nature of wavelet properties, the wavelet based OFDM does not need cyclic prefix to deal with delay spreads of the channel. As a result, it has higher spectral containment than that of Fourier-based OFDM [34]. The input data is processed as per FFT-OFDM. However, the difference

Data out

BPSK

Parallel/Serial Converter

is that the system does not require CP to be added to the OFDM symbol. The output of the inverse discrete wavelet transform (IDWT) can be represented as:

$$d(k) = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} D_m^n 2^{m/2} \Psi(2^m k - n) \quad (5.5)$$

Where D_m^n are the wavelet coefficients and $\psi(t)$ is the wavelet function with compressed factor m times and shifted n times for each subcarrier (number k , $0 \leq k \leq N-1$). The wavelet coefficients are the representation of signals in scale and position or time.

At the receiver side, the process is inverted. The output of discrete wavelet transform (DWT) is:

$$D_m^n = \sum_{k=0}^{N-1} d(k) 2^{m/2} \Psi(2^m k - n) \quad (5.6)$$

Wavelet transforms have been considered as replacing the IFFT/FFT by using the IDWT/DWT blocks. In MATLAB, I used the function “dwt” and “idwt”.

5.1.3 Constellation Mapper

The data are entered serially to the constellation mapper. In MATLAB, constellation mapper support BPSK, grey mapped QPSK, 16QAM, and 64QAM. The complex constellation points are normalized with the specified multiplying factor for different modulation scheme so that equal average power is achieved for the symbols. The constellation mapped data are assigned to all allocated data subcarriers of the OFDM symbols.

5.1.4 Convolutional Encoder

Figure 5-3 shows the implementation of the convolutional encoder with rate of $\frac{1}{2}$ (generate two code bits). The generator polynomials for C1 and C2 are [7, 5] in octal format.

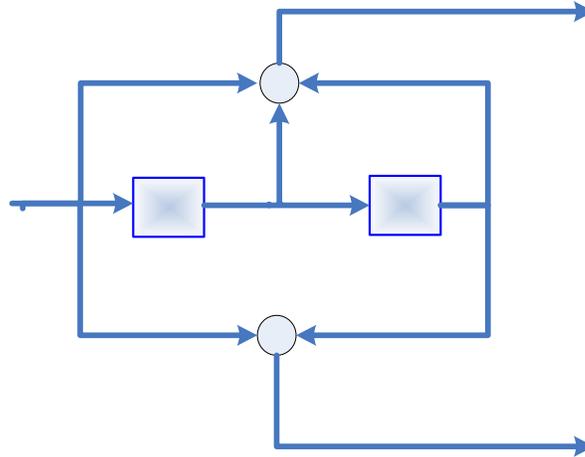


Figure 5-3: 1/2 Coding Rate Convolutional Encoder

X

D

5.1.5 Viterbi Algorithm

Data in

We can view the Viterbi algorithm as a dynamic programming algorithm for finding the shortest path through a trellis, and the algorithm can be broken down into the following steps:

- Weigh the trellis; that is, calculate the branch metrics.
- Recursively compute the shortest paths to time n , in terms of the shortest paths to time $n-1$. In this step, decisions are used to recursively update the survivor path of the signal. This is known as add-compare-select (ACS) recursion.
- Recursively find the shortest path leading to each trellis state using the decisions from Step 2. The shortest path is called the survivor path for that state and the process is referred to as survivor path decode. Finally, when all survivor paths are traced back in time, they merge into a unique path, which is the most likely signal path that we are trying to find.
- Associated with each trellis state S at time n is a state metric $PM_{s,n}$ which is the accumulated metric along the shortest path leading to that state. The state metrics at time n can be recursively calculated in terms of the state metrics of the previous iteration as follows[36]:

$$PM_{i+1} = \min \{ PM_i + BM_{i,i+1}, PM_j + BM_{j,i+1} \} \quad (5.7)$$

Where i is a predecessor state of j and $BM_{ij, n-1}$ is the branch metric on the transition from state i to state j . The qualitative interpretation of this expression is as follows. By definition, the shortest path into state j must pass through a predecessor state. If the shortest path into j passes through i , then the state metric for the path must be given by the state metric for i plus the branch metric for the state transition from i to j . The final state metric for j is given by the minimum of all possible paths. The recursive update described by (5.7) is the ACS operation and is implemented as shown in Figure 5-4 for the four-state trellis example [19].

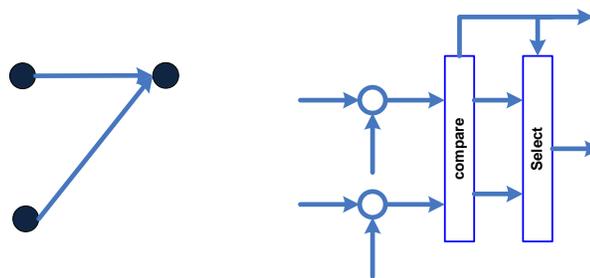


Figure 5-4: (a) Predecessor States of State i (b) State Metric Update for State i Implemented Using 2-way ACS

The update unit is referred to as a two-way ACS unit, because there are two input branches for each state. In general, a state with m -input branches requires an m -way ACS unit. As well as calculating the updated state metric, the ACS unit outputs decision d_s, n , which identify the entering path of the minimum metric [19].

Similarly, taking the look-ahead step $M=2$, skipping a node and each time jump to every third node after adding the respective branch metrics with the state metric. So instead of two possible paths now four have to be calculated which are as follows:

PM_i $BM_{i,i+1}$ PM_{i+1}
 $BM_{j,i+1}$
 PM_j

$$PM_{i+2} = \min \left\{ \begin{array}{l} PM_i + BM_{i,i+1} + BM_{i+1,i+2}, \\ PM_j + BM_{j,i+1} + BM_{i+1,i+2}, \\ PM_k + BM_{k,j+1} + BM_{j+1,i+2}, \\ PM_l + BM_{l,j+1} + BM_{j+1,i+2}, \end{array} \right\} \quad (5.8)$$

5.1.6 Channel Model

Noise exists in all communications systems operating over an analog physical channel, such as radio channel. It is therefore important to study the effects of noise on the communications error rate and some of the tradeoffs that exist between the level of noise and system spectral efficiency. Most types of noise present in radio communication systems can be modeled accurately using AWGN. This noise has a uniform spectral density (making it white), and a Gaussian distribution in amplitude (this is also referred to as a normal distribution). Thermal and electrical noise from amplification, primarily have white Gaussian noise properties, allowing them to be modeled accurately with AWGN. Also most other noise sources have AWGN properties due to the transmission being OFDM. OFDM signals have a flat spectral density and a Gaussian amplitude distribution provided that the number of carriers is large (greater than about 20 subcarriers), because of this the inter-cellular interference from other OFDM systems have AWGN properties. For the same reason ICI, ISI, and IMD also have AWGN properties for OFDM signals.

5.2 Simulation Results

The main contribution of the thesis is divided into two parts. The first part was to optimize Viterbi algorithm which supposed to lead to decrease the decoding time and increase the system throughput. The second one is to increase the BER performance using IDWT/DWT instead of IDFT/DFT. The following sections show the results for the proposed approaches and the overall system enhancement.

5.2.1 Viterbi Decoder Results

MATLAB simulations have been done in order to establish the authenticity of the proposed optimization in the Algorithm. The results are collected for two versions of the simulation, namely:

- Parallel Viterbi Decoding
- Serial Viterbi Decoding

The simulation parameters are selected according to the IEEE 802.16e standard with 1024 FFT size (number of subcarriers), symbol duration 94 (μ s) and the number of symbols per frame 104 symbols. More details on the simulation parameters are listed in Table 5-1. Seeking for simplicity, the constraint length K is chosen to be 3 and the encoding rate $r = 1/2$. Based on system requirement, our results can be generalized to any other suitable parameters according to the system requirements and applications. Another important parameter to be considered through simulation is decoding strategy; here we chose the soft-decision decoding strategy, although it is superior than the hard-decision in broad sense. Block processing; in the soft-decision decoding of data has been used instead of sample by sample processing. Unlike the hard-decision decoding, the branch metrics can be computed in the initial stage without waiting for the next bit to be received.

Table 5-1: Simulation Parameters

| Parameter specification | Parameter value |
|---|-----------------------------------|
| Data length | 10^5 bits |
| FFT size | 32, 64, 128, 512, 1024 |
| Symbol duration | 94 μ s |
| Number of symbols/Frame | 104 |
| Cyclic Prefix | 1/4, 1/8, 1/16, 1/32, 1/64, 1/128 |
| Generating polynomial | [1 1 1; 1 0 1]* |
| Coding rate | $r = 1/2$ |
| Modulation techniques | BPSK, QPSK, 16QPSK, 64QPSK |
| Decision strategy | Soft-decision |
| * The polynomial values showed in octal | |

MATLAB codes for serial and parallel Viterbi algorithms produced promising results; the decoding time of parallel decoder is much less than the time of the serial one, our results given in Figure 5-4, shows the time needed to decode 20000 bits is 0.4 second in case of parallel Viterbi while 1.75 second in the serial case, which reflects a tangible decoding time decreasing. Moreover, our results shows a decreasing time of approximately one forth, i.e., $\frac{1}{4}$ of the serial time. Furthermore, it is worthy to notice that, the tangibility of the difference getting higher in increasing the number of decoded bits. One should note that the ration between the time consuming in both cases, serial and parallel, is increased exponentially with number of decoded bits.

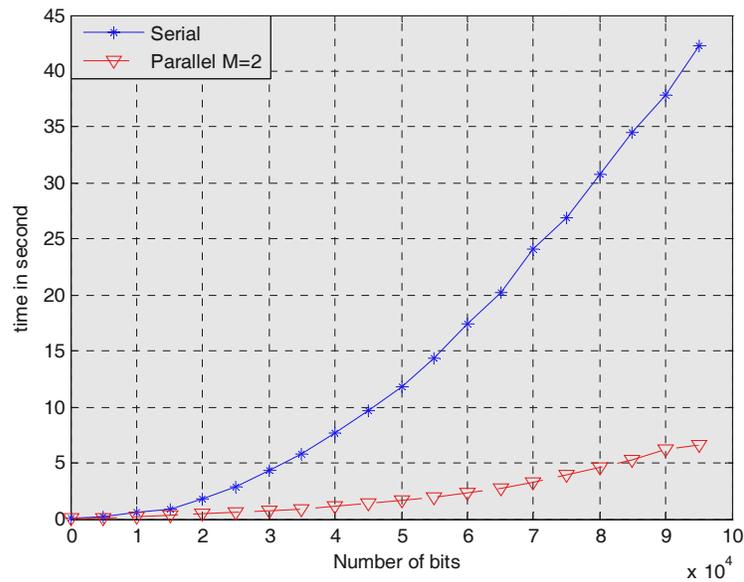


Figure 5-4: Time Needed to Decode Bits Using Serial and Parallel Viterbi Decoder (M=2).

The throughput enhancement of using parallel Viterbi decoding (M=2) over serial Viterbi is shown in Figure 5-5, where as a numerical examples, we can see that the parallel Viterbi

decoder increases the system throughput by 400% at 0 dB of the SNR, 530% at 5 dB and 550% above 7 dB.

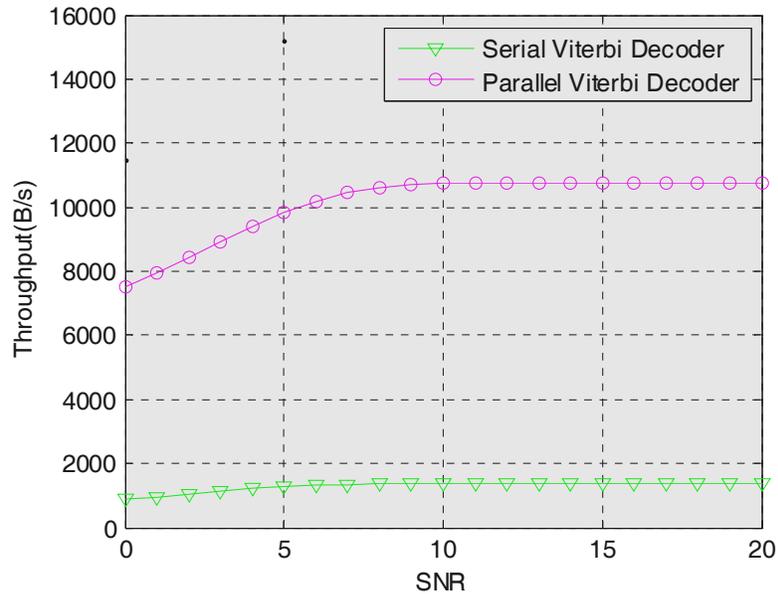


Figure 5-5: Throughput Enhancement for Parallel Viterbi Decoder with M=2.

While increasing the system throughput, for the same simulation parameter, one should care about the system performance expressed in the bit error rate. Comparing serial and parallel BER for the same simulation parameters, Figure 5-6 shows that both serial and parallel Viterbi decoders have approximately similar BER performance. That is to say, achieving throughput enhancement without deteriorating the BER system performance.

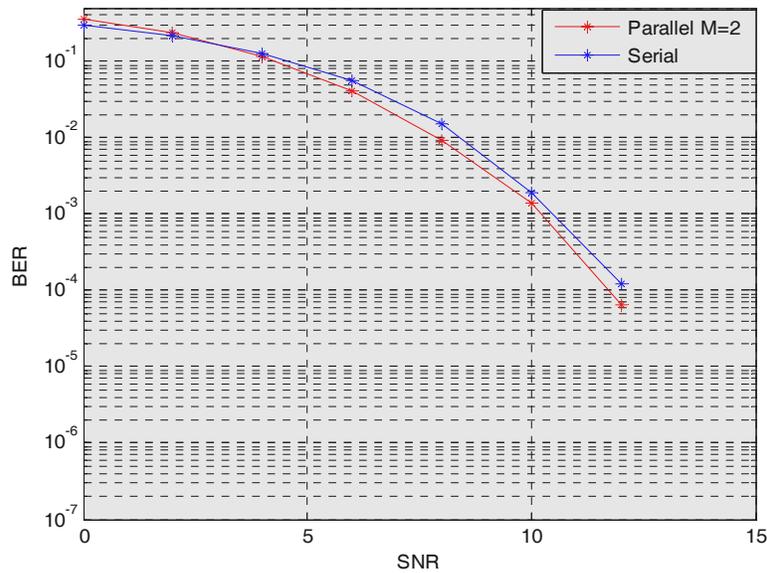


Figure 5-6: BER Performance for Serial and Parallel Viterbi with $M=2$.

Hereinafter, the Decoding Length is defined as the number of trellis steps, M , which is required to determine the decoding block of transmitted bits. Longer decoding lengths require larger amount of memory as paths should be stored before being discarded. The storage requirements grow exponentially with constraint length K . Based on theoretical analysis, numerically, for a coding rate, r , of $1/2$, a set of 2^{K-1} path must be stored after each step.

So far, the previous results were shown for $M = 2$, in Figure 5-7, the number of decoding steps is increased to $M = 3$, and $M = 4$. Our results show that the higher superiority is obtained when $M = 2$, and decreases as long as M getting larger.

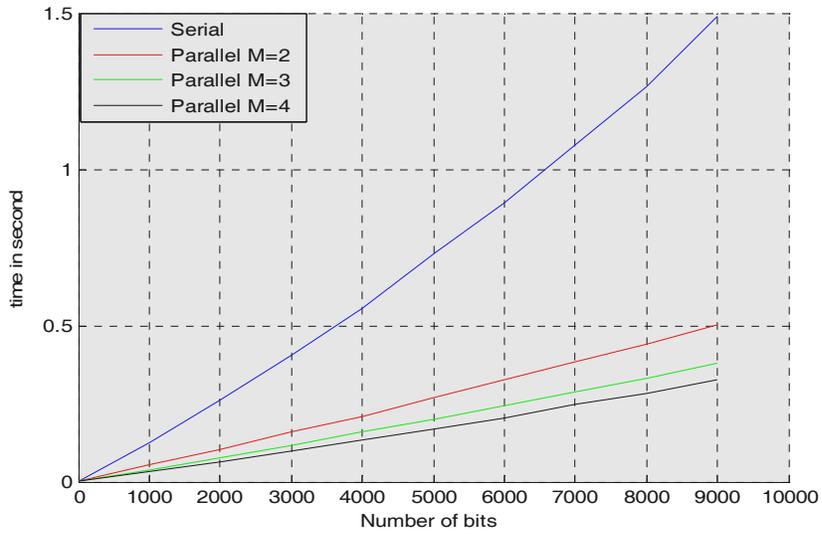


Figure 5-7: The Decoding Time for Serial, Parallel with M=2, 3, 4.

The left side of Figure 5-8 shows that when $M = 2$; each state have to calculate four different paths and chose the minimum weight of them. The other three paths will drop and will not enter the calculations again. That means 12 paths will be dropped each clock cycle in two bits Parallel Viterbi. When $M=3$ we need to calculate another eight paths for each state. When $M=4$, 16 and so on the number of different paths will be 2^n where n is the number of decoded bits.

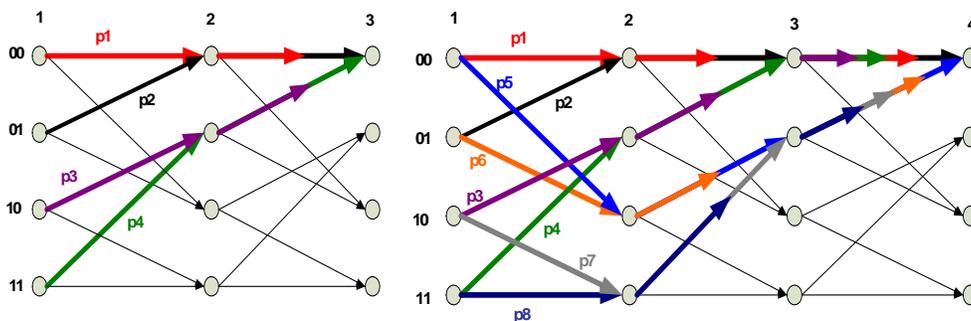


Figure 5-8: Number of Pathes for M=2 and M=3.

When the number of steps increased the number of branches needs to be compared increase exponentially as shown in Figure 5-9.

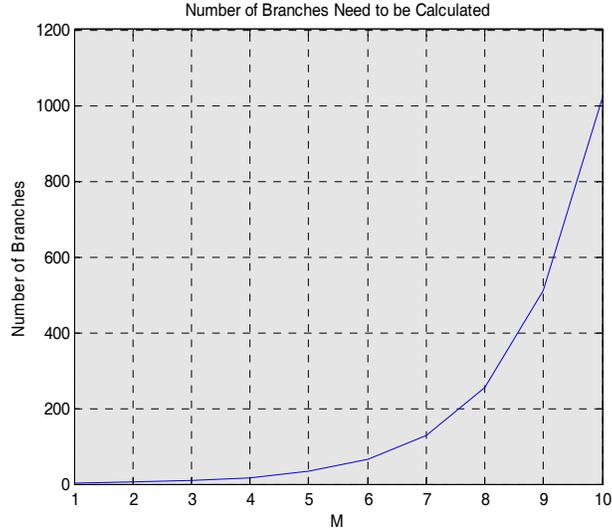


Figure 5-9: Number of branches Need to be calculated.

In terms of hardware complexity; When, $M=2$, $K=3$, the number of states is $2^{K-1} = 4$. The number of required 2-input adders and CS units can be given as $2^{K-1} \times 2^{K-1}$ and $2^{K-1}(2^{K-1} - 1)$ respectively. When, $M=3$, $K=3$, the number of required 2-input adders and CS units can be given as $2^{K-1}(2^2 + 2^K)$ and $2^{K-1} \times 2^{K-1}$ respectively and the latency is $K=3$ clock cycles.

In general for $K=3$, and $M>3$, the number of required 2-input adders and CS units can be given as $2^{K-1} \times 4(2^{K-1} - 1) + 2^{K-1} \times 2^K (M - K + 1)$ and $2^{K-1} \times 2^{K-1}(M - K + 1)$. The latency is $(2M - K) = K + 2(M - K)$ clock cycles. [31]

Figure 5-10 shows the number of Adders, Compare select Units and Latency with number of Viterbi steps M . It is cleared that the hardware complexity and the latency increased when the number of steps increased and affect the performance.

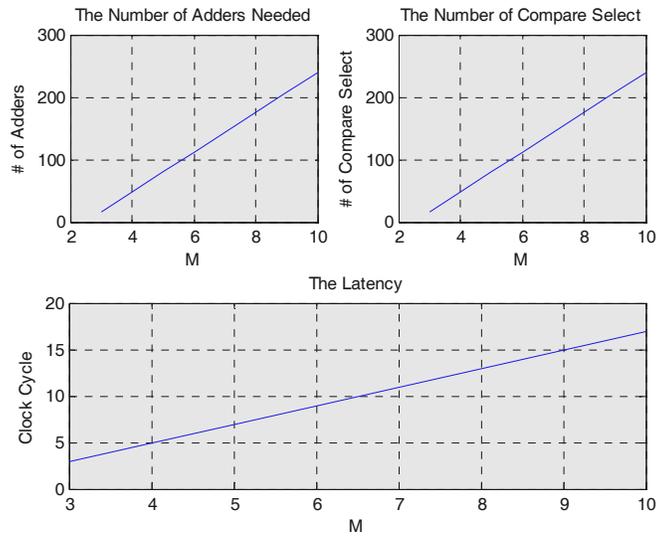


Figure 5-10: The Number of Adders, Compare Select Units and Latency with Number of Viterbi Steps M.

Figure 5-11 shows the system throughput for serial and parallel with M=2, 3 and 4.

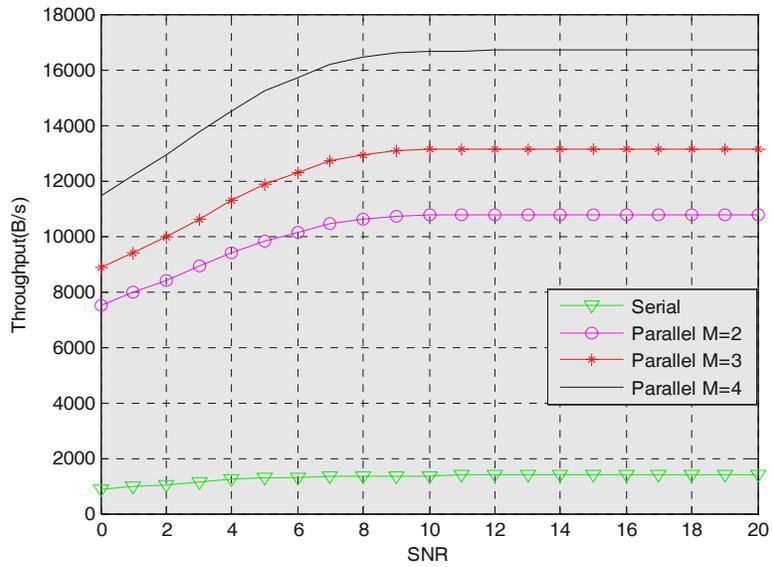


Figure 5-11: Throughput Enhancement for Parallel Viterbi Decoder.

The BER performance not affected when M increased as shown in Figure 5-12.

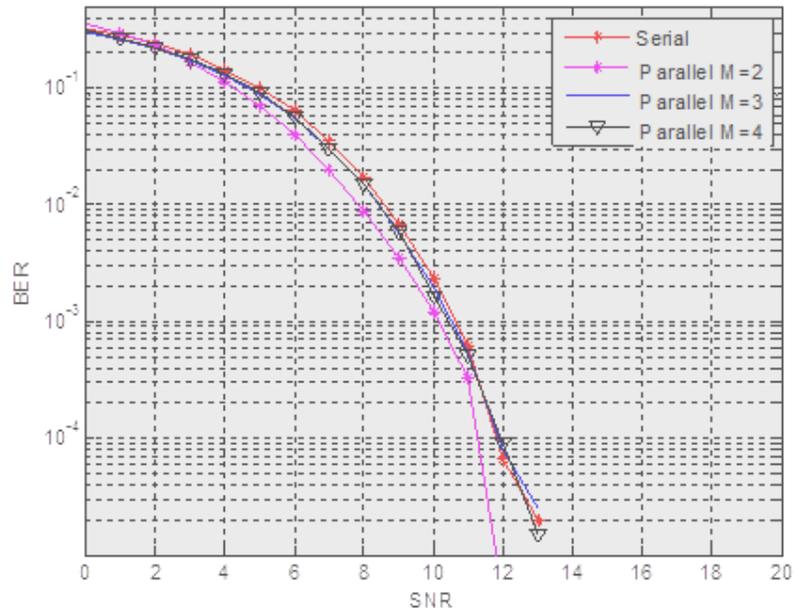


Figure 5-12: BER Performance for Serial, Parallel Viterbi ($M=2, 3, 4$) with QPSK Modulation Scheme.

The data are modulated after they are encoded; it has been found that there is no effect of Parallel Viterbi if different modulation schemes are used. But in term of system enhancement, it's clear from the Figure 5-13 that BPSK has the best BER performance, so it will be used in the next part which using DWT instead of FFT.

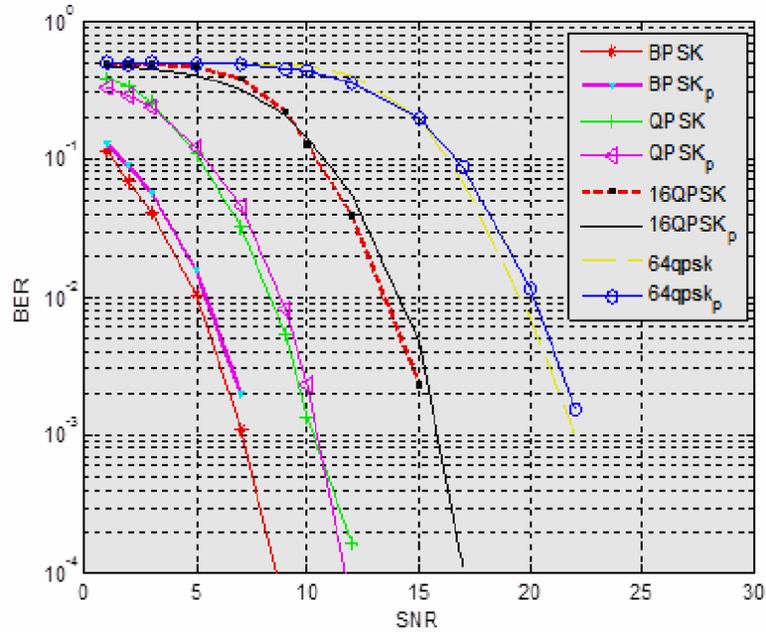


Figure 5-13: BER Performance for Serial and Parallel Viterbi with Different Modulation Techniques.

5.2.2 IDWT Instead of IFFT

The structure and algorithm of IFFT, FFT, the conventional method used in modulation /demodulation process in OFDM system, has matured today. Researchers have developed new transforms in an effort to replace this traditional structure, FFT-based structure. In this section, DWT and FFT will be implemented in MATLAB to acquire their BER performances, and the results will be compared and analyzed.

Figure 5-14 shows that DWT-OFDM performs much better than the FFT-OFDM over AWGN channel.

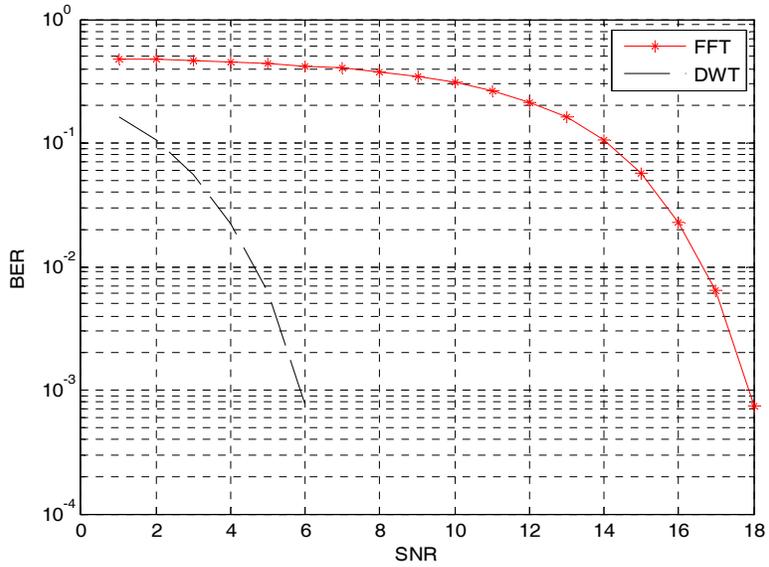


Figure 5-14: BER for OFDM Based FFT vs. OFDM Based DWT (Haar)

In Figure 5-14 the FFT length used is 256 and with no CP. It is well known that the insertion of the CP decrease the BER. Figure 5-15 show the effect of CP on the BER performance, but still the DWT has much better performance.

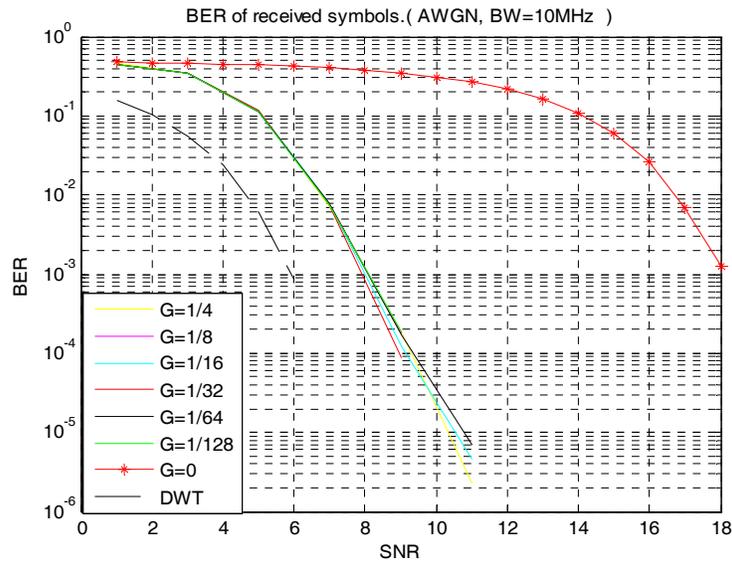


Figure 5-15: FFT with CP vs DWT (Haar).

Figure 5-16 shows the effect of FFT length and DWT length on the BER performance. It's clear from the figure that there is no effect on the DWT.

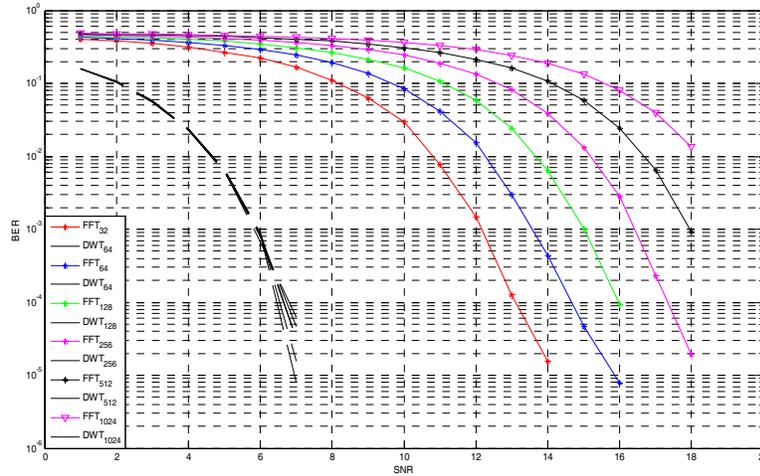


Figure 5-16: FFT and DWT (Haar) size Effect on BER.

The BER performance can be observed for different wavelet families; Biorthogonal, Reverse-biorthogonal are compared with FFT-OFDM.

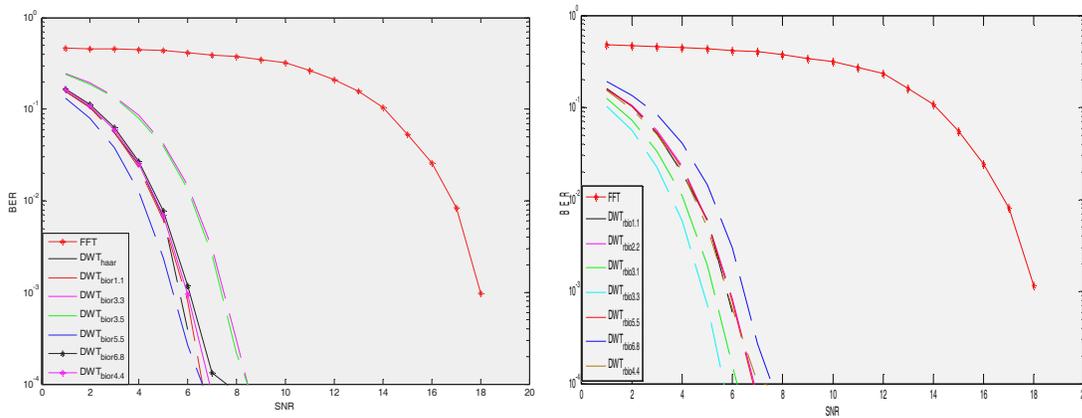


Figure 5-17: DWT Families BER Performance.

Figure 5-17 shows in the left that 'bior5.5' has the best performance in the the Biorthogonal family, the right side of the figure show the Reverse-biorthogonal family and the best is 'rbio3.3'. Comparing the best from the two families with the 'haar' and Daubechies 'db3' as shown in Figure 5-18.

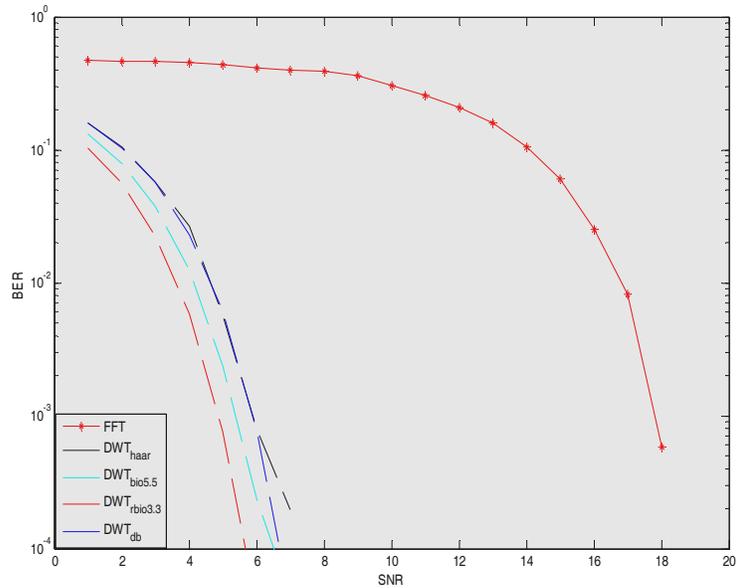


Figure 5-18: BER for OFDM Based FFT vs. OFDM Based DWT Biorthogonal Family.

Now, comparing the overall system BER performance; coded with Serial and Parallel Viterbi and uncoded (serial and parallel Viterbi) DWT and FFT based OFDM are simulated and the results are shown in Figure 5-19.

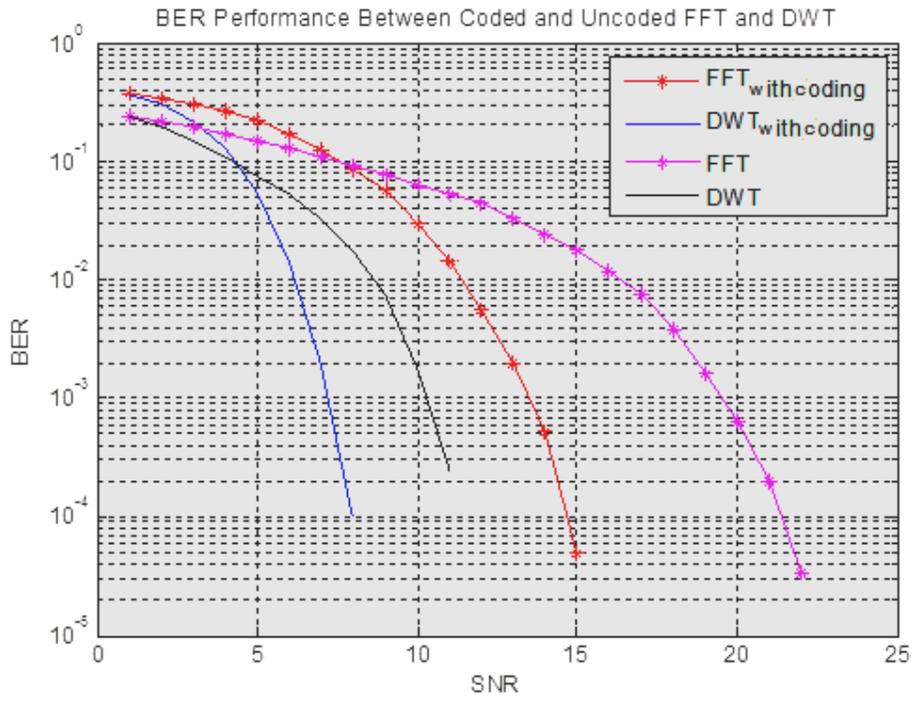


Figure 5-19: BER Performance Comparison between Coded and Uncoded FFT and DWT (Haar).

CHAPTER SIX

Chapter 6 Conclusions and Future Works

In this thesis look-ahead technique has been exploited to create the desired level of concurrency in the sequential design of Viterbi Algorithm. It's clear from the results that Viterbi decoding can be using the Look-Ahead transformation technique to give a vectorized output. It's quite interesting that the decrease in execution time is extremely high with only a single look-ahead step. This may lead the designers to optimize for this approach despite the expected large hardware requirements.

Although this algorithm can be implemented in a few lines of code, its storage space increases exponentially with each additional level of look-ahead. Also the computational complexity increases exponentially to compute the next set of matrices. The table for decoding storage increases exponentially with the increase of look-ahead step. Therefore, for large levels of look-ahead this algorithm would be hard to use.

The next logical step would be to go for a hardware implementation of the proposed algorithm. This will be more challenging as while designing the VA no consideration has been put on the hardware cost.

In BER performance simulation, orthogonal based OFDM systems (DFT-based OFDM, DWT-based OFDM, were compared over AWGN channel. DWT-based OFDM system performs much DFT-based OFDM systems on AWGN channel. It is worthy to introduce the effect of Rayleigh Fading channel and other types of channels.

DWT-based OFDM was only recently introduced to be applied in OFDM system but it holds even better potential to produce low hardware complexity because it doesn't need a CP. With its strong BER performance shown in this study, it is worth starting/continuing the

development of its hardware architecture and algorithm. Also it's interesting to design such wavelet which produces better BER performance.

REFERENCES

- [1] T. S. Rappaport, “Wireless Communications: Principles and Practice”. Prentice Hall, 2002.
- [2] W. C. Y. Lee, “Mobile Communications Engineering: Theory and Applications”, McGraw-Hill Professional; 2 edition, 1997
- [3] J. Korhonen, “Introduction to 3G mobile communications”. Artech House Publishers; 2nd edition, 2003
- [4] “Mobile communications (VIII)”, Advanced Communication Technology, ICACT 2009.
- [5] M. Gerami, “A Survey on WiMAX”. (IJCSIS) International Journal of Computer Science and Information Security, Vol. 8, 2010
- [6] M. Chakraborty and D. Bhattacharyya, “Overview of End-to-End WiMAX Network Architecture”.
- [7] Kas, M. Yargicoglu, B. Korpeoglu, I. Karasan, E. Dept. of Comput. Eng., Bilkent Univ., Bilkent and Turkey,”A Survey on Scheduling in IEEE 802.16 Mesh Modes “.Communications Surveys & Tutorials, IEEE, Vol. 12, 2010
- [8] “LTE vs WiMax: the battle continues [COMMS WiMax vs LTE]”. Engineering & Technology, Vol 5, 2010.
- [9] “LTE vs WiMAX”. Hot Topics Forum: LTE vs WiMAX and Next Generation Internet, Institution of Engineering and Technology, 2007.
- [10] WiMAX Forum, “WiMAX’s technology for LOS and NLOS environments”.
- [11] R. binti ,R. H. binti , M. E. Jiwa, N. bin Hamid, N. B. binti, A. S. Ramli and N. I. binti, “Investigating the distance effects on performance degradation of mobile

WiMAX technology using NCTUns tool". Science and Social Research (CSSR), 2010 International Conference ,2010.

- [12] A. B. Awoseyila and B. G. Evan, " LTE and WiMAX satellite systems: Improving FEC performance using split multicode transmission" . Advanced satellite multimedia systems conference (asma) and the 11th signal processing for space communications workshop (spsc), 2010.
- [13] T. Fuja, C. Heegard, M. Blaum, "Cross parity check convolutional codes". Information Theory, IEEE Transactions ,Vol 35, 1989.
- [14] H. A. Bustamante, I. Kang, C. Nguyen, and R. E. Peile, "Stanford Telecom VLSI of A Convolutional Decoder", in Proc., IEEE MILCOM, Vol. 1, 1989.
- [15] S. Prasad, "OFDM for Wireless Communication Systems". Artech House, 2004
- [16] C. Snow, L. Lampe and R. Schober, "Interference mitigation for coded MB-OFDM UWB". IEEE Trans, 2008.
- [17] Y. Li and G. L. Stuber, "Orthogonal Frequency Division Multiplexing for Wireless Communications". Springer, 2006
- [18] H. Liu G. Li, "OFDM-Based Broadband Wireless Networks Design and Optimization". Wiley, 2005
- [19] J. Proakis, "Digital Communications". McGraw-Hill, 2000
- [20] R. W. Hamming, "Error Detecting and Error Correcting Codes", The Bell System Technical Journal, J Soc, Indust. Appl. Math. Vol. 26, 1950.
- [21] I. S. Reed and G. Solomon, "Polynomial Codes over Certain Finite Fields", J .Soc, Indust. Appl. Math. Vol. 8, 1960.

- [22] Y.S. Wong and N.K., Noordin, "Implementation of convolutional encoder and Viterbi decoder using VHDL of convolutional encoder and Viterbi decoder using VHDL". IEEE Student Conference on, 2009
- [23] S. Nandula, Y.S. Rao, , S.P. Embanath, " High speed area efficient configurable Viterbi Decoder for WiFi and WiMAX systems", IEEE Press, 2007.
- [24] K. Barman, K. Sriram, V.U. Reddy, "Performance of a sub-optimal metric for soft-decision Viterbi with bit interleaving in OFDM based systems". Conference on Convergent Technologies for Asia-Pacific Region, 2003
- [25]G. D. Forney, Jr., "Convolutional codes I: Algebraic structure", IEEE Trans. Inform. Theory, 1970.
- [26] G. D. Forney, "The Viterbi algorithm", Proceedings IEEE, vol. 61, 1973.
- [27] A. J. Viterbi, "Convolutional Codes and Their Performance in Communication Systems", IEEE Trans. Commun., vol. COM-19, 1971.
- [28] McEliece and R.J. Wei Lin, "The trellis complexity of convolutional codes". Information Theory, IEEE Transactions n, vol 42, 1996.
- [29] K. Hong-Du and C. Y. Huang,"A Low-Complexity Viterbi Decoder for Space-Time Trellis Codes". IEEE Trans, 2010.
- [30] S. R. Baig, F. U. Rehman, and M. J. Mughal, "Performance Comparison of DFT, Discrete Wavelet Packet and Wavelet Transforms in an OFDM Transceiver for Multipath Fading Channel", 9th IEEE International Multitopic Conference, 2005.
- [31] C. Cheng, "Hardware Efficient Low-Latency Architecture for High Throughput Rate Viterbi Decoders", IEEE transactions, Vol. 55, 2008.
- [32]A.K.L. Ooi, M. Driberg and V. Jeoti,"DWT based FFT in Practical OFDM Systems". IEEE Region 10 Conference, 2006.

- [33] K. Abdullah, A.Z. Sadik and A.Z. Hussain ,”On the DWT- and WPT-OFDM versus FFT-OFDM”. GCC Conference & Exhibition, 2009.
- [34] R. Mirghani, and M. Ghavami, “Comparison between Wavelet-based and Fourier-based Multicarrier UWB Systems”, IET Communications, Vol. 2, 2008.
- [35] R. Dilmirghani, M. Ghavami, “Wavelet Vs Fourier Based UWB Systems”, 18th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 2007.
- [36] S.C.Kim, J.H.Ryu, and J.D.Cho, “Low Power, High-Rate Viterbi Decoder Employing the SST (Scarce State Transition) Scheme and Radix-4 Trellis”. IDEC Conference, 2000.

APPENDIX MATLAB CODES

```
%Sequential Viterbi
function ipHat_v= vt(y)
ref = [0 0 ; 0 1; 1 0 ; 1 1 ];
ipLUT = [ 0 0 0 0;...
         0 0 0 0;...
         1 1 0 0;...
         0 0 1 1 ];
pathMetric = zeros(4,1); % path metric
survivorPath_v = zeros(4,length(y)/2); % survivor path
p=[];
for ii = 1:length(y)/2
    r = y(2*ii-1:2*ii); % taking 2 coded bits
    % computing the Hamming distance between [00;01;10;11]
    rv = kron(ones(4,1),r);
    hammingDist = sum(xor(rv,ref),2);
    % branch metric and path metric for state 0
    bm1 = pathMetric(1,1) + hammingDist(1);
    bm2 = pathMetric(2,1) + hammingDist(4);
    [pathMetric_n(1,1) idx] = min([bm1,bm2]);
    survivorPath(1,1) = idx;
    % branch metric and path metric for state 1
    bm1 = pathMetric(3,1) + hammingDist(3);
    bm2 = pathMetric(4,1) + hammingDist(2);
    [pathMetric_n(2,1) idx] = min([bm1,bm2]);
    survivorPath(2,1) = idx+2 ;
    % branch metric and path metric for state 2
    bm1 = pathMetric(1,1) + hammingDist(4);
    bm2 = pathMetric(2,1) + hammingDist(1);
    [pathMetric_n(3,1) idx] = min([bm1,bm2]);
    survivorPath(3,1) = idx;
    % branch metric and path metric for state 3
    bm1 = pathMetric(3,1) + hammingDist(2);
    bm2 = pathMetric(4,1) + hammingDist(3);
    [pathMetric_n(4,1) idx] = min([bm1,bm2]);
    survivorPath(4,1) = idx+2;

    pathMetric = pathMetric_n;
    p=[p pathMetric];
    survivorPath_v(:,ii) = survivorPath;
end
% trace back unit
currState = 1;
ipHat_v = zeros(1,length(y)/2);
for jj = length(y)/2:-1:1
    prevState = survivorPath_v(currState, jj);
    ipHat_v(jj) = ipLUT(currState,prevState);
    currState = prevState;
end
for jj = length(y)/2:-1:1
    [xxx id]=min(p(:,jj));

    prevState = id;
    ipHat_v(jj) = ipLUT(currState,prevState);
    currState = prevState;
end
```

```

        ipHat_v=[ipHat_v(1,length(ipHat_v)) ipHat_v(1,1:length(ipHat_v)-1) ]
    ;
end

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

%Parallel Viterbi Decoder

```

function [ipHat_v]= vt_p(y)
oy=0;
oe=0;
% y=[1 1 1 0 0 1 0 0 1 0 1]
ref = [0 0 ; 0 1; 1 0 ; 1 1 ];
[r1,c1]=size(y);
if (mod(c1,4)~=0)
    y=[y zeros(1,4-mod(c1,4))];
end
[r1,c1]=size(y);
yx=reshape(y,4,c1/4);
ix=1;
pathMetric = zeros(4,1); % path metric
p=[];
for ii = 1:2:length(y)/2

    r = yx(:,ix); % taking 2 coded bits
    % computing the Hamming distance between [00;01;10;11]
    rv = kron(ones(4,1),r');
    hammingDist(:,1) = sum(xor(rv(:,1:2),ref),2);
    hammingDist(:,2) = sum(xor(rv(:,3:4),ref),2);
    % branch metric and path metric for state 0
    bm1 = hammingDist(1,1)+hammingDist(1,2);
    bm2 =hammingDist(4,1)+hammingDist(1,2);
    bm3 = hammingDist(3,1)+hammingDist(4,2);
    bm4 = hammingDist(2,1)+hammingDist(4,2);
    pathMetric_n(1,1) = min([bm1,bm2,bm3,bm4]);

    % branch metric and path metric for state 1
    bm1 =hammingDist(4,1)+hammingDist(3,2);
    bm2 = hammingDist(1,1)+hammingDist(3,2);
    bm3 = hammingDist(2,1)+hammingDist(2,2);
    bm4 =hammingDist(3,1)+hammingDist(2,2);
    pathMetric_n(2,1) = min([bm1,bm2,bm3,bm4]);

    % branch metric and path metric for state 2
    bm1 =hammingDist(1,1)+hammingDist(4,2);
    bm2 = hammingDist(4,1)+hammingDist(4,2);
    bm3 = hammingDist(3,1)+hammingDist(1,2);
    bm4 =hammingDist(2,1)+hammingDist(1,2);
    pathMetric_n(3,1) = min([bm1,bm2,bm3,bm4]);

    % branch metric and path metric for state 3
    bm1 = hammingDist(4,1)+hammingDist(2,2);
    bm2 = hammingDist(1,1)+hammingDist(2,2);
    bm3 =hammingDist(2,1)+hammingDist(3,2);
    bm4 = hammingDist(3,1)+hammingDist(3,2);
    pathMetric_n(4,1) = min([bm1,bm2,bm3,bm4]);
pathMetric = pathMetric_n;

```

```

p=[p pathMetric];
ix=ix+1;

end
tic
survivorPath_v =[0 0;1 0; 0 1; 1 1];
ipHat_v = [];
for jj = length(y)/4:-1:1
    [xxx id]=min(p(:,jj));
    ipHat_v = [survivorPath_v(id,:) ipHat_v];
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Convolutional Encoder ; input=1 bit -> output=2 bits with 3 memory elements, Code
%Rate=1/2
function [encoded_sequence]=convlenc(message)
enco_mem=[0 0 0]; %# of memory elements=3
encoded_sequence=zeros(1,(length(message))*2);

    enco_mem(1,3)=enco_mem(1,2);
    enco_mem(1,2)=enco_mem(1,1);
    enco_mem(1,1)=message(1,1);

    temp=xor(enco_mem(1),enco_mem(2));
    o1=xor(temp,enco_mem(3)); %generator polynomial=111
    o2=xor(enco_mem(1),enco_mem(3)); %generator polynomial=101
    encoded_sequence(1,1)=o1;
    encoded_sequence(1,2)=o2;

msg_len=length(message);
c=3;
for i=2:msg_len

    enco_mem(1,3)=enco_mem(1,2);
    enco_mem(1,2)=enco_mem(1,1);
    if(i<=msg_len)
        enco_mem(1,1)=message(1,i);
    else
        enco_mem(1,1)=0;
    end

    temp=xor(enco_mem(1),enco_mem(2));
    o1=xor(temp,enco_mem(3));
    o2=xor(enco_mem(1),enco_mem(3));

    encoded_sequence(1,c)=o1; %o1 generating polynomial(1,1,1)
    c=c+1;
    encoded_sequence(1,c)=o2; %o2 generating polynomial(1,0,1)
    c=c+1;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

%DWT-OFDM vs FFT-OFDM
clear
hold all
c=512; %
bits=1000; %
n=c*bits; % ×
data= 2*round(rand(1,n))-1; %
s = reshape(data,c,[]); %
for i=1:c
    bpsk_sig(i,:)=s(i,:); %
    fin(i,:)=ifft(bpsk_sig(i,:)); %
    fin2(i,:)=idwt(bpsk_sig(i,:),zeros(size(bpsk_sig(i,:))),'haar');
end
transmit=reshape(fin,1,[]);
transmit2=reshape(fin2,1,[]);
snr=[0:2:35];
for t=1:length(snr)
    signal_in_dB=10*log10(std( bpsk_sig(1,:))^2);
    noise_in_dB=signal_in_dB-snr(t);
    noise=wgn(size(transmit, 1),size(transmit, 2),noise_in_dB); %
Generate White Gaussian Noise with variance "noise_in_dB"
    noise2=wgn(size(transmit2, 1),size(transmit2, 2),noise_in_dB); %
Generate White Gaussian Noise with variance "noise_in_dB"
    rxdata=transmit+ noise;
    rxdata2=transmit2+ noise2;
    rec=reshape(rxdata,c,[]);
    rec2=reshape(rxdata2,c,[]);
    for i=1:c
        rxdataf(i,:)=fft(rec(i,:)); %
        uncarry(i,:)=rxdataf(i,:); %
        rxdataf2(i,:)=dwt(rec2(i,:),'haar');
        uncarry2(i,:)=rxdataf2(i,:);
    end
    rdata=sign(real(uncarry));
    num(t)=biterr(rdata+1,s+1)/n;
    rdata2=sign(real(uncarry2));
    num2(t)=biterr(rdata2(:,1:size(s,2))+1,s+1)/n;
end
semilogy(snr,num,'-*r')
semilogy(snr,num2,'-*m')
legend('FFT','DWT')
xlabel('SNR')
ylabel('BER')
grid

```