PERFORMANCE ANALYSIS OF TURBO CODED OFDM IN WIRELESS APPLICATION

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Technology

In

VLSI Design & Embedded System

By

B.BALAJI NAIK

Roll No: 20607011

Department of Electronics & Communication Engineering

National Institute of Technology

Rourkela

2008

PERFORMANCE ANALYSIS OF TURBO CODED OFDM IN WIRELESS APPLICATION

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Technology

In

VLSI Design & Embedded system

By

B.BALAJI NAIK

Roll No: 20607011

Under the guidance of

Prof.S.K.PATRA

Department of Electronics & Communication Engineering

National Institute of Technology

Rourkela

2008

National Institute of Technology

Rourkela

CERTIFICATE

This is to certify that the Thesis Report entitled ‘’**PERFORMANCE ANALYSIS OF TURBO CODED OFDM IN WIRELESS APPLICATION’’** submitted by Mr. **B.BALAJI NAIK (20607011)** in partial fulfillment of the requirements for the award of Master of Technology degree in Electronics and Communication Engineering with specialization in “VLSI Design & Embedded system” during session 2007-2008 at National Institute Of Technology, Rourkela (Deemed University) and is an authentic work by him under my supervision and guidance.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other university/institute for the award of any Degree or Diploma.

Date:

Prof. S.K.PATRA
Dept. of E.C.E
National Institute of Technology
Rourkela-769008

ACKNOWLEDGEMENTS

First of all, I would like to express my deep sense of respect and gratitude towards my advisor and guide **Prof. S.K.Patra**, who has been the guiding force behind this work. I am greatly indebted to him for his constant encouragement, invaluable advice and for propelling me further in every aspect of my academic life. His presence and optimism have provided an invaluable influence on my career and outlook for the future. I consider it my good fortune to have got an opportunity to work with such a wonderful person.

Next, I want to express my respects to **Prof. G.S.Rath, Prof.G.Panda, Prof. K.K. Mahapatra,** and **Dr. S. Meher** for teaching me and also helping me how to learn. They have been great sources of inspiration to me and I thank them from the bottom of my heart.

I would like to thank all faculty members and staff of the Department of Electronics and Communication Engineering, N.I.T. Rourkela for their generous help in various ways for the completion of this thesis.

I would like to thank all my friends and especially my classmates for all the thoughtful and mind stimulating discussions we had, which prompted us to think beyond the obvious. I've enjoyed their companionship so much during my stay at NIT, Rourkela.

I am especially indebted to my parents for their love, sacrifice, and support. They are my first teachers after I came to this world and have set great examples for me about how to live, study, and work.

B.Balaji Naik

CONTENTS

ABSTRACT

Orthogonal Frequency Division Multiplexing (OFDM) has become a popular modulation method in high speed wireless communications. By partitioning a wideband fading channel into flat narrowband channels, OFDM is able to mitigate the detrimental effects of multi path fading using a simple one- tap equalizer. There is a growing need to quickly transmit information wirelessly and accurately.

Engineers have already combine techniques such as OFDM suitable for high data rate transmission with forward error correction (FEC) methods over wireless channels. In this thesis, we enhance the system throughput of a working OFDM system by adding turbo coding. The smart use of coding and power allocation in OFDM will be useful to the desired performance at higher data rates.

Error control codes have become a vital part of modern digital wireless systems, enabling reliable transmission to be achieved over noisy channels. Over the past decade, turbo codes have been widely considered to be the most powerful error control code of practical importance. In the same time-scale, mixed voice/data networks have advanced further and the concept of global wireless networks and terrestrial links has emerged. Such networks present the challenge of optimizing error control codes for different channel types, and for the different qualities of service demanded by voice and data.

LIST OF FIGURES

ABBREVIATIONS

AWGN	additive White Gaussian Noise
ADSL	asymmetric digital subscriber line
AP	access point
BPSK	binary phase shift keying
CCK	complementary code keying
CSMA/CA	Carrier Sense Multiple Access/ Collision Avoidance
CDMA	code division multiple access
DSP	digital signal processors
DAB	digital audio broadcasting
DVB	digital video broadcasting
DFT	discrete Fourier transform
DSSS	direct sequence spread spectrum
EP	extension point
ETSI	European Telecommunications Standards Institute
FCC	Federal Communications Commission
FFT	fast Fourier transform
FDM	frequency division multiplexing
FEC	forward error correction
HDTV	high definition television
IEEE	Institute of Electrical and Electronics Engineers
IFFT	inverse Fourier transform
IDFT	inverse discrete Fourier transform
ISI	inter symbol interference

ICI	inter carrier interference
LAN	local area network
NTSC	National Television Systems Committee
OFDM	orthogonal frequency division multiplexing
PC	personal computer
QPSK	quadrature phase shift keying
QAM	quadrature amplitude modulation
SNR	signal to noise ratio
TDM	time division multiplexing
TDMA	time division multiple access
UHF	ultra high frequency
VLSI	very large scale integration
WLAN	wireless local area networks
TCOFDM	turbo coded orthogonal frequency division multiplexing

Chapter 1

INTRODUCTION

1.1 Introduction

The telecommunications' industry is in the midst of a veritable explosion in Wireless technologies. Once exclusively military, satellite and cellular technologies are now commercially driven by ever more demanding consumers, who are ready for seamless communication from their home to their car, to their office, or even for outdoor activities. With this increased demand comes a growing need to transmit information wirelessly, quickly, and accurately. To address this need, communications engineer have combined technologies suitable for high rate transmission with forward error correction techniques. The latter are particularly important as wireless communications channels are far more hostile as opposed to wire alternatives, and the need for mobility proves especially challenging for reliable communications. For the most part, Orthogonal Frequency Division Multiplexing (OFDM) is the standard being used throughout the world to achieve the high data rates necessary for data intensive applications that must now become routine. Orthogonal Frequency Division Multiplexing (OFDM) is a Multi-Carrier Modulation technique in which a single high rate data-stream is divided into multiple low rate data-streams and is modulated using sub-carriers which are orthogonal to each other. Some of the main advantages of OFDM are its multi-path delay spread tolerance and efficient spectral usage by allowing overlapping in the frequency domain. Also one other significant advantage is that the modulation and demodulation can be done using IFFT and FFT operations, which are computationally efficient.

In this thesis forward error correction is performed by using turbo codes. The combination of OFDM and turbo coding and recursive decoding allows these codes to achieve near Shannon's limit performance in the turbo cliff region.

1.2 Thesis outline

This thesis presents the simulation of turbo-coded OFDM system and analyzes the performance of this system under noisy environment. It is presented as follows:

Chapter 2 discusses definition of wireless LAN, wireless LAN technologies, and wireless LAN standards (IEEE 802.11, IEEE 802.11a, and IEEE 802.11b, IEEE 802.11g) in detail.

Chapter 3 introduces the theory behind OFDM as well as some of its advantages and Functionality issues. We discuss basic OFDM transceiver architecture, cyclic prefix, intersymbol interference, intercarrier interference and peak to average power ratios. We also

present a few results in both Additive White Gaussian Noise, and impulsive noise environments.

Chapter 4 focuses on turbo codes. We explore encoder and decoder architecture, and decoding algorithms (especially the maximum a posteriori algorithm). We elaborate on the performance theory of the codes and find out why they perform so well.

Chapter 5 consist simulated results of our work and a few suggestions are made on how to improve our system. Then, we present our results on the combination of turbo coding and OFDM. The core of our simulation results are found here

Chapter 2

WLAN TECHNOLOGIES & STANDARDS

2.1INTRODUCTION

"A Wireless Local Area Network is a data communications system which transmits and receives data over the air using radio technology." as the name suggests it makes use of wireless transmission medium. In earlier days they were not so popular. The reasons for these included high prices, low data rates and licensing requirements. As these problems have been addressed, the popularity of wireless LANs has grown rapidly.

Wireless LANs redefine the way we view LANs.connecivity no longer implies physical attachment. Users can remain connected to the network as they move around the building or campus. There is no need anymore to bury the network infrastructure in the ground or hide it behind the walls. With wireless networking, the network infrastructure can move and change at the speed of the organization.

Wireless LANs are used both in business and home environments, either as extensions to existing networks, or, in smaller environments, as alternatives to wired networks. They provide all the benefits and features of traditional LANs. Over the last seven years, WLANs have gained strong popularity in a number of vertical markets, including the health-care, retail, manufacturing, warehousing, and academic arenas. These industries have profited from the productivity gains of using hand-held terminals and notebook computers to transmit real-time information to centralized hosts for processing. Today WLANs are becoming more widely recognized as a general-purpose connectivity alternative for a broad range of business customers.

This chapter is organized as follows. Following this introduction, section 2.2 discusses the need for wireless LAN, its advantages over wired LAN. Section 2.3 discusses the different wireless LAN technologies. Section 2.4 discusses the various WLAN standards (IEEE 802.11, IEEE802.11a, IEEE 802.11b, IEEE 802.11g) etc. finally section 2.5 concludes the chapter.

2.2 WHY WIRELESS?

The widespread reliance on networking in business and the meteoric growth of the Internet and online services are strong testimonies to the benefits of shared data and shared resources [11]. With wireless LANs, users can access shared information without looking for a place to plug in, and network managers can set up or augment networks without installing or moving wires. Wireless LANs offer the following productivity, convenience, and cost advantages over traditional wired networks:

- **Mobility:** Wireless LAN systems can provide LAN users with access to real-time information anywhere in their organization. This mobility supports productivity and service opportunities not possible with wired networks.
- **Installation Speed and Simplicity:** Installing a wireless LAN system can be fast and easy and can eliminate the need to pull cable through walls and ceilings.
- **Installation Flexibility:** Wireless technology allows the network to go where wire cannot go.
- **Reduced Cost-of-Ownership:** While the initial investment required for wireless LAN hardware can be higher than the cost of wired LAN hardware, overall installation expenses and life-cycle costs can be significantly lower. Long-term cost benefits are greatest in dynamic environments requiring frequent moves and changes.
- **Scalability:** Wireless LAN systems can be configured in a variety of topologies to meet the needs of specific applications and installations. Configurations are easily changed and range from peer-to-peer networks suitable for a small number of users to full infrastructure networks of thousands of users that enable roaming over a broad area.

2.3 WIRELESS LAN TECHNOLOGIES

The technologies available for use in WLANs include infrared, UHF (narrowband) radios, and spread spectrum radios. Two spread spectrum techniques are currently prevalent: frequency hopping and direct sequence. In the United States, the radio bandwidth used for spread spectrum communications falls in three bands (900 MHz, 2.4 GHz, and 5.7 GHz), which the Federal Communications Commission (FCC) approved for local area commercial communications in the late 1980s. In Europe, ETSI, the European Telecommunications Standards Institute, introduced regulations for 2.4 GHz in 1994, and Hiperlan is a family of standards in the 5.15-5.7 GHz and 19.3 GHz frequency bands [12].

2.3.1 Infrared (IR)

Infrared is an invisible band of radiation that exists at the lower end of the visible electromagnetic spectrum. This type of transmission is most effective when a clear line-of-sight exists between the transmitter and the receiver. Two types of infrared WLAN solutions are available: diffused-beam and direct-beam (or line-of-sight). Currently, direct-beam WLANs offer a

faster data rate than diffused-beam networks, but is more directional since diffused-beam technology uses reflected rays to transmit/receive a data signal, it achieves lower data rates in the 1-2 Mbps range.

Infrared optical signals are often used in remote control device applications. Users who can benefit from infrared include professionals who continuously set up temporary offices, such as auditors, salespeople, consultants, and managers who visit customers or branch offices. These users connect to the local wired network via an infrared device for retrieving information or using fax and print functions on a server.

A group of users may also set up a peer-to peer infrared network while on location to share printer, fax, or other server facilities within their own LAN environment. The education and medical industries commonly use this configuration to easily move networks. Infrared is a short range technology. When used indoors, it can be limited by solid objects such as doors, walls, merchandise, or racking. In addition, the lighting environment can affect signal quality. For example, loss of communications may occur because of the large amount of sunlight or background light in an environment. Fluorescent lights also may contain large amounts of infrared. This problem may be solved by using high signal power and an optical bandwidth filter, which lessens the infrared signals coming from outside sources. In an outdoor environment, snow, ice, and fog may affect the operation of an infrared based system. Because of its many limitations, infrared is not a very popular technology for WLANs.

- **Advantages:**
- No government regulation controlling use.
- Immunity To electro magnetic (EM) and RF interference.
- **Disadvantages:**
- Generally a short range technology(30-50 ft under ideal conditions)
- Signals cannot penetrate solid objects.
- Signal affected by light, snow, ice, fog.
- Dirt can interfere with infrared.

2.3.2 Narrowband technology

A narrowband radio system transmits and receives user information on a specific radio frequency. Narrowband radio keeps the radio signal frequency as narrow as possible just to pass the

information. Undesirable crosstalk between communications channels is avoided by carefully coordinating different users on different channel frequencies.

A private telephone line is much like a radio frequency. When each home in a neighborhood has its own private telephone line, people in one home cannot listen to calls made to other homes. In a radio system, privacy and noninterference are accomplished by the use of separate radio frequencies. The radio receiver filters out all radio signals except the ones on its designated frequency.

- **Advantages:**
- Longest range.
- Low cost solution for large sites with low to medium data throughput requirements.
- **Disadvantages:**
- Large radio and antennas increase wireless client size.
- RF site license required for protected bands.
- No multivendor interoperability.
- Low throughput and interference potential.

2.3.3 Spread Spectrum Technology

Most wireless LAN systems use spread-spectrum technology, a wideband radio frequency technique developed by the military for use in reliable, secure, mission-critical communications systems. Spread-spectrum is designed to trade off bandwidth efficiency for reliability, integrity, and security. In other words, more bandwidth is consumed than in the case of narrowband transmission, but the tradeoff produces a signal that is, in effect, louder and thus easier to detect, provided that the receiver knows the parameters of the spread-spectrum signal being broadcast. If a receiver is not tuned to the right frequency, a spread-spectrum signal looks like background noise. There are two types of spread spectrum radio: frequency hopping and direct sequence.

- **Frequency-Hopping Spread Spectrum Technology**

Frequency-hopping spread-spectrum (FHSS) uses a narrowband carrier that changes frequency in a pattern known to both transmitter and receiver. Properly synchronized, the net effect is to maintain a single logical channel. To an unintended receiver, FHSS appears to be short duration impulse noise.

- **Direct-Sequence Spread Spectrum Technology**

Direct-sequence spread-spectrum (DSSS) generates a redundant bit pattern for each bit to be transmitted. This bit pattern is called a chip (or chipping code). The longer the chip, the greater the probability that the original data can be recovered (and, of course, the more bandwidth required). Even if one or more bits in the chip are damaged during transmission, statistical techniques embedded in the radio can recover the original data without the need for retransmission. To an unintended receiver, DSSS appears as low-power wideband noise and is rejected (ignored) by most narrowband receivers.

2.3.4 Orthogonal frequency division multiplexing

Orthogonal frequency division multiplexing, also called multi carrier modulation uses multiple carrier signals at different frequencies, sending some of bits on each channel. This is similar to FDM.How ever, in the case of OFDM, all sub channels are dedicated to a single data source.

In the OFDM, Suppose we have a data stream operating at R bps and an available bandwidth of $N\Delta f$, centerd at f_0.theentire bandwidth could be used to send data stream, in which case each bit duration would be 1/R.The alternative is to split the data stream into N substreams, using a serial to parallel converter. Each substream has a data rate of R/Nbps and is transmitted on a separate subcarrier, with spacing between adjacent subcarriers of Δf.now the bit duration is N/R.

OFDM has several advantages. First, frequency selective fading only affects some sub channels and not the whole signal. If the data stream is protected by a forward error correcting code, this type of fading is easily handled. More important, OFDM overcome inter symbol interference (ISI) in a multipath environment's has greater impact at higher bit rates, because the distance between bits or symbols is smaller. With OFDM, the data rate is reduced by a factor of N, which increases the symbol time by a factor of N. thus if the symbol period is T_s for the source stream, the period for the OFDM signals is NT_s. This dramatically reduces the effect of ISI.as a design criterion, N is chosen so that NT_s is significantly greater than the root mean square delay spread of the channel.

2.4 STANDARDS

Nowhere in the modern computing field is the proliferation of acronyms and numerical designators more prevalent than in wireless networking. Here is the short version of what you need to know to bring some order to the chaos.

2.4.1 802. What?

The IEEE (Institute of Electrical and Electronics Engineers) is the body responsible for setting standards for computing devices. They have established a committee to set standards for Local Area and Metropolitan Area Networking named the "802 LMSC" (LAN MAN Standards Committee). Within this committee there are workgroups tasked with specific responsibilities, and given a numeric designation such as "11". In this case the 802.11 workgroup is tasked with developing the standards for wireless networking [13].

Within this 802.11 workgroup, there are task groups with even more specific tasks, and these groups are designated with an alphabetic character such as "a", or "b", or "g". There is no apparent logic to the ordering of these characters and none should be inferred. The specific groups and tasks concerning wireless networking hardware standards are outlined below.

Standard	Release date	Op.frequency band	Max.data rate
IEEE 802.11	1997	2.4GHz	2Mbps
IEEE 802.11a	1999	5GHz	54Mbps
IEEE 802.11b	1999	2.4GHz	11Mbps
IEEE 802.11g	2003	2.4GHz	54Mbps
IEEE 802.11n	2007(projected)	2.4GHz or 5GHz	540Mbps

Table 2.1 IEEE 802.11 standards

2.4.2 IEEE 802.11

The original version of the standard IEEE 802.11 released in 1997 specifies two raw data rates of 1 and 2 megabits per second (Mbit/s) to be transmitted via infrared (IR) signals or by either Frequency hopping or Direct-sequence spread spectrum in the Industrial Scientific Medical frequency band at 2.4 GHz. IR remains a part of the standard but has no actual implementations. The original standard also defines Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) as the medium access method. A significant percentage of the available raw channel capacity is sacrificed (via the CSMA/CA mechanisms) in order to improve the reliability of data transmissions under diverse and adverse environmental conditions.

2.4.3 IEEE 802.11a

The 802.11a amendment to the original standard was ratified in 1999. The 802.11a standard uses the same core protocol as the original standard, operates in 5 GHz band, and uses a 52-subcarrier orthogonal frequency-division multiplexing (OFDM) with a maximum raw data rate of 54 Mb/s, which yields realistic net achievable throughput in the mid-20 Mb/s. The data rate is reduced to 48, 36, 24, 18, 12, 9 then 6 Mb/s if required.

802.11a is not interoperable with 802.11b as they operate on separate bands, except if using equipment that has a dual band capability. Nearly all enterprise class Access Points has dual band capability. Since the 2.4 GHz band is heavily used, using the 5 GHz band gives 802.11a a significant advantage. However, this high carrier frequency also brings a slight disadvantage. The effective overall range of 802.11a is slightly less then 802.11b/g, it also means that 802.11a cannot penetrate as far as 802.11b since it is absorbed more readily when penetrating multiple walls. On the other hand, OFDM has fundamental propagation advantages when in a high multipath environment such as an indoor office. And the higher frequencies enable the building of smaller antennae with higher RF system gain which counteract the disadvantage of a higher band of operation. The increased number of useable channels (4 to 8 times as many in FCC countries) and the near absence of other interfering systems (microwave ovens, cordless phones, bluetooth products) makes the 5 GHz band the preferred band for professionals and businesses who require more capacity and reliability and are willing to pay a small premium for it.

2.4.4 IEEE 802.11b

The 802.11b amendment to the original standard was ratified in 1999. 802.11b has a maximum raw data rate of 11 Mb/s and uses the same CSMA/CA media access method defined in the original standard.802.11b products appeared on the market in early 2000, since 802.11b is a direct extension of the DSSS (Direct-sequence spread spectrum) modulation technique defined in the original standard. Technically, the 802.11b standard uses Complementary code keying (CCK) as its modulation technique. The dramatic increase in throughput of 802.11b (compared to the original standard) along with simultaneous substantial price reductions led to the rapid acceptance of 802.11b as the definitive wireless LAN technology.

2.4.5 IEEE 802.11g

In June 2003, a third modulation standard was ratified: 802.11g.This flavor works in the 2.4 GHz band (like 802.11b) but operates at a maximum raw data rate of 54 Mb/s, or about 19 Mb/s net throughput (like 802.11a except with some additional legacy overhead). 802.11g hardware is backwards compatible with 802.11b hardware. Details of making b and g work well together occupied much of the lingering technical process. In an 11g network, however, the presence of an 802.11b participant does significantly reduce the speed of the overall 802.11g network.

The modulation scheme used in 802.11g is orthogonal frequency-division multiplexing (OFDM) for the data rates of 6, 9, 12, 18, 24, 36, 48, and 54 Mb/s, and reverts to CCK (like the 802.11b standard) for 5.5 and 11 Mb/s and DBPSK/DQPSK+DSSS for 1 and 2 Mb/s. Even though 802.11g operates in the same frequency band as 802.11b, it can achieve higher data rates because of its similarities to 802.11a. The maximum range of 802.11g devices is slightly greater than that of 802.11b devices, but the range in which a client can achieve the full 54 Mb/s data rate is much shorter than that of which a 802.11b client can reach 11 Mb/s.

Chapter3

OFDM

3.1 INTRODUCTION

The principle of orthogonal frequency division multiplexing (OFDM) modulation has been in existence for several decades. However, in recent years these techniques have quickly moved out of textbooks and research laboratories and into practice in modern communications systems. The techniques are employed in data delivery systems over the phone line, digital radio and television, and wireless networking systems [14]. What is OFDM? And why has it recently become so popular?

This chapter is organized as follows. Following this introduction, section 3.2, 3.3 gives brief details about single carrier modulation, FDM modulation systems. Section 3.4 discusses definition of orthogonality, and principle of OFDM.section 3.5 discusses the how FFT maintains orthogonality.section 3.6 discusses the generation and reception of OFDM in detail. Section 3.7 addresses about the guard period used in OFDM systems. Section 3.8 presents the advantages, disadvantages and applications of OFDM. Finally section 3.9 concludes the chapter.

3.2 THE SINGLE CARRIER MODULATION SYSTEM

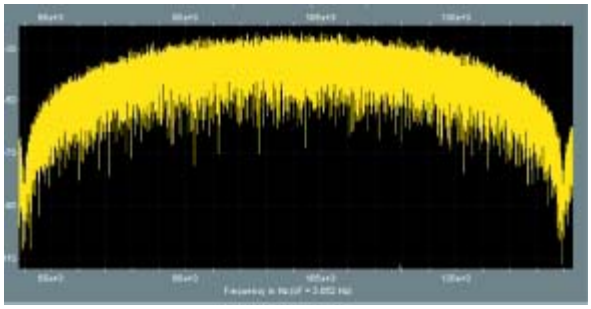


Fig.3.1 Single carrier spectrum

A typical single-carrier modulation spectrum is shown in Figure 3.1. A single carrier system modulates information onto one carrier using frequency, phase, or amplitude adjustment of the carrier. For digital signals, the information is in the form of bits, or collections of bits called symbols, that are modulated onto the carrier. As higher bandwidths (data rates) are used, the duration of one bit or symbol of information becomes smaller. The system becomes more susceptible to loss of information from impulse noise, signal reflections and other impairments. These impairments can impede the ability to recover the information sent. In addition, as the bandwidth used by a single carrier system increases, the susceptibility to interference from other

continuous signal sources becomes greater. This type of interference is commonly labeled as carrier wave (CW) or frequency interference.

3.3 FREQUENCY DIVISION MULTIPLEXING MODULATION SYSTEM

A typical Frequency division multiplexing signal spectrum is shown in figure 3.2.FDM extends the concept of single carrier modulation by using multiple sub carriers within the same single channel. The total data rate to be sent in the channel is divided between the various sub carriers. The data do not have to be divided evenly nor do they have to originate from the same information source. Advantages include using separate modulation demodulation customized to a particular type of data, or sending out banks of dissimilar data that can be best sent using multiple, and possibly different, modulation schemes.

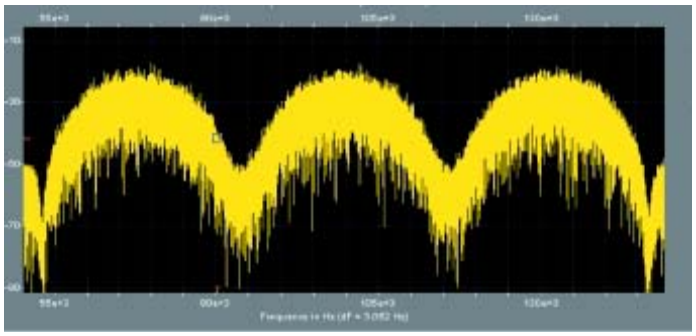


Fig 3.2 FDM signal spectrum

Current national television systems committee (NTSC) television and FM stereo multiplex are good examples of FDM. FDM offers an advantage over single-carrier modulation in terms of narrowband frequency interference since this interference will only affect one of the frequency sub bands. The other sub carriers will not be affected by the interference. Since each sub carrier has a lower information rate, the data symbol periods in a digital system will be longer, adding some additional immunity to impulse noise and reflections. FDM systems usually require a guard band between modulated sub carriers to prevent the spectrum of one sub carrier from interfering with another. These guard bands lower the system's effective information rate when compared to a single carrier system with similar modulation.

3.4 ORTHOGONALITY AND OFDM

If the FDM system above had been able to use a set of sub carriers that were orthogonal to each other, a higher level of spectral efficiency could have been achieved. The guard bands that were necessary to allow individual demodulation of sub carriers in an FDM system would no longer be necessary. The use of orthogonal sub carriers would allow the sub carriers' spectra to overlap, thus increasing the spectral efficiency. As long as orthogonality is maintained, it is still possible to recover the individual sub carriers' signals despite their overlapping spectrums. If the dot product of two deterministic signals is equal to zero, these signals are said to be orthogonal to each other. Orthogonality can also be viewed from the standpoint of stochastic processes. If two random processes are uncorrelated, then they are orthogonal. Given the random nature of signals in a communications system, this probabilistic view of orthogonality provides an intuitive understanding of the implications of orthogonality in OFDM.

OFDM is implemented in practice using the discrete Fourier transform (DFT). Recall from signals and systems theory that the sinusoids of the DFT form an orthogonal basis set, and a signal in the vector space of the DFT can be represented as a linear combination of the orthogonal sinusoids. One view of the DFT is that the transform essentially correlates its input signal with each of the sinusoidal basis functions. If the input signal has some energy at a certain frequency, there will be a peak in the correlation of the input signal and the basis sinusoid that is at that corresponding frequency. This transform is used at the OFDM transmitter to map an input signal onto a set of orthogonal sub carriers, i.e., the orthogonal basis functions of the DFT. Similarly, the transform is used again at the OFDM receiver to process the received sub carriers. The signals from the sub carriers are then combined to form an estimate of the source signal from the transmitter. The orthogonal and uncorrelated nature of the sub carriers is exploited in OFDM with powerful results. Since the basis functions of the DFT are uncorrelated, the correlation performed in the DFT for a given sub carrier only sees energy for that corresponding sub carrier. The energy from other sub carriers does not contribute because it is uncorrelated. This separation of signal energy is the reason that the OFDM sub carriers' spectrums can overlap without causing interference.

3.5 MATHEMATICAL ANALYSIS:

With an overview of the OFDM system, it is valuable to discuss the mathematical definition of the modulation system. It is important to understand that the carriers generated by the IFFT chip are mutually orthogonal. This is true from the very basic definition of an IFFT signal. This will allow understanding how the signal is generated and how receiver must operate. Mathematically, each carrier can be described as a complex wave:

$$S_C(t) = A_C(t)e^{j(\omega_c(t)+\Phi c(t))} \tag{3.1}$$

The real signal is the real part of sc (t). Ac (t) and φc (t), the amplitude and phase of the carrier, can vary on a symbol by symbol basis. The values of the parameters are constant over the symbol duration period t. OFDM consists of many carriers. Thus the complex signal $S_s(t)$ is represented by:

$$s_s(t) = \frac{1}{N}\sum_{n=0}^{N-1} A_N(t)e^{j[\omega_n t+\phi_n(t)]} \tag{3.2}$$

Where

$$\omega_n = \omega_o + n\Delta\omega$$

This is of course a continuous signal. If we consider the waveforms of each component of the signal over one symbol period, then the variables Ac (t) and φc (t) take on fixed values, which depend on the frequency of that particular carrier, and so can be rewritten:

$$\phi_n(t) = \phi_n$$
$$A_n(t) = A_n$$

If the signal is sampled using a sampling frequency of 1/T, then the resulting signal is represented by:

$$s_s(kT) = \frac{1}{N}\sum_{n=0}^{N-1} A_n e^{[j(\omega_0+n\Delta\omega)kT+\phi_n]} \tag{3.3}$$

At this point, we have restricted the time over which we analyze the signal to N samples. It is convenient to sample over the period of one data symbol. Thus we have a relationship: t=NT If we now simplify equation 3.3, without a loss of generality by letting ω0=0, then the signal becomes:

$$s_s(kT) = \frac{1}{N}\sum_{N=0}^{N-1} A_n e^{j\phi_n} e^{j(n\Delta\omega)kT} \tag{3.4}$$

Now equation 3.4 can be compared with the general form of the inverse Fourier transform:

$$g(kT) = \frac{1}{N}\sum_{n=0}^{N-1} G\left(\frac{n}{NT}\right) e^{\frac{2\pi}{N}kn} \tag{3.5}$$

In Equation 3.4 the function $A_n e^{j\phi_n}$ is no more than a definition of the signal in the sampled frequency domain, and s (kT) is the time domain representation. Eqns.4 and 5 are equivalent if:

$$\Delta f = \frac{\Delta\omega}{2\pi} = \frac{1}{NT} = \frac{1}{\tau}$$

This is the same condition that was required for orthogonality Thus, one consequence of maintaining orthogonality is that the OFDM signal can be defined by using Fourier transform procedures.

3.6 OFDM GENERATION AND RECEPTION

OFDM signals are typically generated digitally due to the difficulty in creating large banks of phase locks oscillators and receivers in the analog domain. Fig 3.3 shows the block diagram of a typical OFDM transceiver [15]. The transmitter section converts digital data to be transmitted, into a mapping of subcarrier amplitude and phase. It then transforms this spectral representation of the data into the time domain using an Inverse Discrete Fourier Transform (IDFT). The Inverse Fast Fourier Transform (IFFT) performs the same operations as an IDFT, except that it is much more computationally efficiency, and so is used in all practical systems. In order to transmit the OFDM signal the calculated time domain signal is then mixed up to the required frequency.

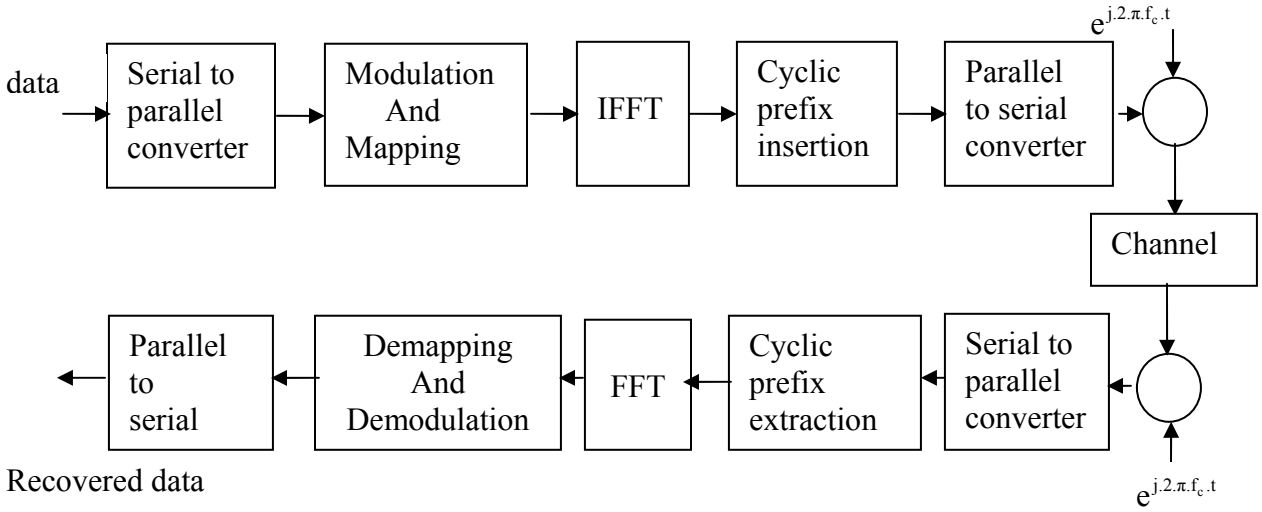


Fig 3.3 Block diagram of a basic OFDM transceiver.

The receiver performs the reverse operation of the transmitter, mixing the RF signal to base band for processing, then using a Fast Fourier Transform (FFT) to analyze the signal in the frequency domain [16]. The amplitude and phase of the sub carriers is then picked out and converted back to digital data. The IFFT and the FFT are complementary function and the most appropriate term depends on whether the signal is being received or generated. In cases where the signal is independent of this distinction then the term FFT and IFFT is used interchangeably.

3.6.1 Serial to parallel conversion

Data to be transmitted is typically in the form of a serial data stream. In OFDM, each symbol typically transmits 40 - 4000 bits, and so a serial to parallel conversion stage is needed to convert the input serial bit stream to the data to be transmitted in each OFDM symbol. The data allocated to each symbol depends on the modulation Scheme used and the number of sub carriers. For example, for a sub carrier modulation of 16 QAM each sub carrier carries 4 bits of data, and so for a transmission using 100 sub carriers the number of bits per symbol would be 400.

At the receiver the reverse process takes place, with the data from the sub carriers being converted back to the original serial data stream. When an OFDM transmission occurs in a multipath radio environment, frequency selective fading can result in groups of sub carriers being heavily attenuated, which in turn can result in bit errors. These nulls in the frequency response of the channel can cause the information sent in neighbouring carriers to be destroyed, resulting in a clustering of the bit errors in each symbol. Most Forward Error Correction (FEC)

schemes tend to work more effectively if the errors are spread evenly, rather than in large clusters, and so to improve the performance most systems employ data scrambling as part of the serial to parallel conversion stage. This is implemented by randomizing the sub carrier allocation of each sequential data bit. At the receiver the reverse scrambling is used to decode the signal. This restores the original sequencing of the data bits, but spreads clusters of bit errors so that they are approximately uniformly distributed in time.

This randomization of the location of the bit errors improves the performance of the FEC and the system as a whole.

3.6.2 Subcarrier modulation

- **Modulation: An Introduction**

One way to communicate a message signal whose frequency spectrum does not fall within that fixed frequency range, or one that is otherwise unsuitable for the channel, is to change a transmittable signal according to the information in the message signal. This alteration is called modulation, and it is the modulated signal that is transmitted. The receiver then recovers the original signal through a process called demodulation.

Modulation is a process by which a carrier signal is altered according to information in a message signal. The carrier frequency, denoted Fc, is the frequency of the carrier signal. The sampling rate, Fs, is the rate at which the message signal is sampled during the simulation. The frequency of the carrier signal is usually much greater than the highest frequency of the input message signal. The Nyquist sampling theorem requires that the simulation sampling rate Fs be greater than two times the sum of the carrier frequency and the highest frequency of the modulated signal, in order for the demodulator to recover the message correctly.

- **Baseband versus Pass band Simulation**

For a given modulation technique, two ways to simulate modulation techniques are called baseband and pass band. Baseband simulation requires less computation. In this thesis, baseband simulation will be used.

❖ **Digital Modulation Techniques**

a) Amplitude Shift Key (ASK) Modulation

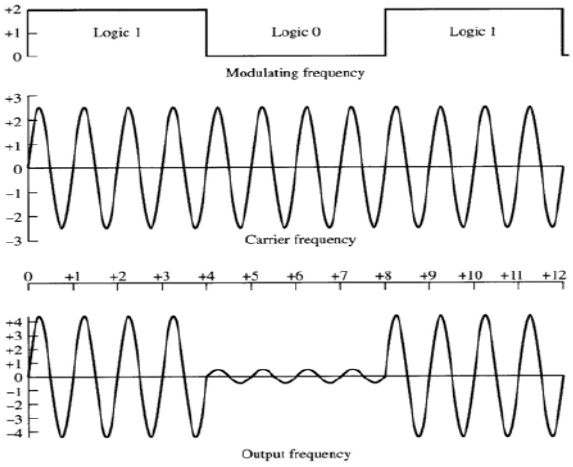


Fig 3.4 ASK modulation

In this method the amplitude of the carrier assumes one of the two amplitudes dependent on the logic states of the input bit stream. A typical output waveform of an ASK modulation is shown in Fig3.4.

b) Frequency Shift Key (FSK) Modulation

In this method the frequency of the carrier is changed to two different frequencies depending on the logic state of the input bit stream. The typical output waveform of an FSK is shown in Fig 3.5. Notice that logic high causes the centre frequency to increase to a maximum and a logic low causes the centre frequency to decrease to a minimum.

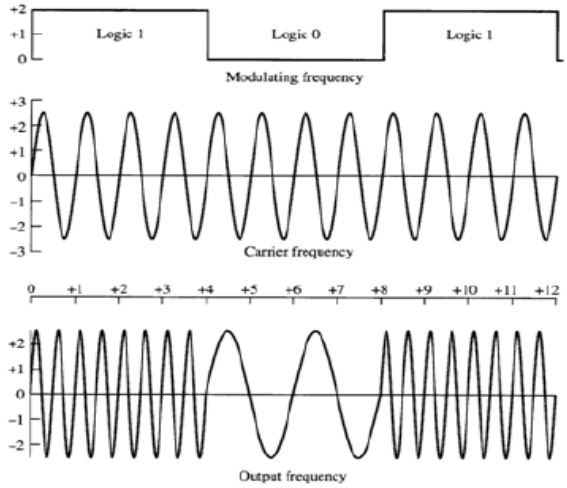


Fig. 3.5 FSK Modulation

c) Phase Shift Key (PSK) Modulation

With this method the phase of the carrier changes between different phases determined by the logic states of the input bit stream. There are several different types of Phase Shift Key (PSK) modulators. These are:

- Two-phase (2 PSK)
- Four-phase (4 PSK)
- Eight-phase (8 PSK)
- Sixteen-phase (16 PSK) etc.

d) Quadrature Amplitude Modulation (QAM)

QAM is a method for sending two separate (and uniquely different) channels of information. The carrier is shifted to create two carriers namely the sine and cosine versions. The outputs of both modulators are algebraically summed and the result of which is a single signal to be transmitted, containing the In-phase (I) and Quadrature (Q) information. The set of possible combinations of amplitudes is a pattern of dots known as a QAM constellation.

Once each subcarrier has been allocated bits for transmission, they are mapped using a modulation scheme to a subcarrier amplitude and phase, which is represented by a complex In-phase and Quadrature-phase (IQ) vector. Fig 3.6 shows an example of subcarrier modulation mapping. This example shows 16-QAM, which maps 4 bits for each symbol. Each combination of the 4 bits of data corresponds to a unique IQvector, shown as a dot on the figure. A large number of modulation schemes are available allowing the number of bits transmitted per carrier per symbol to be varied [17].

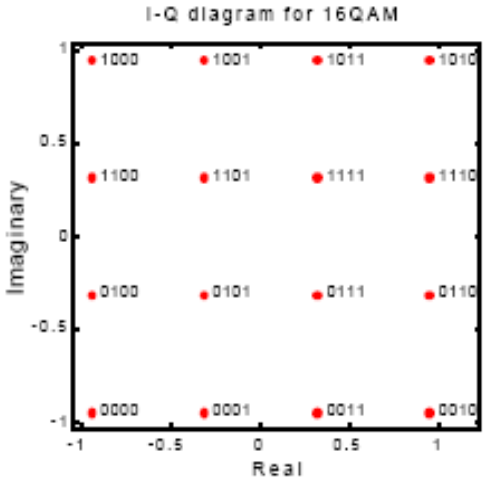


Fig 3.6 IQ modulation constellation, 16-QAM

Subcarrier modulation can be implemented using a lookup table, making it very efficient to implement. In the receiver, mapping the received IQ vector back to the data word performs sub carrier demodulation.

3.6.3 Frequency to time domain conversion

After the subcarrier modulation stage each of the data sub carriers is set to amplitude and phase based on the data being sent and the modulation scheme. All unused sub carriers are set to zero. This sets up the OFDM signal in the frequency domain. An IFFT is then used to convert this signal to the time domain, allowing it to be transmitted. Fig 3.7 shows the IFFT section of the OFDM transmitter. In the frequency domain, before applying the IFFT, each of the discrete samples of the IFFT corresponds to an individual sub carrier. Most of the sub carriers are modulated with data. The outer sub carriers are unmodulated and set to zero amplitude. These zero sub carriers provide a frequency guard band before the nyquist frequency and effectively act as an interpolation of the signal and allows for a realistic roll off in the analog anti-aliasing reconstruction filters.

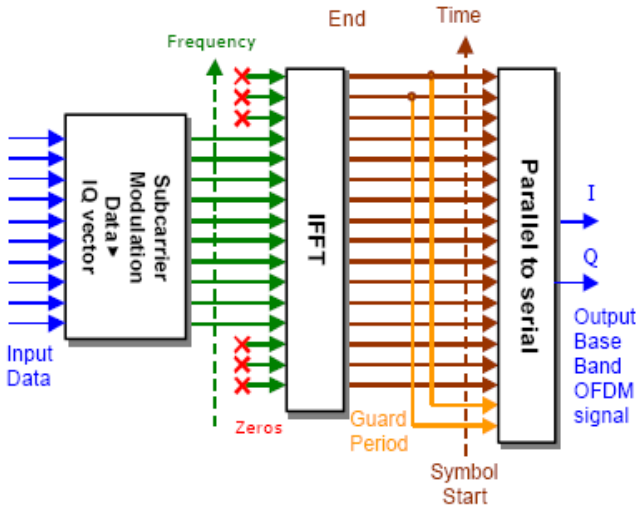


Fig 3.7 OFDM generation, IFFT stage

3.6.4 RF modulation

The output of the OFDM modulator generates a base band signal, which must be mixed up to the required transmission frequency. This can be implemented using analog techniques as shown in Fig 3.8 or using a Digital up Converter as shown in Fig 3.9.

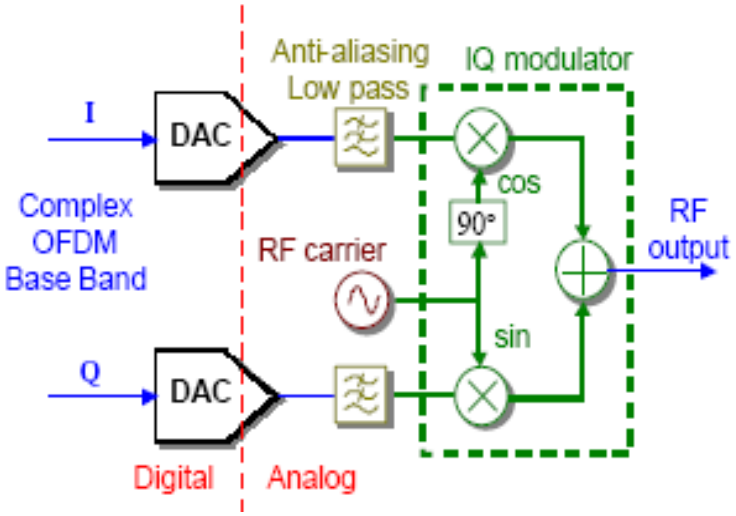


Fig 3.8 RF modulation of complex base band OFDM signal, using analog techniques

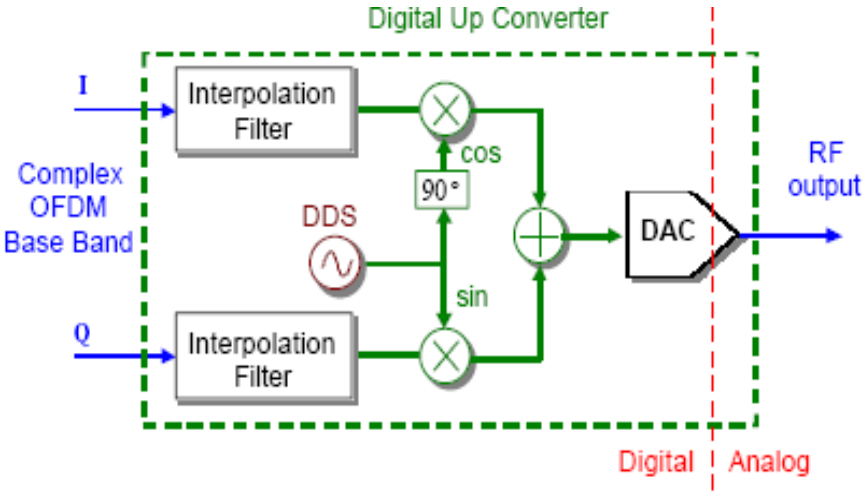


Fig.3.9 RF modulation of complex base band OFDM signal, using digital techniques.

Both techniques perform the same operation, however The performance of the digital modulation will tend to be more accurate due to improved matching between the processing of the I and Q channels, and the phase accuracy of the digital IQ modulator.

3.7 GUARD PERIOD

For a given system bandwidth the symbol rate for an OFDM signal is much lower than a single carrier transmission scheme. For example for a single carrier BPSK modulation, the symbol rate corresponds to the bit rate of the transmission. However for OFDM the system bandwidth is broken up into N_C sub carriers, resulting in a symbol rate that is N_C times lower than the single carrier transmission. This low symbol rate makes OFDM naturally resistant to effects of Inter-Symbol Interference (ISI) caused by multipath propagation. Multipath propagation is caused by the radio transmission signal reflecting off objects in the propagation environment, such as walls, buildings, mountains, etc.

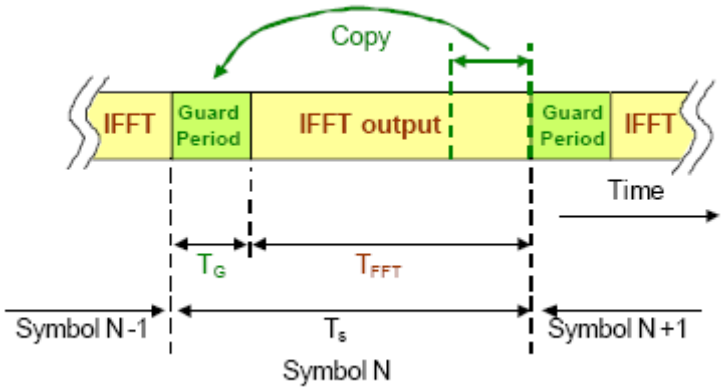


Fig. 3.10 Addition of a guard period to an OFDM signal

These multiple signals arrive at the receiver at different times due to the transmission distances being different. This spreads the symbol boundaries causing energy leakage between them. The effect of ISI on an OFDM signal can be further improved by the addition of a guard period to the start of each symbol. This guard period is a cyclic copy that extends the length of the symbol waveform. Each sub carrier, in the data section of the symbol, (i.e. the OFDM symbol with no guard period added, which is equal to the length of the IFFT size used to generate the signal) has an integer number of cycles. Because of this, placing copies of the symbol end-to-end results in a continuous signal, with no discontinuities at the joins. Thus by

copying the end of a symbol and appending this to the start results in a longer symbol time. Fig 3.10 shows the insertion of a guard period.

The total length of the symbol is $T_S=T_G + T_{FFT}$, where Ts is the total length of the symbol in samples, T_G is the length of the guard period in samples, and T_{FFT} is the size of the IFFT used to generate the OFDM signal. In addition to protecting the OFDM from ISI, the guard period also provides protection against time-offset errors in the receiver. The effects of multipath propagation and how cyclic prefix reduces the inter symbol interference is discussed in detail in chapter4.

3.7.1 Protection against time offset

To decode the OFDM signal the receiver has to take the FFT of each received symbol, to work out the phase and amplitude of the sub carriers. For an OFDM system that has the same sample rate for both the transmitter and receiver, it must use The same FFT size at both the receiver and transmitted signal in order to maintain sub carrier orthogonality. Each received symbol has TG + TFFT samples due to the added guard period. The receiver only needs TFFT samples of the received symbol to decode the signal [18]. The remaining TG samples are redundant and are not needed. For an ideal channel with no delay spread the receiver can pick any time offset, up to the length of the guard period, and still get the correct number of samples, without crossing a symbol boundary. Because of the cyclic nature of the guard period changing the time offset simply results in a phase rotation of all the sub carriers in the signal. The amount of this phase rotation is proportional to the sub carrier frequency, with a sub carrier at the nyquist frequency changing by 180° for each sample time offset. Provided the time offset is held constant from symbol to symbol, the phase rotation due to a time offset can be removed out as part of the channel equalization [19]. In multipath environments ISI reduces the effective length of the guard period leading to a corresponding reduction in the allowable time offset error. The addition of guard period removes most of the effects of ISI. However in practice, multipath components tend to decay slowly with time, resulting in some ISI even when a relatively long guard period is used.

3.7.2 Guard period overhead and sub carrier spacing

Adding a guard period lowers the symbol rate, however it does not affect the sub carrier spacing seen by the receiver. The sub carrier spacing is determined by the sample rate and the FFT size used to analyze the received signal.

$$\Delta f = \frac{F_S}{N_{FFT}} \tag{3.6}$$

In Equation (3.6), Δf is the sub carrier spacing in Hz, F_s is the sample rate in Hz, and N_{FFT} is the size of the FFT. The guard period adds time overhead, decreasing the overall spectral efficiency of the system.

3.7.3 Intersymbol interference

Assume that the time span of the channel is L_c samples long. Instead of a single carrier with a data rate of R symbols/ second, an OFDM system has N subcarriers, each with a data rate of R/N symbols/second. Because the data rate is reduced by a factor of N, the OFDM symbol period is increased by a factor of N. By choosing an

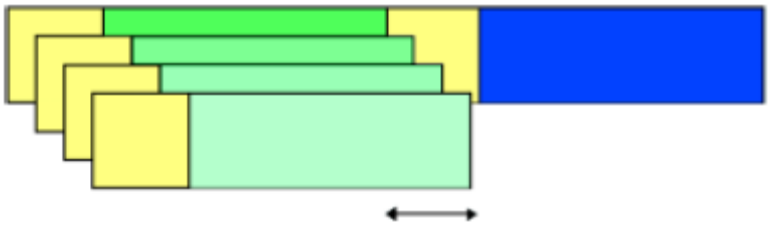


Fig 3.11 Example of intersymbol interference. The green symbol was transmitted first, followed by the blue symbol.

Appropriate value for N, the length of the OFDM symbol becomes longer than the time span of the channel. Because of this configuration, the effect of intersymbol interference is the distortion of the first L_c samples of the received OFDM symbol. An example of this effect is shown in Fig 3.11. By noting that only the first few samples of the symbol are distorted, one can consider the use of a guard interval to remove the effect of intersymbol interference. The guard interval could be a section of all zero samples transmitted in front of each OFDM symbol [20]. Since it does not contain any useful information, the guard interval would be discarded at the receiver. If the length of the guard interval is properly chosen such that it is longer than the time span of the channel, the OFDM symbol itself will not be distorted. Thus, by discarding the guard interval, the effects of intersymbol interference are thrown away as well.

3.7.4 Intrasymbol interference

The guard interval is not used in practical systems because it does not prevent an OFDM symbol from interfering with itself. This type of interference is called intrasymbol interference [21]. The solution to the problem of intrasymbol interference involves a discrete-

time property. Recall that in continuous-time, a convolution in time is equivalent to a multiplication in the frequency-domain. This property is true in discrete-time only if the signals are of infinite length or if at least one of the signals is periodic over the range of the convolution. It is not practical to have an infinite-length OFDM symbol, however, it is possible to make the OFDM symbol appear periodic.

This periodic form is achieved by replacing the guard interval with something known as a cyclic prefix of length L_p samples. The cyclic prefix is a replica of the last L_p samples of the OFDM symbol where $L_p > L_c$. Since it contains redundant information, the cyclic prefix is discarded at the receiver. Like the case of the guard interval, this step removes the effects of intersymbol interference. Because of the way in which the cyclic prefix was formed, the cyclically-extended OFDM symbol now appears periodic when convolved with the channel. An important result is that the effect of the channel becomes multiplicative.

In a digital communications system, the symbols that arrive at the receiver have been convolved with the time domain channel impulse response of Length L_c samples. Thus, the effect of the channel is convolution. In order to undo the effects of the channel, another convolution must be performed at the receiver using a time domain filter known as an equalizer. The length of the equalizer needs to be on the order of the time span of the channel. The equalizer processes symbols in order to adapt its response in an attempt to remove the effects of the channel. Such an equalizer can be expensive to implement in hardware and often requires a large number of symbols in order to adapt its response to a good setting. In OFDM, the time-domain signal is still convolved with the channel response [22]. However, the data will ultimately be transformed back into the frequency-domain by the FFT in the receiver. Because of the periodic nature of the cyclically-extended OFDM symbol, this time-domain convolution will result in the multiplication of the spectrum of the OFDM signal (i.e., the frequency- domain constellation points) with the frequency response of the channel.

The result is that each sub carrier's symbol will be multiplied by a complex number equal to the channel's frequency response at that sub carrier's frequency. Each received sub carrier experiences a complex gain (amplitude and phase distortion) due to the channel. In order to undo these effects, a frequency- domain equalizer is employed. Such an equalizer is much simpler than a time-domain equalizer. The frequency domain equalizer consists of a single complex multiplication for each sub carrier. For the simple case of no noise, the ideal value of the equalizer's response is the inverse of the channel's frequency response [24].

3.8 Advantages and Disadvantages of OFDM as Compared to Single Carrier modulation

3.8.1 Advantages

- Makes efficient use of the spectrum by allowing overlap.
- By dividing the channel into narrowband flat fading sub channels, OFDM is more resistant to frequency selective fading than single carrier systems.
- Eliminates ISI and IFI through use of a cyclic prefix.
- Using adequate channel coding and interleaving one can recover symbols lost due to the frequency selectivity of the channel.
- Channel equalization becomes simpler than by using adaptive equalization techniques with single carrier systems.
- It is possible to use maximum likelihood decoding with reasonable complexity.
- OFDM is computationally efficient by using FFT techniques to implement the modulation and demodulation functions.
- Is less sensitive to sample timing offsets than single carrier systems are.
- Provides good protection against co-channel interference and impulsive parasitic noise.

3.8.2 Disadvantages

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

.3.8.3 Applications of OFDM

- DAB - OFDM forms the basis for the Digital Audio Broadcasting (DAB) standard in the European market.
- ADSL - OFDM forms the basis for the global ADSL (asymmetric digital subscriber line) standard.
- Wireless Local Area Networks - development is ongoing for wireless point-to-point and point-to-multipoint configurations using OFDM technology.
- In a supplement to the IEEE 802.11 standard, the IEEE 802.11 working group published IEEE 802.11a, which outlines the use of OFDM in the 5GHz band.

Chapter 4

TURBO CODES

Introduction :

Turbo codes were first presented at the International Conference on Communications in 1993. Until then, it was widely believed that to achieve near Shannon's bound performance, one would need to implement a decoder with infinite complexity or close. Parallel concatenated codes, as they are also known, can be implemented by using either block codes (PCBC) or convolutional codes (PCCC). PCCC resulted from the combination of three ideas that were known to all in the coding community:

- The transforming of commonly used non-systematic convolutional codes into systematic convolutional codes.
- The utilization of soft input soft output decoding. Instead of using hard decisions, the decoder uses the probabilities of the received data to generate soft output which also contain information about the degree of certainty of the output bits.
- This is achieved by using an interleaver. Encoders and decoders working on permuted versions of the same information.

An iterative decoding algorithm centered around the last two concept would refine its output with each pass, thus resembling the turbo engine used in airplanes. Hence, the name Turbo was used to refer to the process.

4.1 Encoders for Turbo Codes

The encoder for a turbo code is a parallel concatenated convolutional code. Figure 3.1 shows a block diagram of the encoder first presented by Berrou et al [10]. The binary input data sequence is represented by $d_k = (d_{1,}...d_{N)}$ the input sequence is passed into the input of a convolutional encoder [8], ENC_1 and a coded bit stream, $x_{k_1}^p$ is generated. The data sequence is then interleaved. That is, the bits are loaded into a matrix and read out in a way so as to spread the positions of the input bits. The bits are often read out in a pseudo-random manner. The interleaved data sequence is passed to a second convolutional encoder ENC_2, and a second coded bit stream, $x_{k_2}^p$ is generated. The code sequence that is passed to the modulator for transmission is a multiplexed (and possibly punctured) stream consisting of systematic code bits x_k^s and parity bits from both the first encoder $x_{k_1}^p$ and the second encoder $_2$ $x_{k_2}^p$.

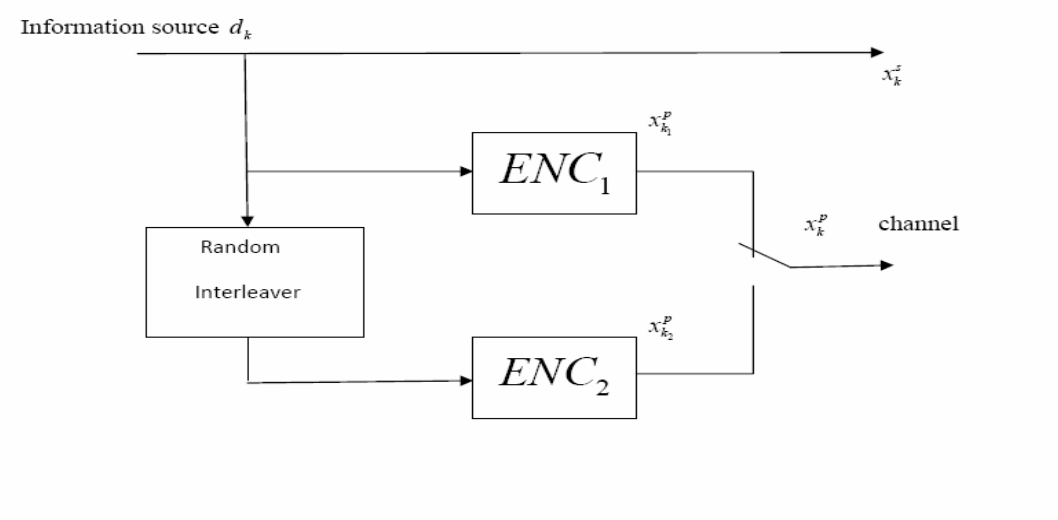
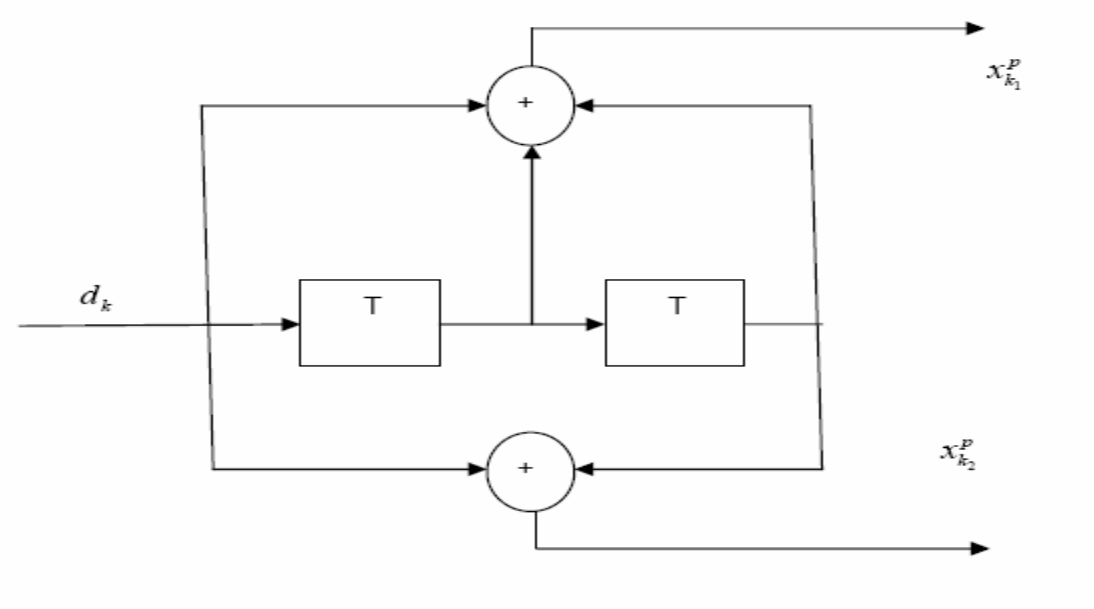


Figure 4.1Structure of a turbo encoder

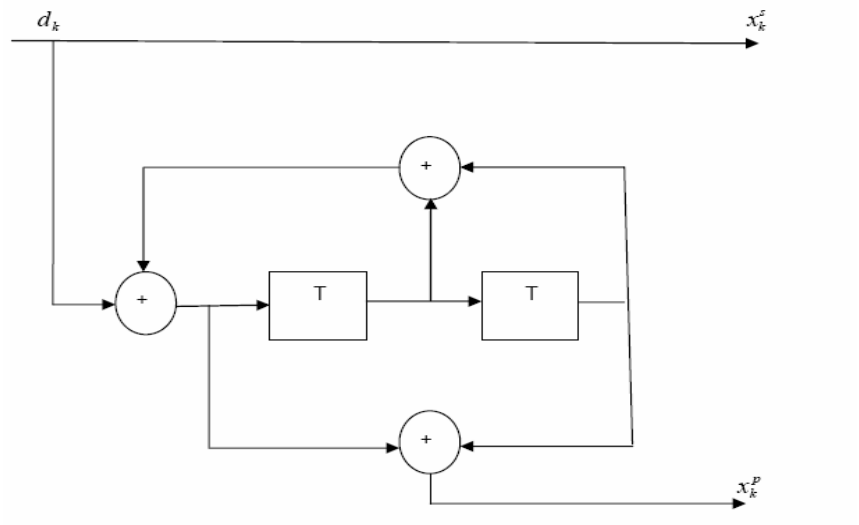
4.1.1 RSC Component Codes

ENC_1 and ENC_2 are Recursive Systematic Convolutional (RSC) codes that is, convolutional codes which use feedback (they are 'recursive') and in which the uncoded data bits appear in the transmitted code bit sequence (they are 'systematic'). A simple RSC encoder is shown in Figure 3.2 along with a non-systematic (NSC) encoder, for comparison. The RSC encoder is rate 1/2, with constraint length K=3, and generator polynomial $G = \{g_1, g_2\} = \{7,5\}$ where g_1 is the feedback connectivity and g_2 is the output connectivity, in octal notation. An RSC component encoder has two output sequences. One is the data sequence; $x_k^s = (x_1^s, ..., x_N^s)$ the other is the parity sequence $x_k^p = (x_1^p, ..., x_N^p)$. To understand why RSC component codes are used in a turbo code encoder, rather than the conventional NSC codes, it is necessary to first discuss the structure of error control codes. The minimum number of errors an error control code can correct is determined by the minimum Hamming distance of the code - the minimum number of bit positions in which any two codewords differ. The linear nature of turbo codes (at least, those using BPSK/QPSK modulations) means that the minimum Hamming distance of the code can be determined by comparing each possible codeword with the all-zeroes codeword. This process simplifies analysis of the code somewhat, and the minimum Hamming distance is then equal to the minimum code weight (number of '1's) which occurs in any codeword. The minimum Hamming distance tends to determine the BER performance of the

code at high SNR - the asymptotic performance. A high minimum Hamming distance results in a steep rate of fall of BER as SNR becomes large, whereas a low value results in a slow rate of fall. In the case of a turbo code, this rate of fall at high SNR is so slow; it is termed an error floor. At low SNR, however, codewords with code weights larger than the minimum value must be considered. It is then that the overall distance spectrum of the code becomes important.



(a) NSC 2



(b)RSC

Figure 4.2 Example of (a) non-systematic convolutional (NSC) and (b) Recursive systematic convolutional (RSC) encoders

This is the relationship between the code weight and the number of codewords with that code weight. Now, RSC codes have an infinite impulse response. That is, if a data sequence consisting of a '1' followed by a series of '0's enters the RSC encoder, a code sequence will be generated containing both ones and zeroes for as long as the subsequent data bits remain zero. This property means that RSC encoders will tend to generate high weight code sequences for groups of data bits spread far apart in the input sequence. An NSC code, however, will return to the all-zeroes state after K-1 input zeroes, where K is the constraint length of the encoder. The infinite impulse response property of RSC codes is complemented in turbo codes by the interleaver between component encoders. An interleaver is a device for permuting a sequence of bits (or symbols) at its input into an alternate sequence with a different ordering at the output.

Turbo codes tend to make use of pseudo-random interleavers, whose role is to ensure that most groups of data bits which are close together when entering one RSC encoder are spread far apart before entering the other RSC encoder. The result is a composite codeword which will often have a high code weight. The details of interleaving will be discussed in more detail in the next section. This does not, however, mean that turbo codes tend to exhibit high minimum Hamming distances, and therefore good asymptotic performance. In fact, the opposite is usually true. We shall see in the following chapter that the pseudo-random nature of most turbo code interleavers tends to result in a mapping such that a few combinations of input bit positions which cause low code weight sequences in one RSC component code are permuted into combinations of positions which generate low code weight sequences in the second RSC code. The result in such a case is a low composite code weight. Such pseudo-random mappings often lead to turbo codes having a low minimum code weight compared to, say, NSC-based convolutional codes, resulting in a marked error floor at high SNR. It is clear from this brief discussion that interleaver design is crucial in ensuring that a turbo code/interleaver combination has the lowest possible error floor. It was mentioned earlier that at low SNR, the distance spectrum of the code as a whole becomes significant in determining BER performance, and that the combination of RSC code and pseudorandom interleaving produces codewords with higher code weights most of the time. This results in there being fewer codewords with relatively low code weights than a comparable convolutional code. It shall be shown later in constructing theoretical bounds for turbo codes that it is the number of codewords at each weight, as well as the actual code weight, which determines the error probability of a code. The low multiplicity of low code weight sequences associated with turbo codes sometimes referred to as spectral thinning, leads to their good BER

performance at low SNR. A full analysis of the turbo code characteristics described here is given in [8].

4.1.2 Interleaving

It was mentioned in the previous section that an interleaver is a device for reordering a sequence of bits or symbols. A familiar role of interleavers in communications is that of the symbol interleaver which is used after error control coding and signal mapping to ensure that fading bursts affecting blocks of symbols transmitted over the channel are broken up at the receiver by a de-interleaver, prior to decoding. Most error control codes work much better when errors in the received sequence are spread far apart. Another role is that of the interleaver between component codes in a serially concatenated code scheme; for example, between a Reed Solomon outer code and a convolutional inner code. The trellis decoding nature of most convolutional codes means that uncorrected errors at the output of the decoder will tend to occur in bursts. The interleaving between the two component codes then ensures that these bursts are adequately spread before entering the outer decoder. In both these examples, the interleaver is typically implemented as a block interleaver. This is a rectangular matrix such that bits or symbols are written in one row at a time, and then read out one column at a time. Thus bits or symbols which were adjacent on writing are spaced apart by the number of rows when reading. The de-interleaving process is simply the inverse of this; writing in column by column and reading out row by row, to achieve the original bit or symbol ordering. Block interleaving is simple to implement, and suitable where the objective is to spread bursts of errors evenly by as large a distance as possible. Block interleavers are not suitable as turbo code interleavers, because they tend to generate large numbers of codewords with a relatively low weight, and therefore with a relatively low hamming distance between them, due to the regularity of the spreading process. Berrou and Glavieux introduced pseudo-random interleaving into turbo codes to solve this problem. A pseudo-random interleaver is a random mapping between input and output positions, generated by means of some form of pseudo-random number generator. Figure 3.3 shows a simple illustration of pseudo-random interleaving. The original data sequence is represented by the sequence of white squares, and the interleaved data sequence is represented by the grey squares. Turbo code BER performance improves with interleaver length - the so called interleaver gain - but the loading and unloading of the interleaver adds a considerable

delay to the decoding process. This would make a 256x256 interleaver unsuitable for, say, and real time speech applications, which are delay sensitive.

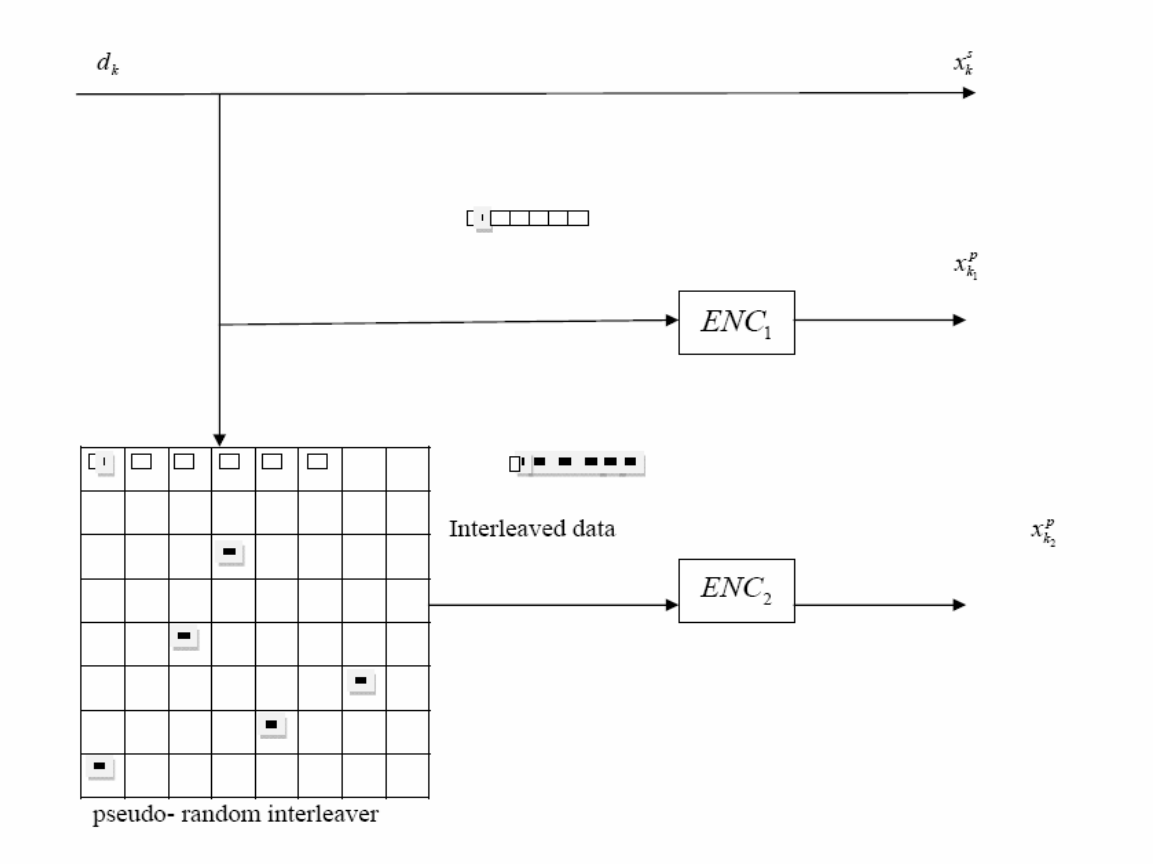


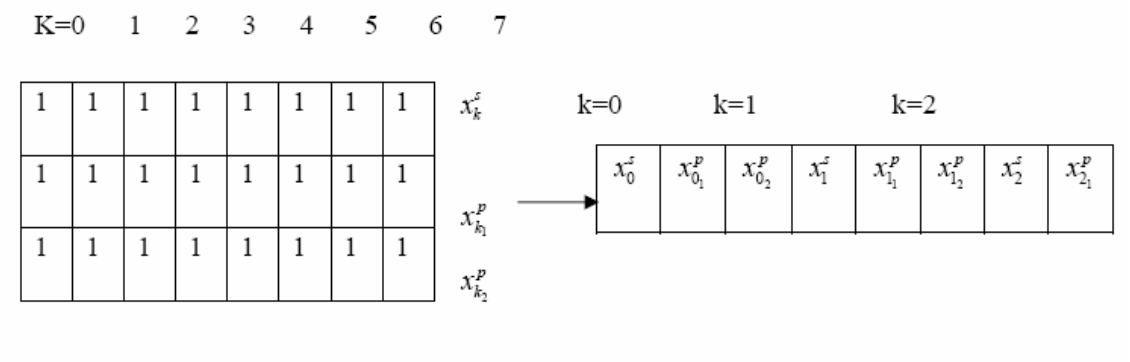
Figure 4.3 Pseudo- Random Interleaving in a Turbo Encoder

4.1.3 Puncturing

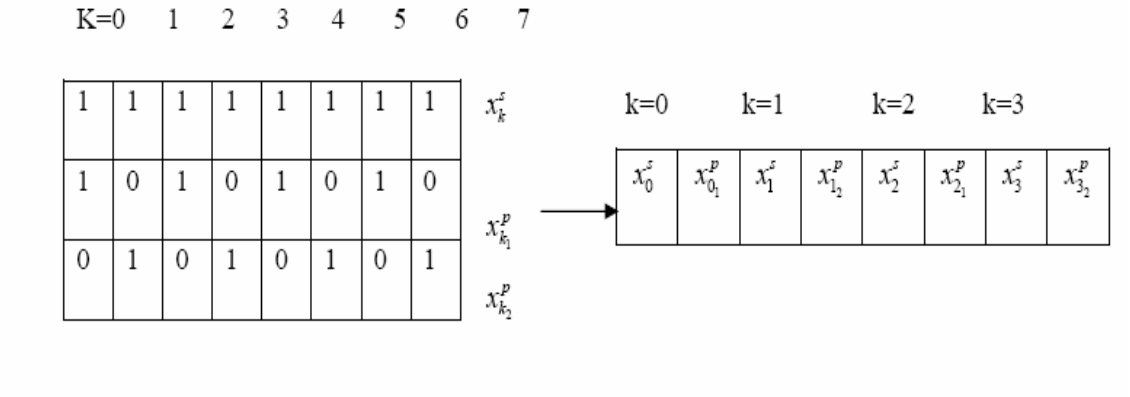
Different code rates are achieved by puncturing the parity bit sequences $x_{k_1}^p$ and $x_{k_2}^p$. Puncturing the data bit sequence $_s\ _k\ x$ leads to a severe degradation in turbo code performance. Figure 3.4 illustrates the puncturing process. A number '1' in the tables represents a code bit that is included in the transmitted code bit sequence, and a number '0' represents a code bit that is excluded, or punctured. On the right of each table is the list of code bits which are included in the transmitted code sequence. In a), the code is unpunctured and is of code rate 1/3 whereas in b), alternate parity bits from each component encoder are punctured at each time interval k. The result is a rate $1/2$ codes. In c), the code is more heavily punctured, to form a rate $3/4$ codes.

4.1.4 Termination

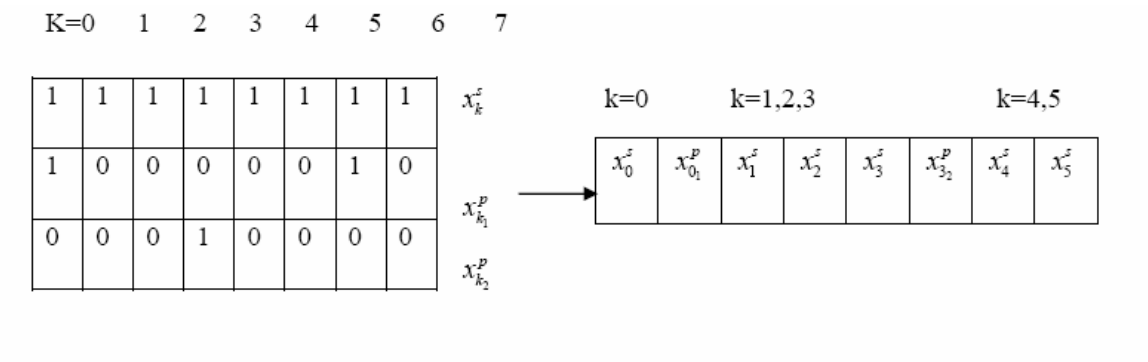
In contrast to block codes, convolutional codes do not have a fixed block length. Convolutional coding is a continuous process, and could span an entire message, rather than a small group of bits. Turbo codes, however, do have a fixed block length, determined by the length of the interleaver. Tail bits are usually appended to each block of data bits entering one or other of the component encoders, to return it to the all-zeroes state at the end of the trellis. This process is called termination, and allows the MAP algorithm to make assumptions about the start and end trellis states. This yields better BER performance. Termination of both component encoders is more difficult, because the terminating sequence for the first encoder is interleaved and may well not, by itself, terminate the second encoder. Interleaver designs have been devised [9] which interleave a terminating sequence in the first encoder into a terminating sequence in the second encoder. This tends to yield better BER performance than a single code terminating process and is another promising area for investigation.



a) Rate 1/3 Turbo Code



b) Rate 1/2 Turbo Code



c) Rate 1/3 Turbo Code

Figure 4.4 Puncture Patterns for Turbo Codes

4.2 Turbo Decoding

A block diagram of a turbo decoder is shown in Figure 3.5. The input to the turbo decoder is a sequence of received code values $R_k = \{y_k^s, y_k^p\}$ from the demodulator. The turbo decoder consists of two component decoders - DEC_1 to decode sequences from ENC_1 , and DEC_2 to decode sequences from ENC_2. Each of these decoders is a Maximum A Posteriori (MAP) decoder. DEC_1 takes as its input the received sequence of systematic values y_k^s and the received sequence of parity values $y_{k_1}^p$ belonging to the first encoder ENC_1. The output of DEC_1 is a sequence of soft estimates EXT_1 of the transmitted data its d_k. EXT_1 is called extrinsic data, in that it does not contain any information which was given to DEC_1 by DEC_2. This information is interleaved, and then passed to the second decoder DEC_2. The interleaver is identical to that in the encoder (Figure 3.1). DEC_2 takes as its input the (interleaved) systematic received values

$_{s\,k}y$ and the sequence of received parity values $_2$ y_k^p from the second encoder ENC_2, along with the interleaved form of the extrinsic information EXT_1, provided by the first decoder. DEC_2 outputs a set of values, which, when de-interleaved using an inverse form of the interleaver, constitute soft estimates EXT_2 of the transmitted data sequence d_k. This extrinsic data, formed without the aid of parity bits from the first code, is feedback DEC_1 . This procedure is repeated in a iterative manner. The iterative decoding process adds greatly to the BER

performance of turbo codes. However, after several iterations, the two decoders' estimates of $_k d$ will tend to converge. At this point, DEC_2 outputs a value $\Lambda(d_k)$; a log-likelihood representation of the estimate of $\Lambda(d_k)$. This log likelihood value takes into account the probability of a transmitted '0' or '1' based on systematic information and parity information from both component codes. More negative values of $\Lambda(d_k)$ represent a strong likelihood that the transmitted bit was a '0' and more positive values represent a strong likelihood that a '1' was transmitted. $\Lambda(d_k)$ is de-interleaved so that its sequence coincides with that of the systematic and first parity streams. Then a simple threshold operation is performed on the result, to produce hard decision estimates, $\Lambda(d_k)$, for the transmitted bits. The decoding estimates EXT_1 and EXT_2 do not necessarily converge to a correct bit decision. If a set of corrupted code bits form a pair of error sequences that neither of the decoders is able to correct, then EXT_1 and EXT_2 may either diverge, or converge to an incorrect soft value. In the next sections, we will look at the algorithms used in the turbo decoding process, within DEC_1 and DEC_2.

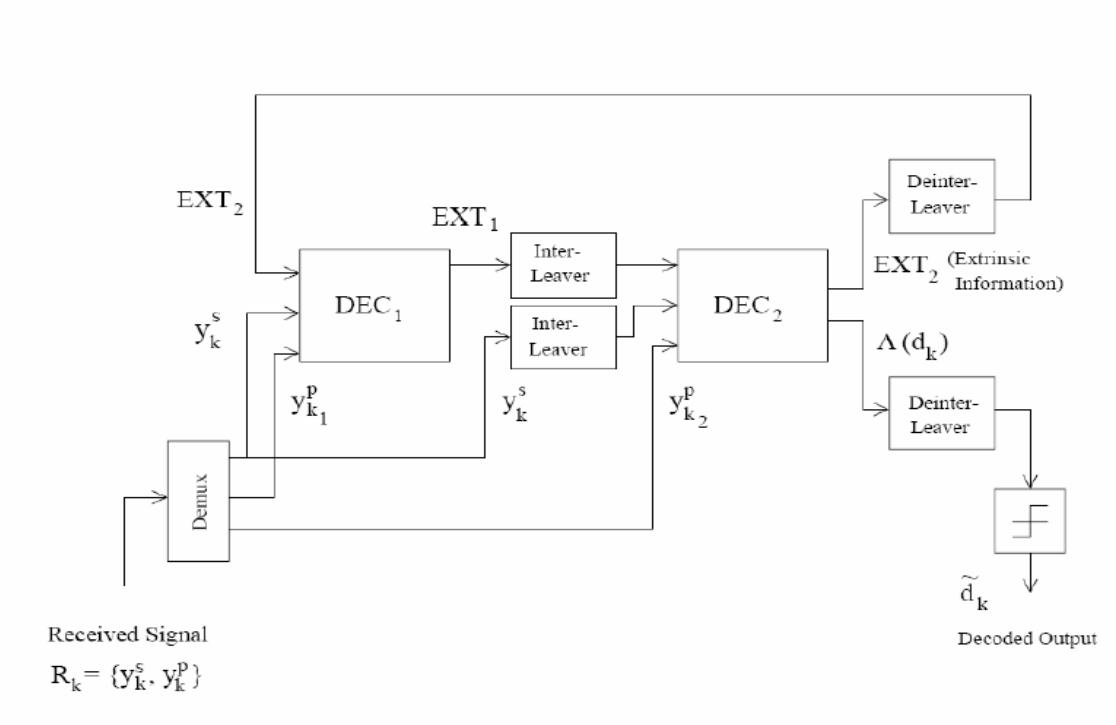


Figure 4.5 Turbo Decoder Structure.

4.3 The MAP Algorithm

We review now the decoding algorithms used within $_1DEC$ and $_2DEC$ to implement the soft-input, soft-output processing needed for iterative decoding. We begin with the modified Bahl, or Maximum A Posteriori (MAP), algorithm presented in Berrou et al.'s original paper [10].

4.3.1 The Need for a Soft Input/Soft Output Algorithm

Decoding of convolutional codes is most frequently achieved using the Viterbi algorithm [11], which makes use of a decoding trellis to record the estimated states of the encoder at a set of time instants. The Viterbi algorithm works by rejecting the least likely path through the trellis at each node, and keeping the most likely one. The removal of unlikely paths leaves us, usually, with a single source path further back in the trellis. This path selection represents a 'hard' decision; on the transmitted sequence. The Viterbi decoder estimates a maximum likelihood sequence. Making hard decisions in this way, at an early point in the decoding process, represents a loss of valuable information. It is frequently advantageous to retain finely-graded probabilities, 'soft decisions', until all possible information has been extracted from the received signal values. The turbo decoding relies on passing information about individual transmitted bits from one decoding stage to the next. The interleaving of the received information sequence between decoders limits the usefulness of estimating maximum likelihood sequences. So, an algorithm is required that can output soft-decision maximum likelihood estimates on a bit-by-bit basis. The decoder should also be able to accept soft decision inputs from the previous iteration of the decoding process. Such a decoder is termed a Soft Input-Soft Output (SISO). Berrou and Glavieux used two such decoders in each stage of their turbo decoder. They implemented the decoders using a modified version of an SISO algorithm proposed by Bahl, Cocke, Jelinek and Raviv [12]. Their modified Bahl algorithm is commonly referred to as the Maximum A Posteriori or MAP algorithm, and achieves soft decision decoding on a bit-by-bit basis by making two passes of a decoding trellis, as opposed to one in the case of the Viterbi algorithm. One pass is made in the forward

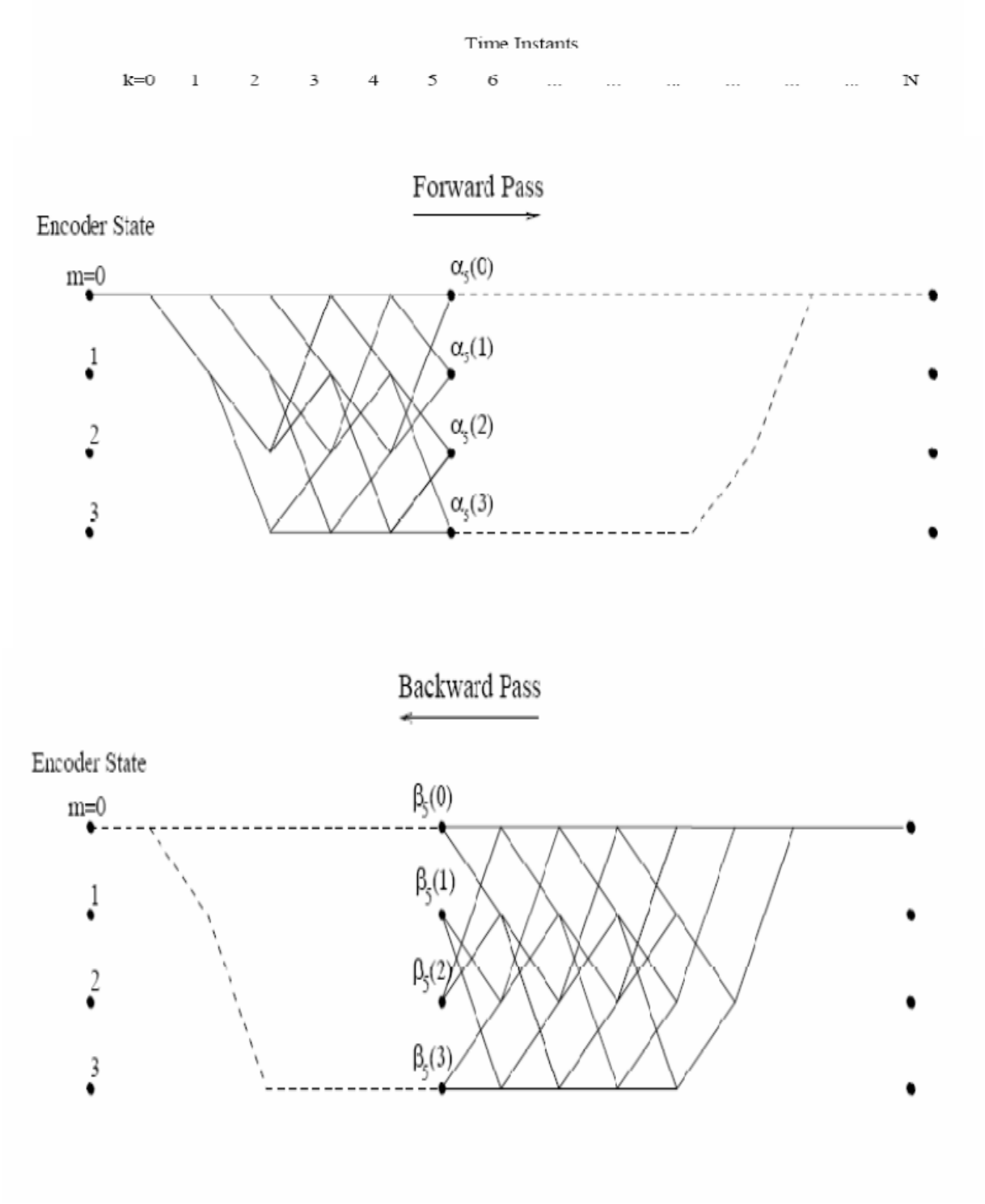


Figure 4.6 Modified Bahl Algorithm.

direction and one in the backward direction, as shown in Figure 3.6. $m = 0....(M-1)$ represents the states of each decoder (DEC_1 or DEC_2), where the total number of states $M = 2^{k-1}$ and K is the constraint length of the component codes ENC_1 and ENC_2. $\alpha_k(m)$ and $\beta_k(m)$ are the state

probabilities that the component code is in state m at time instant k, in the forward and backward directions

4.3.2 Derivation of the MAP Algorithm

The MAP algorithm has been described in [13] and is repeated here:

Forward Pass - Calculation of α State Probabilities

Figure 3.7 illustrates the calculations made at each time interval k, for the simple four states RSC code with trellis connectivity defined by the generator polynomial G={7,5} First, the trellis is traversed in the forward direction. At each node, the current state probability, α is calculated by multiplying the state probability at the previous node $\alpha_{k-1}(m')$ by the branch transition probability, $\gamma_{k-1}(m', m)$ - (Equation 3.3), given the received code pair $R_k = \{y_k^s, y_k^p\}$. This is expressed as follows:

$$\alpha_k(m) = \frac{\sum_{m'} \sum_{i=0}^{1} \gamma_i((y_k^s, y_k^p) m', m) \alpha_{k-1}(m')}{\sum_{m'} \sum_{i=0}^{1} \gamma_i((y_k^s, y_k^p) m', m) \alpha_{k-1}(m')} \qquad (4.1)$$

Where m is the current state, m' is the previous state and i is the data bit ('0' or '1') corresponding to each branch exiting a node.

Backward Pass - Calculation of β State Probabilities

Then the trellis is traversed in the reverse direction. Again the probability of each branch being taken is calculated. The current state probability $\beta_k(m)$ is found by multiplying the probability of arriving in the previous state $\beta_{k+1}(m)$ by the probability of taking the current state transition $\gamma_{k+1}(m', m)$ given the current received values $R_k = \{y_{k+1}^s, y_{k+1}^p\}$. This is expressed as

(4.2)

Where the symbols have the same meaning as before, but β is the backward state probability. The transition probability for each branch between nodes is given by the equation:

$$\gamma_i((y_k^s, y_k^p), m', m) = p((y_k^s, y_k^p) \mid d_k = i, m', m).q(d_k = i \mid m', m).\pi(m \mid m') \qquad (4.3)$$

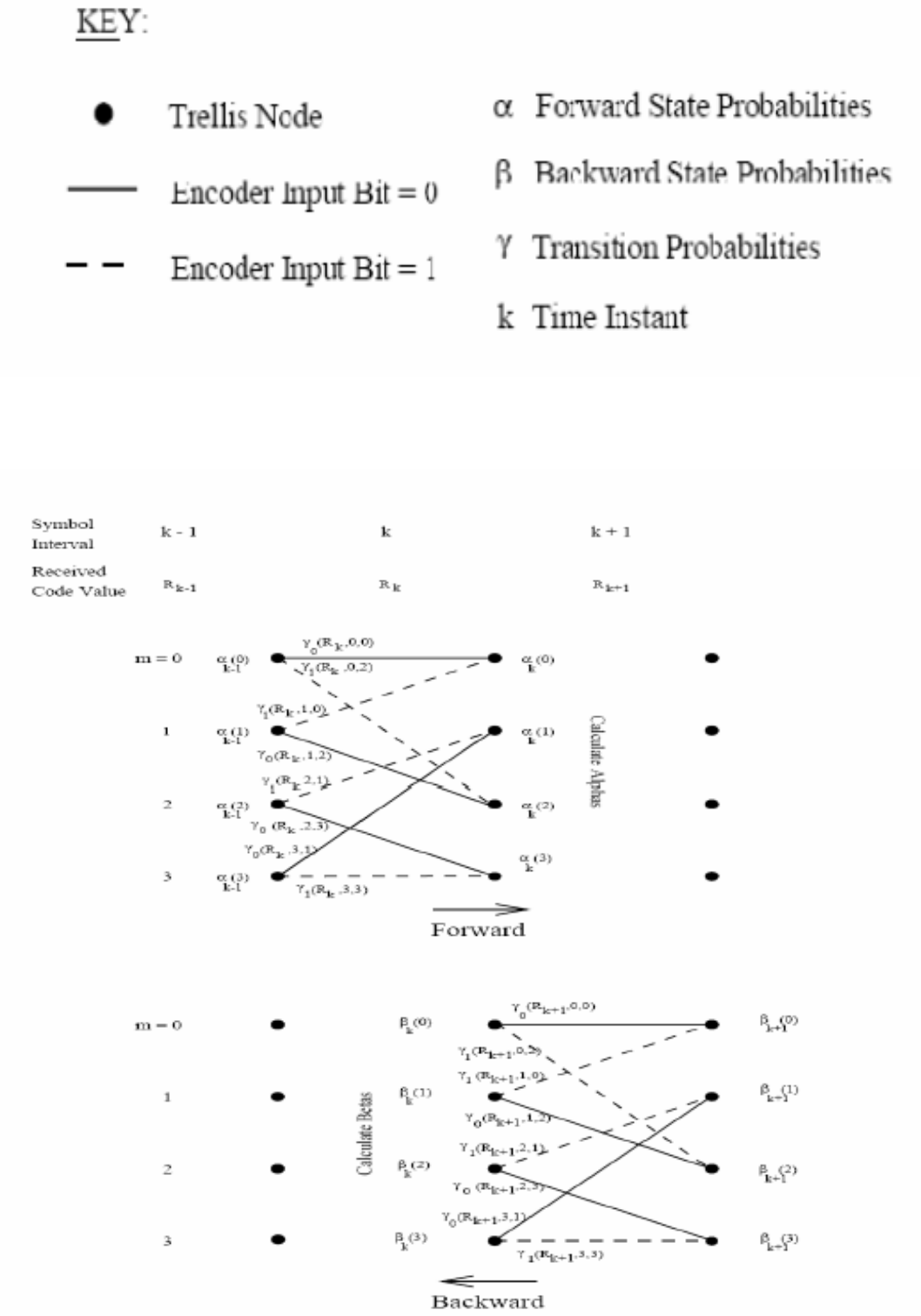


Figure 4.7 Calculations of State Probabilities in the Modified Bahl Algorithm

P (…) is the transition probability of the channel; that is, the probability that a given received symbol will result when symbol *y* is transmitted. This function is defined by the pdf of the channel; for example, a Gaussian pdf in the case of the AWGN channel. q(...) is the probability that any branch from a node can be taken, given the previous state *m',* current state m and the data bit *i* associated with the branch. q (…) is either '0' or '1', depending on the generator polynomial of the RSC encoder represents the a priori information which forms the input to each component MAP decoder from the other MAP decoder, within the iterative decoding process.

Calculation of Log Likelihood Probabilities $\Lambda(d_k)$:

Finally, the forward and backward probabilities at each time interval *k* of the trellises are used to provide a soft estimate of whether the transmitted data bit $_k d$ was a '1' or a '0'. This process is illustrated in Figure 3.8, again for the RSC code with generator polynomial G= {7, 5}. The soft estimate is represented as a log likelihood ratio (LLR), in Figure 3.5, as this is a convenient form for representing a probability which can have a wide dynamic range. It is calculated as follows:

$$\Lambda(d_k) = \ln \frac{\sum_m \sum_{m'} \gamma_1((y_k^s, y_k^p), m', m)\alpha_{k-1}(m')\beta_k(m)}{\sum_m \sum_{m'} \gamma_0((y_k^s, y_k^p), m', m)\alpha_{k-1}(m')\beta_k(m)} \qquad (4.4)$$

$\Lambda(d_k)$ represents the probability that the current data bit is a '0' (if $\Lambda(d_k)$, is negative) or a '1' (if $\Lambda(d_k)$, is positive). After a number of iterations, typically 8…18, the de-interleaved value of $\Lambda(d_k)$, from $_2 DEC$ is converted to a hard decision estimate, of the transmitted data bit. This forms the output of the final turbo decoder stage. The MAP algorithm, as described in part so far, is optimal for estimating the maximum likelihood data sequence on a bit-by-bit basis. However, the MAP algorithm is not usually implemented for reasons of computational complexity. Specifically, although the soft output of the algorithm, $\Lambda(d_k)$, is represented as a log likelihood probability, intermediate values α, β and γ are represented as actual probabilities in the

range0…1,($10^{-\infty}$ …..10^{0}) The dynamic range of these pure probabilities is high, which means that, in terms of implementation, memory needs are also high. Also, Equations 3.1 through 3.4 require a high number of multiplications - operations which are computationally intensive. To simplify the implementation, it is better to work with the logarithmic values of α, β, γ and Λ, rather than their actual probabilities. To save space here, the full implementation of a turbo decoder is now described in terms of these simplified algorithms: The Max Log Map and Log Map algorithms.

4.4 The Max Log-MAP Algorithm

As stated above, working in the logarithmic domain compacts the dynamic range of all the values we are working with. It also converts the multiplication operations in Equations 3.1 through 3.4 to additions. Let us take the logarithm of derived in $\gamma_i((y_k^s, y_k^p)m', m)$ Equation 3.3 and, for the AWGN channel, insert:

$$p(y_k^p \mid d_k = i, m', m) = \frac{1}{\sqrt{\pi N_0}} . \exp(-\frac{1}{N_0}[y_k^p - x_k^p(i, m', m)]^2)$$
$$p(y_k^p \mid d_k = i) = \frac{1}{\sqrt{\pi N_0}} . \exp(-\frac{1}{N_0}[y_k^s - x_k^s(i)]^2) \qquad (4.5)$$

Where x_k^s and x_k^p are the transmitted systematic and parity bits, respectively, corresponding to the current branch at time instant k, y_k^s and y_k^p are the received systematic and parity values, and No is the noise power spectral density. Again ' m and m represent the previous and current states in the trellis, respectively. We can now write the logarithm of Equation 3.3 for *q (...)* = 1 as:

$$\ln \gamma_i[(y_k^s, y_k^p), m', m] = \frac{2 y_k^s x_k^s(i)}{N_0} + \frac{2 y_k^p x_k^p(i)}{N_0} + \ln P_t(m \mid m') + K \qquad (4.6)$$

Note that evaluation of the branch transition γ assumes knowledge of the noise power spectral density N_o. Research has been done into the performance of the MAP algorithm for noisy channel estimates [14], and shows that there is a spread of values around the actual value of No

which will provide good turbo decoding performance. Constant *K* cancels out in the calculation of and $\alpha_k(s_k)$ and $\beta_k(s_k)$ for the logarithm of the α term we get:

$$\ln \alpha_k(m) = \ln\left[\sum_{m'}\sum_{i=0}^{1}\exp(\ln\gamma_i[(y_k^s, y_k^p), m', m]) + \ln\alpha_{k-1}(m'))\right]$$
$$-\ln\left[\sum_{m}\sum_{m'}\sum_{i=0}^{1}\exp(\ln\gamma_i[(y_k^s, y_k^p), m', m]) + \ln\alpha_{k-1}(m'))\right] \quad (4.7)$$

So we have lost the multiplications from Equation 3.1, but we have gained exponential terms, which are themselves computationally intensive. However, we can approximate the sum of a series of log terms by considering only the maximum log value:

$$\ln\left[\exp(\delta_1) + ... + \exp(\delta_n)\right] \approx \max_{i\in(1..n)}\delta_i \quad (4.8)$$

The simplification provides us with the so-called Max Log-MAP algorithm. If we use this approximation in Equation 3.7, we can write $\overline{\alpha}_k(m)$ the log version of $\alpha_k(m)$, as:

$$\overline{\alpha}_k(m) \approx \max_{(m',i)}\left\{\overline{\gamma}_i\left[(y_k^s, y_k^p), m', m\right] + \overline{\alpha}_{k-1}(m')\right\}$$
$$-\max_{(m,m',i)}\left\{\overline{\gamma}_i\left[(y_k^s, y_k^p), m', m\right] + \overline{\alpha}_{k-1}(m')\right\} \quad (4.9)$$

Similarly

$$\overline{\beta}_k(m) \approx \max_{(m',i)}\left\{\overline{\gamma}_i\left[(y_{k+1}^s, y_{k+1}^p), m, m'\right] + \overline{\beta}_{k+1}(m')\right\}$$
$$-\max_{(m,m',i)}\left\{\overline{\gamma}_i\left[(y_{k+1}^s, y_{k+1}^p), m, m'\right] + \overline{\alpha}_k(m)\right\} \quad (4.10)$$

The log-likelihood probability of each bit $\Lambda(d_k)$ is then given approximately by:

$$\Lambda(d_k) \approx \max_{(m,m')}\left\{\overline{\gamma}_1\left[(y_k^s, y_k^p), m', m\right] + \overline{\alpha}_{k-1}(m') + \overline{\beta}_k(m)\right\}$$
$$-\max_{(m,m')}\left\{\overline{\gamma}_0\left[(y_k^s, y_k^p), m', m\right] + \overline{\alpha}_{k-1}(m') + \overline{\beta}_k(m)\right\} \quad (4.11)$$

In accordance with the findings in [10], $\Lambda(d_k)$, must be divided into three components to allow iterative decoding to occur. These components are termed the extrinsic component (that part of $\Lambda(d_k)$, which each decoder derived independently of the other decoder), the a priori component (that part which was derived solely by the other decoder) and the systematic component (that part which derives solely from the systematic part of the received signal). In order to separate the components of $\Lambda(d_k)$, in this way, we must start by defining that part of the branch transition probability γ which is derived purely from the parity component of the received signal. In log form, from Equation 3.3, we can write:

$$\overline{\gamma}_i'(y_k^p, m', m) = \ln p(y_k^p \,|\, d_k = i, m, m') + \ln q(d_k = i \,|\, m, m') \tag{4.12}$$

We can then insert this value into Equation 4.11, to obtain:

$$\begin{aligned}\Lambda(d_k) \approx & \left\{ \max_{(m,m')} \left(\overline{\gamma}_1 \left[(y_k^s, y_k^p), m', m \right] + \overline{\alpha}_{k-1}(m') + \overline{\beta}_k(m) \right) \right. \\ & \left. + \ln p(y_k^s \,|\, d_k = 1) + \ln P_r(d_k = 1) \right\} \\ & - \left\{ \max_{(m,m')} \left(\overline{\gamma}_0 \left[(y_k^s, y_k^p), m', m \right] + \overline{\alpha}_{k-1}(m') + \overline{\beta}_k(m) \right) \right. \\ & \left. + \ln p(y_k^s \,|\, d_k = 0) + \ln P_r(d_k = 0) \right\}\end{aligned} \tag{4.13}$$

Finally, this can be arranged as:

$$\begin{aligned}\Lambda(d_k) \approx & \max_{(m,m')} \left(\overline{\gamma}_1' \left[y_k^p, m', m \right] + \overline{\alpha}_{k-1}(m') + \overline{\beta}_k(m) \right) \\ & - \max_{(m,m')} \left(\overline{\gamma}_0' \left[y_k^p, m', m \right] + \overline{\alpha}_{k-1}(m') + \overline{\beta}_k(m) \right) + \frac{4 y_k^s}{N_0} + L(d_k)\end{aligned} \tag{4. 14}$$

The first two terms form the extrinsic component, the third term is the systematic component and the fourth is the a priori component. It is the extrinsic component which is passed from one MAP decoder to the other, where, after interleaving, it forms the a priori input term of

the new decoder. In Equation 3.6, the log probability of a branch transition was calculated. The unknown term so far in this is the a priori term. We can determine this term, $L(d_k)$ as follows.

$q(d_k = 1/m, m') = 1$ Then

$$L(d_k) = \ln\left(\frac{P_\tau(d_k = 1)}{P_\tau(d_k = 0)}\right) = \ln\left[\frac{P_\tau(m|m')}{1 - P_\tau(m|m')}\right] \tag{4.15}$$

Hence $p_r(m/m') = -\ln(1 + \exp(L(d_k)))$ again, an approximation for $p_r(m/m')$ can be found using equation 4.8

$$\ln P_\tau(m|m') \approx L(d_k) - \max[0, L(d_k)] \tag{4.16}$$

Alternatively if $q(d_k = 1/m, m') = 1$ then:

$$L(d_k) = \ln\left(\frac{P_\tau(d_k = 1)}{P_\tau(d_k = 0)}\right) = \ln\left[\frac{1 - P_\tau(m|m')}{P_\tau(m|m')}\right] \tag{4.17}$$

Hence $p_r(m/m') = -\ln(1 + \exp(l(d_k)))$ Again, an approximation for can be $p_r(m/m')$ can be found using Equation 4.8:

$$\ln P_r(m/m') \approx -\max[0, L(d_k)] \tag{4.18}$$

If The Max Log-MAP algorithm described above allows us to greatly simplify the implementation of the MAP algorithm. However, the approximation in Equation 3.8 makes it sub-optimal and its BER performance is therefore poorer than that of the MAP algorithm. A solution to the problem is called the Log MAP algorithm.

4.5 The Log-MAP Algorithm

Equation 3.8 estimated in $\ln(\exp(\delta_1) + ... + \exp(\delta_n))$ by considering only the maximum exponential term. Robertson et al. used the Jacobian algorithm [16] to improve the approximation:

$$\ln(\exp(\delta_1 + \exp(\delta_2)) = \max(\delta_1, \delta_2) + \ln(1 + \exp(-|\delta_2 - \delta_1|)) = \max(\delta_1, \delta_2) + f_c(|\delta_2 - \delta_1|) \quad (4.19)$$

Where $f_c(...)$ is a correction function. In fact, $\ln(\exp(\delta_1) + ... + \exp(\delta_n))$ the expression can be computed exactly using the Jacobian logarithm, by finding the maximum term and applying the correction recursively through a sequence of δ values. The recursion is initialised with Equation 3.19. we assume $\ln(\exp(\delta_1) + ... + \exp(\delta_n))$ is known. Hence

$$\ln(\exp(\delta_1) + ... + \exp(\delta_n)) \; = \; \ln(\Delta + \exp(\delta_n) \quad (4.20)$$

Where

$$\begin{aligned} \Delta &= \exp(\delta_1) + ... + \exp(\delta_{n-1}) \\ &= \exp(\delta) \\ &= \max(\ln \Delta, \delta_n) + f_c(|\ln \Delta - \delta_n|) \\ &= \max(\delta, \delta_n) + f_c(|\delta - \delta_n|) \end{aligned} \quad (4.21)$$

When implementing the Log-MAP algorithm, all maximizations over two values are augmented by the correction function $f_c(...)$ as defined by Equation 3.19. By correcting the Max Log-MAP algorithm in this way, the precision of the MAP algorithm has been preserved. The incorporation of the correction function increases the complexity slightly relative to the Max Log-MAP algorithm. However, since $f_c(...)$ depends only on $|\delta_2 - \delta_1|$, this effect can be minimized by storing the correction values in a simple look-up table. It was found in that good performance can be achieved with as few as eight correction values.

Chapter 5

RESULTS & CONCLUSION

5.1 Introduction

An OFDM system was modeled using Matlab to allow various parameters of the system to be varied and tested. The aim of doing the simulations was to measure the performance of OFDM under AWGN channel and RAYLEIGH channel conditions, for different modulation schemes like BPSK, QPSK used in IEEE 802.11a wireless LAN standard.

Following this introduction, section 5.2 discusses model used in simulation, steps in OFDM simulation, modulation schemes and their constellation diagrams. Section 5.3 presents the parameters used in simulation. Section 5.4 provides the simulation results of OFDM system for different modulation schemes. It also shows the results to compare the performance of OFDM using coded and uncoded OFDM

5.2 simulation model

Since the main goal of this thesis was to simulate the COFDM system by utilizing turbo code. The block diagram of the entire system is shown in Figure 5.1.

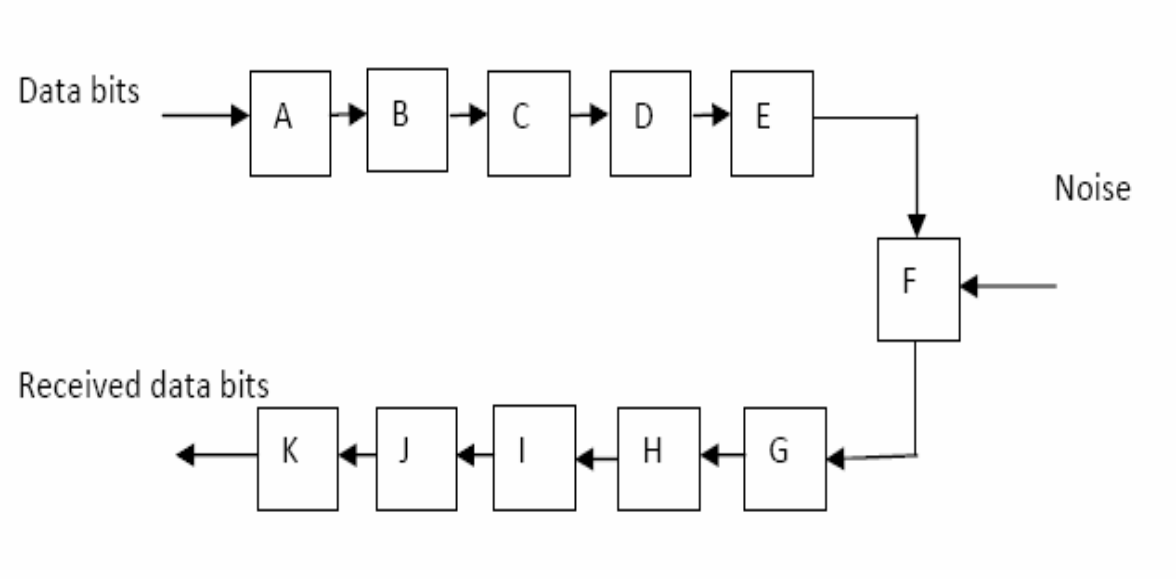


Figure 5.1 Simulation model of turbo coded OFDM

Here A = turbo encoder, B = BPSK/QPSK modulation, C = serial to parallel converter, D = IFFT, E = parallel to serial convertor, F = channel with noise, G = serial to parallel Converterr, H = FFT, I = parallel to serial converterr, J = BPSK/QPSK demodulation and K = turbo decoder.

5.3 simulation parameters

Digital Modulation : BPSK QPSK, QAM

Turbo code rates : 1/3

SISO Decoder :Log-MAP

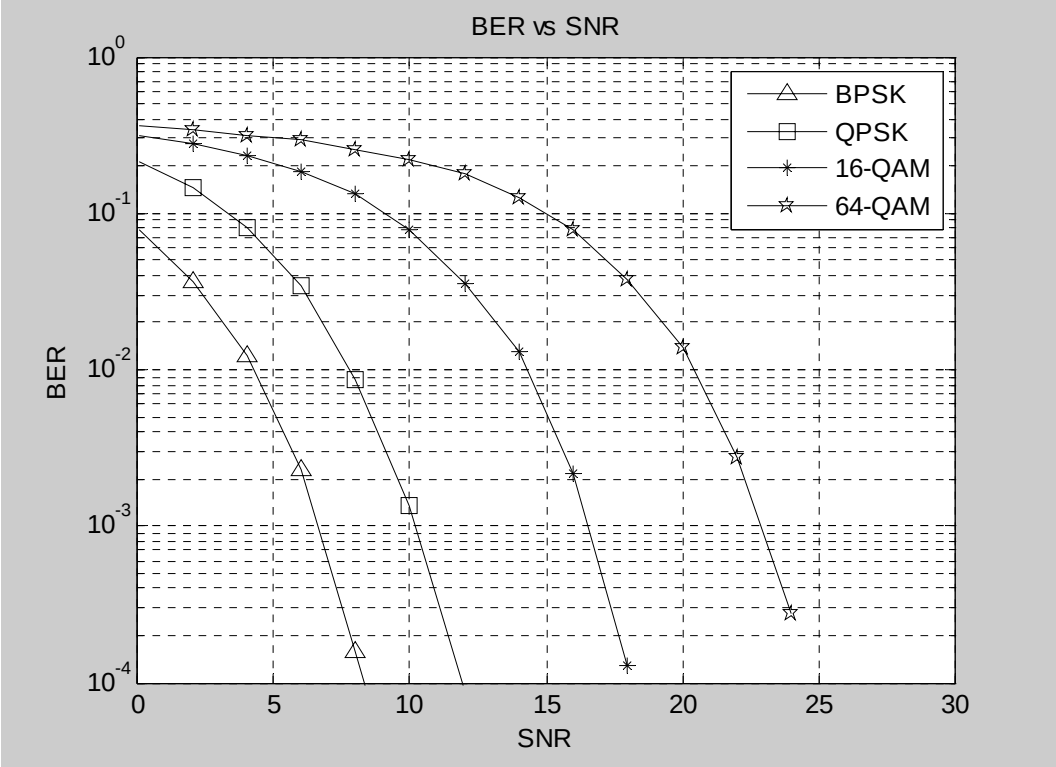
Code Generator : {111, 101}

Interleaver : pseudo random interleaver

We measured the performance of the turbo coded OFDM through MATLAB simulation. The simulation follows the procedure listed below:

1. Generate the information bits randomly.
2. Encode the information bits using a turbo encoder with the specified generator matrix.
3. Use QPSK or different QAM modulation to convert the binary bits, 0 and 1, into complex signals (before these modulation use zero padding)
4. Performed serial to parallel conversion.
5. Use IFFT to generate OFDM signals, zero padding is being done before IFFT.
6. Use parallel to serial convertor to transmit signal serially.
7. Introduce noise to simulate channel errors. We assume that the signals are transmitted over an AWGN (Additive White Gaussian Noise) and Rayleigh channel.
8. At the receiver side, perform reverse operations to decode the received sequence.
9. Count the number of erroneous bits by comparing the decoded bit sequence with the original one.
10. Calculate the BER and plot it.

All the simulations are done to achieve BER at .For simulation results two channel are AWGN and RAYLEIGH are used. The BER performance of TCOFDM system is compared with uncoded OFDM system. As mentioned before, bursty errors deteriorate the performance of the any communications system. The burst errors can happen either by impulsive noise or by deep frequency fades. Fig 5.2 shows the performance of the uncoded OFDM system with AWGN.



.

Fig 5.2 BER vs. SNR plot for OFDM using BPSK, QPSK, 16 QAM, 64 QAM

To improve the performance of this system FEC code can be used. Convolution code is a good example of FEC code. Convolution coding in OFDM can give performance improvement of some 5 db on AWGN channel over the uncoded OFDM system at required BER. Further improvement in the performance can be obtained by applying turbo coding instead of convolution code. As described in chapter 3, turbo code gives better performance at low SNR. The BER performance of TCOFDM system is compared with the respective uncoded system under the fading AWGN channel and RAYLEIGH fading channel. No other channel codes are considered in this thesis as the iterative decoding scheme easily outperforms conventional codes. We have successfully simulated the turbo codes with polynomial generators, (1, 5 / 7)8 which are iteratively decoded by Log- MAP for a number of decoding iterations. We have simulated the polynomial (1, 5 / 7)8 as a reference. The simulated results are shown in from Figure 5.3 to figure5.6. From the results, we observe that both turbo codes (1, 5/7)8 give considerably good BER performance. The overall performance is considered very well in operation under fading

channel which is also efficient in terms of power consumption as compared to the uncoded system.

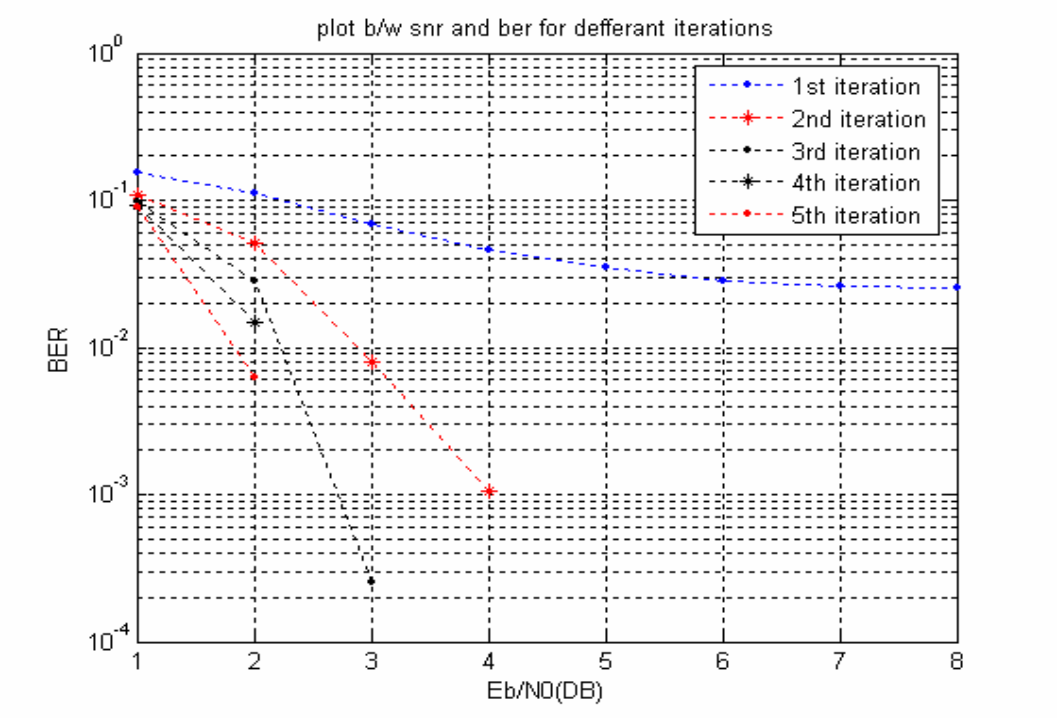


Figure 5.3 BER vs. SNR plot for turbo codes for different iterations

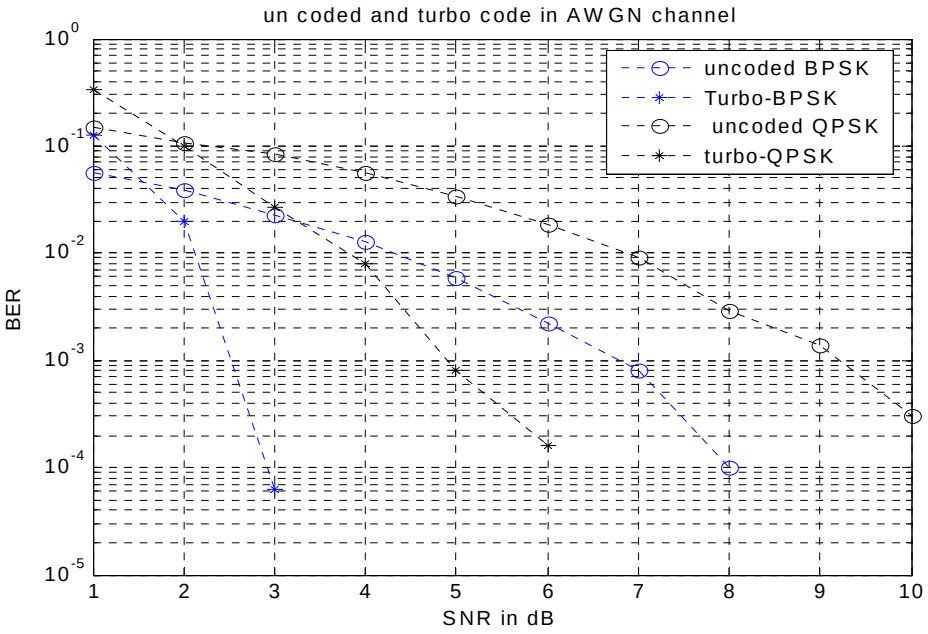


Figure 5.4 BER vs. SNR plot for uncoded and turbo coded OFDM using BPSK and QPSK.

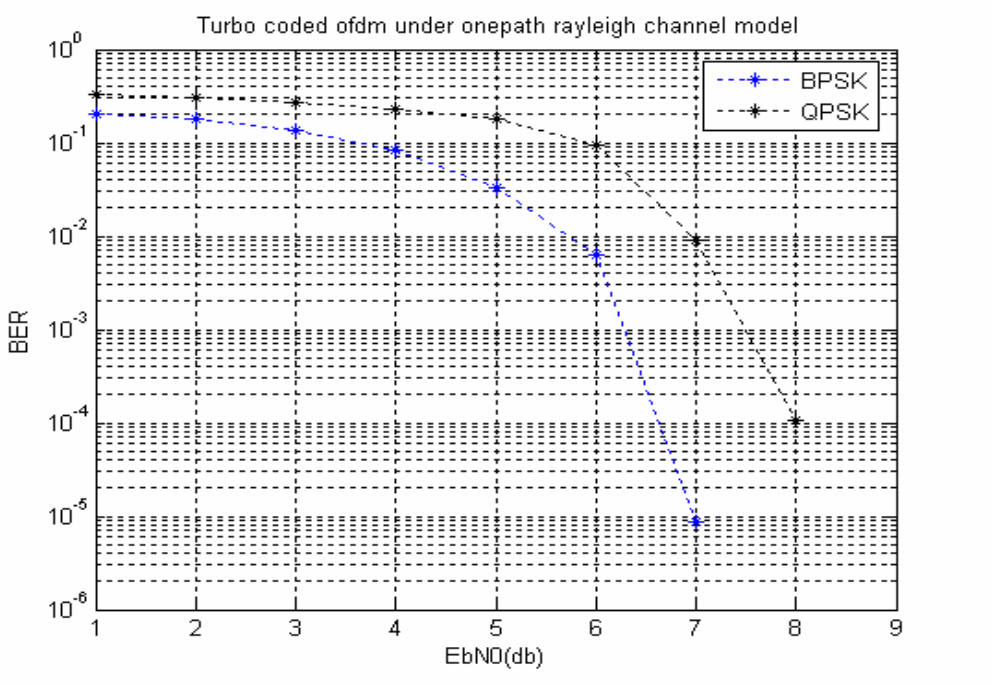


Figure 5.5 BER vs. SNR plot for turbo coded OFDM under one path rayleigh channel .

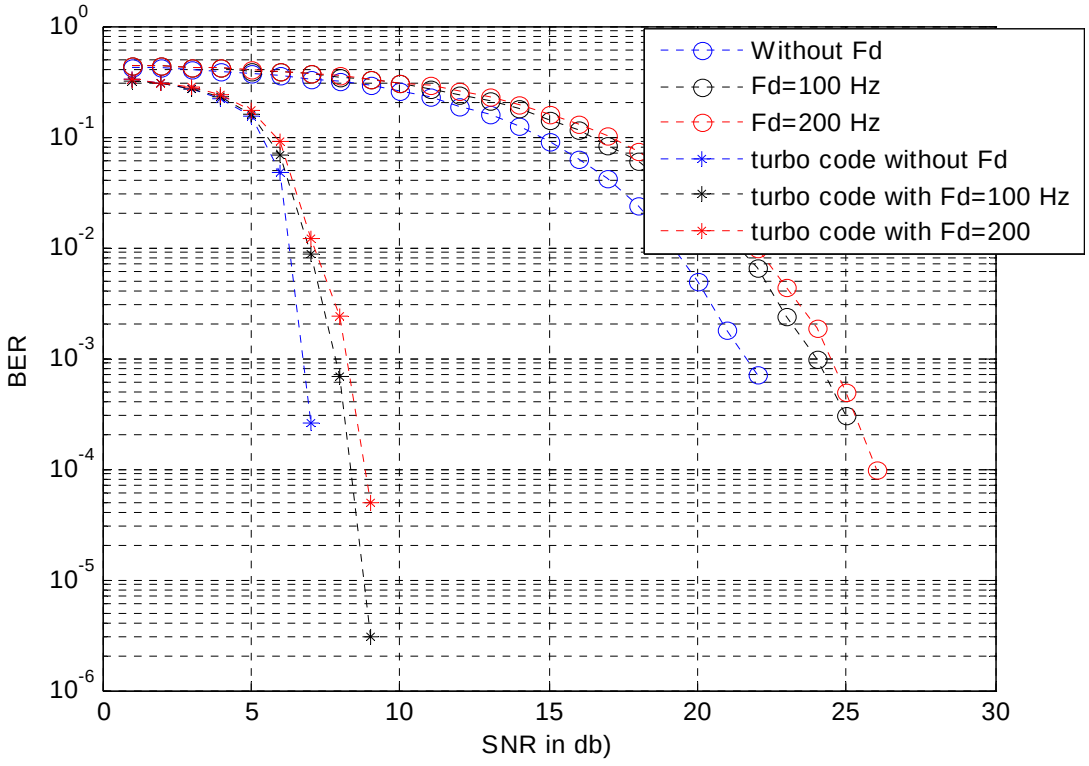


Figure 5.6 BER vs SNR plot for uncoded and turbo coded OFDM under Rayleigh fading channel.

Conclusion:

To conclude, this major project gives the detail knowledge of a current key issue in the field of communications named Orthogonal Frequency Division Multiplexing (OFDM). We focused our attention on turbo codes and their implementation. We described the encoder architecture. In our case, the code is the result of the parallel concatenation of two identical RSCs. The code can be punctured in order to fulfill bit rate requirements. The decoder succeeded in its duty thanks to the decoding algorithms that it is built around. We focused mainly on the study of the MAP. We discovered that the power of the scheme came from the two individual decoders performing the MAP on interleaved versions of the input. Each decoder used information produced by the other as a priori information and outputted a posteriori information. We elaborated on the performance theory of the codes Then we tied concepts of OFDM and turbo coding with a target-based, modulation scheme. First I developed an OFDM system model then try to improve the performance by applying forward error correcting codes to our uncoded system. From the study of the system, it can be concluded that we are able to improve the performance of uncoded OFDM by convolutional coding scheme. Further improvement on the performance has been achieved by applying turbo coding to uncoded OFDM system. Turbo codes with low order decoding iterations have been evaluated. The SNR performance for BER 10^{-2} and 10^{-4}, that are suitable for speed and data applications, are analyzed. As a result, the TCOFDM system with least number of decoding iterations, 3 to 5 iterations are shown to be sufficient to provide good BER performance**.**

Scope of future work:

- VLSI implementation of turbo codes.
- Multi path channel can also be investigated.

REFERENCES:

[1] Ramjee Prasad, "OFDM for Wireless Communications systems", Artech House Publishers, 2004.

[2] L. Hanzo, M. Munster, B.J. Choi, T. Keller, "OFDM & MC-CDMA for Broadband Multiuser Communications, WLANs and Broadcasting" John Wiley Publishers, 2003..

[4] Anibal Luis Intini, "orthogonal Frequency Division Multiplexing For Wireless Networks" Standard IEEE 802.11a, University Of California, Santa Barbara.

[5] Weinstein, S. B. and Ebert, P. M., "Data transmission by frequency division COM-19, multiplexing using the discrete Fourier transform," *IEEE Trans. Comm. Technology*,

. Vol. pp. 282- 289, Oct. 1971.

[8] L. Perez, J. Seghers, and D. Costello, "A Distance Spectrum Interpretation of Turbo Codes", *IEEE Transactions on Information Theory*, vol. 42, no. 6, pp. 1698–1709, Nov. 1996.

[9] W. J. Blackert, E. K. Hall, and S. G. Wilson, "Turbo Code Termination and Interleaver Conditions", *IEE Electronics Letters*, vol. 31, no. 24, pp. 2082–2084, Nov 1995.

[10] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon Limit Error-Correcting Coding: Turbo Codes", *Proceedings of the IEEE International Conference on Communications ICC '93, Geneva.* pp. 1064–1070, May 1993.

[11] P. Robertson, "Improving Decoder and Code Structure of Parallel Concatenated Recursive Systematic (Turbo) Codes", in *IEE Trans. of International Conference on Universal Personal Communications, San Diego*, Sept. 1994, pp. 183–187.

[12] T. A. Summers and S. G. Wilson, "SNR Mismatch and Online Estimation in Turbo Decoding", *IEEE Trans. on Communications*, vol. 46, no. 4, pp. 421–423, April 1998.

[15] A. G. Burr, G. P. White, "Performance of Turbo-coded OFDM" in *IEE Trans. of International Conference on Universal Personal Communications,* 1999.

[16} A. Burr, "Turbo Codes: The Ultimate Error Correction Codes?" Electronics & Communication Engineering Journal, pp. 155-165, August 2001

[17] Nguyen, V. D Kuchenbecker, H P. Interleaving algorithm for soft decision Viterbi decoding in OFDM systems over fading channels. IEEE International Conference on Telecommunication, June 2001, Romania