IMPLEMENTATION OF TURBO CODES

Submitted in partial fulfillment of the requirement for the degree

of

Bachelor of Technology

in

Electronics and Communication Engineering

under the supervision of

Dr. Davinder Singh Saini

By

Meghna Vaidya (081057)
Sweaty Chaudhary (081111)

to

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY

Jaypee University of Information and Technology

Waknaghat, Solan – 173234, Himachal Pradesh

CERTIFICATE

This is to certify that the project report entitled **"IMPLEMENTATION & PERFORMANCE ANALYSIS OF TURBO CODES"** submitted by **Meghna Vaidya** and **Sweaty Chaudhary** in partial fulfillment for the award of degree of Bachelor of Technology in Electronics and Communication Engineering to Jaypee University Of Information Technology, Waknaghat, Solan has been carried out under my supervision. This work has not been submitted partially or fully to any other University or Institute for the award of this or any other degree or diploma.

Date: 31.05.12

Dr. Davinder Singh Saini
Associate Professor
Deptt. of Electronics and
Communication Engineering

ACKNOWLEDGEMENT

Apart from efforts, the success of any project largely depends on the encouragement and the guidelines of many others. Therefore we take the opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We would like to show our appreciation to our project guide DR. D.S. SAINI without whose guidance, tremendous support and continuous motivation the project work should not be have been carried out this well. His kind behavior and motivation provided us the required courage to complete the project.

Special thanks to our project panel because it was their regular concern and appreciation that made this project carried out easily and satisfactory.

Date: May 31, 2012.

Meghna Vaidya(081057)

Sweaty Chaudhary(081111)

CONTENTS

Certificate ii

Acknowledgement iii

List of Figures vii

Abstract ix

CHAPTER 1 INTRODUCTION 1

1.1 Introduction 2

1.2 Thesis Outline 2

CHAPTER 2 WLAN TECHNOLOGIES & STANDARDS 4

2.1 Terms Used 5

2.2 Wireless LAN Technologies 5

2.2.1 Infrared (IR) 6

A Advantages 6

B Disadvantages 6

2.2.2 Narrow Band Technology 7

A Advantages 7

B Disadvantages 7

2.2.3 Spread Spectrum Technology 7

A Frequency Spread Spectrum Technology 8

B Direct Sequence Spread Spectrum Technology 8

2.2.4 How to Choose 802.n? 8

A IEEE 802.11 9

B IEEE 802.11a 9

C IEEE 802.11b 10

D IEEE 802.11g 10

2.3 Channel Conditions 11
2.3.1 Additive White Gaussian Noise 11
2.3.2 Multipath Fading Rayleigh Channel 11
2.3.3 Multipath Fading Rician Channel 12
2.4 Modulation Techniques 12
2.4.1 BPSK 12
2.4.2 QPSK 13

CHAPTER 3 TURBO CODES **15**
3.1 Turbo Codes 16
3.2 Channel Coding 17
3.3 Dawn of Turbo Codes 17
3.4 Why Turbo Codes? 18
3.5 Convolutional Codes 19
3.5.1 Parallel Concatenation Encoding With Interleaving 21

CHAPTER 4 TURBO ENCODER & DECODER **25**
4.1 Encoder for Turbo Codes 26
4.1.1 RSC Component Codes 26
4.1.2 Interleaving 29
4.1.3 Puncturing 30
4.1.4 Termination 32
4.2 Turbo Decoding 32
4.3 Work Description 36

CHAPTER 5 RESULTS **40**
5.1 Introduction 41

5.2 Simulation Model 41

5.3 Simulation Parameters 43

5.4 Simulation Results 43

Conclusion 48

Scope for Future Work 49

List of References 50

Appendix A 51

Appendix B 54

LIST OF FIGURES

2.1 Constellation Diagram for BPSK 13

2.2 Constellation Diagram for QPSK 14

3.1 Basic Structure of a Turbo Encoder 19

3.2 Non Systematic Convolutional Encoder 20

3.3 Recursive Systematic Convolutional Encoder 20

3.4 Generic Structure for Turbo Encoder 21

4.1 Turbo Encoder as presented by Berrou et al 26

4.2 Generic Structure of NSC Encoder 27

4.3 Generic Structure of RSC Encoder 27

4.4 Illustration of Pseudo Random Interleaving 30

4.5 Illustration of Puncturing

A Unpunctured Code 31

B Punctured Code 31

C Heavily Punctured Code 31

4.6 Generic Structure of Turbo Decoder 33

5.1 Block Diagram of Turbo Encoder 41

5.2 Block Diagram of Turbo Decoder 42

5.3 Block Diagram of System with Turbo Encoder and Decoder 42

5.4 Block Diagram of System with Viterbi Decoder 42

5.5 Performance Analysis of Turbo Decoder and Viterbi Decoder 44

5.6 BER versus SNR for Turbo Coded System for different number of Iterations 45

5.7 BER versus SNR curves for different modulation techniques for AWGN Channel conditions 46

5.8 BER versus SNR curves for different modulation techniques for Multipath Rayleigh Fading channel conditions 46

5.9 BER versus SNR curves for different modulation techniques for Multipath Rician Fading channel conditions 47

ABSTRACT

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.

The project is aimed at implementing and understanding the performance of Turbo Codes. First step towards the fulfillment of this project included a thorough ground work regarding the basic fundamentals of Turbo Codes. Once the basics of Turbo Codes were acquired, the next step included generating the Turbo Codes. For this Simulink was chosen as the preferred medium. A Turbo Encoder was constructed using Convolutional Encoders and Interleavers. Once the encoding part was completed, it was followed by the development of a Turbo Decoder- the decoding part generally is based on two approaches, the Viterbi decoding and a Posteriori Probability decoding. A Turbo Decoder was constructed using the A Posteriori Probability Decoder and Interleaver. The Decoders based on both these approaches were integrated into a whole communication system- the preferred modulation and demodulation scheme was Binary Phase Shift Keying and the channel was Additive White Gaussian Noise Channel. After the integration of all the blocks into a complete communication system was achieved, the errors resulting from both the approaches were recorded and the resulting performance was analyzed. BER versus SNR curves were plotted for the already implemented communication systems. The modulation techniques as well as channel conditions were changed and the performance characteristics were observed for Binary Phase Shift Keying and Quadrature Phase Shift Keying, and for different channel conditions such as Additive White Gaussian Noise Channel, Multipath Rayleigh Fading Channel and Multipath Rician Fading Channel. After the analysis part, future prospects and scope of Turbo Codes has been discussed.

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, Forward Error Correction Codes is the standard being used throughout the world to achieve the high data rates necessary for data intensive applications that must now become routine. In this thesis forward error correction is performed by using turbo codes. The combination of turbo coding and recursive decoding and certain modulation techniques like Orthogonal Frequency Division Multiplexing (OFDM) 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 BPSK AND QPSK System and analyzes the performance of this system under Additive White Gaussian Noise Channel, Rayleigh Multipath Fading and Rician Multipath Fading channel conditions. It is presented as follows:

Chapter 2 discusses the fundamentals of wireless communication- definition of wireless LAN, wireless LAN technologies, and wireless LAN standards in detail and various channel conditions and modulation techniques.

Chapter 3 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 codes and find out why they perform so well.

Chapter 4 discusses the various aspects of Turbo Encoding and Decoding as well as some of its advantages and functionality issues.

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

CHAPTER-2
WLAN STANDARDS & TECHNOLOGIES

2.1 Terms Used

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.

2.2 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 Hyper Lan is a family of standards in the 5.15-5.7GHz and 19.3 GHz frequency bands .

2.2.1 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 the1-2 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.

A Advantages

a) No government regulation controlling use.

b) Immunity To electromagnetic (EM) and RF interference.

B Disadvantages

a) Generally a short range technology (30-50 ft under ideal conditions.

b) Signals cannot penetrate solid objects.

c) Signal affected by light, snow, ice, fog.
d) Dirt can interfere with infrared.

2.2.2 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.

A Advantages

a) Longest range.
b) Low cost solution for large sites with low to medium data throughput requirements.

B Disadvantages

a) Large radio and antennas increase wireless client size.
b) RF site license required for protected bands.
c) No multivendor interoperability.
d) Low throughput and interference potential.

2.2.3 Spread Spectrum Technology

Most wireless LAN systems use spread-spectrum technology, a wideband radiofrequency 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.

A 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.

B 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.

2.2.4 How to choose 802.n?

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" (LANMAN 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 work group 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.
IEEE 802.11 standards

A IEEE 802.11

The original version of the standard IEEE 802.11 released in 1997 specifies two raw data rates of 1 and 2mega bits 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.10

B IEEE 802.11a

The 802.11a amendment to the original standard was ratified in 1999. The802.11a 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 gives802.11a 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 8times 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.

C 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.

D 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 about19 Mb/s net throughput (like 802.11a except with some additional legacy overhead). 802.11ghardware 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 802.11gnetwork.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 2Mb/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.

2.3 CHANNEL CONDITIONS

For performance analysis of Turbo codes, various channel conditions are considered, namely, Additive White Gaussian Noise, Multipath Fading Rayleigh and Multipath Fading Rician. The above mentioned three types of channels are described below.

2.3.1 ADDITIVE WHITE GAUSSIAN NOISE CHANNEL

In this channel model, linear addition of white noise is the only impairment to communication. The white noise has a constant power spectral density and a Gaussian distribution of amplitude. This model does not take into account frequency fading, frequency selectivity, nonlinearity, dispersion or interference.

The Additive White Gaussian Noise channel is a suitable model for deep space as well as satellite communication links. It is not a suitable model for terrestrial links because of various factors such as terrain blocking, multipath, interference etc. However, for terrestrial path modeling, this channel is used to simulate background noise of the channel, in addition to terrain blocking, ground clutter, interference, multipath and self interference that modern radio systems encounter in terrestrial operation.

2.3.2 MULTIPATH FADING RAYLEIGH CHANNEL

Rayleigh Fading is a statistical model used to study the effect of propogation environment on a radio signal, generally that used by wireless devices. Rayleigh Fading assumes that the magnitude of the signal, that has passed through a communication channel, will vary randomly or fade, following Rayleigh Distribution – the radial component of the sum of two uncorrelated Gaussian Random Variables.

This is a reasonable model for tropospheric as well as ionospheric signal propogation and the effect of heavily built up urban environments on radio signals. Rayleigh Fading is applicable when there is no dominant propogation along a line of sight between the transmitter and receiver. The requirement that there should be many scatterers present means that Rayleigh Fading is a very useful model in city centres where there is no line of sight between the transmitter and receiver due to heavy construction and there are many objects which reflect, refract and attenuate the signal.

In tropospheric and ionospheric signal propogation, the many particles in the atmospheric layers lead to scattering of the radio signals and this kind of environment approximates Rayleigh Fading.

2.3.3 MULTIPATH FADING RICIAN CHANNEL

Rician Fading is a model for anomaly in propogation of radio signals, caused by partial cancellation of a radio signal by itself – the signal arrives at the receiver by several different paths (thus leading to multipath interference), and at least one of the paths is changing (lengthening or shortening). Rician fading occurs when one of the paths, typically a line of sight signal, is much stronger than the others. In Rician Fading, the amplitude gain is characterized by a Rayleigh distribution.

2.4 MODULATION TECHNIQUES

The modulation techniques used are types of Phase Shift Keying(PSK). It is a digital modulation scheme that conveys data by changing, or modulating, the phase of a reference signal, also called carrier signal.

Phase Shift Keying uses a finite number of phases, each phase encodes an equal number of bits. Each pattern of bits forms the symbol that is represented by the particular phase. The demodulator, which is designed specifically for the symbol set used by the modulator, determines the phase of the received signal and maps it back tp the symbol it represents, thus recovering the original data. This requires the receiver to be able to compare the phase of the received signal to a reference signal – such a system is termed coherent.

2.4.1 BPSK

BPSK (also sometimes called PRK, Phase Reversal Keying, or 2PSK) is the simplest form of phase shift keying (PSK). It uses two phases which are separated by 180° and so can also be termed 2-PSK. It does not particularly matter exactly where the constellation points are positioned, and in this figure they are shown on the real axis, at 0° and 180°. This modulation is the most robust of all the PSKs since it takes the highest level of noise or distortion to make the demodulator reach an incorrect decision. It is, however, only able to modulate at 1 bit/symbol (as seen in the figure) and so is unsuitable for high data-rate applications.

In the presence of an arbitrary phase-shift introduced by the communications channel, the demodulator is unable to tell which constellation point is which. As a result, the data is often differentially encoded prior to modulation.

BPSK is functionally equivalent to 2-QAM modulation.

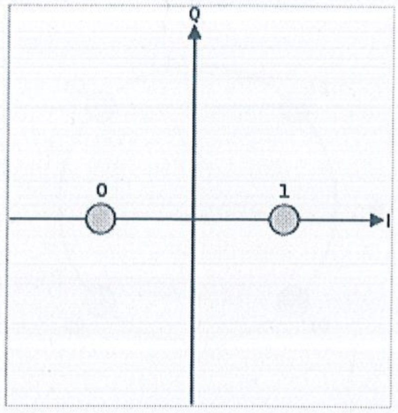

Figure 2.1: Constellation Diagram for BPSK

2.4.2 QPSK

Sometimes this is known as quaternary PSK, quadriphase PSK, 4-PSK, or 4-QAM. (Although the root concepts of QPSK and 4-QAM are different, the resulting modulated radio waves are exactly the same.) QPSK uses four points on the constellation diagram, equispaced around a circle. With four phases, QPSK can encode two bits per symbol, shown in the diagram with gray coding to minimize the bit error rate (BER) — sometimes misperceived as twice the BER of BPSK.

The mathematical analysis shows that QPSK can be used either to double the data rate compared with a BPSK system while maintaining the same bandwidth of the signal, or to maintain the data-rate of BPSK but halving the bandwidth needed. In this latter case, the BER of QPSK is exactly the same as the BER of BPSK - and deciding differently is a common confusion when considering or describing QPSK.

Given that radio communication channels are allocated by agencies such as the Federal Communication Commission giving a prescribed (maximum) bandwidth, the advantage of QPSK over BPSK becomes evident: QPSK transmits twice the data rate in a given bandwidth compared to BPSK - at the same BER. The engineering penalty that is paid is that QPSK transmitters and receivers are more complicated than the ones for BPSK. However, with modern electronics technology, the penalty in cost is very moderate.

As with BPSK, there are phase ambiguity problems at the receiving end, and differentially encoded QPSK is often used in practice.

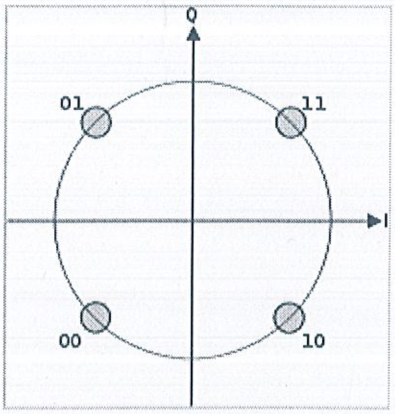

Figure 2.2: Constellation Diagram of QPSK

CHAPTER – 3

TURBO CODES

3.1 Turbo Codes

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 concepts 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.

3.2 Channel Coding

Forward-error-correcting (FEC) channel codes are commonly used to improve the energy efficiency of wireless communication systems. On the transmitter side, an FEC encoder adds redundancy to the data in the form of parity information. Then at the receiver, a FEC decoder is able to exploit the redundancy in such a way that a reasonable number of channel errors can be corrected. Because more channel errors can be tolerated with than without an FEC code, coded systems can afford to operate with a lower transmit power, transmit over longer distances, tolerate more interference, use smaller antennas, and transmit at a higher data rate.

A binary FEC encoder takes in k bits at a time and produces an output (or code word) of n bits, where n > k. While there are 2n possible sequences of n bits, only a small subset of them, 2k to be exact, will be valid code words. The ratio k/n is called the code rate and is denoted by r.

Lower rate codes, characterized by small values of r, can generally correct more channel errors than higher rate codes and are thus more energy efficient. However, higher rate codes are more bandwidth efficient than lower rate codes because the amount of overhead (in the form of parity bits) is lower. Thus the selection of the code rate involves a tradeoff between energy efficiency and bandwidth efficiency.

For every combination of code rate (r), code word length (n), modulation format, channel type, and received noise power, there is a theoretical lower limit on the amount of energy that must be expended to convey one bit of information. This limit is called the channel capacity or Shannon capacity, named after Claude Shannon, whose 1948 derivation of channel capacity .It is considered to have started the applied mathematical field that has come to be known as information theory.

Since the dawn of information theory, engineers and mathematicians have tried to construct codes that achieve performance close to Shannon capacity. Although each new generation of FEC code would perform incrementally closer to the Shannon capacity than the previous generation, as recently as the early 1990s the gap between theory and practice for binary modulation was still about 3 dB in the most benign channels, those dominated by additive white Gaussian noise (AWGN). In other words, the practical codes found in cell phones, satellite systems, and other applications required about twice as much energy (i.e., 3 dB more) as the theoretical minimum amount predicted by information theory. For fading channels, which are harsher than AWGN, this gap was even larger.

3.3 The Dawn of Turbo Codes

A major advancement in coding theory occurred in 1993, when a group of researchers working in France developed (or, in the parlance of coding theorists, "discovered") turbo codes. The initial results showed that turbo codes could achieve energy efficiencies within only a half decibel of the Shannon capacity. This was an extraordinary result that at first was met with skepticism. But once other researchers began to validate the results independently, a massive research effort was soon underway with the goal of explaining and, better yet, enhancing the remarkable performance of turbo codes. Much of this research focused on improving the practicality of turbo codes, which, as will be discussed shortly, have some peculiarities that make implementation less than straightforward. By the end of the

1990s, the virtues of turbo codes were well known, and they began to be adopted in various systems. Now they are incorporated into standards used by NASA for deep space communications (CCSDS), digital video broadcasting (DVB-T), and both third-generation cellular standards (UMTS and cdma2000).

3.4 Why Turbo Codes?

In order to understand why turbo codes work so well, one must first understand what makes for a good code in general. But first, two terms must be defined. A linear code is a code for which the modulo-2 sum of two valid code words (found by XOR-ing each bit position) is also a valid code word. Most codes, turbo codes included, are linear. The Hamming weight (also simply known as weight) of a code word is the number of ones that it contains. Thus a simple linear code could be composed of two code words, {000} and {111}, where the Hamming weight of the first code word is 0 and the Hamming weight of the second code word is 3. Note that all linear codes must contain the all-zeros codeword, since any code word XOR-ed with itself will produce all zeros. A "good" linear code is one that has mostly high-weight code words (except, of course, the mandatory all-zeros code word). High-weight code words are desirable because it means that they are more distinct, and thus the decoder will have an easier time distinguishing among them. While a few low-weight code words can be tolerated, the relative frequency of their occurrence should be minimized. One way to reduce the number of low-weight code words is by using a turbo encoder.

Since the weight of the turbo code word is simply the sum of the weights of the input and the parity outputs of the two constituent code words, we can allow one of these parity outputs to have low weight (as long as the other has high weight). Because the second encoder's input has been scrambled by the interleaver, its parity output is usually quite different from the first encoder's. Thus, although it is possible that one of the two encoders will occasionally produce a low-weight output, the probability that both encoders simultaneously produce a low-weight output is extremely small. This improvement is called the interleaver gain and is one of the main reasons that turbo codes perform so well. Coding theorists say that turbo codes have a "thin distance spectrum," that is the distance spectrum, which is a function that describes the number of code words of each possible nonzero weight (from 1 to n), is thin in the sense that there are not very many low-weight words present .

3.5 Convolutional Codes

Although almost any type of encoder could be used for the two constituent encoders shown in Figure 3.1 , in practice turbo codes almost always use recursive systematic convolutional (RSC) encoders. RSC codes are very similar in nature to conventional convolutional codes, also called nonsystematic convolutional codes (NSC), which are commonly used for 2G cellular systems, data modems, satellite communications, and many other applications. An example of an NSC encoder is shown in Figure 3.2.

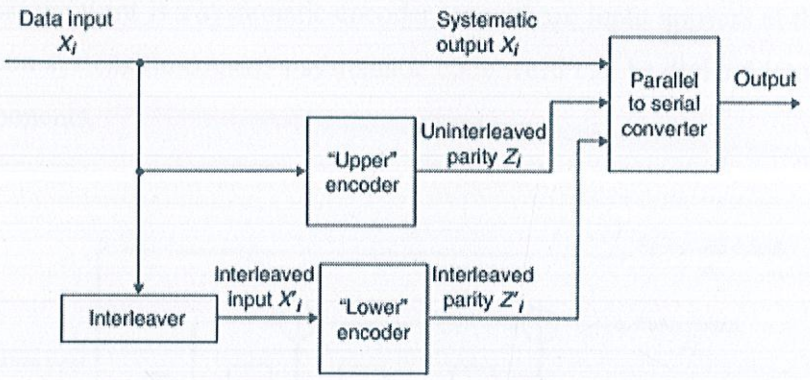

Figure 3.1: Basic Structure of a Turbo Encoder

Data {Xi} enter from the left and are stored in a linear shift register (D denotes a D flip-flop). Each time a new data bit arrives, the data is shifted to the right into the next flip-flop. Each of the two output bits {Z1,i, Z2,i} is computed by XOR-ing a particular subset of the three bits stored in the shift register with the bit at the encoder's input. Because there are two output bits for each input bit, this is a rate r = 1/2 encoder. The constraint length, denoted by K, is the maximum number of input bits (past and current) that either output can depend on. In this case, since each output depends on up to four bits (the three in the shift register plus the one at the input), the constraint length is K = 4. One problem with the encoder shown in Figure 3.2 is that it is nonsystematic; that is, the encoder's input bits do not appear at its output. Thus, the code word contains only parity bits and therefore cannot be divided into the separate data and parity fields desired by the turbo encoder.

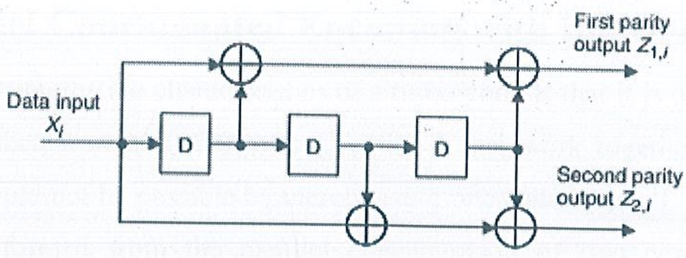

Figure 3.2: Non Systematic Convolutional Encoder

Instead, what we want is a systematic encoder, one whose input appears at the output. Unlike its non systematic cousin, a systematic code word can be divided into data and parity components.

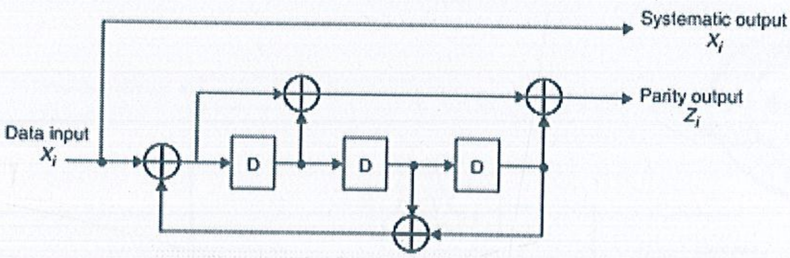

Figure 3.3: Recursive Systematic Convolutional Encoder

As shown in Figure 3.3 ,a recursive systematic convolutional code can be created from an NSC by simply feeding one of the two parity outputs back to the input (it is this feedback that makes it recursive). Since one of the parity outputs is fed into the input, only the other parity output needs to be transmitted. This allows the data input (also called the systematic output) to be transmitted along with the parity output that was not fed back, while still maintaining a code rate of r = 1/2. Also, it turns out that the feedback within the encoder is necessary for the turbo encoder to obtain the maximum interleaver gain. Note that, as indicated by Figure 3.1, only the parity outputs of the two RSC encoders are actually transmitted. The systematic outputs are not needed because they are identical to each other (although ordered differently) and to the turbo code input (which becomes the systematic part of the overall turbo code word).

3.5.1 Parallel Concatenated Encoding with Interleaving

One of the most interesting characteristics of a turbo code is that it is not just a single code. It is, in fact, a combination of two codes 1 that work together to achieve a synergy that would not be possible by merely using one code by itself. In particular, a turbo code is formed from the parallel concatenation of two constituent codes separated by an interleaver. Each constituent code may be any type of FEC code used for conventional data communications. Although the two constituent encoders may be different, in practice they are normally identical. A generic structure for generating turbo codes is shown in Figure 3.4 As can be seen, the turbo code consists of two identical constituent encoders, denoted as ENC #1 and ENC #2. The input data stream and the parity outputs of the two parallel encoders are then serialized into a single turbo code word.

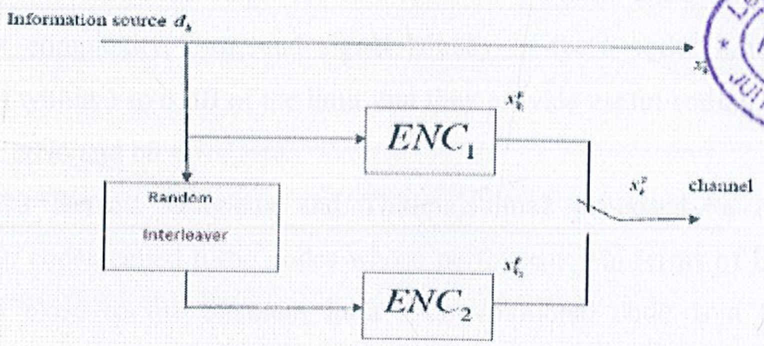

Figure 3.4: Generic Structure for Turbo Encoder

The interleaver is a critical part of the turbo code. It is a simple device that rearranges the order of the data bits in a prescribed, but irregular, manner. Although the same set of data bits is present at the output of the interleaver, the order of these bits has been changed, much like a shuffled deck of cards (although each input word is shuffled in exactly the same way). Without the interleaver, the two constituent encoders would receive the data in the exact same order and thus—assuming identical constituent encoders—their outputs would be the same. This would not make for a very interesting (or powerful) code. However, by using an interleaver, the data {Xi} is rearranged so that the second encoder receives it in a different order, denoted {Xi}.

Thus, the output of the second encoder will almost surely be different than the output of the first encoder—except in the rare case that the data looks exactly the same after it passes through the interleaver. Note that the interleaver used by a turbo code is quite different than the rectangular interleavers that are commonly used in wireless systems to help break up deep fades. While a rectangular channel interleaver tries to space the data out according to a regular pattern, a turbo code interleaver tries to randomize the ordering of the data in an irregular manner.

Shannon Theorem states that "Theoretical possibility of transmitting information nearly without error at any rate below a limiting rate, C." Noisy channel with Channel Capacity C and information transmitted at the rate R. If R < C; There exist codes that allow the probability of error at the receiver to be made arbitrarily small. However, error correcting channel codes must exist which allow maximum capacity to be achieved.

Many well-considered channel codes inch towards the Shannon limit, but all contenders have required large block lengths to perform close to the limit. The consequent complexity, cost, and signal latency of these codes have made them impractical within 3 to 5 dB of the limit, but they provide useful coding gain at higher values of Eb/No and bit error rate.

In 1993 Berrou, Glavieux and Thitimajshima2 proposed "a new class of convolution codes called turbo codes whose performance in terms of Bit Error Rate (BER) are close to the Shannon limit". Convolutional code is a type of error correcting code: Where m bit information to be encoded is transformed into an n bit symbol, where m/n is the code rate and m>n. Transformation is a function of k symbols, where k is the constraint length.

The principle of turbo code is theoretically possible to approach the Shannon limit by using a block code with large block length or a convolutional code with a large constraint length. The processing power required to decode such long codes makes this approach impractical. Turbo codes overcome this limitation by using recursive coders and iterative soft decoders. The recursive coder makes convolutional codes with short constraint length appear to be block codes with a large block length, and the iterative soft decoder progressively improves the estimate of the received message.

The theory of error correcting codes has presented a large number of code constructions with corresponding decoding algorithms. However, for applications where very strong error correcting capabilities are required these constructions all result in far too complex decoder solutions. The way to combat this is to use concatenated coding, where two (or more) constituent codes are used after each other or in parallel - usually with some kind of interleaving. The constituent codes are decoded with their respective decoders, but the final decoded result is usually sub-optimal. This means that better results might be achieved with a more complicated decoding algorithm - like the brute-force trying of all possible codewords. However, concatenated coding offers a nice trade of between error correcting capabilities and decoder complexity.

The idea of concatenated coding fits well with Shannon's channel coding theorem, stating that as long as we stay on the right side of the channel capacity we can correct everything - if the code is long enough. This also means that if the code is very long, it does not have to be optimal. The length in itself gives good error correcting capabilities, and concatenated coding is just a way of constructing - and especially decoding - very long codes.

Turbo codes are a class of convolution code which exhibit the properties of large block codes through the use of recursive coders. Coder performance is heavily dependent on the design of the interleaver, which must ensure adequate weight for at least one of the codes. The coding gain depends on the number of iterations; typically 5 to 10 iterations generate most of the improvement.

Turbo codes play a major role in the error channel coding scheme used in wireless cellular networks. Turbo codes emerged in 1993 and have since become a popular area of communications research. It is due to their exceptional performance that turbo codes are being accepted as 3GPP standard in personal communications. In next era of wireless communications, mainly the 4G applications, there is a need to provide the best QOS (Quality of Service) provisioning. The 4G cellular systems are aimed at supporting various applications like voice, data and multimedia over packet switched networks. Different applications have varied QOS requirements in terms of the data rate, bit error rate (BER), frame size and the packet error rate.

For certain type of transmission like text transmission, the packet loss is intolerable while delay is acceptable. But for real time video, there can be an acceptable degradation in the video, but delay in the system cannot be accepted..

Turbo codes is the most adaptable error coding scheme used to adapt to the varying QOS requirement .Hence there is a need to analyze the performance of Turbo codes by varying all the parameters which can be made adaptable. Based on the analysis, the most suitable parameters are chosen . In this project we analyze the behavior of Turbo codes for various interleaver size and structure. The simulation is done for the various noise levels and the graph is plotted against the bit error rate and the signal to noise ratio.

There are various applications of turbo codes due to its many important advantages. Turbo codes are used extensively in 3G AND 4G mobile telephony standards and in satellite communication also. NASA missions such as Mars Reconnaissance Orbiter are using turbo codes. They are also used in IEEE 802.16 (WiMAX).

CHAPTER - 4

TURBO ENCODER AND DECODER

4.1 Encoders for Turbo Codes

The encoder for a turbo code is a parallel concatenated convolutional code. Figure 4.1 shows a block diagram of the encoder first presented by Berrou.

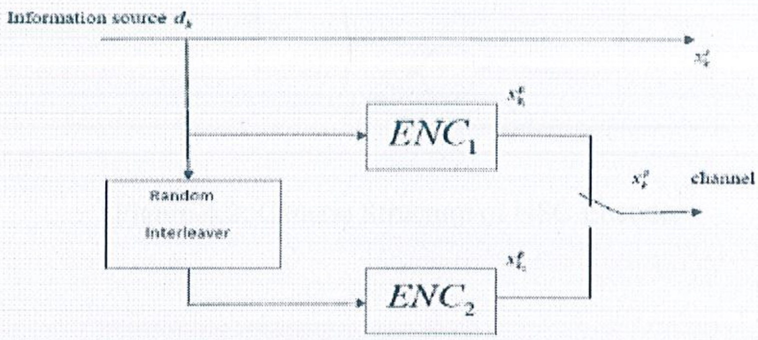

Figure 4.1: Turbo Encoder as presented by Berrou et al

The binary input data sequence is represented by $d_k = (d_1,\ldots,d_n)$ the input sequence is passed into the input of a convolutional encoder , and ENC_1 a coded bit stream, $x_{k1}{}^{p}$ 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, and a second coded bit stream ENC_2, $x_{k2}{}^{p}$ 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 $x_{k1}{}^{p}$ and the second encoder $x_{k2}{}^{p}$.

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 4.3 along with a non-systematic (NSC) encoder in Figure 4.2, for comparison.

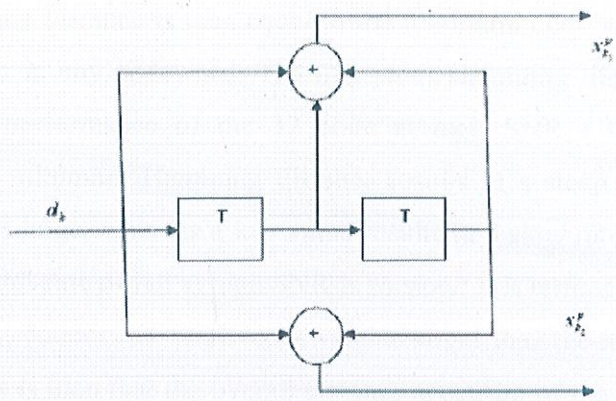

Figure 4.2: Generic Structure of NSC Encoder

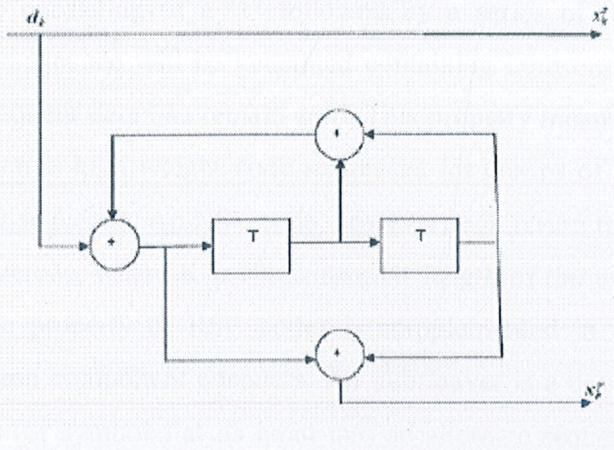

Figure 4.3 : Generic Structure of RSC Encoder

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 code words 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 32 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 codeweights larger than the minimum value must be considered. It is then that the overall distance spectrum of the code becomes important.

This is the relationship between the code weight and the number of code words 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 code words with higher code weights most of the time. This results in there being fewer code words 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 code words 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.

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 4.3 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.

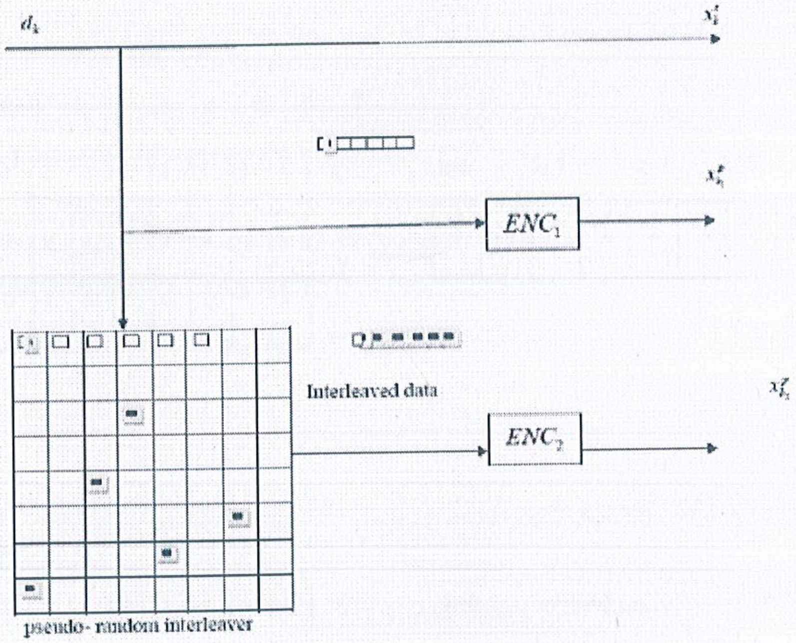

Figure 4.4: Illustration of Pseudo Random Interleaving

4.1.3 Puncturing

Different code rates are achieved by puncturing the parity bit sequences $x_{k1}{}^{p}$ and $x_{k2}{}^{p}$. Puncturing the data bit sequence $x_k{}^{s}$ leads to a severe degradation in turbo code performance. Figure 4.4 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 .

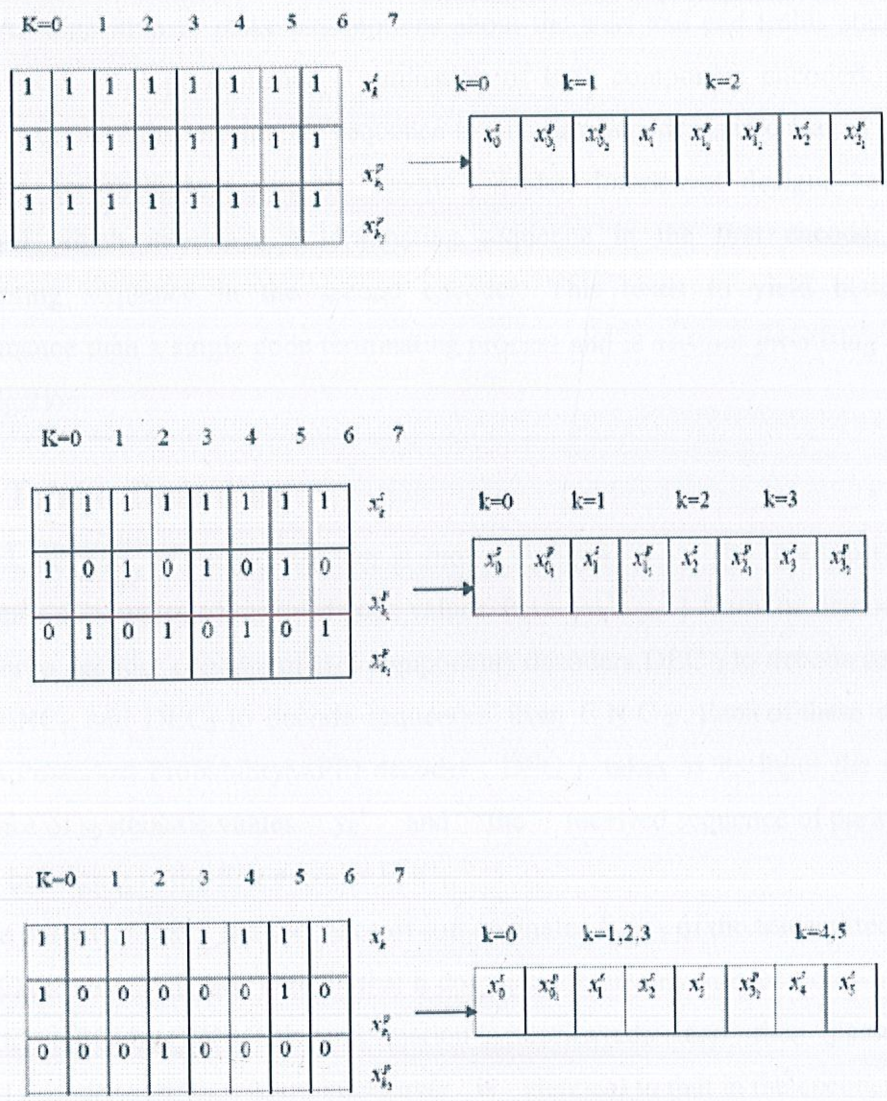

Figure 4.5: Illustration of Puncturing (a) Unpunctured Code (b) Punctured Code (c) Heavily Punctured Code

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 designed 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.

4.2 Turbo Decoding

A block diagram of a turbo decoder is shown in Figure 4.5 . 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 components decoders DEC_1 to decode sequences from ENC_1, and DEC_2 to decode sequences from ENC_2. Each of these decoders is a A Posteriori Probability(APP) decoder . DEC_1 takes as its input the received sequence of systematic values y_k^s and the received sequence of parity values y_{k1}^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 . DEC_2 takes as its input the (interleaved) systematic received values y_k^s and the sequence of parity , y_k^p from the second encoder ENC_2 , along with the interleaved form of the extrinsic information EXT_1, DEC_2 provided by the first decoder outputs a set of values, which when de-interleaved using an inverse form of the interleaver, constitute soft estimate EXT_2 of the transmitted data sequence d_k .

This extrinsicdata 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 d_k 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 and 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 and mayei ther 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.

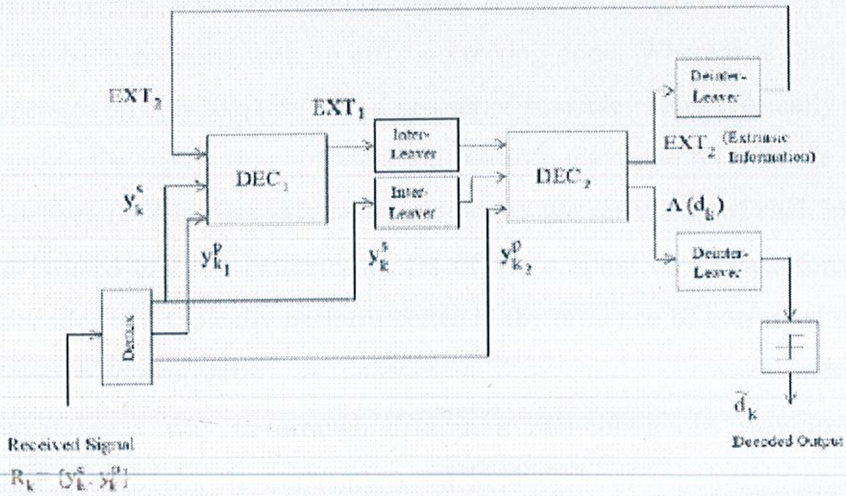

Figure 4.6: Generic Structure of Turbo Decoder

Hard decision decoding takes a stream of bits say from the 'threshold detector' stage of a receiver where each bit is considered definitely one or zero. Eg. for binary signaling, received pulses are sampled and the resulting voltages are compared with a single threshold. If a voltage is greater than the threshold it is considered to be definitely a 'one' say regardless of how close it is to the threshold. If its less, its definitely zero.

Soft decision decoding requires a stream of 'soft bits' where we get not only the 1 or 0 decision but also an indication of how certain we are that the decision is correct. One way of implementing this would be to make the threshold detector generate instead of 0 or 1, say: 000 (definitely 0),001 (probably 0),010 (maybe 0),011 (guess 0), 100 (guess 1), 101 (maybe 1),110 (probably 1) ,111(definitely 1).

One may call the last two bits 'confidence' bits. This is easy to do with eight voltage thresholds rather than one. This helps when we anticipate errors and have some 'forward error correction' coding built into the transmission. The Viterbi algorithm can take these 'soft bit' words and compute distances etc. as easily as it deals with hard bits. No great additional complexity apart from dealing with words (in this example 3-bit words) rather than one bit words. But the decisions are likely to be much much better with the greater reliability being placed on bits we are certain about than on but we are more uncertain about.

In information theory, a soft-decision decoding is a class of algorithm used to decode data that has been encoded with an error correcting code. Whereas a hard decision decoder operates on data that take on a fixed set of possible values (typically 0 or 1 in a binary code), the inputs to a soft-decision decoder may take on a whole range of values in-between. This extra information indicates the reliability of each input data point, and is used to form better estimates of the original data. Therefore, a soft-decision decoder will typically perform better in the presence of corrupted data than its hard-decision counterpart.

Interleaving is a way to arrange data in a non-contiguous way to increase performance. Interleaving is frequently used in digital communication and storage systems to improve the performance of forward error correcting codes. Many communication channels are not memoryless: errors typically occur in bursts rather than independently. If the number of errors within a code word exceeds the error-correcting code's capability, it fails to recover the original code word. Interleaving ameliorates this problem by shuffling source symbols across several code words, thereby creating a more uniform distribution of errors.

The analysis of modern iterated codes, like turbo codes and, typically assumes an independent distribution of errors. Systems using turbo codes therefore typically employ additional interleaving across the symbols within a code word.

For turbo codes, an interleaver is an integral component and its proper design is crucial for good performance. The iterative decoding algorithm works best when there are not short cycles in the factor graph that represents the decoder; the interleaver is chosen to avoid short cycles.

Interleaver designs include: Rectangular (or uniform) interleavers ,diagonal interleavers,Random interleavers (where the interleaver is a known random permutation). The random interleaver uses a fixed random permutation and maps the input sequence according to the permutation order.S-random interleaver (where the interleaver is a known random permutation with the constraint that no input symbols within distance S appear within a distance of S in the output). The block interleaver is the most commonly used interleaver in communication system. It writes in column wise from top to bottom and left to right and reads out row wise from left to right and top to bottom.

The choice of interleaver size for varying SNRs also play a major role in theperformance of Turbo codes.To reduce the error floor that occurs in Turbo codes,the interleaver size is increased. The interleaver can reduce the error coefficients of low weight codewords through a process called Spectral Thinning.This distance spectrum results in a reduced bit error probability by a factor 1/N which is the interleaving gain.N is the size of interleaver . At low SNR's , the interleaver size is the only important factor ,as the code performance is dominated by the interleaver gain.The effects induced by changing the interleaver structure at low SNR region are not significant. At high SNRs , the interleaver structure determines the code performance The interleaver structure affects the mapping of low weight input sequences to the interleaver output,and hence the first several distance spectral lines of the turbo code distance spectrum.This plays an important role in determining the code performance at high SNRs.

The Viterbi algorithm is a dynamic programming algorithm for finding the most likely sequence of hidden states – called the Viterbi path – that results in a sequence of observed events, especially in the context of Markov information sources, and more generally, hidden Markov models. The forward algorithm is a closely related algorithm for computing the probability of a sequence of observed events. These algorithms belong to the realm of information theory.

A Viterbi decoder uses the Viterbi algorithm for decoding a bitstream that has been encoded using forward error correction based on a convolutional code.There are other algorithms for decoding a convolutionally encoded stream (for example, the Fano algorithm). The Viterbi algorithm is the most resource-consuming, but it does the maximum likelihood decoding. It is most often used for decoding convolutional codes with constraint lengths $k<=10$, but values up to $k=15$ are used in practice.

The algorithm makes a number of assumptions. First, both the observed events and hidden events must be in a sequence. This sequence often corresponds to time. Second, these two sequences need to be aligned, and an instance of an observed event needs to correspond to exactly one instance of a hidden event. Third, computing the most likely hidden sequence up to a certain point t must depend only on the observed event at point t, and the most likely sequence at point $t-1$.These assumptions are all satisfied in a first-order hidden Markov model.

The terms "Viterbi path" and "Viterbi algorithm" are also applied to related dynamic programming algorithms that discover the single most likely explanation for an observation. For example, in statistical parsing a dynamic programming algorithm can be used to discover the single most likely context-free derivation (parse) of a string, which is sometimes called the "Viterbi parse".

It is now also commonly used in speech recognition, keyword spotting, computational linguistics, and bioinformatics. For example, in speech-to-text (speech recognition), the acoustic signal is treated as the observed sequence of events, and a string of text is considered to be the "hidden cause" of the acoustic signal. The Viterbi algorithm finds the most likely string of text given the acoustic signal.

4.3 WORK DESCRIPTION

Once the basic ideas regarding turbo codes were acquired, the implementation of turbo encoder and decoder followed. Generally, maximum turbo encoder designs follow these ideas-

- The two encoders used are normally identical;

- The code is in a systematic form, that is the input bits also occur at the output.
- The interleaver reads the bits in a pseudo-random order.

To implement a turbo decoder in matlab, we make use of the convolutional encoder block. The polynomial description of the convolutional encoder describes the connections among shift registers and modulo 2 adders. This description has three components(for a feedback type encoder)-

- Constraint length
- Generator polynomial
- Feedback connection polynomial

The value of constraint length is one plus the number of shift registers for that input. It forms a vector whose length is number of inputs in the encoder diagram.
If the encoder has a inputs and b outputs, the code generator matrix is an a-by-b matrix. To build this matrix, place a 1 in each spot where a connection line from the shift register feeds into the adder and a 0 elsewhere. Convert this binary representation into an octal representation.

For a feedback encoder, we have a vector of feedback connection polynomials. The length of this vector is the number of inputs in the encoder. The elements of this vector indicate the feedback connection for each input.

The poly2trellis function accepts a polynomial description of a convolutional encoder and returns the corresponding trellis structure description- indicating the number of input and output symbols.

The output of the first convolutional encoder is fed in a random interleaver which rearranges the elements of the input vector using a random permutation. The number of elements parameter indicates how many numbers are present in the input vector. The initial seed parameter initializes the random number generator that the block uses to determine the permutation. The output of this random interleaver is fed to an identical convolutional encoder, as mentioned above. The pairing block for a turbo encoder, that is a turbo decoder is also constructed on the same platform. The input is fed to a zero order hold- it samples and holds the input for a specified time period. If the input is a vector, all elements of the vector are held for the same sample period.

The output of the zero order hold is fed to an APP decoder which performs the a posteriori probability decoding of a convolutional code. The inputs represent the sequence of log-likelihoods of encoder input bits as well as code bits. The outputs are updated versions of these sequences. To define the turbo or convolutional encoder that produced the coded input, we use the poly2trellis function- which takes up the details of constraint length, generator matrix and feedback connections. To indicate how the encoder treats the trellis at the beginning and end of each frame, we set the termination method parameter to either truncated or terminated. The truncated option indicates that the encoder resets to the all-zeroes state at the beginning of each frame. The terminated option indicates that the encoder forces the trellis to end each frame in the all-zeros state.

The decoding algorithm can be controlled using the algorithm parameter. APP implements the a posteriori probability decoding. To gain speed, one can use max or max*. Max* option also enables the scaling bits. This is used to avoid losing precision during the computations.

The updated sequence of log-likelihoods of encoder bits is fed to a random deinterleaver which has the same parameters as the interleaver. The output of this interleaver is fed to the next APP decoder along with a constant signal, which is identical to the above mentioned decoder. This is further fed to a hard decision mask, which converts the likelihoods of the channel bits into binary. The other output of likelihoods of the code bits is fed to a random interleaver(having identical parameters as the deinterleaver) and a delay unit; which is ultimately fed to the first decoder.

The binary output is fed to a zero order hold and stored as the output. Once the turbo encoder and decoder are implemented, the next step is to integrate them into a complete communication system. Input to the system is given using a Bernoulli Binary Generator that generates random binary numbers using a Bernoulli distribution. This distribution with parameter p produces zero with probability p and one with probability 1-p. The output of this generator is fed to the already constructed turbo decoder. Once the input is coded, it is modulated using binary phase shift keying and is transmitted over an additive white Gaussian channel. This channel block adds white Gaussian noise to a real or complex input signal. The variance of the noise generated by the AWGN block uses the ratio of bit energy to noise power spectral density, number of bits per symbol, input signal power and symbol period.

Once the transmitted signal reaches the receiver, it is demodulated by the binary phase shift key demodulator. The demodulated signal is fed to the turbo decoder, which has been implemented already. The Bernoulli random generator output and both the outputs of the APP decoders are compared to calculate the error rate.

The performance of the Viterbi decoder is also compared to the Turbo decoder. The same communication system is created- with the difference of having a convolutional encoder instead of a turbo encoder and a viterbi decoder instead of a turbo decoder. The input and output are compared to calculate the error rate, as above.

CHAPTER 5

RESULTS AND CONCLUSIONS

5.1 INTRODUCTION

A Turbo encoded system was modeled using Matlab Simulink to allow the generation of Turbo Codes and various parameters of the system to be varied and tested. The aim of doing the simulations was to measure the performance of Turbo Codes under Additive White Gaussian Noise, Rayleigh Multipath Fading, Rician Multipath Fading Channel conditions, for different modulation schemes like Binary Phase Shift Keying and Quadrature Phase Shift Keying used in IEEE 802.11a wireless LAN Standard.

Following this introduction, section 5.2 discusses the model of the communication system used in simulation, steps in Turbo Code Simulation, modulation techniques and different channel conditions. Section 5.3 presents the parameters used in simulation. Section 5.4 provides the simulation results of Turbo coded system for different modulation schemes and channel conditions. It also shows the results to compare the performance of system using Turbo decoder and Viterbi decoder.

5.2 SIMULATION MODEL

Since the main goal of this thesis was to simulate the communication system by utilizing Turbo codes. The block diagram of the entire system is shown in figures 5.1 to 5.4.

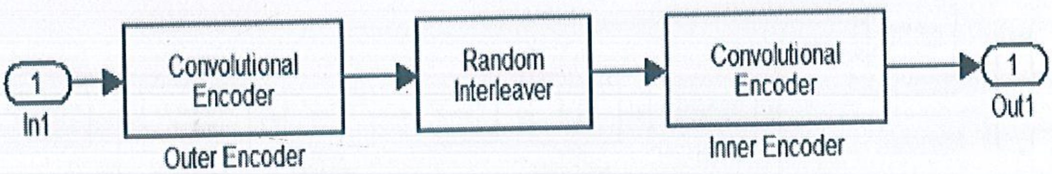

Figure 5.1: Block Diagram of Turbo Encoder

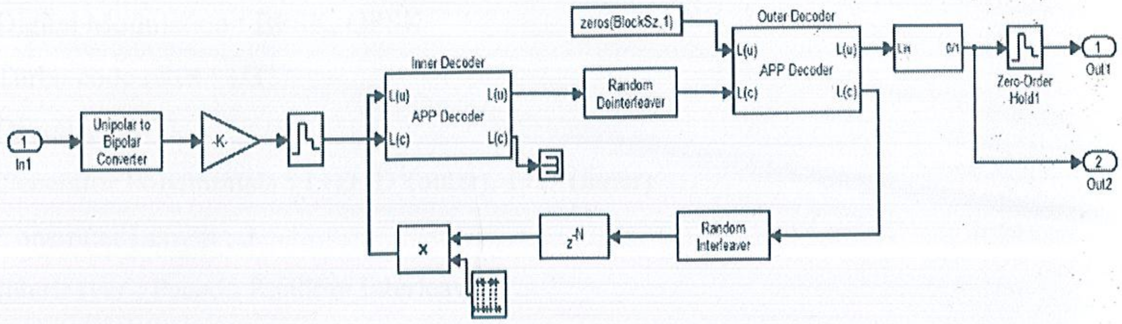

Figure 5.2: Block Diagram of Turbo Decoder

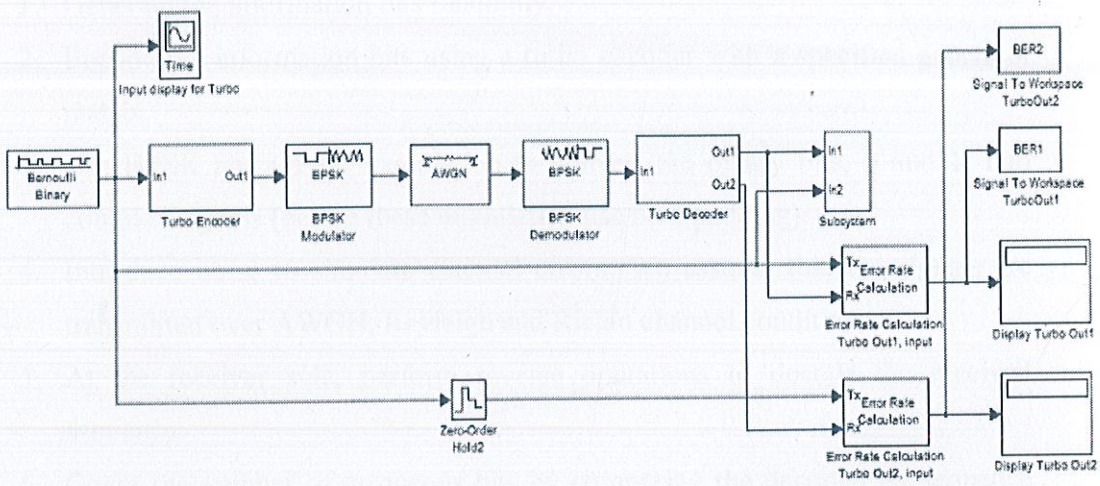

Figure 5.3: Block Diagram of System with Turbo Encoder and Decoder

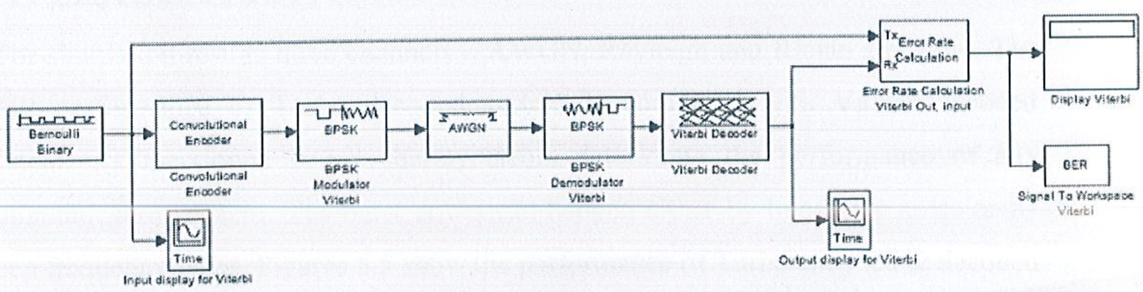

Figure 5.4: Block Diagram of System with Viterbi Decoder

5.3 SIMULATION PARAMETERS

Digital Modulation : BPSK, QPSK

Turbo code rates : 1/3

Decoder : APP

Generator Polynomials : $1+D+D^2$(outer), $1+D^2$(inner)

Constraint Length : 3

Interleaver : Pseudo Random Interleaver

We measured the performance of Turbo coded BPSK and QPSK for different channel conditions 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 a specified generator matrix.
3. Use BPSK and QPSK modulation to convert the binary bits, 0 and 1, into complex signals (before these modulation use zero padding).
4. Introduce noise to simulate channel errors. We assume that the signals are transmitted over AWGN, Rayleigh and Rician channel conditions.
5. At the receiver side, perform reverse operations to decode the received sequence.
6. Count the number of erroneous bits by comparing the decoded bit sequence with the original one.
7. Calculate the BER and plot it.

5.4 SIMULATION RESULTS

For simulation results three channels – AWGN, Rayleigh and Rician are used. The BER performance of Turbo decoded system is compared with Viterbi decoded system. As mentioned before, bursty errors deteriorate the performance of any communication system. The burst errors can happen either by impulsive noise or by deep frequency fades. Figures 5.5 show the performance of Turbo encoded & decoded and Convolution encoded & Viterbi decoded system with AWGN channel conditions; in terms of error rates with respect of number of symbols compared. As seen in Figure 5.5 the performance of system with Turbo Decoder is significantly better than the performance of system with Viterbi Decoder.

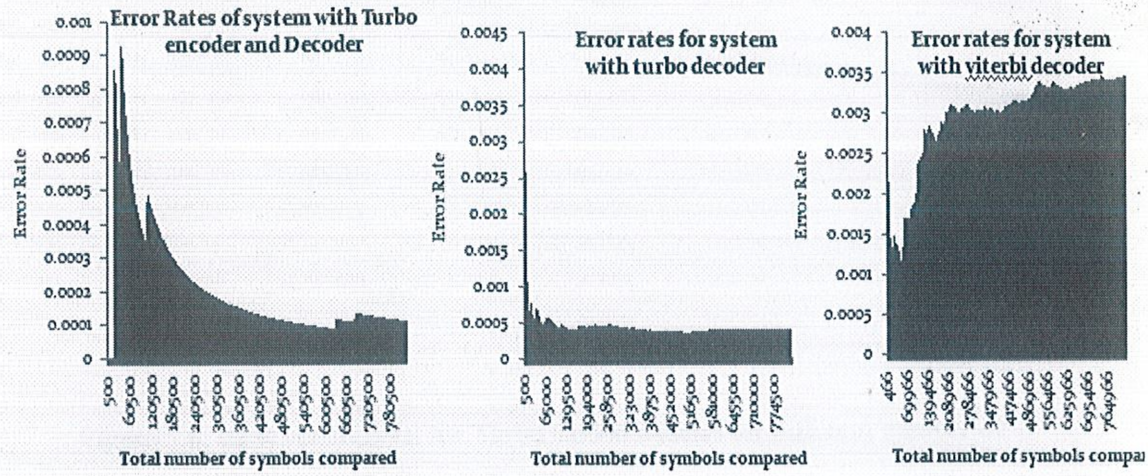

Figure 5.5: Performance Analysis of Turbo Decoder with Viterbi Decoder

To improve the performance of any communication system Forward Error Correction(FEC) can be used. Convolution code is a good example of FEC code. Convolution coding in a communication system can give performance improvement of some 5 db on AWGN channel over the uncoded system at required BER. Further improvement in the performance can be obtained by applying turbo coding instead of convolutional code. Turbo codes give better performance at low SNR.

The performance of Turbo Codes also varies with the number of Iterations. The trends followed by Turbo Codes upon changing the number of iterations and its result on the BER versus SNR plots are shown in Figure 5.6. As seen in the figure, the performance of the system increases with the increase in number of iterations, as the Bit Error Rate reduces significantly.

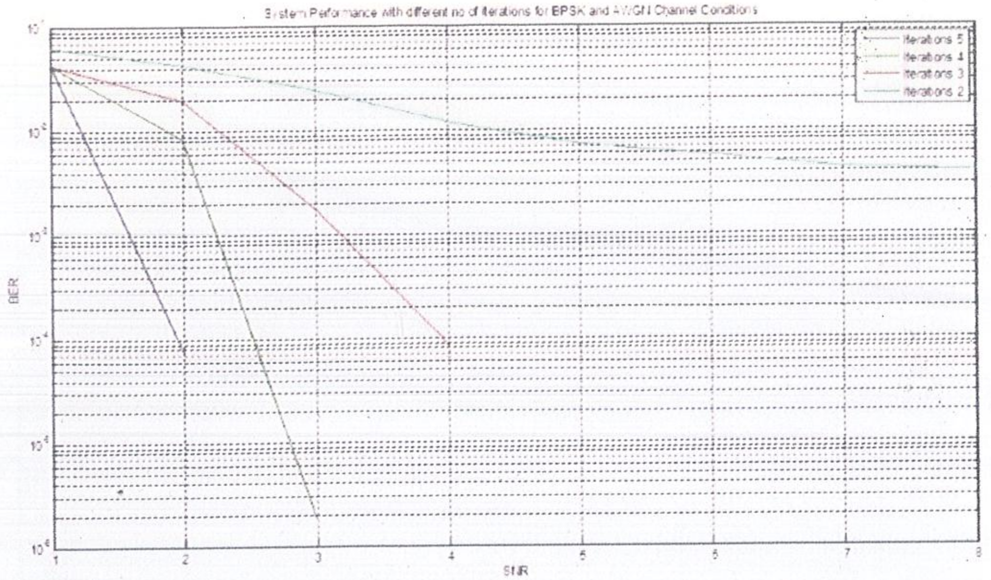

Figure 5.6: BER versus SNR for Turbo Coded System for different number of Iterations

The BER performance of Turbo coded BPSK and QPSK system is compared is compared under the fading AWGN channel, Rayleigh multipath fading channel and Rician multipath 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 $1+D+D^2$ AND $1+D^2$ which are iteratively decoded by APP(A Posteriori Probability) for a number of decoding iterations. The simulated results are shown from figure 5.7 to figure 5.9. From the results, we observe that turbo codes 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. As evident from the figures, BPSK outperforms QPSK in all the three channel conditions. And the performance of system is better with Rayleigh and Rician channel conditions as compared to AWGN channel.

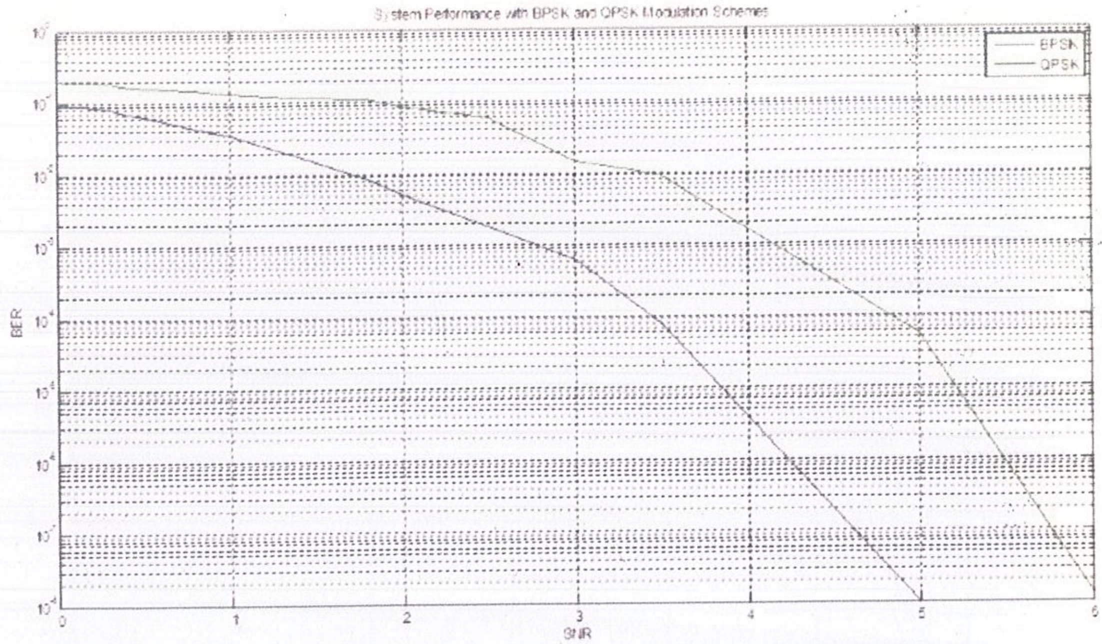

Figure 5.7: BER versus SNR curves for different modulation techniques for AWGN channel conditions

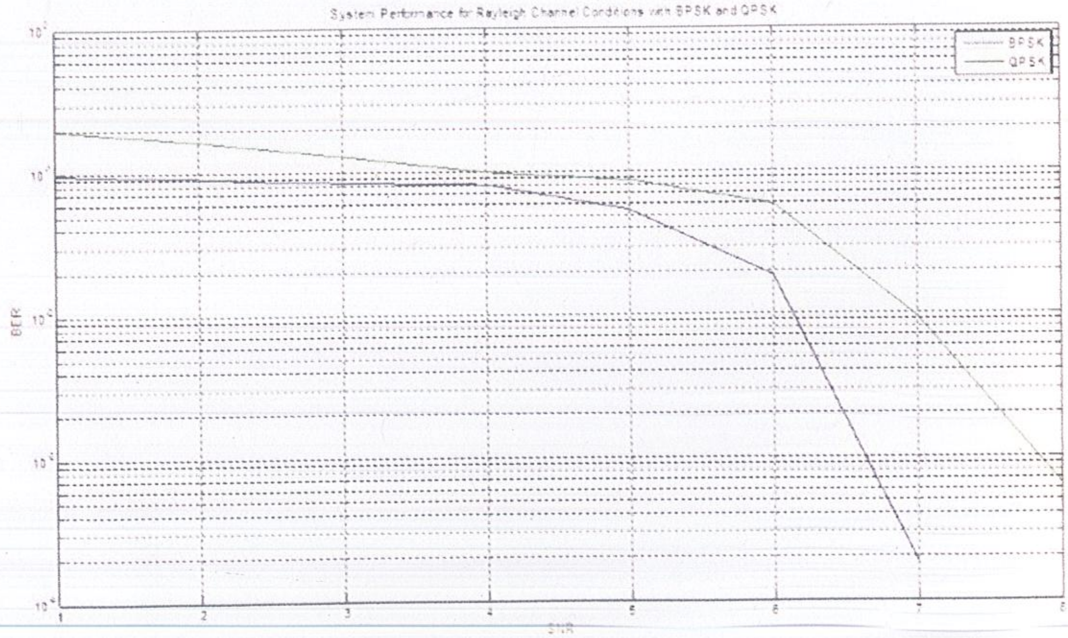

Figure 5.8: BER versus SNR curves for different modulation techniques for Multipath Rayleigh Fading

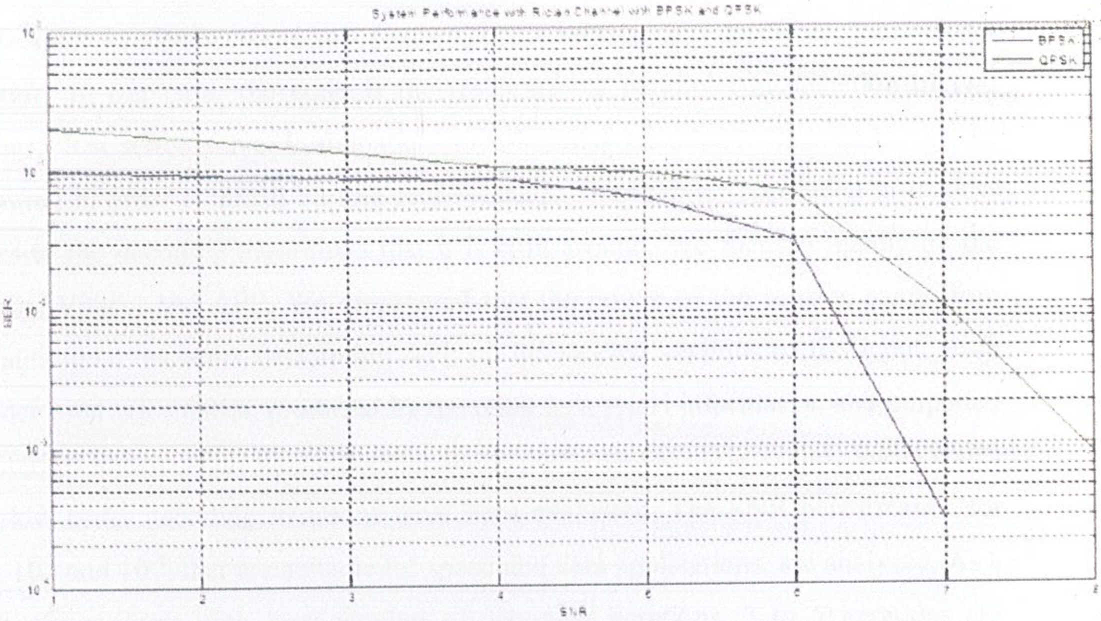

Figure 5.9: BER versus SNR curves for different modulation techniques for Multipath Rician Fading

CONCLUSION

To conclude, this major project gives the detail knowledge of a current key issue in the field of communications named Turbo Codes. We focused our attention on Turbo codes- their implementation and performance analysis. We described the encoder structure. In our case, the code is the result of the parallel concatenation of two identical RSCs(Recursive Systematic Convolutional encoders). 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 Viterbi and APP. We discovered that the power of the scheme came from two individual decoders performing APP 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 of the codes. 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 systems with least number of decoding iterations, 3 to 5 iterations are shown to provide good BER performance. The systems having BPSK modulation scheme are shown to have outperformed the systems having QPSK modulation scheme for all the three channel conditions- AWGN, Rayleigh and Rician. Also, systems have better performance with Rayleigh and Rician channel conditions as compared to AWGN channel conditions.

FUTURE SCOPE

Future telecommunication systems are moving towards higher data rates but simultaneously keeping in mind the complexity issues. Turbo codes have abundant scope in deep space as well as satellite communication. Other areas of scope are VLSI implementation of turbo codes as well as multipath channel investigation.

LIST OF REFERENCES

[1] Kulat K, Varade S, "Performance Analysis of MIMO-OFDM System using Space Time Turbo Codes and Adaptive Beamforming", International Journal of Computer Applications, Volume 23-No. 3, May 2011.

[2] Duman T, Stefanov A, "Turbo Coded Modulation for Systems with Transmit and Receive Antenna Diversity over Block Fading Channels: System Model, Decoding Approaches, and Practical Considerations", IEEE Journal on Selected Areas in Communication, Volume 19- No. 5, May 2001.

[3] Oberg M, Siegel P, "Lowering the Error Floor for Turbo Codes", University of California, San Diego, 1997.

[4] Shannon C, "Communication in the presence of Noise", IEEE, Volume 86-No. 2, Feb 1998.

[5] Berrou C, Glavieux A, Thitimajshima P., "Near Shannon Limit Error- Correcting Coding and Decoding: Turbo Codes", IEEE, 1993.

[6] Berrou C, Adde P, Angui E, Faudeil S, "A Low Complexity, Soft Output, Viterbi Decoder Architecture", IEEE, May 1993.

[7] Wang C, "On the Performance of Turbo Codes", IEEE, 1998.

[8] Yuan J, Vucetic B, Feng W, "Combined Turbo Codes and Interleaver Design", IEEE, 1998.

[9] Jung P, Nasshan M, "Comprehensive Comparison of Turbo Code Decoders", IEEE, 1995.

[10] Costello D, Takeshita O, "New Deterministic Interleaver Designs for Turbo Codes", IEEE Transactions on Information Theory, Volume 46- No. 6, Sept 2000.

[11] Jiang F, Perez L, Vucetic B, "Recent Advances in Turbo Code Design and Theory", IEEE Explore, May 2011.

[12] A Turbo Tutorial by Jakob Dahl Andersen, Technical University of Denmark.

[13] On the Design of Turbo Codes, Dlvsalar D, Pollara F, Communications Syatem and Research Section.

[14] Turbo Code Concepts made easy, or How I learned to Concatenate and Reiterate, Sklar B.

[15] An Introduction to Turbo Codes by MC Valenti.

[16] Wireless Communications- Principles and Practices by Thedore Rapperbort.

[17] http://www.ee.vt.edu/yufei/turbo.html & www.mathworks.com

APPENDIX A

Block diagrams of different communication systems made by modifying the original system

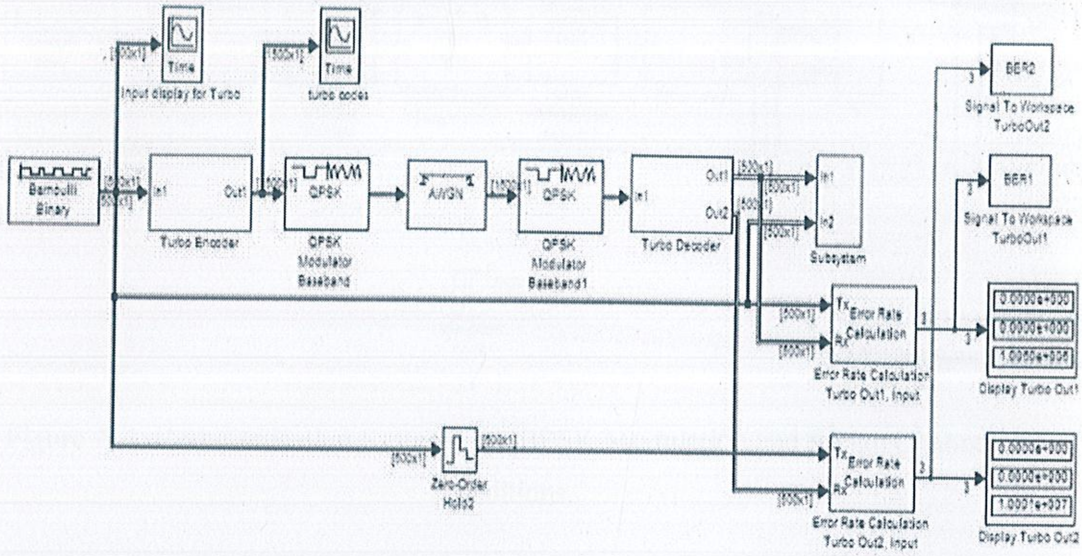

Figure A.1: Communication System with QPSK modulation and AWGN Channel Conditions

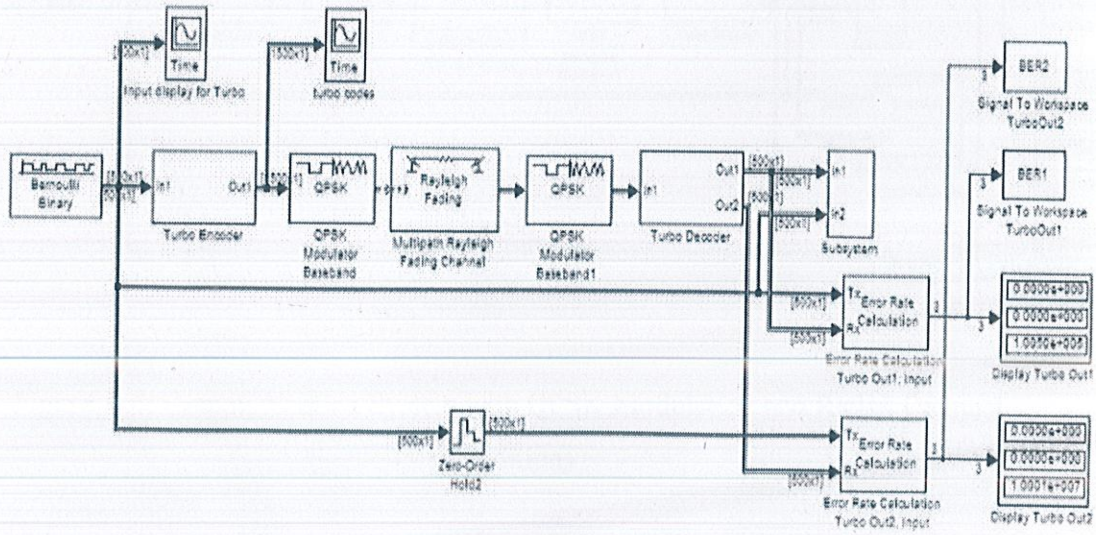

Figure A.2: Communication System with QPSK modulation and Rayleigh Channel Conditions

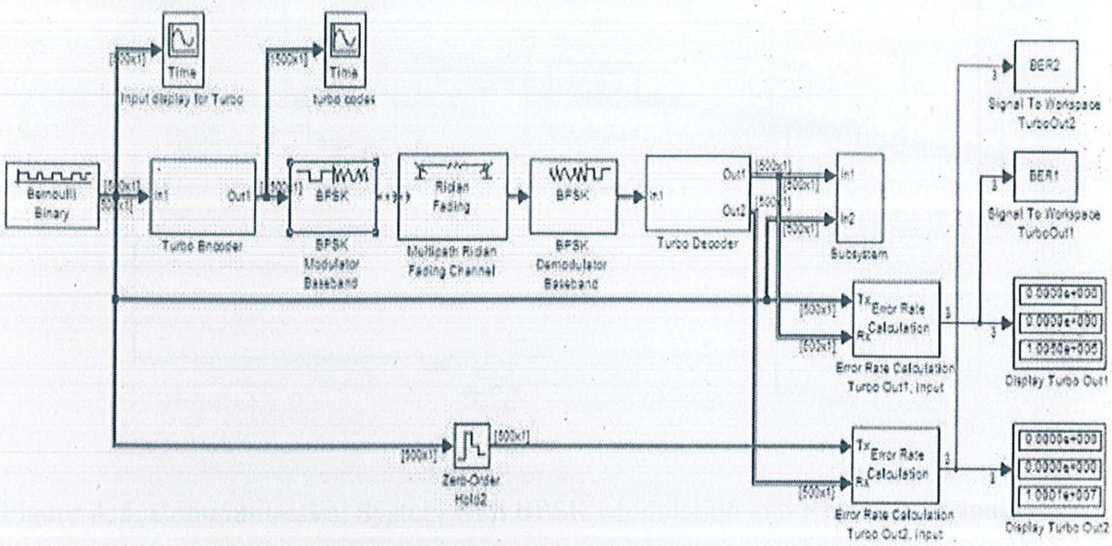

Figure A.3: Communication System with BPSK Modulation and Rician Channel Conditions

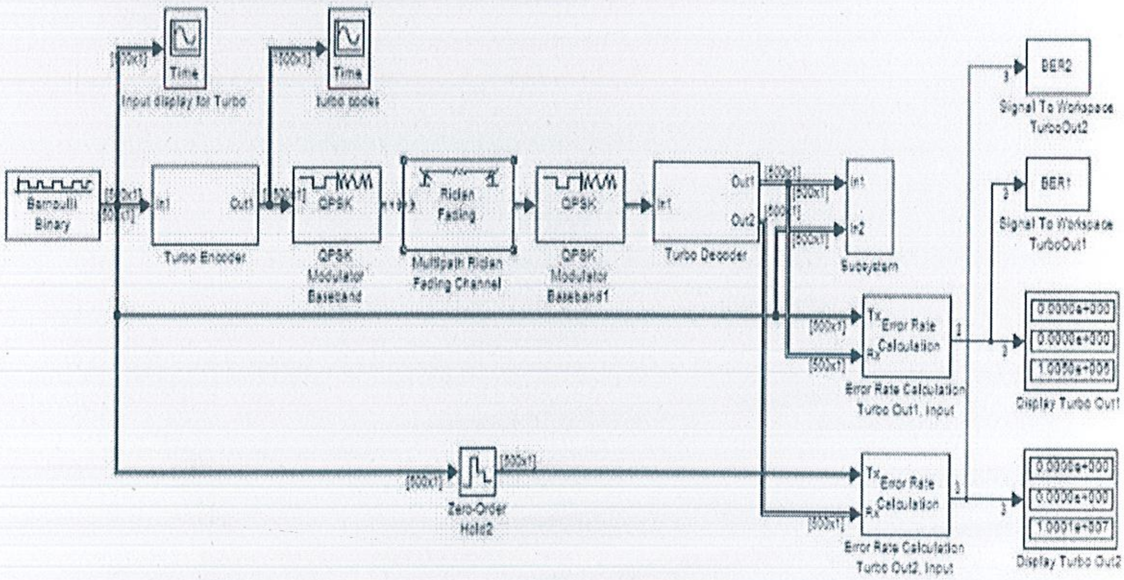

Figure A.4: Communication System with QPSK Modulation and Rician Channel Conditions

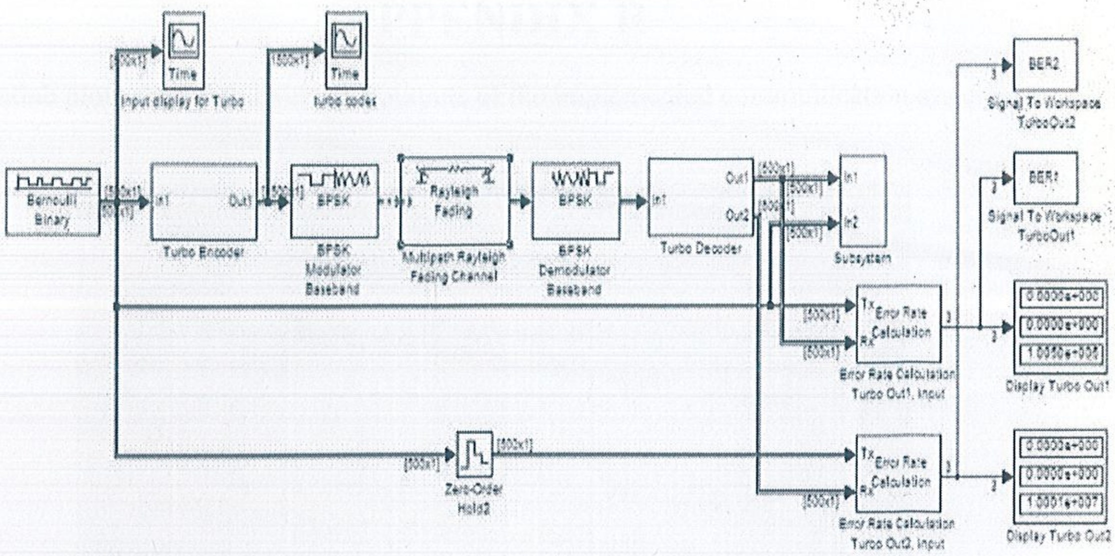

Figure A.5: Communication System with BPSK Modulation and Rayleigh Channel Conditions

APPENDIX B

Matlab plots showing inputs and outputs of the implemented communication systems

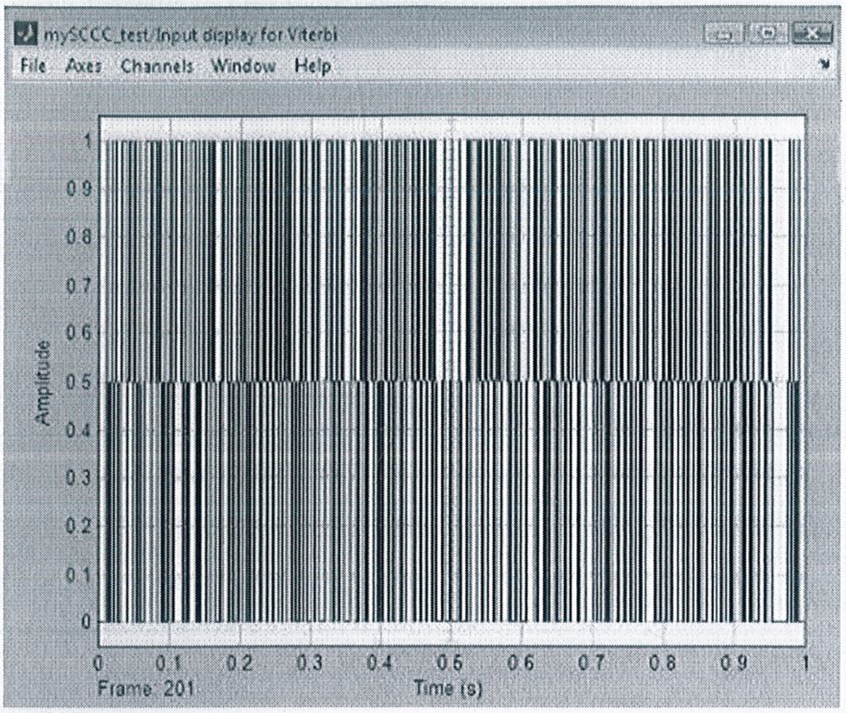

Figure B.1: Input as given to Communication System with Viterbi Decoder

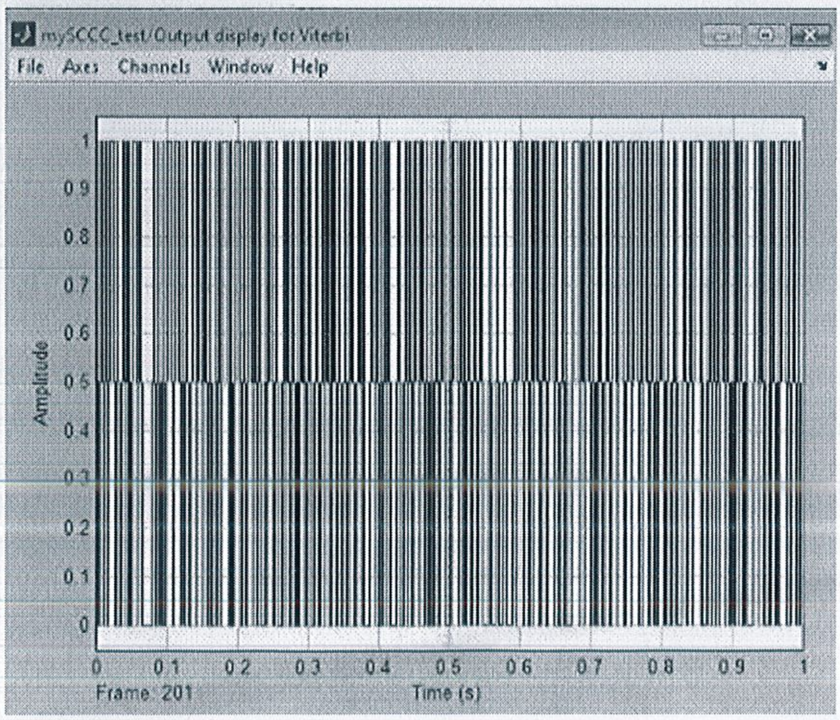

Figure B.2: Output as obtained for Communication System with Viterbi Decoder

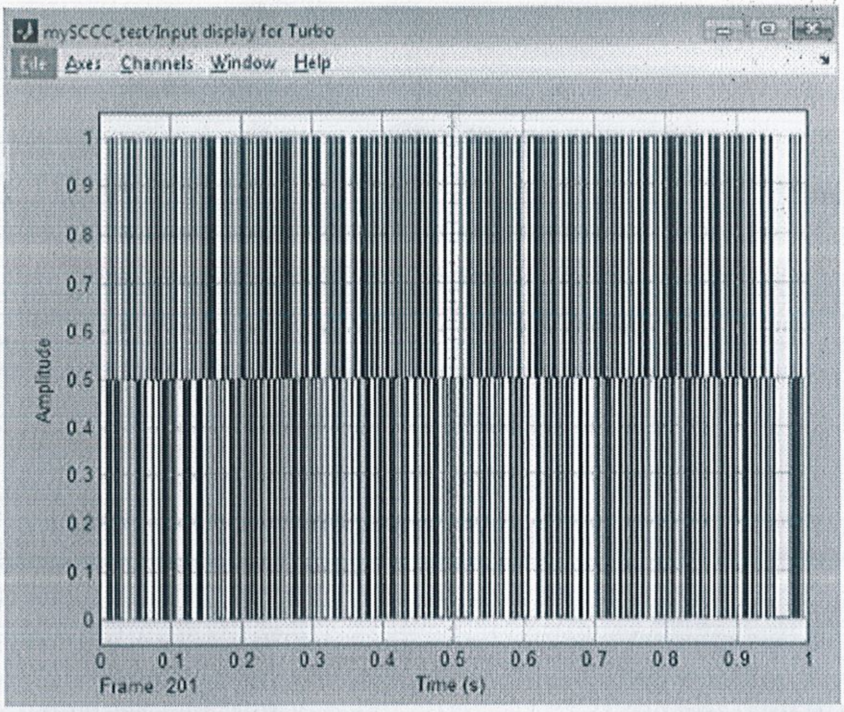

Figure B.3: Input as given to Communication System with Turbo Encoder and Decoder

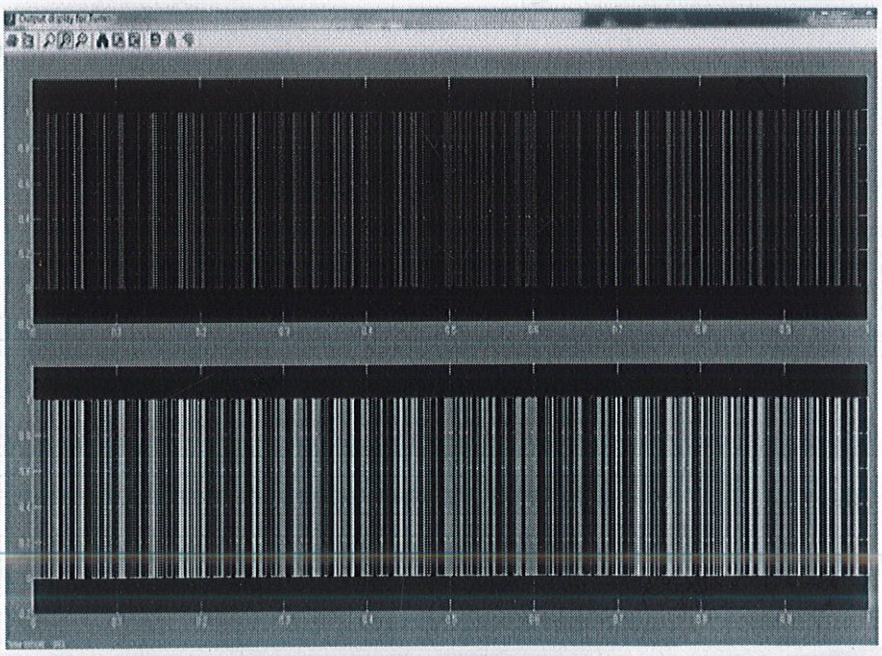

Figure B.4: Output as obtained from the Communication System with Turbo Encoder and Decoder

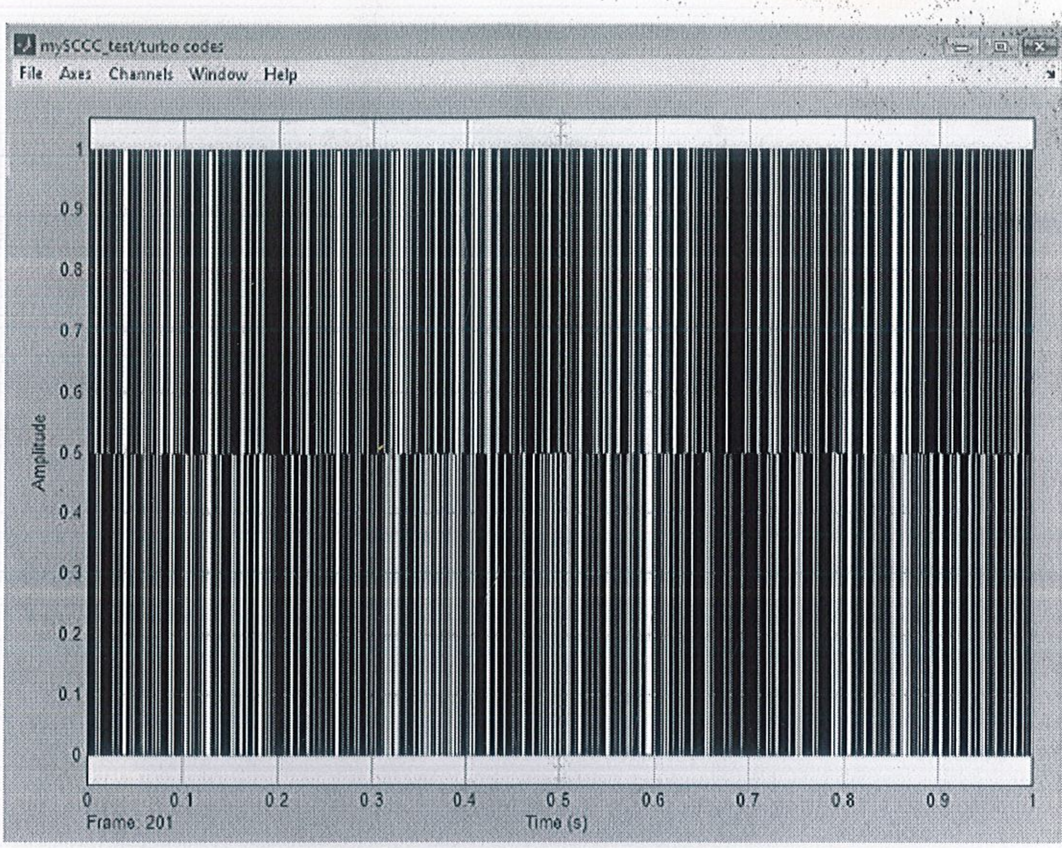

Figure B.5: Generation of Turbo Codes