SIMULATION OF ERROR CORRECTION CODES ON WIRELESS COMMUNICATION SYSTEMS

By

IBRAHIM SARIYYA AWWAL

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
AHMADU BELLO UNIVERSITY, ZARIA
NIGERIA

APRIL 2012

SIMULATION OF ERROR CORRECTION CODES ON WIRELESS COMMUNICATION SYSTEMS

By

IBRAHIM SARIYYA AWWAL

M.SC/ENG/15052/2007-08

A THESIS SUBMITTED TO THE SCHOOL OF POSTGRADUATE STUDIES, AHMADU BELLO UNIVERSITY ZARIA, IN PARTIAL FULFILMENT OF THE REQUIREMENTS OF MASTER OF SCIENCE IN ELECTRICAL ENGINEERING, DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING, AHMADU BELLO UNIVERSITY, ZARIA
NIGERIA

APRIL 2012

DECLARATION

I declare that the work in the thesis entitled '**S**imulation of Error Correction Codes on Wireless Communication Systems' has been performed by me in the Department of Electrical and Computer Engineering under the supervision of Dr. M.B. Mu'azu and Dr. D.D.Dajab.

The information derived from the literature has been duly acknowledged in the text and a list of references provided. No part of this thesis was previously presented for another degree or diploma at any university.

Ibrahim, Sariyya Awwal

Name of student	Signature

CERTIFICATION

This thesis entitled "Simulation of Error Correction Codes on Wireless Communication Systems" by Ibrahim, Sariyya Awwal meets the regulations governing the award of the degree of Master of Science of Ahmadu Bello University, Zaria and is approved for its contribution to knowledge and literary presentation.

---------- ----------

Dr. M.B. Mu'azu Date
Supervisor

---------- ----------

Dr. D.D. Dajab Date
Supervisor

---------- ----------

Dr. M.B. Mu'azu Date
Head of Department

---------- ----------

Prof. A.A. Joshua Date
Dean, School of Post-Graduate Studies

DEDICATION

To my mum, with much love and appreciation……I miss you.

ACKNOWLEDGEMENTS

Praise be to Almighty Allah for seeing me through this study successfully. I am indeed grateful to a number of people for all the help rendered in the course of this research. First and foremost, my profound gratitude goes to my supervisors Dr. M.B.Muazu and Dr. D.D.Dajab for taking time to guide me through this thesis. My appreciation also goes to the post graduate coordinator of the Department Dr. Suleiman Garba and all distinguished lecturers of the Department. They have added GREAT meaning to my engineering study at A.B.U. Zaria. To my dear friends Nana Hauwa, Hussaina and course mates whom I have benefited immensely from their kindness and generosity during the course of my study I say thank you very much. I would like to thank the most important persons in my life. I owe everything I have to my family. My sincere appreciation goes to my parents for their love and support throughout my study. May Allah reward you with Jannatil Firdausi Finally my deepest love to my husband (always supportive and understanding) and son who this study has turned to “engineers”; words cannot express my appreciation for you walking down this lane with me. I am very grateful. May the Almighty Allah reward you all abundantly, Ameen.

ABSTRACT

Error correction codes are widely used in almost all digital systems as they provide a method for dealing with the unknown, *noise*. This research has investigated the performance of error correction codes in wireless communication systems with the use of MATLAB. The error correction code employed was the convolution error correction codes. The performance of the codes were evaluated based on key performance indicators like gain, Bit Error Rate (BER) e.t.c. by transmitting randomly generated data through a Gaussian channel. For the verification of proposed approach, computer simulation results are included. The results showed a comparison of the performance of the convolution codes (with different code rates) in terms of their BER performance. Based on the results of the plots, up to 70% and 74% improvement on coding gains were achieved between reference points of 10^{-2} and 10^{-4} respectively for the two code rates (1/2 and 1/3) used. The results also showed that with the BER decreased, the coded system can transmit data signals with at least 3dB less power thus making the performance of the coded system better than the uncoded system.

TABLE OF CONTENTS

PAGE

TITLE PAGE i

DECLARATION iii

CERTIFICATION iv

DEDICATION v

ACKNOWLEDGEMENTS vi

ABSTRACT vii

CHAPTER ONE

INTRODUCTION 1

1.1 GENERAL INTRODUCTION 1

1.2 MOTIVATION 3

1.3 STATEMENT OF PROBLEM 4

1.4 SIGNIFICANCE OF RESEARCH 4

1.5 OBJECTIVES OF THE STUDY 5

1.6 THESIS OUTLINE 6

CHAPTER TWO

LITERATURE REVIEW AND THEORETICALBACKGROUND 7

2.0 LITERATURE REVIEW 7

2.1 SHANNON LIMIT 9

2.2 ERROR CORRECTING CODES ……..10

2.3 ERROR-CORRECTION CODE SELECTION……..11

2.4 TYPES OF ERROR CORRECTION CODES…….....12

2.4.1 BLOCK CODES.……..12

2.4.2 CONVOLUTION CODES……...13

2.5 PROPERTIES OF CONVOLUTION CODES…….……..14

CHAPTER THREE

DESIGN METHODOLOGY……..…….19

3.0 INTRODUCTION……..19

3.1 IMPLEMENTATION OF CODES……19

3.1.1 ENCODER REPRESENTATION……..20

3.1.2 ENCODER STRUCTURE……..20

3.1.3 DECODING CONVOLUTIONAL CODES……...22

3.2 PERFORMANCE MEASURE……...23

3.3 PERFORMANCE ANALYSIS……..26

3.4 SIMULINK BLOCKS……29

CHAPTER FOUR

RESULTS AND ANALYSIS.....33

4.0 SIMULATION RESULTS……..33

4.1 RESULT DISCUSSION…….……34

4.2 SIMULATION PLOTS…….……..35

4.3 SIMULATION OF RESULT……...…….……...38

CHAPTER FIVE

SUMMARY, CONCLUSION AND RECOMMENDATIONS…….39

5.1 INTRODUCTION..……..……..……..…….39

5.2 SUMMARY….……..……...….39

5.3 LIMITATION…….…..……39

5.4 CONCLUSION……..40

5.5 RECOMMENDATIONS FOR FURTHER WORK…...……...……....40

REFERENCES……..…..…….42

APPENDIX A……...……...…….46

APPENDIX B……...……...50

LISTS OF TABLES

PAGE

Table 4.1: The Convolution Encoder Characteristics used in the Simulation……33

Table 4.2: Simulation Results……34

LIST OF FIGURES

PAGE

Figure 1.1: A sample communication system……2

Figure 2.1: Block diagram of error correction code……...12

Figure 2.2: System diagram……....14

Figure 2.3: Model of a coding system……....18

Figure 3.1: Procedure of encoding and decoding in a communication system…….....19

Figure 3.2: A Convolutional encoder……..…...20

Figure 3.3: A Trellis Structure……..…...21

Figure 3.4: Convolutional coding flowchart……..….....27

Figure 3.5: Convolutional Coded System…….....28

Figure 3.6: An Uncoded System…….....29

Figure 4.1: Simulation Result for 1/3 Convolutional Code.....….35

Figure 4.2: Simulation Result for rate 1/2 Convolutional Code..……...…..35

Figure 4.3: Simulation Result for Uncoded System……..…….....36

Figure 4.4: Simulation Result for 1/2, 1/3 and Uncoded Systems……..……..36

Figure 4.5: Simulation Result for 1/2 Code and 1/3 Code……..……....37

LIST OF ABBREVIATIONS

ADC - Analogue-to-Digital converter

ASIC - Application-Specific Integrated Circuits

AWGN - Additive White Gaussian Noise

BCH - Bose–Chadhuri–Hocquenghem

BER - Bit Error Rate

BPSK - Binary Phase Shift Key

CDMA - Code Division Multiple access

ECC - Error Correction Code

FPGA - Field programmable gate array

GSM - Global System for Mobile

GUI - Graphical User Interphase

IDS - Insertion/Deletion/Substitution

MAP - Maximum A Posteriori

SNR - Signal to Noise Ratio

VHDL - VHSIC (very high speed integrated circuits) hardware description language

1-D - 1-Dimension

2-D - 2-Dimension

3-D - 3-Dimension

3-G - Third generation

LIST OF SYMBOLS

Π — Pi

∞ - Infinity

σ - Sigma

CHAPTER ONE

INTRODUCTION

1.1 BACKGROUND INFORMATION

The last thirty years have seen a dramatic change in the way communication is achieved around the world. Wireless communication has evolved from being an expensive and rare technology for the few in the 70's, to becoming a widespread and economical means for facilitating domestic, commercial, as well as public service communications. One of the major reasons for the continuous growth in the use of wireless communication is its increasing ability to provide efficient communication links to almost any location, at constantly reducing costs with increasing power efficiency (Jemibewon, 2000).

Wireless communication is one of the most active areas of technological development. This development is being driven primarily by the transformation of what has been a medium for supporting voice telephony into a medium for supporting other services such as transmission of video, images, text, and data e.t.c. (Wang and Poor, 2003). Basically, a communication system deals with information or data transmission from one point to another (Du and Swamy, 2009). Over the years, there has been a tremendous growth in digital communications especially in the fields of cellular, satellite, and computer communications. In these communication systems, the information is represented as a sequence of binary bits. The binary bits are then mapped (modulated) to analog signal waveforms and transmitted over a communication channel. The communication channel introduces noise and interference to corrupt the transmitted signal. At the receiver end, the channel corrupted transmitted signal is mapped back to binary bits. The received binary information is an estimate of the transmitted binary information (Huang, 1997). Normally, during signal transmission through noisy channels, errors can occur on the received data. Bit errors may result due to this transmission and the number of bit errors depends on the amount of noise and interference in the communication channel. These errors can be detected and corrected using coding techniques (Huang, 1997). Noise is any undesired signal in a communication circuit. Noise can also be unwanted disturbances superimposed on a useful signal, which tends to obscure its information content. There are many varieties of noise; however, the three general types of noise can be categorized as follows:

1) **Gaussian noise** is usually not dependent on time, meaning that it is random and not systematically planned. The amplitude of the frequency can vary, making a crackling notation or sound. Some examples of Gaussian noise can include splashing in a tank or unplanned interruption in a sensing device.
2) **Drift noise** is correlated to time, and has random movement. Examples of drift noise can include fouling or catalyst decay.

3) **Shot noise** may be defined as sporadic and short bursts of noise, in which the amplitude is similar among the bursts of noise. Examples of shot noise include partial clogging or jamming in the process in which the same amplitude will be seen by the noise whenever the clogging or jamming in the process occurs.

A sample communication system is shown in Figure 1.1

Destination

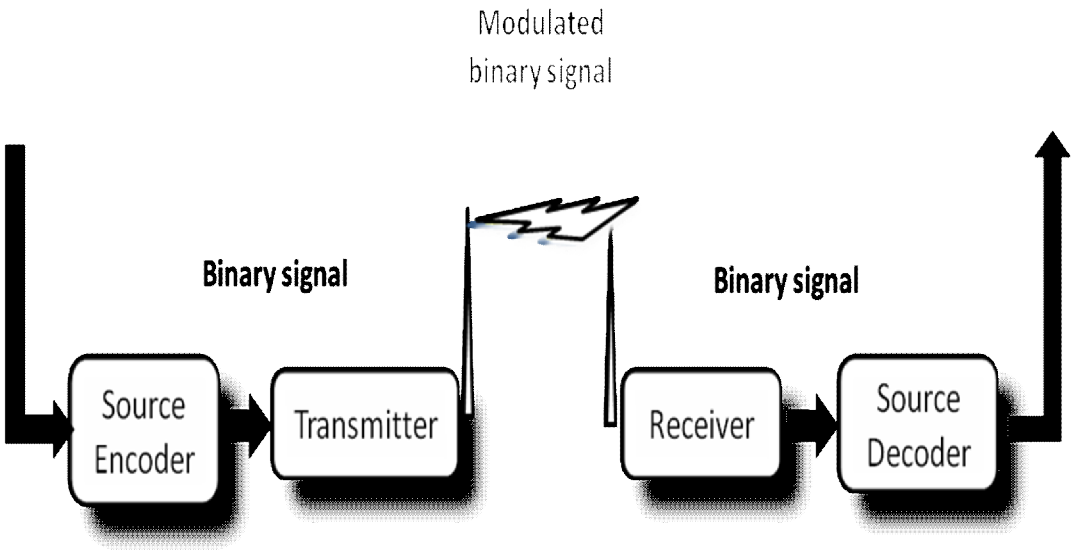


Figure 1.1: A sample communication system

The information source is where the message originates and the destination is where it is received. (Saha, 2003). Wireless communications encompasses various types of fixed, mobile, and portable two-way radios, cellular telephones, personal digital assistants (PDAs), and wireless networking. Other examples of wireless technology include GPS units, Garage door openers or garage doors, wireless

computer mice, keyboards and Headset (audio), headphones, radio receivers, satellite television, broadcast television and cordless telephones

There are three common types of transmission channels:

i. Wireless channels
ii. Guided electromagnetic waves channels
iii. Optical channels

The wireless channel can be the atmosphere. Due to its open nature, various noise sources are added to the channel (Du and Swamy, 2009). Some error causing effects include:

- Multipath propagation of signals
- Interference from other communication devices.
- Propagation path loss.

1.2 MOTIVATION

Today, wireless communication has become an integral part of everyone's life. Communication in adverse channels could have distortions such as dynamic noise, jamming, multi access interference, severe fading, and dropped calls, thus prove unreliable. Wireless communication applications require error correction schemes in order to maintain high quality of service, reliable transmission, improve safety and provide better performance. As communication revolution evolves, there is an increasing number of communications thriving, although with a growing shortage of bandwidth and the possibility of errors corrupting transmission systems, there is a need to improve communication systems with regards to protecting these systems from corruption of errors, while simultaneously enhancing the efficiency of bandwidth use as well as providing highly reliable and error free systems which should not be taken for granted given the nature of wireless communication channels. Error correction coding makes wireless communications more robust in the presence of noise. The noise source is the main focus in implementing error correcting codes.

1.3 STATEMENT OF PROBLEM

Various researches have been carried out with a view to securing and optimizing the usage of wireless communication systems. With bandwidth requirements in communication systems, the system capacity is depending on the level of interference that can be tolerated, and it is inversely proportional to the signal to noise ratio (SNR) (Qian, 1999). In other words, the system capacity increases as SNR decreases. In order to reduce SNR, error correction codes need to be integrated into the system. Error correction codes decrease SNR by introducing the coding gain for the communication link. The coding gain measures the amount of additional SNR required to provide the same BER performance for an uncoded message signal (Qian, 1999).

In 1948, Shannon proved that for a band limited Additive White Gaussian Noise (AWGN) channel, with bandwidth B, there exist families of coding schemes that can achieve arbitrarily small probability of error at the receiving end at a communication rate less than the capacity of the channel, which can be described as follows:

$$C = B \log_2 \left(1 + \frac{S}{N}\right) bits/sec \qquad (1.1)$$

Where C represents the capacity of the channel, S and N are the average signal power and noise power respectively. The implication of equation 1.1 is that if information rate can be dropped below the capacity of transmission channel, with proper error protection means, such as error correction codes, error free transmission is possible. In this research, error correction is achieved using convolution codes.

1.4 SIGNIFICANCE OF RESEARCH

One of the goals of wireless communication systems is to deliver information or data efficiently and reliably comparable to wired systems, at reasonable costs and convenience to as many users as possible. A system with good performance should be able to deliver clear, uncorrupted data with very little latency and minimal power consumption. In digital systems, the ability to achieve uncorrupted data transmission is directly proportional to a system's probability of bit error, which is in turn inversely proportional to the power of data transmission. One of the major drives in wireless communication is the effort to increase power efficiency, thus delivering reliable links at lower power levels. Many practices have been developed to assist in the achievement of higher power efficiency, one of which is the implementation of error-correction codes.

This practice involves the transmission of extra bits for redundancy in order to reduce the probability of bit error. As expected, the transmission of extra bits requires the use of additional spectrum, which reduces the available bandwidth for transmitting information bits, thereby resulting in a drop in bandwidth efficiency. This illustrates a very important dilemma in wireless communication systems: the tradeoff between power efficiency and bandwidth efficiency. Cost, convenience and mobility issues also need to be considered in order to provide the best wireless communication systems possible. Ultimately, each wireless communication needs to make the best tradeoff possible between all these requirements in order to remain relevant in today's communication environment (Jemibewon, 2000). The cost of using error correction coding to protect the information is a reduction in data rate or an expansion in bandwidth.

1.5 OBJECTIVES OF THE STUDY

The main objectives chosen for this thesis are highlighted as follows:

- Understand the concept of error correction in wireless communication systems.
- Implementation of error correction codes in a noisy channel using convolution codes to compensate transmission impairment in order to increase quality of transmission.
- Creation of the experimental simulation model of a communication system in MATLAB® 2008/Simulink to oversee the process of error correction implementation.
- Investigate the performance of the chosen codes (convolution codes) with different code rates.

1.6 THESIS OUTLINE

There are five chapters contained in this project: Chapter one is the general introduction to the research. Chapter two entails literature reviews of related researches on error correction codes, an overview of a wireless communication system emphasizing the important types of error correction codes. Chapter three describes the methodology for the error correction coding in this research which is achieved using convolution codes. Chapter four consists of the simulation results obtained in investigating the performance of convolution codes (for two code rates), analysis and comparison of coded and uncoded systems. Chapter five contains the summary, conclusions and suggestions for further work.

CHAPTER TWO

LITERATURE REVIEW AND THEORETICAL BACKGROUND

2.0 LITERATURE REVIEW

Emerging wireless communication systems transmit high data rates to provide wide range of services including high quality voice, data, images and other multimedia applications. Unfortunately, wireless communication signals propagate through harsh communication channels encountering various obstacles resulting in data errors at the receiver end. Error correction codes are a means of including redundancy in a stream of information bits to allow the detection and correction of symbol errors during transmission. The approach to error correction coding taken by modern digital communication systems started in the late 1940's with the ground breaking work of Shannon, Hamming and Golay (Valenti, 1999). The birth of error correction coding showed that Shannon's channel capacity could be achieved when transmitting information through a noisy channel (Jemibewon, 2000).

Several researches have been carried out on error correction codes for different types of systems. In Huang (1997), the evaluation of soft output decoding for turbo codes was carried out in which the performance of turbo codes were evaluated through computer simulation. Coding theory related to this research was also studied, including convolution encoding and Viterbi decoding.

Jemibewon (2000), studied the smart implementation of turbo decoding for improved power efficiency in which the technical feasibility of adapting the resolution of analog-to-digital converter (ADC) in a decoder and the number of decoding iterations in order to obtain the best trade-off between BER performance and power dissipation in a communication system with the aid of computer simulations.

Thomas *et al.* (2003) presented an article on integrated circuits for channel coding in third generation (3G) cellular mobile wireless systems. The article considered decoder design problems and using case studies described 3G decoder solutions using application-specific integrated circuits (ASIC).

Source and Channel Coding Techniques necessary for reliable transmission of Digital Mammograms over a wireless AWGN channel was investigated by Hosseinzadeh *et al.* (2004). Lossless source compression techniques were investigated. Error correction and detection mechanisms were also discussed focusing on schemes that achieve the lowest bit error rates to ensure reliable data transfer over a noisy wireless channel.

Vats and Kacewicz (2005) studied error correcting codes for speech channels of cellular phones with emphasis laid on the most popular cell phone standard, Global System for Mobile communication (GSM). Simulations using MATLAB® 2008 were carried out to show the power of convolution codes and interleaving done in GSM for speech signals. Codes used in the competing code division multiple access (CDMA) standard were also evaluated.

Cheng (2005), in his research investigated error protection using convolution encoder and Viterbi decoder system against insertion, deletion and substitution errors (IDS). Two independent IDS error correcting designs working on the convolution encoder and the Viterbi decoder were presented.

Balakrishnan *et al.* (2007) studied the performance analysis of various error control codes in terms of their BER performance and power consumption on different platforms. Error control codes with different constraints were implemented and simulated using VHSIC (Very high speed integrated circuits) hardware description language (VHDL). Implementation on field-programmable gate array (FPGA) and ASIC design were also carried out and the energy consumption was measured. The error control performance of these codes was then evaluated in terms of Bit Error Rate (BER) by transmitting randomly generated data through a Gaussian channel.

Cheong (2007) also carried out a study on block error correction codes and convolution codes. The paper gave an overview on error control coding employed in forward error correction, particularly block codes and convolution codes.

Simulation of error correction codes on wireless communication systems investigates the performance of error correcting codes for data signals used in wireless systems; cellular phones, internet, geolocation/GPS, emergency services, radio and television broadcasting e.t.c. This research work employed the use of convolution error correction codes in order to deal effectively with noise. With the use of MATLAB®2008 for simulation and analysis, the error correction performance of the convolution code (with different code rates) was evaluated in terms of Bit Error Rate (BER) by transmitting randomly generated data through a Gaussian channel. Bit-error-rate (BER) performance and power dissipation are two important measures of performance used to characterize communication systems.

2.1 SHANNON LIMIT

In 1948, Claude Shannon proved that any communication channel could be characterized by maximum theoretical capacity, C. If the source information rate, R, is less than C (R<C), then there exist channel-encoding method such that information can be reliably transmitted. This theorem set a theoretical limit on possible information rate for achieving reliable (error-free) transmission through the channel by appropriate coding. On the other hand, if R>C, reliable transmission is not possible regardless of the amount of signal processing performed at the transmitter and receiver. Thus if the required transmission rate R (measured in bits per second) is less than C, it is possible to design a communication system for that channel and with the help of error-control coding, a very small probability of output error can be achieved for that channel. The capacity C in bits per second (b/s) depends on only two parameters, the channel bandwidth and the signal-to-noise ratio. In practice, it has proved to be remarkably difficult to find classes of constructive codes that can be decoded by feasible decoding algorithms at rates, which come at all close to the Shannon limit (Arasteh, 2006).

2.2 ERROR CORRECTING CODES

At present, the use of communication systems has grown in large quantities. For that reason, it is required that communication systems make data transmission safe, more efficient and with a better performance (faster transmission and more efficient use of bandwidth). In other words, communication systems need to be more immune to noise and allow for transfer of large amounts of information. Error correction coding involves the transmission of additional redundant bits in the stream of information bits, in order to allow the detection and correction of some symbol errors at the receiver. Although the use of channel coding improves the error performance of communication systems, the transmission of additional bits results in a reduction in bandwidth efficiency (Jemibewon, 2000).

Error correcting codes add redundancy into bits to improve the performance of a system. They are widely used in almost all digital systems as they provide a method for dealing with the unknown, *noise*. But this capability has several costs and trade-offs. Apart from needing an extra encoder and decoder, we have to transmit more information in the same amount of time *i.e.* increased data rates which translates into more bandwidth (Vats and Kacewicz, 2005).

Error Correction Codes (ECC) encodes data in such a way that a decoder can identify and correct certain errors in the data. Channel capacity is a theoretical measure of the fastest rate at which error free transmission can be realized. Traditional modulation techniques deliver performance significantly inferior to that predicted by Shannon's work, but when incorporated with error correction coding, most digital modulation schemes can achieve performance that approaches channel capacity (Jemibewon, 2000). Shannon also established the fundamental limits in transmission speeds in the systems of digital communications and led to the search for coding techniques for approaching the limit of this capacity (Morales, 2008).

When digital data is stored in a memory, it is crucial to have a mechanism that can detect and correct a certain number of errors. Usually data strings are encoded by adding a number of redundant bits to them. When the original data is reconstructed, a decoder examines the encoded message, to check for any errors (Valenti, 1999).

Practically all systems that transmit or store digital data use some form of error correction technique. In the presence of real-world phenomenon, designers of these systems have to face the certain outcome that some transmitted or stored bits of information will not be received the way they were intended. The choice of how to handle errors ranges from doing nothing to using elaborate error detection and correction methods. Selecting between the choices depends on the accuracy, speed, and latency requirement of the information, and whether or not there is simultaneous bidirectional communications between the sender and receiver (Rossi, 1998).

2.3 ERROR-CORRECTION CODE SELECTION

Thorough research, both theoretical and numerical, must precede the design of error correction codes. There are many error-correcting codes and their combinations, each fitting a particular task. The basic issues to be taken into account in order to choose an error-correcting code (ECC) are:

- Physical channel characteristics (Signal to noise ratio, presence of memory, e.g. *fading*)
- Modulation type
- Desirable channel characteristics (information rate, BER - bit error probability).

A simple measure for the quality of a code is the information rate which is the measure of how much of the code is the message. The rate is given as:

$$\frac{\log 2C}{n} \qquad (2.1)$$

Where,

c represents the number of code words in the code.

2.4 TYPES OF ERROR CORRECTION CODES

There are two main types of error correction codes (also known as channel codes), as shown in Figure 2.1, namely:

- Block codes
- Convolution codes

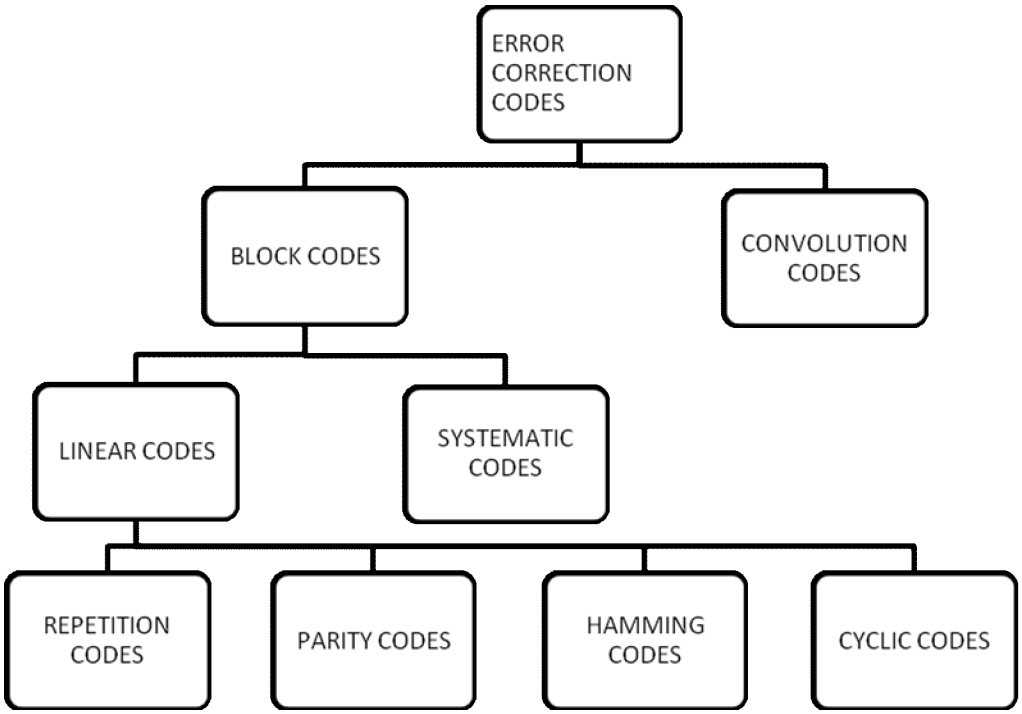


Figure 2.1: Block diagram of error correction code

2.4.1 Block Codes

Block codes are codes that require data being compiled into blocks before coding or decoding can take place (Houghton, 1997). They are based rigorously on finite field arithmetic and abstract algebra. They can be used to either detect or correct errors. Each codeword in the set is a linear combination of a set of generator code words. They have very high code rates and are useful in situations where bit error rate (BER) of a channel is relatively low, bandwidth availability is limited (Shelton, 1999). Block codes are also referred to as **(n, k)** codes where a block of **k** data bits is encoded to become a block of **n** bits called a code word (Valenti, 1999). By predetermined rules, n-k redundant bits are added to the k information bits to form the n coded bits. Block codes, also known as redundancy codes, are thought of as giving external protection in the encoding process. Some of the commonly used block codes are Hamming

codes, Golay codes, Bose–Chadhuri–Hocquenghem (BCH) codes, and Reed Solomon codes (which uses non binary symbols) (Vats and Kacewicz, 2005).

2.4.2 Convolution Codes

These are alternatives to block coding in which coding and decoding can take place on a continuous data bit stream (Houghton, 1997). They are used in applications that require good performance with low implementation cost. They operate on data stream not static block. They are usually denoted by (**n**, **k, l**) (Er. Liu, 2004). The code words produced depend on both the data message and a given number of previously encoded messages. The encoder changes state with every message processed. The length of the code word is usually constant (Valenti, 1999). In almost all applications of digital communications, convolution codes are used as the main error correction code because of its ease of implementation compared to an equivalent block code. Convolution codes convert the entire data stream into one single codeword. The encoded bits depend not only on the current input bits but also on past input bits. The main decoding strategy for convolution codes is based on the widely used Viterbi algorithm (Huang, 1997).

Figure 2.2 shows the typical architecture of a system that uses an error correcting code.

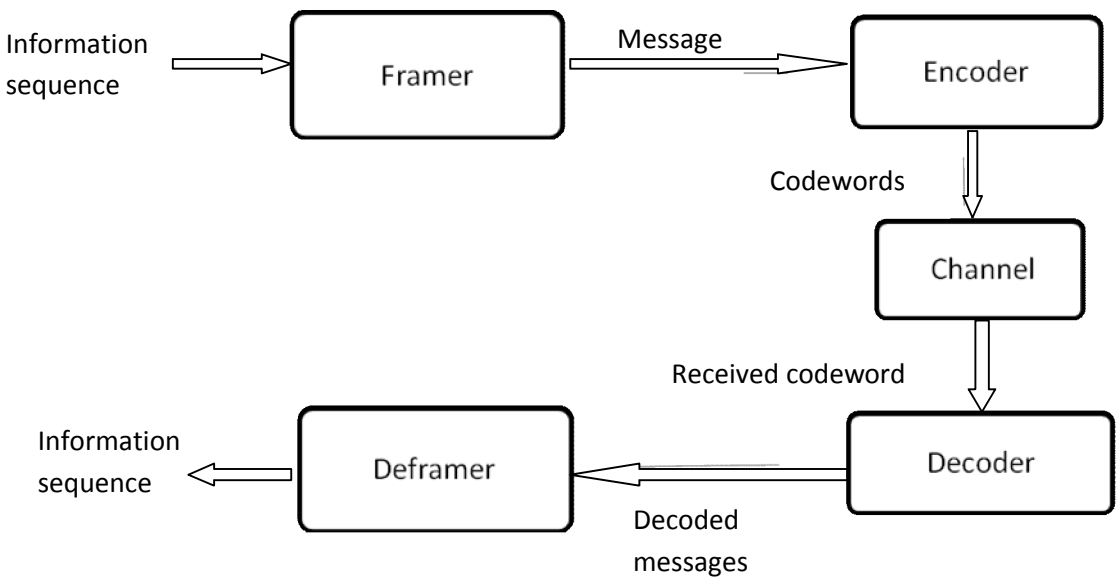


Figure 2.2: System diagram

At the transmitter, the framer converts the information sequence into smaller blocks called messages. The messages are then encoded using an error correcting code into codewords which are then transmitted over an unreliable channel. The channel delivers a noisy version of the codewords to the receiver. The receiver performs the reverse process of:

1) Recovering the original messages, i.e., decoding, and
2) Reconstructing of the original information sequence from the decoded messages, i.e., deframing (Nazer and Gastpar, 2008).

2.5 PROPERTIES OF CONVOLUTION CODES

Passing the information sequence to be transmitted through a linear finite state shift register generates convolution codes. Convolution codes can be summed up as codes without a block structure (Vats and Kacewicz, 2005). Convolution codes are one of the most widely used channel codes in practical communication systems. These codes are developed with a separate strong mathematical structure and are primarily used for real time error correction. They are usually denoted by (**n**, **k, l)**

Where,

n- Number of output bits

k- Number of input bits

l- Code memory depth

To encode a block code, an infinite data stream is divided into blocks and then the code is applied on each individual block. The final output depends on the corresponding block it came from in the input stream. For convolution codes, the stream is again divided into blocks, but the blocks are much smaller. The final output for convolution codes depends on multiple blocks from the input stream (Vats and Kacewicz, 2005).

There are three parameters which define the convolution code:
(a) **Rate**: This is the ratio of the number of input bits to the number of output bits. That is to say that code rate R is determined by input rate and output rate:

$$R = \frac{k}{n} \quad < \quad 1 \qquad (2.2)$$

Where,

k is the number of input bits and **n** is the number of output bits

(b) **Constraint length**: Constraint length is defined as the number of symbols in the data stream that have an effect on the output of the encoder.

The constraint length K for a convolution code is defined as:

$$k = m + 1 \qquad (2.3)$$

Where

m is the maximum number of stages (memory size) in any shift register.

The shift registers store the state information of the convolution encoder and the constraint length relates the number of bits upon which the output depends.

(c) **Generator polynomial**: This is a polynomial involved in wiring of the input sequence with the delay elements to form the output (Sankar, 2009).

According to Shannon, the band limited Additive White Gaussian Noise channel with bandwidth B has the capacity C:

$$C = B \log_2 \left(1 + \frac{S}{N}\right) bits/sec \qquad (2.4)$$

Where *S* and *N* are the average signal power and noise power respectively.

The data transmission can be protected from errors induced by noise in the channel provided that the information rate R is less than the capacity C,

$$R < C \qquad (2.5)$$

Since

$$N = N_0 B \qquad (2.6)$$

If the two-sided power spectrum density of the noise is:

$$\frac{N_0}{2} Watts/\mathrm{Hz}$$

Equation (2.4) becomes:

$$C = B \log_2 \left(1 + \frac{S}{N}\right) bits/sec$$

$$= B \log_2 \left(1 + \frac{S}{N_0 B}\right)$$

$$= \frac{S}{N_0} \log_2 \left(1 + \frac{S}{N_0 B}\right)^{\frac{N_0 B}{S}} bits/sec \quad (2.7)$$

From equation (2.7)

$$\lim_{b \to \infty} c = \frac{S}{N} \log_2 e \quad (2.8)$$

The energy per bit can be written as

$$E_b = \frac{S}{R} \quad (2.9)$$

Combining 2.5, 2.8 and 2.9 yields

$$\frac{S}{E_b} < \frac{S}{N_0} \log_2 e \quad (2.10)$$

Therefore

$$\frac{E_b}{N_0} > \ln 2 \approx -1.6dB \quad (2.11)$$

Not considering its bandwidth efficiency, binary phase shift keying (BPSK) is known to be the optimum binary modulation scheme, because it is one kind of antipodal sampling. Consequently, it is often used in communications theory as a benchmark for comparison. For the uncoded BPSK,

$$BER = Q\left(\sqrt{2}\frac{E_b}{N_0}\right) \quad (2.12)$$

Where

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{\frac{-y^2}{2}} dy \quad (2.13)$$

And BER denotes the bit error rates of the received sequence (Cheng, 2005).

In information theory, the data transmission can be simplified as shown in Figure 2.3.

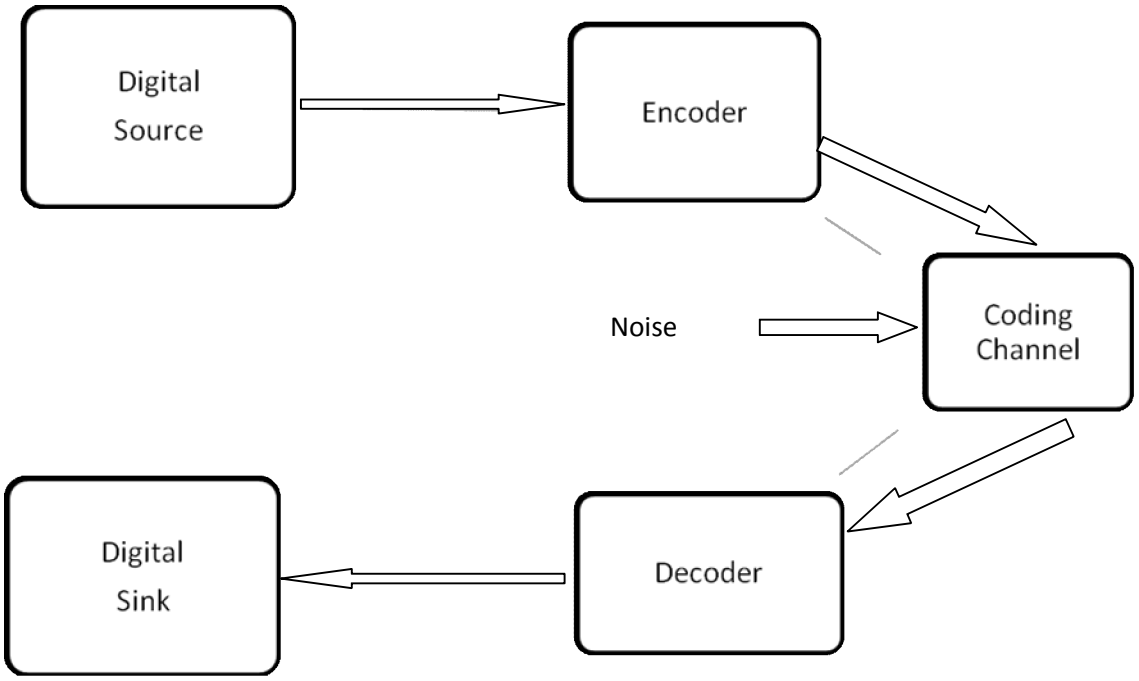


Figure 2.3: Model of a coding system

CHAPTER THREE

DESIGN METHODOLOGY

3.0 INTRODUCTION

In this research, a strategy to present convolution codes used for error correction in wireless communication systems was introduced, without using mathematical structures. Methods used in the study are explained in the following sections:

3.1 IMPLEMENTATION OF CODES

The convolution error correction code is implemented by simulations in MATLAB®2008. Figure 3.1 illustrates the procedure for encoding and decoding in a communication system, where u is the information word, v is the codeword, v' is the received word and u' is the decoded word.

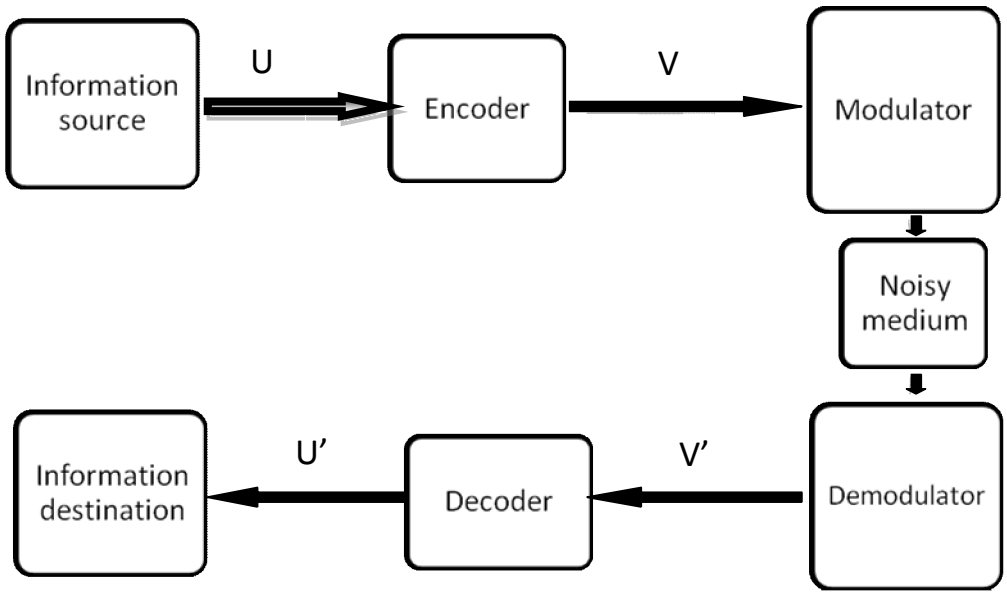
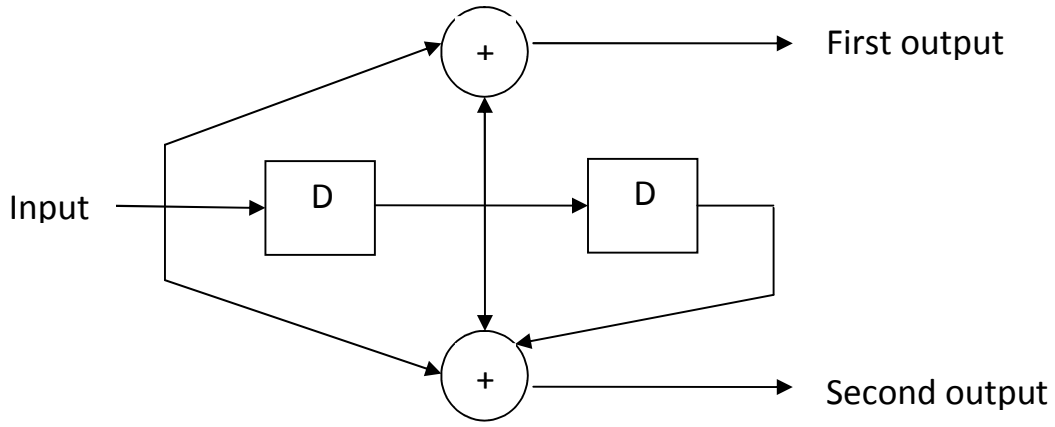


Figure 3.1: Procedure of encoding and decoding in a communication system.

A convolution code can become very complicated with various code rates and constraint lengths. As a result, a simple convolution code will be used to describe the code properties as shown in Figure 3.2 (Huang, 1997).



D – Shift register

Figure 3.2: A Convolution Encoder

3.1.1 ENCODER REPRESENTATION

The encoder can be represented in several different but equivalent ways. They are

1. Generator Representation

2. Tree Diagram Representation

3. State Diagram Representation

4. Trellis Diagram Representation (Huang, 1997).

3.1.2 ENCODER STRUCTURE

As with any binary code, convolution codes protect information by adding redundant bits. A rate-*k/n* convolution encoder processes the input sequence of *k*-bit information symbols through one or more

binary shift registers (possibly employing feedback). The convolution encoder computes each *n*-bit symbol (*n* > *k*) of the output sequence from linear operations on the current input symbol and the contents of the shift register(s). Thus, a rate *k/n* convolution encoder processes a *k*-bit input symbol and computes an *n*-bit output symbol with every shift register update (Wesel, 2003). Convolution encoding is used for coding of the transmitted bits. A trellis structure is used to describe convolution codes in this section. A trellis description of a convolution encoder shows how each possible input to the encoder influences both the output and the state transitions of the encoder. Figure 3.3 depicts the trellis of the convolution encoder shown in Figure 3.2.

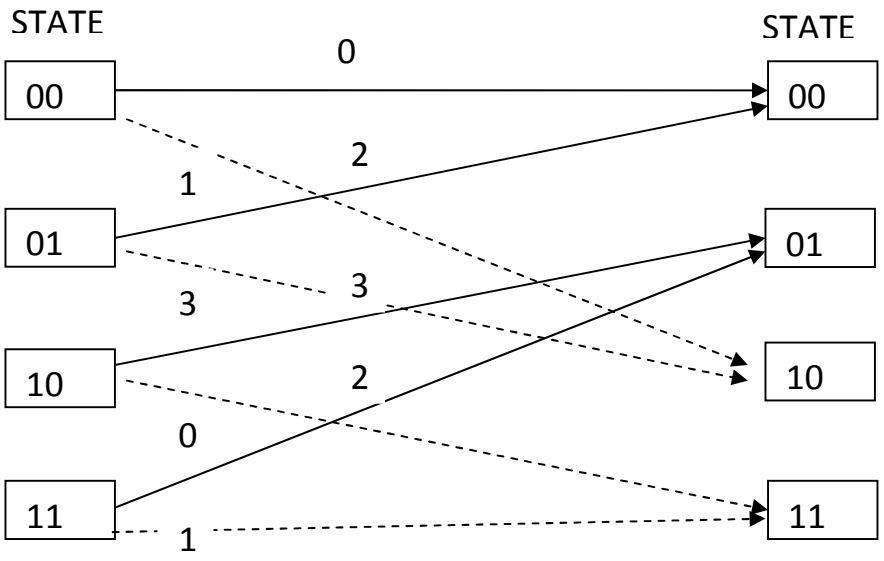


Figure 3.3: A Trellis Structure

The encoder has four states (numbered in binary from 00 to 11), a one-bit input, and a two-bit output (The ratio of input bits to output bits make this encoder a rate-1/2 encoder). Each solid arrow shows how the encoder changes its state if the current input is zero, and each dashed arrow shows how the encoder changes its state if the current input is one. The octal numbers above each arrow indicate the current output of the encoder.

As an example of interpreting this trellis diagram, if the encoder is in the 10 state and receives an input of zero, then it outputs the code symbol 3 and changes to the 01 state. If it is in the 10 state and

receives an input of one, then it outputs the code symbol 0 and changes to the 11 state (MATLAB®2008 7.1, 2005).

To specify a trellis in MATLAB®2008, a specific form of a MATLAB®2008 structure called a trellis structure is used (MATLAB®2008 7.1, 2005). A poly2trellis function is used to convert generator polynomials to a Trellis structure thus:

Trellis = poly2trellis (Constraint Length, Code Generator)

A convolution encoder is called so because it performs a convolution of the input stream with encoder's impulse responses:

$$y_i^j = \sum_{k=0}^{\infty} h_k^j x_{i-k} \qquad (3.1)$$

Where x is an input sequence, y^j is a sequence from output j and h^j is an impulse response for output j.

3.1.3 DECODING CONVOLUTION CODES

Convolution code decoding algorithms infer the values of the input information sequence from the stream of received distorted output symbols. There are three major families of decoding algorithms for convolution codes: Sequential, Viterbi, and Maximum a posteriori (MAP). Wozencraft proposed sequential decoding in 1957. Fano in 1963 and Zigangirov in 1966 further developed sequential decoding (Wesel, 2003).

Viterbi originally described the decoding algorithm that bears his name in 1967 introducing the trellis structure and showing that Viterbi decoding is maximum-likelihood in the sense that it selects the sequence that makes the received sequence most likely. The goal of the Viterbi algorithm is to find the transmitted sequence (or codeword) that is closest to the received sequence. As long as the distortion is not too severe, this will be the correct sequence (Wesel, 2003). The Viterbi decoder block works

according to the maximum likelihood decoding. It means finding the most probable transmitted symbol stream from the received codeword.

The Viterbi decoder (Shooshtary, 2008) defines a metric for each path and makes a decision based on this metric. The most common metric is the Hamming distance metric. When two paths come together on one node, the shortest hamming distance is kept. The number of trellis branches is defined as trace back depth. To define a convolution decoder in MATLAB®2008 simulation, a poly2trellis function is used to convert convolution code to trellis description (MATLAB®2008 7.1, 2005).

Trellis = poly2trellis (Constraint Length, Code Generator).

3.2 PERFORMANCE MEASURE

The error correcting capability of the implemented codes which is given by BER needs to be measured. This is obtained by dividing the number of erroneous bits with the total number of transmitted bits. BER is affected by several factors including noise in the channel, quantization technique used, code rate *R*, transmitter power level P_{out} and energy per symbol to noise ratio which is given by:

$$\frac{E_s}{N_0} \tag{3.2}$$

The code rate *R* is given by:

$$R = \frac{k}{n} \tag{3.3}$$

Where k is the number of bits at the input of the encoder and n is the number of bits at the output of the encoder. The BER is shown to be directly proportional to the code rate and inversely proportional to energy per symbol noise ratio and transmitter power level (Balakrishnan *et al.*, 2007). The encoder encodes the data with code rate *R* and transmits it over the noisy channel. If the transmitter power level P_{out} is unchanged, then the received energy per symbol *E* decreases to *R*×*E*. Hence, the BER measured at the input of the decoder is larger than the BER of the data transmitted without coding (Balakrishnan *et al.*, 2007).

This increase in BER is overcome by using a decoder that can correct errors. Proper choice of error correction codes will reduce the BER to several orders of magnitude. The difference in BER achieved by using error correction codes to that of uncoded transmission is referred to as coding gain.

To evaluate the performance of the convolution codes in the noisy channel, an Additive White Gaussian Noise (AWGN) channel is modeled. White noise is a random signal (or process) with a flat power spectral density. Gaussian noise is statistical noise that has a probability density function of the normal distribution (also known as Gaussian distribution). In other words, the values that the noise can take on are Gaussian-distributed. It is most commonly used as additive white noises to yield additive white Gaussian noise (AWGN). Examples include splashing in a tank or unplanned interruption in a sensing device.

Adding Gaussian noise to the encoded data is done by generating Gaussian random numbers with desired energy per symbol to noise ratio (Balakrishnan *et al.*, 2007). The variance σ^2 of additive Gaussian noise which has the power spectrum of *No/2(Watts*/Hz) is given by:

$$\sigma^2 = \frac{N_0}{2} \tag{3.4}$$

If the energy per symbol *Es* is set to 1, then:

$$\frac{E_s}{N_0} = \frac{1}{2}\sigma^2 \tag{3.5}$$

And the standard deviation σ is given by:

$$\sigma = \sqrt{\left(\frac{1}{2}\left(\frac{E_s}{N_0}\right)\right)} \tag{3.6}$$

Hence, the standard deviation σ with desired *Es/No* is calculated. The received symbols are fed to the Viterbi decoder to obtain the information bits. The decoded data is then compared with the corresponding input given to the encoder and the BER is calculated. To calculate theoretical bounds for the bit error rate P_b of a convolution code, the estimate of equation (3.7) can be used (MATLAB® 2008 7.1, 2005):

$$P_b < \sum_{d=f}^{\infty} C_d P_d \qquad (3.7)$$

Where c_d is the sum of bit errors for error events of distance d, and f is the free distance of the code. The quantity P_d is the pair wise error probability, given by:

$$P_d = \frac{1}{2} erfc \left[\sqrt{dR \frac{E_b}{N_o}} \right] \qquad (3.8)$$

Where R is the code rate and erfc is the MATLAB®2008 complementary error function, defined by:

$$erfc(x) = \frac{2}{\sqrt{\pi}} \int_{x}^{\infty} e^{-a} \qquad (3.9)$$

The minimum free distance of the code determines the performance of the convolution code. d_{free} is the minimum Hamming distance between two different code words which is also called free distance (Shooshtary, 2008).

3.3 PERFORMANCE ANALYSIS

The performance analysis of convolution codes is based on the notion of an "error event". Suppose that the transmitted code sequence is y (D) and the detected code sequence is y' (D). Each of these sequences specifies a unique path through a minimal code trellis. Typically these paths will agree for long periods of time, but will disagree over certain finite intervals. An error event corresponds to one of these finite intervals. It begins when the path y' (D) first diverges from the path y (D), and ends when these two paths merge again.

The error sequence is the difference (over this interval):

$$e(D) = y'(D) - y(D) \quad (3.10)$$

Where,

y' (D) is the detected code sequence and y (D) the transmitted code sequence.

The BER is used to obtain the coding gain, which is a measure of the difference between signal-to-noise ratios (SNR) between the uncoded system and the coded system to reach the same level of BER. Convolution coding is performed by simulations in MATLAB®2008 following the steps as illustrated in Figure 3.4.

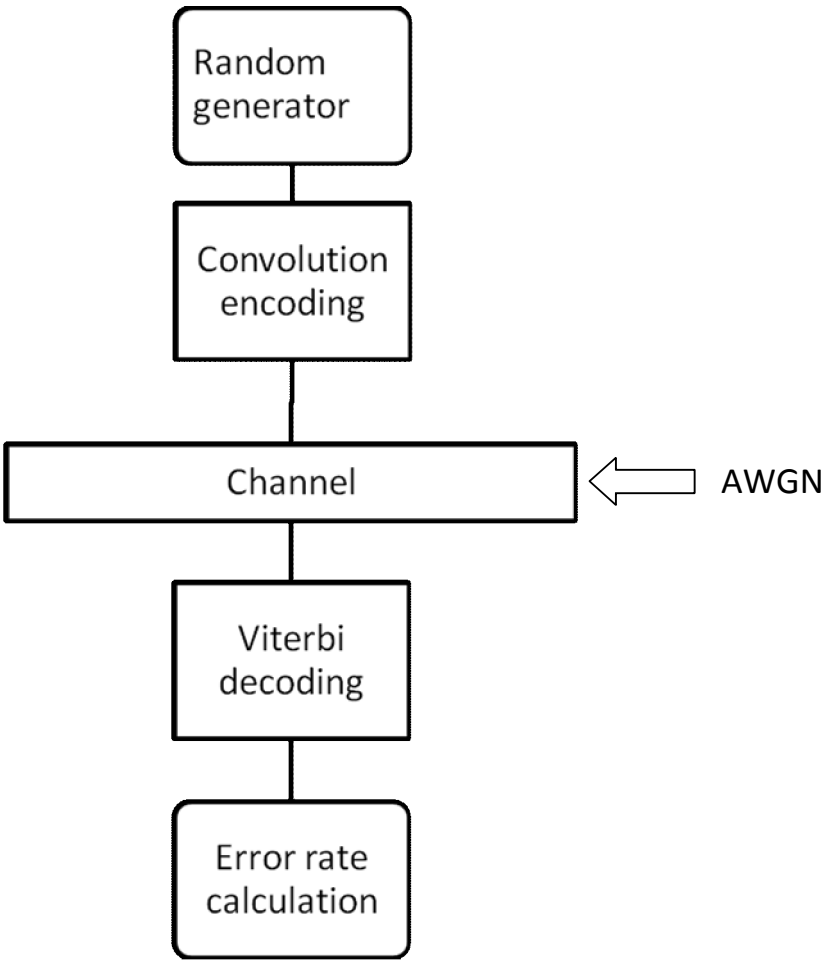


Figure 3.4: Convolution Coding Flowchart

(1) Generating the data: The data to be transmitted through the channel is generated using *random integer* generator. It generates random uniformly distributed integers in the range [0, M-1].

(2) Convolution encoding: This convolutionally encodes binary data. This is achieved by the *poly2trellis* function. It accepts a polynomial description of a convolution encoder and returns the corresponding trellis structure description. The parameters for the *poly2trellis* function are "constraint length" and "code generator polynomial".

(3) Adding noise to the transmitted symbols: The AWGN function adds White Gaussian Noise to the channel symbols produced by the encoder. The parameters for this function are the coded symbols, SNR (signal to noise ratio), the state, and the power type (whether in "dB" or "linear").

(4) Decoding: The decoding is done using the Viterbi algorithm to decode convolutionally encoded data.

(5) Error rate calculation: This is used for calculating the number of errors, number of symbols compared and the bit error rate. It computes the error rate of the received data by comparing it to a delayed version of the transmitted data.

Figures 3.5 and 3.6 show the MATLAB®2008/Simulink representative models for convolutionally coded system and an uncoded system respectively. First the information bits are generated using a random number generator. The randomly generated data is then sent to the convolution encoder and encoded into code words, which are transmitted over the noisy channel. Before transmitting, the encoded data is modulated using Binary Phase Shift Keying (BPSK) method.

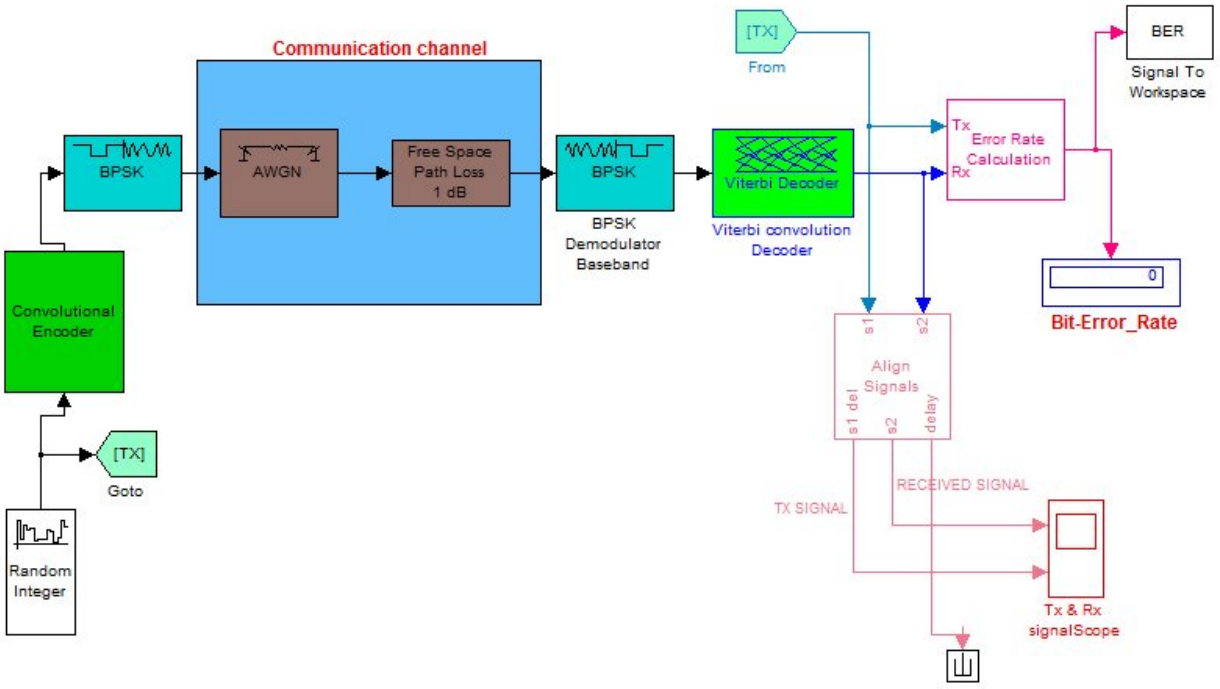


Figure 3.5: Convolution Coded System

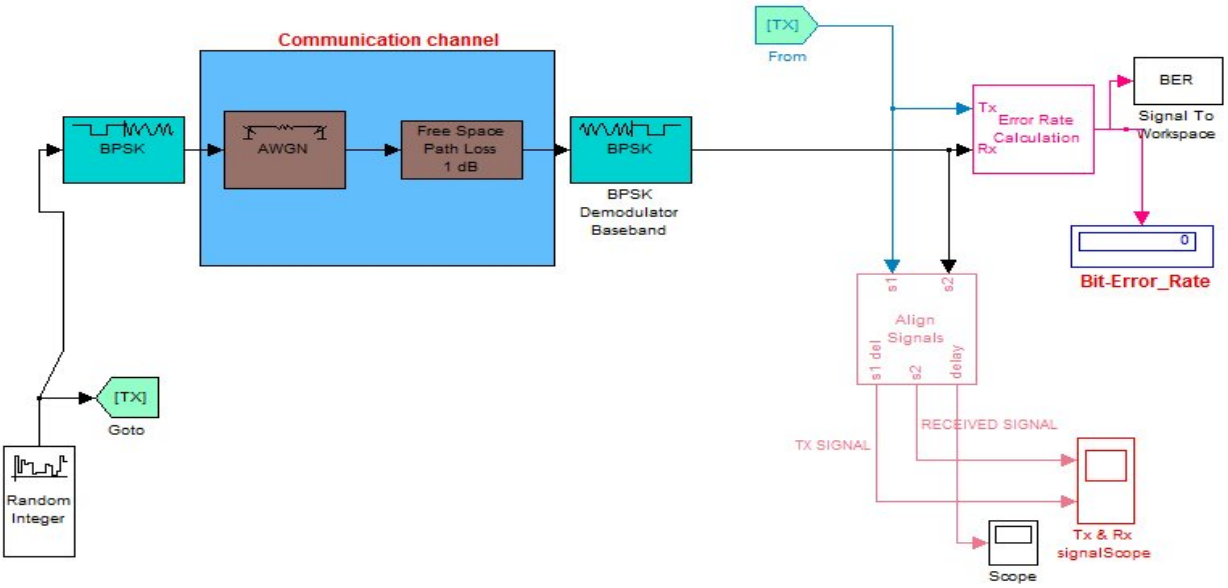


Figure 3.6: An Uncoded System

3.4 SIMULINK BLOCKS

Simulink provides a graphical user interface (GUI) for building models as block diagrams. After a model is defined and represented by the block diagram, it is simulated. Using scopes and other display blocks, the simulation results can be viewed while the simulation is running. In addition, many parameters can be changed and the effect seen with "what if" exploration. The simulation results can be put in the MATLAB® 2008 workspace for post processing and visualization. The blocks used in the study are as follows:

1. Random Integer Generator: The Random Integer Generator block generates uniformly distributed random integers in the range [0, M-1], where M is the M-ary number defined in the dialog box. The M-ary number can be either a scalar or a vector. In the model, the information (in the form of random integers) transmitted from source to destination is an abstract conception. In the real world, it can be images, computer data, voice, e.t.c

2. Convolution Encoder: This block creates convolution codes from binary data by encoding a sequence of binary input vectors to produce a sequence of binary output vectors. It can also process multiple symbols.

3. The BPSK Modulator Baseband block: This block modulates signals using the binary phase shift keying method. The output is a baseband representation of the modulated signal. For both integer and bit inputs, this block can accept the data types int8, uint8, int16, uint16, int32, uint32, Boolean, single, and double. The input must be a discrete-time binary-valued signal.

4. The AWGN Channel block: This block adds White Gaussian Noise to a real or complex input signal. When the input signal is real, this block adds real Gaussian noise and produces a real output signal. When the input signal is complex, this block adds complex Gaussian noise and produces a complex output signal. It inherits its sample time from the input signal. The Initial seed parameter in this block initializes the noise generator. Initial seed can be either a scalar or a vector whose length matches the number of channels in the input signal. The signal inputs can only be of type single or double. The port data types are inherited from the signals that drive the block.

5. The Free Space Path Loss block: This block simulates the loss of signal power due to the distance between transmitter and receiver. The block reduces the amplitude of the input signal by an amount that is determined in either of two ways:

- By the Distance (km) and Carrier frequency (MHz) parameters
- By the Loss (dB) parameter

6. The BPSK Demodulator Baseband block: This block demodulates the signal that was modulated using the binary phase shift keying method. The input is a baseband representation of the modulated signal. The input can be either a scalar or a frame-based column vector. The block can accept the data types single and double.

7. The Viterbi Decoder block: This block decodes input symbols to produce binary output symbols. It uses the Viterbi algorithm to decode convolutionally encoded input data. It can process several symbols at a time for faster performance.

8. The Align Signals block: The block aligns a signal with a delayed, and possibly distorted, version of itself. The block is particularly useful when there is a need for comparing a transmitted and received signal to find the bit error rate, but the delay in the received signal is unknown. The input port labeled s1 receives the original signal, while the input port labeled s2 receives the delayed version of the signal. The two input signals must have the same sample times. The block calculates the delay between the two signals, and then

- Delays the first signal, s1, by the calculated value, and outputs it through the port labeled s1 del.
- Outputs the second signal s2 without change through the port labeled s2.
- Outputs the delay value through the port labeled delay.

9. The Error Rate Calculation block: This block computes error rate of the received data by comparing it to a delayed version of the transmitted data. The block output is a three-element vector consisting of the error rate, followed by the number of errors detected and the total number of symbols compared. This vector can then be sent to either the workspace or an output port. The stop simulation option stops the simulation upon detecting a target number of errors or a maximum number of symbols, whichever comes first.

10. The Signal to Workspace block: This is a sink block that writes data from the simulation into an array in the MATLAB® 2008 main workspace. However, the data is not available until simulation is stopped or paused. The output array can be 2 dimension (2-D) or 3 dimension (3-D), depending on whether the data is 1 dimension (1-D), sample based, or frame based.

11. The Display block: This block shows the value of its input on its icon. The amount of data displayed and the time steps at which the data is displayed are determined by block parameters.

12. The Scope block: This block displays signals generated during a simulation. It displays its input with respect to simulation time. The Scope block can have multiple axes (one per port); all axes have a common time range with independent y-axes. The Scope allows you to adjust the amount of time and the range of input values displayed.

CHAPTER FOUR

RESULTS AND ANALYSIS

4.0 SIMULATION RESULTS

Computer simulations were used in this research to evaluate the performance of the error correction scheme presented in the last three chapters.

Simulation results of the error correction using convolution codes are based on Bit Error Rate (BER) performance over a range of signal-to-noise ratios (Eb/No). The error rates of the received data were computed by comparing it to a delayed version of the transmitted data. This comparison (as in Table 4.1) was done for the uncoded system and two different codes:

(1) Rate 1/2 code

(2) Rate 1/3 code

Table 4.1: The Convolution Encoder Characteristics used in the Simulation

RATE	CONSTRAINT LENGTH	GENERATOR POLYNOMIAL 1(octal form)	GENERATOR POLYNOMIAL 2(octal form)	GENERATOR POLYNOMIAL 3(octal form)
1/2	3	6	7	-
1/3	3	4	5	7

The results were captured from the Simulink model and then tabulated. The tabulated result includes the bit error rates of the two codes and that of the uncoded system, the errors detected and the total number of symbols compared. Table 4.2 shows results obtained.

Table 4.2: Simulation Results

	Coded System						Uncoded System		
S/N0	Code rate ½			Code rate 1/3				No. of Errors Detected	No. of Symbols Compared
	BER	No. of Errors Detected	No. of Symbols Compared	BER	No. of Errors Detected	No. of Symbols Compared	BER		
1	0.05	2	40	0	0	40	0.575	23	40
2	0.0488	2	41	0.02	1	41	0.585	24	41
3	0.0476	2	42	0.05	2	42	0.571	24	42
4	0.0465	2	43	0.05	2	43	0.581	25	43
5	0.0455	2	44	0.05	2	44	0.568	25	44
6	0.0444	2	45	0.04	2	45	0.578	26	45
7	0.0435	2	46	0.04	2	46	0.565	26	46
8	0.0426	2	47	0.04	2	47	0.574	27	47
9	0.0417	2	48	0.06	3	48	0.583	28	48
10	0.0408	2	49	0.06	3	49	0.571	28	49

4.1 RESULTS DISCUSSION

From the simulation results shown in Table 4.2, it can be seen that with an increase in symbols, there is an increase in errors. The start points of the BER for the 1/2 and 1/3 codes are 0.05 and 0.0 respectively. These decrease moderately to the end points of 0.0408 and 0.06. As compared to the uncoded system which has a BER start point of 0.575 and end point of 0.571, this is about 50% more than the coded system.

4.2 SIMULATION PLOTS

Monte-Carlo technique was used in plotting graphs of bit error-rate (BER) versus signal to noise ratio (E_b/N_o) in order to further characterize the performance of the convolution codes. The simulation results are shown in the Figures 4.1 to 4.5.

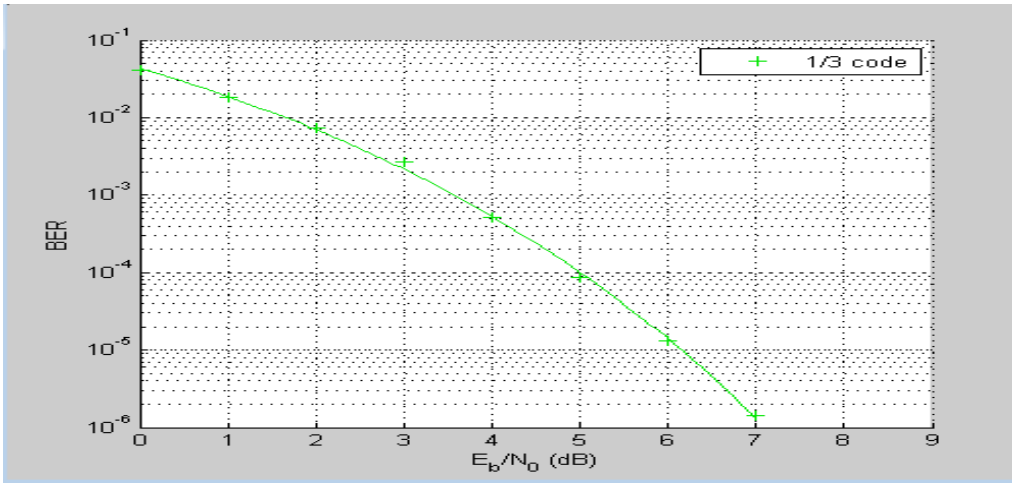


Figure 4.1: Simulation Result for 1/3 Convolution Code

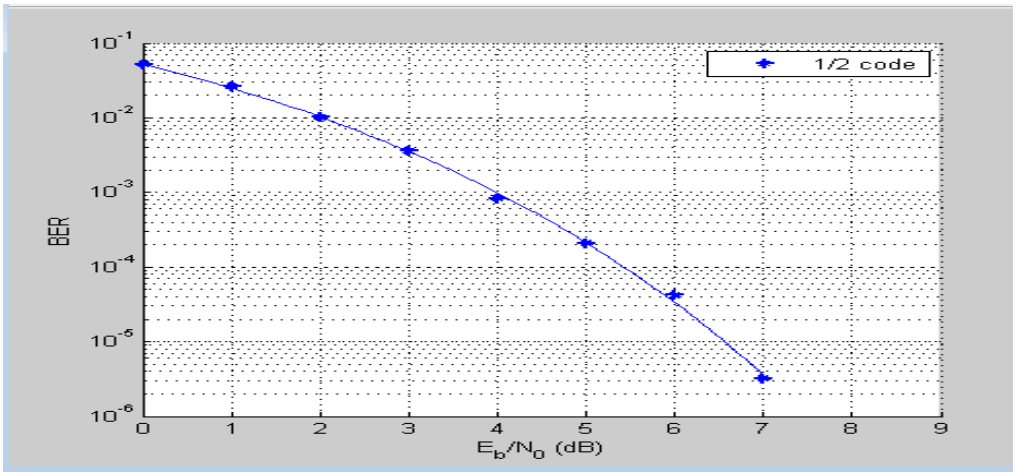


Figure 4.2: Simulation Result for rate 1/2 Convolution Code

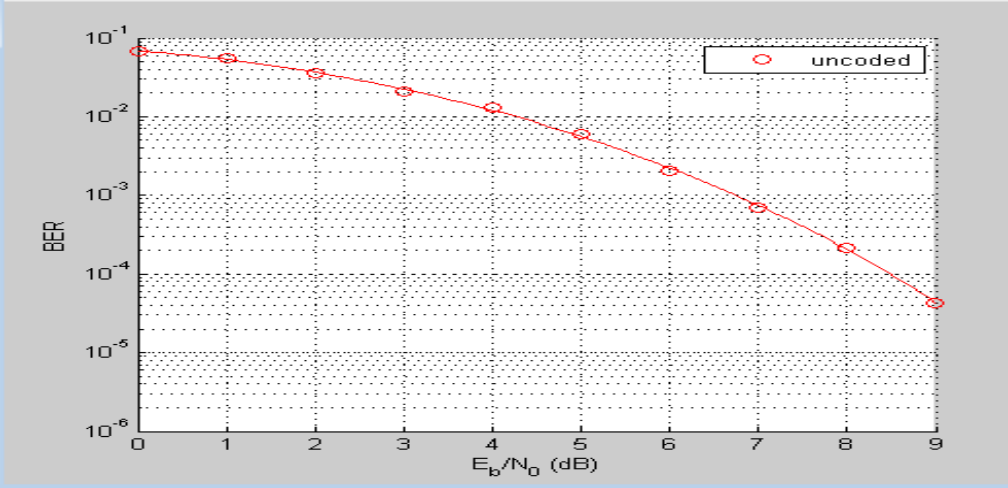


Figure 4.3: Simulation Result for Uncoded System

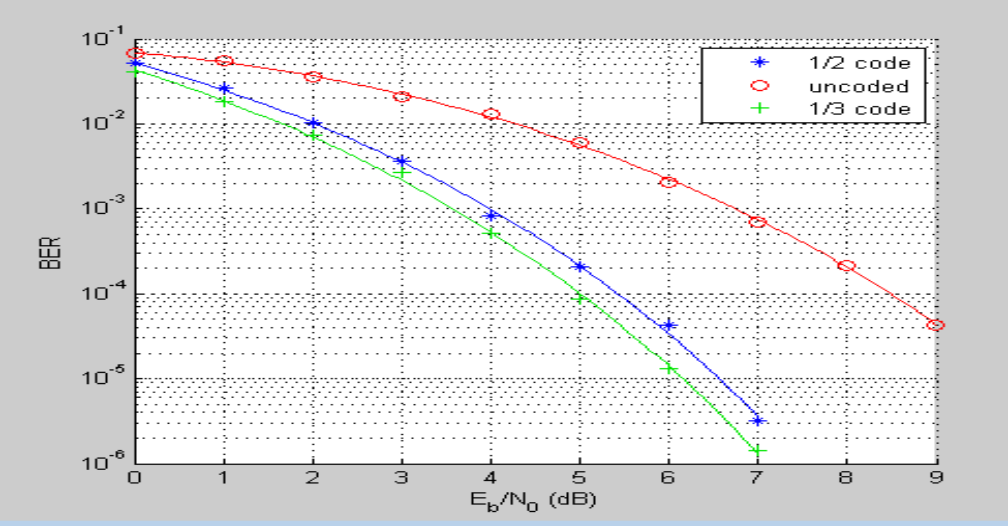


Figure 4.4: Simulation Result for 1/2, 1/3 and Uncoded Systems

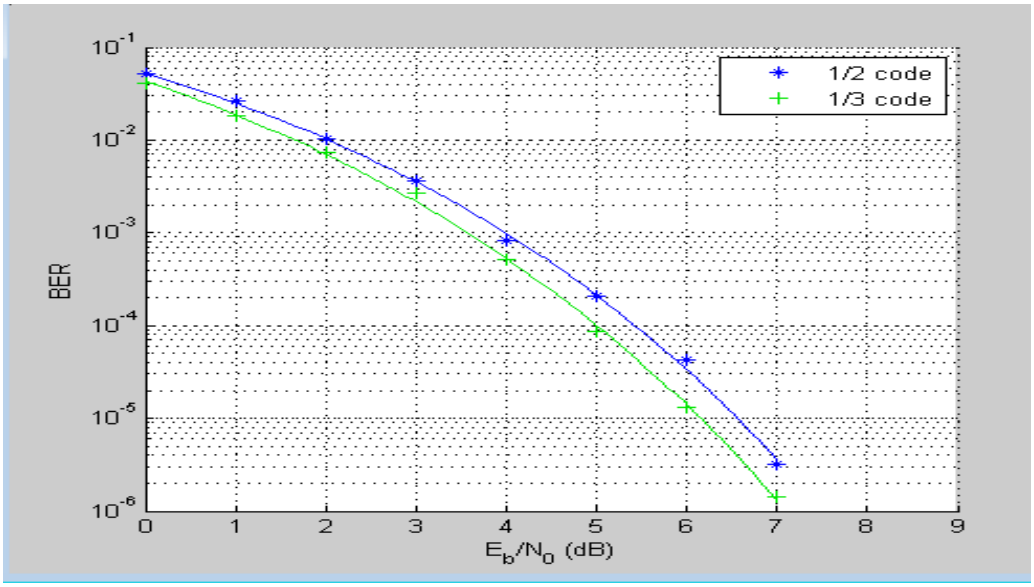


Figure 4.5: Simulation Result for 1/2 Code and 1/3 Code

The vertical axis shows the bit error rate (BER) and horizontal axis shows the signal-to-noise ratio (E_b/N_o). The signal to noise ratio was varied from 0dB to 9dB.

The plots are summarized as follows:

a) Error correction system using rate 1/3 convolution encoder is shown in Figure 4.1.

b) Error correction system using rate 1/2 convolution encoder is shown in Figure 4.2.

c) The system without error correction is shown in Figure 4.3.

d) Comparison between rates 1/2, 1/3 and uncoded systems is shown in figure 4.4

e) Comparison between 1/2 code rate and 1/3 code rate coded systems is shown in figure 4.5.

From Figures 4.1, 4.2, and 4.3 it can be seen that the curves depend upon E_b/N_o. The curves show that BER decreases with an increase in E_b/N_o.

A comparison of the effect of convolution coding for the two different convolution code rates (1/2 and 1/3) and an uncoded system is displayed in Figure 4.4. The plots show start points of the BER for the 1/2 and 1/3 codes are at 0.0520 and 0.0409 respectively. These decrease moderately to the end points of 3.21e-6 and 1.41e-6. As compared to the uncoded system whose BER starts at 0.0689 and ends at 4.24e-5. For example, coding with rate 1/2 results in a saving of about 3dB and with rate 1/3 more than 3.5 dB

(example is for BER = 10^{-4}). Thus to operate a system at 10^{-4} BER, over 8 dB would be needed without error correction or only about 5 dB for 1/3 code rate and 5.5 dB for 1/2 code rate with error correction. The coding gains of 2.131 dB and 3.011 dB for 1/2 code rate, 2.556 and 3.435 for 1/3 code rate were exhibited over the system without error correction. The reference points for these coding gains are at Bit Error Rates of 10^{-2} and 10^{-4} respectively.

Figure 4.5 shows a comparison between 1/3 and 1/2 rates. 1/3 requires 0.354dB and 0.424 dB less of signal to noise ratio than 1/2 to achieve the same BER at reference points of 10^{-3} and 10^{-4} respectively making the 1/3 code performance better than the 1/2 code. Thus it can be said that the coded system outperforms the uncoded system because it gives a lower BER for the same number of symbols being compared. It is evident that the introduction of the convolution coding scheme has helped to decrease the Bit Error Rate significantly.

4.3 SIGNIFICANCE OF RESULT

The comparative analysis of the rate 1/2 code, the rate 1/3 code and the uncoded systems were shown in Figures 4.1 to 4.5. With a rate 1/2 and 1/3 convolution coding and a constraint length of 3, a data signal can be transmitted with at least 3 dB less power. This can reduce the transmitter or antenna cost, or permits increased data rates for the same transmitter power and antenna size. Higher coding rate results in a better BER at the expense of more bits being processed and transmitted. When more bits are being processed, the processing time will be larger. However, a rate 1/2 and 1/3 coding results in an increase of bandwidth by a factor of 2 and 3 respectively. In general, the bandwidth expansion factor of a convolution code is simply *n/k*, where *k/n* is the code rate expressed as a ratio of the number of bits in the convolution encoder (*k*) to the number of channel symbols output by the convolution encoder (*n*) in a given encoder cycle. Slower transmission (rate <1) is the main price paid for nearly error-free transmission and a code defines a trade-off between the two.

CHAPTER FIVE

SUMMARY, CONCLUSION AND RECOMMENDATIONS

5.1 INTRODUCTION

Error correction coding is a key element of any digital wireless communication system, minimizing the effects of noise and interference on the transmitted signal at the physical layer. Error prone communications systems are unreliable, unreliable services are often discarded. There is therefore the need to improve the reliability of such services. Since it is practically impossible to build all potential designs to test and compare performances, it is best to obtain mathematical models that describe the behavior of different parts of the considered systems, run computer simulations to obtain estimates of the results. Computer simulations use mathematical models to imitate the functionality of a real process. They are also used to compare the performance of alternative designs before implementation.

5.2 SUMMARY

The research carried out in this work provides a simulating tool that outlines efficiently error correction in wireless communication systems with the use of convolution codes. Convolution code is widely used in digital wireless communication system for detecting and correcting errors in received signal message bits. Simulations for two rates of convolution codes have been presented and the bit error rate for each type was determined.

5.3 LIMITATIONS

Due to low frequency of errors as Eb/No increases the amount of time it will take to gather enough data to accurately measure the BER is monumental. Hence wise estimation and extrapolation techniques need to be used to make a precise estimation of BER in a short duration. Other concerns include computational complexity, encoding and decoding delays. Longer decoding depths do not provide significant BER improvement. Accurate characterization by Monte Carlo simulation at very low BER e.g. less than 10^{-10}, is not computationally feasible.

5.4 CONCLUSION

With the detailed description of the convolution code encoder and decoder presented in Chapters 2 and 3 respectively, the performance of convolution codes was investigated through extensive computer simulation. To validate the convolution codes simulation, comparisons were made between the coded and uncoded bit error rate (BER) results. Based on the results of the graphs, it is evident that there was a 70% and 74% improvement on coding gains between reference points of 10^{-2} and 10^{-4} for the two code

rates used. This improvement can be attributed to the introduction of the convolution codes. In general, it is observed that the introduction of the convolution coding scheme has helped to decrease the Bit Error Rate significantly which in turn will result in the transmission of signals of specified quality with a smaller transmit power. In other words, it leads to higher power efficiency, but on the other hand, the bit rate is half (or third) that of the uncoded scheme thus lowering bandwidth efficiency. It can thus be concluded that a coded system offers better channel efficiency than the uncoded system.

5.5 RECOMMENDATIONS FOR FURTHER WORK

The following are suggested for further work:

(1) Implementation of other versions of error correction codes in order to meet different design requirements can improve this work. This will involve investigation of BER performance for other coding schemes e.g. Block codes; repetition codes, parity codes *n*-redundancy codes e.t.c, using other code rates and using real time data such as images, voice and so on to validate the accuracy and authenticity of the developed correction model.

(2) Hardware Implementation of the coding scheme used in this research using real data is also another area that needs further investigation. This can be achieved by implementing the "real-life" part of the thesis; building a real-life communication system to verify the functionality and the results obtained through the simulated system.

(3) Additional MATLAB®2008 functions can be created to process various types of applications (images, speech, texts, and video e.t.c) in place of randomly generated signals, however some applications may require larger processing time.

REFERENCES

Arasteh D. (2006). Teaching Convolution Coding Using MATLAB®2008 in Communication Systems. Proceedings of the ASEE Gulf South-west Annual Conference Southern University, Baton Rouge, LA USA.

Asif S, Abdullah S.M and Anisul Islam, A.M. (2009). Comparison of BER between Uncoded Signal and Coded Signal (Using Convolution Code) over Slow Rayleigh Fading Channel. Journal of Theoretical and Applied Information Technology 2005-2009. Through: http://www.jatit.org

Babale, S.A. (2010). Modeling and Analysis of System Capacity against Radio Frequency Impairment. Unpublished M.Sc Thesis. Department of Electrical and computer Engineering, Ahmadu Bello University, Zaria, December, 2011.

Balakrishnan G, Yang M, Jiang Y and Kim Y. (2007). Performance analysis of error control codes for wireless sensor networks. Paper presented at the Information Technology (ITNG 2007) 4th International Conference. Retrieved 3rd October, 2009. Pp 876-879.

Bystrom, M. and Modestino, W. (1997). Combined source-channel coding schemes for video transmission over an additive white Gaussian noise channel. Journal of Center for Image Processing Research, Rensselaer Polytechnic Institute, Troy, NY 1218.

Cheng, L. (2005). Pruned Convolution codes and Viterbi decoding with levenshtein distance metric. Journal of the Department of Electrical and Electronics Engineering, University of Johannesburg, Johannesburg, South Africa. 97(2). 140-146

Cheong, M. Y. (2007). Block Error Correction Codes and Convolution Codes. Postgraduate course in Radio communication, Wireless Local Area Network (WLAN), Helsinki University of Technology. S-72.333.

Du, K and Swamy, M.N.S. (2009). Wireless Communication System. Cambridge University Press. Pp 3.

Er. Liu. (2004). Convolution Coding and Viterbi Algorithm. Postgraduate seminar on Radio Communications S-72.333. Helsinki University of Technology, Porland.

Hosseinzadeh, D, April K and Sri K. (2004). Source and Channel Coding Techniques for Faithful transmission of Digital Mammograms. *IEEE* Canadian review (Autumn, 2004).

Houghton, A.D. (1997). The Engineers Error Coding Handbook. Chapman & Hall, London, UK. 1st edition, 1997. Pp134-141

Huang, F. (1997). Evaluation of Soft Output Decoding for Turbo Codes. M.Sc thesis. Faculty of the Virginia Polytechnic Institute and State University. Retrieved 21st March, 2010.

Jemibewon, A. (2000). A smart implementation of turbo decoding for improved power efficiency. M.Sc Thesis. Faculty of the Virginia Polytechnic Institute and State University. Retrieved 21st March, 2010.

Ma. F. and Knight J. (2001). Convolution Codes. Laboratory. 97.478. Fall 2001. Retrieved on 22/01/2011 through: http://www.informatik.uni-trier

Mahmood A. (2008). Method to Improve Channel Coding Using Cryptography. Department of Electrical Engineering and Computer Science, the University of Siegen, Siegen, Germany.

MATLAB® 2008 7.1 (2005). Convolution encoder. Communication Toolbox. Retrieved through: www.mathworks.com/access/helpdesk/help/toolbox/commblks

Mohammad, S. M. and Salem, S. (2010). Bit Error Rate Analysis for BPSK Modulation in Presence of Noise and Two Co-channel Interferer. IJCSNS. 10(5).152-156

Morales, C. A. (2008). VLSI Architectures for Turbo Codes Adapted for the Standard Mobile Communications WCDMA (3GPP TS 25.212). M.Sc. Thesis, University of Guadalajara, Mexico.

Qian N. (1999). Performance Gain of Convolution Encoding vs. Direct Spreading in the CDMA Schemes. A research paper, A & M University, Texas. Retrived on 6th February, 2011 through: http://www.utdallas.edu

Rossi, R. (1998). Digital Channel Error Correction Coding Design Tools. Synthesis Research Inc., SR – TN012

Saha J. (2003). Error correcting codes. Group theory seminar presentation. Open computing facility, University of Califonia. Fall, 2003. Retrived on 23rd November, 2009 through: http://www.ocf.berkeley.edu

Sankar K. (2009). Convolution Code. Signal Processing for Communication article. Retrieved on 07/10/09 through: http://www.dsplog.com/2009/01/04/convolutioncode

Shelton, C.P. (1999). Coding for Error Detection and Correction. Dependable Embedded Systems, Carnegie Mellon University. 18(84). 9b.

Shooshtary, S. (2008). Development of a MATLAB®2008 Simulation Environment for Vehicle-to-Vehicle and Infrastructure Communication. M.Sc Thesis. Vienna University of Technology, Austria.

Error Correcting Codes.(2008). Signal and information processing laboratory, Swiss Federal Institute of Technology. Retrieved on 12th December, 2008.

Thomas, C; Mark, A. B; Linda, M. D; Thomas, P. B; David, G. and Chris N. (2003). Integrated Circuits for Channel Coding in 3G Cellular Mobile Wireless Systems. *IEEE* Communications Magazine (August, 2003).

Valenta, V. (2000). Error performance of BPSK and QPSK in Satellite Communications. Department of Electrical and Electronic Technology, FEEC, BUT.

Valenti, M. C. (1999). The Evolution of Error Control Coding. Department of Electrical and Computer Engineering, Virginia Tech, USA.

Vats, D. and Kacewicz, A. (2005). Error-Correcting Codes on Cellular Phones. M.Sc Thesis. Dept. of Electrical and Computer Engineering, Cornell University, Ithaca, NY.

Wang, X and Poor, V. H. (2003). Wireless Communication Systems: Advanced Techniques for Signal Reception. Prentice Hall Publishing. 2004, from inform IT Network. Through: http://www.informit.com

Wesel, R. (2003). Convolution Codes. Online Encyclopedia of Telecommunications. Retrieved on 23/01/2011 through: www.ee.ucla.edu/~**wesel**/documents

APPENDIX A: Simulation results for 100 data points

	Coded System						Uncoded System		
S/NO	Code rate ½			code rate 1/3					
	BER	No. of Errors Detected	No. of Symbols Compared	BER	No. of Errors Detected	No. of Symbols Compared	BER	No. of Errors Detected	No. of Symbols Compared
1	0.0476	1	21	0	0	21	0.571	12	21

2	0.0909	2	22	0	0	22	0.591	13	22
3	0.087	2	23	0	0	23	0.609	14	23
4	0.0833	2	24	0	0	24	0.625	15	24
5	0.08	2	25	0	0	25	0.6	15	25
6	0.0769	2	26	0	0	26	0.615	16	26
7	0.0741	2	27	0	0	27	0.593	16	27
8	0.0714	2	28	0	0	28	0.571	16	28
9	0.069	2	29	0	0	29	0.552	16	29
10	0.0667	2	30	0	0	30	0.567	17	30
11	0.0645	2	31	0	0	31	0.548	17	31
12	0.0625	2	32	0	0	32	0.531	17	32
13	0.0606	2	33	0	0	33	0.545	18	33
14	0.0588	2	34	0	0	34	0.559	19	34
15	0.0571	2	35	0	0	35	0.571	20	35
16	0.0556	2	36	0	0	36	0.583	21	36
17	0.0541	2	37	0	0	37	0.595	22	37
18	0.0526	2	38	0	0	38	0.579	22	38
19	0.0513	2	39	0	0	39	0.59	23	39
20	0.05	2	40	0	0	40	0.575	23	40
21	0.0488	2	41	0.02	1	41	0.585	24	41
22	0.0476	2	42	0.05	2	42	0.571	24	42
23	0.0465	2	43	0.05	2	43	0.581	25	43
24	0.0455	2	44	0.05	2	44	0.568	25	44
25	0.0444	2	45	0.04	2	45	0.578	26	45

26	0.0435	2	46	0.04	2	46	0.565	26	46
27	0.0426	2	47	0.04	2	47	0.574	27	47
28	0.0417	2	48	0.06	3	48	0.583	28	48
29	0.0408	2	49	0.06	3	49	0.571	28	49
30	0.04	2	50	0.06	3	50	0.56	28	50
31	0.0392	2	51	0.06	3	51	0.569	29	51
32	0.0385	2	52	0.06	3	52	0.558	29	52
33	0.0377	2	53	0.06	3	53	0.547	29	53
34	0.037	2	54	0.06	3	54	0.537	29	54
35	0.0364	2	55	0.05	3	55	0.545	30	55
36	0.0357	2	56	0.05	3	56	0.554	31	56
37	0.0351	2	57	0.05	3	57	0.544	31	57
38	0.0345	2	58	0.05	3	58	0.552	32	58
39	0.0339	2	59	0.05	3	59	0.542	32	59
40	0.0333	2	60	0.05	3	60	0.55	33	60
41	0.0492	3	61	0.05	3	61	0.557	34	61
42	0.0645	4	62	0.05	3	62	0.548	34	62
43	0.0635	4	63	0.05	3	63	0.556	35	63
44	0.0781	5	64	0.05	3	64	0.563	36	64
45	0.0923	6	65	0.05	3	65	0.569	37	65
46	0.0909	6	66	0.05	3	66	0.576	38	66
47	0.0896	6	67	0.04	3	67	0.582	39	67
48	0.0882	6	68	0.04	3	68	0.574	39	68
49	0.087	6	69	0.04	3	69	0.58	40	69

50	0.0857	6	70	0.04	3	70	0.571	40	70
51	0.0845	6	71	0.04	3	71	0.577	41	71
52	0.0833	6	72	0.04	3	72	0.583	42	72
53	0.0822	6	73	0.04	3	73	0.575	42	73
54	0.0811	6	74	0.04	3	74	0.581	43	74
55	0.08	6	75	0.04	3	75	0.587	44	75
56	0.0789	6	76	0.04	3	76	0.592	45	76
57	0.0779	6	77	0.04	3	77	0.584	45	77
58	0.0769	6	78	0.04	3	78	0.59	46	78
59	0.0759	6	79	0.04	3	79	0.595	47	79
60	0.075	6	80	0.04	3	80	0.6	48	80
61	0.0741	6	81	0.04	3	81	0.593	48	81
62	0.0732	6	82	0.04	3	82	0.598	49	82
63	0.0723	6	83	0.04	3	83	0.59	49	83
64	0.0714	6	84	0.04	3	84	0.583	49	84
65	0.0706	6	85	0.04	3	85	0.576	49	85
66	0.0698	6	86	0.03	3	86	0.57	49	86
67	0.069	6	87	0.03	3	87	0.575	50	87
68	0.0682	6	88	0.03	3	88	0.568	50	88
69	0.0674	6	89	0.03	3	89	0.573	51	89
70	0.0667	6	90	0.03	3	90	0.578	52	90
71	0.0659	6	91	0.03	3	91	0.582	53	91
72	0.0652	6	92	0.03	3	92	0.576	53	92
73	0.0645	6	93	0.03	3	93	0.57	53	93

74	0.0638	6	94	0.03	3	94	0.574	54	94
75	0.0632	6	95	0.03	3	95	0.579	55	95
76	0.0625	6	96	0.03	3	96	0.583	56	96
77	0.0619	6	97	0.03	3	97	0.588	57	97
78	0.0612	6	98	0.03	3	98	0.592	58	98
79	0.0606	6	99	0.03	3	99	0.586	58	99
80	0.06	6	100	0.03	3	100	0.59	59	100
81	0.0594	6	101	0.03	3	101	0.584	59	101
82	0.0588	6	102	0.03	3	102	0.578	59	102
83	0.0583	6	103	0.03	3	103	0.573	59	103
84	0.0577	6	104	0.03	3	104	0.577	60	104
85	0.0571	6	105	0.03	3	105	0.571	60	105
86	0.0566	6	106	0.04	4	106	0.566	60	106
87	0.0561	6	107	0.04	4	107	0.57	61	107
88	0.0556	6	108	0.04	4	108	0.565	61	108
89	0.055	6	109	0.04	4	109	0.569	62	109
90	0.0545	6	110	0.04	4	110	0.564	62	110
91	0.0541	6	111	0.04	4	111	0.568	63	111
92	0.0536	6	112	0.04	5	112	0.571	64	112
93	0.0531	6	113	0.04	5	113	0.566	64	113
94	0.0526	6	114	0.04	5	114	0.57	65	114
95	0.0522	6	115	0.04	5	115	0.574	66	115
96	0.0517	6	116	0.04	5	116	0.578	67	116
97	0.0513	6	117	0.04	5	117	0.581	68	117

98	0.0508	6	118	0.04	5	118	0.585	69	118
99	0.0504	6	119	0.04	5	119	0.58	69	119
100	0.05	6	120	0.04	5	120	0.583	70	120

APPENDIX B: Generated MATLAB® 2008 Codes

Generation of the rate 1/2 code Model

```
EbNo = 0:9; linEbNo = 10.^(EbNo(:).*0.1);

M = 4; codeRate = 1/2; constlen = 3; k = log2(M); codegen = [6 7];

tblen = 34;     % traceback length

trellis = poly2trellis(constlen, codegen);

dspec = distspec(trellis, 7);

expVitBER = bercoding(EbNo, 'conv', 'hard', codeRate, dspec);

 numSymb = 100; numPlot = 20;
```

```
Nsamp = 4;      % oversampling rate
EbNoDemo = 3; EsN0 = EbNoDemo + 10*log10(k);
seed = [37];
rand('state', seed(1));
msg_orig = randsrc(numSymb, 1, 0:1);
 [msg_enc_bi] = convenc(msg_orig, trellis);
numEncPlot = numPlot / codeRate; tEnc = (0:numEncPlot-1) * codeRate;
 randn('state');
msg_enc = bi2de(reshape(msg_enc_bi, ...
   size(msg_enc_bi,2)*k,size(msg_enc_bi,1) / k)');
grayencod = bitxor(0:M-1, floor((0:M-1)/2));
msg_gr_enc = grayencod(msg_enc+1);
msg_tx = pskmod(msg_gr_enc, M, pi/4);
msg_tx = rectpulse(msg_tx, Nsamp);
msg_rx = awgn(msg_tx, EsN0-10*log10(1/codeRate)-10*log10(Nsamp));
 msg_rx_int = intdump(msg_rx, Nsamp);
msg_gr_demod = pskdemod(msg_rx_int, M, pi/4);
[dummy graydecod] = sort(grayencod); graydecod = graydecod - 1;
msg_demod = graydecod(msg_gr_demod+1)';
msg_demod_bi = de2bi(msg_demod,k)'; msg_demod_bi = msg_demod_bi(:);
 msg_dec = vitdec(msg_demod_bi, trellis, tblen, 'cont', 'hard');
 [nChnlErrs BERChnl] = biterr(msg_enc, msg_demod);
[nCodErrs BERCoded] = biterr(msg_orig(1:end-tblen), msg_dec(1+tblen:end));
 semilogy(EbNo, expVitBER, 'g');  xlabel('Eb/No (dB)'); ylabel('BER');
title('Performance for R=1/2, K=3 Convolution Code');
```

```
grid on; axis([0 9 10e-7 10e-2]); legend('1/2 code', 0);
```

Generation of the rate 1/3 code Model

```
EbNo = 0:9; linEbNo = 10.^(EbNo(:).*0.1);
M = 4; codeRate = 1/3; constlen = 3; k = log2(M); codegen = [4 5 7];
tblen = 34;     % traceback length
trellis = poly2trellis(constlen, codegen);
dspec = distspec(trellis, 7);
expVitBER = bercoding(EbNo, 'conv', 'hard', codeRate, dspec);
 numSymb = 100; numPlot = 20;
Nsamp = 4;      % oversampling rate
EbNoDemo = 3; EsN0 = EbNoDemo + 10*log10(k);
seed = [37];
rand('state', seed(1));
msg_orig = randsrc(numSymb, 1, 0:1);
 [msg_enc_bi] = convenc(msg_orig, trellis);
numEncPlot = numPlot / codeRate; tEnc = (0:numEncPlot-1) * codeRate;
 randn('state');
msg_enc = bi2de(reshape(msg_enc_bi, ...
   size(msg_enc_bi,2)*k,size(msg_enc_bi,1) / k)');
grayencod = bitxor(0:M-1, floor((0:M-1)/3));
msg_gr_enc = grayencod(msg_enc+1);
msg_tx = pskmod(msg_gr_enc, M, pi/4);
msg_tx = rectpulse(msg_tx, Nsamp);
msg_rx = awgn(msg_tx, EsN0-10*log10(1/codeRate)-10*log10(Nsamp));
```

```
 msg_rx_int = intdump(msg_rx, Nsamp);
msg_gr_demod = pskdemod(msg_rx_int, M, pi/4);
[dummy graydecod] = sort(grayencod); graydecod = graydecod - 1;
msg_demod = graydecod(msg_gr_demod+1)';
msg_demod_bi = de2bi(msg_demod,k)'; msg_demod_bi = msg_demod_bi(:);
 msg_dec = vitdec(msg_demod_bi, trellis, tblen, 'cont', 'hard');
 [nChnlErrs BERChnl] = biterr(msg_enc, msg_demod);
[nCodErrs BERCoded] = biterr(msg_orig(1:end-tblen), msg_dec(1+tblen:end));
 semilogy(EbNo, expVitBER, 'g');  xlabel('Eb/No (dB)'); ylabel('BER');
title('Performance for R=1/3, K=3 Convolution Code');
grid on; axis([0 9 10e-7 10e-2]); legend('1/3 code', 0);
```