Performance of Turbo-Coded DS-CDMA Systems in Fading and Burst Channels

by
Telex Magloire Ngatched Nkouatchah
University of Natal
2001

Submitted in fulfilment of the academic requirements for the degree of MScEng in the School of Electrical and Electronic Engineering, University of Natal, 2001.

Abstract

Turbo codes are a class of forward error correction (FEC) codes that offer energy efficiencies close to the limits predicted by information theory. The features of turbo codes include parallel code concatenation, recursive convolutional encoding, nonuniform interleaving, and an associated iterative decoding algorithm. The excellent performance of turbo codes explains why much of the current research is focused on applying turbo codes to different systems.

This dissertation first outlines a new simple criterion for stopping the iterative process of the turbo decoder for each individual frame immediately after the bits are correctly estimated and thus prevents unnecessary computations and decoding delay.

The dissertation then considers the performance of turbo coded DS-CDMA systems. The performance analysis begins with simulation results for turbo coded DS-CDMA over a multi-path Rayleigh fading channel. The channel is then modeled using the Gilbert-Elliott channel model and analytical expressions for the performance of the system are derived. The influence of various parameters such as the Doppler frequency, the signal-to-noise ratio threshold on the system performance are analyzed and investigated.

To God be the Glory

Preface

The research work discussed in this dissertation was performed by Mr. Telex Magloire Ngatched Nkouatchah, under the supervision of Professor Fambirai Takawira, at the University of Natal's School of Electrical and Electronic Engineering Centre for Radio Access Technologies, which is sponsored by Alcatel Altech Telecoms and Telkom South Africa Ltd through the Centre of Excellence Programme.

Parts of the work presented in this dissertation have been published in the IEE Electronics Letters and at two SATNAC conferences, and submitted for publication in IEEE Transactions in Vehicular Technology.

The whole dissertation, unless specifically indicated to the contrary in the test, is the author's work, and has not been submitted in part, or in whole to any other University.

As the candidate's supervisor, I have approved this dissertation for submission.

Signed: ..... Name: ..... Date: .....

Acknowledgements

First and foremost thanks goes to Professor Fambirai Takawira for his inspiration with the topic of this dissertation, careful guidance, constant support and the many encouragements during the past two years. His patience, critical insight and his advice were instrumental in the completion of this work. He has been an excellent supervisor for me and I certainly hope that his culture of excellence will inspire me.

Secondly, I wish to express my sincere appreciation to my famiiy and friends for their support, understanding, love and trust.

Thanks are also owed to Alcatel Altech Telecoms and Telkom South Africa for their valued financial support.

Finally, I would like to thank all my postgraduate friends for their assistance and for making our time together enjoyable.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF ACRONYMS

AMPS Advanced Mobile Phone System

APP A Posteriori Probability

AWGN Additive White Gaussian Noise (channel)

BCH Bose Ray-Chaudhuri and Hocquenghem (code)

BER Bit Error Rate

bps bits per second

BPSK Binary Phase Shift Keying (modulation)

BCJR Bahl Cocke Jelinek Raviv

BS Base Station

BVD Big Viterbi Decoder

CCSDS Consultative Committee for Space Data Systems

CD Compact Disk

CDMA Code Division Multiple Access

CDMA2000 Code Division Multiple Access by 2000

CDPD Cellular Digital Packet Data

CE Cross Entropy

CRC Cyclic Redundancy Check (code)

CSI Channel State Information

DFT Discrete Fourier Transform

DS Direct Sequence

DS-CDMA Direct Sequence Code Division Multiple Access

DSN Deep Space Network

DSSS Direct Sequence spread Spectrum

DVD Digital Versatile Disk

EIA Electronic Industry Association

ESA European Space Agency

FEC Forward Error Correction

FER Frame Error Rate

FIR Finite Impulse Response (filter)

FFT Fast Fourier Transform

GE Gilbert Elliott (channel)

GSM Global System for Mobile Communications

HAD Hard Decision Aided

HGA High Gain Antenna

ICC International Conference on Communications

IDTF Inverse Discrete Fourier Transform

IEEE Institute of Electrical and Electronics Engineering

IFFT Inverse Fast Fourier Transform

IHDA Improved Hard Decision Aided

i.i.d independent and identically distributed

IIR Infinite Impulse Response (filter)

IMT-2000 International Mobile Telecommunications by 2000

INMARSAT International Maritime Satellite Organization

INTELSAT International Telecommunications Satellite Organization

IS Interim Standard

IS-95 EIA Interim Standard for U.S Code Division Multiple Access

ISI InterSymbol Interference

ITU International Telecommunications Union

LGA Low Gain Antenna

LLR Log-Likelihood Ratio

LOS Line Of Site

MA Moving Average

MAI Multiple Access Interference

MAP Maximum A Posteriori

MF Matched Filter

MIP Multipath Intensity Profile

ML Maximum Likelihood

MLSE Maximum Likelihood Sequence Estimator

MMSE Minimum Mean-Square Error

MRC Maximal Ratio Combining

MS Mobile Station

NAMPS Narrowband Advanced Mobile Phone System

NASA National Aeronautics and Space Agency

NMT Nordic Mobile Telephone

PCCC Parallel Concatenated Convolutional Codes

PCS Personal Communication System

pdf probability density function

PHS Personal Handyphone System

PSD Power Spectrum Density

RM Reed Muller (code)

RMD RAM Mobile Data

RS Reed Solomon (code)

RSC Recursive Systematic Convolutional (code)

SCCC Serial Concatenated Convolutional Codes

SCR Sign Change Ratio

SDR Sign Difference Ratio

SI Side Information

SI Self Interference

SISO Soft-Input, Soft-Output

SNR Signal to Noise power Ratio

SOVA Soft-Output Viterbi Algorithm

SSS Strict Sense Stationary

TCM Trellis Coded Modulation

TDMA Time Division Multiple Access

USDC United States Digital Cellular

VA Viterbi Algorithm

WCDMA Wideband Code Division Multiple Access

WGN White Gaussian Noise (process)

WLAN Wireless Local Area Network

WSS Wide Sense Stationary

CHAPTER 1

INTRODUCTION

1.1 THE GOAL OF COMMUNICATIONS

The purpose of any communication system is to provide the framework which allows the transmission of information from an *information source* to a *destination* via a *communication channel*. If the communication has to be done without requirements on location, one system of interest is wireless communications. To improve performance, coverage, and efficiency, modern wireless communication systems utilize digital signaling techniques. In general, the channel distorts the transmitted information in a random manner by a variety of possible factors. Among these factors are attenuation, nonlinearities, bandwidth limitations, multipath propagation, and noise. Depending on the type and the degree of such distortions, the transmitted information may be received incorrectly. In a digital system, this translates to an abundance of bit errors. If these bit

errors go unchecked, then the system would become practically useless. The goal of the designer of a digital communication system is thus to provide a cost-effective facility which improves the reliability and quality of such wireless transmissions.

At an acceptable data rate, one method of improving the error performance between two users is to increase the power of the transmitted signal. In such cases, the receiver can more easily determine the content of the communicated message. However, there can be substantial costs which motivate the need to consider other methods of improving communications. One cost is the resulting interference to other users within a system, for the case of multiuser system. While the intended target receives the message more easily, the amplified message may interfere with the communications between other users. Another cost is that, in the case of a mobile user, expending too much power on signal transmission will result in a shorter battery life. In addition, each transmitter has a limitation on its average power.

With digital signaling, an alternative way to mitigate the effects of channel distortion is to use error-control coding. Error-control coding for data integrity may be exercised by means of forward error correction (FEC). Essentially, by introducing structural redundancy in the data stream, the receiver can reduce its susceptibility to channel distortion. Hence, to achieve the same performance criteria, error-control coding can provide several decibels (dB) of signal-to-noise ratio (SNR) gain over uncoded systems. It has been said that each decibel is worth one millions dollars. In some cases, the gain of error control coding over the uncoded case can exceed 8 decibels. A survey of the progress made in designing and applying error-control codes to digital communication systems is given in the next section.

1.2 ERROR CORRECTION CODING

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 [1], Hamming [2],

and Golay [3]. In his paper, Shannon set forth the theoretical basis for coding which has come to be known as *information theory*. By mathematically defining the entropy of an information source and the capacity of a communications channel, he showed that it was possible to achieve reliable communications over a noisy channel provided that the source's entropy is lower than the channel's capacity. This came as a surprise to the communications community which at the time thought it impossible to achieve both arbitrarily small error probability and a nonzero data transmission rate [4]. Shannon did not explicitly state how channel capacity could be practically reached, only that it was attainable.

1.2.1 Block Codes

At the same time Shannon was defining the theoretical limits of reliable communication, Hamming and Golay were busy developing the first practical error control schemes. Their work gave birth to a flourishing branch of applied mathematics known as *coding theory*. Richard Hamming is generally credited with discovering the first error correcting code [4]. In 1946, Hamming, a mathematician by training, was hired by Bell Labs to work on elastic theory. However, Hamming found that he spent much of his time working on computers which at the time were highly unreliable. The computers were equipped with error detection capabilities, but upon detecting an error would simply halt execution of the program. Frustrated that his programs would rarely finish without being prematurely halted, Hamming searched for ways to encode the input so that the computer could correct isolated errors and continue running. His solution was to group the data into sets of four information bits and then calculate three check bits which are a linear combination of the information bits. The resulting seven bit codeword was what was fed into the computer. After reading in the codeword, the computer ran through the algorithm than not only detected errors, but also could determine the location of a single error. Thus the Hamming code was able to correct a single error in a block of seven encoded bits[1].

[1] Hamming presented a generalized class of single error correcting codes in his 1950 paper [4]. In general, a hamming code word is 2^m-1 bits long, where $m \geq 2$, and the number of information bits is 2^m-1-m.

While Hamming's code was a great advancement, it had some undesirable properties. First of all, it was not very efficient, requiring three check bits for every four data bits. Second, it only had the ability to correct a single error within the block. These problems were addressed by Marcel Golay, who generalized Hamming's construction. In the process, Golay discovered two very remarkable codes that now bear his name. The first code, the binary Golay code, groups data into blocks of twelve bits and then calculates eleven check bits. The associated decoding algorithm is capable of correcting up to three errors in the 23 bit codeword. The second code is the ternary Golay code, which operates on ternary, rather than binary, numbers. The ternary Golay code protects blocks of six ternary symbols with five ternary check symbols and has the capacity to correct two errors in the resulting eleven symbol codeword [5].

The general strategy of Hamming and Golay's codes was the same: to group q-ary symbols into blocks of k and then add n-k check symbols to produce an n symbol codeword. The resulting code has the ability to correct t errors, and has *code rate* $r = k/n$. A code of this type is known as a *block code*, and is referred to as a (q,n,k,t) block code for brevity. Furthermore, the Hamming and Golay codes are *linear*, since the modulo-q sum of any two codewords is itself a codeword. In the 50 years since error correcting block codes were first introduced, many new classes of block codes have been discovered and many applications have been found for these codes. For instance, the binary Golay code provided error control during the Jupiter fly-by of Voyager I [5]. However, Golay codes have been replaced in most current communication applications by more powerful codes.

The next main class of linear block codes to be discovered were the Reed-Muller codes, which were first described by Muller in 1954 [7] in the context of Boolean logic design, and later recognized to be a new class of error correcting codes by Reed, who proposed the associated decoding algorithm [8]. Reed-Muller (RM) codes were an important step beyond the Hamming and Golay codes because they allowed more flexibility in the size of the codeword and the number of correctable errors per codeword. Whereas the Hamming and Golay codes were specific codes with particular values for q, n, k, and t,

the RM codes were a *class* of binary codes with a wide range of allowable design parameters. Reed-Muller codes saw extensive application between 1969 and 1977 in the *Mariner* missions to Mars, which used a $(q=2, n=32, k=6, t=7)$ RM code [5]. After the Mariner missions, RM codes fell out favor within the coding community due to the discovery of more powerful codes. Recently there has been a resurging interest in RM codes because the high-speed decoding algorithms are suitable for optical communications [5].

Following the discovery of RM codes came the discovery of *cyclic* codes, which were first discussed in 1957 by Prange of the Air force Cambridge Research Center [9]. Cyclic codes are linear block codes that possess the additional property that any cyclic shift of a codeword is also a codeword. The cyclic property adds a considerable amount of structure to the code, which can be exploited by reduced complexity encoders and (more importantly) reduced complexity decoders. Another benefit of cyclic codes is that they can be compactly specified by a polynomial of degree *n-k*, denoted by *g(D)* and called the *generator polynomial*. Cyclic codes are also called cyclic redundancy check (CRC) codes, and can be decoded by the Meggitt decoder [10]. Meggitt decoders have a complexity that increases exponentially with the number of correctable errors *t*, and are typically only used to correct single and double bit errors. For this reason, CRC codes are primarily used today for error detection applications rather than error correction.

An important subclass of the cyclic codes was discovered almost simultaneously by Hocquenghen in 1959 [11] and by the team of Bose and Ray-Chaudhuri in 1960 [12]. These codes are known as BCH codes, and have length $n=q^m-1$, where m is an integer valued design parameter. The number of errors that the binary $(q=2)$ BCH code can correct is bounded by $t<(2^m-1)/2$. BCH codes were extended to the nonbinary case $(q>2)$ by Reed and Solomon in 1960 [13]. Reed Solomon (RS) codes constituted a major advancement because their nonbinary nature allows for protection against burst of errors. However, it was not until Berlekamp introduced an efficient decoding algorithm in 1967 [14] that RS began to find practical applications. Since that time, RS codes have

found extensive applications in such systems as Compact Disk (CD) players, Digital Versatile Disk (DVD) players, and the cellular Digital Packet data (CDPD) standard [15].

1.2.2 Convolutional Codes

Despite the success of block codes, there are several fundamental drawbacks to their use. First, due to the frame oriented nature of block codes, the entire codeword must be received before decoding can be completed. This can introduce an intolerable latency into the system, particularly for large block lengths. A second drawback is that most codes require precise frame synchronization.[2] A third drawback is that most algebraic-based decoders for block codes work with hard-bit decisions, rather than with the unquantized, or "soft", outputs of the demodulator. With hard-decision decoding typical for block codes, the output of the channel is taken to be binary, while with soft-decision decoding the channel output is continuous-valued. However, in order to achieve the performance bounds predicted by Shannon, a continuous-valued channel output is required.[3] So while block codes can achieve impressive performance over relatively benign channels, they are typically not very efficient, and therefore exhibit rather poor performance when the signal-to-noise ratio is low. Note that the poor performance of block codes at low signal-to-noise ratios is not a function of the code itself, but rather a function of the suboptimality of hard-decision decoding. It is actually possible to perform soft-decision decoding of block codes, although until recently soft-decision decoding has been regarded as too complex. This thinking has begun to change with recent work in the area of error-and-erasures decoding for Reed Solomon codes [5] and trellis-based soft-decision decoding algorithms for other classes of block codes [16].

The drawbacks of block codes can be avoided by taking a different approach to coding, that of *convolutional coding* which was first introduced in 1955 by Elias [17]. Rather than segmenting data into distinct blocks, convolutional encoders add redundancy to a

[2] Frame synchronization implies that the decoder has knowledge of which symbol is the first symbol in a received code word or frame.

[3] Information theory tells us that by using soft-decisions, rather than hard decisions, the power efficiency can be improved by 2.5 dB for rate 1/2 codes.

continuous stream of input data by using a linear shift register. Each set of n output bits is a linear combination of the current set of k input bits and the m bits stored in the shift register. The total number of bits that each output depends on is called the *constraint length*, and denote by K_c. The *rate* of a convolutional encoder is the number of data bits k taken by the encoder in one coding interval, divided by the number of code bits n output during the same interval. Just as the data is continuously encoded, it can be continuously decoded with only nominal latency. Furthermore, the decoding algorithm can make full use of soft-decision information from the demodulator. The first practical decoding algorithm was the sequential decoder of Wozencraft and Reiffen in 1961 [18], which was later modified in 1963 by Fano [19] and by Jelinek in 1969 [20]. While the work of Fano and Jelinek represented an improvement in the decoding algorithm, it was not until the introduction of the Viterbi algorithm in 1967 [21] that an optimal solution (in a maximum likelihood sense) became practical.

After the development of the Viterbi algorithm, convolutional coding began to see extensive application in communication systems. The constraint length $K_c = 7$ "Odenwalder" convolutional code, which operates at rates $r = 1/3$ and $r = 1/2$, has become a standard for commercial satellite applications [22], [15]. Convolutional codes were used by several deep space probes such as *Voyager* and *Pioneer* [5]. All of the second generation digital cellular standards incorporate convolutional coding; GSM uses a $K_c = 5$, $r = 1/2$ convolutional code, USDC use $K_c = 6$, $r = 1/2$ convolutional code, and IS-95 uses a $K_c = 9$ convolutional code with $r = 1/2$ on the downlink and $r = 1/3$ on the uplink [137]. Globalstar also uses a $r = 1/2$, $K_c = 9$ convolutional code, while Iridium uses a $r = 3/4$, $K_c = 7$ convolutional code [23]. Furthermore, all of the third generation standards that are under consideration incorporate convolutional coding [6] for some modes of operation.

1.2.3 Concatenated Codes

A key weakness of convolutional codes is their susceptibility to burst errors. This weakness can be alleviated by using an *interleaver*, which scrambles the order of the

code bits prior to transmission. A *deinterleaver* at the receiver places the received code bits back in the proper order after demodulation and prior to decoding. By scrambling the code bits' order at the transmitter and then reversing the process at the receiver, burst errors can be spread out so that they appear independent to the decoder. The most common type of interleaver is the *block interleaver*, which is simply an M_b x N_b bit array. Data is placed into the array column-wise and then read out row-wise. A bust error of length up to N_b bits can be spread out by a block interleaver such that only one error occurs every M_b bits. All of the second generation digital cellular standards use some form of block interleaving. A second type of interleaver is the *cross* or convolutional interleaver, which allows continuous interleaving and deinterleaving and is ideally suited for use with convolutional codes [24].

It should be noted that in many ways convolutional codes have properties that are complimentary to those of Reed-Solomon codes. While convolutional codes are susceptible to burst errors, RS codes handle burst errors quite well. However, convolutional codes with soft-decision decoding generally outperform RS codes of similar complexity at low signal-to-noise (SNR) [5]. In severely power limited channels, an interesting and efficient system design can be obtained by using both an RS code and convolutional code concatenated in series as shown in Fig. 1.1. Data is encoded by an RS "outer" encoder which then feeds an "inner" convolutional encoder. At the receiver, the

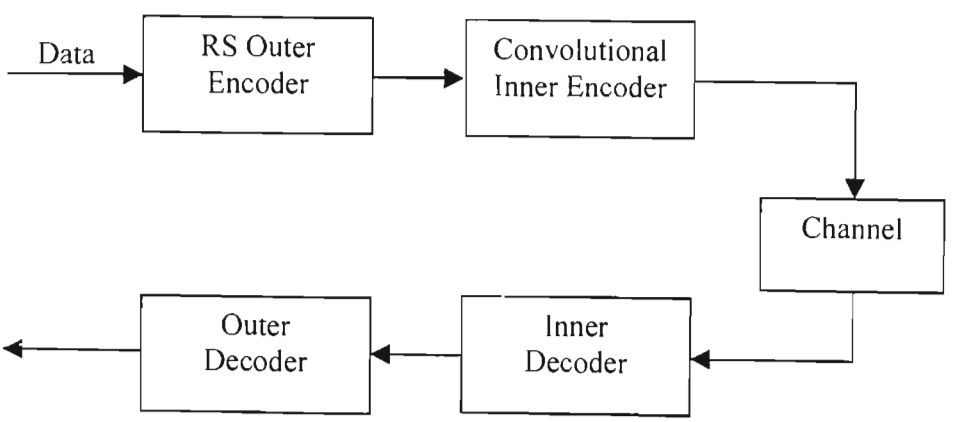


Fig. 1.1 Serial concatenated code

"inner" convolutional decoder cleans up the data received over the noisy channel. The output of the convolutional decoder has a much improved SNR, but due to the nature of convolutional codes, errors are typically grouped into bursts. The "outer" RS decoder completes the decoding process by decoding data output from the convolutional decoder. thus each decoder works with the appropriate type of data; the convolutional decoder works at low SNR with mostly independent errors, while the RS decoder works at high SNR with mostly burst errors. This type of serial code concatenation was first proposed by David Forney in 1966 [25], and is used both by the National Aeronautics and Space Agency (NASA) and the European Space Agency (ESA) for the Deep Space Network (DSN) standard established in 1987. In this standard, a $(q=8, n=255, k=223, t=16)$ RS code is used along with the Odenwalder convolutional code. In extreme cases such as NASA's *Galileo* mission and ESA's *Giotto* mission, a block interleaver can be placed between the convolutional and RS encoders in order to spread very long error burst across several RS code words [5].

1.2.4 Trellis Coded Modulation

Up until the mid 1970's, coding and modulation were regarded as two separate processes. Ungerboeck changed this thinking in 1976 with the introduction of trellis coded modulation (TCM) [26]. TCM uses convolutional coding and multidimensional signal constellations to achieve reliable communications over bandlimited channels. It was TCM that enabled telephone modems to break the 9600 bits per second (bps) barrier, and today all high speed modems use TCM. In addition, TCM has been used for many satellite communication applications [5]. TCM comes remarkably close to achieving Shannon's promise of reliable communications at channel capacity, and is now used for channels with high signal-to-noise ratios that require high bandwidth efficiency.

1.2.5 Turbo Codes

The gap between practical coding systems and Shannon's theoretical limit closed even further in June 1993 at the International Conference on Communications (ICC) in Geneva

Switzerland. At this conference two papers were presented that discussed a new class of codes and the associated decoding technique. Berrou, Glavieux and Thtimajshima coined the term "turbo codes" to describe this new class of codes, and are generally credited with their discovery [27]. However in [28], which was presented at the same conference, an independent group presented a similar technique. Berrou and Glavieux formalized their findings in a 1996 Transactions on Communications paper [29], which was later awarded the 1997 Information Theory Society Paper Award by the Institute of Electrical and Electronics Engineers' (IEEE) Information Theory society [30].

The most surprising element of turbo coding success is its simplicity. The encoder is formed using a parallel concatenation of two or more component encoders. In its original form, the constituent codes were from a subclass of convolutional codes known as recursive systematic convolutional (RSC) codes [27]. If the input block has N information bits, the encoded bit stream is made up of the uncoded information bits and the parity bits of the component encoders. The key element of the encoder is the use of an interleaver which permutates the information sequence and then uses this as the input to the second component encoder. In general, this permutation allows low weight outputs of the first component encoder to result in high weight outputs of the second component encoder. Thus, the combination of the encoders might contain favorable distance properties, even if each component encoder does not.

Due to the presence of the interleaver, optimal (maximal likelihood) decoding of turbo codes is incredibly complex and therefore impractical. Note that a turbo decoder does not perform maximum likelihood decoding directly, but attempts to achieve maximum likelihood decoding in an iterative way. The original turbo decoder [27] used two decoders, each using the maximum a posteriori (MAP) algorithm. The idea behind the decoding strategy is to break the overall decoding problem into smaller problems (decoding each of the constituent codes) with locally optimal solutions and to share information in an iterative fashion. The MAP decoder is described in Chapter 2 and is also describe in Appendix A. There are other less complex algorithms that can be used in place of the MAP algorithm for each decoder such as SOVA and Max-Log-MAP [32].

However, because these other algorithms are suboptimal, they reduce the complexity of decoding at the cost of performance. Hence, for the purpose of this dissertation, we will consider the turbo decoder where each component decoder uses the MAP algorithm to calculate a posteriori likelihood estimates for each bit.

The original turbo code of [27] used constraint length $K_c = 5$ RSC encoders and a 65, 536 bit interleaver. The parity bits were punctured such that the overall code was a $(n = 131{,}072, k = 65{,}532)$ linear block code.[4] Simulation results showed that a bit error rate of 10^{-5} could be attained at an E_b/N_o ratio of just 0.7 decibels (dB) after 18 iterations of decoding. Thus the authors claimed that turbo codes could come within a 0.7 dB of the Shannon limit.[5] These results were so good that they were met with much skepticism by the coding community. Since then, however, similar results have been reproduced by other researchers for a variety of block sizes [33-35] and even improved [36]. However, it was found that the performance of turbo codes degrades as the length of the code n decreases (or equivalently as the size of the interleaver decreases).

Other researchers soon began to look at using other concatenation configurations and other types component codes. It was found that serial concatenated codes offer performance comparable to, and in some cases superior to that of parallel concatenated codes [36]. Also, it was found that the performance with convolutional component codes could be matched or exceeded with block component codes such as Hamming codes [38], BCH codes [39], and Reed-Solomon codes [40]. As a result, it soon became apparent that the real breakthrough from the introduction of turbo codes was not the code construction, but rather the method of iterative decoding.

[4] The size of the input data frame is 4 less than that of the interleaver because 4 bits were reserved to terminate the upper encoder.

[5] The Shannon limit for the additive white Gaussian noise (AWGN) channel with rate 1/2 coding and binary input symbols is actually 0.2 dB, not 0 dB as reported in [27], and thus the original turbo code actually comes within 0.5 dB of the appropriate limit.

1.2.6 Comparison of Coding Systems

A performance comparison of the error correction codes found in several deployed systems is shown in Fig. 1.2 [23]. Here, the x-axis measures power efficiency in terms of the ratio of energy per bit to one sided noise spectral density, E_b/N_o, while the y-axis measures spectral efficiency in terms of the code rate, r. An AWGN is assumed. All points on the curve assume binary phase shift keying (BPSK) modulation and a bit error rate (BER) of $P_b=10^{-5}$. The curve labeled "Shannon Capacity Bound" is the theoretical minimum E_b/N_o required to achieve reliable communication as a function of code rate. Since BPSK modulation is assumed, the actual limit is the curve labeled "BPSK Capacity Bound".

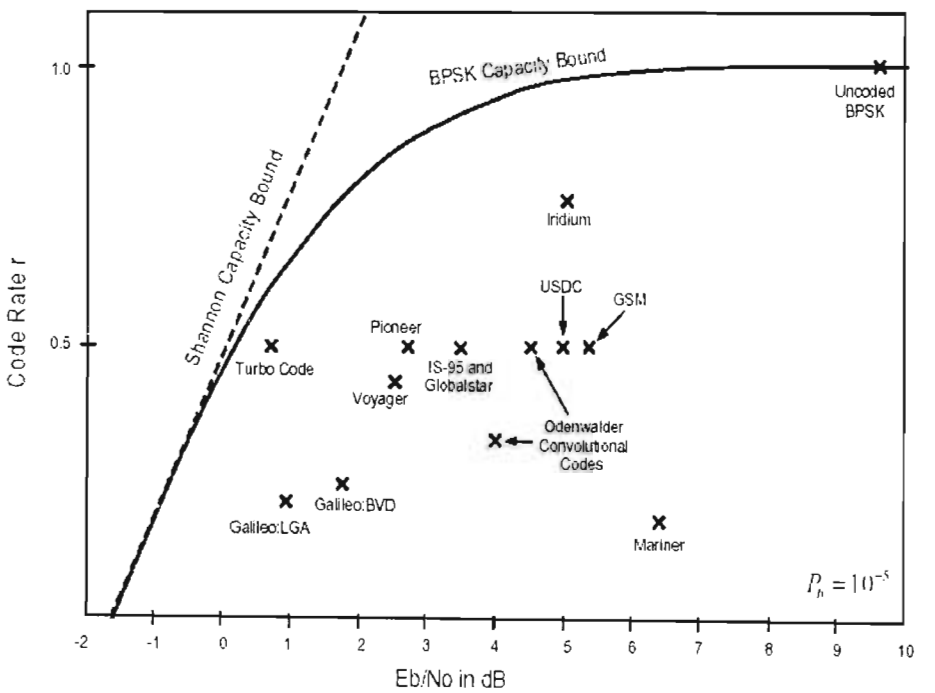


Fig. 1.2 E_b/N_o versus code rate for several error correction coding systems using BPSK modulation in an AWGN channel (taken from [23]).

When BPSK modulation is used without any error correction coding, E_b/N_o = 9.6 dB is required to achieve $P_b=10^{-5}$. If coding is used, then the required value of E_b/N_o can be

reduced, although at the expense of reduced spectral efficiency and increased receiver complexity. The difference between the E_b/N_o required when a code is used and the E_b/N_o required for uncoded BPSK is called the *coding gain*. The first operational code in Fig. 1.2 was the (32,6) Reed Muller code used during the 1969 Mariner mission to Mars. This could provide 3.2 dB of coding gain, but the code rate was only $r = 0.1875$. Although this was a modest coding gain, it was significant because at the time each decibel in coding gain reduced the overall system cost by about $1,000,000 (USD) [23]. The code used by the Pioneer 10, 11, and 12 missions[6] was significantly more powerful than that for Mariner. Pioneer used a sequentially decoded, rate r = 1/2, constraint length K_c = 32 convolutional code which provided 6.9 dB of coding gain.

Many of the systems shown on Fig. 1.2 use convolutional codes. The constraint length K_c = 7 Odenwalder codes of rates 1/2 and 1/3 have been adopted by NASA and ESA as part of the DSN standard, and have become a de facto industry standard for many satellite communication applications. The rate 1/2 Odenwalder code provides 5.6 dB of coding gain. The K_c = 5 GSM code, K_c = 6 USDC code, and K_c = 9 IS-95 code provide coding gain of 4.3, 4.6, and 6.1 dB, respectively [42].[7] The code used for Globalstar is identical to that of IS-95, while Iridium's r =3/4, K_c = 7 convolutional code achieves a 4.6 dB coding gain. The Big Viterbi Decoder (BVD) system was developed for the Galileo mission to Jupiter, and is composed of a constraint length K_c = 15, rate 1/4 convolutional code capable of achieving a 7.9 dB coding gain [43].

The system used by the Voyager missions[8] was a serial concatenation of the r = 1/2 Odenwalder code and a $(q = 2^8, n = 255, k = 223, t = 16)$ Reed Solomon code [23]. The system provided 7.1 dB of coding gain at a rate of r = 0.44. To compensate for long burst errors produced by the convolutional decoder, a block interleaver with depths between 2

[6] Pioneer 10 was launched in 1972 and flew by Jupiter in 1973, Pioneer 11 was launched in 1973 flew by Saturn in 1997, and Pioneer 12 was launched and orbited Venus in 1978 [41].
[7] Note that these standards do not necessarily use BPSK modulation. The performance points shown here are for the actual codes used by the system, but for comparison purposes all data points assume BPSK modulation.
[8] Voyager 1 and 2 were launched in 1979 with the mission of flying by Jupiter, Saturn, Uranus, and Neptune [41].

and 8 outer-code blocks was placed between the RS and convolutional encoders. A concatenation of a convolutional code and the $(q=2^8, n=255, k=223, t=16)$ RS code was also used by the Galileo mission to Jupiter, although for this mission the r = ¼ BVD convolutional code was used. The concatenated system proved to be essential to the success of Galileo because, due to failure of the high gain antenna (HGA), only the low gain antenna (LGA) could be used.

Also shown on Fig. 1.2 is the performance of the original turbo code [27]. As can be seen in the figure, turbo codes approach the capacity limit much more closely than any of the other codes. Further, the complexity of the decoder used by the turbo code is approximately the same as the complexity of the BVD decoder [44]. This excellent performance of turbo codes explains why much of the current research is focused on applying turbo codes to different systems.

In the next section, we describe a specific type of communication system used for both commercial and military applications. In addition, the need for error control coding in this communication system is motivated.

1.3 SPREAD SPECTRUM COMMUNICATIONS AND ERROR CONTROL CODING

One communication system of interest is spread spectrum communication. A spread spectrum can be characterized as a system whose transmitted signal satisfies two criteria. First, the bandwidth of the transmitted signal must be greater than the message bandwidth. This by itself, however, is not sufficient because there are many modulation methods that achieve it. For example, frequency modulation, pulse code modulation, and delta modulation may have bandwidth that are much greater than the message bandwidth. Hence the second criterion is that the transmitted bandwidth must be determined by some function that is independent of the message and is known to the receiver. There are many

advantages to this seemingly wasteful allocation of bandwidth. Important attributes of spread spectrum systems include:

- Antijam capability — particularly for narrow-band jamming.
- Interference rejection.
- Multiple-access capability.
- Multipath protection.
- Covert operation or low probability of intercept (LPI) and detection (LPD).
- Secure communications.
- Improved spectral efficiency — in special circumstances.
- Ranging.

The basic elements of a spread spectrum digital communication system are illustrated in Fig. 1.3. We observe that the channel encoder and decoder and the modulator and

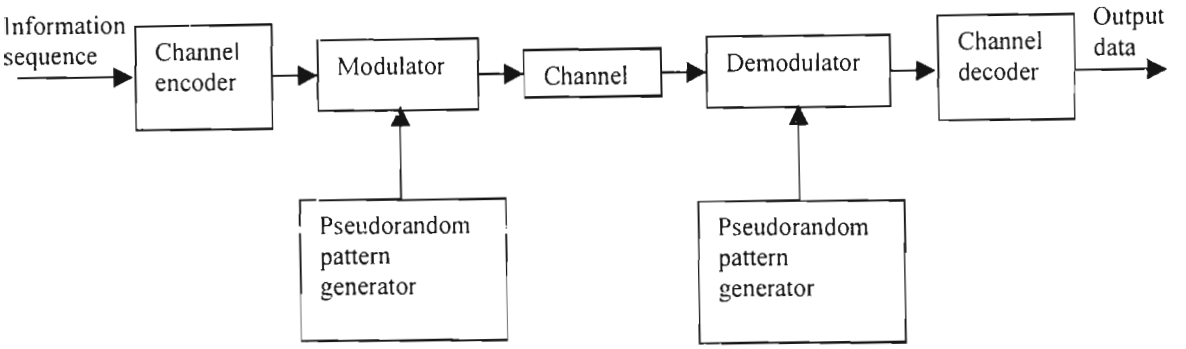


Fig. 1.3. Model of spread spectrum digital communications system

demodulator are the basic elements of a conventional digital communication system. In addition to these elements, a spread spectrum system employs two identical pseudorandom sequence generators, one which interfaces with the modulator at the transmitting end and the second with interfaces with the demodulator at the receiving end. These two generators produce a pseudorandom or pseudonoise (PN) binary-valued sequence that is used to spread the transmitted signal in frequency at the modulator and to despread the received signal at the demodulator. Phase shift keying (PSK) is appropriate for applications where phase coherence between the transmitted signal and the received signal can be maintained over a time interval that span several symbol (or bit) intervals.

When the PN sequence generated at the modulator is used in conjunction with the PSK modulation to shift the phase of the PSK signal pseudorandomly at a rate that is an integer multiple of the bit rate, the resulting modulated signal is called *direct-sequence* (DS) *spread spectrum signal.*

By multiplying the received signal with a synchronized replica of the PN code signal, the desired signal is despread back to a narrow bandwidth while any interference signals are spread over a wide bandwidth. The net effect is a reduction in the interference power by an amount equal to the bandwidth expansion factor, which is called the *processing gain* of the spread spectrum system. The reduction in interference power is the basic reason for using spread spectrum signals to transmit digital information over channels with interference.

The enhancement in performance obtained from a DS spread spectrum signal through the processing gain can be used to enable many DS spread spectrum signals to occupy the same channel bandwidth provided that each signal has its own pseudorandom (signature) sequence. This type of digital communication in which each transmitter/receiver user pair has its own distinct signature code for transmitting over a common channel bandwidth is called *code division multiple access* (CDMA). This CDMA technique is combined with *direct sequence spread spectrum* (DSSS) modulation to yield what is know as *direct sequence code division multiple access* communication (DS-CDMA). Fig.1.4 shows a conceptual block diagram of a wireless DS-CDMA system. The unmodulated users' data, $d_u(t)$, can be either a serial binary bit stream (single user case) or a serial multilevel data stream. This data stream is then modulated by a higher rate code sequence, $p(t)$ which increases the bandwidth of the baseband data signal. This baseband spread spectrum (SS) signal, $d_m(t)$, is finally modulated with a radio frequency (RF) carrier, w_c, before being transmitted into the air (RF channel). During transmission through the channel the signal gets corrupted by the addition of multipath interference, random noise, and other interfering signals which are simply represented as $n(t)$. The received signal,

$r_{rf}(t)$, is then RF demodulated yielding the baseband SS signal, $r_m(t)$. The final baseband demodulation despreads and recovers the original data signal resulting in $r_d(t)$.

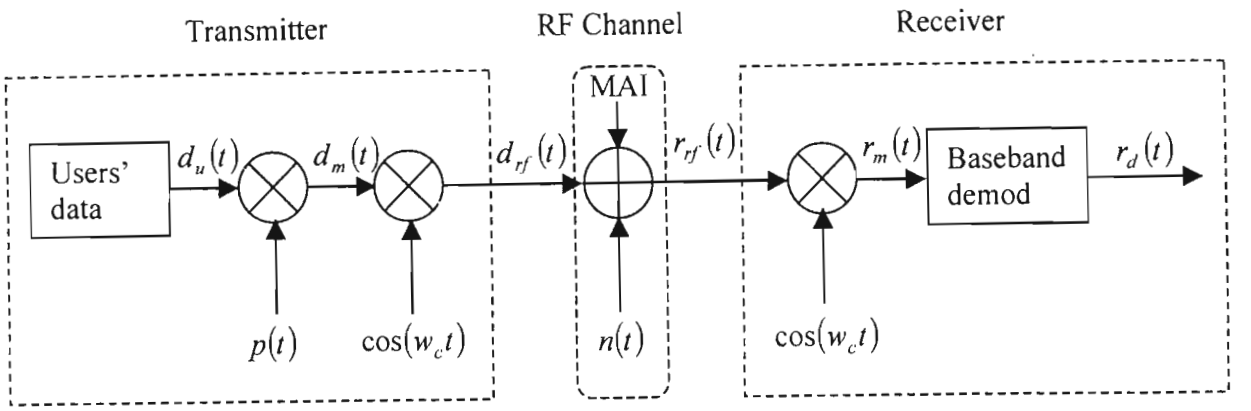


Fig. 1.4 Conceptual Block Diagram of DS-CDMA System

DS-CDMA is the multiple access scheme chosen for the third generation personal communication systems. The most important reason for this is the flexibility in supporting different services with different data rates and quality [80]. Other reasons for using DS-CDMA are less system administration, mitigation of multipath propagation, and higher capacity.

In spite of the many gains associated with spread spectrum, the performance of these systems may not be sufficient. As mention earlier, forward error correction coding techniques are normally employed in digital communication systems to improve transmission performance. In particular, it was emphasized by Viterbi [81] that spread spectrum CDMA systems could exhibit their full potential when they are combined with FEC. The different FEC codes that have been applied in CDMA systems are presented in the next section.

1.4 CODED CDMA SYSTEMS

A number of FEC codes have been considered for CDMA systems. In IS-95 (Electronic Industry Association (EIA) for U.S Digital Cellular (USDC)), a convolutional code with constraint length of 9 is used for both the downlink and the uplink channel. The generator functions for this code are 753 (octal) and 561 (octal). In the downlink channel, for the synchronization channel, the paging channel and rate set 1 on the traffic channel, the convolutional code rate is 1/2. For the rate set 2, an effective code rate of 3/4 is achieved by puncturing two out of every six symbols after the symbol repetition. In the uplink channel, for the access channel and rate set 1 on the traffic channels, the convolutional code rate is 1/3. For rate set 2 on the traffic channel, a code of rate 1/2 is used.

In UMTS/CDMA2000, convolutional codes and turbo codes are used. The convolutional code has a constraint length of 9 and rates 1/2, 1/3 and 1/4. A Turbo code of rate 1/2, 1/3 and 1/4 and constraint length of 4 is preferred for data transmission over 14.4 kb/s on supplemental channel. The turbo code characteristics are presented in Section 2.6.

Although the performance characterization of turbo-coded DS-CDMA systems on a Rayleigh fading channel has been considered by several researchers (e.g. [82, 83]), they have all considered an ideally interleaved Rayleigh fading channel (i.e. the channel is assumed to be memoryless), since this allows for a theoretical analysis. Most practical channels, however, do exhibit memory. In this dissertation we will, among others, consider the performance characterization of turbo-coded DS-CDMA systems on a channel with memory.

1.5 DISSERTATION OUTLINE

The outline for the remainder of this dissertation is as follows. In Chapter 2, turbo codes are reviewed, beginning with a description of the encoding and the decoding operations. The MAP decoding algorithm is described and a heuristic explanation of iterative

decoding is presented. A brief performance analysis of turbo codes is presented and the factors affecting turbo code performance are discussed and investigated. These factors include the number of decoding iteration used, the interleaver size or frame-length of the input data, the structure of the interleaver, the generator polynomials and constraint lengths of the component codes, the component decoding algorithm and the puncturing of component codes. Some extensions of the idea of turbo coding are presented. Finally, the standard turbo codes proposed for various communications systems are presented.

In Chapter 3, we develop a new simple and effective criterion for stopping the iterative process of the turbo decoder for each individual frame immediately after the bits are correctly estimated and thus prevent unnecessary computations and decoding delay. The proposed scheme extends the existing hard-decision-aided (HDA) technique. The new improved hard-decision-aided (IHDA) compares the hard decision of the information bits based on the information produce by the two component decoders at each iteration and terminates the iterative process if they agree with each other for the entire block. Unlike the HDA, it requires no extra data storage. We compare the new scheme with four known stopping criteria. It is shown that the proposed IHDA scheme achieves similar performance in terms of the BER but is more effective in terms of the average number of iterations for small interleaver sizes, while requiring lower complexity. The proposed scheme has the additional advantage that its performance is independent of the choice of any parameter.

In Chapter 4, we examine the use of models in narrowband wireless communication systems. A model is a simple representation of a system used to help describe and predict the system's behavior. Because of a model's simplifying assumptions, no model can represent a system completely. Thus different types of models have different strengths and weaknesses in system analysis. We describe the J_0 fading model (where J_0 arises from Bessel function notation) which is popular in representing narrowband fading. The J_0 statistical model is considered to be the model most closely representing narrowband channel in this dissertation. We look at simulation models, which are suitable for Monte Carlo simulation of the system. In particular, we describe a fast Fourier transform (FFT)

method of generating correlated Rayleigh fading samples. This method describes how we simulate the narrowband wireless channel in this dissertation.

In Chapter 5, we investigate the effect of mobile velocity/Doppler frequency on the performance of a turbo-coded DS-CDMA system over multipath, frequency selective Rayleigh fading channel. This chapter focuses on the modification of the turbo decoder for this model. The approach is to modify the calculation of branch transition probabilities inherent to the original turbo decoder. A new expression for the channel reliability factor is derived. The effects of the channel correlation and the number of resolvable paths on the performance of the system are investigated. Finally, the influence of the choice of the multipath intensity profile on the performance of the system is assessed.

In Chapter 6, we consider the design and performance of a turbo-coded DS-CDMA system in a generic channel with memory, the Gilbert-Elliott (GE) burst channel model. The GE channel is described; it is a two state hidden Markov model where one state represents a bad channel state which typically has high error probabilities and the other state represents a good channel state which has low error probabilities. Some known properties of the channel are recapitulated and useful statistical properties of the channel are derived. The channel parameters are matched to the slowly varying Rayleigh fading channel and the effect of channel interleaving on the GE channel model is assessed. The accuracy of using the GE model is demonstrated by comparing simulation results on both the Rayleigh fading channel and the GE model, showing that both models are equivalent for a broad class of parameters and that moderate variations in the GE model only caused small variations in the bit error rate. Using the transfer function bounding techniques, new analytical expressions for the performance bound of the system are derived and the influence of various parameters such as the Doppler frequency, the signal-to-noise ratio threshold on the system performance are analyzed and investigated. Finally, the GE model is used to compare the performance of different interleaver sizes to see the effect of imperfect interleaving. It is shown that a combination of a short code interleaver with a channel interleaver of appropriate depth could outperform a very large code interleaver

yet resulting in an acceptable delay for even delay-sensitive services such as interactive voice and video and making turbo codes suitable for such services.

Finally, conclusions are drawn and topics for future work are discussed in Chapter 7.

1.6 ORIGINAL CONTRIBUTIONS IN THE DISSERTATION

The original contributions in the dissertation include:

1. In Chapter 3, we develop a simple criterion for stopping the iterative process of the turbo decoder for each individual frame immediately after the bits are correctly estimated and thus prevent unnecessary computations and decoding delay. This work has been published in *IEE Electronic Letters* in October 2001 [139].
2. In Chapter 5, we investigate the effects of the mobile velocity/Doppler frequency on the performance of a turbo-coded DS-CDMA system and derive an expression for the channel reliability factor that takes into account the appropriate channel statistics. This work has been published in *SATCAM* in September 2000 [141].
3. In Chapter 6, we derive closed form expressions for the error probabilities in each state of the Gilbert-Elliott channel model for BPSK modulation. This is followed by a derivation of an expression for the pairwise-error probability for the Gilbert-Elliott channel model. We then use the transfer function bounding techniques to obtain upper bounds on the bit-error rate for maximum likelihood decoding of turbo codes constructed with random permutations on the Gilbert-Elliott channel. We apply this union bound technique to obtain upper bounds on the bit-error rate of a turbo-coded DS-CDMA system. The influence of various parameters such as the Doppler frequency, the signal-to-noise ratio threshold on this bound are analyzed and investigated. Finally, we assess the effect of imperfect interleaving on the GE channel model and compare the performance of different channel interleaver sizes on the Gilbert-

Elliott channel model with imperfect interleaving. We show that a combination of a short code interleaver with a channel interleaver of appropriate depth could outperform a very large interleaver, yet resulting in an acceptable delay for even delay-sensitive services and thus making turbo codes suitable for such services. This work is under review for publication by *IEEE Transactions in Vehicular Technology* [140].

1.7 PUBLISHED WORK

The following publications have resulted from this work.

1.7.1 Journal Papers

- T. M. N. Ngatched and F. Takawira, "A simple stopping criterion for turbo decoding," *IEE Electronic Letters*, vol. 37, no. 22, pp. 1350-1351, Oct. 25, 2001 [139].
- T. M. N. Ngatched and F. Takawira, "Performance analysis of turbo-coded DS-CDMA system on a Gilbert-Elliott channel," *IEEE Trans. Veh. Technol.*, submitted [140].

1.7.2 Conference Papers

- T. M. N. Ngatched and F. Takawira, "Effect of mobile Doppler frequency on the performance of coherent turbo-coded DS-CDMA over frequency selective multi-path Rayleigh fading channel," in *Proc. SATCAM*, 10-13 Sept. 2000, Cape Town, South Africa [141].
- T. M. N. Ngatched and F. Takawira, "Comparison of stopping criteria for turbo decoding," in *Proc. South African Telecommunication Networks & Applications Conference*, 2-5 Sept. 2001, Wild Coast Sun, South Africa [138]

CHAPTER 2

TURBO CODES

2.1 INTRODUCTION

The emergence of turbo codes has had a strong effect on how coding theorists approach the design of codes. Traditionally, the major design objective for an error control code was to maximize the minimum distance, $d_{\min}$, amongst codes with comparable complexity. Yet despite having relatively low $d_{\min}$, turbo codes have been shown to achieve bit error rates (BER) of 10^{-5} at SNRs smaller than codes with higher $d_{\min}$.

The importance of minimum distance can be explained by applying the union bound[9] to the AWGN channel. For AWGN channel with double-sided power spectral density, $N_o/2$, the union bound on the word error probability for an (n, k) block code is [45]

[9] The union bound is an upper bound on the probability of error (for both symbols and bits) and is obtained by taking the sum of the error probabilities of all the error events.

$$P_w \leq \sum_{d=d_{min}}^{n} A_d Q\left(\sqrt{2dr\,E_b/N_o}\right), \qquad (2.1)$$

where $r = k/n$ is the rate of the code, E_b is the received energy per bit, and A_d is the weight enumerator of the code.

Because $Q(x) \leq e^{-x^2/2}$ diminishes quickly for increasing values of x, (2.1) is dominated by the value of d_{min} when $E_s/N_o = r\,E_b/N_o$ is large. If more than one code possesses the same value of d_{min}, the code with smaller $A_{d_{min}}$ is chosen.

In the low SNR regime, however, weights greater than d_{min} may provide a strong contribution to the error rates because $Q(.)$ falls off more slowly. In particular, if the values of A_d grow quickly enough in d, codewords with weight greater than d_{min} may contribute more to the error rate than lower weight codewords. For bit error rates, this effect is compounded if higher weight codewords have higher information weight. Hence, in the low SNR region, d_{min} may no longer dominate the code performance. In other words, for low SNRs, a code with smaller d_{min} may actually yield better performance because its distance profile beyond d_{min} is important to code performance.

The above discussion is not new. It is well known in coding theory that d_{min} dominates performance at high SNRs, but is not as important for low SNRs. The difference is that most codes of interest yield their desired performance in the high SNR domain where d_{min} plays a key role. Those codes that do yield good performance in the low SNR region have prohibitively high decoding complexity [45]. Meanwhile, turbo codes with their relatively low d_{min} have been shown to yield good performance at low SNRs. In fact, turbo codes achieve performance near to Shannon limit and they do so with reasonable decoding complexity. In this chapter, turbo codes are described in greater detail, giving insights regarding their performance at low SNRs.

2.2 TURBO ENCODER

A standard turbo encoder is formed by parallel concatenation of two *recursive systematic convolutional* (RSC) encoders separated by an *N*-bit random interleaver or permuter together with an optional puncturing mechanism [27].

The encoder structure is called a parallel concatenation because the two encoders operate on the same set of input bits, rather than one encoding the output of the other. Thus turbo codes are also referred to as parallel concatenated convolutional codes (PCCC) [46]. A block diagram of a turbo encoder is shown in Fig. 2.1. Clearly, without the puncturer, the encoder is rate 1/3, mapping *N* data bits to *3N* code bits.

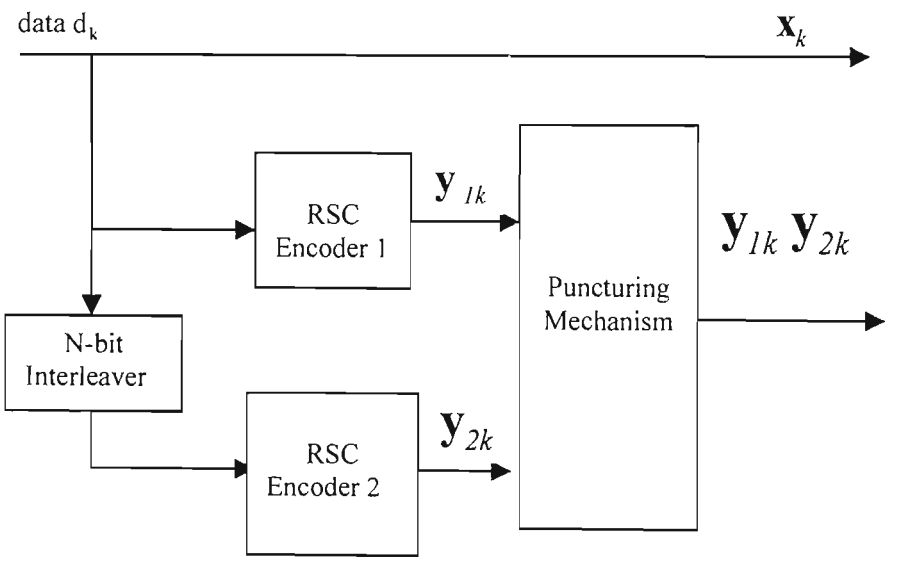


Fig. 2.1: A Turbo Encoder

The generator matrix of a rate 1/2 component RSC code can be represented as

$$\mathbf{G}(D) = \left[1 \quad \frac{\mathbf{g}_1(D)}{\mathbf{g}_0(D)} \right], \tag{2.1}$$

where $\mathbf{g}_0(D)$ and $\mathbf{g}_1(D)$ are feedback and feedforward polynomials with degree ν, respectively. In the encoder, the same information sequence is encoded twice but in a different order. For each input information sequence block $\mathbf{d} = \{d_1, d_2, \ldots\ldots, d_N\}$ of length

N, whose elements take on values of 0 or 1 equiprobably, the first encoder operates directly on it and produces two sequences. The first output sequence, denoted $\mathbf{x}$, is equal to the input sequence since the encoder is systematic. The other output is the first parity check sequence $\mathbf{y}_1 = \{y_{11}, y_{12}, \ldots\ldots\ldots, y_{1N}\}$. The second encoder operates on the interleaved version of $\mathbf{d}$ and produces two sequences as well, but only the second parity sequence $\mathbf{y}_2 = \{y_{21}, y_{22}, \ldots\ldots\ldots, y_{2N}\}$ is transmitted. The information sequence $\mathbf{x}$ and the parity check sequences of the two encoders, $\mathbf{y}_1$ and $\mathbf{y}_2$, are multiplexed to generate the turbo code sequence. So the overall code rate is 1/3. Before describing further details of the turbo encoder in its entirety, we shall first discuss its individual components.

2.2.1. The Recursive Systematic Encoders

Convolutional codes have usually been used in their feed-forward form, like $(\mathbf{g}_0(D), \mathbf{g}_1(D)) = (1 + D^2, 1 + D + D^2)$. However, for these codes a sequence with a single 1, i.e. the sequence [...0001000...], will give a codeword which is exactly the generator vector and the weight of this codeword will in general be very low. It is clear that a single 1 will propagate through any interleaver as a single 1, so the conclusion is that if we use the codes in the feed-forward form in the turbo scheme the resulting code will have a large number of codewords with very low weight.

The trick is to use the codes in their recursive systematic form where we divide the generator matrix with one of the generator vectors. Our example gives $(1, \mathbf{g}_1(D)/\mathbf{g}_0(D)) = (1, (1 + D + D^2)/(1 + D^2))$. This operation does not change the set of encoded sequences, but the mapping of input sequences to output sequences is different. The code is said to be the same, meaning that the distance properties are unchanged, but the encoding is different. Fig. 2.2 depicts an encoder on the recursive systematic form.

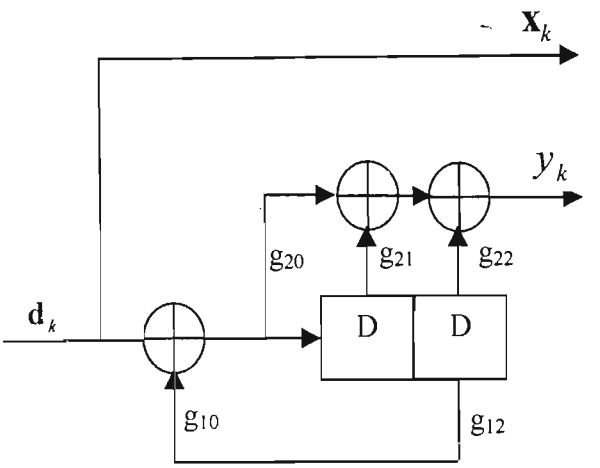


Fig. 2.2: Recursive Systematic Convolutional Encoder

The output sequence we got from the feed-forward encoder with a single 1 is now obtained with the input $1+D^2 = \mathbf{g}_0(D)$. More important is the fact that a single 1 gives a codeword of semi-infinite weight, so with the recursive systematic encoders we may have a chance to find an interleaver where the information patterns giving low weight words from the first encoder are interleaved to patterns giving words with high weight from the second encoder. The most critical input patterns are now patterns of weight 2. For the example code the information sequence [...01010...] will give an output of weight 5.

Notice that the fact the codes are systematic is just a coincidence, although it turns out to be very convenient for several reasons. One of these is that the BER after decoding of a systematic code cannot exceed the BER on the channel. Imagine that the received parity symbols were completely random, then the decoder would of course stick to the received version of the information. If the parity symbols at least make some sense we would gain information on the average and the BER after decoding will be below the BER on the channel.

One thing is important concerning the systematic property, though. If we transmit the systematic part from both encoders, this would just be a repetition, and we know that we can construct better codes than repetition codes. The information part should only be transmitted from one of the constituent codes, so if we use a constituent code with rate

1/2 the final rate of the turbo code becomes 1/3. If more redundancy is needed, we must select constituent codes with lower rates.

2.2.2 The Interleaver

The interleaver in the turbo encoder is a pseudo-random block scrambler defined by a permutation of N elements with no repetitions. The first role of the interleaver is to generate a long block code from small memory convolutional codes, as long codes can approach the Shannon capacity limit. Secondly, it decorrelates the inputs to the two decoders so that an iterative suboptimum decoding algorithm based on information exchange between the two component decoders can be applied. If the input sequences to the two component decoders are decorrelated there is a high probability that after correction of some errors in one decoder some of the remaining errors should become correctable in the second decoder. The final role of the interleaver is to break low weight input sequences, and hence increase the code free Hamming distance or reduce the number of codewords with small distances in the code distance spectrum.

Unlike the classical interleaver (e.g., block or convolutional interleaver), which rearranges the bits in some systematic fashion, it is important that the interleaver sort the bits in a manner that lacks apparent order, although it might be tailored in a certain way for weight-two and weight-three inputs. Also important is that N be selected quite large. The importance of these two requirements will be illuminated below. It should be noted that one pseudo-random interleaver will perform about as well as any other provided N is large. The pseudo-random interleaving pattern must be available at the decoder as well.

2.2.3 Trellis Termination

A turbo encoder operating on each block of information bits can be regarded as a block encoder. Since the component codes are recursive, it is not possible to terminate the trellis by transmitting ν zero tail bits. Trellis termination means driving the encoder to the

all-zero state. This is required at the end of each block to make sure that the initial state for the next block is the all-zero state.

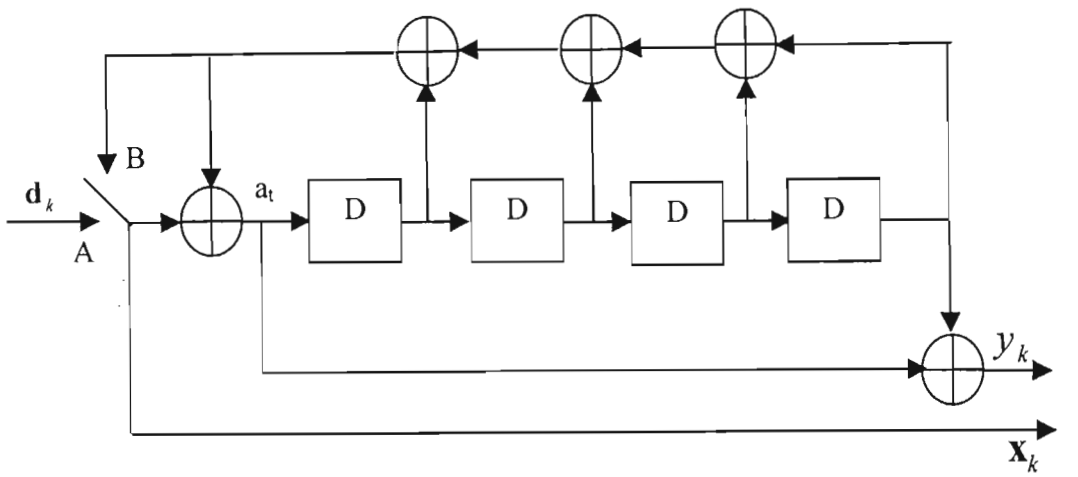


Fig. 2.3: Trellis Termination

The tails bits depend on the state of the component encoder after N information bits. A simple solution to this problem is shown in Fig.2.3 [47]. A switch in each parallel component encoder is in position "A" for the first N clock cycles and in position "B" for ν additional cycles. This will drive the encoder to the all-zero state. Trellis termination is based on setting the input "a_t" to the shift register to zero. This will flush the register with zeros after ν shifts.

With a pseudo-random interleaver it is highly unlikely that this method will terminate both encoders simultaneously with only ν tail bits since the codes are recursive. Usually, only the first component encoder is forced to return to the all zero-state. The performance degradation produced by an unknown final state of the second encoder is negligible for a large interleaver size N [33]. A rate 1/3 turbo code with trellis termination is equivalent to a $(3(N+\nu), N)$ linear systematic block code.

In the remainder of the dissertation we will assume that the first encoder is forced to the all-zero state. Special interleaver structures that can drive both encoders to the all-zero state are discussed in [48-51].

2.2.4 The Puncturer

While for deep space applications low-rate codes are appropriate, in other situations such as satellite communications, a rate of 1/2 or higher is preferred. The role of the turbo code puncturer is identical to that of its convolutional code counterpart, to periodically delete selected bits to reduce coding overhead. For the case of iterative decoding to be discussed below, it is preferable to delete only parity bits as indicated in Fig. 2.1, but there is no guarantee that this will maximize the minimum codeword distance. For example, to achieve a rate 1/2, one might delete all even parity bits from the top encoder and all odd parity bits from the bottom one. The puncturing matrix in this case is given by

$$\mathbf{P} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \tag{2.2}$$

where the puncturing period is 2.

An alternative approach in high rate turbo code design is to employ high rate component codes [52, 53].

In the next section, the structure and the computations associated with turbo decoding algorithm is described.

2.3 TURBO DECODER

The problem of decoding turbo codes involves the joint estimation of two Markov processes, one for each constituent code. While in theory it is possible to model a turbo code as a single Markov process, such a representation is extremely complex and does not lend itself to computationally tractable decoding algorithms. Turbo decoding proceeds instead by first independently estimating the individual Markov processes. Because the two Markov processes are driven by the same set of data, sharing information between the two decoders in an iterative fashion can refine the estimates.

More specifically, the output of one decoder can be used as a priori information by the other decoder. If the outputs of the individual decoders are in the form of hard-bit decisions, then there is little advantage to sharing information. However, if the individual decoders produce soft-bit decisions, considerable performance gains can be achieved by executing multiple iterations of decoding.

2.3.1 Iterative Decoder Structure

A general structure of iterative turbo decoder is shown in Fig. 2.4. Two component decoders are linked by interleavers in a structure similar to that of the encoder. As seen in the figure, each decoder takes three inputs: 1) the systematically encoded channel output bits; 2) the parity bits transmitted from the associated component encoder; and 3) the information from the other component decoder about the likely values of the bits concerned. This information from the other component decoder is referred to as *a-priori* information. The component decoders have to exploit both the inputs from the channel and this *a-priori* information. They must also provide what are known as soft outputs for the decoded bits. This means that as well as providing the decoded output bit sequence, the component decoders must also give the associated probabilities for each bit that it has been correctly decoded. A decoding algorithm that accepts *a-priori* information at its input and produces *a-posteriori* information at its output is called soft-input soft-output (SISO) decoding algorithm. Two suitable decoders are the MAP [54] algorithm of Bahl et al. and the so-called SOVA proposed by Hagenauer and Hoeher [55] which will be described later.

The soft outputs from the component decoders are typically represented in terms of the so-called Log Likelihood Ratios (LLRs), the magnitude of which gives the sign of the bit, and the amplitude the probability of correct decision. The LLRs are simply, as their name implies, the logarithm of the ratio of two probabilities. For example, the LLR $L(u_k)$ for the value of a decoded bit u_k is given by

$$L(u_k) = \log\left(\frac{P(u_k = +1)}{P(u_k = -1)}\right), \tag{2.3}$$

where $P(u_k = +1)$ is the probability that the bit $u_k = +1$, and similarly for $P(u_k = -1)$. Notice that the two possible values of the bits u_k are taken to be +1 and −1, rather then 1 and 0, as this simplifies the derivations that follow.

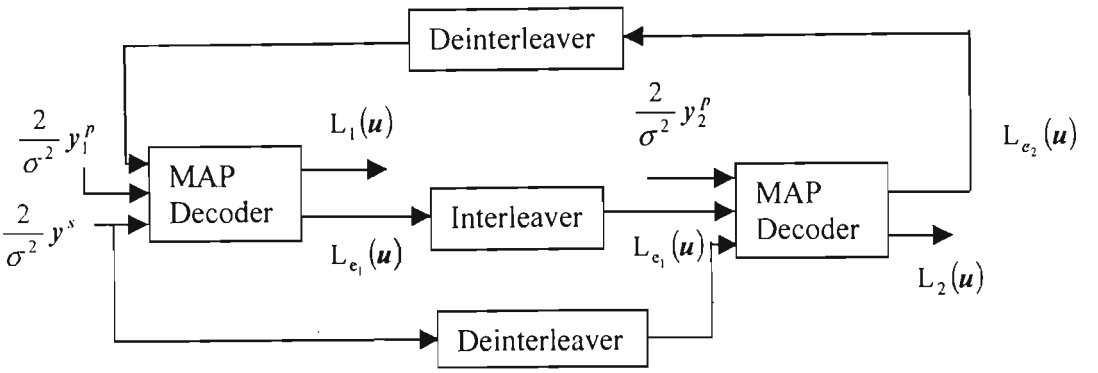


Fig. 2.4: Turbo Decoder

The decoder of Fig. 2.4 operates iteratively, and in the first iteration the first component decoder takes channel output values only, and produces a soft output as its estimate of the data bits. The soft output from the first encoder is then used as additional information for the second decoder, which uses this information along with the channel outputs to calculate its estimate of the data bits. Now the second iteration can begin, and the first decoder decodes the channel outputs again, but now with additional information about the values of the input bits provided by the output of the second decoder in the first iteration. This additional information allows the first decoder to obtain a more accurate set of soft outputs, which are then used by the second decoder as *a-priori* information. This cycle is repeated, and with every iteration the BER of the decoded bits tends to fall. However, the improvement in performance obtained with increasing numbers of iterations decreases as the number of iterations increases. Hence, for complexity reasons, usually only about eight iterations are used.

Due to the interleaving used at the encoder, care must be taken to properly interleave and de-interleave the LLRs which are used to represent the soft values of the bits, as seen in Fig. 2.3. Furthermore, because of the iterative nature of the decoding, care must be taken

not to re-use the same information more than once in each decoding step. For this reason the concept of so-called extrinsic and intrinsic information was used in their seminal paper by Berrou et al. describing iterative decoding of turbo codes [27]. These concepts are described in a later section. Having considered the basic decoder structure, let us now focus our attention on the MAP algorithm in the next section.

2.3.2. The Maximum A-Posteriori Algorithm

The derivation of this algorithm has been well documented in previous papers [27, 33, 54, 56] and is also included in Appendix A. For notational purposes, the algorithm is briefly reviewed here.

The MAP algorithm gives, for each decoded bit u_k, the probability that this bit was +1 or −1, given the received symbol sequence $\mathbf{y}$. This is equivalent to finding the a-posteriori LLR $L(u_k/\mathbf{y})$, where

$$L(u_k/\mathbf{y}) = \log\left(\frac{P(u_k = +1/\mathbf{y})}{P(u_k = -1/\mathbf{y})}\right) \tag{2.4}$$

Let S be the set of all 2^m constituent encoder states (m is the memory) and $s_k \in S$ be the state of the encoder at time k. Furthermore, let S^+ be the set of ordered pairs (s',s) corresponding to all state transitions $(s_{k-1} = s') \rightarrow (s_k = s)$ caused by data input $u_k = +1$, and S^- be similarly defined for $u_k = -1$. Then as shown in [56],

$$L(u_k/\mathbf{y}) = \log\left(\frac{\sum_{S^+} \tilde{\alpha}_{k-1}(s').\gamma_k(s',s).\tilde{\beta}_k(s)}{\sum_{S^-} \tilde{\alpha}_{k-1}(s').\gamma_k(s',s).\tilde{\beta}_k(s)}\right), \tag{2.5}$$

where the $\tilde{\alpha}$'s are computed recursively as

$$\tilde{\alpha}_k(s) = \frac{\sum_{s'} \tilde{\alpha}_{k-1}(s')\gamma_k(s',s)}{\sum_s \sum_{s'} \tilde{\alpha}_{k-1}(s')\gamma_k(s',s)}, \tag{2.6}$$

with initial conditions

$$\tilde{\alpha}_0(0) = 1 \text{ and } \tilde{\alpha}_0(s \neq 0) = 0. \tag{2.7}$$

(These conditions state that the decoder is expected to start in state 0.) and the $\tilde{\beta}'s$ are computed in a "backward" recursion as

$$\tilde{\beta}_{k-1}(s') = \frac{\sum_s \tilde{\beta}_k(s)\gamma_k(s',s)}{\sum_s \sum_{s'} \tilde{\alpha}_{k-1}(s')\gamma_k(s',s)}, \tag{2.8}$$

with boundary conditions

$$\tilde{\beta}_N(0) = 1 \text{ and } \tilde{\beta}_N(s \neq 0) = 0. \tag{2.9}$$

(The encoder is expected to end in state 0 after N input bits, implying that the last m input bits, called *termination bits*, are also selected.) The probability $\gamma_k(s',s)$ in (2.5), (2.6) and (2.8) is defined as

$$\gamma_k(s',s) \equiv p(s_k = s, y_k / s_{k-1} = s'), \tag{2.10}$$

and represents the branch transition probabilities used by the MAP algorithm. They are of particular interest and essentially measure the probability of channel outputs given a particular state transition along a branch in the encoder trellis. The computation of branch transition probabilities depends on the channel, so they play a key role in the design of the turbo decoder for DS-CDMA systems.

Assuming a memoryless Gaussian channel with BPSK modulation, it is shown in [56] that for a systematic code such as a RSC code,

$$\begin{aligned} \gamma_k(s',s) &= P(u_k)p(y_k/u_k) \\ &= \exp\left[\frac{1}{2}u_k\left(L^e(u_k) + L_c y_k^s\right)\right].\gamma_k^e(s',s), \end{aligned} \tag{2.11}$$

where

$$L_c \equiv \frac{4aE_c}{N_0}, \tag{2.12}$$

and where

$$\gamma_k^e(s',s) \equiv \exp\left[\frac{1}{2}L_c y_k^p x_k^p\right]. \tag{2.13}$$

In (2.12) and (2.13), x_k^p and y_k^p are the individual bits within the transmitted and received codewords $\mathbf{y}_k$ and $\mathbf{x}_k$, $E_c = rE_b$ is the energy per channel bit, $\sigma^2 = N_0/2E_c$ is

the noise variance and a is the fading amplitude (we have $a=1$ for nonfading AWGN channels).

2.3.3 Summary of the MAP algorithm

From the description given above, we see that the MAP decoding of a received sequence $\mathbf{y}$ to give the *a-posteriori* LLR $L(u_k/\mathbf{y})$ can be carried out as follows. As the channel values y_k are received, they and the *a-priori* LLRs $L(u_k)$ (which are provided in an iterative turbo decoder by the other component decoder—see next section) are used to calculate $\gamma_k(s',s)$ according to (2.11). As the channel values y_k are received, and the $\gamma_k(s',s)$ values are calculated, the forward recursion from (2.6) can be used to calculate $\tilde{\alpha}_k(s',s)$. Once all the channel values have been received, and $\gamma_k(s',s)$ has been calculated for all $k=1,2.....N$, the backward recursion from (2.8) can be used to calculate the $\tilde{\beta}_k(s',s)$ values. Finally, all the calculated values of $\tilde{\alpha}_k(s',s)$, $\tilde{\beta}_k(s',s)$ and $\gamma_k(s',s)$ are used in (2.5) to calculate the values of $L(u_k/\mathbf{y})$. These operations are summarized in the flowchart of Fig. 2.5.

In the next section we will describe the principles behind the iterative decoding of turbo codes, and how the MAP algorithm described in this section can be used in such a scheme.

2.3.4 Iterative Turbo Decoding Principles

Combining (2.11) with (2.5) it can be shown that, for a systematic code such as a RSC code, the output from the MAP decoder can be re-written as

$$L(u_k/\mathbf{y}) = L(u_k) + L_c y_k^s + L_e(u_k), \tag{2.14}$$

where

$$L_e(u_k) = \log\left(\frac{\sum_{S^+} \tilde{\alpha}_{k-1}(s').\gamma_k^e(s',s).\tilde{\beta}_k(s)}{\sum_{S^-} \tilde{\alpha}_{k-1}(s').\gamma_k^e(s',s).\tilde{\beta}_k(s)}\right). \quad (2.15)$$

Here, $L(u_k)$ is the *a-priori* LLR given by (2.3), and L_c defined by (2.12) is called the channel reliability measure, y_k^s is the received version of the transmitted systematic bit $x_k^s = u_k$.

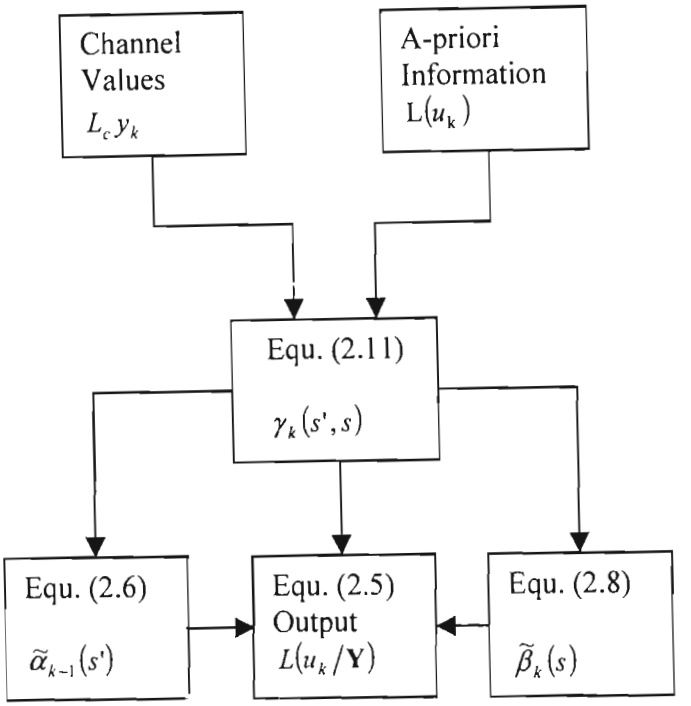


Fig. 2.5: Summary of the key Operations in the MAP Algorithm

Thus, we can see that the *a-posteriori* LLR $L(u_k/\mathbf{y})$ calculated with the MAP algorithm can be thought of as comprising of three terms- $L(u_k)$, $L_c y_k^s$, and $L_e(u_k)$. The *a-priori* LLR term $L(u_k)$ comes from $P(u_k)$ in the expression of the branch transition probability $\gamma_k(s',s)$ in (2.11). This probability should come from an independent source. In most cases we will have no independent or *a-priori* knowledge of the likely value of the bit u_k, and so the *a-priori* LLR $L(u_k)$ will be zero, corresponding to an *a-priori* probability $P(u_k)$=0.5. However, in the case of an iterative turbo decoder, each component decoder

can provide the other decoder with an estimate of the *a-priori* LLR $L(u_k)$, as described later. The second term $L_c y_k^s$ in (2.14) is the soft output of the channel for the systematic bit u_k, which is directly transmitted across the channel and received as y_k^s. The final term in (2.14), $L_e(u_k)$, is derived using the constraints imposed by the code used, from the *a-priori* information sequence $L(u_k)$ and the received channel information sequence $\mathbf{y}$, excluding the received systematic bit y_k^s and the a-priori information $L(u_k)$ for the bit u_k. Hence, it is called the *extrinsic* LLR for the bit u_k. Equation (2.14) shows that the *extrinsic* information from a MAP decoder can be obtained by subtracting the *a-priori* information $L(u_k)$ and the received systematic channel input $L_c y_k^s$ from the soft output $L(u_k/\mathbf{y})$ of the decoder.

We summarize below what we meant by the terms *a-priori*, *a-posteriori*, and *extrinsic information* which are central concepts behind the iterative decoding of turbo codes.

a-priori: The *a-priori* information about a bit is information known before decoding starts, from a source other than the received sequence or the code constraints. It is also sometimes referred to as intrinsic information to contrast with the extrinsic information described next.

extrinsic: The extrinsic information about a bit u_k is the incremental information provided by a decoder based on the received sequence and on *a-priori* information excluding the received systematic bit y_k^s and the *a-priori* information $L(u_k)$ for the bit u_k. Typically, the component decoder provides this information using the constraints imposed on the transmitted sequence by the code used. It processes the received bits and *a-priori* information surrounding the systematic bit u_k, and uses this information about the value of u_k.

a-posteriori: The a-posteriori information about a bit is the information that the decoder gives taking into account *all* available sources of information about u_k. It is the *a-posteriori* LLR, i.e., $L(u_k/\mathbf{y})$, that the MAP algorithm gives as its output.

We now describe how iterative decoding of turbo codes is carried out. Consider initially the first component decoder in the first iteration. This decoder receives the channel sequence $\mathbf{y}^{(1)}$ containing the received versions of the transmitted systematic bits $L_c y_k^s$, and the parity bits $L_c y_k^p$, from the first encoder. Usually, to obtain a half rate code, half of these parity bits will have been punctured at the transmitter, and so the turbo decoder must insert zeros in the soft channel output $L_c y_k^p$ for these punctured bits. The first component decoder can then process the soft channel inputs and produce its estimate $L_{11}(u_k/\mathbf{y})$ of the conditional LLRs of the data bits u_k, $k = 1,2,\ldots,N$. In this notation, the subscript 11 in $L_{11}(u_k/\mathbf{y})$ indicates that this is the *a-posteriori* LLR in the first iteration from the first component decoder. Note that in this first iteration the first component decoder will have no *a-priori* information about the bits, and hence $P(u_k)$ in (2.11) giving $\gamma_k(s',s)$ will be 0.5.

Next, the second component decoder comes into operation. It receives the channel sequence $\mathbf{y}^{(2)}$ containing the *interleaved* version of the received systematic bits, and the parity bits from the second component encoder. Again the turbo-decoder will need to insert zeroes into this sequence if the parity bits generated by the encoder are punctured before transmission. However, now, in addition to the received sequence $\mathbf{y}^{(2)}$, the decoder can use the conditional LLR $L_{11}(u_k/\mathbf{y})$ provided by the first component decoder to generate *a-priori* LLRs $L(u_k)$ to be used by the second component decoder. Metaphorically speaking, these *a-priori* LLRs $L(u_k)$—which are related to bit u_k—would be provided by an "independent conduit of information, in order to have two independent channel-impaired opinions" concerning bit u_k. This would provide a "second channel-impaired" as regards to bit u_k. In an iterative turbo decoder, the extrinsic information $L_e(u_k)$ from the other component decoder is used as the *a-priori* LLRs, after being interleaved to arrange the decoded data bits $\mathbf{u}$ in the same order as the second encoder encoded them. The second component decoder thus uses the received channel sequence $\mathbf{y}^{(2)}$ and the *a-priori* LLRs $L(u_k)$ (derived by interleaving the extrinsic LLRs $L_e(u_k)$ of

the first component decoder) to produce its *a-posteriori* LLRs $L_{12}(u_k/\mathbf{y})$. This is the end of the first iteration.

For the second iteration the first component decoder again processes its received channel sequence $\mathbf{y}^{(1)}$, but now it also has *a-priori* LLRs $L(u_k)$ provided by the extrinsic portion $L_e(u_k)$ of the a-posteriori LLRs $L_{12}(u_k/\mathbf{y})$ calculated by the second component decoder, and hence it can produce an improved a-posteriori LLR $L_{21}(u_k/\mathbf{y})$. The second iteration continues with the second component decoder using the improved *a-posteriori* LLRs $L_{21}(u_k/\mathbf{y})$ from the first component encoder to derive, through (2.14), improved *a-priori* LLRs $L(u_k)$ which it uses in conjunction with its received channel sequence $\mathbf{y}^{(2)}$ to calculate $L_{22}(u_k/\mathbf{y})$.

This iterative process continues with ever-updating extrinsic information to be exchanged between the two component decoders, and with each iteration on the average the BER of the decoded bits will fall. However, in [57, Fig.9], the improvement in performance for each additional iteration carried out falls as the number of iterations increases. Hence, for complexity reasons usually around six to eight iterations are carried out, as no significant improvement in performance is obtained with higher number of iterations.

When the series of iterations finishes the output from the turbo decoder is given by the de-interleaved *a-posteriori* LLRs of the second component decoder, $L_{i2}(u_k/\mathbf{y})$, where i is the number of iteration used. The sign of these *a-posteriori* LLRs gives the hard decision output, i.e., whether the decoder believes that the transmitted data bit u_k was +1 or −1, and in some applications the magnitude of these LLRs, which gives the confidence the decoder has in its decision, may also be useful.

Ideally, for the iterative decoding of turbo codes, the *a-priori* information used by a component decoder in the context of bit u_k should be based on a "conduit of information" independent from the channel outputs used by that decoder. More explicitly, for example, an original systematic information bit and the parity-related information

smeared across a number of bits by the encoder due to the code constraints imposed are affected by the channel differently. Hence, even if the original systematic information bit was badly corrupted by the channel, the surrounding parity information may assist the decoder in obtaining a high-confidence estimate concerning the channel-impaired original systematic bit. However, in turbo decoders the extrinsic LLR $L_e(u_k)$ for the bit u_k, as explained above, uses all the available received parity bits and all the received systematic bits except the received values y_k^s associated with the bit u_k. The same received systematic bits are also used by the other component decoder, which uses the interleaved or de-interleaved version of $L_e(u_k)$ as it's *a-priori* LLRs. Hence, the *a-priori* LLRs $L(u_k)$ are not truly independent from the channel outputs $\mathbf{y}$ used by the component decoders. However, due to the fact that the component convolutional codes have a short memory, usually of only 4 b or less, the extrinsic LLR $L_e(u_k)$ is only significantly affected by the received systematic bits relatively close to the bit u_k. When this extrinsic LLR $L_e(u_k)$ is used as the *a-priori* LLR $L(u_k)$ by the other component decoder, because of the interleaving used, the bit u_k and its neighbors will probably have been well separated. Hence, the dependence of the *a-priori* LLRs $L(u_k)$ on the received systematic channel values $L_c y_k^s$ which are also used by the other component decoder will have relatively little effect, and the iterative decoding provides good results.

Another justification for using the iterative arrangement described above is how well it has been found to work. In the limited experiments that have been carried out with optimal decoding of turbo codes [37, 58, 59], it has been found that optimal decoding performs only a fraction of a decibel (around 0.35-0.5 dB) better than iterative decoding with the MAP algorithm. Furthermore, various turbo coding schemes have been found [37], [60], that approach the Shannonian limit, which gives the best performance theoretically available, to similar fraction of a decibel. Therefore, it seems that, for a variety of codes, the iterative decoding of turbo codes gives an almost optimal performance. Hence, it is this iterative decoding structure, which is almost exclusively used with turbo codes.

2.3.5 Pseudo-Code for the Iterative Decoder

We do not give pseudo-code for the encoder here since this is much more straightforward. However, it must be emphasized that at least the first encoder must be terminated correctly to avoid serious degradation. That is, the last m bits of the N-bit information word to be encoded must force the first encoder to the zero state by the N^{th} bit.

The pseudo-code given below for iterative decoding of turbo code follows directly from the development recapitulated above and is detailed in Appendix A. implicit is the fact that each decoder must have a table (array) containing the input bits and parity bits for all possible state transitions $s' \rightarrow s$. Also required are permutation and de-permutation functions (arrays) since the first decoder (DECODER1) and the second decoder (DECODER2) will be sharing information about each u_k, but DECODER2's information is permuted relative to DECODER1. We denote these arrays by $P[.]$ and $P_{inv}[.]$, respectively. For example, the permuted word **u'** is obtained from the original word **u** via the pseudo-code statement: for $k = 1{:}\ N$, $u'_k = u_{P[k]}$, end. It should be noted that due to the presence of L_c in $L(u_k)$, knowledge of the noise variance by each MAP decoder is necessary.

Initialization

DECODER1:

- $\tilde{\alpha}_0^{(1)}(s) = \begin{cases} 1 \text{ for } s = 0 \\ 0 \text{ for } s \neq 0 \end{cases}$
- $\tilde{\beta}_N^{(1)}(s) = \begin{cases} 1 \text{ for } s = 0 \\ 0 \text{ for } s \neq 0 \end{cases}$
- $L_{21}^c(u_k) = 0 \text{ for } k = 1,2,\ldots,N$

DECODER2:

- $\tilde{\alpha}_0^{(2)}(s) = \begin{cases} 1 \text{ for } s = 0 \\ 0 \text{ for } s \neq 0 \end{cases}$
- $\tilde{\beta}_N^{(2)}(s) = \tilde{\alpha}_N^{(2)}(s)$ for all s (set after the computation of $\{\tilde{\alpha}_N^{(2)}(s)\}$ in the first iteration).

- $L_{12}^{e}(u_k)$ is to be determined from DECODER1 after the first half-iteration and so need not be initialized.

The n^{th} iteration

DECODER 1:

for $k = 1: N$

- get $y_k = (y_k^s, y_k^{1p})$ where y_k^{1p} is a noisy version of the first encoder parity.
- compute $\gamma_k(s', s)$ from (2.11) for allowable state transitions $s' \rightarrow s$ (u_k in (2.11) is set to the value of the encoder input which caused the transition $s' \rightarrow s$; $L^e(u_k)$ is in this case $L_{21}^{e}(u_{Pinv[k]})$, the de-permuted extrinsic information from the previous DECODER2 iteration).
- compute $\tilde{\alpha}_k^{(1)}(s)$ for all s using (2.6).

end.

for $k = N: -1: 2$

- compute $\tilde{\beta}_{k-1}^{(1)}(s)$ for all s using (2.8)

end.

for $k = 1: N$

- compute $L_{12}^{e}(u_k)$ using

$$L_{12}^{e}(u_k) = \log\left(\frac{\sum_{S^+} \tilde{\alpha}_{k-1}^{(1)}(s').\gamma_k^e(s', s).\tilde{\beta}_k^{(1)}(s)}{\sum_{S^-} \tilde{\alpha}_{k-1}^{(1)}(s').\gamma_k^e(s', s).\tilde{\beta}_k^{(1)}(s)}\right)$$

end.

DECODER2:

for $k = 1: N$

- get $y_k = (y_{P[k]}^s, y_k^{2p})$ where y_k^{2p} is a noisy version of the second encoder parity.
- compute $\gamma_k(s', s)$ from (2.11) for allowable state transitions $s' \rightarrow s$ (u_k in (2.11) is set to the value of the encoder input which caused the transition $s' \rightarrow s$; $L^e(u_k)$ is in this case $L_{12}^{e}(u_{P[k]})$, the permuted extrinsic information from the previous DECODER1 iteration; y_k^s is the permuted systematic value, $y_{P[k]}^s$).

- compute $\tilde{\alpha}_k^{(2)}(s)$ for all s using (2.6).

end.

for $k = N : -1 : 2$

- compute $\tilde{\beta}_{k-1}^{(2)}(s)$ for all s using (2.8)

end.

for $k = 1 : N$

- compute $L_{21}^e(u_k)$ using

$$L_{12}^e(u_k) = \log\left(\frac{\sum_{S^+} \tilde{\alpha}_{k-1}^{(1)}(s').\gamma_k^e(s',s).\tilde{\beta}_k^{(1)}(s)}{\sum_{S^-} \tilde{\alpha}_{k-1}^{(1)}(s').\gamma_k^e(s',s).\tilde{\beta}_k^{(1)}(s)}\right)$$

end.

After the last iteration

for $k = 1 : N$

- compute

$$L_1(u_k) = L_c y_k^s + L_{21}^e\left(u_{Pinv[k]}\right) + L_{12}^e(u_k)$$

- if $L_1(u_k) > 0$

  decide $u_k = +1$

  else

  decide $u_k = -1$

end.

2.3.6 Modifications of the MAP Algorithm

The MAP algorithm is, in the form described above, extremely complex due to the multiplications and divisions needed in (2.6) and (2.8) for the recursive calculation of $\tilde{\alpha}_k(s',s)$ and $\tilde{\beta}_k(s',s)$, the multiplications and exponential operations required to calculate $\gamma_k(s',s)$ using (2.11), and the multiplication and logarithm operations required to calculate $L(u_k/\mathbf{y})$ using (2.5). However, much work has been done to reduce this complexity without affecting its performance. Initially the Max-Log-MAP algorithm was

proposed by Koch and Baier [61] and Erfanian et al. [62]. This technique simplified the MAP algorithm by transferring the recursions into the log domain and invoking an approximation to dramatically reduce the complexity. Because of this approximation its performance is sub-optimal compared to that of the MAP algorithm. However, Robertson et al. [32] in 1995 proposed the Log-MAP algorithm, which corrected the approximation used in the Max-Log-MAP algorithm and hence gave a performance identical to that of the MAP algorithm, but at a fraction of its complexity.

2.3.7 The SOVA Algorithm

The Viterbi Algorithm (VA) minimizes the sequence error probability. Its output is in the form of hard-quantized estimation of symbols in the most likely transmitted sequence. In concatenated systems, with multiple signal processing stages, the overall receiver performance is improved if the inner stages produce soft output estimation. The SOVA algorithm [55], [63] is a variation of the Viterbi Algorithm which allows it to be used as a component decoder for turbo codes. This algorithm has two modifications over the classical VA algorithm. Firstly, the path metrics used are modified to take account of *a-priori* information when selecting the ML path through the trellis. Secondly, the algorithm is modified so that it provides a soft output in the form of the *a-posteriori* LLR $L(u_k/\mathbf{y})$ for each decoded bit.

2.4 THE EFFECT OF VARIOUS CODEC PARAMETERS

In this section we present simulation results for turbo codes using Binary Phase Shift Keying (BPSK) over Additive White Gaussian Noise (AWGN) channels. Unless otherwise specified, a MAP decoding algorithm with a total of 8 decoding iterations is used, the size of the interleaver is 1024 and all codes are rate 1/3 with component encoders of memory three and octal generator polynomial (13, 15). We show that there are many parameters, some of which are interlinked, which affect the performance of turbo codes. Some of these parameters are:

- The number of decoding iterations used.
- The interleaver size or frame-length of the input data.
- The structure of the interleaver used.
- The generator polynomials and constraint lengths of the component codes.
- The component decoding algorithm used.
- Puncturing the component codes.

2.4.1 The Effect of the Number of Iterations on Turbo Code Performance

Let us consider the performance of turbo codes with random interleavers when the number of iterations in the iterative decoder is variable. As an example, a rate 1/3, 8 state turbo code with octal generator polynomials (13, 15) and interleaver sizes of 1024 and 10240 is simulated. The simulation results based on the iterative MAP decoding algorithm are shown in Figs. 2.6 and 2.7 for the interleaver sizes of 1024 and 10240, respectively.

Initially, increasing the number of iterations has a considerable effect on the performance. However, there are diminishing returns when the number of iterations becomes high. For interleaver size 1024, there is no significant improvement if the number of iterations increases above 8, while for interleaver size 10240 this threshold is about 12.

As the decoding approaches the performance limit of a given turbo code, any further iteration results in very little improvement. Therefore, it is important to devise an efficient criterion to stop the iteration process and prevent unnecessary computations and decoding delay. We will address this issue in more details in Chapter 3.

2.4.2 The Effect of Interleaver Size on Turbo Code Performance

In the original paper on turbo coding by Berrou et *al.* [27], a pseudo-random block interleaver is used where information is written row by row and read out following a non-uniform rule, based on randomly generated numbers. In this paper, and many of the

subsequent papers, impressive results have been presented for coding with very large frame lengths, The interleaver length is critical for the code performance, particularly at low SNR's. The interleaver structure is important for the performance at high SNR's as it affects the distance properties of the overall turbo code. It determines the code free distance which has a dominant effect on the asymptotic performance.

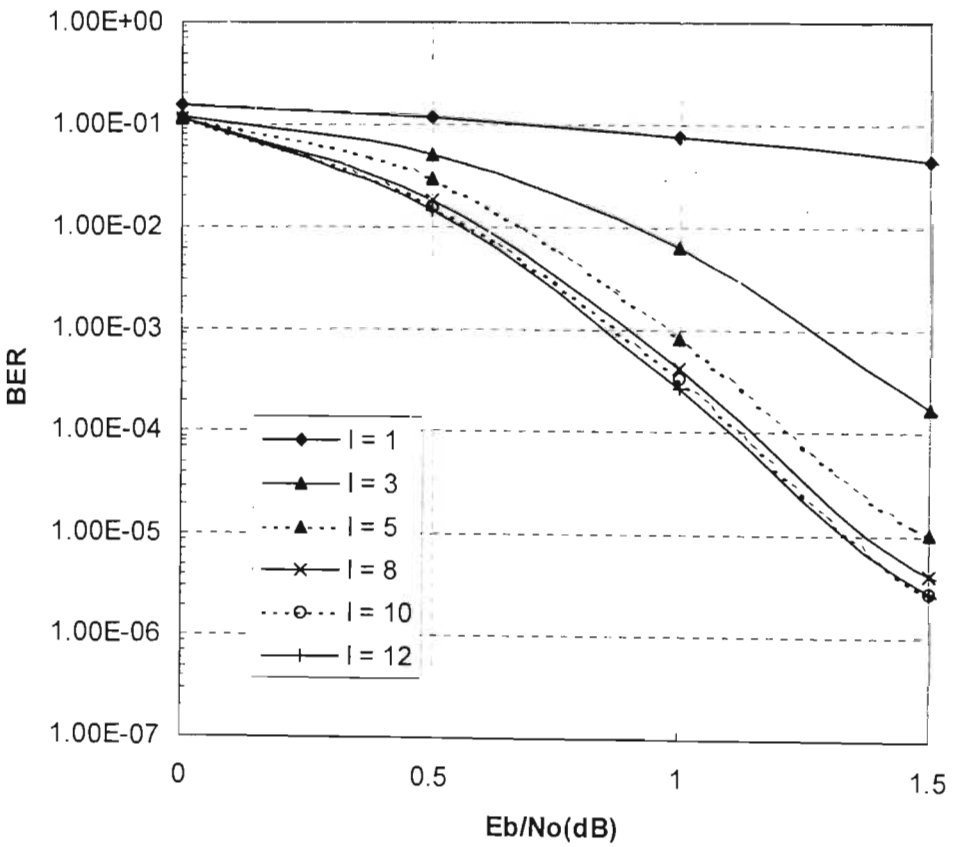


Fig. 2.6: BER performance of a 8 state, rate 1/3 turbo code with MAP algorithm on an AWGN channel, interleaver size 1024 bits, variable number of iterations

As the interleaver enables the information exchange between the two component decoders, increasing the interleaver size has the effect of randomizing the information sequence at the input of the second decoder. Consequently, the inputs to the two component decoders become less correlated, with respect to noise, improving the decoding performance.

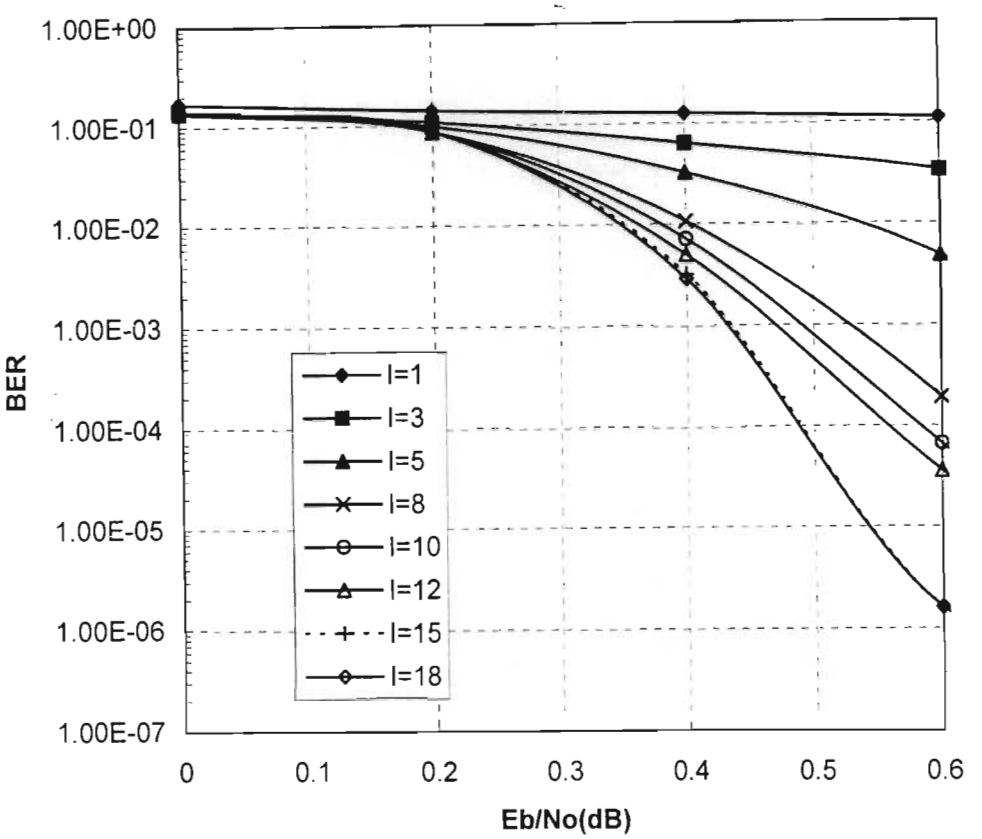


Fig. 2.7: BER performance of a 8 state, rate 1/3 turbo code with MAP algorithm on an AWGN channel, interleaver size 5120 bits, variable number of iterations.

For feedback component codes, interleaving modifies the code weight distribution relative to the uninterleaving input. This is most significant for weight 2 information sequence which has the dominant effect on the performance. If an input weight 2 sequence to the first encoder generates a low weight path in the first code trellis, the interleaved sequence at the input of the second encoder will produce a higher weight path, thus improving the code performance.

The simulation results for a rate 1/3, 8-state turbo code with octal generator polynomials (13, 15) and a pseudo-random interleaver of various lengths are shown in Fig. 2.8. Clearly, the performance is consistently improved by increasing the interleaver size. This is consistent with the expression of the analytical theoretical performance limits as a function of the coded frame length shown in [64]. However, for many applications, such

as speech transmission systems, the large delays inherent in using high frame-lengths are unacceptable. Therefore, an important area of turbo coding research, is achieving as impressive results with short frame-lengths, as have been demonstrated for long frame-systems. The larger frame-length systems would be useful in data or nonreal time transmission systems.

2.4.3 The Effect of Puncturing Component on Turbo Code Performance

One way on increasing the rate of a turbo code is by puncturing the encoder outputs of the basic 1/3 rate turbo code. In this way it is possible to design rate 1/2, 2/3, 3/4, 5/6 etc. codes. The performance result for a rate 1/2 obtained from the rate 1/3, 8-state turbo code with octal generator polynomials (13, 15) and interleaver size of 1024 is shown in Fig. 2.9. It is obtained by simulation with an iterative MAP decoding method. The puncturing matrix for this code is given by (2.2). The performance loss relative to the 1/3 rate code is about 0.6 dB, in terms of E_b/N_0, at the bit error rate of 10^{-4}. This corresponds to a gain about 2.4 dB in terms of channel SNR. Very similar gains are seen for turbo codes with different frame-lengths.

2.4.4 The Effect of The Component Codes

Both the constraint length and the generator polynomials used in the component codes of turbo codes are important parameters. Often in turbo codes the generator polynomials which lead to the largest minimum free distance for ordinary convolutional codes are used, although when the effect of interleaving is considered these generator polynomials do not necessarily lead to the best minimum free distance for turbo codes. Fig. 2.10 shows the huge difference in performance that can result from different generator polynomials being used in the component codes. It can be seen from Fig. 2.10 that the order of these generator polynomial is also important. Swapping (15, 13) to (13, 15) gives a significant degradation in performance. It should be noticed that the performance of a convolutional code (both regular and recursive systematic codes) would be unaffected by the swapping.

The effect of increasing the constraint length of the component codes is shown in Fig. 2.11. It can be seen from this figure that increasing the constraint length of turbo codes does improve its performance, with the $K_c = 4$ code performing about 0.3 dB better than the $K_c = 3$ code at a bit error rate of 10^{-4}, and the $K_c = 5$ code giving a further improvement of about 0.1 dB. However, these improvements are provided at the cost of approximately doubling or quadrupling the decoding complexity.

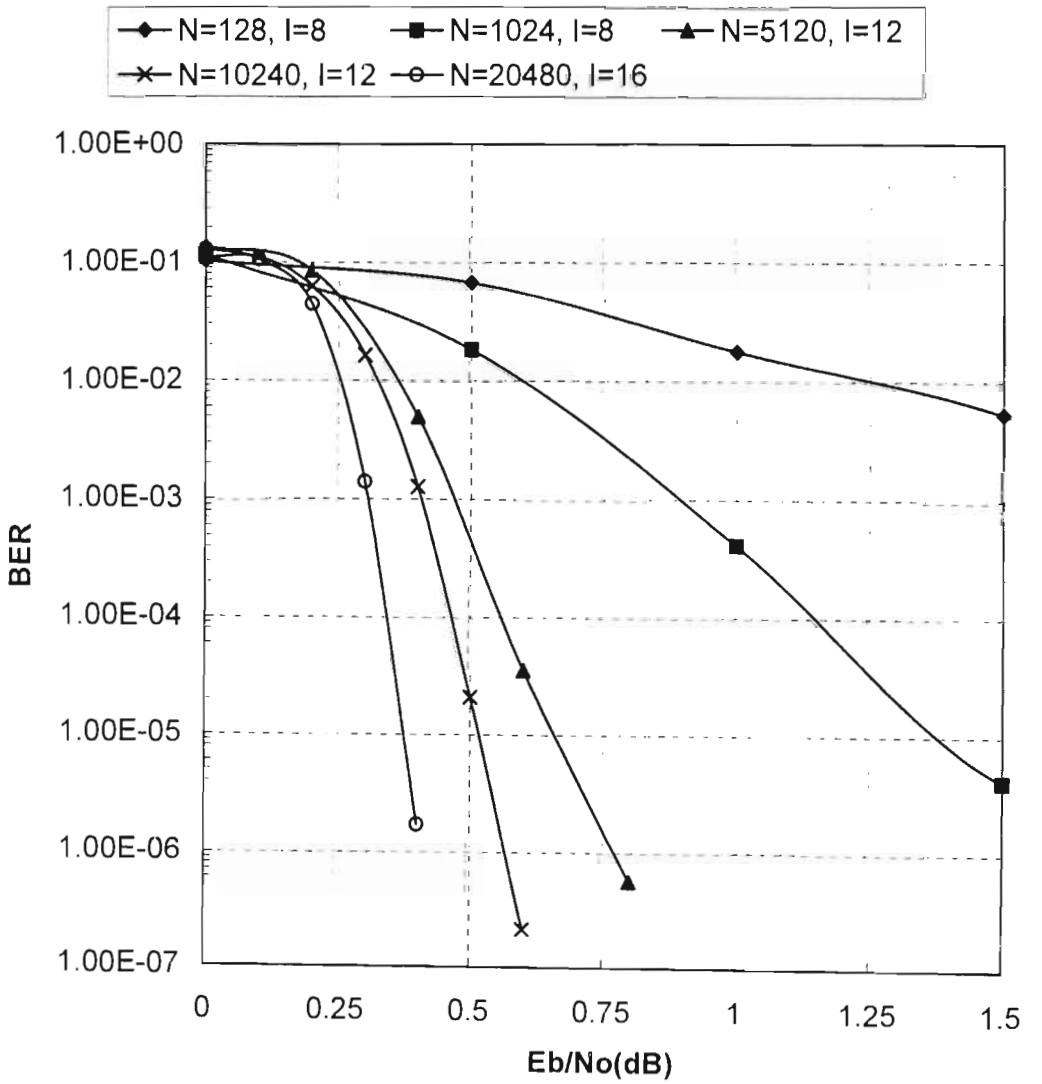


Fig. 2.8: BER performance of a 8 state, rate 1/3 turbo code with MAP algorithm on an AWGN channel, interleaver size N, number of iterations I.

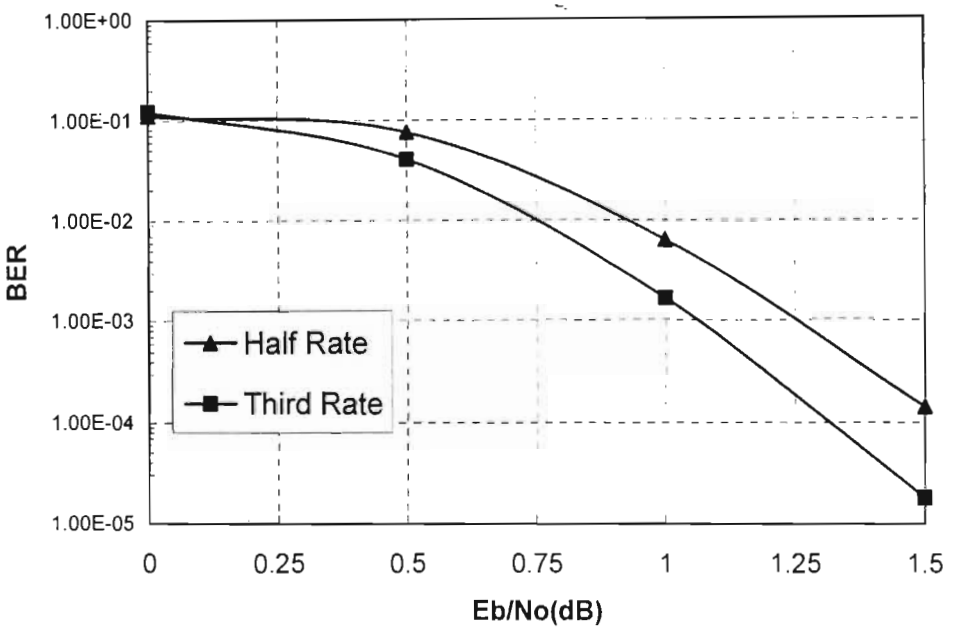


Fig. 2.9: BER performance comparison between 1/3 and 1/2 rate turbo code with MAP algorithm on an AWGN channel, interleaver size 1024.

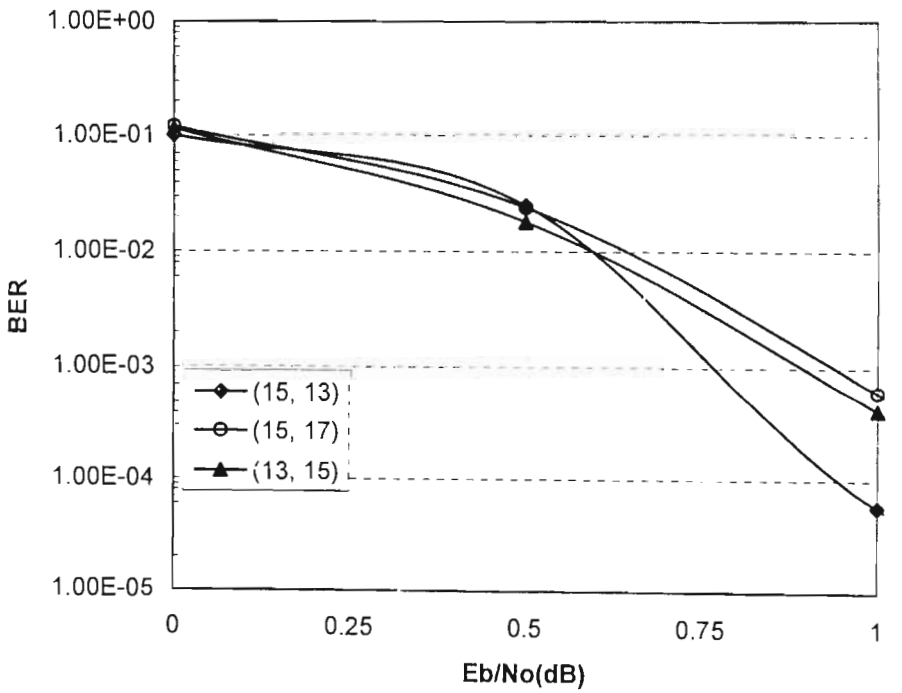


Fig. 2.10: BER performance of a rate 1/3 turbo code with MAP algorithm on an AWGN channel, interleaver size 1024 and octal generators (g_1, g_2).

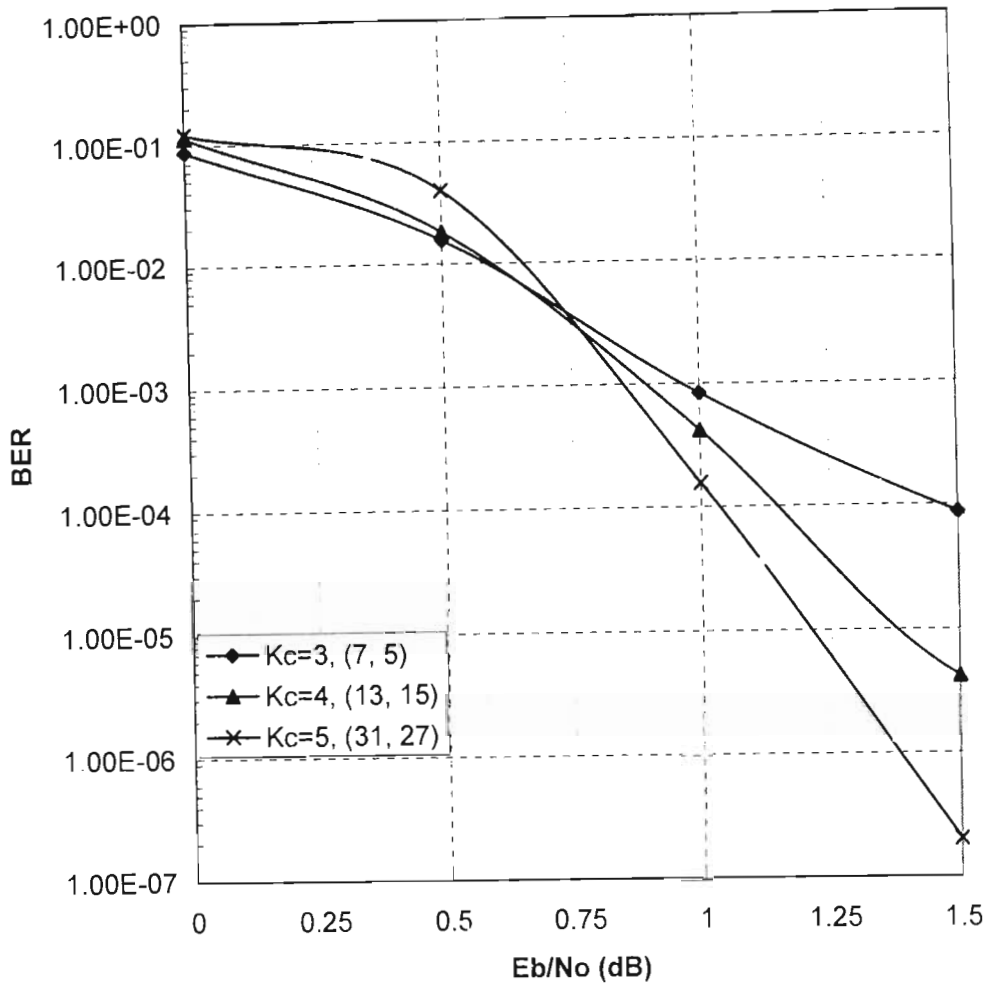


Fig. 2.11: BER performance of a rate 1/3 turbo code with MAP algorithm on an AWGN channel, interleaver size 1024, constraint length K_c and octal generators (g_1, g_2).

2.4.5 The Effect of Interleaver Structure on Code Performance

It is well known that the interleaver used in turbo codes has a vital influence on the performance of the code. As we discussed in a previous section, the turbo code performance at high SNR's is dominated by the code first several distance spectral lines which are produced by low weight input sequences. 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. It plays an important role in determining the code performance at high SNR's.

For example, we consider an input sequence to the first component encoder generating a low weight parity check sequence. It is desirable that the interleaver is capable of breaking this input pattern. That means that the interleaver does not produce the same input pattern to the second encoder, or an input sequence which generates a finite weight code sequence. In such a case, the input sequence to the second component encoder will most likely produce a high weight parity check sequence. This will result in an increase in the turbo codeword weight. If an interleaver is designed to break the low weight input sequences so that the resulting turbo code has a large minimum free distance, the error performance at high SNR's can be improved.

At low SNRs, 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. However, both the interleaver size and structure affect the turbo code minimum free distance and first several distance spectral lines. They play an important role in determining the code performance at high SNRs, and consequently, the asymptotic performance of the turbo code. It is possible to design particular interleavers which can result in good code performance at high SNRs. Several algorithms have been proposed and can be classified into four types of interleaving techniques. They are block interleavers, convolutional interleavers, random interleavers, and code matched interleavers. Details on the design of interleavers and comparison between turbo codes using different interleavers can be found in [33, 48, 35, 65, 67].

2.4.6 The Effect of the Component Decoding algorithm Used

In [68, Figs. 15, 16 and 17], simulations results show that the Max Log MAP and the SOVA algorithms both give degradation in performance compared to the MAP and the Log MAP algorithms. At a BER of 10^{-4} this degradation is about 0.1 dB for the Max Log MAP algorithm, and about 0.6 dB for the SOVA algorithm.

2.5 VARIATION OF TURBO CODES

Conventional turbo codes are composed of two RSC codes, concatenated in parallel, and separated by an interleaver. Parallel concatenation was extended to more than two codes in [69] to obtain *multiple turbo codes*. The additional gain from this was shown to be minimal. Using the same ingredients, namely convolutional encoders and interleavers, *serially* concatenated convolutional codes (SCCCs) have shown to yield performances comparable and, in some cases, superior to turbo codes [70]. In fact any arbitrary topology composed of combinations of parallel and serial concatenation of multiple encoders can be used to create a powerful code [71]. There is no compelling reason why the type of code should be restricted to just RSC codes. Thus the underlying concepts behind turbo codes can be extended to a general class of codes composed of a heterogeneous mix component codes concatenated according to any arbitrary topology. Related work in the area of graphical models has recently been used to describe the general class of concatenated codes [72].

2.6 APPLICATIONS OF TURBO CODES

Turbo and serial concatenated convolutional codes have been proposed for various communication systems, such as deep space, cellular mobile and satellite communication networks. The standard turbo codes are presented in this section.

2.6.1 Turbo Codes for Deep Space Communications

NASA's next generation deep space transponder will contain a turbo code [73]. The Consultative Committee for Space Data Systems (CCSDS) has adopted 16-state rate 1/2, 1/3, 1/4 and 1/6 turbo codes as a new standard for telemetry channel coding [74, 75]. The rate 1/3 turbo code is obtained by parallel concatenation of two identical 19-state rate 1/2 RSC encoders, where the systematic information bit of the second component encoder is eliminated. The generator matrix of the RSC code is given by

$$\mathbf{G}(D)=\left[1 \quad \frac{1+D+D^3+D^4}{1+D^3+D^4}\right],$$

The rate 1/2 turbo code is obtained by puncturing every other parity symbol of each component code in the rate 1/3 turbo encoder. The rate 1/4 turbo code is obtained by parallel concatenation of a 16-state rate 1/3 RSC encoder and a 16-state rate 1/2 RSC encoder whose systematic information bit is eliminated. The generator matrix of the rate 1/3 RSC code is given by

$$\mathbf{G}(D)=\left[1 \quad \frac{1+D^2+D^4}{1+D^3+D^4} \quad \frac{1+D+D^2+D^3+D^4}{1+D^3+D^4}\right].$$

The rate 1/2 RSC code is the same as the component code in the rate 1/3 turbo encoder.

The rate 1/6 turbo code is obtained by parallel concatenation of a 16-state rate 1/4 RSC encoder and a 16-state rate 1/3 RSC encoder whose systematic information bit is eliminated. The generator matrices of the rate 1/4 and 1/3 codes are given by

$$\mathbf{G}(D)=\left[1 \quad \frac{1+D+D^3+D^4}{1+D^3+D^4} \quad \frac{1+D^2+D^4}{1+D^3+D^4} \quad \frac{1+D+D^2+D^3+D^4}{1+D^3+D^4}\right],$$

and

$$\mathbf{G}(D)=\left[1 \quad \frac{1+D+D^3+D^4}{1+D^3+D^4} \quad \frac{1+D+D^2+D^3+D^4}{1+D^3+D^4}\right]$$

respectively.

2.6.2 Turbo Codes for CDMA2000

In the CDMA2000 proposal, turbo codes are recommended for both forward and reverse supplemental channels in the 3rd generation of wideband code division multiple access (CDMA) cellular mobile systems [76]. The reverse link turbo encoder is based on two identical (3, 1, 3) RSC codes with generator matrix

$$\mathbf{G}(D)=\left[1 \quad \frac{1+D^2+D^3}{1+D+D^3} \quad \frac{1+D+D^2+D^3}{1+D+D^3}\right].$$

By puncturing the component encoder outputs, variables code rates of 1/4, 1/2 and 1/3 can be achieved.

The forward link turbo encoder is based on two identical (3, 1, 3) RSC codes with generator matrix

$$\mathbf{G}(D)=\left[1 \quad \frac{1+D+D^3}{1+D^2+D^3} \quad \frac{1+D+D^2+D^3}{1+D^2+D^3}\right].$$

By puncturing the component encoder outputs, variables code rates of 1/4, 1/2 and 1/3 can also be achieved.

2.6.3 Turbo Codes for 3GPP

There are two candidates for the 3rd generation cellular mobile communications based on the 3rd Generation Partnership Project (3GPP) [77]. They include an 8-state rate 1/3 and 1/2 turbo code and a 4-state rate 1/3 serial concatenated convolutional code.

The 8-state turbo code is generated by parallel concatenation of two identical rate 1/2 RSC encoders with generator matrix

$$\mathbf{G}(D)=\left[1 \quad \frac{1+D+D^3}{1+D^2+D^3}\right].$$

The output of the turbo encoders is punctured to produce coded digits corresponding to the desired code rate of 1/3 or 1/2.

The inner code of the 4-state rate 1/3 serial concatenated convolutional code is a rate 1/2 RSC code with generator matrix

$$\mathbf{G}(D)=\left[1 \quad \frac{1+D^2}{1+D+D^2}\right].$$

The outer code is of rate 2/3 obtained by puncturing a rate 1/2 RSC code with the same generator matrix as for the inner code.

The trellis termination is performed in both the turbo and serial concatenated encoders. In the serial encoder, the tailing bits of the outer encoder are included in the interleaver.

2.6.4 Turbo Codes for Satellite Communications

Turbo codes are also recommended for INMARSAT's new mobile multimedia service [78, 79]. A data rate of 64 kbits/s is provided based on the combined turbo coding and 16-QAM modulation. In addition, INTELSAT is currently investigating the application of turbo code for digital services.

2.7 CONCLUSION

This chapter presents an overview of turbo codes, beginning with the description of the encoding and the decoding operations. It is shown that turbo code is the parallel concatenation of two Recursive Systematic Convolutional (RSC) codes that are given the same data in permuted order. We describe the MAP decoding algorithm and present a heuristic explanation of iterative decoding. A brief performance analysis of turbo codes based on simulation results is presented and some of the factors affecting turbo code performance are discussed. Some extensions of the idea of turbo coding are presented. Finally the standard turbo codes proposed for various communications systems are presented.

CHAPTER 3

A SIMPLE STOPPING CRITERION FOR TURBO DECODING

In this chapter, we propose a simple and effective criterion for stopping the iterative process of the turbo decoder for each individual frame immediately after the bits are correctly estimated and preventing unnecessary computations and decoding delay. The proposed scheme extends the existing hard-decision-aided (HDA) technique. The new improved hard-decision-aided (IHDA) compares the hard decision of the information bits based on information produce by the two component decoders at each iteration and terminates the iterative process if they agree with each other for the entire block. Unlike the HDA, it requires no extra data storage. Simulations comparing the new technique with other well known stopping criteria show that the proposed IHDA scheme achieves

similar performance in terms of BER but is in some cases more effective in terms of the average number of iterations, while requiring lower complexity.

We start with a short introduction to the problem. Then we present a brief review of existing stopping criteria. We describe the new stopping criterion in Section 3. Simulation results are presented in Section 4 to compare the new criterion with the existing schemes. Finally, we conclude in Section 5.

3.1 INTRODUCTION

Iterative decoding is a key feature of turbo codes. Each decoding iteration results in additional computations and decoding delay. As the decoding approaches the performance limit of a given turbo-code, any further iteration results in very little improvement. Often, a fixed number M is chosen and each frame is decoded for M iterations. Usually, M is set with the worst corrupted frame in mind. Most frames, however, need fewer iterations to converge. Therefore, it is important to devise an efficient criterion to stop the iteration process for each individual frame immediately after the bits are correctly estimated and prevent unnecessary computations and decoding delay. For each decoded frame, the number of iterations performed is determined by the number of passes before a certain condition or rule for stopping is satisfied. The stopping condition attempts to determine when a frame can be reliably decoded with no further iterations, and it is computed based on data available to the decoder during the decoding of each specific frame. More explicitly, at the end of each iteration, the decoder performs a check on the condition for stopping. If the condition is true, the iterative process on the frame is terminated, and the decoded sequence from the current iteration is sent to the output: otherwise, the iterative process continues to the next iteration. To prevent an endless loop should the stopping rule never be satisfied, we require that the decoder cease after the maximum number of iterations, M.

One such stopping criterion was devised based on the cross entropy (CE) between the distributions of the estimates at the outputs of the decoders at each iteration [84, 85]. This criterion is known as CE criterion. It effectively stops the iteration process with very little performance degradation. Two other simple and effective criteria for stopping the iteration process in turbo decoding with a negligible degradation of the error performance were proposed in [86]. Both criteria, known as the sign change ratio (SCR) and the hard-decision-aided (HDA) respectively, were devised based on the CE criterion and require much less and simpler computations. A simple stopping criterion, known as the sign difference ratio (SDR) that extends the sign change ration (SCR) technique was recently proposed in [87]. Unlike the SCR, the new sign difference ratio (SDR) requires no extra data storage.

In this chapter, we propose a simple stopping criterion for turbo decoding that extends the existing HDA technique. The new improved hard-decision-aided (IHDA) compares the hard decisions of the information bits at the output of each decoder at each iteration and terminates the iterative process if they agree with each other for the entire block. Just like the SDR, it requires no extra data storage. Furthermore, its performance is independent of the choice of any parameter.

3.2 REVIEW OF EXISTING STOPPING CRITERIA

For simplicity, we consider a turbo-code that consists of two rate-1/n recursive systematic convolutional (RSC) codes with feedback. Let $u_k, k \in \{1,.....,N\}$, be the information bits, whose code bits are BPSK modulated and transmitted through an $N(0,\sigma^2)$ AWGN channel. At the receiver, $(y_k^s, y_{1k}^p, y_{2k}^p)$ are signals corresponding to u_k, where y_k^s is the systematic signal, y_{1k}^p and y_{2k}^p are parity for RSC1 and RSC2 respectively. They are sent to the soft-input soft-output (SISO) MAP decoders DEC1 and DEC2 to produce the estimates $\hat{u}_k$, as shown in Fig.3.1.

At the ith iteration, let $L_j^{(i)}(\hat{u}_k)$ and $L_{ej}^{(i)}(\hat{u}_k)$ denote the log-likelihood-ratio (LLR) and the extrinsic values of the estimated information bit $\hat{u}_k$ delivered by decoder j, respectively, with $j = 1, 2$. It is shown in [84] that

$$L_1^{(i)}(\hat{u}_k) = L_{e_2}^{(i-1)}(\hat{u}_k) + \frac{2}{\sigma^2} y_k^s + L_{e_1}^{(i)}(\hat{u}_k), \tag{3.1}$$

$$L_2^{(i)}(\hat{u}_k) = L_{e_1}^{(i)}(\hat{u}_k) + \frac{2}{\sigma^2} y_k^s + L_{e_2}^{(i)}(\hat{u}_k), \tag{3.2}$$

where $\frac{2}{\sigma^2} y_k^s$ is the channel soft value.

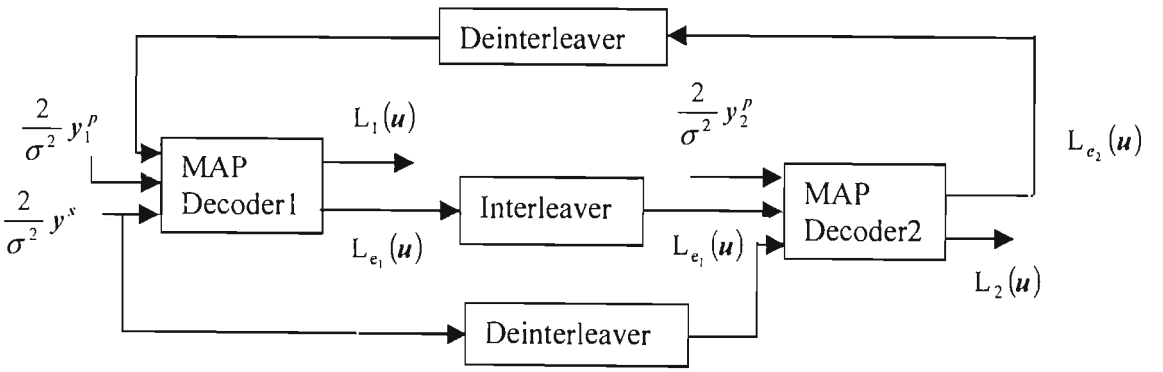


Fig.3.1: Diagram of iterative (turbo) decoder for a two dimensional turbo-code which uses two MAP decoders operating cooperatively.

Four known dynamic stopping criteria are reviewed in the following.

3.2.1 Cross Entropy (CE)

The probability distribution $p_j^{(i)}(\hat{u}_k)$ at the output of the jth decoder is given by [84]

$$p_j^{(i)}(\hat{u}_k = \pm 1) = \frac{\exp\left[\pm L_j^{(i)}(\hat{u}_k)\right]}{1 + \exp\left[\pm L_j^{(i)}(\hat{u}_k)\right]}. \tag{3.3}$$

At iteration i, the CE between the distributions $p_1^{(i)}(\hat{\mathbf{u}})$ and $p_2^{(i)}(\hat{\mathbf{u}})$ of the outputs of decoders one and two for an independently, identically distributed source $\mathbf{u}$ is defined as [84]

$$T(i) \equiv E_{p_2^{(i)}}\left\{\log\frac{p_2^{(i)}(\hat{\mathbf{u}})}{p_1^{(i)}(\hat{\mathbf{u}})}\right\} = \sum_{k=1}^{N} E_{p_2^{(i)}}\left\{\log\frac{p_2^{(i)}(\hat{u}_k)}{p_1^{(i)}(\hat{u}_k)}\right\}, \tag{3.4}$$

where $E[X]$ denotes the expectation of a random variable X. This CE can be used to stop the iteration process in turbo decoding.

Let

$$\Delta L_{e_2}^{(i)}(\hat{u}_k) = L_2^{(i)}(\hat{u}_k) - L_1^{(i)}(\hat{u}_k) = L_{e_2}^{(i)}(\hat{u}_k) - L_{e_2}^{(i-1)}(\hat{u}_k). \tag{3.5}$$

Suppose that the decoding iteration converges, and at iteration i the decoding process can be terminated. Then the following assumptions on the LLR's and the extrinsic values at the outputs of the two decoders can be made [84].

1) Hard decisions of the information bits based on their LLR values do not change anymore, i.e., $\text{sign}\left(L_1^{(i)}(\hat{u}_k)\right) = \text{sign}\left(L_2^{(i)}(\hat{u}_k)\right) = \hat{u}_k^{(i)} = \pm 1$.
2) The magnitudes of $L_1^{(i)}(\hat{u}_k)$ and $L_2^{(i)}(\hat{u}_k)$ are very large so that by (3.3) either $p_j^{(i)}(\hat{u}_k = 1) \approx 1.0$ or $p_j^{(i)}(\hat{u}_k = -1) \approx 1.0$, for $j = 1, 2$.
3) $\Delta L_{e_2}^{(i)}(\hat{u}_k)$ has the same sign as $\hat{u}_k^{(i)}$.
4) The difference between the magnitudes of $L_{e_2}^{(i-1)}(\hat{u}_k)$ and $L_{e_2}^{(i)}(\hat{u}_k)$ are very small and less than 1.0. Hence, when there is no sign change between them, the values of $\Delta L_{e_2}^{(i)}(\hat{u}_k)$ are negligible compared with those with sign changes, i.e., when $k \in \Lambda_x \equiv \left\{m : \text{sign}\left(L_{e_2}^{(i-1)}(\hat{u}_m)\right) = \text{sign}\left(L_{e_2}^{(i)}(\hat{u}_m)\right), m \in \Lambda\right\}$, $\Delta L_{e_2}^{(i)}(\hat{u}_k) \approx 0$.

Based on these assumptions, the CE in (3.4) can be approximated as follows [84]:

$$T(i) \approx \sum_k \frac{\left|\Delta L_{e_2}^{(i)}(\hat{u}_k)\right|^2}{\exp\left(\left|L_1^{(i)}(\hat{u}_k)\right|\right)}. \tag{3.6}$$

In [84] it is shown that when $T(i)$ drops to a value of $(10^{-2}\text{to}10^{-4})T(1)$, the distributions of $p_1^{(i)}(\hat{\mathbf{u}})$ and $p_2^{(i)}(\hat{\mathbf{u}})$ are "close enough" to terminate the iterative process with very little performance degradation.

3.2.2 Sign Change Ratio (SCR)

Let $C(i)$ denote the number of sign changes in $L_{e_2}(\hat{\mathbf{u}}) \equiv (L_{e_2}(\hat{u}_1), L_{e_2}(\hat{u}_1), ..., L_{e_2}(\hat{u}_N))$ from iteration $(i-1)$ to iteration i. The following criterion is a direct result of the CE criterion.

The approximation of $T(i)$ given by (3.6) can be written as the sum of two parts [86]

$$\begin{aligned} T(i) &\approx \sum_{k\in\Lambda_s} \frac{\left|\Delta L_{e_2}^{(i)}(\hat{u}_k)\right|^2}{\exp\left(\left|L_1^{(i)}(\hat{u}_k)\right|\right)} + \sum_{k\in\Lambda/\Lambda_s} \frac{\left|\Delta L_{e_2}^{(i)}(\hat{u}_k)\right|^2}{\exp\left(\left|L_1^{(i)}(\hat{u}_k)\right|\right)} \\ &= T_1(i) + T_2(i). \end{aligned} \tag{3.7}$$

Based on assumption 4), the values $\left|\Delta L_{e_2}^{(i)}(\hat{u}_k)\right|^2$ which contribute to $T_1(i)$ are much more smaller than those that contribute to $T_2(i)$. Furthermore, $\left|L_1^{(i)}(\hat{u}_k)\right|$ in $T_1(i)$ have much larger average values than those in $T_2(i)$. Therefore, $T_1(i)$ is assumed negligible compared with $T_2(i)$, and we have

$$T(i) \approx \sum_{k\in\Lambda/\Lambda_s} \frac{\left|\Delta L_{e_2}^{(i)}(\hat{u}_k)\right|^2}{\exp\left(\left|L_1^{(i)}(\hat{u}_k)\right|\right)} \approx \delta_s C(i), \tag{3.8}$$

where δ_s is defined as the average value of $\dfrac{\left|\Delta L_{e_2}^{(i)}(\hat{u}_k)\right|^2}{\exp\left(\left|L_1^{(i)}(\hat{u}_k)\right|\right)}$ for $k \in \Lambda/\Lambda_s$.

Equation (3.8) shows that the number of sign changes in $L_{e_2}^{(i)}(\hat{u}_k)$ between two consecutive iterations directly relates to the CE between distributions $p_1^{(i)}(\hat{\mathbf{u}})$ and $p_2^{(i)}(\hat{\mathbf{u}})$. This relationship provides a stopping criterion for iterative decoding based on the sign

changes $C(i)$ in $L_{e_2}(\hat{\mathbf{u}})$. Simulation shows that if $C(i) \le (0.005 \sim 0.03)N$, iterative decoding can be stopped with about the same performance degradation as the CE criterion described above [86]. The ratio $C(i)/N$ is called the ratio of sign changes. The stopping criterion is referred to as the sign-change-ration (SCR) criterion.

3.2.3 Hard-Decision-Aided (HDA)

As the decoding iteration converges to the final stage, the first assumption of the CE can be modified as follows [86].

1') $\text{sign}\left(L_2^{(i-1)}(\hat{u}_k)\right) = \text{sign}\left(L_2^{(i)}(\hat{u}_k)\right) = \hat{u}_k^{(i)} = \pm 1$.

Based on assumptions 1') and 2), the CE between $p_2^{(i-1)}(\hat{\mathbf{u}})$ and $p_2^{(i)}(\hat{\mathbf{u}})$ becomes

$$T'(i) = \sum_{k\in\Lambda} E_{p_2^{(i)}}\left\{\log\frac{p_2^{(i)}(\hat{u}_k)}{p_2^{(i-1)}(\hat{u}_k)}\right\}$$

$$\approx \sum_{k\in\Lambda}\left\{\hat{u}_k \Delta L_2^{(i)}(\hat{u}_k) + \log\frac{1+\exp\left(\left|L_2^{(i-1)}(\hat{u}_k)\right|\right)}{1+\exp\left(\left|L_2^{(i)}(\hat{u}_k)\right|\right)}\right\}, \qquad (3.9)$$

where $\Delta L_2^{(i)} \equiv L_2^{(i)}(\hat{u}_k) - L_2^{(i-1)}(\hat{u}_k)$. Simulation shows that when $\text{sign}\left(L_2^{(i)}(\hat{u}_k)\right) = \text{sign}\left(L_2^{(i-1)}(\hat{u}_k)\right)$ for all $k \in \Lambda$, $T'(i)$ is small enough for terminating the iterative process.

At iteration $(i-1)$, we store the hard decisions of the information bits based on $L_{e_2}^{(i-1)}(\hat{u}_k)$ and check the hard decision based on $L_{e_2}^{(i)}(\hat{u}_k)$ at iteration i. If they agree with each other for the entire block, the iterative process is terminated at iteration i. This stopping criterion is called the hard-decision-aided (HDA) criterion.

3.2.4 Sign Difference Ratio (SDR)

This technique is a modification of the SCR technique that obviates the need for storage of values from the previous iteration. At iteration i it computes D_{ji}, which is the number of sign differences between $L_{ej}^{(i)}(\hat{u}_k)$. Decoding is terminated when $D_{ji} \prec p \times N$, where p is a constant usually chosen to be $0.001 \leq p \leq 0.01$, and N is the frame size [87].

The CE method requires $(6N-1)$ real number operations [including N additions to get $L_1^{(i)}(\hat{u}_k)$] and $(N+2)$ real number memory units for storage. The SCR technique requires only N binary additions, a counter no greater than N, and N bits to store the sign bits of the extrinsic information. At each iteration, the HDA criterion requires N binary operations to obtain the signs of $L_2^{(i)}(\hat{u}_k)$ for all k, at most N logic operations to check the sign changes and N integer memory units for storing the signs of $L_2^{(i)}(\hat{u}_k)$ for all k. Just as with the SCR, the SDR scheme requires N binary additions of sign bits and a counter no greater than N to check the criterion. The efficiency of both the SCR and the SDR is subjected to and appropriate choice of the constants q and p respectively.

Although the SDR method is very simple, we present in the following a modified scheme of the HDA which obviates the need for storage of values from the previous iteration as well and which is more efficient than the other techniques. We call this new technique the improved hard-decision-aided (IHDA).

3-3 A NEW STOPPING CRITERION

Although iterative decoding improves the LLR value for each information bit through iterations, the hard decision of the information bit is ultimately made based on the sign of its LLR value. The hard decisions of the information sequence at the end of each iteration provide information on the convergence of the iterative decoding process.

According to (3.1) and (3.2) the output of each MAP decoder is composed of three terms: the *a-priori* value ($L_{e_2}^{(i-1)}(\hat{u}_k)$ for the first decoder and $L_{e_1}^{(i)}(\hat{u}_k)$ for the second decoder), the *extrinsic* information obtained based on the code constraint ($L_{e_1}^{(i)}(\hat{u}_k)$ for the first decoder and $L_{e_2}^{(i)}(\hat{u}_k)$ for the second decoder) and the soft output from the channel ($\frac{2}{\sigma^2} y_k^s$ for both decoders). Among the three estimates, $\frac{2}{\sigma^2} y_k^s$ is fixed for every iteration, while the *a-priori* and *extrinsic* values are updated from iteration to iteration.

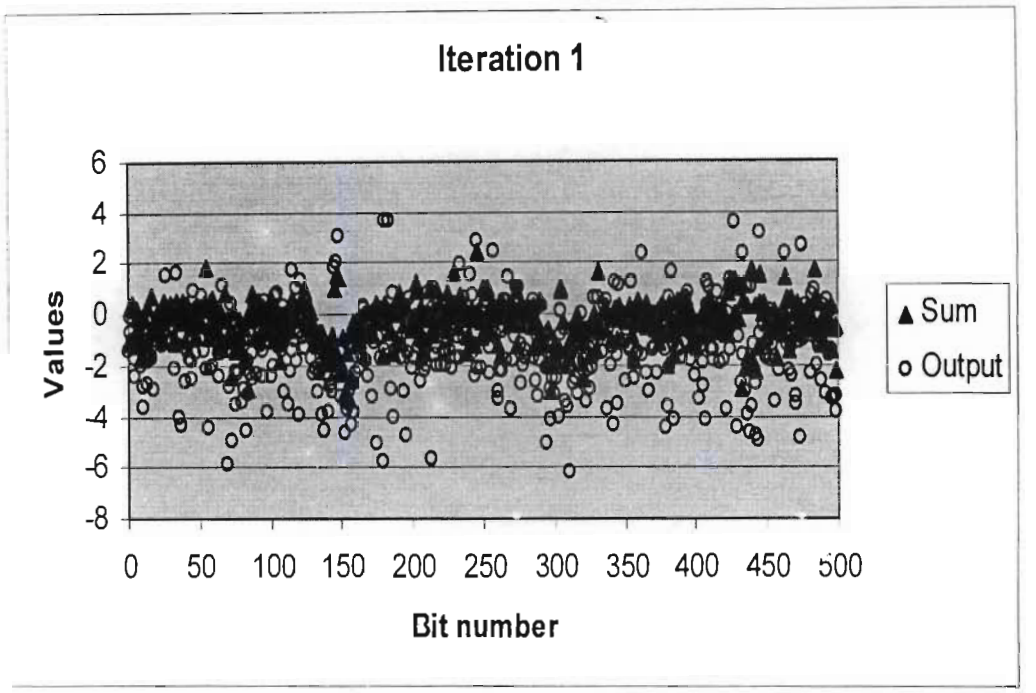
From repeated simulations, it was observed that, the influence of each of these terms on the *a-posteriori* LLR delivered by the MAP algorithm depends on whether the frame is "good" (easy to decode) or "bad" (hard to decode). Fig. 3.2 shows how the LLR value and the *a-priori* and *extrinsic* sum values for each information bit of a "bad" frame from the component decoders in an iterative decoder vary with the number of iterations used. The output from the second component decoder is shown after one, three, five and seven iterations. The output from the first component decoder (not shown here) follows the same pattern. The input sequence consisted entirely of –1 values, hence negative LLR values correspond to a correct hard decision, and positive values to an incorrect hard decision. The encoded bits were transmitted over an AWGN channel. It can be seen that as the number of iterations used increases, the output, i.e. the magnitudes of the *a-posteriori* values, do not increase significantly, but stay close to the magnitudes of the *a-priori* and *extrinsic* sum. Because the soft output from the channel, $\frac{2}{\sigma^2} y_k^s$, is fixed for every iteration, this observation tells us that both the *a-priori* and *extrinsic* do not increase significantly but stay close to or lower than the soft output from the channel. In this case, therefore, the soft output from the channel will contribute on the *a-posteriori* LLR delivered by the MAP algorithm.

Fig. 3.3 shows how the LLR value and the *a-priori* and *extrinsic* sum values for each information bit of a "good" frame from the component decoders in an iterative decoder

vary with the number of iterations used. The output from the second component decoder is shown after one, three, five and seven iterations. The output from the first component decoder (not shown here) follows the same pattern. Again, the input sequence consisted entirely of –1 values, hence negative LLR values correspond to a correct hard decision, and positive values to an incorrect hard decision. The encoded bits were transmitted over an AWGN channel. It can now be seen that as the number of iterations used increases the number of positive LLR values, and hence the BER, decreases until after five iterations there are no incorrectly decoded values. Furthermore, as the number of iterations increases, the decoders become more certain about the value of the bit and hence the magnitudes of the LLRs gradually become larger. A comparison with the magnitudes of the *a-priori* and *extrinsic* sum on the same figure shows that this sum primarily determined the *a-posteriori* value. Consequently, because the soft output from the channel, $\frac{2}{\sigma^2} y_k^s$, is fixed for every iteration, the increase in the magnitudes of the LLRs is due to increases in the magnitude of the extrinsic information. Hence, the soft output of the channel will have less impact on the a-*posteriori* LLR delivered by the MAP algorithm in this case. Because the extrinsic information keeps increasing as the number of iteration i increases, and from (3.1) and (3.2), it is reasonable to expect that, for a "good" frame, both the *a-priori* and *extrinsic* will agree on the hard estimation as the iteration converges.

In consequence, for any arbitrary frame, the new terminating scheme is: At iteration i, we compare the hard decisions of the information bit based on $\left(\frac{2}{\sigma^2} y_k^s + \mathrm{L}_{e_1}^{(i)}(\hat{u}_k)\right)$ with the hard decision of the information bit based on $\mathrm{L}_2^{(i)}(\hat{u}_k)$. If they agree with each other for the entire block, we simply terminate the iterative process at iteration i.

Just as with the SCR, the IHDA eliminates the storage from the previous iteration. At each iteration, the IHDA criterion requires N binary additions of sign bits and a counter not greater than N to check the sign changes.



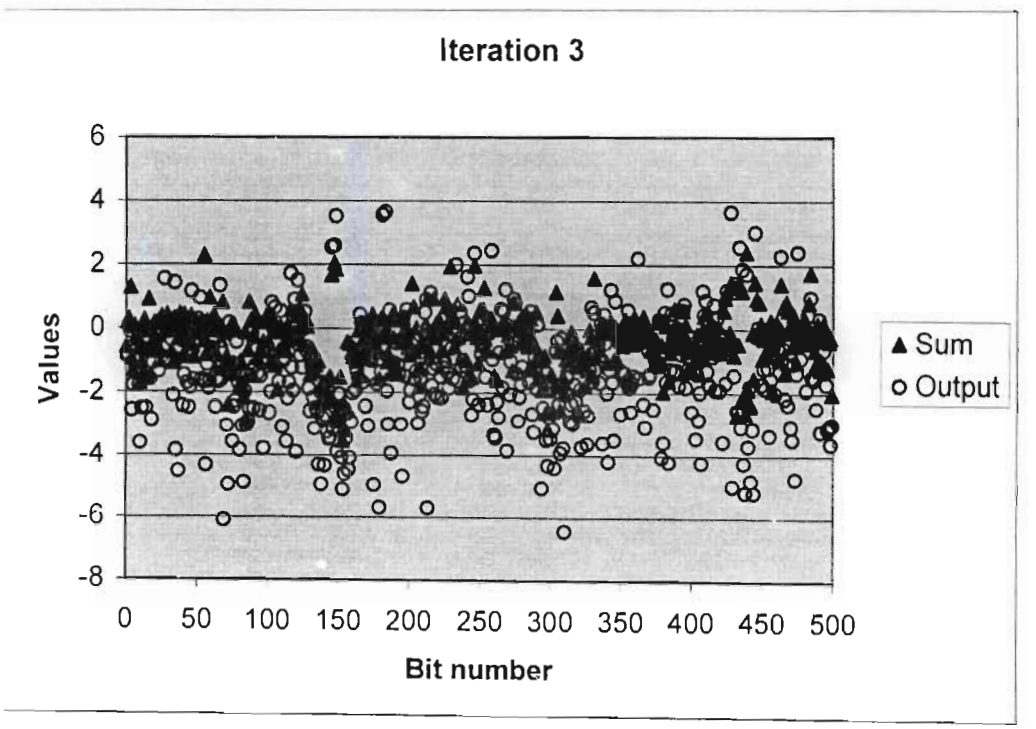


Fig. 3.2a: Soft outputs from the MAP decoder in an iterative turbo decoder for a transmitted "bad" stream of all –1.

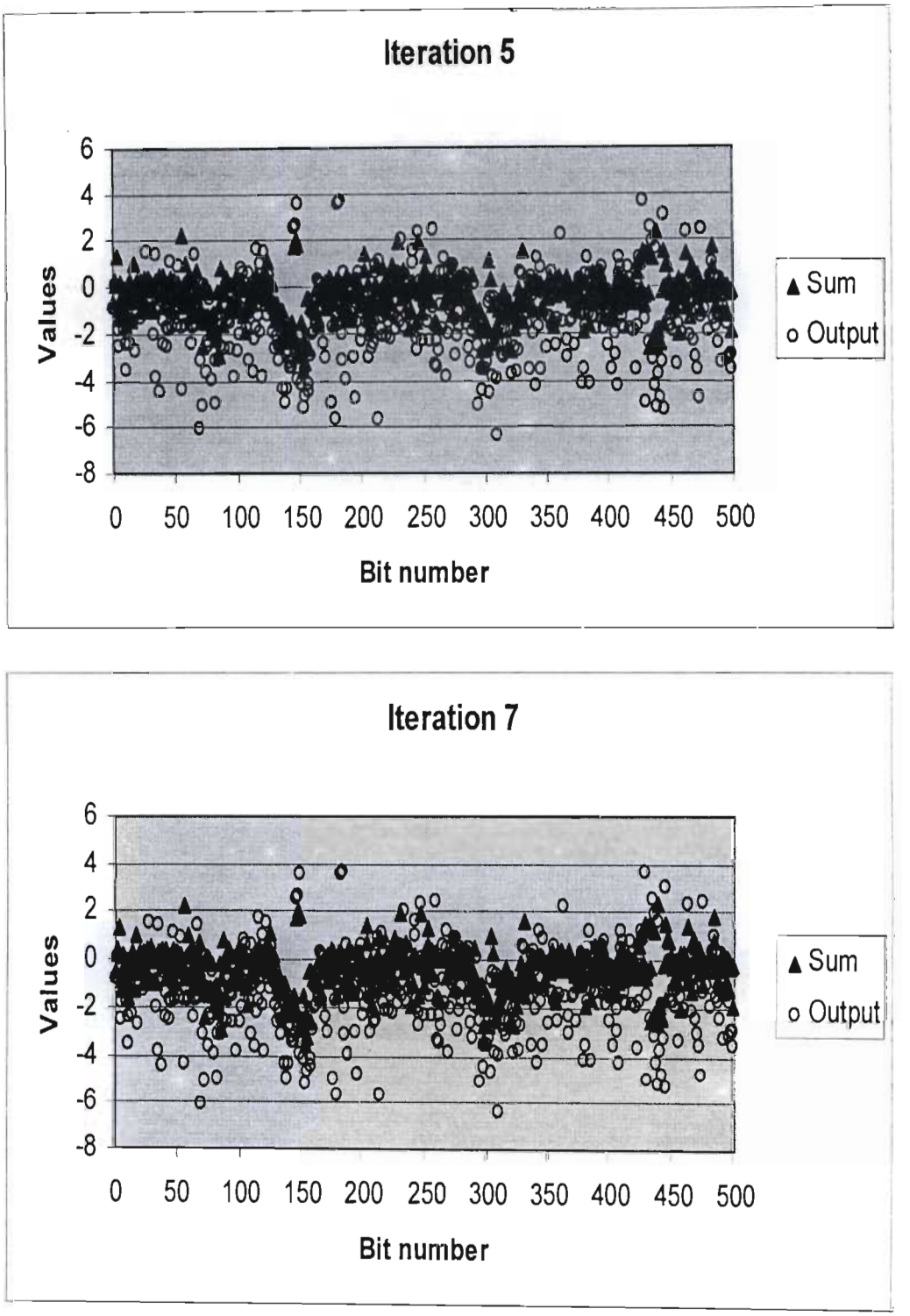


Fig. 3.2b: Soft outputs from the MAP decoder in an iterative turbo decoder for a transmitted "bad" stream of all −1.

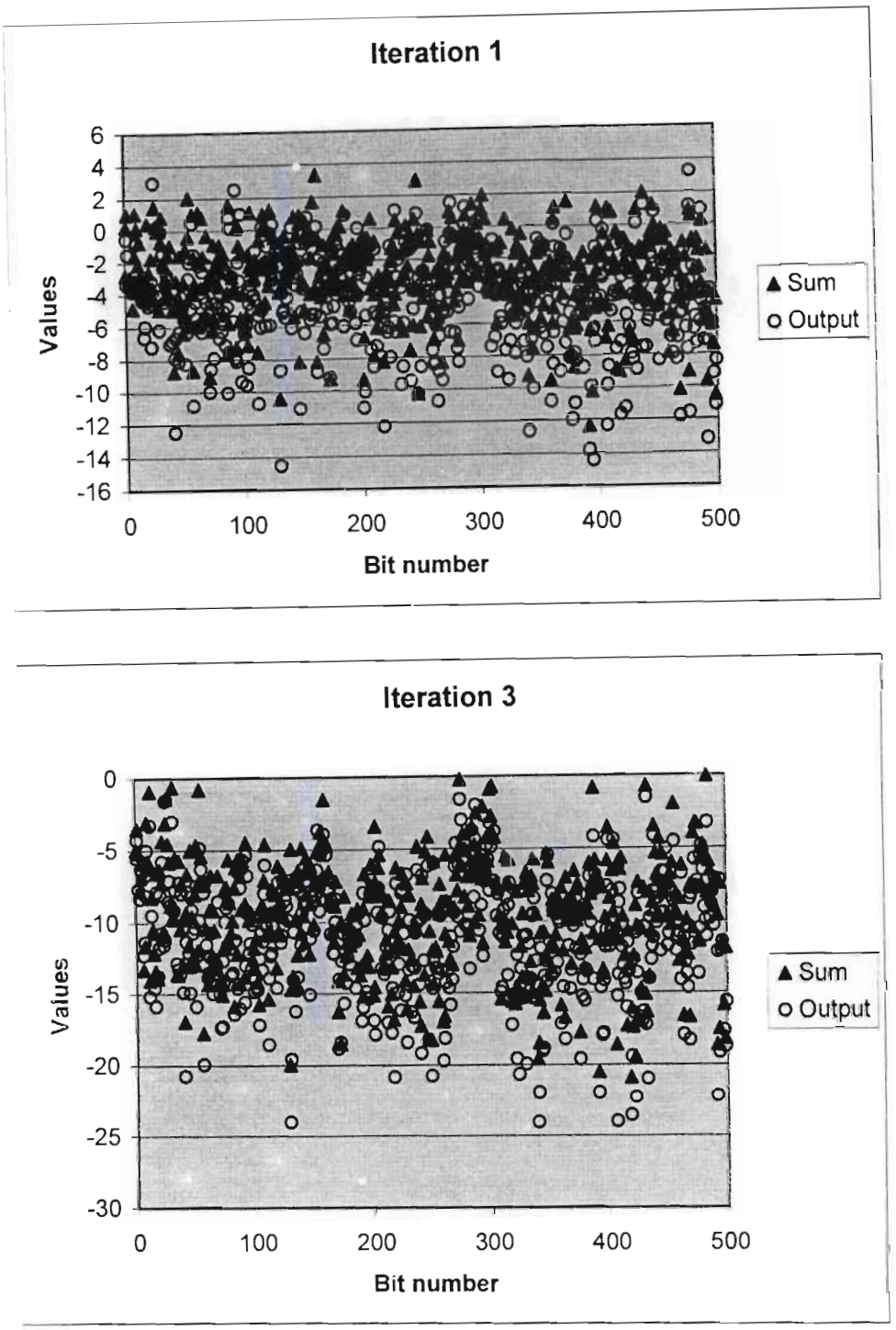


Fig. 3.3a: Soft outputs from the MAP decoder in an iterative turbo decoder for a transmitted "good" stream of all –1.

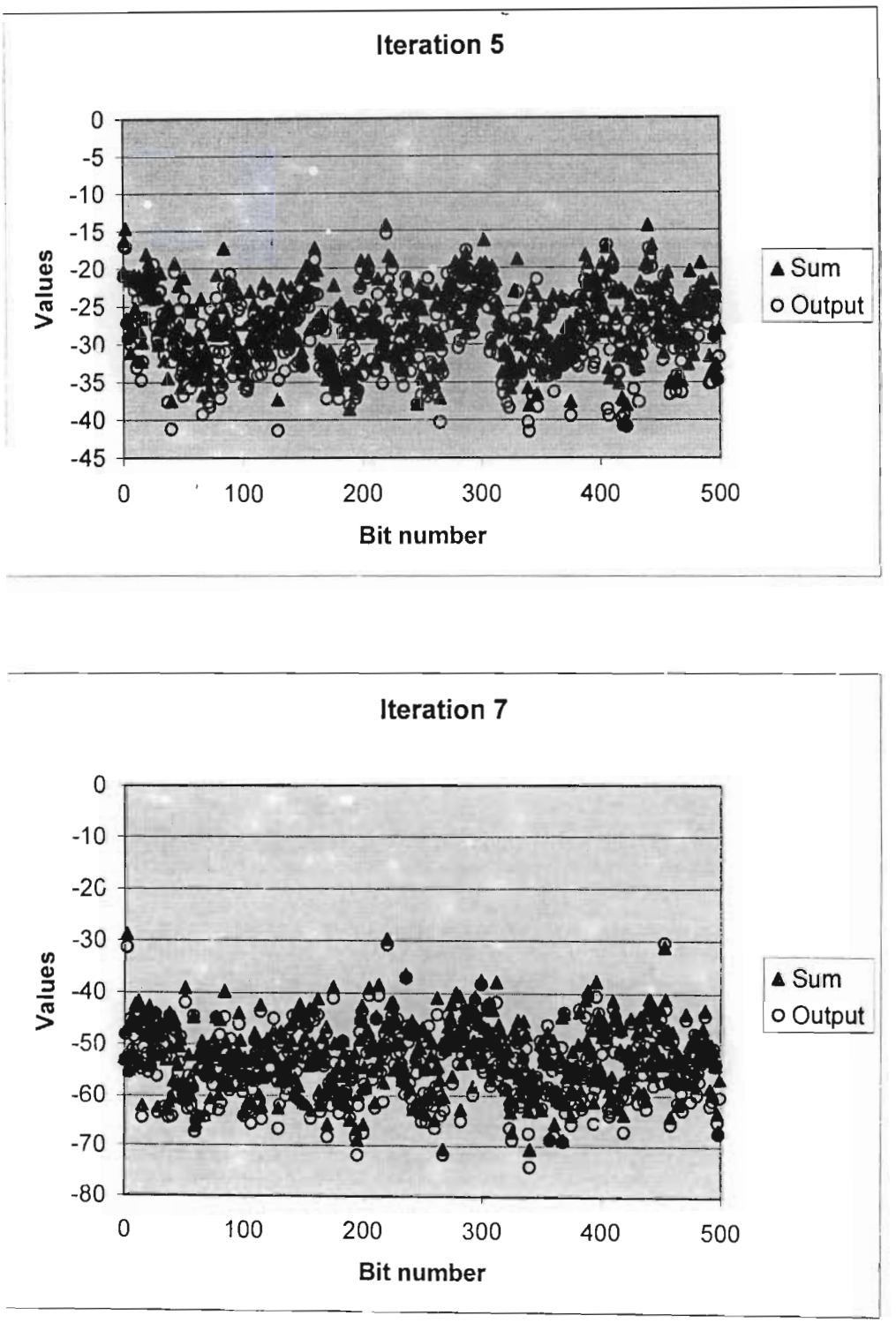


Fig. 3.3b: Soft outputs from the MAP decoder in an iterative turbo decoder for a transmitted "good" stream of all –1.

3.4 SIMULATION MODEL AND RESULTS

For all simulations, the component encoders are rate one-half, recursive, systematic convolutional encoders with memory three and octal generators $(15,13)$. A maximum of 8 turbo decoding iterations was used. Five terminating schemes were studied: CE, SCR, HDA, SDR and IHDA. For the CE criterion, the threshold $T(i)$ is set to $10^{-3}T(1)$. For the SCR criterion, q is set to 10^{-3}. For the SDR criterion, p is set to 10^{-4}. The "GENIE" case, where the information bits are known and the iteration is stopped immediately after the frame is correctly decoded, is shown as the limit of all possible schemes.

In Figs 3.4, 3.5 and 3.6, the performance in terms of the BER is shown for the interleaver sizes of 128, 1024 and 5120 respectively. All six schemes exhibit similar BER performance. The simple IHDA technique is as efficient as SDR methods in terms of the BER. However, at high SNRs, the HDA presents a slight degradation in performance compared to the other schemes.

In Figs. 3.7, 3.8 and 3.9, the average number of iterations versus the signal-to-noise ratio is shown for the interleaver sizes of 128, 1024 and 5120 respectively. It is observed that the IHDA saves more iterations than all the other schemes for small interleaver sizes. As the interleaver size increases both the CE and the HDA save more iterations, but the IHDA method remain as efficient as the SCR technique and outperforms the SDR. The HDA technique, while requiring significantly less computation than the CE method, requires extra data storage.

It should be note that the performance and efficiency of the CE method, the SCR method and the SDR method depend on the threshold, the parameter q and the parameter p respectively. Thus, they should be chosen carefully in order not to compromise the performance. The IHDA method is not dependent on any parameter.

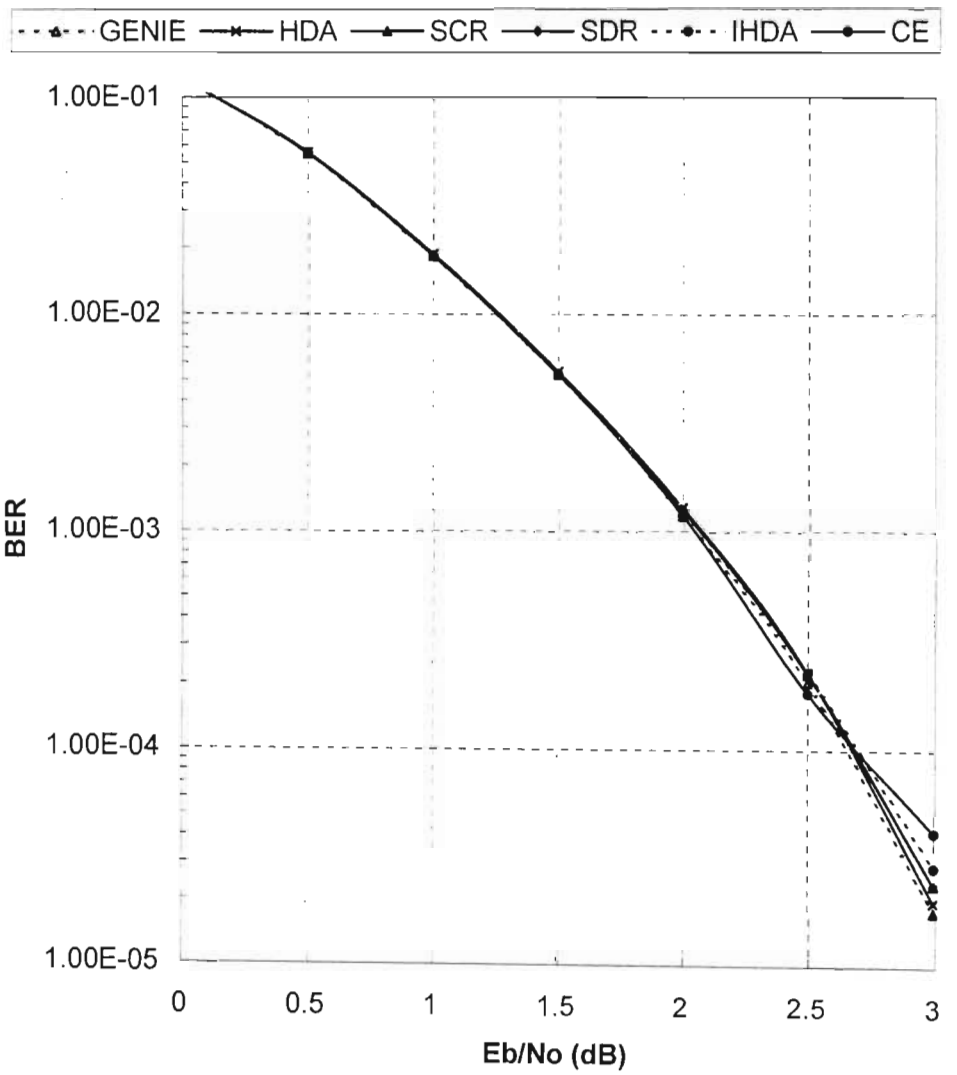


Fig. 3.4: Simulated BER performance for six stopping schemes: GENIE, CE, SCR, HDA, SDR, and IHDA. Interleaver size: 128.

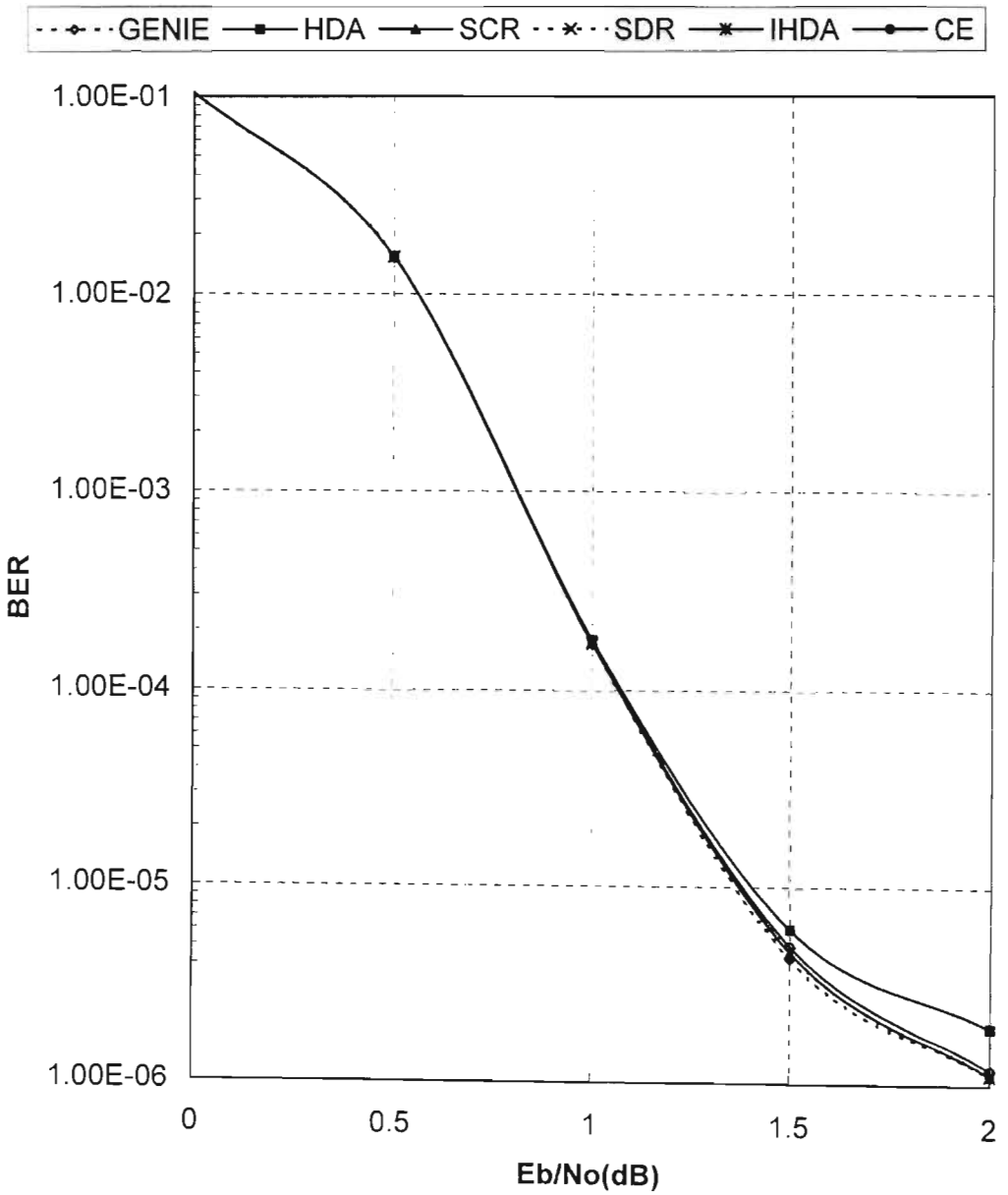


Fig. 3.5: Simulated BER performance for six stopping schemes: GENIE, CE, SCR, HDA, SDR, and IHDA. Interleaver size: 1024.

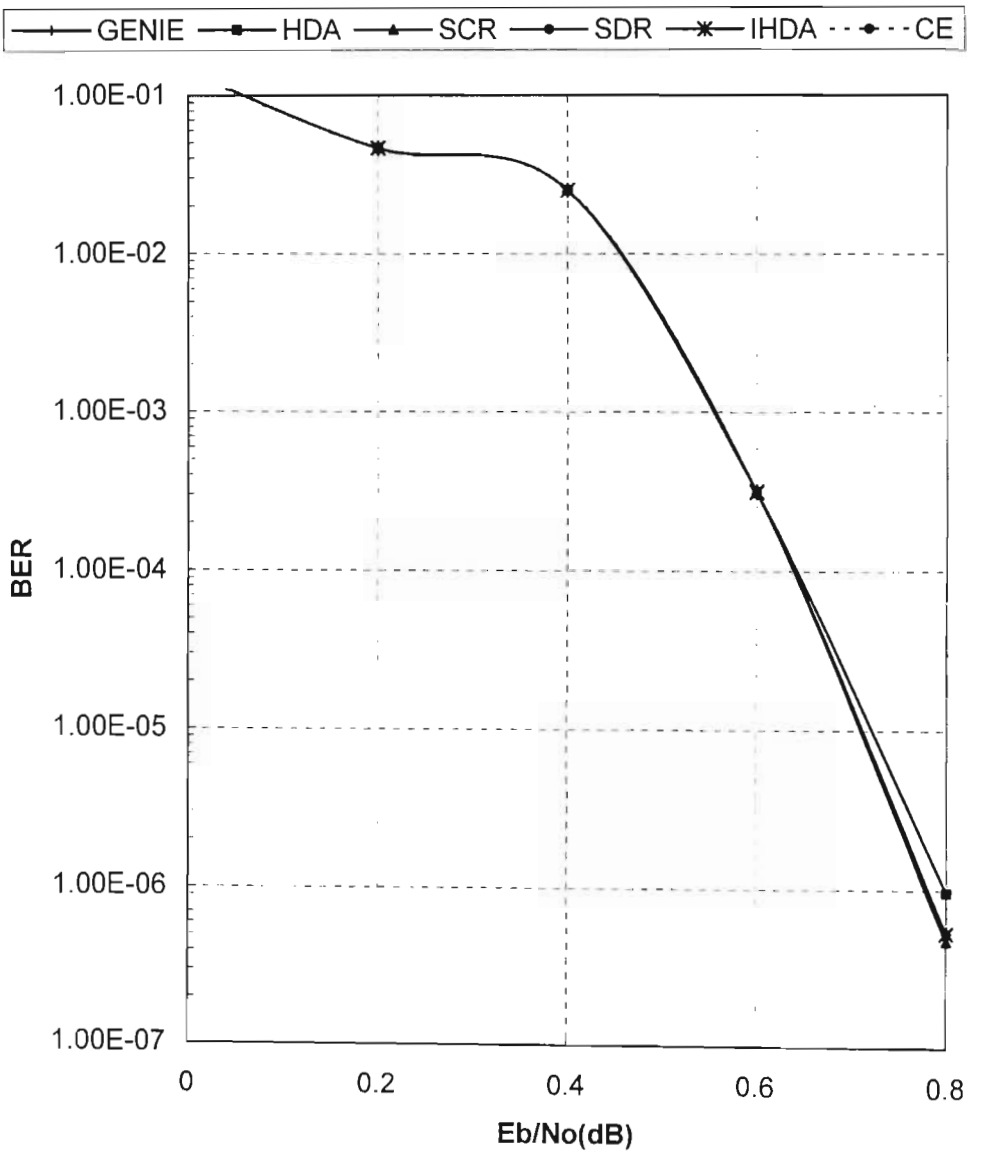


Fig. 3.6: Simulated BER performance for six stopping schemes: GENIE, CE, SCR, HDA, SDR, and IHDA. Interleaver size: 5120.

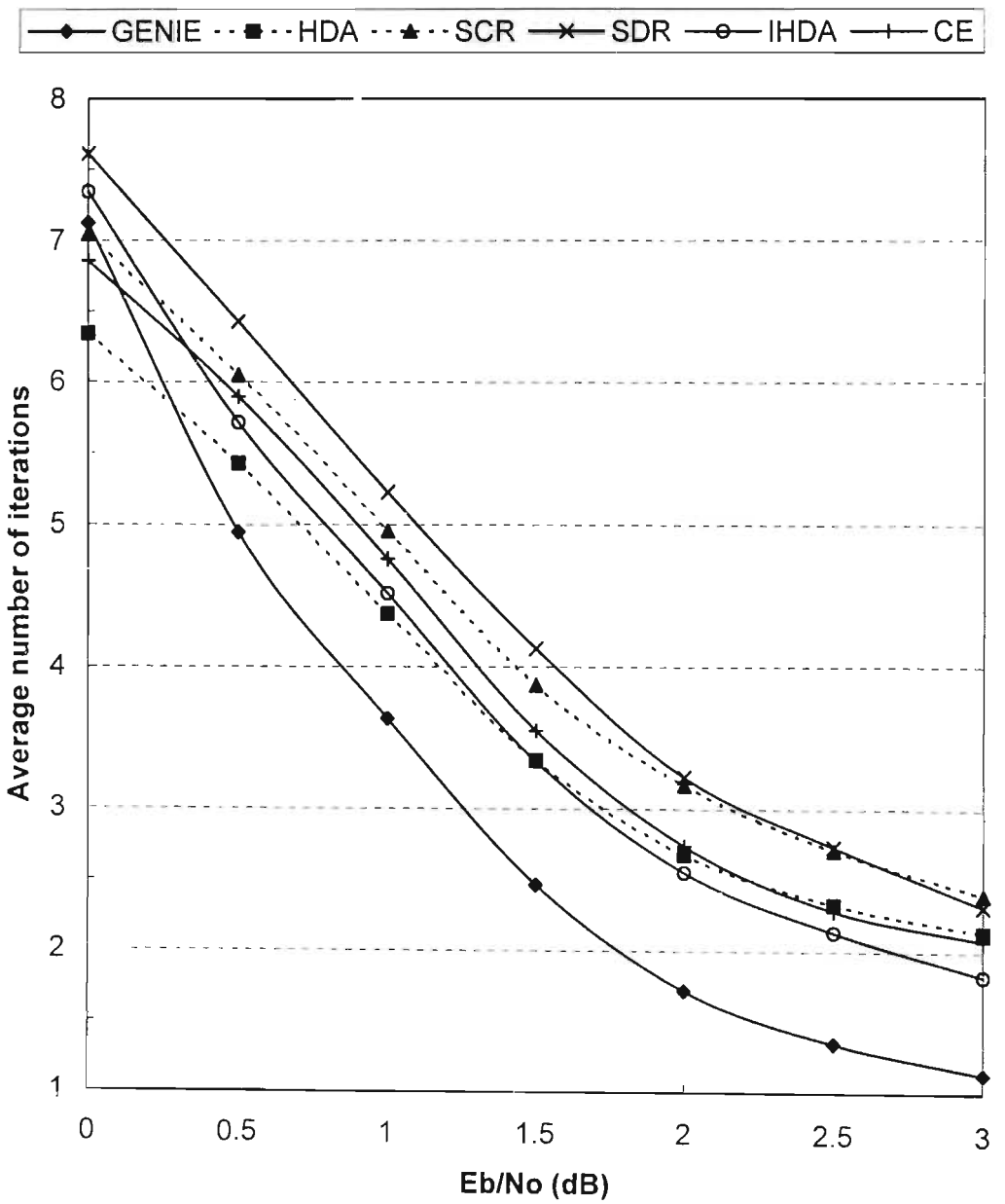


Fig. 3.7: Simulated average number of iterations for six stopping schemes: GENIE, CE, SCR, HDA, SDR, and IHDA. Interleaver size: 128.

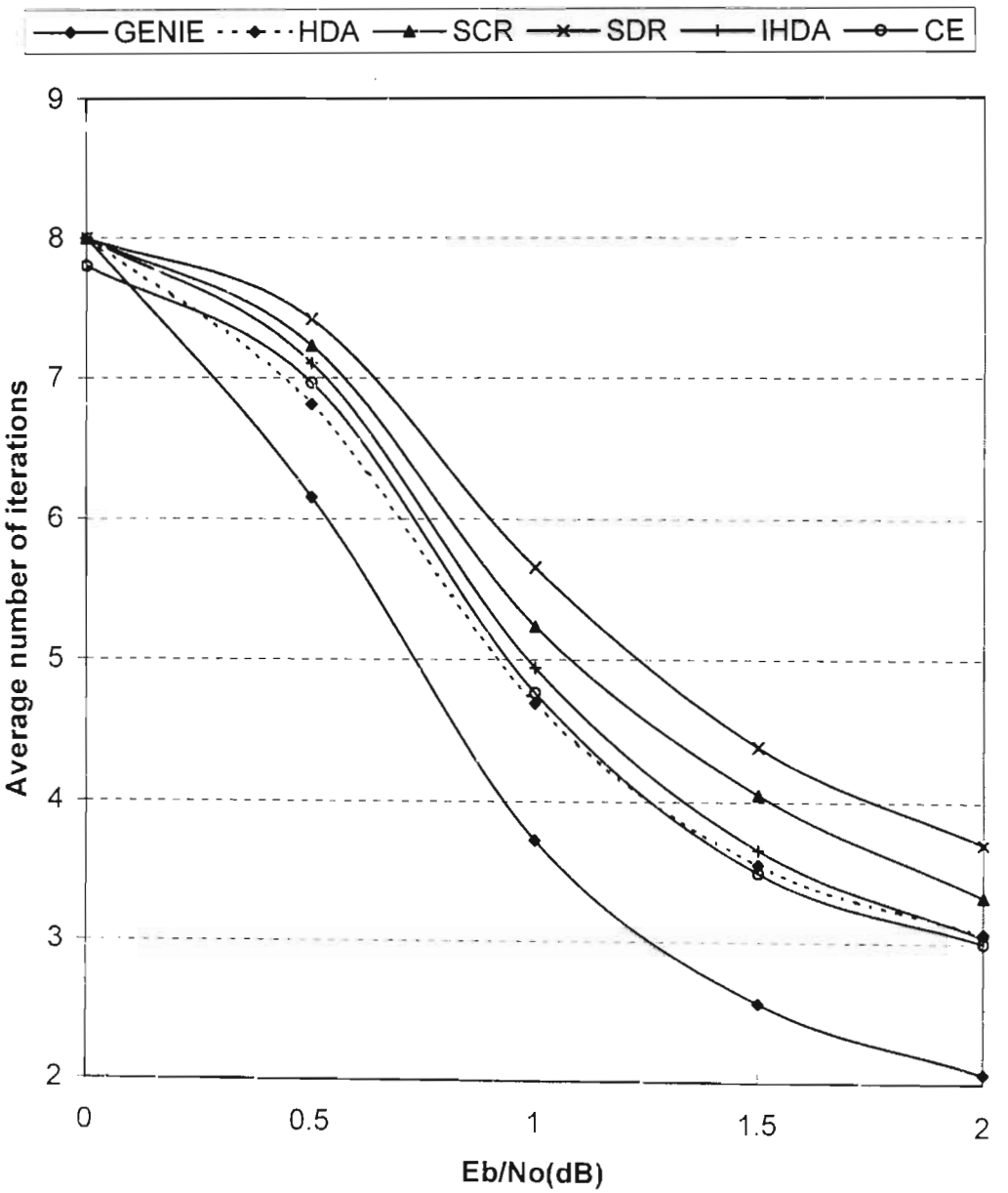


Fig. 3.8: Simulated average number of iterations for six stopping schemes: GENIE, CE, SCR, HDA, SDR, and IHDA. Interleaver size: 1024.

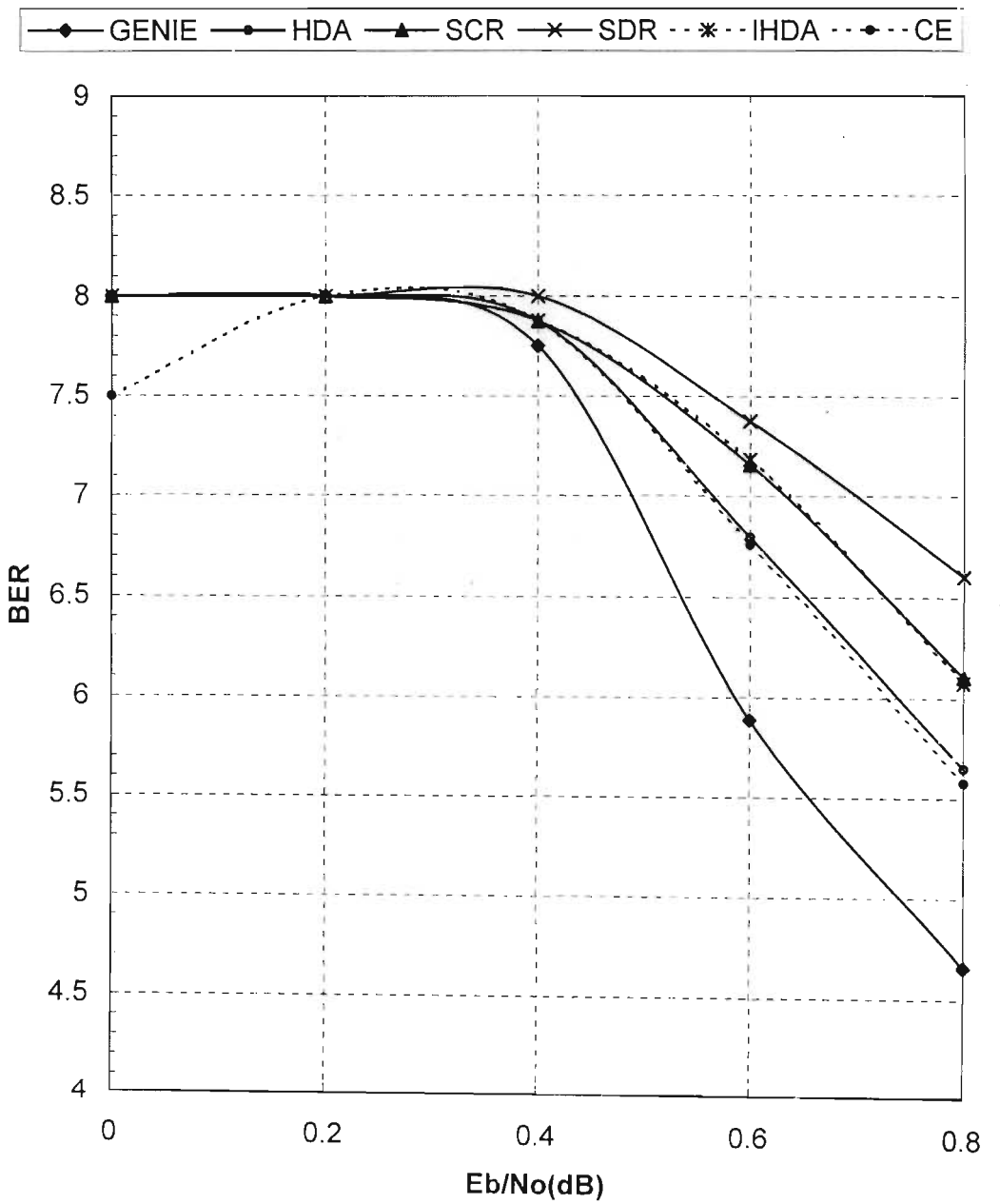


Fig. 3.9: Simulated average number of iterations for six stopping schemes: GENIE, CE, SCR, HDA, SDR, and IHDA. Interleaver size: 5120.

3.5 CONCLUSION

In this chapter, we presented a new stopping criterion; the improved hard decision aided, and compared it with four existing stopping schemes. It has been shown through simulations that the IHDA criterion performs with negligible difference from the SCR, the SDR, and the CE technique in terms of the bit-error rate. More importantly, it was observed that the IHDA saves more iterations than all the other schemes for small interleaver sizes. As the interleaver size increases both the CE and the HDA save more iterations, but the IHDA method remains as efficient as the SCR technique and outperforms the previously reported SDR. The HDA technique, while requiring significantly less computation than the CE method, requires extra data storage. Like the SCR and the SDR, the new IHDA method requires significantly less computation than the CE method. Both the SDR and the IHDA have the advantage of reduced storage requirement. The IHDA method, however, has the additional advantage that its performance is independent of the choice of any parameter.

CHAPTER 4

MODELING THE WIRELESS CHANNEL

4.1 INTRODUCTION

A model is a simplified description, especially a mathematical one, of a system or process, to assist calculations and predictions. It is necessarily less complex than the system, and has corresponding strengths and weaknesses in representing the system.

A model is *statistical* if it is used to provide a mathematical description of the system based on simple physical assumptions. The strength of statistical models is that they attempt to resemble the actual physical process. Their weakness is that this close representation is often not easy to analyze mathematically, and often no solution can be found without resorting to numerical methods. A model is *analytical* if it is used to

provide a mathematically tractable solution to system problems. The strength of analytical models is that they can provide closed-form or analytical solutions to problems. Their weakness is that often some compromise is made in terms of the representation of the actual process to make the mathematics tractable. Often, analytical models are used to generate solutions for a specific instance of a statistical model, becoming a " model of a model". A *simulation* model is used to provide Monte Carlo simulation of the system. The strength of simulation models is that they provide numerical solutions to problems while maintaining a close representation of the actual process. The weakness of simulation models is that simulation solutions are not as desirable as analytical solutions. Analytical solutions are more desirable because they show the significant parameters that affect a system, and because implementation errors in a simulation model can be difficult to detect.

In Section 4.2, we describe the J_0 fading model which is popular in representing narrowband fading. We will use the J_0 process as the model most closely representing the narrowband wireless channel in this dissertation. In Section 4.3 we look at the use of software simulation models. The section describes a fast Fourier transform method of generating correlated Rayleigh random variates. This technique will be used in Chapters 5 and 6 respectively. Finally, we conclude in Section 4.4.

4.2 THE J_0 STOCHASTIC PROCESS FOR MODELING THE WIRELESS CHANNEL

Statistical models provide a mathematical description of a system based on simple general or physical system assumptions. A common statistical model for the signal strength of communication channels is the stochastic process [42]. In this section we describe a popular stochastic process for modeling wireless channels.

Several multipath models have been suggested to explain the observed statistical characteristics of the electromagnetic fields and the associated signal envelope and phase. One of the most widely quoted is that proposed by Clarke [88] and is based on scattering propagation. We call this model the J_0 fading model, after its autocorrelation function. One station, representing the cellular system base station, is fixed in space while the other station, representing the cellular mobile user, is moving. When no direct line-of-sight transmission between the base station and the mobile exists, the signal will propagated by *scattering*, either by radio waves reflecting off obstructions or diffracting around obstacles. We first describe the J_0 fading model, starting with its physical assumptions of scattered waves, then describe the first-order statistical properties of the model, and finally describe the second-order properties of the model.

4.2.1 The Scattering Model and Fundamental Assumptions

Clarke developed this model from a suggestion by Gilbert [89] and assumed that the field incident on the mobile antenna is composed of a number of horizontally-traveling plane waves of random phase, these plane waves being vertically polarized with spatial angles of arrival and phase angles which are random and statistically independent. Furthermore the phase angles are assumed to have a uniform probability density function (PDF) in the interval $[0, 2\pi]$. A diagram of this simple model is shown in Fig. 4.1 with plane waves from stationary scatters incident on a mobile traveling in the x-direction with velocity v. The x-y plane is assumed to be horizontal.

At every receiving point we assume the signal to be the resultant of N_W plane waves. Clarke originally assumed constant amplitude waves, but Jakes has generalized the model for random amplitude waves [90]. Jakes assumes that the angles, amplitudes and phases are mutually independent and that the transmitted, scattered and received radiation is vertically polarized, so that the electric field appears normal to the horizontal plane.

We restrict our discussion of the J_0 model to vertical monopole antennae, commonly used in modern cellular systems [91]. Clarke has generalized the model to include the

two-dimensional horizontal magnetic field components, but for the vertical monopole antenna only the vertical electric field component is relevant. Furthermore, a common simplifying assumption is that the radio waves' angles of arrival are uniformly distributed, or equivalently, that the scattering is isotropic. This is generally true in urban environments.

With these assumptions, the resultant electric field component at the mobile is

$$e_z(t) = E_0 \sum_{n=1}^{N_R} A_n \cos[(\omega_f + \omega_{Dn})t + \phi_n], \tag{4.1}$$

where $E_0 A_n$ is the real amplitude of the nth wave in the electric field, ω_f is the carrier frequency of the transmitted signal, ϕ_n is the uniformly distributed random phase angle, and ω_{Dn} is the Doppler shift in the nth wave. This Doppler shift is

$$\omega_{Dn} = \frac{2\pi v}{\lambda}\cos(\alpha_n), \tag{4.2}$$

where v is the speed of the mobile, λ is the wavelength of the carrier frequency, and α_n

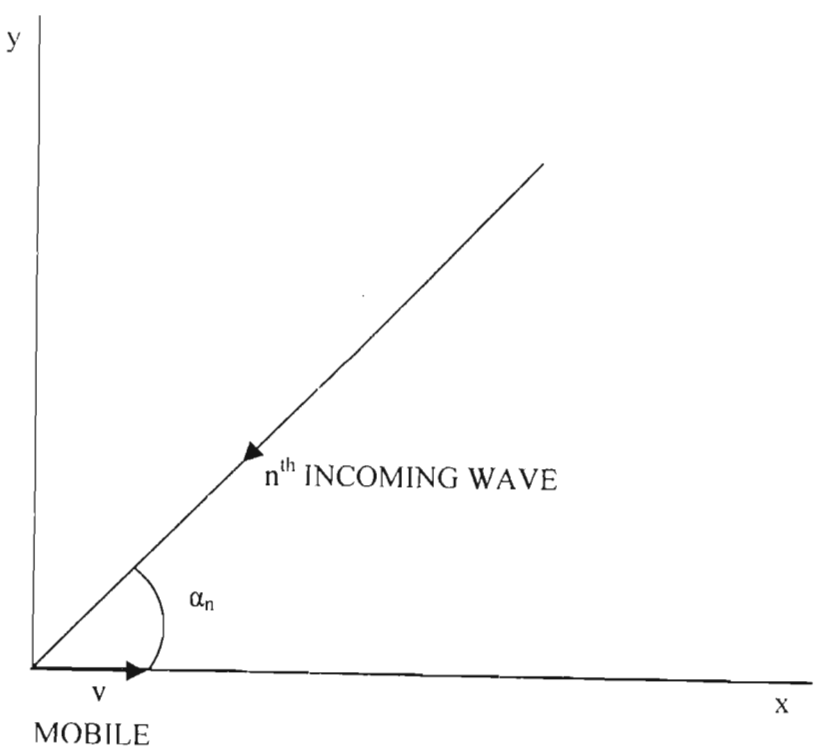


Fig. 4.1: A typical component incident wave on the mobile receiver

is the azimuthal angle between the mobile direction and direction of the arriving wave, as in Fig. 4.1. The amplitudes A_n are normalized so that the ensemble average

$$E\left\{\sum_{n=1}^{N_W} A_n^2\right\}=1. \tag{4.3}$$

From the Central Limit Theorem [92], for large N_W the electric field component is a complex Gaussian stochastic process, and because the Doppler shift is bounded by $\pm\frac{2\pi v}{\lambda}$, it is also a narrowband process. Thus $e_z(t)$ can be described as

$$e_z(t)=e_{zc}(t)\cos(\omega_f t)-e_{zs}(t)\sin(\omega_f t), \tag{4.4}$$

where the in-phase and quadrature narrowband equivalent components are real Gaussian processes,

$$e_{zc}(t)=E_0\sum_{n=1}^{N_W} A_n\cos(\omega_{Dn}t+\phi_n), \tag{4.5}$$

$$e_{zs}(t)=E_0\sum_{n=1}^{N_W} A_n\sin(\omega_{Dn}t+\phi_n). \tag{4.6}$$

Jakes shows that for large N_W, $e_{zc}(t)$ and $e_{zs}(t)$ have zero ensemble mean, and equal variance

$$E\left\{e_{zc}(t)^2\right\}=E\left\{e_{zs}(t)^2\right\}=b=\frac{E_0^2}{2}, \tag{4.7}$$

where the expectation is taken over α_n, ϕ_n and A_n. Clarke shows that the two processes $e_{zc}(t)$ and $e_{zs}(t)$ are mutually independent of each other. Over short period of time, Clarke assumes $e_{zc}(t)$ and $e_{zs}(t)$ are wide-sense stationary (WSS), and thus are strict-sense stationary (SSS), as they are Gaussian. Over long periods of time, Clarke has shown that stationarity cannot be assumed, according to experimental measurements.

4.2.2 First-Order Properties

Practical radio receivers do not normally have the ability to detect the components $e_{zc}(t)$ and $e_{zs}(t)$; they respond to the envelope and phase of the complex signal $e_z(t)$. As $e_{zc}(t)$ and $e_{zs}(t)$ are Gaussian, the envelope of $e_z(t)$, $r(t)=|e_z(t)|$ is given by

$$r(t)=\sqrt{e_{zc}(t)^2+e_{zs}(t)^2}\ . \tag{4.8}$$

It is well known [93] that $r(t)$ is Rayleigh distributed, so we call $r(t)$ a *Rayleigh process*. We reserve this term for Rayleigh distributed processes with identical correlation functions. In the J_0 model, $r(t)$ represents the amplitude of the mobile user's received signal envelope as a function of time. For a mobile traveling at a constant velocity, there is a direct correspondence between variations in $r(t)$ as a function of distance and variation in $r(t)$ as a function of time. Conceptually, we can generalize the traditional additive Gaussian noise channel model block diagram by post-multiplying the channel by $r(t)$, as shown in Fig. 4.2. As (4.8) is a memoryless transformation, it can be shown that $r(t)$ is also SSS [92]. The first order density function of $r(t)$ is

$$f_r(r;t)=\frac{r(t)}{b}\exp\left(-\frac{r(t)^2}{2b}\right)U[r(t)]\ , \tag{4.9}$$

where $U(x)$ is the unit step function,

$$U(x)=\begin{cases}1, & x\geq 0\\ 0, & x<0.\end{cases} \tag{4.10}$$

The mean of the Rayleigh process $r(t)$ is

$$\begin{aligned}\mu_r(t)&=\int_{-\infty}^{\infty} r f_r(r;t)dr\\ &=\sqrt{\frac{\pi}{2}}b\ .\end{aligned} \tag{4.11}$$

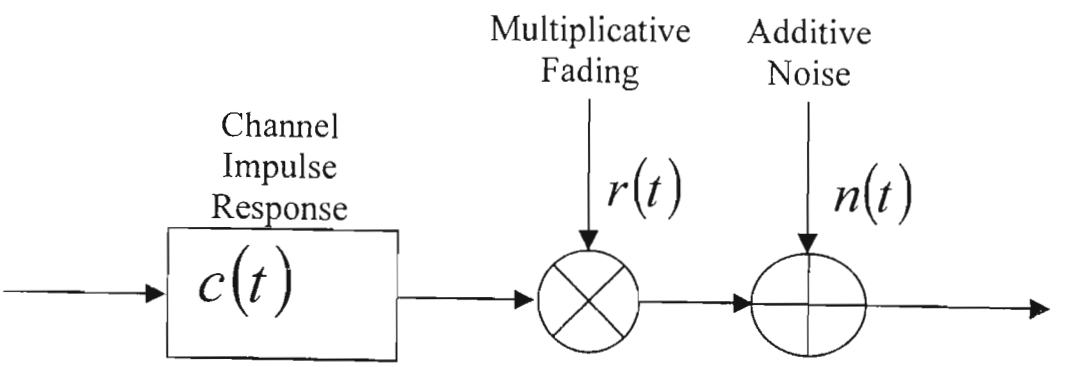


Fig. 4.2: The Multiplicative Fading Communications Channel Model.

4.2.3 Second-Order Properties

Clarke shows that for large N_W and for isotropic scattering, the power spectra of the electric field component $e_z(t)$ is

$$S_{e_z}(f)=\begin{cases}\dfrac{b}{\pi f_D}\dfrac{1}{\sqrt{1-\left(\dfrac{f-f_c}{f_D}\right)^2}}, & f_c-f_D\le f\le f_c+f_D\\ 0, \quad \text{otherwise,}\end{cases} \tag{4.12}$$

where $f_D=\dfrac{v}{\lambda}$ is the maximum Doppler frequency spread. We can take the inverse cosine and sine transform to obtain the electric field component autocorrelation functions,

$$E\{e_{zc}(t)e_{zc}(t+\tau)\}=E\{e_{zs}(t)e_{zs}(t+\tau)\}=\int_{f_c-f_D}^{f_c+f_D}S_{e_z}(f)\cos[2\pi(f-f_c)\tau]df$$
$$=bJ_0(2\pi f_D\tau), \tag{4.13}$$

and

$$E\{e_{zc}(t)e_{zs}(t+\tau)\}=-E\{e_{zs}(t)e_{zc}(t+\tau)\}=\int_{f_c-f_D}^{f_c+f_D}S_{e_z}(f)\sin[2\pi(f-f_c)\tau]df$$
$$=0, \tag{4.14}$$

where $J_0(x)$ is the Bessel function of the first kind of order zero, which according to [94] can be written as

$$J_0(x)=\sum_{k=0}^{\infty}\frac{\left(-\frac{1}{4}x^2\right)^k}{(k!)^2}. \tag{4.15}$$

In [95], it is shown that the autocorrelation function of the envelope, $r(t)$, can be derived from the autocorrelation functions of the underlying Gaussian electric field processes to be

$$R_r(\tau)=\frac{\pi b}{2}\,{}_2F_1\left(-\frac{1}{2},-\frac{1}{2};1;J_0(2\pi f_D\tau)^2\right), \tag{4.16}$$

where ${}_2F_1(\alpha,\beta;\gamma;z)$ is the Gaussian hypergeometric function, which can be represented by [94]

$${}_2F_1(\alpha,\beta;\gamma;z)=\frac{\Gamma(\gamma)}{\Gamma(\alpha)\Gamma(\beta)}\sum_{k=0}^{\infty}\frac{\Gamma(\alpha+k)\Gamma(\beta+k)}{\Gamma(\gamma+k)}\frac{z^k}{k!}, \tag{4.17}$$

and where $\Gamma(x)$ is the gamma function, defined in [94] as

$$\Gamma(x)=\begin{cases}\int_0^\infty t^{x-1}\exp(-t)dt, & \mathrm{Re}(x)\\ \lim_{n\to\infty}\dfrac{n!n^x}{x(x+1)...(x+n)}, & x\neq 0,-1,-2,...\end{cases} \tag{4.18}$$

Similarly, the autocovariance function of $r(t)$ is

$$C_r(\tau)=\frac{\pi b}{2}\left[\frac{3}{2}{}_2F_1\left(-\frac{1}{2},-\frac{1}{2};1;J_0(2\pi f_D\tau)^2\right)-1\right]. \tag{4.19}$$

It is well known that the Gaussian process is completely defined in terms of its mean and autocorrelation functions [92]. As the Rayleigh process is formed by the deterministic and memoryless transform in (4.8) of two mutually independent zero-mean Gaussian processes with identical autocorrelation functions, the Rayleigh process is completely defined by the autocorrelation function of the underlying Gaussian process.

4.2.4 Variation of the J_0 Fading Model

The J_0 fading model contains simplifying assumptions and omissions about the physical wireless channel for mathematical simplicity. The principal constraint on the model treated by Clarke is its restriction to the case when the incoming waves are traveling horizontally, i.e. it is a two-dimensional model. It is obvious that in practice, diffraction and scattering from oblique surfaces create waves that do not travel horizontally. It is equally clear, however, that those waves which make a major contribution to the received signal do indeed travel in an approximately horizontal direction because the two-dimensional model successfully explains almost all the observed properties of the signal envelope and phase. Nevertheless, there are differences between what is observed and what is predicted; in particular the observed envelope spectrum shows differences at low frequencies and around $2f_m$, where f_m is the maximum Doppler frequency. Complicated models have been proposed that may more accurately model the wireless channel by taking into account other radio phenomena like shadowing, and by challenging Clarke's assumptions that all incoming waves travel horizontally and that the scattering is

isotropic. We focus the dissertation on the J_0 fading model due to its simplicity and popularity. We discuss two variations commonly found in literature.

The J_0 fading model has a Rayleigh first-order distribution. When the signal from the base station to the mobile user has a direct line-of-sight path component, the Rayleigh distribution is no longer suitable [88]. The Ricean distribution has been used to model the wireless channel in this instance, so that the first-order distribution of $r(t)$ is

$$f_r(r;t)=\frac{r(t)}{b}\exp\left[-\frac{r(t)^2+s^2}{2b}\right]I_0\left(\frac{r(t)s}{b}\right)U(r(t)), \tag{4.20}$$

where s is a parameter on the strength of the line-of-sight component, and I_0 is the modified Bessel function of the first kind of order zero, which according to [94] can be represented by

$$I_0(x)=\sum_{k=0}^{\infty}\frac{\left(\frac{1}{2}x^2\right)^k}{(k!)^2}. \tag{4.21}$$

A more general distribution also commonly used is the Nakagami-m distribution, which includes the Rayleigh distribution as a special case [96]. This distribution was originally conceived to model fast fading in ionospheric and tropospheric propagation. The first-order distribution of $r(t)$ with Nakagami-m distribution is

$$f_r(r;t)=\frac{r(t)^{2m-1}}{\frac{1}{2}\left(\frac{2b}{m}\right)^m\Gamma(m)}\exp\left[-\frac{r(t)^2}{\frac{2b}{m}}\right]U(r(t)), \tag{4.22}$$

where m is the fading parameter of the Nakagami distribution. When m = 1/2 the first-order distribution is one-sided Gaussian and when m = 1 the first-order distribution is Rayleigh. The Nakagami-m distribution has been used to model the wireless channel [97].

4.3 SIMULATION MODELS

A common tactic in communications research is to compare theoretical results with simulated results. Rarely are theoretical results compared to the actual process. First, it is usually less expensive and time consuming to construct a simulation than conduct field experimentation. Second, simulation gives control over the testing environment to allow for balanced performance comparisons between possible solutions. Software simulations are inexpensive and easy to construct. A number of different techniques to simulate correlated Rayleigh random variates (J_0 fading process) have been proposed and used in the literature. These range from the inverse Discrete Fourier Transform (IDFT) [98, 99] to the filtering of two independent zero-mean white Gaussian noise (WGN) processes [100-102] and the superposition of Sinusoids ([90] and [103]). Rapid generation of those variates is of considerable practical value in communication systems simulation. A good summary of the different techniques is given in [104, 105].

In Smith's [98] FFT based algorithm for generating correlated Rayleigh random variates, quadrature correlated Gaussian random variates are generated at an intermediate stage. Smith's paper gives an algorithm only; neither the basis of the method, nor the limitations of the method are discussed. Young and Beaulieu [99] have modified Smith's algorithm for greater computational efficiency and have provided a statistical analysis of the technique. From this analysis, it is clear that the filter specification of [98] is not unique. Beaulieu and C.C. Tan [106, 107] have studied an FFT technique to generate bandlimited Gaussian random variates. They showed that the FFT method only requires $O(n)$ memory elements and $O(n \log_2 n)$ operations to generate each sequence of variates. They specifically examined the important case of ideally bandlimited Gaussian processes, but showed the method can be generalized to any correlation function. In this Section, we apply the FFT based method of [106, 107] to generate correlated Rayleigh random variates. We show that the aliasing error introduced in the correlation function of the generated variates is negligible for practical simulation.

4.3.1 Background

Rayleigh-distributed variates are obtained as the magnitude of variates having a complex distribution. The real and imaginary parts of the complex sequence must be independent and each must have zero mean for Rayleigh fading. In a widely accepted model, the Gaussian process is assumed to have a bandlimited non-rational power spectrum density (PSD), given by [90]

$$S_y(f)=\begin{cases}\dfrac{1}{\pi f_D\sqrt{1-(f/f_D)^2}}, & |f|<f_D\\ 0, & |f|\geq f_D\end{cases} \tag{4.23}$$

where f represents the frequency in Hz and the parameter f_D is the maximum Doppler frequency in Hz, given by $f_D=v/\lambda$, where v is the vehicle velocity in m/s and λ is the carrier wavelength in m. The normalized (unit-variance) continuous time autocorrelation of the scattered received signal under these conditions is [90]

$$R_y(\tau)=J_0(2\pi f_D\tau). \tag{4.24}$$

where τ is the sequence separation, in seconds, between observation times and $J_0(.)$ is the zero-order Bessel function of the first kind.

4.3.2 The Problem of Generating Coloured Gaussian Noise

Let $u(t)$ denote a wide-sense stationary (WSS), zero-mean, unit-variance, continuous-time, white Gaussian noise process, and let $y(t)$ denote the stochastic process at the output of filter $H(w)$ with input $u(t)$. It can be shown [92] that passing zero mean WSS white Gaussian noise through a linear time-invariant filter with transfer function $H(w)$ results in a zero-mean WSS Gaussian noise with PSD

$$S_y(w)=|H(w)|^2. \tag{4.25}$$

If $S_y(w)$ is bandlimited to frequency B, then $S_y(w)=0$ for $|w|\geq B$. The autocorrelation function and PSD are related through the continuous-time Fourier transform [92], so the autocorrelation function of the noise is

$$R_y(\tau)=F^{-1}\{S_y(w)\}=F^{-1}\{|H(w)|^2\}. \tag{4.26}$$

The process $y(t)$ is thus WSS, continuous-time and Gaussian with autocorrelation $R_y(\tau)$ and PSD $S_y(w)$.

Computer simulation of communication systems requires discrete-time modeling of signals and noise. Monte Carlo simulations require a large number of trials to estimate system parameters accurately. A single trial will have an integer number of dependent symbols, α. We require N_T samples to represent each symbol, where N_T is an integer. Let T denote the symbol time interval and so the simulation sampling rate is $\frac{1}{T_s}=\frac{N_T}{T}$. The discrete-time sampled process $y[n]=y(nT_s)$ will be WSS and zero-mean with correlation function $R_y[d]$. It can be shown that [92].

$$R_y[d]=R_y(dT_s). \tag{4.27}$$

Correspondingly, the PSD of $y[n]$ is the discrete-time Fourier transform (DTFT) of $R_y[d]$, which can be shown to be [92]

$$S_y(e^{jw})=\frac{1}{T_s}\sum_{n=-\infty}^{\infty}S_y\left(\frac{w+2\pi n}{T_s}\right). \tag{4.28}$$

Also note that corresponding to (4.25), if $H(e^{jw})$ is the DTFT of $h[n]$ then [92],

$$S_y(e^{jw})=|H(e^{jw})|^2 \tag{4.29}$$

4.3.3 The Method

To model the continuous-time WSS Gaussian noise for $0\leq t\leq \alpha T$, the problem is therefore to generate $N_s=\alpha N_T$ discrete-time WSS Gaussian noise variates with a discrete autocorrelation function

$$R_y[d] = J_0(2\pi f_m |d|) \tag{4.30}$$

In (4.30) f_m is the maximum Doppler frequency normalized by the sampling rate, and d is the sampling lag as previously defined.

We assume that the simulation sampling rate is much greater than the Nyquist rate, so the DTFT of $R_y[d]$ is the 2π-periodic waveform [108]

$$S_y(e^{jw}) = \begin{cases} \dfrac{1}{\pi f_m \sqrt{1-(w/2\pi f)^2}} & \text{if } |w| < 2\pi f_m \\ 0 & \text{if } 2\pi f_m \le |w| \le \pi \end{cases} \tag{4.31}$$

over $-\pi \le w \le \pi$.

Since linear transformations of jointly Gaussian random variates generate jointly Gaussian random variates [92], the random variates can be generated in the frequency domain and transformed into the time domain using the discrete Fourier transform (DFT). One way of proceeding would be to transform the autocorrelation sequence $R_y[d]$ into the corresponding discrete frequency PSD sequence. The square root of this PSD would then give a filter characteristic possessing the desired autocorrelation $R_y[d]$. This cannot, however, be done in general, because the DFT of the finite time windowed sequence $R_y[d]$ has negative values owing to the Gibbs phenomenon. To avoid this problem, we start with the desired discrete frequency PSD. We use the DFT to sample $S_y(e^{jw})$ [108]. Using the filter symmetry contributed by [99], we must ensure that we sample $S_y(e^{jw})$ so that the frequency domain sequence $S_y[k]$ maintains a symmetry such that $S_y[k] = S_y[2N_s - k]$, $0 < k < N_s$, leaving $S_y[0]$ and $S_y[N_s]$ unconstrained. This ensures that the IDFT of $S_y[k]$ will be real [108].

$$S_y[k] = S_y(e^{jw})\Big|_{w=\frac{2\pi k}{2\alpha N_T}} \tag{4.32}$$

$$= \begin{cases} \dfrac{1}{\pi f_m \sqrt{1-(k/2\alpha N_T f_m)^2}}, & 0 \le k < 2\alpha N_T f_m \\ 0, & 2\alpha N_T f_m \le k \le 2N_s - 2\alpha N_T f_m \\ \dfrac{1}{\pi f_m \sqrt{1-(2N_s - k/2\alpha N_T f_m)^2}}, & 2N_s - 2\alpha N_T f_m < k \le 2N_s - 1 \end{cases} \quad (4.33)$$

Since $R_y[d]$ is not of finite length, the inverse DFT (IDFT) of $S_y(e^{jw})$ will be a time-aliased version of $R_y[d]$. The IDFT of (4.33) will be an aliased version of $\tilde{R}_y[d] = J_0(2\pi f_m |d|)$, $d = 0,1,.....,2N_s - 1$.

4.3.4 Procedure

Given $X_1[k]$ and $X_2[k]$, $0 \le k \le 2N_s - 1$, two uncorrelated sequences of $2N_s$ zero-mean, unit-variance, white Gaussian variates, we generate $Y_1[n]$ and $Y_2[n]$, two sequences of N_s random, zero-mean Gaussian samples having discrete frequency PSD $S_y[k]$ in the following steps.

Step 1. Generate and store $2N_s$ spectral weights that are the square roots of the magnitudes of the sampled power spectral density $S_y[k]$ scaled by the number of samples,

$$H[k] = \sqrt{2N_s S_y[k]}\ ,\ k = 0,1,2,.....,2N_s - 1\ . \quad (4.34)$$

This stage will take $2N_s$ multiplications and $2N_s$ square roots for a total of $4N_s$ operations. This stage requires $2N_s$ memory elements to store $H[k]$.

Step 2. Add $X_1[k]$ and $X_2[k]$ in quadrature and multiply them by the spectral weight $H[k]$ to form $V[k]$ so

$$V[k] = H[k](X_1[k] + jX_2[k])\ ,k = 0,1,.....,2N_s - 1. \quad (4.35)$$

Step 3. Apply the complex inverse FFT (IFFT) to $V[k]$ to obtain the first N_s points of its IDFT, which we define as sequence S_n

$$S_n = \frac{1}{2N_s} \sum_{k=0}^{2N_s-1} V[k] \exp\left(\frac{j2\pi}{2N_s} kn\right) \quad , n = 0,1,\ldots\ldots,N_s - 1. \qquad (4.36)$$

Generate two sets of N_s correlated zero-mean Gaussian random variates by separating S_n into its real and imaginary sequences. The two sets of N_s are:

$$Y_1[n] = \mathrm{Re}(S_n) \quad , n = 0,1,\ldots\ldots,N_s - 1, \qquad (4.37a)$$

and

$$Y_2[n] = \mathrm{Im}(S_n) \quad , n = 0,1,\ldots\ldots,N_s - 1. \qquad (4.38b)$$

The sequences $Y_1[n]$ and $Y_2[n]$ are then used to generate the Rayleigh-distributed variates.

With a FFT algorithm like the decimation-in-time FFT [108] this complex IDFT for the first N_s points of $V[k]$ will take $2N_s \log_2(2N_s)$ complex multiplication and $N_s \log_2(2N_s)$ complex additions, or $14N_s \log_2(2N_s)$ real operations. Storing S_n requires N_s memory elements. Thus, the algorithm requires an asymptotic order of $O(N_s)$ operations for initialization and $O(N_s \log_2 N_s)$ operations for the generation phase. The entire procedure requires $O(N_s)$ memory elements.

4.3.5 Analysis of the Method

4.3.5.1 Statistical Properties of $Y_1[n]$ and $Y_2[n]$

It can be shown [107] that $Y_1[n]$ and $Y_2[n]$ are Gaussian and zero-mean, that the autocorrelation function of $Y_1[n]$ and $Y_2[n]$ is the IDFT of $S_y[k]$, and that the cross-correlation between $Y_1[n]$ and $Y_2[n]$ is zero.

4.3.5.2 Aliasing Error Analysis

As stated earlier, it is not obvious that the aliasing error incurred by frequency sampling ideally bandlimited white noise is small. However, for sampling rates that are integer

multiples of the noise bandwidth as is the case for α number of samples per trial, the aliasing error is finite and small as will be shown.

We will analyze the aliasing error. The IDFT of (4.33) is

$$\tilde{R}_y[d] = \frac{1}{2N_s} \sum_{k=0}^{2N_s-1} S_y[k] \exp\left(\frac{j2\pi}{2N_s} kd\right). \tag{4.39}$$

The absolute error between $\hat{R}_y[d]$ and the desired correlation function is

$$\varepsilon_d = \left| R_y[d] - \tilde{R}_y[d] \right|. \tag{4.40}$$

In Fig. 4.3, the maximum correlation error is plotted as a function of the number of symbols per trial, α, for different values of f_m and N_T = 15. These values of f_m encompass the whole range of mobile communication environments ranging from cellular telephony to personal communication system (PCS). Observe that the maximum error in the correlation function is insignificant for practical applications. For example, for f_m = 0.1, when 100 symbols are generated in one trial, the aliasing error is 3.83×10^{-3}. The aliasing error decreases when more symbols are generated per trial, down to 1.2×10^{-3} for 1000 symbols. It is, however, observed that the correlation error increases as the parameter f_m decreases. This suggest that, for smaller values of f_m, the number of samples per symbol, N_T, must be higher to obtain a negligible correlation error. That this is the case can be seen in Fig. 4.4 where the maximum correlation error is plotted as a function of the number of symbols per trial, α, for different values of N_T and f_m= 0.005. Fig. 4.5 shows both the desired correlation function $R_y[d]$ and $\tilde{R}_y[d]$ for f_m=0.1, N_T=15 and α=100 symbols per trial. Note that the two are virtually indistinguishable.

An extreme case of the Doppler spectrum is the uniform model, which has been studied in [109, 110]. For this model,

$$S_y''(f) = \begin{cases} \dfrac{1}{2f_D} & |f| < f_D \\ 0 & |f| \geq f_D \end{cases} \tag{4.41}$$

The continuous correlation function and discrete correlation of this model are

$$R_y^u(\tau) = \text{sinc}(2\pi f_D \tau), \tag{4.42}$$

and

$$R_y^u[d] = \text{sinc}(2\pi f_m d) \tag{4.43}$$

respectively.

The discrete frequency PSD, $S_y^u[k]$ are

$$S_y^u[k] = \begin{cases} \dfrac{1}{2f_m}, & 0 \le k \le 2\alpha N_T f_m \\ 0, & 2\alpha N_T f_m \le k \le 2N_s - 2\alpha N_T f_m \\ \dfrac{1}{2f_m}, & 2N_s - 2\alpha N_T f_m < k \le 2N_s - 1 \end{cases} \tag{4.44}$$

The IDFT of (4.42) is

$$\tilde{R}_y^u[d] = \frac{1}{2N_s} \sum_{k=0}^{2N_s-1} S_y^u[k] \exp\left(\frac{j2\pi}{2N_s} kd\right)$$

$$= \frac{1}{2N_s} \left\{ \sum_{k=0}^{2\alpha N_s - 1} S_y^u[k] \left[\exp\left(\frac{j2\pi}{2N_s} kd\right) + \exp\left(-\frac{j2\pi}{2N_s} kd\right) \right] - S_y^u[0] \right\}. \tag{4.45}$$

Using the identity $\exp(jx) + \exp(-jx) = 2\cos(x)$, we obtain

$$\tilde{R}_y^u[d] = \frac{1}{4N_s f_m} \left\{ 2 \sum_{k=0}^{2N_s-1} \cos\left(\frac{2\pi}{2N_s} kd\right) - 1 \right\}. \tag{4.46}$$

Using the identity given in [111] that

$$\sum_{k=0}^{n} \cos kx = \frac{1}{2} \left(1 + \frac{\sin\left(\left(n + \frac{1}{2}\right)x\right)}{\sin\left(\frac{x}{2}\right)} \right), \tag{4.47}$$

gives

$$\tilde{R}_y^u[d] = \frac{1}{4N_s f_m} \frac{\sin\left(2\pi f_m d - \frac{2\pi}{4N_s} d\right)}{\sin\left(\frac{2\pi}{4N_s} d\right)}. \tag{4.48}$$

The trigonometric identity $\sin(a-b)=\sin a\cos b-\cos a\sin b$ is used to obtain

$$\tilde{R}_y^u[d]=\frac{1}{4N_s f_m}\left[\sin(2\pi f_m d)\cot\left(\frac{2\pi}{4N_s}d\right)-\cos(2\pi f_m d)\right]. \tag{4.49}$$

The absolute error between $\tilde{R}_y^u[d]$ and the desired correlation is

$$\varepsilon_d^u=\left|R_y^u[d]-\tilde{R}_y^u[d]\right| . \tag{4.50}$$

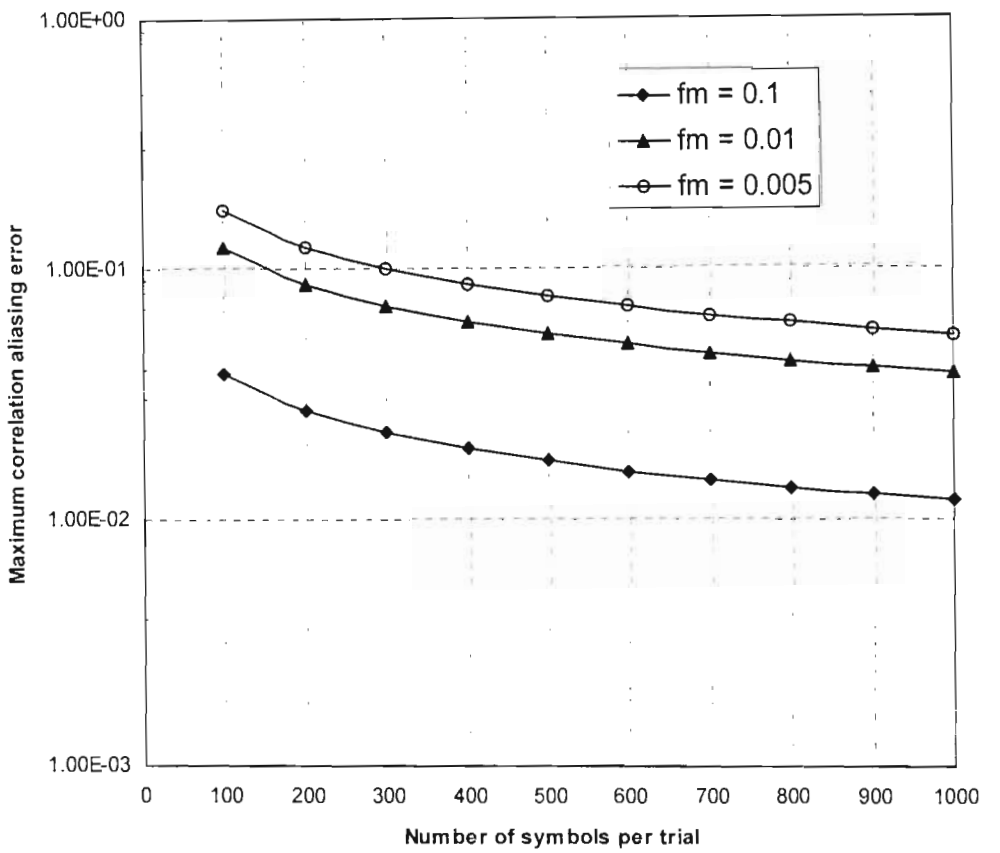


Fig. 4.3: Maximum correlation error due to aliasing for different values of *fm* and $N_T = 15$.

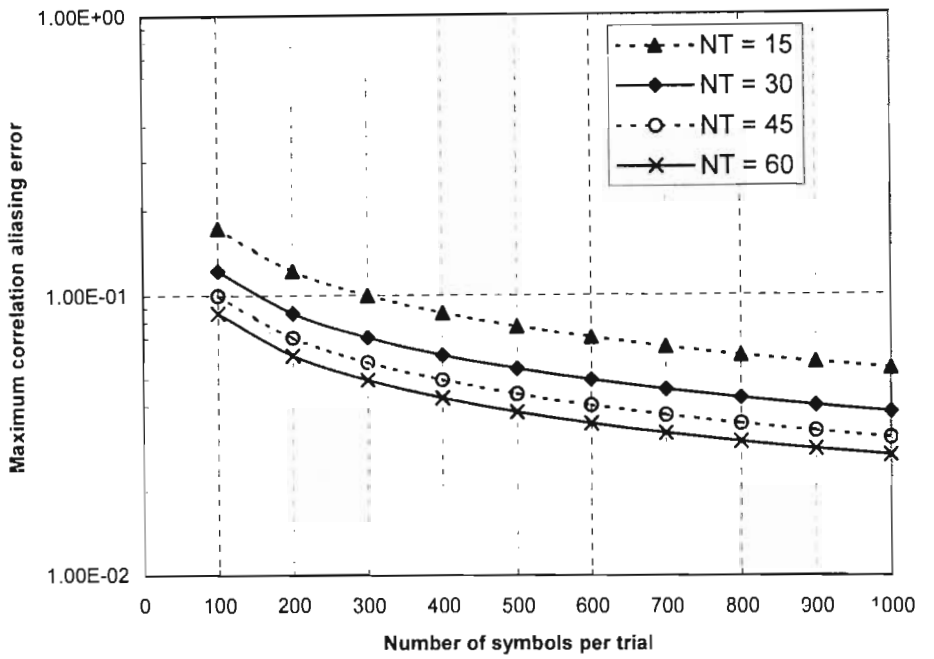


Fig. 4.4 Maximum correlation error due to aliasing for different values of N_T and $fm = 0.005$.

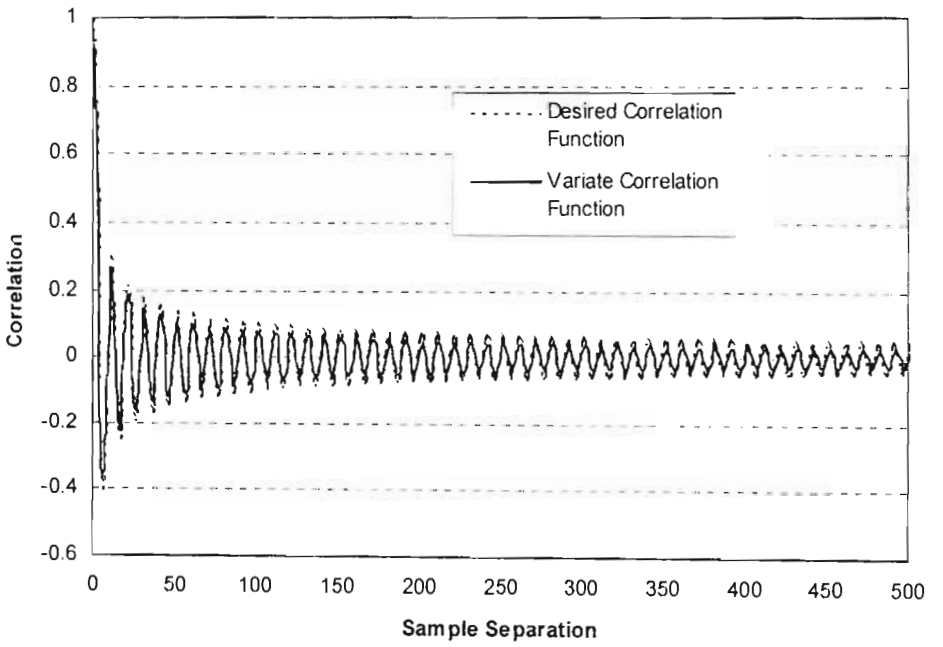


Fig. 4.5: Correlation functions $R_y[d]$ and $\tilde{R}_y[d]$, fm = 0.1, N_T=15, α = 100.

4.4 CONCLUSION

In this chapter, we introduce the models used in this dissertation. The widely used *Jo* statistical model is considered to be the model closely representing the narrowband wireless channel in this dissertation. We also present a FFT method of generating correlated Rayleigh random variates. This technique will be used in Chapters 5 and 6 respectively. As pointed out in the introduction, analytical models are often used to generate solution for specific instance of statistical model. A common analytical model is the Markov chain. In Chapter 6, we will use a first order Markov chain to model a slowly varying Rayleigh fading channel and derive analytical results of the performance of Turbo-coded DS-CDMA system. Classifying a model as statistical, analytical, or simulated-based depends on the model's use.

CHAPTER 5

PERFORMANCE OF TURBO-CODED DS-CDMA SYSTEM

In this chapter, we investigate the effect of mobile velocity/Doppler frequency on the performance of a turbo-coded Direct Sequence Code Division Multiple Access (DS-CDMA) system over a multi-path, frequency selective, and Rayleigh fading channel. A single cell scenario is adopted where there is perfect channel side information. The turbo decoder is modified to take into account the presence of the multiuser interference and self-interference. The effect of channel correlation and channel side information are examined.

This chapter is organized as follows. In the next section, we present an introduction. We describe the system model and adopt the notation in Section 5.2. The decoder metric

is derived in Section 5.3. In Section 5.4, the simulation model and the results are presented. Finally, we conclude in Section 5.5.

5.1 INTRODUCTION

Since its introduction in 1993 by Berrou et al. [27], turbo coding has raised great interest in the communication community. This interest is motivated by the fact that they perform, in AWGN, near the Shannon limit and that these codes have an efficient, sub-optimal decoding structure, making them practical. Turbo codes can provide significant coding gain by utilizing two relatively simple constituent convolutional encoders, which are concatenated in parallel through an interleaver. The decoder consists of the corresponding component decoders employing a maximum-a-posteriori (MAP) algorithm and is designed to iteratively exploit extrinsic information. Performance is proportional to the number of decoding iterations and the length of the information packet, which is limited by the length of the interleaver. Furthermore, turbo-codes also offer promising alternatives for fading channels [112, 113]. Therefore, their performance in different environments is worth investigation.

Code Division Multiple Access (CDMA) is becoming increasingly popular for PCS applications, since it promises spectrum efficiency, economical and high quality services for the third generation cellular systems [80].

5.2 SYSTEM MODEL

The system model is shown in Fig. 5.1. The source is a random binary data generator. The input data is then encoded by the turbo encoder, which gives out one systematic bit and two parity bits. The transmitter consists of three paths: turbo encoder, BPSK modulator, DS spreader.

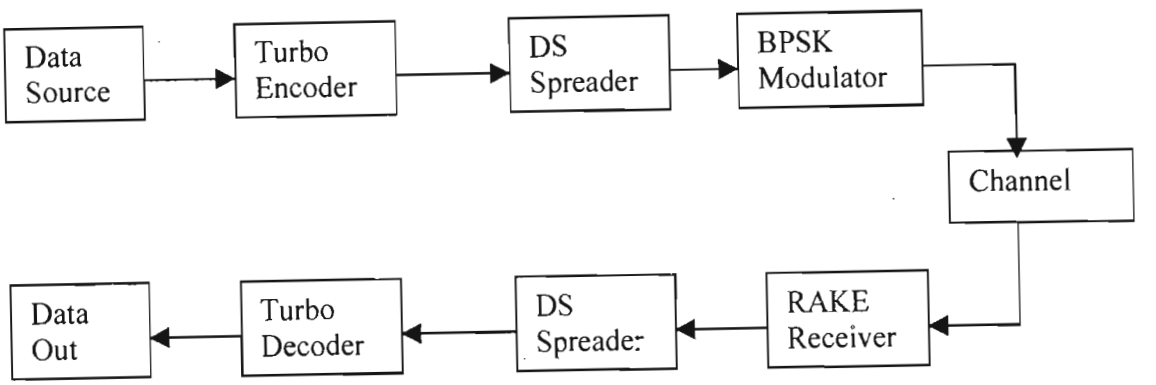


Fig. 5.1: System Model

5.2.1 Transmitter Model

In a DS-CDMA system, we assume that there are K users transmitting data asynchronously. The transmitted signal for the k^{th} user is a phase-coded carrier, which may be written as [114, 115]

$$s^{(k)}(t) = \sqrt{2P}a^{(k)}(t)b^{(k)}(t)\cos[w_c + \phi^{(k)}], \tag{5.1}$$

where $a^{(k)}(t)$ is the code sequence of the k^{th} user and may be expressed as

$$a^{(k)}(t) = \sum_{j=-\infty}^{\infty} a_j^{(k)} p_a(t - jT_c), \; a_j^{(k)} \in \{-1,1\}, \tag{5.2}$$

and $b^{(k)}(t)$ is the data waveform, which can be expressed as

$$b^{(k)}(t) = \sum_{j=-\infty}^{\infty} a_j^{(k)} p_b(t - jT), \; b_j^{(k)} \in \{-1,1\}. \tag{5.3}$$

In (5.1), P is the average transmitted power, which is common to all the users, w_c is the common carrier frequency, and $\phi^{(k)}$ is the phase angle of the k^{th} modulator. The phase angle is assumed to be uniformly distributed in $[0,2\pi]$. In (5.2) and (5.3), T_c is the chip duration, T is the data bit duration and $p_a(t)$ and $p_b(t)$ are rectangular pulses of unit height and duration T_c and T respectively.

5.2.2 Channel Model

The link between the k^{th} user and the base station is characterized by a low pass equivalent transfer function given by

$$h_k(\tau) = \sum_{l=0}^{L_k} \beta_l^{(k)} e^{j\theta_l^k} \delta(\tau - \tau_l^k). \tag{5.4}$$

The l^{th} path of the k^{th} user can be characterized by three random variables: the gain $\beta_l^{(k)}$, the delay τ_l^k and the phase θ_l^k. We assume that $\beta_l^{(k)}$ and τ_l^k are statistically independent for different values of k and l. The gain $\beta_l^{(k)}$ is Rayleigh distributed, θ_l^k is uniformly distributed in $[0,2\pi]$ and τ_l^k is uniformly distributed in [0,T].

5.2.3 Receiver Model

The receiver model is presented in Fig. 5.2. We assume that BPSK modulated signals are detected coherently. The received signal of the k^{th} user, for the general case of multiple asynchronous users that occupy the mobile channel, in a single-cell system, can be written as [114, 115]:

$$r(t) = n(t) + \sqrt{2P} \sum_{k=1}^{K} \sum_{l=0}^{L_k - 1} \beta_l^k a^{(k)}\left[t - \tau_l^{(k)}\right] b^{(k)}\left[t - \tau_l^{(k)}\right] \cos\left[w_c t + \varphi_l^k\right], \tag{5.5}$$

where L_k is the number of multipath of the k^{th} user, $\varphi_l^{(k)} = \phi^{(k)} + \theta_l^{(k)} - w_c \tau_l^{(k)}$ is the phase of the lth path of the kth carrier, and $n(t)$ is AWGN with two sided power spectral density $N_o/2$. It is assumed that there are $L^{(k)}$ paths present at the kth receiver, each corresponding to a different propagation path between transmitter and receiver.

Assuming that acquisition has been accomplished for the user of interested (k=1) and that the matched filter is synchronized to the first path of the desired signal, then the output of the correlation receiver at each sampling time can be written as [114, 115].

$$U = \sum_{n=0}^{I_{cr}} \int_{nT_c}^{T+nT_c} r(t) \beta_n^1 a^{(1)}(t - nT_c) \cos\left[w_c t + \varphi_n^1\right] dt$$

$$= \sum_{n=0}^{L_r-1} \left\{ S^{(n)} + I_{mai}^{(n)} + I_{si}^{(n)} + I_{ni}^{(n)} \right\}, \tag{5.6}$$

where

$$S^{(n)} = \sqrt{\frac{P}{2}} b_0^{(1)} T \left\{ \beta_n^{(1)} \right\}^2 , \tag{5.7a}$$

$$I_{mai}^{(n)} = \sqrt{\frac{P}{2}} \sum_{k=2}^{K} \sum_{l=0}^{L^{(k)}} \beta_n^{(1)} \beta_l^{(k)} \left\{ b_{-1}^{(k)} R_{k1}\left[\tau_{nl}^{(k)}\right] + b_0^{(k)} \hat{R}\left[\tau_{nl}^{(k)}\right] \right\} \cos\left[\varphi_{nl}^{k}\right] , \tag{5.7b}$$

$$I_{si}^{(n)} = \sqrt{\frac{P}{2}} \sum_{l=0, l\neq n}^{L^{(1)}-1} \beta_n^{(1)} \beta_l^{(1)} \left\{ b_{-1}^{(1)} R_{11}\left[\tau_{nl}^{(1)}\right] + b_0^{(1)} \hat{R}\left[\tau_{nl}^{(1)}\right] \right\} \cos\left[\varphi_{nl}^{(1)}\right] , \tag{5.7c}$$

$$I_{ni}^{(n)} = \int_{nT_c}^{T+nT_c} n(t) \beta_n^{(1)} a^{(1)}(t - nT_c) \cos\left[w_c t + \varphi_n^{[1]}\right] dt , \tag{5.7d}$$

with $b_0^{(1)}$ being the information bit to be detected, and $b_{-1}^{(1)}$ the preceding bit.

$\tau_{nl}^{(k)} = \tau_l^{(k)} - \tau_n^{(1)}$, $\varphi_{nl}^{k} = \varphi_l^{k} - \varphi_n^{1}$, k =1,2,.....K, and

$$R_{k1}(\tau) = \int_0^{\tau} a^{(k)}(t - \tau) a^{(1)}(t) dt , \tag{5.8a}$$

$$\hat{R}_{k1}(\tau) = \int_{\tau}^{T} a^{(k)}(t - \tau) a^{(1)}(t) dt . \tag{5.8b}$$

$R_{k1}(\tau)$ and $\hat{R}_{k1}(\tau)$ are partial correlation functions.

Thus, from (5.6), we see that the output of the nth branch, n =0,1...L_r-1, consists of four terms. The first term represents the desired signal component to be detected. The second term represents the multiple-access interference (MAI) from the (K-1) other simultaneous users in the system. The third term is the self-interference (SI) for the reference user due to side lobes of the autocorrelation function of the spread-spectrum code that is assigned. Finally, the last term in (5.6) is the Gaussian random variable due to the AWGN process.

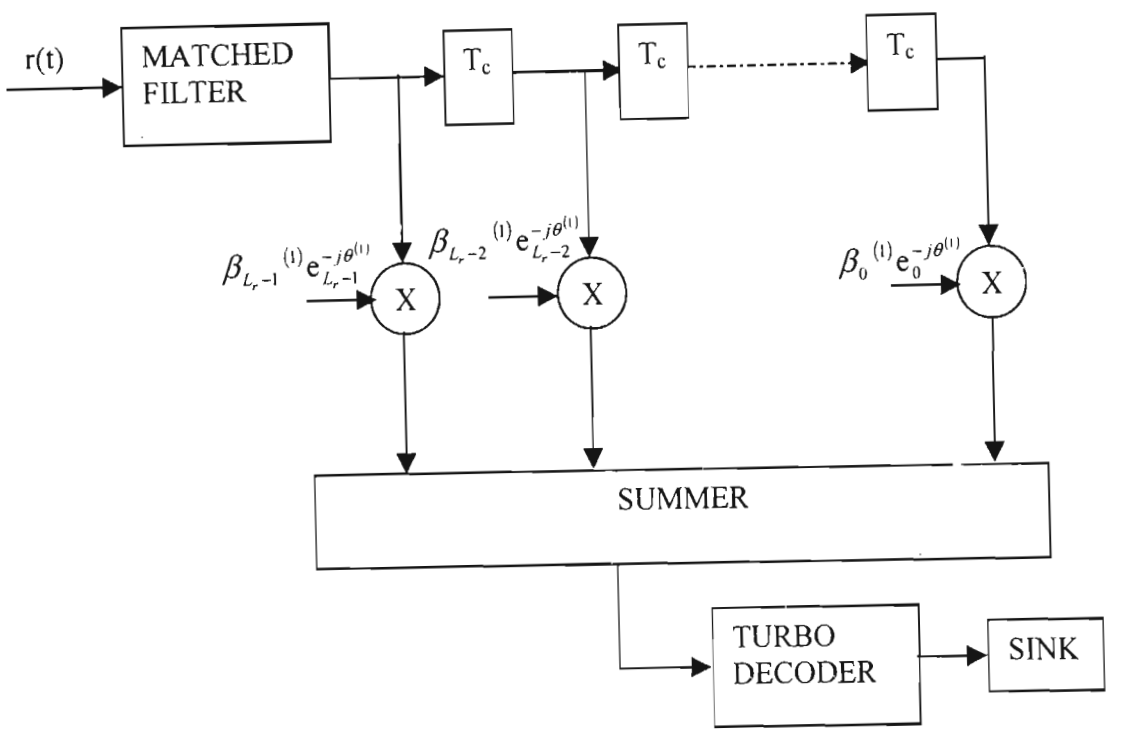


Fig. 5.2: RAKE Receiver Model

5.3 TURBO DECODER METRIC

For multiuser Rayleigh fading channels, the turbo-decoder must be modified to take into account the appropriate channel statistics. In the MAP algorithm, this corresponds to formulating the transition metric γ. This results in a new expression for the channel reliability factor L_c.

5.3.1 Gaussian Assumption

In the performance analysis of DS-CDMA systems, the interference terms are usually assumed to be Gaussian-distributed. The use of Gaussian assumption is very common, since it is found to be quite accurate, even for small values of K (<10) when the BER is 10^{-3} or greater [114]. A number of approaches are available to improve the accuracy of

the Gaussian approximation [116]. In the sequel, we employ the Gaussian approximation and model the MAI and the SI terms as AWGN processes with equal variance equal to the MAI and SI variances, respectively. It is well known that, conditioned on $\beta_n^{(1)}$, the variance of MAI in the nth branch, $\sigma_{mai,n}^2$, is given by [114], [117]

$$\sigma_{mai,n}^2 = \frac{P}{2}\sum_{k=2}^{K}\sum_{l=0}^{L^{(k)}-1}\left\{\beta_n^{(1)}\right\}^2 E\left[\left\{\beta_l^{(k)}\right\}^2\right].E\left[\left\{b_{-1}^{(k)}R_{k1}\left(\tau_{nl}^{(k)}\right)+b_0^{(k)}\hat{R}_{k1}\left(\tau_{nl}^{(k)}\right)\right\}^2\right].E\left[\cos^2\left(\varphi_{nl}^{(k)}\right)\right]$$

$$= \frac{P}{4}\left\{\beta_n^{(1)}\right\}^2\sum_{k=2}^{K}\frac{T^2}{3N^3}r_{k1}(N)\sum_{l=0}^{L^{(k)}-1}.E\left[\left\{\beta_l^{(k)}\right\}^2\right]. \tag{5.9}$$

where $r_{k1}(N)$ is the average cross-correlation parameter over all the possible $\{K(K-1)/2\}$ combinations of sequences k and one among the set, defined in [117]. In order to bypass the problem of using specific codes, we assume that the period of the CDMA signature sequences is much larger than the processing gain, so that the sequences can be modeled as random binary sequences. For random signature sequences, $r_{k1} \approx 2N^2$ and (5.9) becomes [117]

$$\sigma_{mai,n}^2 = \frac{E_b T}{6N}\left\{\beta_n^{(1)}\right\}^2\sum_{k=2}^{K}\sum_{l=0}^{L^{(k)}-1}\Omega_l^{(k)}, \tag{5.10}$$

where $\Omega_l^{(K)} = E\left[\left\{\beta_l^{(k)}\right\}^2\right]$. Similarly, the variance of $I_{si}^{(n)}$, conditioned on $\beta_n^{(1)}$, is well approximated by [114]

$$\sigma_{si,n}^2 \approx \frac{E_b T}{4N}\left\{\beta_n^{(1)}\right\}^2\sum_{l=1}^{I_c^{(1)}-1}\Omega_l^{(1)}, \tag{5.11}$$

where $E_b = PT$ is the signal energy per bit. From (5.7d), the variance of the Gaussian noise term, conditioned on $\beta_n^{(1)}$, is given as

$$\sigma_{n,n}^2 \approx \frac{T\eta_0}{4}\left\{\beta_n^{(1)}\right\}^2. \tag{5.12}$$

Also, conditioned on $\beta_n^{(1)}$, $n = 0,1,\ldots,L_r-1$, the output of the combining receiver U is a Gaussian random process, with mean

$$U_s = \sqrt{\frac{E_b T}{2}}\sum_{n=0}^{L_r-1}\left\{\beta_n^{(1)}\right\}^2, \tag{5.13}$$

and variance equal to the sum of the variances of all the interference terms. Therefore, from (5.10), (5.11) and (5.12), we have

$$\sigma_T^2 = \sum_{n=0}^{L_r-1}\left(\sigma_{mai,n}^2 + \sigma_{si,n}^2 + \sigma_{ni,n}^2\right)$$

$$= (E_b T)\left\{\frac{\sum_{k=2}^{K}\sum_{l=0}^{L^{(k)}-1}\Omega_l^{(k)}}{6N} + \frac{\sum_{l=1}^{L^{(1)}-1}\Omega_l^{(1)}}{4N} + \frac{No}{4E_b}\right\}\cdot\sum_{n=0}^{L_r-1}\left\{\beta_n^{(1)}\right\}^2, \qquad (5.14)$$

where $\Omega_l^k = \mathrm{E}\left[\left\{\beta_l^{(k)}\right\}^2\right]$, N is the processing gain of the spread-spectrum system and $E_b=PT$ is the signal energy per bit. Assuming that all the users have the same average signal power at the receiver as a result of perfect power-control strategy, we set $\Omega_l^{(k)} = \Omega_l$ and $L^{(k)}=L$ for all users. A multipath fading channel is sometimes characterized by its multipath intensity profile (MIP), which is the average power at the output of the channel as a function of path delay [42]. Actual measurements made by Turin [118] in an urban environment indicate that the MIP is exponential. However, despite this indication, both the constant [119] and Gaussian [120] intensity profiles have also been used in the literature. Here, we assume both the exponential MIP distribution and the constant distribution. For an exponential MIP distribution,

$$\Omega_l = \Omega_0 \exp(-l\delta), \qquad \delta \geq 0 \qquad (5.15)$$

where Ω_0 is the average signal strength corresponding to the first incoming path and δ is the rate of average power decay. A constant distribution corresponds to $\delta = 0$.

5.3.2 Derivation of the Metric

Applying the above Gaussian assumption, equation (5.6) can be rewritten as

$$U = \sum_{n=0}^{L_r-1}\sqrt{\frac{P}{2}}b_0^{(1)}T\left\{\beta_n^{(1)}\right\}^2 + n\ , \qquad (5.16)$$

where n is a Gaussian random variable with zero mean and variance σ_T^2.

We normalized U by $\sqrt{\frac{P}{2}}T$ to obtain

$$y = \frac{U}{\sqrt{\frac{P}{2}}T} = b_0^{(1)} \sum_{n=0}^{L-1} \left\{ \beta_n^{(1)} \right\}^2 + n_1 , \tag{5.17}$$

where n_1 is a Gaussian random variable with zero mean and variance

$$\sigma_1^2 = \sigma_T^2 \Big/ \left(\sqrt{\frac{P}{2}}T \right)^2 = \frac{2\sigma_T^2}{(E_b T)} . \tag{5.18}$$

5.3.2.1 Original Turbo Decoder

Because the optimal decoder is too complex, the turbo decoder provides a suboptimal alternative that iteratively passes log-likelihood information between a pair of MAP decoders matched to each of the component encoders. The turbo decoding algorithm has been well documented in Chapter 2 and Appendix A, thus it will not be repeated here. Of particular interest, however, are the branch transition probabilities which are needed for turbo decoder calculations. The computation of branch transition probabilities depends on the channel, so they play a key role in the design of the turbo decoder for our system model. Let s_k be the state of the first encoder at time k. The branch transition probabilities used by the MAP algorithm are calculated as

$$\gamma_k(s', s) = p\left(s_k = s, y_k^s, y_k^p / s_{k-1} = s'\right), \tag{5.19}$$

where we have defined y_k^s and y_k^p as the received decision variables originating from the systematic and parity symbols respectively. The above equation can be rewritten as

$$\begin{aligned} \gamma_k(s', s) &= Pr(s/s') p\left(y_k^s, y_k^p / s', s\right) \\ &= Pr(u_k) p\left(y_k^s / u_k\right) p\left(y_k^p / u_k\right), \end{aligned} \tag{5.20}$$

where u_k corresponds to the event $s' \rightarrow s$. It can be shown that

$$Pr(u_k) = A_k \exp\left[u_k L^e(u_k)/2\right], \tag{5.21}$$

where A_k is a constant not depending on u_k and

$$L^e(u_k) \equiv \log\left(\frac{Pr(u_k = +1)}{Pr(u_k = -1)} \right) \tag{5.22}$$

is the extrinsic information.

The terms (y_k^s, y_k^p) and u_k are in the y's and $b_0^{(1)}$'s of equation (5.17) respectively.

5.3.2.2 Modification to the Turbo Decoder

The turbo decoding algorithm is dependent on what information is available to the turbo decoder. We will assume that the fading amplitudes are perfectly known to the decoder. If we let

$$a = \sum_{n=0}^{L-1} \{\beta_n^{(1)}\}^2 , \tag{5.23}$$

then $p(y_k^i / u_k, a)$ has density $N(au_k, \sigma_1^2)$ where $i \in \{s, p\}$. Thus (5.20) can be computed using

$$p(y_k^i / u_k, a) = \frac{1}{\sqrt{2\pi\sigma_1^2}} \exp\left[-\frac{1}{2\sigma_1^2} (y_k^i - au_k)^2 \right] . \tag{5.24}$$

Following reasoning similar to the one in Chapter 2 and Appendix A, the channel reliabiiity factor is obtained as

$$L_c = \left[\frac{(K-1)\sum_{l=0}^{L-1} \Omega_l}{6N} + \frac{\sum_{l=1}^{L-1} \Omega_l}{4N} + \frac{N_0}{4E_b} \right]^{-1} . \tag{5.25}$$

5.4 SIMULATION MODEL AND RESULTS

For all simulations, the components encoders are rate 1/2 recursive systematic convolutional encoders with memory 3 and octal generators (13, 15). The frame size is 1024 bits. Gold spreading sequences of length N = 63 are assumed. The channel is a six path independent Rayleigh fading channel having a delay equal to the symbol duration. The number of RAKE branches is equal to the number of propagation paths that arrive at the receiver unless otherwise specified. We assume perfect channel side information. This means that the instantaneous amplitudes for the fading channel are known to the

turbo decoder. The paths strength and phases are assumed constant during any bit interval. The number of users is 10. In order to represent a wide range of mobile communication environments, the product $f_D T_s$ was considered as an independent parameter f_m and simulations were performed for f_m = 0.001, 0.01 and 0.1. These values encompass the whole range of mobile communication environments ranging from cellular telephony to PCS. The MAP algorithm and a total of 8 decoding iteration were used.

Figs. 5.3 and 5.4 show the effect of the velocity/Doppler frequency on the performance of the system when $\delta = 0.2$ (exponential MIP) and $\delta = 0$ (constant MIP) respectively. In both cases, the performance of the system improves as the mobile velocity/Doppler frequency increases. Looking at the bit error rate of 10^{-4} we see that a performance gain of about 6 dB is obtained when increasing f_m from 0.001 to 0.01 and that a further improvement of about 2.5 dB is obtained when f_m is increased to 0.1. These gains are the same for both MIP. It can be seen, however, that the system performs better when a constant MIP is assumed. Note that, at a constant carrier frequency and data rate, increasing f_m is equivalent to increasing the speed of the mobile user. These results make intuitive sense because when the signal experiences a fast fading channel, successive symbols in a codeword will suffer from nearly uncorrelated fading. Hence the turbo decoder can correct intermittent symbols errors due to the channel. However, as the speed or the fading rate decreases, the channel becomes more correlated. If a deep fade corrupts the channel, the fade may leads to a burst of errors that surpasses the error correcting capability of the turbo decoder.

Figs. 5.5 and 5.6 show the effect of the number of RAKE branches, L_r, on the performance of the system when $\delta = 0.2$ (exponential MIP) and $\delta = 0$ (constant MIP) respectively. As expected, the performance of the system improves as the number of RAKE branches increases. It is well known that the performance of the RAKE is optimum when all the paths are resolved and combined at the receiver, i.e. when $L_r = L$.

We however notice that, contrary to the previous observation, when $L_r < L$, the system performs better when $\delta = 0.2$ than when $\delta = 0$. This is probably due to the fact that the reduced multiple-access interference noise has more impact on the RAKE performance than the reduced signal power. The reverse applies when $L_r = L = 6$, as the sum of the signal powers of the different paths affects the RAKE performance more favorably than the increase in the interference noise.

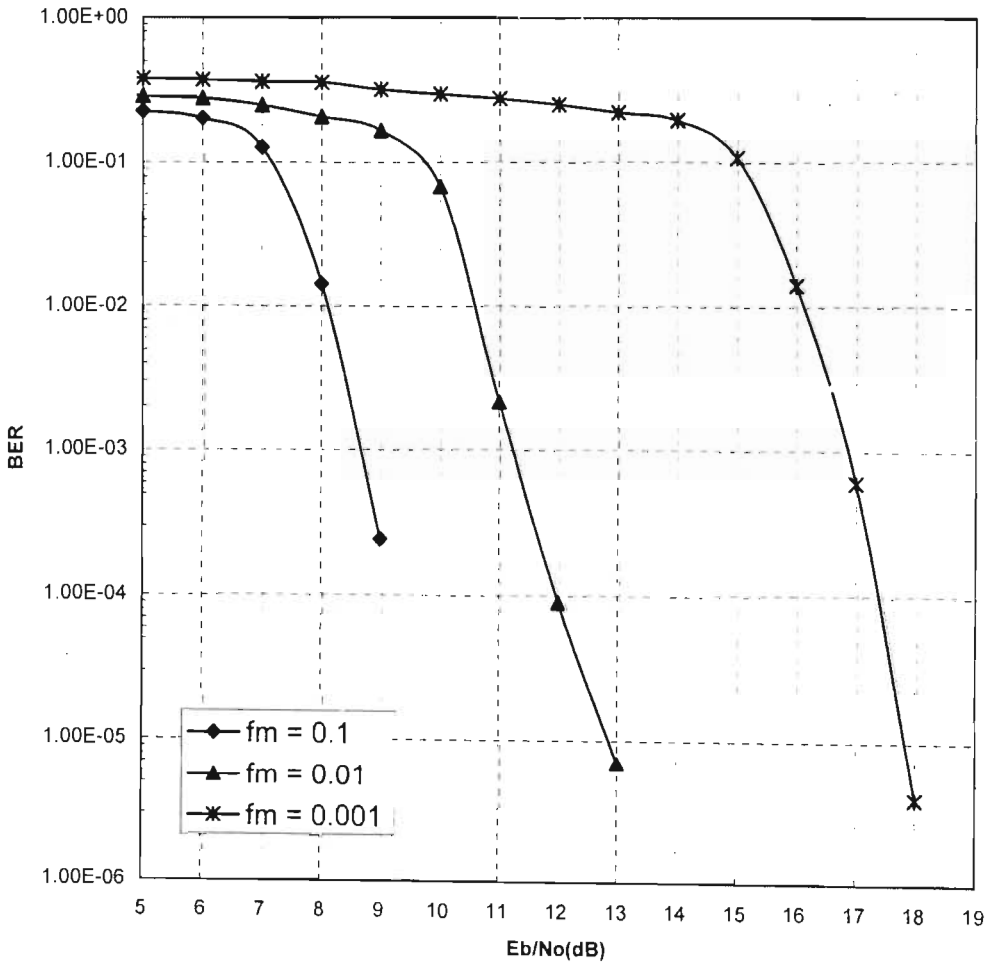


Fig. 5.3: Simulated bit error rate performance for different values of *fm* and $\delta = 0.2$.

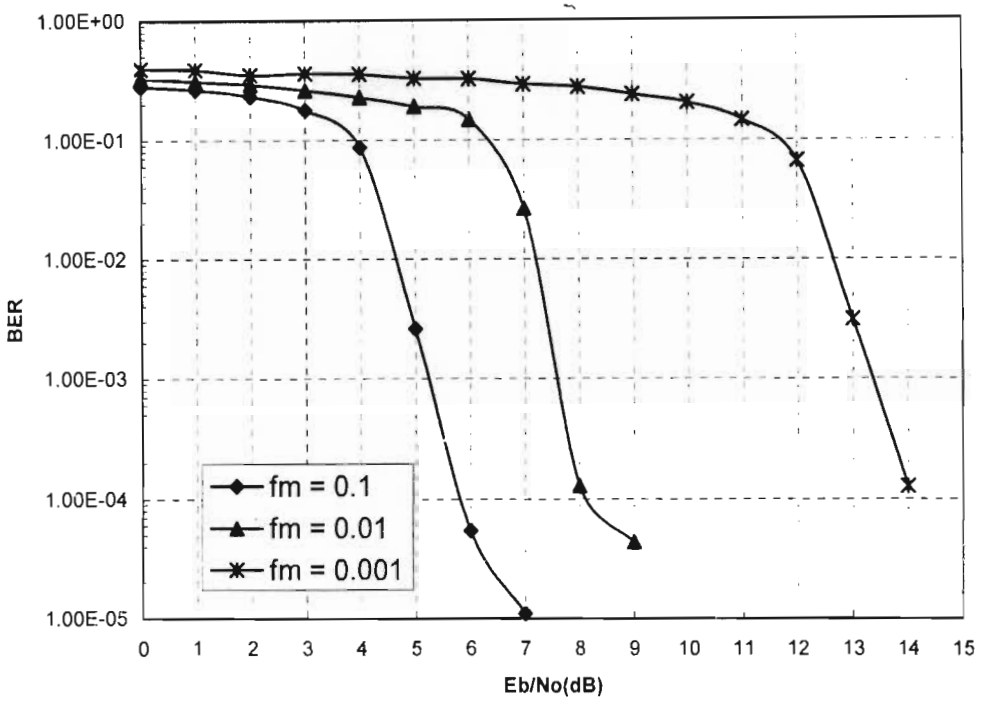


Fig. 5.4: Simulated bit error rate performance for different values of *fm* and $\delta = 0$.

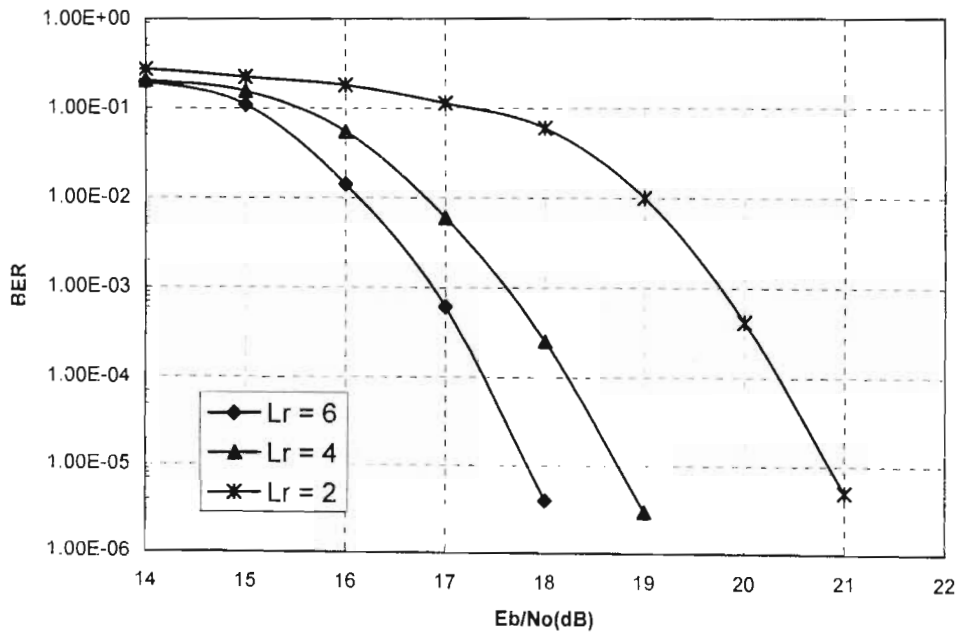


Fig. 5.5: Simulated bit error rate performance for different number of RAKE branches L_r, $f_m = 0.001$ and $\delta = 0.2$.

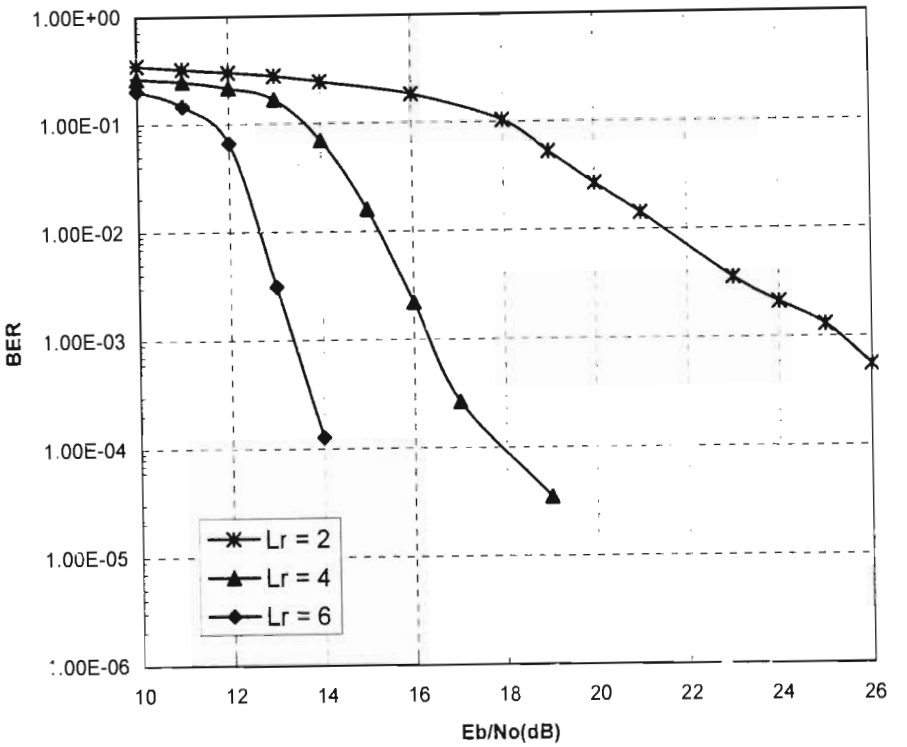


Fig. 5.6: Simulated bit error rate performance for different number of RAKE branches L_r, $f_m = 0.001$ and $\delta = 0$.

5.5 CONCLUSION

In this chapter, we investigate the effect of mobile velocity/Doppler frequency on the performance of a turbo-coded direct sequence code division multiple access (DS-CDMA). Applying the Gaussian assumption for the MAI and SI terms, we derive an expression for the turbo decoding metric. This results in a new expression for the channel reliability factor. It has been shown that the correlation of the channel has a large influence on the performance of the system. It was observed that when all the paths are resolved and combined at the receiver, i.e. when the number of RAKE branches is equal to the number of paths ($L_r = L$), the system performs better when a constant MIP is assumed. The reverse was observed when $L_r < L$.

CHAPTER 6

PERFORMANCE OF TURBO-CODED DS-CDMA SYSTEM ON A GILBERT-ELLIOTT CHANNEL

In this chapter, we consider the design and performance of a turbo-coded DS-CDMA system in a slowly varying fading channel and an equivalent Gilbert-Elliott model. Analytical expressions for the performance bound of the system are derived and the influence of various parameters such as the Doppler frequency, the signal-to-noise ratio threshold on the system performance, is analyzed and investigated. Comparisons between the results obtained by this analytical expression and results obtained through computer simulations show that that the analytical results are accurate for a broad range of channel parameters. Furthermore we show that a combination of a small code interleaver with a channel interleaver could outperform codes with very large interleavers, making turbo coding suitable for even delay-sensitive services.

The chapter is organized as follows. We first present a brief introduction to motivate the problem. We then present our system model. The Gilbert-Elliott channel model is described in Section 3; some known properties of the channel are recapitulated and useful statistical properties of the channel are derived. The channel parameters are matched to the slowly varying Rayleigh fading channel in Section 4, and the effect of channel interleaving on the Gilbert-Elliott channel model is assessed in Section 5. Analytical expressions for the performance bounds are derived in Section 6. Section 7 considers the modification to the turbo decoder necessary for the Gilbert-Elliott channel model. The simulation model and results are presented in Section 8 and conclusions are drawn in Section 9.

6.1 INTRODUCTION

In the previous chapter, we considered the performance of turbo-coded DS-CDMA system over multi-path Rayleigh fading channel. A common feature of fading communication channels is that they all have memory. Studies of the performance of error correcting codes are most often concerned with situations where the channel is assumed to be memoryless, since this allows for a theoretical analysis. In situation where memory is accounted for, on the other hand, the analytical results are few, and studies of the performance are therefore often obtained through simulations.

To deal with complicated channel model, it is sometimes possible to use a less complex one that still reflects the essential (for that particular study) properties of the complicated model. For a channel with memory, the Gilbert-Elliott (GE) channel [121, 122] is a useful and one of the simplest discrete models that has been studied in considerable detail in the literature. In this model, for a slowly varying channel, the channel is assumed to either be in a good state, where the probability of errors is small, or in a bad state, where the probability of error is significantly larger. The dynamics of the channel are modeled as a first- order Markov chain, a model which Wang and Moayeri [123] and Wang and Chang [124], in spite of its simplicity, showed to be very accurate for a Rayleigh fading

channel. The model parameters of the GE channel have been related to the fade statistics by several researchers [125, 127].

The GE channel has been used by numerous authors to evaluate the performance of random-error-correcting codes (i.e. [128, 129]) and burst-error-correcting codes [130]. In [131], the authors consider the design and performances of turbo codes for a Gilbert-Elliott burst channel and in a realistic fading channel and propose the necessary modifications to the turbo decoder. Three different decoding structures are also proposed in [132]. No analytical work is done in these papers.

6.2 SYSTEM MODEL

We consider the model of the asynchronous binary PSK direct-sequence CDMA system that allows K users to share a channel. For the sake of simplicity, we only discuss the case of a single resolvable path, but our method can be extended to a multipath scenario. The actual signal received at a given receiver in this model is given by (see [114][117][133, 134] for the details on the DS/SSMA and DS/CDMA systems):

$$r(t) = \sqrt{2P}\sum_{k=1}^{K}\beta_k(t)a_k(t-\tau_k)b_k(t-\tau_k)\cos(\omega_c t+\varphi_k)+n(t)\ , \qquad (6.1)$$

where n(t) is the Gaussian noise of the double-sided power spectral density $N_o/2$ and P is the power of the transmitted signal, which is common to all users. β_k is the fading amplitude experienced by the signal of the mobile k which is complex Gaussian random variable with zero mean and unit variance. $a_k(t)$ is the spreading waveform of the mobile k and may be expressed as

$$a_k(t) = \sum_{j=-\infty}^{\infty} a_{k,j}\,p_a(t-jT_c)\ ,\ a_{k,j}\in\{-1,1\}. \qquad (6.2)$$

$b_k(t)$ is the data waveform, which can be expressed as

$$b_k(t) = \sum_{j=-\infty}^{\infty} b_{k,j}\,p_b(t-jT)\ ,\ b_{k,j}\in\{-1,1\}. \qquad (6.3)$$

In (6.1) ω_c is the carrier frequency, τ_k is the delay which is uniformly distributed over [0,T). The phase angle φ_k is assumed to be uniformly distributed in $[0,2\pi]$.

In (6.2) and (6.3), T_c is the chip duration, T is the data bit duration, and $p_a(t)$ and $p_b(t)$ are rectangular pulses of unit height and duration T_c and T respectively. β_k, τ_k and φ_k are each assumed to be independent to each other with different K.

If the received signal $r(t)$ is the input to a correlation receiver matched to the ith user, the output of the matched filter at each sampling instant is:

$$Z_i = \int_0^T r(t)a_i(t+\tau_k)\cos\omega_c t dt \tag{6.4}$$

$$= \int_{\tau_k}^{\tau_k+T}\left[n(t)+\sqrt{2P}\sum_{k=1}^{K}\beta_k(t)a_k\sum_{j=-\infty}^{\infty}b_{k,j}p_T(t-jT-\tau_k)\cos(\omega_c t+\varphi_k)\right]a_i(t-\tau_k)\cos\omega_c t dt$$

$$= \int_{\tau_k}^{\tau_k+T} n(t)a_i(t-\tau_k)\cos\omega_c t dt + \sqrt{\frac{P}{2}}b_{i,0}T\beta_i +$$

$$\sqrt{\frac{P}{2}}\sum_{\substack{k=1\\k\neq i}}^{K}\beta_k\left[b_{k,-1}\int_0^{\tau_k}a_k(t-\tau_k)a_i(t)dt + b_{k,0}\int_{\tau_k}^{T}a_k(t-\tau_k)a_i(t)dt\right]\cos\varphi_k$$

$$= \sqrt{\frac{P}{2}}b_{i,0}T\beta_i + \sqrt{\frac{P}{2}}\sum_{\substack{k=1\\k\neq i}}^{K}\beta_k\left[b_{k,-1}R_{k,i}(\tau_k)+b_{k,0}\hat{R}_{k,i}(\tau_k)\right]\cos\varphi_k + n'(t)\ . \tag{6.5}$$

We define the continuous-time partial cross-correlation functions

$$R_{k,i}(\tau_k) = \int_0^{\tau_K} a_k(t-\tau_k)a_i(t)dt\ , \tag{6.6}$$

$$\hat{R}_{k,i}(\tau_k) = \int_{\tau_k}^{T} a_k(t-\tau_k)a_i(t)dt\ . \tag{6.7}$$

In (6.5), the first term is the desired (signal) component, the second term is the multiuser interference component and the last component is the AWGN component.

We can write (6.5) as

$$Z_i = \sqrt{\frac{P}{2}}b_{i,0}T\beta_i + Z'\ , \tag{6.8}$$

where

$$Z' = \sqrt{\frac{P}{2}} \sum_{\substack{k=1 \\ k \neq i}}^{K} \beta_k \left[b_{k,-1} R_{k,i}(\tau_k) + b_{k,0} \hat{R}_{k,i}(\tau_k) \right] \cos\varphi_k + n'(t) . \qquad (6.9)$$

In (6.9), $n'(t)$ is a Gaussian random variable with zero mean and variance $N_0T/4$.

Since $E[cos\varphi_k] = 0$, $E\{Z'\} = 0$.

Therefore $var\{Z'\} = \sigma^2_{Z'} = \frac{P}{2} \sum_{\substack{k=1 \\ k \neq i}}^{K} E\{\beta_k^2\} E\left\{ \left[b_{k,-1} R_{k,i}(\tau_k) + b_{k,0} \hat{R}_{k,i}(\tau_k) \right]^2 \right\} E[cos^2(\varphi_k)] + \frac{N_0 T}{4}$

$$= \frac{P}{4} \sum_{\substack{k=1 \\ k \neq i}}^{K} \frac{T^2}{3N^3} r_{kl}(N) E\{\beta_k^2\} + \frac{N_0 T}{4}, \qquad (6.10)$$

where $r_{kl}(N)$ is the average cross-correlation parameter over all possible $\{K(K-1)/2\}$ combinations of sequences K and one among the set. In order to bypass the problem of using specific codes, we assume that the period of the CDMA signature sequence is much larger than the processing gain, so that the sequences can be modeled as random binary sequences. For random signature sequences, $r_{kl} \approx 2N^2$ and (6.10) becomes,

$$\sigma^2_{Z'} = \frac{P}{4} \sum_{\substack{k=1 \\ k \neq i}}^{K} \frac{T^2}{3N^3} 2N^2 E\{\beta_k^2\} + \frac{N_0 T}{4}$$

$$= \frac{PT^2}{6N} \sum_{\substack{k=1 \\ k \neq i}}^{K} E\{\beta_k^2\} + \frac{N_0 T}{4} . \qquad (6.11)$$

The SNR at the output of the receiver is therefore

$$\gamma = \frac{(E\{Z_i/\beta_i\})^2}{\mathrm{var}\{Z_i/\beta_i\}} = \frac{\beta_i^2 \frac{P}{2} T^2}{\frac{PT^2}{6N} \sum_{\substack{k=1 \\ k \neq i}}^{K} E\{\beta_k^2\} + \frac{N_0 . T}{4}}$$

$$= \frac{\beta_i^2}{\frac{1}{3N} \sum_{\substack{k=1 \\ k \neq i}}^{K} E\{\beta_k^2\} + \frac{N_0}{2P.T}} \qquad (6.12)$$

and its expected value is

$$E\{\gamma\} = \bar{\gamma} = \frac{1}{\frac{(K-1)}{3N} + \frac{N_0}{2P.T.E\{\beta_i^2\}}} \qquad (6.13)$$

6-3 THE GILBERT-ELLIOTT CHANNEL MODEL

The GE channel is a first-order, discrete-time, stationary, Markov chain with two states, one good and one bad, appropriately denoted by G and B. In the good state errors occur with low probability P_G while in the bad state they occur with high probability P_B. The probabilities that the channel state changes from G to B and from B to G are denoted by b and g, respectively (see Fig. 6.1). The steady state probabilities of being in states G and B are $\pi_G = \frac{g}{b+g}$ and $\pi_B = \frac{b}{b+g}$, respectively.

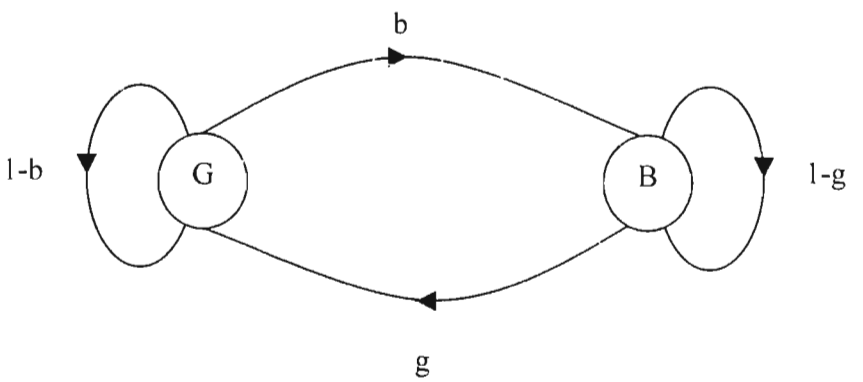


Fig. 6.1: The Gilbert-Elliott Channel Model

Fig. 6.2 is useful in interpreting the significance of the two states. It shows the received SNR, γ, vs. time. If the SNR drops below some pre-determined threshold γ_T, then the channel goes into the bad state else it resides in the good state.

In either state, the channel exhibits the properties of a binary symmetric channel. Figs. 6.3a and 6.3b show the channel bit error characteristics in the two states. In the bad state the probability of error is $P_e(B)$ and in the good state the probability of error is $P_e(G)$. Normaly,

$$P_e(G) << P_e(B).$$

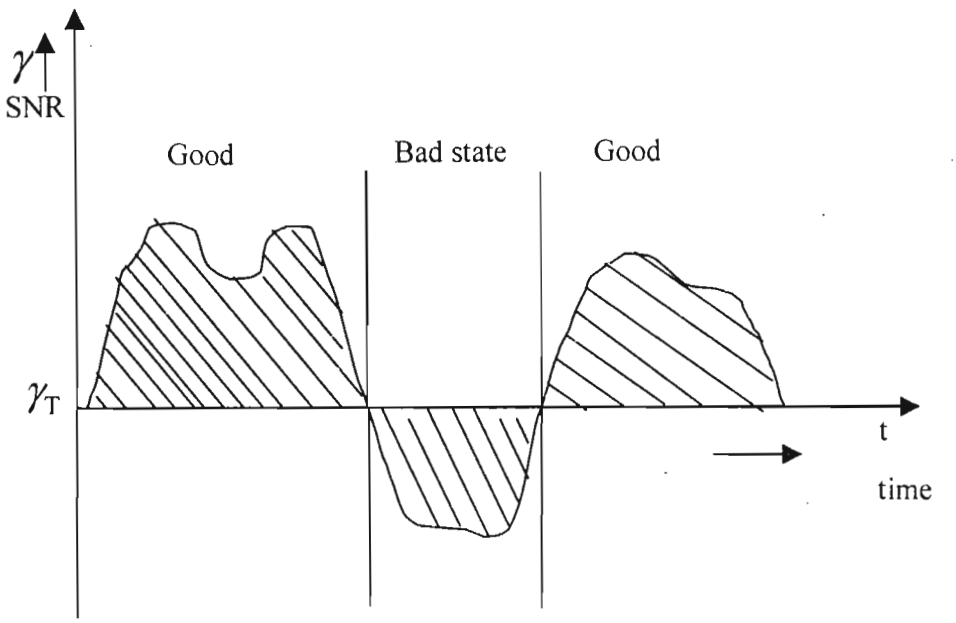


Fig. 6.2: Physical interpretation of the states in the Gilbert-Elliott model.

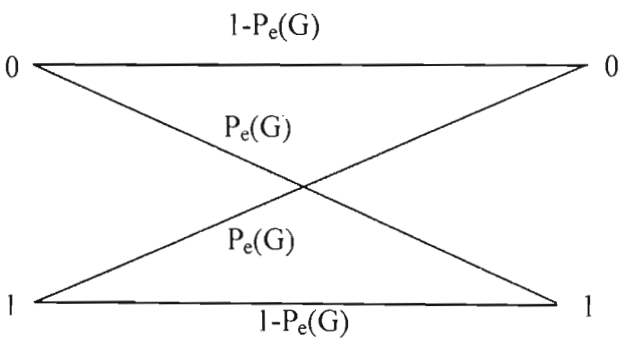


Fig. 6.3a: Good State Binary Symmetric Channel

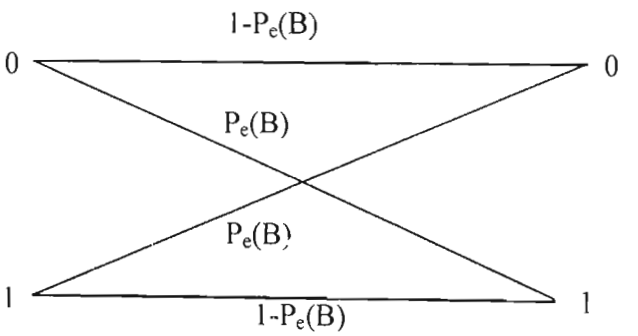


Fig. 6.3b: Bad state Binary Symmetric Channel

At the receiver the detector measures the received SNR and uses it to estimate the state of the channel. This information is passed over to the decoder. The received SNR constitutes the channel state information (CSI) for the decoder.

Let *T(G)* and *T(B)* be the number of time units the channel spends in the good and bad state respectively. Their averages, $\overline{T}(G)$ and $\overline{T}(B)$, can be obtain as follows.

Assume the channel is in the good state and let

$$Y = \begin{cases} 1, \text{if the next transition is from } G \text{ to } B \\ 0, \text{if the next transition is from } G \text{ to } G \end{cases}$$

$$\begin{aligned} \overline{T}(G) = E[T(G)] &= E[E[T(G)/Y]] \\ &= E[T(G)/Y=1]P\{Y=1\} + E[T(G)/Y=0]P\{Y=0\} \\ &= bE[T(G)/Y=1] + (1-b)E[T(G)/Y=0] \\ &= b + (1-b)(1+E[T(G)]). \end{aligned}$$

Therefore

$$\overline{T}(G) = E[T(G)] = \frac{1}{b}. \tag{6.14}$$

Likewise, assume the channel is in the bad state and let

$$X = \begin{cases} 1, \text{if the next transition is from } B \text{ to } G \\ 0, \text{if the next transition is from } B \text{ to } B \end{cases}$$

$$\begin{aligned} \overline{T}(B) = E[T(B)] &= E[E[T(B)/X]] \\ &= E[T(B)/X=1]P\{X=1\} + E[T(B)/X=0]P\{X=0\} \\ &= gE[T(B)/X=1] + (1-g)E[T(B)/X=0] \\ &= g + (1-g)(1+E[T(B)]). \end{aligned}$$

Therefore

$$\overline{T}(B) = E[T(B)] = \frac{1}{g}. \tag{6.15}$$

In words *T(G)* and *T(B)* are geometric random variables with parameter b and g respectively.

6-4 MATCHING THE GILBERT-ELLIOTT CHANNEL MODEL TO THE LAND MOBILE CHANNEL

Our aim now is to relate the generative GE model to the analogue Rayleigh fading model. To do this, we need some results concerning the memory of the analogue model. As can be seen in Fig. 6.4, the signal envelope only occasionally experiences very deep fades. Shallow fades are more likely to occur. A quantitative expression of this property is the level crossing rate, *h(.)*, which is defined as the expected rate at which the envelope crosses a specified threshold, γ_T, in the positive direction. If we use the SNR, γ, instead of the signal envelope level, R, we get the following expression for the level crossing rate [90],

$$h(\gamma_T) = f_D \sqrt{2\pi \frac{\gamma_T}{\bar{\gamma}}}\, exp(-\gamma_T/\bar{\gamma}) , \tag{6.16}$$

where γ_T is the specified SNR threshold and $\bar{\gamma}$ is the average SNR of the received signal. The maximum Doppler shift, f_D, is given by

$$f_D = \mathrm{v}.\lambda , \tag{6.17}$$

where v is the velocity of the moving vehicle and λ is the carrier wavelength. Related to the level crossing rate is the expected duration of the fade below the specified threshold, γ_T, which is given by

$$E[\tau_h] = \frac{1}{h(\gamma_T)}\, pr[\gamma \le \gamma_T]. \tag{6.18}$$

In (6.18), $pr[\gamma \le \gamma_T]$ represents the fraction of time the Rayleigh fading channel is below γ_T and is given by

$$pr[\gamma \le \gamma_T] = \int_0^{\gamma_T} f(\gamma)d\gamma , \tag{6.19}$$

where $f(\gamma)$ is the distribution of the received SNR. Note that the Rayleigh fading results in an exponentially distributed multiplicative distortion of the signal. As a result, the probability density function of the signal-to-noise ration (SNR), γ, is given by

$$f(\gamma) = \frac{1}{\bar{\gamma}} \exp(-\gamma/\bar{\gamma}),\ \gamma \ge 0 . \tag{6.20}$$

Substituting (6.20) into (6.19) we obtain

$$pr[\gamma \le \gamma_T] = 1\text{-}\exp(-\rho), \tag{6.21}$$

where $\rho = \gamma_T / \bar{\gamma}$. So the expected fade duration in a Rayleigh fading environment is

$$E[\tau_b] = \frac{e^{\rho} - 1}{f_D \sqrt{2\pi\rho}}. \tag{6.22}$$

In the same way, the average non-fade duration $E[\tau_g]$ above the threshold is given by

$$E[\tau_g] = \frac{1}{h(\gamma_T)} pr[\gamma \ge \gamma_T]$$

$$= \frac{1}{f_D \sqrt{2\pi\rho}}. \tag{6.23}$$

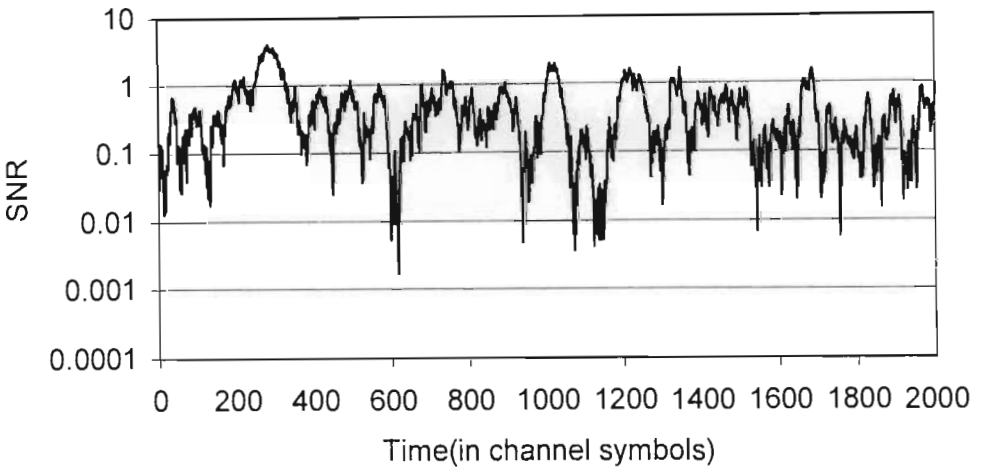


Fig. 6.4: An example of the received signal envelope on a typical simulated Rayleigh fading (fm=0.001).

We want to relate the discrete channel model to the analogue one in such a way that the discrete model should generate approximately the same error distribution as the analogue channel (including the modulator and the demodulator). A natural way to match the two models is to relate the state sequence to the fading signal envelope. Let the bad state represents the situation when the signal envelope is below some threshold and let the good state represents the situation when the signal envelope is above the threshold. We then let the average number of time units the channel spends in the Good (Bad) states be

equal to the expected non-fade (fade) duration, normalized with the symbol time-interval T_S,

$$\overline{T}(G)=E[\tau_g]\frac{1}{T_S}, \tag{6.24}$$

$$\overline{T}(B)=E[\tau_b]\frac{1}{T_S}. \tag{6.25}$$

Hence the transition probabilities are given by

$$b = f_D T_S \sqrt{2\pi\rho}\ , \tag{6.26}$$

$$g = \frac{f_D T_S \sqrt{2\pi\rho}}{e^{\rho}-1}\ . \tag{6.27}$$

The next step is to calculate the error probabilities in each state. They are taken to be the conditional error probabilities of the Rayleigh fading channel, conditioned on being in the respective state, i.e.

$$P_e(B)=\frac{\int_0^{\gamma_T} f(\gamma)P(\gamma)d\gamma}{\int_0^{\gamma_T} f(\gamma)d\gamma}, \tag{6.28a}$$

and

$$P_e(G)=\frac{\int_{\gamma_T}^{\infty} f(\gamma)P(\gamma)d\gamma}{\int_{\gamma_T}^{\infty} f(\gamma)d\gamma}, \tag{6.28b}$$

where $P(\gamma)$ is the symbol error probability for a given value of γ, which depends on the modulation format used. For BPSK, the conditional probability of a code symbol error, conditioned on the received SNR, is given by [42]

$$P(\gamma)=\frac{1}{2}\mathrm{erfc}(\sqrt{\gamma}), \tag{6.29}$$

where $\mathrm{erfc}(x)=\frac{2}{\sqrt{\pi}}\int_x^{\infty}\exp(-t^2)dt$ is the complementary error function.

Substituting (6.20) and (6.29) into (6.28a), the probability of error in the bad state, $P_e(B)$ is

$$P_e(B) = \frac{\int_0^{\gamma_T} \frac{1}{2}\operatorname{erfc}\left(\sqrt{\gamma}\right)\frac{1}{\bar{\gamma}}\exp(-\gamma/\bar{\gamma})\,d\gamma}{\int_0^{\gamma_T} \frac{1}{\bar{\gamma}}\exp(-\gamma/\bar{\gamma})\,d\gamma}$$

$$= \frac{\int_0^{\gamma_T} \frac{1}{2}\left(\int_{\sqrt{\gamma}}^{\infty} \frac{2}{\sqrt{\pi}}\exp\left(-t^2\right)dt\right)\frac{1}{\bar{\gamma}}\exp(-\gamma/\bar{\gamma})\,d\gamma}{\exp(-\gamma/\bar{\gamma})\big|_{\gamma_T}^{0}}$$

$$= \frac{\frac{1}{2}\mathrm{I}}{1-\exp(-\gamma_T/\bar{\gamma})}. \tag{6.30}$$

The integral, I, in the numerator defined as

$$\mathrm{I} = \int_0^{\gamma_T}\left(\int_{\sqrt{\gamma}}^{\infty}\frac{2}{\sqrt{\pi}}\exp\left(-\mathrm{t}^2\right)\mathrm{dt}\right)\frac{1}{\bar{\gamma}}\exp(-\gamma/\bar{\gamma})\,\mathrm{d}\gamma \tag{6.31}$$

can be evaluated by resorting to integration by parts.

We obtain

$$\mathrm{I} = -\operatorname{erfc}\sqrt{\gamma}\,\exp(-\gamma/\bar{\gamma})\big|_0^{\gamma_T} - \int_0^{\gamma_T}\frac{1}{\sqrt{\pi}}\exp\left[-\left(1+\frac{1}{\bar{\gamma}}\right)\gamma\right]\frac{\mathrm{d}\gamma}{\sqrt{\gamma}}$$

$$= 1-\operatorname{erfc}\sqrt{\gamma_T}\,\exp(-\gamma_T/\bar{\gamma}) - \mathrm{I}_1. \tag{6.32}$$

To evaluate I_1 in equation (6.32), we use the substitution

$$\mathrm{u}^2 = \left(1+\frac{1}{\bar{\gamma}}\right)\gamma, \tag{6.33}$$

which gives

$$\mathrm{I}_1 = \int_0^{\gamma_T}\frac{1}{\sqrt{\pi}}\exp\left[-\left(1+\frac{1}{\bar{\gamma}}\right)\gamma\right]\frac{\mathrm{d}\gamma}{\sqrt{\gamma}}$$

$$= \frac{1}{\sqrt{1+\frac{1}{\bar{\gamma}}}}\int_0^{\sqrt{(1+1/\bar{\gamma})\gamma_T}}\frac{2}{\sqrt{\pi}}\exp\left(-\mathrm{u}^2\right)\mathrm{du}$$

$$= \frac{1}{\sqrt{1+\frac{1}{\bar{\gamma}}}}\left[1-\operatorname{erfc}\left(\sqrt{\gamma_T\left(1+\frac{1}{\bar{\gamma}}\right)}\right)\right]. \tag{6.34}$$

Therefore,

$$I = 1 - \text{erfc}\sqrt{\gamma_T}\,\exp(-\gamma_T/\bar{\gamma}) - \frac{1}{\sqrt{1+\frac{1}{\bar{\gamma}}}}\left[1 - \text{erfc}\left(\sqrt{\gamma_T\left(1+\frac{1}{\bar{\gamma}}\right)}\right)\right]. \tag{6.35}$$

Substituting the value of I in equation (6.30), the simplified expression of $P_e(B)$ is

$$P_e(B) = \frac{1 - \text{erfc}\left(\sqrt{\gamma_T}\right)\exp(-\rho) - \frac{1}{\sqrt{1+\frac{1}{\bar{\gamma}}}}\left[1 - \text{erfc}\left(\sqrt{\gamma_T+\rho}\right)\right]}{2[1-\exp(-\rho)]}, \tag{6.36}$$

where $\rho = \gamma_T / \bar{\gamma}$.

The probability of error in the good state, $P_e(G)$, can similarly be expressed as

$$P_e(G) = \frac{\int_{\gamma_T}^{\infty} f(\gamma)P(\gamma)d\gamma}{\int_{\gamma_T}^{\infty} f(\gamma)d\gamma}$$

$$= \frac{\int_{\gamma_T}^{\infty} \frac{1}{2}\left(\int_{\gamma}^{\infty}\frac{2}{\sqrt{\pi}}\exp(-t^2)dt\right)\frac{1}{\bar{\gamma}}\exp(-\gamma/\bar{\gamma})\,d\gamma}{\int_{\gamma_T}^{\infty}\frac{1}{\bar{\gamma}}\exp(-\gamma/\bar{\gamma})\,d\gamma}$$

$$= \frac{\frac{1}{2}I_2}{\exp(-\gamma_T/\bar{\gamma})}, \tag{6.37}$$

where

$$I_2 = \int_{\gamma_T}^{\infty}\left(\int_{\gamma}^{\infty}\frac{2}{\sqrt{\pi}}\exp(-t^2)dt\right)\frac{1}{\bar{\gamma}}\exp(-\gamma/\bar{\gamma})\,d\gamma. \tag{6.38}$$

I_2 can also be integrated by parts as was done for I yielding

$$I_2 = -\text{erfc}\sqrt{\gamma}\,\exp(-\gamma/\bar{\gamma})\Big|_{\gamma_T}^{\infty} - \int_{\gamma_T}^{\infty}\frac{1}{\sqrt{\pi}}\exp\left[-\left(1+\frac{1}{\bar{\gamma}}\right)\gamma\right]\frac{d\gamma}{\sqrt{\gamma}}$$

$$= \text{erfc}\sqrt{\gamma_T}\,\exp(-\gamma_T/\bar{\gamma}) - \frac{1}{\sqrt{1+\frac{1}{\bar{\gamma}}}}\int_{\sqrt{(1+1/\bar{\gamma})\gamma_T}}^{\infty}\frac{2}{\sqrt{\pi}}\exp(-u^2)du$$

$$= \text{erfc}\sqrt{\gamma_T}\,\exp(-\gamma_T/\bar{\gamma}) - \frac{1}{\sqrt{1+\frac{1}{\bar{\gamma}}}}\left[\text{erfc}\left(\sqrt{\gamma_T\left(1+\frac{1}{\bar{\gamma}}\right)}\right)\right]. \tag{6.39}$$

Hence $P_e(G)$ can be expressed in the following form-

$$P_e(G)=\frac{1}{2}\left[\operatorname{erfc}\left(\sqrt{\gamma_T}\right)-\frac{\exp(\rho)}{\sqrt{1+\frac{1}{\bar{\gamma}}}}\operatorname{erfc}\left(\sqrt{\gamma_T+\rho}\right)\right]. \tag{6.40}$$

6.5 THE EFFECT OF INTERLEAVING

Like most of the well-known codes, turbo codes are very effective when the errors caused by the channel are statistically independent, i.e. for random errors. However, there are channels that exhibit bursty error characteristics. One example is the class of channels characterized by multipath and fading, which we described in the last chapter. We saw that correlated fading channels tend to produce burst errors. Such error clusters are not usually corrected by codes that are optimally designed for statistically independent errors.

An effective method to cope with burst errors is to insert an interleaver between the channel encoder and the channel. This process is known as interleaving and may alleviate part of the fading problem caused by the correlation between adjacent received codesymbol energies. It has been widely used in conjunction with error control coding for channel that exhibits bursty error characteristics [42]. Interleaving consists essentially of scrambling the codesymbols in a one-to-one deterministic format before transmission. The inverse of this process is deinterleaving which restores the received sequence to its original order. The effect is to randomize the errors for a particular codeword in order to render the code symbols less dependant, and consequently make them more amenable to correction with a random error-correcting code. A block diagram of a system that employs interleaving is shown in Fig. 6.5. Note that the channel interleaver is different from the nonuniform interleaver required within the turbo encoder. An interleaver or a deinterleaver is characterized by its delay and storage capacity. The interleaving or deinterleaving delay is the maximum delay encountered by any symbol before it is

inserted into the output sequence. The storage capacity is the number of symbols stored by the interleaver or deinterleaver.

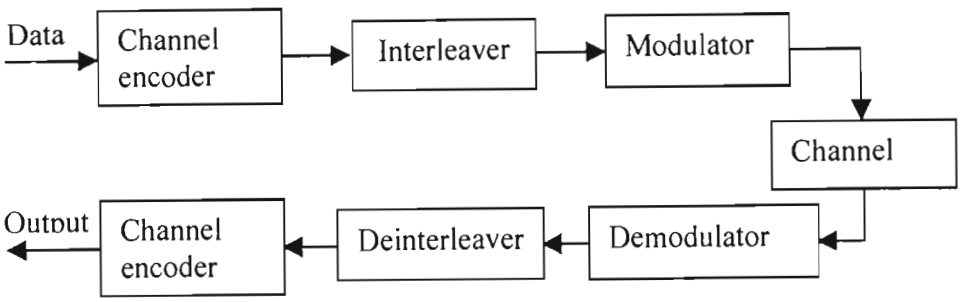


Fig. 6.5: Block diagram of system employing interleaving for burst error channel.

The interleaver can take one of two forms: a block structure or a convolutional structure. Here we assume that block interleaving is used. The encoded symbols are written row by row into a matrix of m rows and n columns. The channel symbols to be transmitted are then read column by column from the matrix and after transmission, the received codesymbols are reordered in the reverse manner. This is illustrated in Fig. 6.6. Thus two adjacent codesymbols are separated by m-1 symbols during transmission.

How effective an interleaver is depends primarily on the extent the interleaved symbols are dependent, which, in turn, depends on how large m, the interleaver depth is chosen. The larger the value of m, the better the interleaver can be expected to work, and if m is infinite, the performance would be the same as for a memoryless channel. In a practical situation, the size of the interleaver will typically be determined by how much delay can be tolerated. We can show that interleaving a code to degree m, has exactly the same effect, from an error performance point of view, as transmitting at a lower rate or increased symbol duration of Tm. In order words, two consecutive symbols of a codeword are spaced apart by m symbols times. This holds true for any stationary slow-fading channel. So, if we know the m-step transition probabilities, that is the probabilities that the channel has changed from one step to the order if observed m moments of time later, it is clear that the original GE channel with an interleaver will be equivalent to a GE

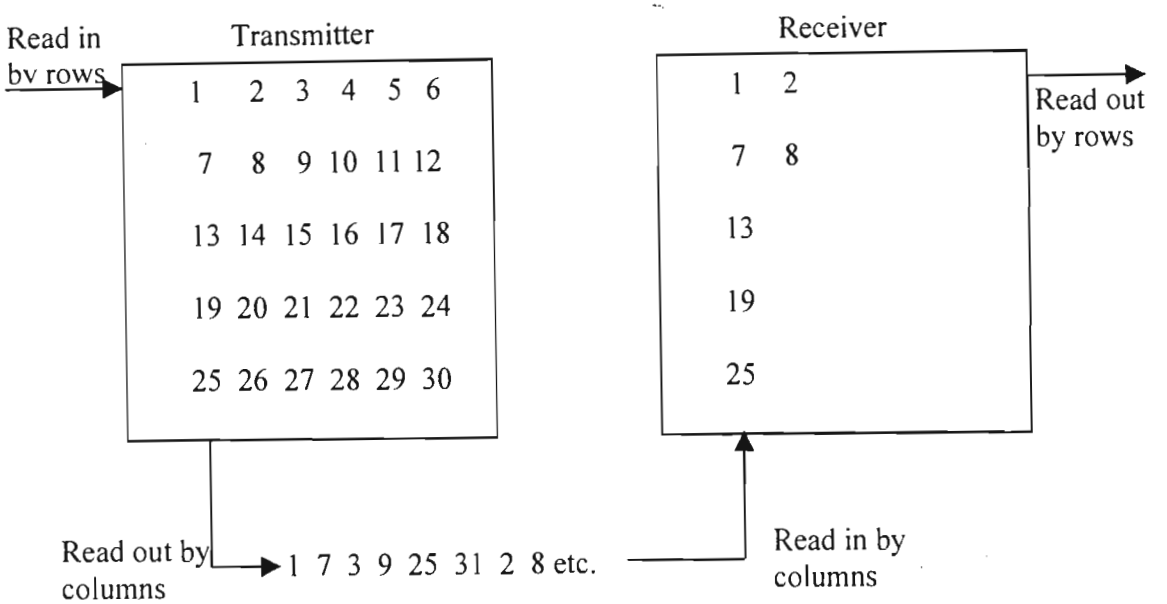


Fig. 6.6: The Structure of a Block Interleaver.

channel where the corresponding transition probabilities are the m-step transition probabilities of the original model. The m-step transition probabilities can be obtain form the Chapman-Kolmogrov equation, which, for our two-state Markov chain, takes on the form [135]

$$P^{(m+1)}(x/x') = P(x/G)P^{(m)}(G/x') + P(x/B)P^{(m)}(B/x') \,, \tag{6.41}$$

and using the fact that $P^{(m)}(B/x')$=1- $P^{(m)}(G/x')$ we obtain a simple recurrence relation

$$P^{(m+1)}(x/x') = [P(x/G) - P(x/B)]P^{(m)}(G/x') + P(x/B), \tag{6.42}$$

where x and x' are states belonging to the state set S= $\{G,B\}$.

By repeated substitution, we then find the general form of the m-step transition probabilities

$$P^{m}(G/G) = \frac{g}{b+g} + \frac{b}{b+g}(1-b-g)^{m}, \tag{6.43}$$

$$P^{m}(G/B) = \frac{g}{b+g} - \frac{g}{b+g}(1-b-g)^{m}, \tag{6.44}$$

and

$$P^{m}(B/G) = \frac{b}{b+g} - \frac{b}{b+g}(1-b-g)^{m}, \tag{6.45}$$

$$P^m(B/B) = \frac{b}{b+g} + \frac{g}{b+g}(1-b-g)^m. \tag{6.46}$$

As expected, if m $\rightarrow \infty$, $P^m(B/G) = \frac{b}{b+g} = \pi_B$ and $P^m(G/B) = \frac{g}{b+g} = \pi_G$ as it would be for a memoryless channel.

6.6 PERFORMANCE ANALYSIS

There are two main tools for the performance evaluation of turbo codes. Monte Carlo simulation and the standard union bound. Monte Carlo simulation generates reliable probability of error estimates as low as 10^{-6} and is useful for rather low signal-to-noise ratios since the error probabilities for larger signal to noise ratio are, typically, too small to simulate. The union bound provides an upper bound on the performance of turbo codes with maximum likelihood decoding averaged over all possible interleavers [46][136]. The premise is that there is at least one interleaver which performs at least as well as the average. Here we apply this bound to the Gilbert-Elliott channel.

6.6.1 Derivation of the Average Upper Bound

The union upper bound is a popular and effective method of bounding block code performance provided that the weight distribution, *A(d)*, is known. For turbo codes, deriving this weight distribution for a particular interleaving scheme is very difficult. Therefore, the authors in [46] and [136], have advanced the idea of forming an average weight function, where the average is over all possible interleaving schemes. In this context, it is useful to view the turbo scheme as the concatenation of multiple "code fragments". From Fig. 6.7, one of the code fragments is the input frame, $\mathbf{u}^s$, while the other fragments are generated by the constituent encoders, $\mathbf{u}^{p1}$ and $\mathbf{u}^{p2}$, and are influenced by the weight of the input frame and how this weight is permuted by the interleavers, I_1 and I_2.

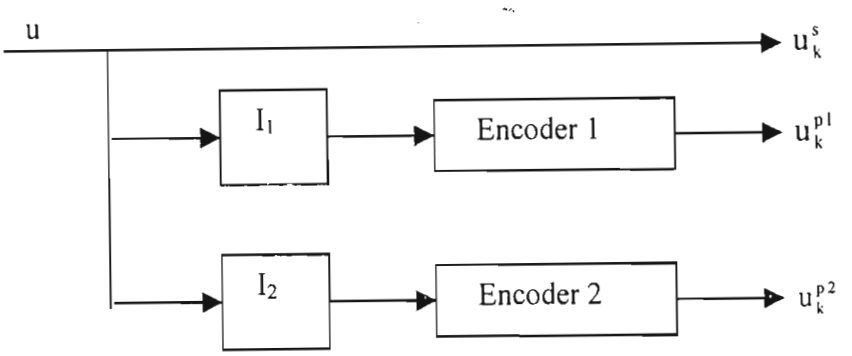


Fig. 6.7: Random interleaving view of turbo codes

Without loss of generality, consider sending the all-zeros codeword. If the interleavers I_1 and I_2 are known, the traditional union upper bound for ML-decoding of an (n, k) block over a memoryless channel is given as:

$$P_{word}(I_1, I_2) \leq \sum_{d=d_{min}}^{n} A(d/I_1, I_2) P_2(d), \tag{6.47}$$

where $P_2(d)$ is the probability of choosing a specific incorrect codeword with weight d and $A(d/I_1, I_2)$ is the number of codewords having Hamming weight d. For a specific interleaver pair (I_1, I_2), the only way to construct $A(d/I_1, I_2)$ is by encoding all valid input sequences and tallying output weight. Therefore, consider forming the word error probability averaged over all interleaver pairs.

$$\begin{aligned}
\overline{P}_{word} &= \sum_{I_1} \sum_{I_2} P(I_1, I_2) P_{word}(I_1, I_2) \\
&= \sum_{I_1} \sum_{I_2} P(I_1, I_2) \sum_{d=d_{min}}^{n} A(d/I_1, I_2) P_2(d) \\
&= \sum_{d=d_{min}}^{n} P_2(d) \sum_{I_1} \sum_{I_2} P(I_1, I_2) A(d/I_1, I_2) \\
&= \sum_{d=d_{min}}^{n} P_2(d) \overline{A}(d).
\end{aligned} \tag{6.48}$$

Here, $P(I_1, I_2)$ is the probability of a specific interleaver and $\overline{A}(d)$ is the average weight distribution function. This average function is formed as the expectation over all

interleaver pairs. If the interleaver pairs are assumed chosen independently and uniformly from the pool of all possible permutations of k elements, the average weight distribution can be rewritten as

$$\overline{A}(d) = \sum_{I_1} \sum_{I_2} P(I_1) P(I_2) A(d/I_1, I_2). \tag{6.49}$$

Furthermore, we can write $A(d/I_1, I_2)$ as

$$A(d/I_1, I_2) = \sum_{i} \binom{k}{i} P(d/i, I_1, I_2), \tag{6.50}$$

where $P(d/i, I_1, I_2)$ is the conditional probability of producing a codeword of weight d from an input of weight i and interleaver pair (I_1, I_2). Furthermore, since turbo codes represent the concatenation of three code fragments, the codeword weight is the sum of the weights of the fragments, $d = i + d_1 + d_2$, where d_1 and d_2 are the Hamming weights of the sequences $\mathbf{u}^{p1}$ and $\mathbf{u}^{p2}$ from Fig. 6.7. Therefore, $A(d/I_1, I_2)$ can be rewritten as

$$A(d/I_1, I_2) = \underbrace{\sum_{i} \sum_{d_1} \sum_{d_2}}_{d=i+d_1+d} \binom{k}{i} P(i, d_1, d_2/i, I_1, I_2). \tag{6.51}$$

Noting the independence of interleaver choices, we can write $P(i, d_1, d_2/i, I_1, I_2)$ as $P(i/i)P(d_1/i, I_1)P(d_2/i, I_2)$ and rewrite the average weight distribution function as,

$$\begin{aligned}
\overline{A}(d) &= \sum_{I_1} \sum_{I_2} P(I_1) P(I_2) \underbrace{\sum_{i} \sum_{d_1} \sum_{d_2}}_{d=i+d_1+d_2} \binom{k}{i} P(i/i) P(d_1/i, I_1) P(d_2/i, I_2) \\
&= \underbrace{\sum_{i} \sum_{d_1} \sum_{d_2}}_{d=i+d_1+d_2} \underbrace{\sum_{I_1} P(I_1) P(d_1/i, I_1)}_{P(d_1/i)} \underbrace{\sum_{I_2} P(I_2) P(d_2/i, I_2)}_{P(d_2/i)} \\
&= \underbrace{\sum_{i} \sum_{d_1} \sum_{d_2}}_{d=i+d_1+d_2} \binom{k}{i} P(d_1/i) P(d_2/i).
\end{aligned} \tag{6.52}$$

Therefore, the complete expressions for the average word and bit error probabilities are given as

$$\overline{P}_{word} \le \sum_{d=d_{\min}}^{n} \overline{A}(d) P_2(d)$$

$$= \sum_{d=d_{\min}}^{n} \underbrace{\sum_{i}\sum_{d_1}\sum_{d_2}}_{d=i+d_1+d_2} \binom{k}{i} P(d_1/i)P(d_2/i)P_2(d),$$

and

$$P_{bit} \le \sum_{d=d_{\min}}^{n} \underbrace{\sum_{i}\sum_{d_1}\sum_{d_2}}_{d=i+d_1+d_2} \frac{i}{k}\binom{k}{i} P(d_1/i)P(d_2/i)P_2(d). \tag{6.53}$$

To calculate the expression of equation (6.53) the distribution of the parity sequences d_1 and d_2 is required. This distribution can be given as [136] (the derivation is described in Appendix B)

$$P(d_p/i) = \frac{t(k,i,d_p)}{\sum_{d_p} t(k,i,d_p)} = \frac{t(k,i,d_p)}{\binom{k}{i}}, \tag{6.54}$$

where $t(l,i,d)$, which can be found from the code's transfer function, is the number of paths of length l, input weight i , and output weight d, starting and ending in the all zero state. With $P(d/i)$, the performance of turbo codes can be studied on various statistical channels by formulating the two-codeword probability $P_2(d)$ for the channel of interest and using (6.53). In the sequel, we derive $P_2(d)$ for the Gilbert-Elliott channel.

6.6.2 $P_2(d)$ for the Gilbert-Elliott Channel

If the channel state is known exactly to the decoder we assume that amongst the d bits in which the wrong path and the correct path differ, there are d_B in the bad state and $d_G = d - d_B$ in the good state. Amongst the bits in the bad states, there are e_B bits in error and amongst the "good" bits e_G are in error.

Let $CM^{(1)}$ and $CM^{(0)}$ be the metric of the wrong path and the correct path respectively.

$$CM^{(1)} = e_B \log(1 - P_e(B)) + (d_B - e_B)\log P_e(B) + e_G \log(1 - P_e(G)) + (d_G - e_G)\log P_e(G) \tag{6.55}$$

$$CM^{(0)} = (d_B - e_B)\log(1 - P_e(B)) + e_B \log P_e(B) + (d_G - e_G)\log(1 - P_e(G)) + e_G \log P_e(G) \tag{6.56}$$

The probability of error in the pairwise comparison of the metrics $CM^{(1)}$ and $CM^{(0)}$ is

$$P_2(d) = P_r\left(CM^{(1)} > CM^{(0)}\right)$$

$$= P_r\left(CM^{(l)} - CM^{(0)} > 0\right). \tag{6.57}$$

If both metric values are equal, a random choice with probability 1/2 is made.

If we substitute equation (6.55) and (6.56) into (6.57) we obtain

$$P_2(d) = P_r\{(d_G + Cd_B) < 2(e_G + Ce_B)\}, \tag{6.58}$$

where C is the metric ratio defined as

$$C = \frac{\log[(1 - P_e(B))/P_e(B)]}{\log[(1 - P_e(G))/P_e(G)]}. \tag{6.59}$$

To evaluate $P_2(d)$, we need the probability distribution for being in the bad state d_B times out of d, $P_d(d_B)$, and the distribution for being in the good state d_G times out of d, $P_d(d_G)$. We show that (see proof in Appendix C)

$$P_d(d_B) = \begin{cases} (1-b)^{d-1}\pi_G, & d_B = 0 \\ [P_d(d_B/GG) + P_d(d_B/GB)]\pi_G + [P_d(d_B/BG) + P_d(d_B/BB)]\pi_B, & 1 \le d_B < d \\ (1-g)^{d-1}\pi_B, & d_B = d \end{cases} \tag{6.60}$$

where

$$P_d(d_B/GG) = \sum_{i=2}^{\min(d_B+1,\,d-d_B)} \binom{d-d_B-1}{i-1}\binom{d_B-1}{i-2}(1-b)^{d-d_B-i}b^{i-1}(1-g)^{d_B-i+1}g^{i-1}, \tag{6.61a}$$

$$P_d(d_B/GB) = \sum_{i=1}^{\min(d_B,\,d-d_B)} \binom{d-d_B-1}{i-1}\binom{d_B-1}{i-1}(1-b)^{d-d_B-i}b^{i}(1-g)^{d_B-i}g^{i-1}, \tag{6.61b}$$

$$P_d(d_B/BG) = \sum_{i=1}^{\min(d_B,\,d-d_B)} \binom{d-d_B-1}{i-1}\binom{d_B-1}{i-1}(1-b)^{d-d_B-i}b^{i-1}(1-g)^{d_B-i}g^{i}, \tag{6.61c}$$

$$P_d(d_B/BB) = \sum_{i=2}^{\min(d_B,\,d-d_B+1)} \binom{d-d_B-1}{i-2}\binom{d_B-1}{i-1}(1-b)^{d-d_B-i+1}b^{i-1}(1-g)^{d_B-i}g^{i-1}. \tag{6.61d}$$

Here $P_d(d_B/GG)$ is the conditional probability of being d_B times in the bad state, conditioned on being in the good state both the first and the last instants of time, and the other conditional probabilities are defined accordingly.

Likewise,

$$P_d(d_G) = \begin{cases} (1-g)^{d-1}\pi_B, & d_G = 0 \\ [P_d(d_G/GG) + P_d(d_G/GB)]\pi_G + [P_d(d_G/BG) + P_d(d_G/BB)]\pi_B, & 1 \le d_G < d \\ (1-b)^{d-1}\pi_G, & d_G = d \end{cases} \tag{6.62}$$

where

$$P_d(d_G/GG)=\sum_{i=2}^{\min(d_G,d-d_G+1)}\binom{d-d_G-1}{i-2}\binom{d_G-1}{i-1}(1-b)^{d_G-i}b^{i-1}(1-g)^{d-d_G+1}g^{i-1} \quad (6.63a)$$

$$P_d(d_G/GB)=\sum_{i=1}^{\min(d_G,d-d_G)}\binom{d-d_G-1}{i-1}\binom{d_G-1}{i-1}(1-b)^{d_G-i}b^{i}(1-g)^{d-d_G-i}g^{i-1} \quad (6.63b)$$

$$P_d(d_G/BG)=\sum_{i=1}^{\min(d_G,d-d_G)}\binom{d-d_G-1}{i-1}\binom{d_G-1}{i-1}(1-b)^{d_G-i}b^{i-1}(1-g)^{d-d_G-i}g^{i} \quad (6.63c)$$

$$P_d(d_G/BB)=\sum_{i=2}^{\min(d_G+1,d-d_G)}\binom{d-d_G-1}{i-1}\binom{d_G-1}{i-2}(1-b)^{d_G-i+1}b^{i-1}(1-g)^{d-d_G-i}g^{i-1} \quad (6.63d)$$

The conditional probabilities here are also defined as above.

Thus

$$P_2(d)=\sum_{d_B+dG=d}\sum_{e_B}\sum_{e_G}\binom{d_B}{e_B}P_e(B)^{e_B}(1-P_e(B))^{d_B-e_B}P_d(d_B)\binom{d_G}{e_G}P_e(G)^{e_G}(1-P_e(G))^{d_G-e_G}P_d(d_G) \quad (6.64)$$

if we restrict the summation over e_B and e_G to those value e_B and e_G which fulfill the inequality in (6.58) and introduce the factor 1/2 in case of equality.

6.7 TURBO DECODER FOR GILBERT-ELLIOTT CHANNEL

6.7.1 Original Turbo Decoder

Because the optimal decoder is too complex, the turbo decoder provides a suboptimal alternative that iteratively passes log-likelihood information between a pair of MAP decoders matched to each of the component encoders. The turbo decoding algorithm has been well documented in Chapter 2 and Appendix A, thus it will not be repeated here. Of particular interest, however, are the branch transition probabilities which are needed for turbo decoder calculations. The computation of branch transition probabilities depend on the channel, so they play a key role in the design of the turbo decoder for the Gilbert-

Elliott channel model. Let s_k be the state of the first encoder at time k. The branch transition probabilities used by the MAP algorithm are calculated as

$$\gamma_k(s',s) = p(s_k = s, z_k^s, z_k^p / s_{k-1} = s'), \tag{6.65}$$

where we have defined z_k^s and z_k^p as the received decision variables originating from the systematic and parity symbols respectively. The above equation can be rewritten as

$$\begin{aligned}\gamma_k(s',s) &= Pr(s/s')p(z_k^s, z_k^p / s', s) \\ &= Pr(u_k)p(z_k^s / u_k)p(z_k^p / u_k),\end{aligned} \tag{6.66}$$

where u_k corresponds to the event $s' \rightarrow s$. It can be shown that

$$Pr(u_k) = A_k \exp[u_k L^e(u_k)/2], \tag{6.67}$$

where A_k is a constant not depending on u_k and

$$L^e(u_k) \equiv \log\left(\frac{Pr(u_k = +1)}{Pr(u_k = -1)}\right) \tag{6.68}$$

is the extrinsic information.

6.7.2 Modification to the Turbo Decoder

The turbo decoding algorithm is dependent on what information is available to the turbo decoder. We consider two cases: known channel state and unknown state but known steady state probabilities π_G and π_B.

If the state, G or B, is known, then the modification to the turbo decoder is straightforward. The turbo decoder can simply use the relevant noise variance to calculate the branch transition probabilities. Thus, (6.66) can be calculate using

$$p(z_k^i / u_k) = \frac{1}{\sqrt{\pi N_j}} \exp\left[-\frac{1}{N_j}(z_k^i - u_k^i)^2\right], \tag{6.69}$$

where $i \in \{s, p\}$ and $j \in \{G, B\}$.

If the channel state is unknown but the steady state probabilities are known, then (6.66) can be calculated by invoking the total probability with respect to the channel state.

$$p(z_k^i / u_k) = \frac{1}{\sqrt{\pi N_G}} \exp\left[-\frac{1}{N_G}(z_k^i - u_k^i)^2\right]\pi_G + \frac{1}{\sqrt{\pi N_B}} \exp\left[-\frac{1}{N_B}(z_k^i - u_k^i)^2\right]\pi_B, \tag{6.70}$$

where $i \in \{s,p\}$.

6.8 SIMULATION MODEL AND RESULTS

For all simulations, the component encoders are rate one-half, recursive, systematic convolutional encoders with memory four and octal generators (7, 5). Gold spreading sequences of length $N = 63$ are assumed. The number of users is 10. In order to represent a wide range of mobile communication environments, the product $f_D T_S$ was considered as an independent parameter fm and simulations were performed for fm = 0.001, 0.01, and 0.1. These values encompass the whole range of mobile communication environments ranging from cellular telephony to PCS. We assumed perfect channel side information. This means that the state of the Gilbert-Elliot burst channel and the instantaneous amplitudes for the fading channel are known to the turbo decoder. The MAP algorithm and a total of 8 decoding iterations were used. The frame size is $N1$ = 1024 bits unless otherwise specified. The signal-to-noise threshold on the Gilbert-Elliott channel model is 10 dB for fm = 0.1, 0.01 and 14 dB for fm = 0.001 unless otherwise specified.

In Fig. 6.8, the performance of the system is simulated for both the Rayleigh fading channel and the equivalent Gilbert-Elliott channel model.

Since the threshold is chosen somewhat ad hoc, it is desirable that the performance of the system is not dramatically affected by a minor variation of this parameter. That this is the case can be seen in Fig. 6.9 where the performance of the system is simulated for different values of the threshold and the parameter fm. It can be noted, however, that the degradation in performance increased as the normalized Doppler frequency decreases. For fm = 0.001, the performance of the system at high signal-to-noise ratio is greatly affected by the choice of the threshold.

In Fig. 6.10, the bound on the bit error rate is evaluated for different values of the normalized Doppler frequency and for various block lengths. Equations (6.36), (6.40), (6.53) and (6.64) are used. It can be noticed that the bound is useless at low signal-to-noise ratio. This behavior mimics that of similar bounds applied to totally random codes, which turbo codes resemble. This divergence is an artifact of the bound, as the actual performance of the system does not diverge at low signal–to-noise ratio. In computing these bounds, we realized that only a handful of terms ($i \leq 10$, $d \leq 20$) are needed for convergence, and that this is almost independent of the values of $N1$ and the normalized Doppler frequency.

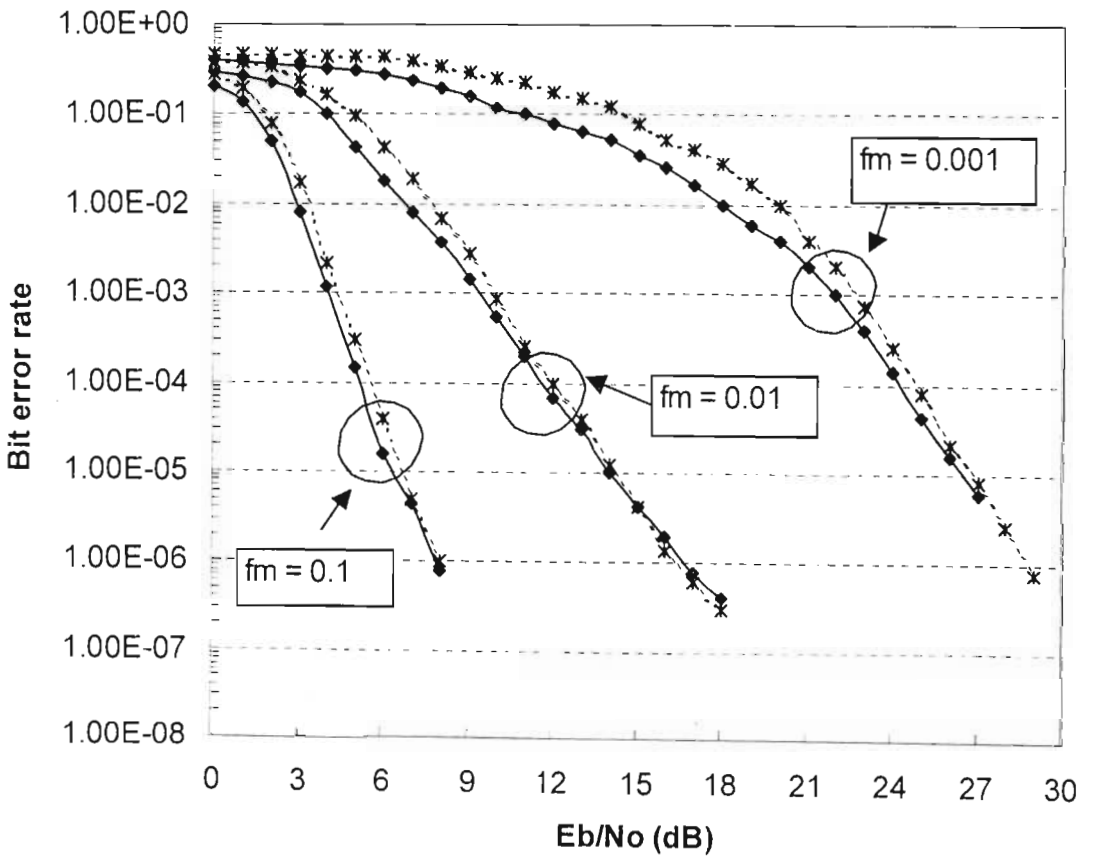


Fig. 6.8: Simulated bit error rate performance on the Rayleigh fading channel (dashed lines) and on the Gilbert-Elliott model (solid lines).

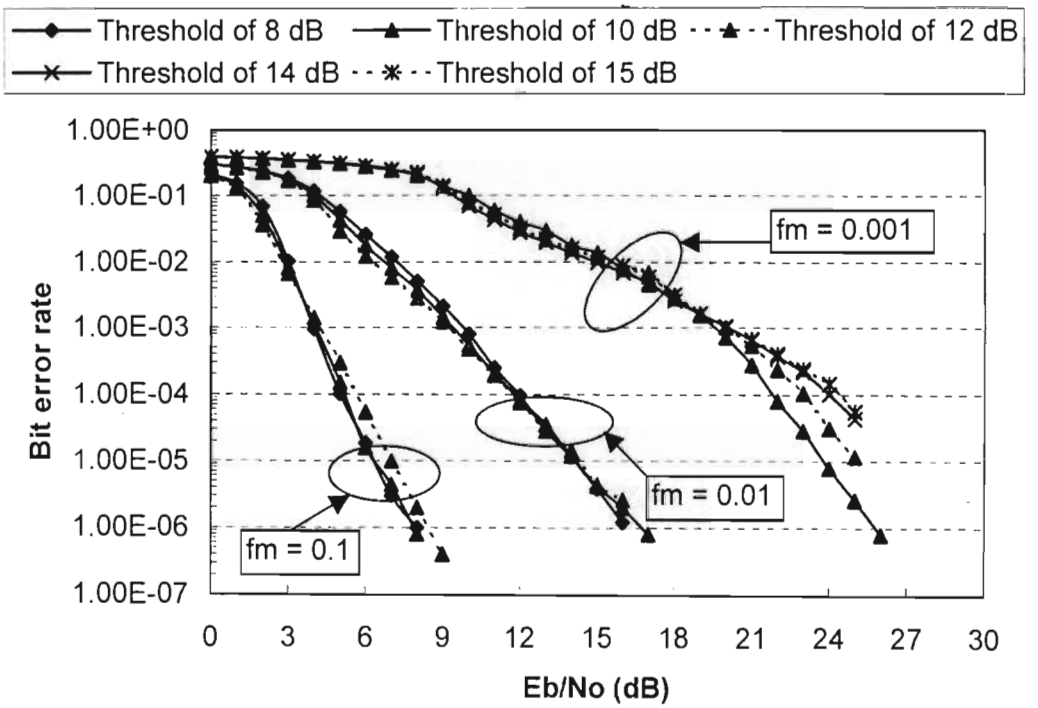


Fig. 6.9: Simulated bit error rate performance on the Gilbert-Elliot channel for different values of the threshold.

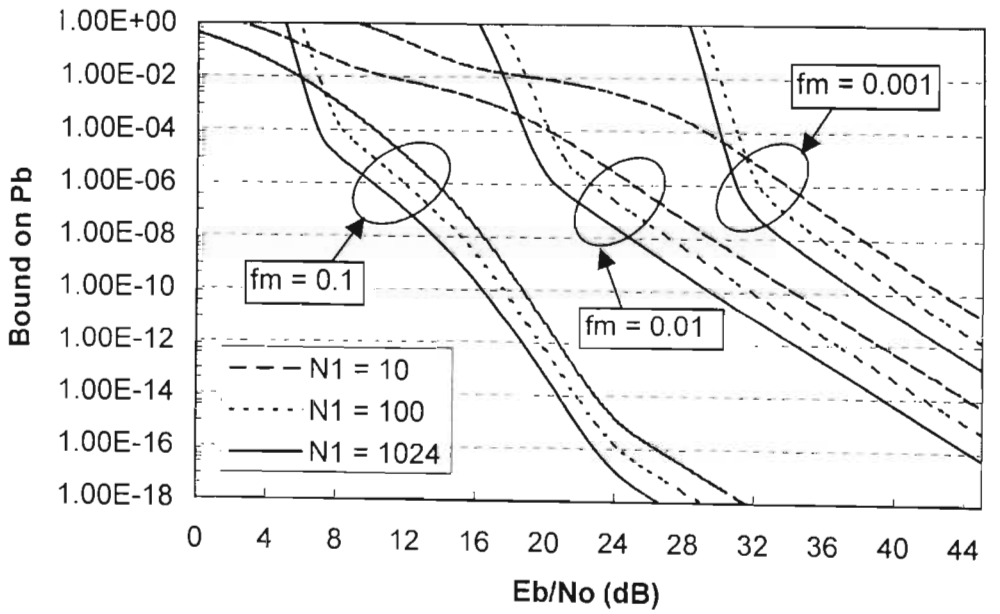


Fig. 6.10: Bounds on the bit error rate for various block lengths N1 and different values of the parameter fm.

In Fig. 6.11, we assess the effect of the threshold on the accuracy of the bound. We see that a minor variation in this parameter does not dramatically affect the accuracy of the bound.

A comparison of the transfer function bound with the simulated results in Fig. 6.12 confirms that the former can be used to predict the system performance in the range where obtaining sufficient data from simulations is impractical and can therefore be used to determine the coding gain.

In Fig. 6.13, we investigate the effect of the interleaver depth on the performance of the system. As should be expected, the performance of the system improves as the depth of the interleaver increases. Looking at the bit error rate of 10^{-3}, we see that the gain of using an interleaver of depth 10, 20, 40, and 60 is 5 dB, 6 dB, 6.5 dB and 7 dB respectively. However, as the interleaver depth increases beyond 60, the improvement in performance becomes negligible. This observation was shown to be independent of the interleaver size. It is well known that reaching the Shannon limits with turbo codes seems to be approachable with only very large interleavers, causing tremendous transmission delays and prohibiting an application for delay-sensitive services.

In Fig. 6.14, we compare the performance of a code interleaver of size 100 and channel interleaver of depth 50 with the performance of very large code interleavers without channel interleaver. At an acceptable data rate (greater or equal to 9.6 kbits/s), the delay resulting from such a combination would still be acceptable even for delay-sensitive services. It can be seen that this combination outperforms a code interleaver size of even 60000. This result is very interesting in the sense that it makes turbo codes suitable for even delay-sensitive services.

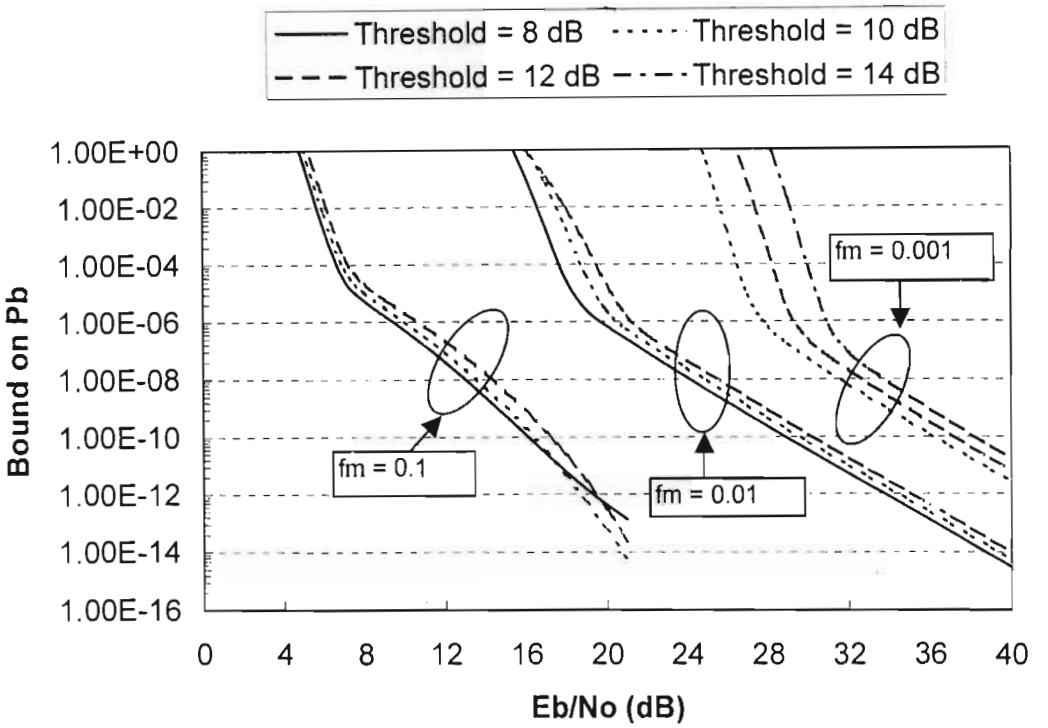


Fig. 6.11: Bounds on the bit error rate for various values of the threshold

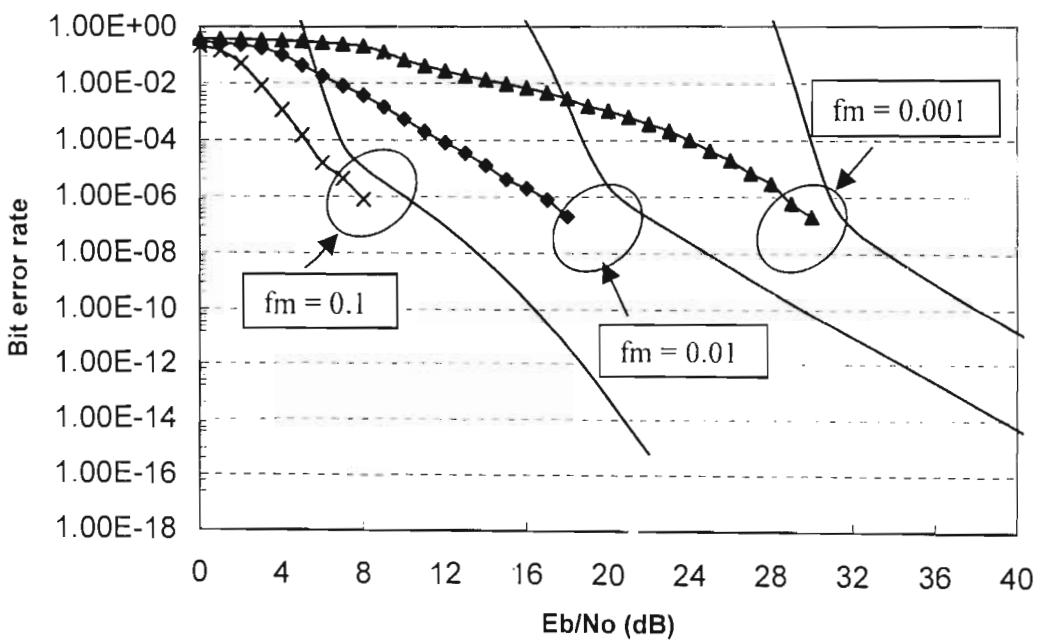


Fig. 6.12: Transfer function bound (solid lines) versus simulation for various values of fm.

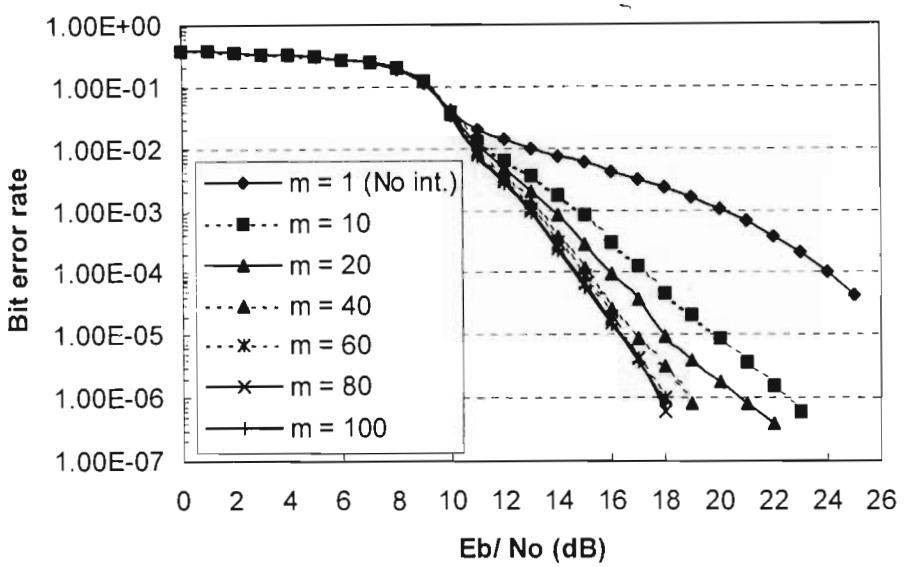


Fig. 6.13: Simulated bit error rate on the interleaved Gilbert-Elliott channel model for different values of the interleaver depth. fm = 0.001.

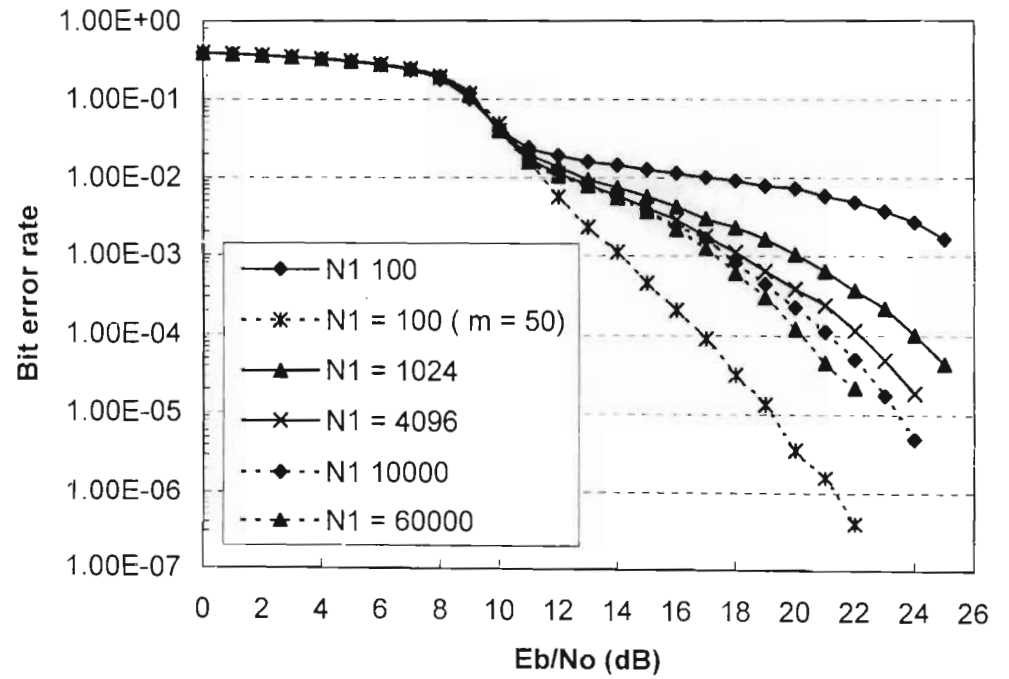


Fig. 6.14: Simulated bit error rate comparison of the performance of a code interleaver of size 100 on the interleaved Gilbert-Elliott channel model with a channel interleaver of depth 50 and the performance of larger code interleavers of various sizes without channel interleaving.

6.9 CONCLUSIONS

In this chapter, we consider the design and performance of a turbo-coded DS-CDMA system in a slowly varying Rayleigh fading channel and an equivalent Gilbert-Elliott model. The accuracy of using the Gilbert-Elliott model is demonstrated by comparing simulation results on both the Rayleigh fading channel and the Gilbert-Elliott model, showing that both models are equivalent for a broad class of parameters and that moderate variations in the Gilbert-Elliott model only caused small variations in the bit error rate.

We derive closed form expressions for the error probabilities in each state of the Gilbert-Elliott channel model for BPSK modulation. We also derive an expression for the pairwise-error probability for the Gilbert-Elliott channel model and expressions for the probability distribution for being in the bad state d_B times out of d and the distribution for being in the good state d_G out of d on the Gilbert-Elliott channel model. Using the transfer function bounding techniques, we obtain upper bounds on the bit-error rate for a maximum likelihood decoding of turbo codes constructed with random permutations. We obtain this upper bound as a function of the channel parameters, the interleaver parameters and the code. This bound is very useful in establishing the decoder performance in the range where obtaining sufficient data from simulations is impractical.

Finally, we assess the effect of imperfect interleaving on the GE channel model. Using the Chapman-Kolmogrov equation, we obtain the transition probabilities of an interleaved GE channel in terms of the transition probabilities of the original model. The Gilbert-Elliott model was then used to compare the performance of different interleaver sizes to see the effect of imperfect interleaving. It was found that a combination of a short code interleaver with a channel interleaver of appropriate depth could outperform a very large interleaver yet resulting in an acceptable delay for even delay-sensitive services such as interactive voice and video and thus making turbo codes suitable for such services.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

In this dissertation, the concepts of turbo coding have been presented and applied to a practical digital communication system over a wireless channel. Turbo codes are a novel error correction technique that was first introduced in 1993 by a group of French researchers [27]. The essential features of turbo codes are parallel code concatenation, recursive encoding, pseudo-random interleaving, and iterative decoding.

The first phase of this research investigated stopping criteria for turbo decoding. Iterative decoding is a key feature of turbo codes. Each decoding iteration results in additional computations and decoding delay. As the decoding approaches the performance limit of a given turbo-code, any further iteration results in very little improvement. Often, a fixed number M is chosen and each frame is decoded for M iterations. Usually M is set with the worst corrupted frames in mind. Most frames need fewer iterations to converge. It would reduce the average computation substantially without performance degradation if the decoder terminated the iterations for each individual frame immediately after the bits are correctly estimated. Four stopping criteria have been proposed and used in the literature.

We have compared the four schemes and designed a new and simple stopping criterion for turbo decoding. Our scheme was shown to be more effective, requiring no extra data storage and saving more iterations for small interleaver sizes. In addition, its performance is independent of the choice of any parameter. This work has been presented at the 2001 SATNAC conference [138], and has been published in Electronic Letters [139].

The second phase of this research involved the characterization of turbo-coded DS-CDMA systems in channels with memory. Studies of the performance of turbo-coded DS-CDMA systems are most often concerned with situations where the channel is assumed to be memoryless, since this allows for a theoretical analysis. In situations where memory is accounted for, on the other hand, the analytical results are few, and studies of the performance are obtained via simulations. We approximated the memory of the channel by a regenerative two state Markov chain, the Gilbert-Elliott channel model. The two states in this model correspond to the faded and non-faded state and the transition probabilities can be related to the communication system parameters. The dynamics of the channel are modeled as a first-order Markov chain. The accuracy of using the Gilbert-Elliott model is demonstrated by comparing simulation results on both the Rayleigh fading channel and the Gilbert-Elliott model, showing that both models are equivalent for a broad class of parameters and that moderate variations in the Gilbert-Elliott model only caused small variations in the bit error rate. Using the transfer function bounding techniques, we obtain upper bounds on the bit-error rate for a maximum likelihood decoding of turbo codes constructed with random permutations. We obtain this upper bound as a function of the channel parameters, the interleaver parameters and the code. Finally, the Gilbert-Elliott model was used to compare the performance of different interleaver sizes to see the effect of imperfect interleaving. It was found that a combination of a short code interleaver with a channel interleaver of appropriate depth could outperform a very large code interleaver yet resulting in an acceptable delay for even delay-sensitive services such as interactive voice and video and thus making turbo codes applicable to such services. This work has been submitted to IEEE transactions on Vehicular Technologies [140].

Future research should extend the work presented in Chapter 6 on the performance of turbo-coded DS-CDMA system to multipath fading channel and consider various channel impairments. It is also necessary to develop and investigate channel estimation schemes.

The concept of iterative decoding, although originally developed as a practical means for decoding turbo codes has since then found application for many other coding structures and other functionalities of a communication system. For other applications, such as multiuser detection, the complexity of full iterative decoding is prohibitive. So far, little effort has been directed towards reduced complexity alternatives for iterative decoding. Future research should aim at developing strategies for complexity reduction and evaluating them in terms of complexity/performance trade-offs.

Although bounds based on analytical methods using the uniform interleaver concept can be computed for turbo codes, the actual performance depends on the particular interleaver used in the turbo encoder. Discrepancies of two to three orders of magnitude are common. Another area of future interest would be to investigate the application of importance sampling techniques to turbo codes. Importance sampling is a Monte-Carlo technique used to reduce significantly the amount of computations and hence, the duration of simulations, required to obtain a specified precision. The basic idea of importance sampling is to generate the random inputs using a *simulation distribution* which increases the relative frequency of "important" events. The observed simulation data are then weighted by the *importance sampling weight* in order to obtain an unbiased estimator.

APPENDICES

APPENDIX A

Derivation of Equations Used in the Turbo Decoder

This is a summary of the equations used in the turbo decoder. A more complete discussion can be found in [54] and [56].

We first discuss a modified version of the BCJR algorithm for performing symbol-by-symbol MAP decoding. We then show how this algorithm is incorporated into an iterative decoder employing two BCJR-MAP decoders. We shall require the following definitions:

- E1 is a notation for encoder 1
- E2 is a notation for encoder 2
- D1 is a notation for decoder 1
- D2 is a notation for decoder 2
- m is the constituent encoder memory
- S is the set of all 2^m constituent encoder states
- $\mathbf{x}^s = (x_1^s, x_2^s, ..., x_N^s) = (u_1, u_2, ..., u_N)$ is the encoder input word
- $\mathbf{x}^p = (x_1^p, x_2^p, ..., x_N^p)$ is the parity word generated by a constituent encoder
- $y_k = (y_k^s, y_k^p)$ is a noisy (AWGN) version of (x_k^s, x_k^p)
- $\mathbf{y}_a^b = (y_a, y_{a+1}, ..., y_b)$
- $\mathbf{y} = \mathbf{y}_1^N = (y_1, y_2, ..., y_N)$ is the noisy received codeword

A.1 The Modified BCJR Algorithm

In the symbol-by-symbol MAP decoder, the decoder decides $u_k = +1$ if $P(u_k = +1/y) > P(u_k = -1/y)$, and it decides $u_k = -1$ otherwise. More succinctly, the decision $\hat{u}_k$ is given by

$$\hat{u}_k = \text{sign}[L(u_k)]; \tag{A.1}$$

where $L(u_k)$ is the log *a posteriori* probability (LAPP) ratio defined as

$$L(u_k) = \log\left(\frac{P(u_k = +1/\mathbf{y})}{P(u_k = -1/\mathbf{y})}\right). \tag{A.2}$$

Incorporating the code's trellis, this may be written as

$$L(u_k) = \log\left(\frac{\sum_{S^+} p(s_{k-1} = s', s_k = s, \mathbf{y})/p(\mathbf{y})}{\sum_{S^-} p(s_{k-1} = s', s_k = s, \mathbf{y})/p(\mathbf{y})}\right), \tag{A.3}$$

where $s_k \in S$ is the state of the encoder at time k, S^+ is the set of ordered pairs (s', s) corresponding to all state transitions $(s_{k-1} = s') \rightarrow (s_k = s)$ caused by data input $u_k = +1$, and S^- is similarly defined for $u_k = -1$.

Observe that we may cancel $p(\mathbf{y})$ in (A.3) which means we require only an algorithm for computing $p(s', s, \mathbf{y}) = p(s_{k-1} = s', s_k = s, \mathbf{y})$. The BCJR algorithm [54] for doing this is

$$p(s', s, \mathbf{y}) = \alpha_{k-1}(s')\gamma_k(s', s)\beta_k(s), \tag{A.4}$$

where $\alpha_k(s) \equiv p\left(s_k = s, \mathbf{y}_1^k\right)$ is computed recursively as

$$\alpha_k(s) = \sum_{s' \in S} \alpha_{k-1}(s')\gamma_k(s', s), \tag{A.5}$$

with initial conditions

$$\alpha_0(0) = 1 \text{ and } \alpha_0(s \neq 0) = 0. \tag{A.6}$$

(These conditions state that the encoder is expected to start in state 0.) The probability $\gamma_k(s', s)$ in (A.5) is defined as

$$\gamma_k(s', s) \equiv p(s_k = s, y_k / s_{k-1} = s'), \tag{A.7}$$

and will be discussed further below. The probabilities $\beta_k(s) \equiv p\left(\mathbf{y}_{k+1}^N / s_k = s\right)$ in (A.4) are computed in a "backward" recursion as

$$\beta_{k-1}(s') = \sum_{s \in S} \beta_k(s)\gamma_k(s', s), \tag{A.8}$$

with boundary conditions

$$\beta_N(0) = 1 \text{ and } \beta_N(s \neq 0) = 0. \tag{A.9}$$

(The encoder is expected to end in state 0 after N input bits, implying that the last m input bits, called *termination bits*, are also selected.)

Unfortunately, canceling the divisor $p(\mathbf{y})$ in (A.3) leads to a numerically unstable algorithm. We can include division by $p(\mathbf{y})/p(y_k)$ in the BCJR algorithm[10] by defining the modified probabilities

$$\tilde{\alpha}_k(s) = \alpha_k(s)/p\left(\mathbf{y}_1^k\right),$$

and

$$\tilde{\beta}_k(s) = \beta_k(s)/p\left(\mathbf{y}_{k+1}^N/\mathbf{y}_1^k\right).$$

Dividing (A.4) by $p(\mathbf{y})/p(y_k) = p\left(\mathbf{y}_1^{k-1}\right)p\left(\mathbf{y}_{k+1}^N/\mathbf{y}_1^k\right)$, we obtain

$$p(s',s/\mathbf{y})p(y_k) = \tilde{\alpha}_{k-1}(s')\gamma_k(s',s)\tilde{\beta}_k(s). \tag{A.10}$$

Note since $p\left(\mathbf{y}_1^k\right) = \sum_{s\in S}\alpha_k(s)$, the values $\tilde{\alpha}_k(s)$ may be computed from $\{\alpha_k(s): s\in S\}$ via

$$\tilde{\alpha}_k(s) = \alpha_k(s)\Big/\sum_{s\in S}\alpha_k(s). \tag{A.11}$$

But since we would like to avoid storing both $\{\alpha_k(s)\}$ and $\{\tilde{\alpha}_k(s)\}$, we can use (A.5) in (A.11) to obtain a recursion involving only $\{\tilde{\alpha}_k(s)\}$,

$$\begin{aligned}\tilde{\alpha}_k(s) &= \frac{\sum_{s'}\alpha_{k-1}(s')\gamma_k(s',s)}{\sum_s\sum_{s'}\alpha_{k-1}(s')\gamma_k(s',s)} \\ &= \frac{\sum_{s'}\tilde{\alpha}_{k-1}(s')\gamma_k(s',s)}{\sum_s\sum_{s'}\tilde{\alpha}_{k-1}(s')\gamma_k(s',s)}.\end{aligned} \tag{A.12}$$

where the second equality follows by dividing the numerator and denominator by $p\left(\mathbf{y}_1^{k-1}\right)$.

The recursion for $\tilde{\beta}_k(s)$ can be obtained by noticing that

$$p\left(\mathbf{y}_k^N/\mathbf{y}_1^{k-1}\right) = p\left(\mathbf{y}_1^k\right)\frac{p\left(\mathbf{y}_{k+1}^N/\mathbf{y}_1^k\right)}{p\left(\mathbf{y}_1^{k-1}\right)}$$

[10] Unfortunately, dividing by simply $p(\mathbf{y})$ to obtain $p(s',s/\mathbf{y})$ also leads to an unstable algorithm. Obtaining $p(s',s/\mathbf{y})\,p(y_k)$ instead of the APP $p(s',s/\mathbf{y})$ presents no problem since an APP ratio is computed so that the unwanted factor $p(y_k)$ cancels; see equation (A.14) below.

$$= \sum_{s} \sum_{s'} \alpha_{k-1}(s') \gamma_k(s', s) . \frac{p\left(\mathbf{y}_{k+1}^{N} / \mathbf{y}_{1}^{k}\right)}{p\left(\mathbf{y}_{1}^{k-1}\right)}$$

$$= \sum_{s} \sum_{s'} \tilde{\alpha}_{k-1}(s') \gamma_k(s', s) . p\left(\mathbf{y}_{k+1}^{N} / \mathbf{y}_{1}^{k}\right),$$

so that dividing (A.8) by this equation yields

$$\tilde{\beta}_{k-1}(s') = \frac{\sum_{s} \tilde{\beta}_k(s) \gamma_k(s', s)}{\sum_{s} \sum_{s'} \tilde{\alpha}_{k-1}(s') \gamma_k(s', s)} . \tag{A.13}$$

In summary, the modified BCJR-MAP algorithm involves computing the LAPP ratio $L(u_k)$ by combining (A.3) and (A.10) to obtain

$$L(u_k) = \log\left(\frac{\sum_{S^+} \tilde{\alpha}_{k-1}(s') . \gamma_k(s', s) . \tilde{\beta}_k(s)}{\sum_{S^-} \tilde{\alpha}_{k-1}(s') . \gamma_k(s', s) . \tilde{\beta}_k(s)} \right), \tag{A.14}$$

where the $\tilde{\alpha}$'s and $\tilde{\beta}$'s are computed recursively via (A.12) and (A.13) respectively. Clearly the $\{\tilde{\alpha}_k(s)\}$ and $\{\tilde{\beta}_k(s)\}$ share the same boundary conditions as their counterparts as given in (A.6) and (A.9). Computation of the probabilities $\gamma_k(s', s)$ will be discussed shortly.

A.2 Iterative MAP Decoding

From the Bayes' rule, the LAPP ratio for an arbitrary MAP decoder can be written as

$$L(u_k) = \log\left(\frac{P(\mathbf{y} / u_k = +1)}{P(\mathbf{y} / u_k = -1)} \right) + \log\left(\frac{P(u_k = +1)}{P(u_k = -1)} \right),$$

with the second term representing *a-priori* information. Since $P(u_k = +1) = P(u_k = -1)$ typically, the *a-priori* term is usually zero for conventional decoders. However, for *iterative* decoders, D1 receives *extrinsic* or *soft* information for each u_k from D2 which serves as *a-priori* information. Similarly, D2 receives extrinsic information from D1 and the decoding iteration proceeds as D1→D2→D1→D2..., with the previous decoder passing soft information along to the next decoder at each half-iteration except for the

first. The idea behind extrinsic information is that D2 provides soft information to D1 for each u_k, using only information not available to D1 (i.e., E2 parity); D1 does likewise for D2.

An iterative decoder using component BCJR-MAP decoders was shown in Fig. 2.4. Observe how interleavers and de-interleavers are involved in arranging systematic, parity, and extrinsic information in the proper sequence for each decoder.

We now show how extrinsic information is extracted from the modified-BCJR version of the LAPP ratio embodied in (A.14). We first observe that $\gamma_k(s',s)$ may be written as (cf. equation (A.7))

$$\begin{aligned}\gamma_k(s',s) &= P(s/s')p(y_k/s',s)\\ &= P(u_k)p(y_k/u_k)\end{aligned}$$

where the event u_k corresponds to the event $s' \to s$. Defining

$$L^e(u_k) \equiv \log\left(\frac{P(u_k=+1)}{P(u_k=-1)}\right),$$

observe that we may write

$$\begin{aligned}P(u_k) &= \left(\frac{\exp\left[-L^e(u_k)/2\right]}{1+\exp\left[-L^e(u_k)\right]}\right).\exp\left[u_k\, L^e(u_k)/2\right]\\ &= A_k \exp\left[u_k\, L^e(u_k)/2\right]. \qquad \text{(A.15)}\end{aligned}$$

As for $p(y_k/u_k)$, we may write (recall $y_k = \left(y_k^s, y_k^p\right)$ and $x_k = \left(x_k^s, x_k^p\right) = \left(u_k, x_k^p\right)$)

$$\begin{aligned}p(y_k/u_k) &\propto \exp\left[-\frac{\left(y_k^s-u_k\right)^2}{2\sigma^2}-\frac{\left(y_k^s-x_k^p\right)^2}{2\sigma^2}\right]\\ &= \exp\left[-\frac{y_k^{s\,2}+u_k^2+y_k^{p\,2}+x_k^{p\,2}}{2\sigma^2}\right].\exp\left[\frac{u_k y_k^s + x_k^p y_k^p}{\sigma^2}\right]\\ &= B_k \exp\left[\frac{y_k^s u_k + y_k^p x_k^p}{\sigma^2}\right],\end{aligned}$$

so that

$$\gamma_k(s',s) \propto A_k B_k \exp\left[u_k L^e(u_k)/2\right]\exp\left[\frac{u_k y_k^s + x_k^p y_k^p}{\sigma^2}\right]. \qquad (A.16)$$

Now since $\gamma_k(s',s)$ appears in the numerator (where $u_k = +1$) and denominator (where $u_k = -1$) of (A.14), the factor $A_k B_k$ will cancel out as it is independent of u_k. Also, since we assume transmission of the symbols ± 1 over the channel, $\frac{E_c}{N_0/2} = \frac{1}{\sigma^2}$ so that $\sigma^2 = N_0/2E_c$ where $E_c = rE_b$ is the energy per channel bit. From (A.16), we then have

$$\gamma_k(s',s) \sim \exp\left[\frac{1}{2}u_k\left(L^e(u_k) + L_c y_k^s\right) + \frac{1}{2}L_c y_k^p x_k^p\right]$$

$$= \exp\left[\frac{1}{2}u_k\left(L^e(u_k) + L_c y_k^s\right)\right].\gamma_k^e(s',s), \qquad (A.17)$$

where $L_c \equiv \frac{4E_c}{N_0}$ and where

$$\gamma_k^e(s',s) \equiv \exp\left[\frac{1}{2}L_c y_k^p x_k^p\right].$$

Combining (A.17) with (A.14) we obtain

$$L(u_k) = \log\left(\frac{\sum_{S^+} \tilde{\alpha}_{k-1}(s').\gamma_k^e(s',s).\tilde{\beta}_k(s).C_k}{\sum_{S^-} \tilde{\alpha}_{k-1}(s').\gamma_k^e(s',s).\tilde{\beta}_k(s).C_k}\right)$$

$$= L_c y_k^s + L^e(u_k) + \log\left(\frac{\sum_{S^+} \tilde{\alpha}_{k-1}(s').\gamma_k^e(s',s).\tilde{\beta}_k(s)}{\sum_{S^-} \tilde{\alpha}_{k-1}(s').\gamma_k^e(s',s).\tilde{\beta}_k(s)}\right). \qquad (A.18)$$

where $C_k \equiv \exp\left[\frac{1}{2}u_k\left(L^e(u_k) + L_c y_k^s\right)\right]$. The second equality follows since $C_k(u_k = +1)$ and $C_k(u_k = -1)$ can be factored out of the summations in the numerator and denominator, respectively. The first term in (A.18) is called the *channel value*, the second term represents any *a-priori* information about u_k provided by a previous decoder, and the third term represents *extrinsic information* that can be passed on to a subsequent decoder.

APPENDIX B

Calculation of $p(d/i)$ for the Union Bound

In this section, the calculation of $p(d/i)$ which is essential to the computation of the union bound for turbo codes is described [136]. The state transition matrix details the input/output relationship of the encoder. A sample state transition matrix is shown below for a convolutional code with generator polynomial (7, 5) in octal notation.

$$\mathbf{A}(L,I,D)=\begin{pmatrix} L & LID & 0 & 0 \\ 0 & 0 & LD & LI \\ LID & L & 0 & 0 \\ 0 & 0 & LI & LD \end{pmatrix}. \tag{B.1}$$

The rows index the states of the encoder before a transition while the columns index the states of the encoder after a transition. The monomial $L^l I^i D^d$ describes the input/output relationship for a specific transition. The variables i and d are equal to 0 or 1 depending on whether the corresponding input and output bits, respectively, are 0 or 1 and l, which represents the length of the path is always equal to 1.

If we define $t(l,i,d)$ as the number of paths of length l, input weight i, and output weight d which start and end in the all-zero state (represented by 0^m), then we define the transfer function as follows:

$$T(L,I,D)=\sum_{l\geq 0}\sum_{i\geq 0}\sum_{d\geq 0} L^l I^i D^d t(l,i,d)\ . \tag{B.2}$$

According to [142], $T(L,I,D)$ is the $(0^m, 0^m)$ entry in the matrix

$$\left(\mathbf{I}+\mathbf{A}(L,I,D)+\mathbf{A}(L,I,D)^2+\mathbf{A}(L,I,D)^3\ldots\right)\mathbf{A}(L,I,D)^m\ . \tag{B.3}$$

The factor $\mathbf{A}(L,I,D)^m$ takes care of the termination edges. Since $\mathbf{I}+\mathbf{A}+\mathbf{A}^2+\mathbf{A}^3+\ldots=(\mathbf{I}-\mathbf{A})^{-1}$, it follows from Eq. (B.3) that

$$T(L,I,D)=\left[(I-A)^{-1}A(1,I,D)^{m}\right]_{0^m,0^m}. \tag{B.4}$$

Using Eq. (B.4) and by omitting the termination factor $\mathbf{A}(L,I,D)^m$, we can get the following approximation for the example state transition shown above:

$$T(L,I,D)\approx\frac{1-LD-L^2D+L^3\left(D^2-I^2\right)}{1-L(1+D)+L^3\left(D+D^2-I^2-I^2D^3\right)-L^4\left(I^4D^2-I^2D^4+D^2-I^2\right)}. \tag{B.5}$$

If we multiply both sides of Eq. (B.5) by the denominator of the right-hand side, and take the coefficient of $t(l,i,d)$ of both sides of the resulting equation, we obtain the following recursion

$$\begin{aligned}t(l,i,d)&=t(l-1,i,d-1)+t(l-1,i,d)\\&+t(l-3,i-2,d-3)+t(l-3,i-2,d)-t(l-3,i,d-2)-t(l-3,i,d-1)\\&+t(l-4,i-4,d-2)-t(l-4,i-2,d-4)-t(l-4,i-2,d)+t(l-4,i,d-2)\\&+\delta(l,i,d)-\delta(l-1,i,d-1)-\delta(l-2,i,d-1)+\delta(l-3,i,d-2)-\delta(l-3,i-2,d).\end{aligned} \tag{B.6}$$

By assuming a uniform interleaver, we can compute $p(d/i)$ for each code fragment.

$$p(d/i)=\frac{t(N,i,d)}{\sum_{d'}t(N,i,d')}=\frac{t(N,i,d)}{\binom{N}{i}}. \tag{B.7}$$

Finally, we can calculate $p(d/i)$ for the entire code by using

$$p(d=d_0+d_1+d_2/i)=p_0(d_0/i)p_1(d_1/i)p_2(d_2/i). \tag{B.8}$$

APPENDIX C

Derivation of $P_d(d_B)$ and $P_d(d_G)$ for the Pairwise Error Probability

Recall that we want to find the probability of being in state B exactly d_B out of d times, $P_d(d_B)$, and the probability of being in state G exactly d_G out of d times, $P_d(d_G)$.

Consider $P_d(d_B)$. For the cases $d = 0$ and $d = n$, the result is trivial, since $d = 0$ means that the channel starts in the good state and never leaves it, which will happen with probability $(1-b)^{d-1}\pi_G$, and $d = n$ means that the channel starts in the bad state and remains there, which will happen with probability $(1-g)^{d-1}\pi_B$. Henceforth, we may therefore assume that $1 \le d \le n$.

The channel behavior can be depicted by one of the following four cases:

- The channel starts in the good state and ends in the good state, which will happen with probability $P_d(d_B/GG)$.
- The channel starts in the good state and ends in the bad state, which will happen with probability $P_d(d_B/GB)$.
- The channel starts in the bad state and ends in the bad state, which will happen with probability $P_d(d_B/BB)$.
- The channel starts in the bad state and ends in the good state, which will happen with probability $P_d(d_B/BG)$.

$P_d(d_B)$ is found by summing the conditional probabilities, weighted appropriately, i.e.

$$P_d(d_B) = [P_d(d_B/GG) + P_d(d_B/GB)]\pi_G + [P_d(d_B/BG) + P_d(d_B/BB)]\pi_B \quad \text{(C.1)}$$

Now, consider $P_d(d_B/GG)$, and let i be the number of sojourns in the good state. The number of sojourns in the bad state is then i-1. If the channel is in the bad state exactly d_B

out of d times, then it is in the good state exactly d-d_B. Clearly $i \leq d - d_B$, otherwise the channel would be in the good state too many times, and $i - 1 \leq d_B$, otherwise the channel would be in the bad state too many times. The probability, apart from d_B and d, depends on the number of times the channel state changes, not on the exact behavior of the channel. In this case, we will have:

- i-1 transitions from a good state to a bad state, each of which will happen with probability b.
- d-d_B-i transitions from a good state to a good state, each of which will happen with probability 1-b.
- i-1 transitions from a bad state to a good state, each of which will happen with probability g.
- d_B-i+1 transitions from a bad state to a bad state, each of which will happen with probability 1-g.

The probability for this specific channel behavior is given by

$$(1-b)^{d-d_B-i} b^{i-1} (1-g)^{d_B-i+1} g^{i-1} \tag{C.2}$$

Now, since the number of ways d_B can be expressed as a sum of i-1 positive intergers is

$$\binom{d_B - 1}{i - 2}, \tag{C.3}$$

and the number of ways that d-d_B can be expressed as a sum of i positive integers is

$$\binom{d - d_B - 1}{i - 1}, \tag{C.4}$$

we have that

$$P_d(d_B/GG) = \sum_{i=2}^{\min(d_B+1,\, d-d_B)} \binom{d - d_B - 1}{i - 1} \binom{d_B - 1}{i - 2} (1-b)^{d-d_B-i} b^{i-1} (1-g)^{d_B-i+1} g^{i-1} \quad . \tag{C.5}$$

By using similar arguments, it is straightforward to derive the other conditional probabilities, and hence $P_d(d_B)$, which concludes the proof.

$P_d(d_G)$ is also derived similarly.

[1] C. E. Shannon, "A mathematical theory of communication," *Bell Sys. Tech. J.,* vol. 27, pp. 379-423 and 623-656, 1948.

[2] R. W. Hamming, "Error detecting and correcting codes," *Bell Sys. Tech. J.,* vol. 29, pp. 147-160, 1950.

[3] M. J. E. Golay, "Notes on digital coding," *Proc. IEEE*, vol. 37, p. 657, 1949.

[4] R. W. Lucky, *Silicon Dreams: Information, Man, and Machine*. New York, NY: St. Martin's Press, 1989.

[5] S. Wicker, *Error Control Systems for Digital Communications and Storage*. Englewoods cliffs, NJ: Prentice Hall, Inc., 1995.

[6] E. Berruto, M. Gudmundson, R. Menolascino, W. Mohr, and M. Pizarroso, "Research activities on UMTS radio interface, network architectures, and planning," *IEEE Commun. Magazine*, vol. 36, pp. 82-95, Feb. 1998.

[7] D. E. Muller, "Application of Boolean algebra to switching circuit design," *IEEE Trans. On Computers*, vol.3, pp. 6-12, Sept. 1954.

[8] I. S. Reed, "A class of multiple-error-correcting codes and decoding structure," *IEEE trans. Inform. Theory*, vol. 4, pp. 38-49, Sept. 1954.

[9] E. Prange, "Cyclic error-correcting codes in two symbols," Tech. Rep. TN-57-103, Air Force Cambridge Research Center, Cambridge, MA, Sept. 1957.

[10] J. E. Meggitt, "Error correcting codes and their implementation," *IRE Trans. Inform. Theory*, vol. 7, pp. 232-244, Oct. 1961.

[11] A. Hocquenghem, "Codes correcteurs d'erreurs," *Chiffres*, vol. 2, pp. 147-156, 1959.

[12] R. C. Bose and D. K. Ray-Chaudhuri, "On a class of error correcting binary group codes," *Information and Control*, vol. 3, pp. 68-79, Mar. 1960.

[13] I. S. Reed and G. Solomon, "Polynomial codes over certain fields," *SIAM Journal on Applied Mathematics*, vol. 8, pp. 300-304, 1960.

[14] E. R. Berlekamp, "Nonbinary BCH decoding," in *IEEE Int. Symp. On Inform. Theory*, (san Remo, Italy), 1967.

[15] E. R. Berlekamp, R. E. Peil, and S. P. Pope, "The application of error control to communications," *IEEE Commun. Magazine*, vol. 25, pp. 44-57, Apr. 1987.

[16] S. Lin, T. Kasami, T. Fujiwara, and M. Fossorier, *Trellises and trellis-based decoding algorithms for linear block codes*. Kluwer Academic Publishers, 1998.

[17] P. Elias, "Coding for noisy channels," *IRE Conv. Record*, vol. 4, pp. 37-47, 1955.

[18] J. M. Wozencraft and B. Reiffen, *Sequential Decoding*. Cambridge, MA: MIT Press, 1961.

[19] R. M. Fano, "A heuristic discussion of probability decoding," *IEEE Trans. Inform. Theory*, vol. 9, pp. 64-74, Apr. 1963.

[20] F. Jelinek, "An upper bound on moments of sequential decoding effort," *IEEE trans. Inform. Theory*, vol. 15, pp. 464-468, July 1969.

[21] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inform. Theory*, vol. 13, pp. 260-269, Apr. 1967.

[22] J. P. Odenwalder, *Error Control Coding Handbook*. Linkabit Corporation, 1976.

[23] D. J. Costello, J. Hagenauer, H. Imai, and S. B. Wicker, "Application of error-control coding," *IEEE Trans. Inform. Theory*, vol. 44, pp. 2531-2560, Oct. 1998.

[24] J. L. Ramsey, "Realization of optimum interleavers," *IEEE Trans. Inform. Theory*, vol. 16, pp. 338-345, May 1970.

[25] G. D. Forney, *Concatenated Codes*. Cambridge, MA: MIT Press, 1966.

[26] G. Ungerboeck and I. Csajka, "On improving data link performance by increasing the channel alphabet and introducing sequence coding," in *Proc., IEEE Int. Symp. On Inform. Theory*, (Ronneby, Sweden), June 1976.

[27] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes(1)," in *Proc. ICC'93*, Genava, Switzerland, May 1993, pp. 1064-1070.

[28] J. Lodge, R. young, P. Hoeher, and J. Hagenauer, "Separable MAP 'filters' for the decoding of product and concatenated codes," in *Proc., IEEE Int. Conf. On Commun.*, (Geneva, Switzerland), pp. 1740-1745, 1993.

[29] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo-codes," *IEEE Trans. Commun.*, vol. 44, pp. 1261-1271, Oct. 1996.

[30] IEEE Information Theory Society, "1997 Information Theory Society Paper Award," *IEEE Trans. Inform. Theory*, vol. 44, p. 445, Mar. 1998.

[31] P. Robertson, "Improving decoder and code structure of parallel concatenated recursive systematic (turbo) codes," in *Proc., IEEE Int. Conf. On Universal Personal Commun.*, pp. 183-187, 1994.

[32] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," in *Proc. Int. Conf. Communications*, June 1995, pp. 1009-1013.

[33] P. Robertson, "Illuminating the structure of parallel concatenated recursive systematic (turbo) codes," in *Proc., IEEE GLOBECOM'94*, San Francisco, CA, pp. 1298-1303, 1994.

[34] J. Hagenauer, P. Robertson, and L. Papke, "Iterative (turbo) decoding of systematic convolutional codes with the MAP and SOVA algorithms," in *Proc., ITG Conf. Sept.* 1994.

[35] P. Jung and M. Naβhan, "Performance evaluation of turbo codes for short frame transmission systems," *Electronics Letters*, vol. 30, pp. 111-113, Jan. 20th 1994.

[36] S. Benedetto and G. Montorsi, "Serial concatenation of block and convolutional codes," *Electronics Letters*, vol. 32, pp. 887-888, May 9th 1996.

[37] C. Berrou, "Some clinical aspects of turbo codes," in Proc. *Int. Symp. Turbo Codes and Related Topics,* Brest, France, Sept. 1997, pp. 26-31.

[38] J. F. Cheng and R. J. McEliece, "Unit-memory Hamming turbo codes," in *Proc., IEEE Int. Symp. On Inform. Theory*, p. 33, 1995.

[39] R. Pyndiah, A. Glavieux, A. Picart, and S. Jacq, "Near optimum decoding of product codes," in *Proc., IEEE GLOBECOM*, pp. 339-343, 1994.

[40] O. Aitsab and R. Pyndiah, "Performance of Reed-Solomon block turbo code," in *Proc., IEEE GLOBECOM*, (London, UK), pp. 121-125, Nov. 1996.

[41] C. Sagan, *Pale Blue Dot*. New York, NY: Random House, 1994.

[42] J. G. Proakis, *Digital Communications*, McGraw-Hill, Inc., New York, Third edition, 1995.

[43] O. M. Collins, "The subtleties and intracies of building a constraint length 15 convolutional decoder," *IEEE Trans. Commun.*, vol. 40, pp. 1810-1819, Dec. 1992.

[44] L. C. Perez, "Turbo codes," in *Trellis Coding* (C. Schelgel, ed.), ch. 8, pp. 233-262, New York, NY: IEEE Press, 1997.

[45]: S. Lin and D. Costello, *Error control coding: Fundamentals and Applications,* Prentice Hall, 1983.

[46] S. Benedetto and G. Montorsi, "Unveilling turbo-codes: Some results on parallel concatenated coding schemes," *IEEE Trans. Inform. Theory*, vol. 42, no. 2, Mar. 1996, pp. 409-428.

[47] D. Divsalar and F. Pollara, "Turbo codes for PCS applications," in *Proc. ICC'95*, Seattle, WA, June 1995, pp. 54-59.

[48] A. S. Barbulescu and S. S. Pietrobon, "Interleaver design for turbo codes," *Electron. Lett.*, vol. 30, no. 25, Dec. 1994, p. 2107.

[49] A. S. Barbulescu and S. S. Pietrobon, "Terminating the trellis of turbo-codes in the same state," *Electron. Lett.*, vol. 31, no. 1, Jan. 1995, pp. 22-23.

[50] A. S. Barbulescu and S. S. Pietrobon, "Interleaver design for three dimensional turbo codes," in *Proc. 1995 IEEE ISIT*, Whistler, BC, Canada, sep. 1995, p.37.

[51] O. Joerssen and H. Meyr, "Terminating the trellis of turbo-codes," *Electron. Lett.*, vol. 30, no. 16, Aug. 1994, pp. 1285-1286.

[52] D. Divsalar and F. Pollara, "On the design of turbo codes," *TDA Progress Report* 42-123, Jet Propulsion Lab., Nov. 1995, pp. 99-121.

[53] S. Benedetto, R. Garello, and G. Montorsi, "A search for good convolutional codes to be used in the construction of turbo codes," *IEEE Trans. Commun.*, vol. 46, no. 9, Sep. 1998, pp. 1101-1105.

[54] L. R. Bahl, J. Cocke, F. Jelinek and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. 20, pp. 284-287, Mar. 1974.

[55] J. Hagenauer and P. Hoeher, "A Viterbi algorithm with soft-decision outputs and its applications," *IEEE Globecom*, pp. 1680-1686, 1989.

[56] W. E. Ryan, "A turbo code tutorial," *New Mexico state University*, Las cruces, unpublished paper, available at http://WWW.ece.arizona.edu/~Wryan/.

[57] B. Sklar, "A primer on turbo code concepts," *IEEE Commun. Mag.*, pp. 94-102, Dec. 1997.

[58] M. Breiling and L. Hanzo, "The super-trellis structure of turbo codes," *IEEE Trans. Inform. Theory*, vol. 46, no. 6, Sept. 2000, pp. 2212-2228.

[59] M. Breiling and L. Hanzo, "Optimum noniterative turbo-decoding," in *Proc. PIMRC'97*, Helsinki, Findland, Sept. 1997, pp. 714-718.

[60] H. Nick, J. Hagenauer, and F. Burkett, "Approaching Shannon's capacity limit by 0.27 dB using simple Hamming codes," *IEEE Commun. Lett.*, vol. 1, pp. 130-132, Sept. 1997.

[61] W. Koch and A. Baier, "Optimum and sub-optimum detection of coded data disturbed by time-varying inter-symbol interference," *IEEE Globecom*, pp. 1679-1684, Dec. 1990.

[62] J. A. Erfanian, S. Pasupathy, and G. Gulak, " Reduced complexity symbol detectors with parallel structures for ISI channels," *IEEE Trans. Commun.*, vol. 42, pp. 1661-1671, 1994.

[63] C. Berrou, P. Adde, E. Angui. And S. Faudeil, "A low complexity soft-output Viterbi decoder architecture," in *Proc. Int. Conf. Communications*, May 1993, pp. 737-740.

[64] S. Dolinar, D. Divsalar, and F. Pollara, "Code Performance as a Function of Block Size," SPL, NASA. [online]. Available: http://tmo.jpl.nasa.gov/tmo/progress_report/42-133/title.htm.

[65] P. Jung and M. Naßhan, "Dependence of the error performance of turbo-codes on the interleaver structure in short frame transmission systems," *IEE Electron. Lett.*, pp. 287-288, Feb. 1994.

[66] H. Herzberg, "Multilevel turbo coding with short interleavers," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 303-309, Feb. 1998.

[67] B. Vucetic and J. Yuan, *Turbo Codes Principles and Applications,* Kluver Academic Publishers: Boston, Dordrecht, London, 2000.

[68] J. P. Woodard and L. Hanzo, "Comparative Study of Turbo Decoding Techniques: An Overview," *IEEE Trans. Veh. Technol.*, vol. 49, No. 6, Nov. 2000.

[69] D. Divsalar and F. Pollara, "Multiple turbo codes," in *Proc. IEEE MILCOM*, pp. 279-285, Nov. 1995.

[70] S. Benedetto, G. Montorsi, D. Divsalar, and F. Pollara, "Serial Concatenation of Interleaved Codes: Performance Analysis, Design, and Iterative Decoding," *JPL TDA Progress report*, vol. 42, Aug. 1996.

[71] S. Benedetto and G. Montorsi, "Generalized concatenated code with interleavers," in *Proc. Int. Symp. On Turbo Codes and related Topics*, (Brest, France), pp. 32-39, Sept. 1997.

[72] F. R. Kschischang and B. J. Frey, "Iterative decoding of compound codes by probability propagation in graphical models" IEEE J. Select. Areas Commun., vol. 16, pp. 219-230, Feb. 1998.

[73] C. D. Edwards, C.T. Steilzvield, L. J. Deutsch and L. Swanson, "NASA's deep-space telecommunications road MAP," TMO Progess Report 42-126, Feb. 1999.

[74] D. Divsalar and F. Pollara, "Turbo codes for deep-space communications," TDA Progress report, 42-120, Feb. 1995.

[75] D. Divsalar and F. Pollara, "Multiple turbo codes for deep-space communications," TDA Progress Report, 42-121, May. 1995.

[76] The cdma2000 ITU-R RTT Candidate Submission (0.18), Jul. 1998.

[77] 3rd Generation Partnerhip Project (3GPP) Technical Specification group: Radio Access Networking Group 1, Multiplexing and Channel Coding, TS 25. 212 v1. 1. 0, June 1999.

[78] H. Feldman and D. V. Ramana, "An introduction to Inmarsat's new mobile multimedia service," in *Proc. Sixth Int. mobile satellite Conf.*, Ottawa, June 1999, pp. 226-229.

[79] E. Trachman and T. Hart, "Research elements leading to the development of Inmarsat's new mobile multimedia services," in *Proc. Sixth Int. Mobile satellite Conf.*, Ottawa, June 1999, pp. 209-212.

[80] A. Baïer, U. C. Fielbüg, W. Granzow, P. Teder, and J. Thielecke, "Design study for a CDMA-based third generation mobile radio system," *IEEE J. Select. Areas Commun.*, vol. 2, no. 4, pp. 773-743, May 1994.

[81] A. Viterbi, "When not to spread spectrum— A sequel," *IEEE Commun. Mag.*, vol. 23, pp. 12-17, Apr. 1985.

[82] V. Kühn, "Evaluating the Performance of Turbo Codes and Turbo-Coded Modulation in a DS-CDMA Environment," *IEEE J. Select. Areas Commun.*, vol. 17, No. 12, pp. 2138-2147, Dec. 1999.

[83] K. Tang, L. B. Milstein, and P. H. Siegel, "Combined MMSE Interference Suppression and Turbo Coding for a Coherent DS-CDMA System." *IEEE J. Select. Areas Commun.*, vol. 19, no. 9, pp. 1793-1803, Sept. 2001.

[84] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 429-445, Mar. 1996.

[85] M. Moher, "Decoding via cross entropy minimization," in Proc. *IEEE Globecom Conf.*, Houston, TX, Dec. 1993, pp. 809-813.

[86] R. Y. Shao, S. Lin, and M. P. C. Fossorier, "Two simple stopping criteria for turbo decoding," *IEEE Trans. Commun.*, vol. 47, pp. 1117-1120, Aug. 1999.

[87] Yufei Wu, B. D. Woerner and W. J. Ebel, "A simple stopping criterion for turbo decoding" *IEEE Communications Letters*, vol. 4, No. 8, pp. 258-260, August 2000.

[88] R. H. Clarke, "A statistical theory of Mobile radio reception," *Bell Syst. Tech.* J., 47, pp. 957-1000, 1968.

[89] E. N. Gilbert, "Energy reception for mobile radio," *Bell Syst. Tech. J.*, 44, pp. 1779-1803, 1965.

[90] W. C Jakes, *Microwave mobile communications*, New York: Willey, 1974.

[91] V. H. MacDonald, "Advanced mobile phone service: The cellular concept," *Bell Syst. Techn. J.*, 58(1), pp 15-42, 1979.

[92] A. Papoulis, *Probability, Random variables and Stochastic Processes*, McGraw-Hill, Inc., New York, Third edition, 1991.

[93] S. O. Rice, "Mathematical analysis of random noise," *Bell Syst. Tech. J.*, 23, pp-292-332, 1944.

[94] M. Abranowitz and I.A. Stegun, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Dover publications, New York, 9th edition, 1972.

[95] W. B. Davenport Jr. and W. L. Root, *An Introduction to the theory of random signals and noise*, IEEE Press, New York, 1958.

[96] M. Nakagami, "The *m*-distribution: a general formula of intensity distribution of rapid fading," in W. C. Hoffman, editor, *Statistical Methods in Radio wave propagation*, pp. 3-36, Pergamon Press, New York, 1960.

[97] W. R. Brawn and U. Dersch, "A physical mobile radio channel model," *IEEE Trans. Veh. Technol.*, vol. 40, No. 2, pp. 472-482, 1991.

[98] J. I. Smith, "A computer generated multipath fading simulation for mobile radio," IEEE Trans. Commun, vol. COM-22, pp. 1607-1617, October 1974.

[99] D. J. Young and N. C. Beaulieu, "The generation of correlated Rayleigh random variates by inverse discrete Fourier transform," IEEE Trans. Commun, vol. 48, No. 7, pp. 1114-1127, July 2000.

[100] C. Loo and N. Secord, "Computer models for fading channels with applications to digital transmission," IEEE. Trans. Veh. Technol., vol VT-40, no. 4, pp. 700-707, Nov. 1991.

[101] P. McLane, "Two-stage Doppler-phasor-corrected TCM/DMPSK for shadowed mobile satellite channels," IEEE. Trans. Commun, vol. 41, no. 8, pp. 1137-1141, Aug. 1993.

[102] D. Verdin and T. Tozer, "Generating a fading process for the simulation of land-mobile communications," Electronics Letters, vol. 29, no. 23, pp. 2012-2012, 11 Nov. 1993.

[103] P. Hoeher, " A statistical discrete-time model for the WSSUS multipath channel," IEEE. Veh. Technol., vol. VT-41, no. 4, pp. 461-468, Nov. 1992.

[104] D. J. Young, "The generation of correlated Rayleigh random variates by discrete Fourier transform and quality measures for random variate generation," M. S. thesis, Queen's University, Kingston, Canada, 1997.

[105] D. J. Young and N. Beaulieu, "A Quantitative Evaluation of Generation Methods for Correlated Rayleigh Random Variates," *Globecom 1998*, Sydney Australia, Conference Record 8-12 November, Vol.6, pp. 3332-3337.

[106] N. C Beaulieu and C. C. Tan "Generating Bandlimited Gaussian Noise Variates with the FFT," In *proceeding of the IEEE pacific Rim conference on communication*, computers and signal processing, pp. 28-35, Victoria, British Columbia, Canada, August 1997.

[107] N. C Beaulieu and C. C. Tan "An FFT method for generating bandlimited Gaussian noise variates," *In proceeding of the IEEE Globcl communication conference* (GLOBECOM) pp 684-688, phoenix, Arizona, U.S.A, November 1997.

[108] A. V. Oppenheim and R. S. Schaffer, dicrete-Time Signal Processing, prentice Hall, 1989.

[109] P. Robertson and S. Kaiser, " The effects of Doppler spreads in OFDMA(A) mobile radio systems," in prof. VTC'99-Fall, 1999, pp. 329-333.

[110] Y. G. Li and L. J. Cimini, " Bounds on the interchannel interference of OFDM in time-varying impairments" IEEE Trans. Commun. Vol. 49, No. 3, march 2001.

[111] I. S. Gradshteyn and I. M. Ryzhilk, Table of integrals, series, and products, Academic press, 5th edition, 1994.

[112] E. Hall and S.G. Wilson, " Design and Analysis of Turbo codes on Rayleigh Fading Channels," *IEEE J. Select. Areas Commun*, vol. JSAC-16 no.2, pp.160-174, February 1998.

[113] P. Komulainen and K. Pehkonen, "Performance Evaluation of Superorthogonal Turbo Codes in AWGN and Flat Rayleigh Fading Channels," *IEEE J. Select. Areas Commun,* vol. JSAC-16 no.2, pp. 196-205, February 1998.

[114] T. Eng and L. B. Milstein, " Coherent DS-CDMA Performance in Nakami multipath fading," *IEEE Trans. commun.,* vol.43, nos. 2-4, pp. 1134-1143, 1995.

[115] Georges P.Efthymoglou, Valentine A. Aalo, and Henry Helmken, "Performance Analysis of Coherent DS-CDMA Systems in Nkagami Fading Channel with Arbitrary Parameters," *IEEE Trans. Veh. Technol.*, vol.46, No.2 May 1997.

[116] J. M. Holtzman, "A simple, accurate method to calculate spread-spectrum multiple-access error probabilities," *IEEE Trans. Commun.*, vol. 40, no. 3, pp. 461-464, 1992.

[117] M. Pursley, "Performance evaluation for phase coded spread system multiple access communication—part I: system analysis," *IEEE Trans. Commun.,* vol. COMM-25, no. 8, pp. 795-799, 1977.

[118] G. Turin, F. Clapp, T. Johnston, S. Fine, nad D. Lavry, "A statistical model of urban multipath propagation," *IEEE Trans. Veh. Technol.*, vol. VT-21, no. 1, pp. 1-9, 1972.

[119] H. Xiang, "Binary code-division multiple-access systems operating in multipath fading, noisy channel," IEEE Trans. Commun., vol. COMM-33, pp. 775-784, Aug. 1985.

[120] M. V. Clark, L. J. Greenstein, W. K. Kennedy, and M. Shafi, "Matched filter performance bounds for diversity combining receivers in digital mobile radio," *IEEE Trans. Veh. Technol.*, vol.41, no. 4, pp. 356-362, 1992.

[121] E. N. Gilbert, "Capacity of a burst noise channel," *Bell Syst. Tech. J.*, vol. 39, pp. 1253-1266, Sept. 1960.

[122] E. O. Elliott, "Estimates of error rates for codes on burst-noise channels," *Bell Syst. Tech. J.*, vol. 42, pp. 1977-1997, Sept. 1963.

[123] H. S. Wang and N. Moayeri, "Finite-state Markov channel—A useful model model for radio communication channels," *IEEE Trans. Veh. Technol.*, vol. 44, pp. 163-171, Feb. 1995.

[124] H. S. Wang and P.C. Chang, "On verifying the first-order Markovian assumption for a Rayleigh fading channel model," *IEEE Trans. Veh. Technol.*, vol. 45, pp. 353-357, May 1996.

[125] L. Ahlin, "Coding methods for the mobile radio channel," presented at Nordic Seminar on Digital Land Mobile Communications, Espoo, Finland, Feb. 1985.

[126] R. Krishnamurthi, An Analytical Study of Block Codes in a Portable Digital Cellular System, Ph.D. Thesis, SMU, 1990.

[127] G. Sharma, A. Dholakia, and A. A. Hassan, "Simulation of error trapping decoders on a fading channel," in Proc. 1996 IEEE Vehicular Technology Conf., Atlanta, GA, Apr. 28-May 1, 1996, pp. 1361-1365.

[128] Leif Wilhelsson and Laurence B. Milstein, "On the effect of imperfect interleaving for the Gilbert-Elliot channel," *IEEE Trans. Commun.* Vol. 47, No. 5, pp. 681-688 May 1999.

[129] J. R. Yee and E. J. Weldon, " Evaluation of the Performance of Error-Correcting Codes on a Gilbert Channel," *IEEE Trans. Commun.* Vol. 43, No. 8, August 1995.

[130] G. Sharma, A. A. Hassan and A. Dholakia, " Performance Evaluation of Burst-Error-Correcting Codes on a Gilbert_Elliott Channel," IEEE Trans. Comm. Vol. 46, No. 7, JULY 1998.

[131] J. H. Kang, W. E. Stark and A. O. Hero, " Turbo codes for Fading and Burst Channels," Proceedings of the Communications Theory Miniconference, Globecom'99, pp. 40-45, November 1998.

[132] J. Garcia-Frias and J. D. Villasenor, " Turbo Codes for Continous Hidden Markov Channels With Unknown Parameters" Proceedings of the Global Telecommunications Conference, Globecom'99, pp. 2363-2368, 5-9 December 1999, Rio de Janeiro, Brazil.

[133] M.B. Pursley and H. E. A. Roefs, " Numerical evaluation of correlation parameters for optimal phases of binary shift-register sequences," *IEEE Trans. Commun.*, vol. COM-27, pp. 1597-1604, Oct. 1979.

[134] M. B. Pursley, D. V. Sarwate, and W. Stark, "Error probability for direct-sequence spread-spectrum multiple-access communications—Part I: Upper and lower bounds," *IEEE Trans. Commun.*, vol. COM-30, pp. 975-984, May 1982.

[135] H. J. Larson and B. O. Shuber, *Probabilistic Models in Engineering Sciences*, vol II John Wiley&Sons, New York Chichester Brisbane Toronto.

[136] D. Divsalar, S. Dollinar, R. J. Mc Eliece and F. Pollara, "Transfert function bounds on the performance of turbo codes," *TDA Progr.* Rep. 42-121, JPL, Cal Tech, Aug. 1995.

[137] T.S Rappaport, *Wireless Communications: Principles and Practice*. Upper Saddle River, NJ: Prentice Hall PTR, 1996.

[138] T. M. N. Ngatched and F. Takawira, "Comparison of stopping criteria for turbo decoding," in *Proc. South African Telecommunication Networks & Applications Conference*, 2-5 Sept. 2001, Wild Coast Sun, South Africa.

[139] T. M. N. Ngatched and F. Takawira, "A simple stopping criterion for turbo decoding," *IEE Electronic Letters*, vol. 37, no. 22, pp. 1350-1351, Oct. 25, 2001.

[140] T. M. N. Ngatched and F. Takawira, "Performance analysis of turbo-coded DS-CDMA system on a Gilbert-Elliott channel," *IEEE Trans. Veh. Technol.*, submitted.

[141] T. M. N. Ngatched and F. Takawira, "Effect of mobile Doppler frequency on the performance of coherent turbo-coded DS-CDMA over frequency selective multi-path Rayleigh fading channel," in *Proc. SATCAM*, 10-13 Sept. 2000, Cape Town, South Africa.

[142] R. Stanley, *Enumerative Combinatorics*, vol. 1, Wadsworth & Brooks/Code, Monterey, California, 1986.