REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	14.Jan.03	THESIS

4. TITLE AND SUBTITLE
A DENSITY EVOLUTION ANALYSIS OF TURBO PRODUCT CODES

5. FUNDING NUMBERS

6. AUTHOR(S)
2D LT DURHAM LAURA M

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

8. PERFORMING ORGANIZATION REPORT NUMBER
CI02-802

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
THE DEPARTMENT OF THE AIR FORCE
AFIT/CIA, BLDG 125
2950 P STREET
WPAFB OH 45433

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
Unlimited distribution
In Accordance With AFI 35-205/AFIT Sup 1

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

DISTRIBUTION STATEMENT A
Approved for Public Release
Distribution Unlimited

20030220 129

14. SUBJECT TERMS	15. NUMBER OF PAGES
	113
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT

THE VIEWS EXPRESSED IN THIS ARTICLE ARE THOSE OF THE AUTHOR AND DO NOT REFLECT THE OFFICIAL POLICY OR POSITION OF THE UNITED STATES, DEPARTMENT OF DEFENSE, OR THE U.S. GOVERNMENT

A Density Evolution Analysis of Turbo Product Codes*

by

Laura M. Durham

B.S., Electrical Engineering
United States Air Force Academy, 2000

Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of Master of Science in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2002

Author ..... Laura M. Durham
Department of Electrical Engineering and Computer Science
May 22, 2002

Certified by ..... Lori L. Jeromin
Lori L. Jeromin
Technical Staff, MIT Lincoln Laboratory
Thesis Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

*This work was sponsored by the Department of the Air Force under Contract F19682-00-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the author and not necessarily endorsed by the United States Air Force.

A Density Evolution Analysis of Turbo Product Codes*

by

Laura M. Durham

Submitted to the Department of Electrical Engineering and Computer Science
on May 22, 2002, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering and Computer Science

Abstract

Turbo product codes (TPC) are a promising approach for power-efficient communications, particularly in satellite and terrestrial wireless systems. These codes use an iterative decoding method similar to turbo codes. TPCs have been shown to have a bit error rate (BER) performance within a couple of dB of turbo codes without the error floor, however other performance measures of turbo product codes are not well developed. This thesis applies the Extrinsic Information Transfer (EXIT) chart analysis, developed for turbo codes, to turbo product codes. The EXIT chart analysis allows for examination of the evolution of the probability densities of the information passed from iteration to iteration of the decoder. The analysis begins with the EXIT chart analysis for two-dimensional TPCs, similar to the turbo code results, and then extends the analysis to three-dimensional TPCs. Binary phase-shift keying (BPSK) and Gaussian minimum-shift keying (GMSK) modulations are examined in both an unfaded additive white Gaussian noise (AWGN) as well as Rayleigh faded channel. In addition, BER results are predicted in the low E_b/N_o region, convergence thresholds determined, and lastly a new code construction for a rate 1/2 TPC is designed.

Thesis Supervisor: Lori L. Jeromin
Title: Technical Staff, MIT Lincoln Laboratory

List of Figures

List of Tables

Chapter 1

Introduction

In digital communications, several aspects must be considered when designing a forward error correcting code within a defined system. Some of these aspects include the code's block size and rate, the decoder's complexity of implementation, and finally the performance of that code. In general, we wish to maximize the performance and data rate of the code, while minimizing the complexity and size of the code. The above factors are especially true in the area of satellite communications where onboard resources, such as power, weight/mass, spectrum allocation, and bandwidth may also be limited. Above all, the communication system must be reliable. Our ability to both accurately and quickly transmit large amounts of information over some distance determines the reliability of the communication system. We use forward error correction (FEC) in satellite communications to improve our performance, and thus increase the reliability of our communication system.

There has been a large focus on finding error correcting codes to meet our communication needs. During the 1990s, concatenated convolutional codes with iterative decoding, or turbo codes, were found to perform well. However, a high degree of complexity is needed to decode these codes [4]. In recent years, turbo product codes (TPC) have also performed well without the high complexity of decoding required of the turbo codes [7]. To meet the goals of good performance, low complexity, and increasing reliability we want to use the best code for our defined system.

However, how do we determine the best code to use? Richardson and Urbanke [16]

proposed tracking the evolution of the density of the information messages through the decoding process. In [17] and [18], ten Brink extended the ideas of density evolution to the Extrinsic Information Transfer (EXIT) chart. The EXIT chart examines the evolution of the mutual information between the extrinsic information and the source bits from one iteration to the next. He showed that the EXIT chart could not only be used to analyze the decoding process, but could also be used as a tool aiding the search to determine good codes for a given system.

1.1 Channel Capacity

In addition to meeting the power, space, and bandwidth requirements of a given communication system, one of the primary goals of coding is to maximize our use of the communication channel. In general, the channel capacity C is

$$C = \max_{p(x)} I(X;Y) \tag{1.1}$$

where $I(X;Y)$ gives the information gain between the transmitted signal X and the received signal Y [12]. This information gain is the mutual information between X and Y and is defined as

$$I(X;Y) = \sum_X \sum_Y p(x,y) \log_r \left[\frac{p(x,y)}{p(x)p(y)} \right] \tag{1.2}$$

where $r = 2$ for a binary system, $p(x,y)$ is the probability of the x and y occurring jointly, and $p(x)$ and $p(y)$ are the probabilities of x and y occurring individually. The channel capacity C gives the maximum amount of information for a given channel [12]. The capacity of the channel is further defined to be the maximum rate at which error-free communication exists, and is often referred to as the Shannon Limit. The maximum rate is usually measured in bits per second.

By placing constraints on the average signal power and using the additive white

Gaussian noise (AWGN) channel,

$$\frac{C}{W} = \log_2\left(1 + \frac{R}{W}\frac{E_b}{N_o}\right) \text{bits/sec/Hz.} \tag{1.3}$$

The channel capacity C is a function of the bandwidth W and the information bit energy to noise density ratio E_b/N_o [14]. By setting $C = R$, where R is the code rate, and using Equation 1.3, we plot the Shannon limit as a function of E_b/N_o in Figure 1-1. The ratio R/W of code rate to bandwidth is the bandwidth efficiency, which is upper bounded by the Shannon Limit [14].

In Figure 1-1, as you move to the left on the x-axis with reduced E_b/N_o, the code is more power efficient. The code is more bandwidth efficient as you move up along the y-axis. We desire a code that is not only power efficient, but bandwidth efficient as well. The desired performance area lies just below the Shannon Limit curve in the upper left section of the plot. This area is highlighted in the figure.

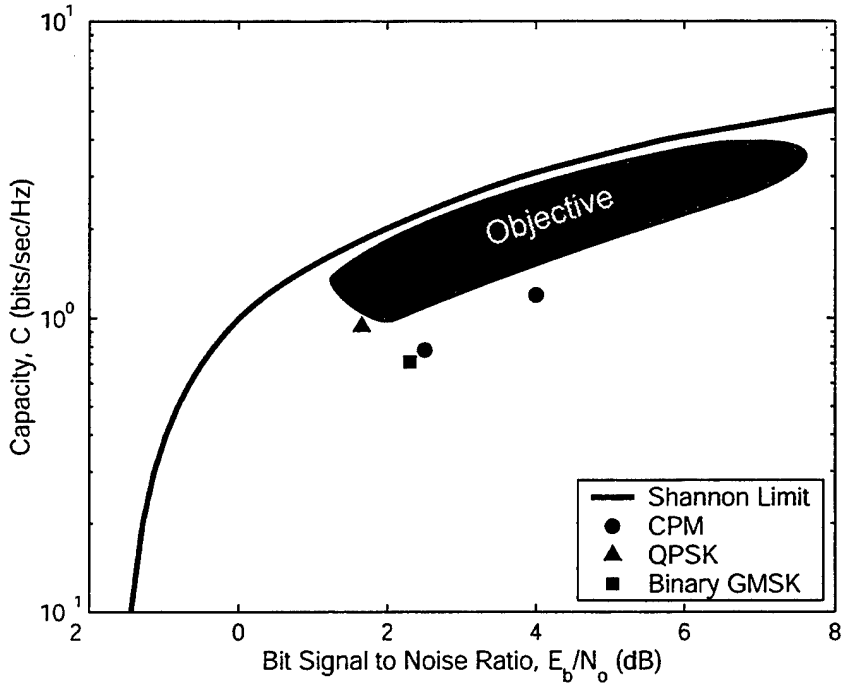


Figure 1-1: Comparison of Some Current Satcom Techniques with Shannon Capacity

Also shown are several codes, with associated modulation schemes, being consid-

ered for use in satellite communications. These codes include rate 1/2 and 2/3 turbo codes with quaternary phase-shift keying (QPSK) modulation and continuous phase modulation (CPM), and a rate 1/2 turbo code with binary Gaussian minimum-shift keying (GMSK) modulation. The E_b/N_o values are those required for a bit error rate (BER) of 10^{-5}. For our purposes, we use the E_b/N_o required to obtain a BER of 10^{-5} as our performance measure, and the smaller the E_b/N_o value, the better the performance of the code.

1.2 Objective

As shown in Figure 1-1, these codes and modulations have left room for improvement. We will examine TPCs to fill in some of those gaps. The examination will involve analyzing the TPC decoder developed by Advanced Hardware Architectures, Inc. (AHA) and Efficient Channel Coding, Inc. (ECC), with AHA the hardware developer and ECC the software developer. The EXIT chart method developed by ten Brink in [17, 18] will be used in this analysis.

There are several goals of this examination. The examination will cover the performance of the code in the AWGN environment and the Rayleigh fading environment, indicative of satellite and terrestrial wireless communication links as well. Also, we will use the results to determine if the common code combinations are in fact good codes, or which code performs best with a limited number of iterations. Finally, we will use our density evolution results to aid in the design of even better performing TPCs.

We were successful in extending the EXIT chart from turbo codes to TPCs. Also, the EXIT chart analysis was performed for both two-dimensional and three-dimensional TPC structures. This examination included binary phase-shift keying (BPSK), Gaussian minimum-shift keying (GMSK), and a fast Rayleigh faded channel as well as the AWGN channel. Finally, we verified the use of the EXIT chart as a code design tool to compare various rate one-half codes.

The remainder of the thesis is as follows. Chapter 2 will provide some background

information on relevant error correcting codes, modulation schemes, and decoding. Chapter 3 will summarize ten Brink's previous applicable work for turbo codes. The details of our application of ten Brink's methods will be given in chapter 4. Chapters 5 and 6 will cover results of the density evolution analyses in the AWGN channel and the Rayleigh fading channel, respectively. Chapter 7 will extend these analyses to multidimensional codes Chapters 8 will summarize certain results by comparing BER predictions and convergence analysis. Chapter 9 will use the EXIT Chart in a code design example. Finally, chapter 10 will give our conclusions and suggestions for possible future work.

Chapter 2

System Model

A typical communication system with forward error correction is shown in the block diagram in Figure 2-1. The system begins with an information source which supplies the message to be transmitted to the destination at the termination of the system. The message can be in any form, but for our purposes we will restrict our discussion to digital binary information sources. The information message first enters the encoder. The encoder transforms the message into an encoded sequence of code words. These code words then enter the modulator. The modulator converts the code words into a waveform, or signal, for transmission through the communication channel. The channel degrades the signal by adding noise and possible distorting the signal. Typically, the noise is AWGN, but the signal can also suffer from fading in the channel. Fading is common in terrestrial wireless and some satellite communication channels.

After passing through the channel, the received signal enters the demodulator. The demodulator performs the opposite function of the modulator by converting the signal back into code words. The code words then enter the decoder. The decoder transforms the code words back into an information message. This is the opposite function of the encoder. Finally, the received information message reaches the destination. If the received message is the same as the transmitted message, the transmission is considered a success. Otherwise, the transmission is said to be in error, with the error rate given in terms of BER.

The functions which occur prior to the channel can be grouped together as trans-

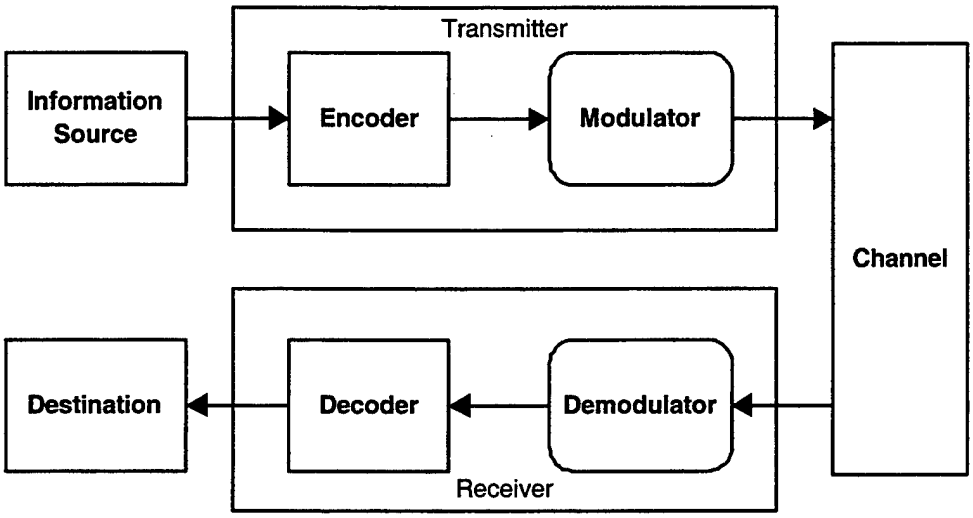


Figure 2-1: A Basic Communication System

mitter functions. Similarly, the functions which occur after the channel can be grouped together as receiver functions. The following will discuss the transmitter and receiver functions in further detail. Sections concerning the encoding, modulation and demodulation, channel statistics, and decoding will be covered.

2.1 Encoding

FEC is the concept of applying some form of redundancy, or coding, to the information we wish to transmit prior to the data being transmitted. This redundancy allows for the detection and correction of transmission errors at the receiver [13]. Error correcting codes accomplish the goals of FEC. We will consider only binary codes in our examination; that is codes composed solely of the binary elements, or bits, 0 and 1. This section will review some basic error correcting codes, including linear block codes and convolutional codes. Turbo codes and turbo product codes will be addressed as well.

2.1.1 Error Correcting Codes

The most basic class of error correcting codes is the binary block code. A binary block code consists of the set of all fixed-length vectors known as code words. The code maps information bit sequences of length k to code words of length $n > k$ such that there are 2^k code words in the code. Each code word has n total bits. In systematic codes, k bits are information bits, and the remaining $n - k$ bits are called parity bits. For an (n, k) binary block code, if all 2^k length n linear combinations of the bits are linearly independent, we say that the code is an (n, k) binary linear block code. Finally, the code rate is defined as $R = k/n$. This rate gives the number of information bits per transmitted symbol, or code word [13, 14].

A useful property of binary linear block codes is the Hamming distance of the code. The Hamming distance between two code words equals the number of corresponding positions where the two code words differ [14]. The minimum distance between any two code words in the code is denoted as d_{min}. The minimum Hamming distance of the code determines the code's error correcting capability [2]. For an (n, k) binary linear block code with minimum distance d_{min}, the code can detect up to $(d_{min} - 1)$ errors. However, the same code is capable of correcting $\lfloor \frac{1}{2}(d_{min} - 1) \rfloor$[1] errors [14].

We will concentrate our discussion on a few types of block codes: single parity check codes, Hamming codes, and extended Hamming codes. These codes are fairly simple and straightforward to use, and are the constituent codes from which the TPCs used in this analysis will be constructed.

The first type of linear block code we will consider is the single parity check (SPC) code. We define SPCs such that

$$(n, k) = (m, m - 1) \tag{2.1}$$

where m is any positive integer, however, we will restrict our attention to codes where m is a power of two. SPCs are formed by first taking $m - 1$ binary information bits. Then, the number of ones in the information bits are counted and an overall parity

[1] $\lfloor x \rfloor$ denotes the largest integer $\leq x$

bit is appended such the total number of ones in the code word is an even number. These codes have a minimum distance of two. Therefore, they can detect one error in a code word, but cannot correct any errors in a code word.

A second type of linear block code is the Hamming code. A Hamming code is defined such that

$$(n, k) = (2^m - 1, 2^m - 1 - m) \tag{2.2}$$

where m is any positive integer. These codes have a minimum distance of three and can detect up to two errors and correct one error per code word. A third type of linear block code is created by adding an overall parity bit to the Hamming code. This new code is the extended Hamming code. These codes have a minimum distance of four and can now detect up to three errors, but they can still only correct one error per code word. Figure 2-2 shows an $(8, 4)$ extended Hamming code which has four information bits I and four parity bits P.

$$I \quad I \quad I \quad I \quad P \quad P \quad P \quad P$$

Figure 2-2: (8,4) extended Hamming code

A second class of codes are convolutional codes. Like linear block codes, the (n, k) representation is used where each code word has n total bits and k information bits. However, convolutional codes have memory. For a convolutional code with memory m, the current code word depends not only upon the current k information bits, but upon the previous m information bits as well. The code rate is again $R = k/n$ [13]. Typically, k and n are smaller for convolutional codes than for block codes.

Concatenated codes are formed as the result of combining two or more, separate codes so that a larger code is formed [14]. Either the block codes or convolutional codes mentioned previously may be concatenated together. Turbo codes are one type of concatenated code where the constituent codes, which are convolutional codes, are separated by a non-uniform interleaver. These codes typically perform quite well by performing close to the Shannon Limit at low bit error rates. However, these codes

require a high level of decoding complexity [4].

In recent years, a new type of error correcting code has created much interest. These codes have also performed well, but without the same high-degree of decoding complexity as turbo codes. This class of codes is known as turbo product codes (TPC). While TPCs have recently gained interest due to the iterative decoding method introduced by Pyndiah in [15], Elias first mentioned the idea of such a block product error correcting code in 1954 [8].

2.1.2 Product Code Encoding

A TPC is a multidimensional array composed of linear block codes along each dimension, usually resulting in a two- or three-dimensional code. A two-dimensional code is encoded by using one linear block code along a horizontal axis and another along a vertical axis [4]. When encoding higher dimensional codes, the same process is used except additional axes are added until the higher dimension is reached. The constituent codes are usually comprised of extended Hamming codes or single parity check codes. The use of such codes helps to keep the encoding and decoding complexity low [7, 2].

As an example, we will describe the encoding of the $(8,4)^2$ two-dimensional TPC. Since the two constituent codes have the same length, the resulting TPC will be square. The (8,4) code is an extended Hamming code that has four information bits and four parity bits (see Figure 2-2). The TPC is encoded by building first along one axis and then along the second. The resulting TPC is an 8×8 matrix, shown in Figure 2-3, with I being an information bit and P being a parity bit. P_H are the parity bits for the horizontally constructed code words, P_V are the parity bits for the vertically constructed codewords, and P_{VH} are the parity bits of the parity bits [7, 2]. These values of P_{VH} will be the same regardless of which axis is used to computer P_{VH}. The rate of the code is $R = 1/4$.

I	I	I	I	P_H	P_H	P_H	P_H
I	I	I	I	P_H	P_H	P_H	P_H
I	I	I	I	P_H	P_H	P_H	P_H
I	I	I	I	P_H	P_H	P_H	P_H
P_V	P_V	P_V	P_V	P_{VH}	P_{VH}	P_{VH}	P_{VH}
P_V	P_V	P_V	P_V	P_{VH}	P_{VH}	P_{VH}	P_{VH}
P_V	P_V	P_V	P_V	P_{VH}	P_{VH}	P_{VH}	P_{VH}
P_V	P_V	P_V	P_V	P_{VH}	P_{VH}	P_{VH}	P_{VH}

Figure 2-3: Turbo Product Code Construction

2.2 Modulation and Demodulation

In certain systems, especially the next generation satellite systems, the bandwidth available is limited. By using an appropriate modulation scheme, we can use the allocated bandwidth more efficiently. This section will review the two forms of modulation used in our examination of TPCs. The first is BPSK and the second is GMSK. While GMSK is a bandwidth efficient modulation used in the communication systems of interest, BPSK will be used as a benchmark for performance comparisons because it is a standard modulation with straight forward analysis.

2.2.1 Binary Phase-Shift Keying

BPSK is one form of pulse amplitude modulation (PAM) and is relatively simple to implement. For an encoded code word x, the bits $\{0, 1\}$ are mapped to $\{-1, +1\}$. This modulation scheme requires a large amount of bandwidth, and as a result is not very bandwidth efficient, with a 99 percent bandwidth of 1. The transmitted signal is

$$s(t) = A\sin(\omega_o t + \phi) \tag{2.3}$$

where $\phi = \{0, \pi\}$ and $\omega_o = 2\pi f_o$ is the carrier.

The signal may be demodulated by performing matched filter detection and making a binary decision on the filter output. This process is known as hard decision demodulation. For an antipodal modulation system, if the sign of the received bit is

negative, the demodulated bit is a -1, and if the sign is positive, the demodulated bit is a $+1$ [14].

2.2.2 Gaussian Minimum-Shift Keying

GMSK is a form of continuous phase modulation (CPM) which gives the signal a continuous phase and constant envelope. The transmitted passband signal is

$$s(t) = \mathrm{Re}\{s_{bb}(t)\exp\{j2\pi f_o\}\} \tag{2.4}$$

where $s_{bb}(t)$ is the complex baseband envelope of the signal $s(t)$, and f_o is the carrier frequency. The baseband envelope is

$$s_{bb}(t) = \sqrt{\frac{2ER}{T_p}}\exp\left\{j2\pi h\sum_{i=n-L+1}^{n}\alpha_i q\left(t-\frac{iT_p}{R}\right)+j\theta_n\right\} \tag{2.5}$$

where E is the signal energy, T_p is the pulse width, and R is the fractional pulsed chipping rate, with $R=1$ for GMSK. α_i takes values from the set $\{-1,1\}$ with the phase transition

$$q(t) = \frac{1}{2}+\frac{t_1Q(\sigma t_1)-t_2Q(\sigma t_2)}{2T_p}-\frac{\exp(-\sigma^2/2)t_1^2-\exp(-\sigma^2/2)t_2^2}{2T_p\sigma\sqrt{2\pi}} \tag{2.6}$$

where B is the 3 dB bandwidth of the Gaussian filter, Q is the error function, and $\sigma = 2\pi B/\sqrt{2\pi}$ [1].

The modulation occurs when the encoded code words are sent through a pulse shaping filter, the output of which modulates the phase of the transmitted signal. An interleaver reduces the amount of inter-symbol interference present in the signal after demodulation. Demodulation may be accomplished with an application of the soft output Viterbi algorithm (SOVA). While GMSK is more complex to implement than BPSK, it has two important advantages. First, GMSK is much more bandwidth efficient than unfiltered BPSK, with a 99 percent bandwidth of 0.7. Second, additional performance gains can be made through the use of the Viterbi algorithm in the

demodulation of the received signal.

2.3 Channel Statistics

The channel represents the path the signal takes to get from the transmitter to the receiver. For a land-based telephone system, the channel is a wire. A wire can be easily modeled. However, for terrestrial wireless and satellite communication systems, the channel is free space. Due to many factors, including the natural environments, buildings, and other users, the channel is more difficult to model. Usually, we wish to model an average channel, and we use AWGN for this purpose. White noise has a flat power spectral density and is uncorrelated with respect to time. However, when channel is faded, the noise is characterized by a non-Gaussian statistic. In a fading channel, the noise does not have a constant power spectral density and is both time varying and correlated with respect to time. When modeling such channels, we use a statistic such as the Rayleigh probability density function (PDF). The following will discuss both AWGN and Rayleigh channel statistics.

2.3.1 Additive White Gaussian Noise Channel

The AWGN channel is the standard channel used when designing communication systems. The noise is characterized by the Gaussian PDF as

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-x^2/2\sigma^2} \tag{2.7}$$

where σ^2 is the variance of the zero-mean noise. White noise is also defined to have a constant, or flat, power spectral density over the entire frequency band [14]. For AWGN, the one-sided noise spectral density is given as $N_o = 2\sigma^2$, and the information bit energy to noise density ratio is given as E_b/N_o in dB.

In the analysis, complex Gaussian noise is added to the modulated signal such

that the resulting complex received signal is given as

$$z_c = x_c + n_c \tag{2.8}$$

where z_c is the received complex signal, x_c is the modulated complex signal, and n_c is the complex AWGN. Both the real and imaginary components of n_c are independent zero-mean Gaussian random variables with variance σ^2.

2.3.2 Rayleigh Faded Channel

The Rayleigh PDF is usually used to model those channels which are characterized by several multiple paths with no significant line of sight (LOS) path present. The fade is Rayleigh distributed according to

$$p_{rayl}(a) = \frac{a}{\sigma^2} e^{-a^2}/2\sigma^2 \tag{2.9}$$

where a is the magnitude of the fade and σ^2 is the variance, which is different from the variance of the AWGN.

In our analysis, the modulated signal was multiplied by the Rayleigh distributed fade and then complex AWGN was added to the signal. The fading envelope was generated using two independent Gaussian random variables, x_1 and x_2, as

$$a(t) = \sqrt{x_1^2(t) + x_2^2(t)} \tag{2.10}$$

The resulting complex received signal can be expressed as

$$z_c = a_c \cdot x_c + n_c \tag{2.11}$$

where z_c is the complex received signal, a_c is the complex Rayleigh envelope as in Equation 2.10, x_c is the transmitted signal, and n_c is the complex AWGN.

Rayleigh fading can be applied in one several ways, and two methods will be described in the following. First, we can model a slow fading scenario. This is when

the duration of the fade is slow with respect to the signaling interval, or symbol duration, and was modeled by applying the same fade over an entire block of data. This will be referred to as block fading. A second application of Rayleigh fading is usually in a fast fading scenario. Since the duration of the fades are fast with respect to the signaling interval, or symbol duration, we modeled this fade by applying an independent Rayleigh distributed fade to each symbol within a given block. This will be referred to as symbol fading.

2.4 Decoding

Decoding estimates which code word was transmitted through the communication system. Several decoding methods exist – ranging from the simple, hard decision decoding, to the complex, soft input-soft output (SISO) iterative decoding. Our desire to maximize performance while minimizing complexity and the code structure determine which decoding method is used. The following will discuss hard decision decoding and soft decision and SISO iterative decoding.

For a system with inputs $\{-1, +1\}$, the hard decision decoder uses only the sign of the received bits. For our purposes, we used the $\{0, 1\} \rightarrow \{-1, +1\}$ modulation, and as a result, if the sign of the received bit is negative, the input to the decoder is a -1, and if the sign is positive, the input to the decoder is a $+1$. The decoder chooses the code word that is closest in Hamming distance to the received word. If the decoded word matches the transmitted word, then the block was decoded correctly, otherwise the block is said to be in error.

In contrast to the hard decision decoder, a soft decision decoder uses the sign of the received bits as well as additional soft input information to make the decoding decision. This additional information is usually given as reliability information which gives a confidence value regarding the hard decision. One decoding rule that is used in soft decision decoding is the maximum likelihood (ML) decoding rule, which minimizes the probability of error by maximizing the probability of the received word given a transmitted word, or $P(\mathbf{r}|\mathbf{v})$, where $\mathbf{r}$ is the received code word and $\mathbf{v}$ is the

transmitted code word [13]:

$$\hat{\mathbf{v}} = \text{argmax} \prod_i P(r_i|v_i) \tag{2.12}$$

where r_i and v_i are the bits of the code words $\mathbf{r}$ and $\mathbf{v}$ and we assume that the received symbols are independent. ML decoding is considered to be an optimum decoding rule [13].

Even though soft decision decoding is more complex to implement than hard decision decoding, it is often used due to the significant performance improvements that result. However, to further improve performance iterative decoding algorithms have been developed. These algorithms make use of not only soft input information, but soft output information as well. For more information on Pyndiah's, Chase's, and an application of Viterbi's iterative decoding algorithms, please refer to [15], [5], and [11], respectively.

In the following examination of TPCs, Thesling's cyclic-2 pseudo maximum likelihood decoding algorithm will be used [19]. This algorithm is sub-optimal, but for TPCs composed of extended Hamming codes and single parity check codes with BPSK in AWGN, the algorithm performs almost at the true correlation decoding value [7]. The first step of this algorithm involves making a hard decision on the received signal. The sign and magnitude, or reliability, information that results are then sent through the iterative process. The rows are decoded first with the soft output adjusted by a weighting function. This adjusted soft output information then becomes the soft input to the column decoder. The soft output from the column decoder is adjusted by a weighting function as well. The weighting function following each axis iteration need not be the same; the parameters of these functions are controlled by an optimized feedback value [3]. The extrinsic soft output and input information, noted as $\text{soft}_{\text{row,in}}$, $\text{soft}_{\text{row,out}}$, $\text{soft}_{\text{column,in}}$, and $\text{soft}_{\text{column,out}}$ in Figure 2-4, are passed between the two decoders in an iterative fashion. The decoding process continues until no further corrections can be made to the code word, or until the maximum number of iterations has been reached [2]. For our purposes, the weighting function

is considered to be part of the decoder function and is not shown in the figure.

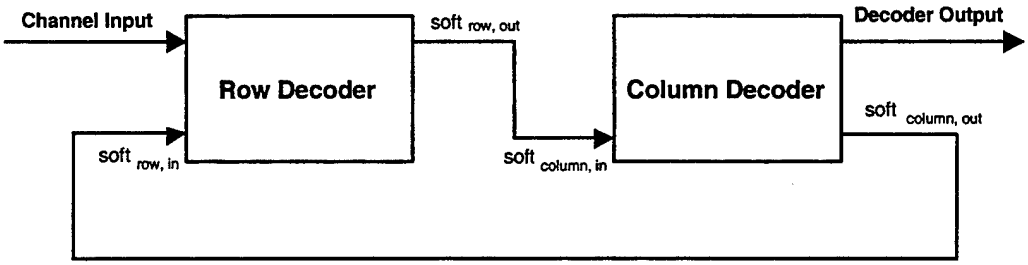


Figure 2-4: Diagram of the Iterative Decoding Process

Using iterative decoding, various two- and three-dimensional TPCs with code rates of 1/3 to 4/5 have performed within 2.5 dB of the Shannon Limit at a bit error rate (BER) of 10^{-5}. Figure 2-5 shows the bit error rate performance curves for a two-dimensional TPC in a BPSK modulated AWGN channel. The given results were achieved using only a few iterations. On average, this TPC was decoded in four to six complete iterations, with a complete iteration including both one row and column decoding. This code will be analyzed further in later sections of this thesis.

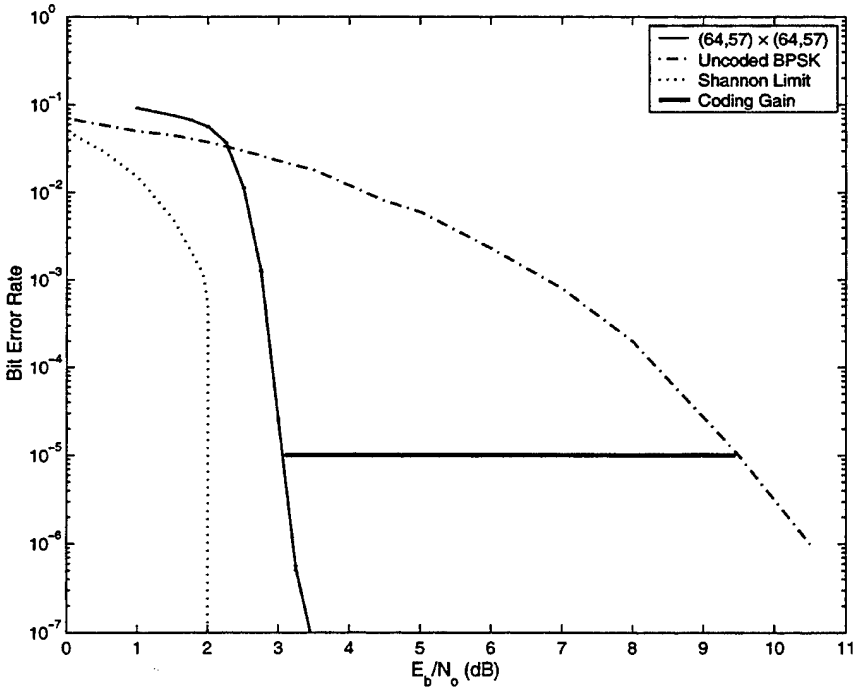


Figure 2-5: Turbo Product Code Performance Curves

The coding gain is based upon a code's performance improvement over the performance of uncoded data with similar conditions. Figure 2-5 not only gives an uncoded BPSK curve, but shows the Shannon Limit for codes of rate 4/5, which is at an E_b/N_o of 2 dB [7]. The coding gain is upper bounded by the distance from the uncoded curve to the Shannon Limit for the code at a given error rate. For the codes of rate 4/5, the maximum coding gain is 7.5 dB at the error rate of 10^{-5}. With BPSK modulation, the $(64, 57)^2$, rate 0.793 TPC has a coding gain of about 6.3 dB at the error rate of 10^{-5}. In contrast, the difference from the Shannon Limit curve to the code's performance curve shows coding, or power, inefficiency. For this code, we have an inefficiency of about 1.2 dB.

To give a measure of the accuracy of the performance curves used throughout the remainder of the thesis, error bars showing the 95 percent confidence intervals are shown for the curve for the $(64, 57)^2$ TPC in Figure 2-5. However, because the error bars are quite small in the preceding figure, a closer view of the BER curve is shown in Figure 2-6. As shown, the confidence interval is quite close to the BER curve, increasing only slightly at the lowest error rates plotted. These intervals are representative of the accuracy of the data used throughout this thesis.[2]

In the next chapter, we will begin our discussion density evolution and EXIT charts by examining ten Brink's methods for convolutional codes.

[2]This interval size assumes that all bit errors are independent, when they actually occur as part of an errored block. Thus, the error bars illustrated are somewhat optimistic.

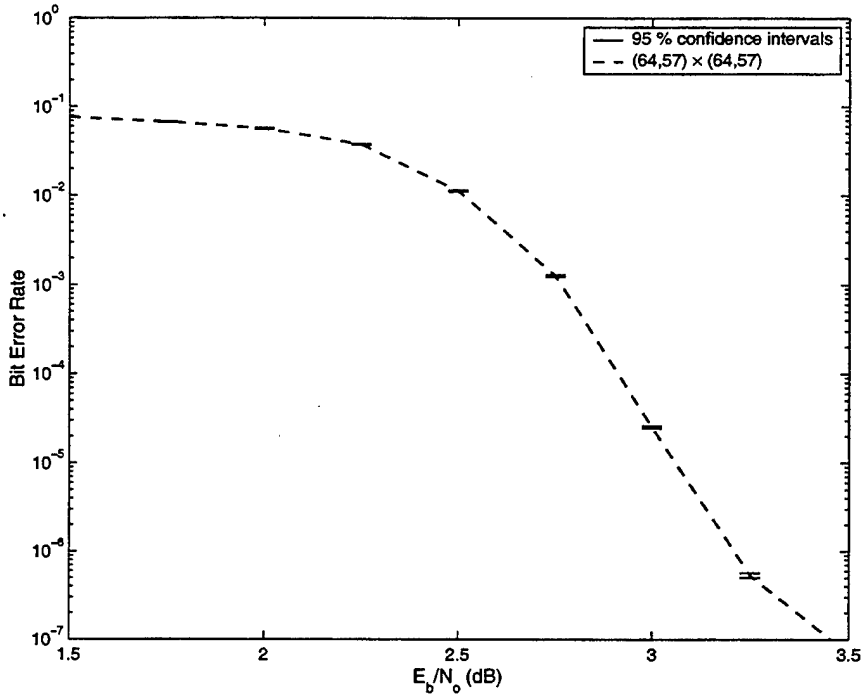


Figure 2-6: $(64, 57)^2$ TPC Performance Curves with Error Bars

Chapter 3

ten Brink's Methods for Turbo Codes

We have briefly examined how TPCs are constructed and decoded. We have also been given some preliminary performance results. Next, an analysis method is needed to determine which codes will perform best while meeting the system requirements. One such method is density evolution. Density evolution tracks the extrinsic information of the decoders from iteration to iteration in order to map the overall convergence of the decoder. The extrinsic information is the soft output information gained during the iterative decoding process. Divsalar, Dolinar, and Pollara applied a density evolution analysis to turbo codes and low density parity check codes. They tracked the evolution of the signal-to-noise ratios (SNR) of the decoders to analyze the iterative decoding process [6]. In [17] and [18], ten Brink examined the evolution of the extrinsic information through its mutual information using an Extrinsic Information Transfer (EXIT) Chart. A common goal of both analyses was to use their density evolution method to aid in code design for iteratively decoded systems. We chose to use ten Brink's methods in our density evolution analysis.

In his examination, ten Brink used the EXIT chart to visually demonstrate the evolution of the extrinsic information through a soft input/output iterative decoding process. By using the standard deviation of the soft input and soft output information, ten Brink calculated the mutual information between the extrinsic information and

the transmitted bits at the various stages of the iterative decoding process. For each constituent decoder, a plot of the mutual information at the input versus the mutual information at the output represents its transfer function. By plotting the transfer functions of the two decoders against each other, he was able to generate the EXIT chart[17, 18].

Using the EXIT chart, one could then plot the decoding trajectory. The decoding trajectory shows the evolution of the mutual information through the decoding process as a block is completes decoding. The trajectory is a stair-step like line connecting the two transfer characteristics. However, if the two transfer functions cross, we are unable to plot the decoding trajectory, and we know that that block was not decoded correctly. In addition, ten Brink showed that it was also possible to use the EXIT charts to predict the BER at low E_b/N_o values [18]. Finally, he listed several possible applications of the EXIT chart, including using it as a tool for code design.

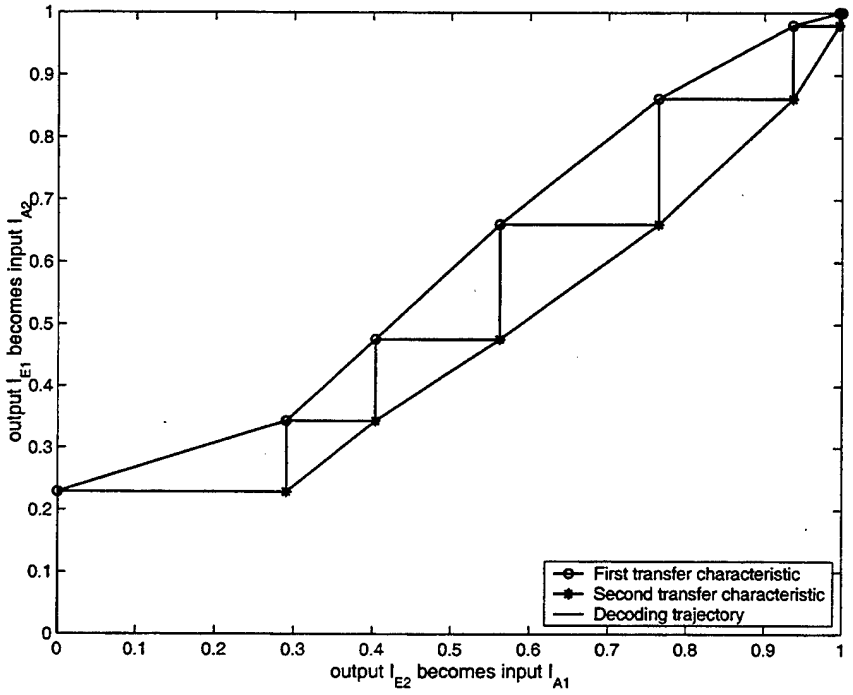


Figure 3-1: Example EXIT Chart and Decoding Trajectory

Figure 3-1 shows an EXIT chart for the $(64, 57)^2$, rate 0.793 TPC using GMSK over an AWGN channel at an E_b/N_o of 3.0 dB. As shown, the decoder is able to

correctly decode the block at this instance. The decoding trajectory is allowed to step through the tunnel created by the two transfer characteristics.

ten Brink developed his methods for examining the density evolution of an iterative decoding process using both serial and parallel concatenated turbo codes. Recently, in [10] his ideas have been applied to non-binary codes. In this thesis, his ideas will be applied to TPCs. However, we will first begin with an in-depth review of the process required to create an EXIT chart. The EXIT chart is a graphical description of the density evolution of the decoding process, and it is the primary tool that will be used for not only the density evolution analyses, but performance, or BER, analysis and convergence analysis as well. This section also sets up the required background and nomenclature used in the TPC analysis.

3.1 Mutual Information Analysis

3.1.1 The Iterative Decoder

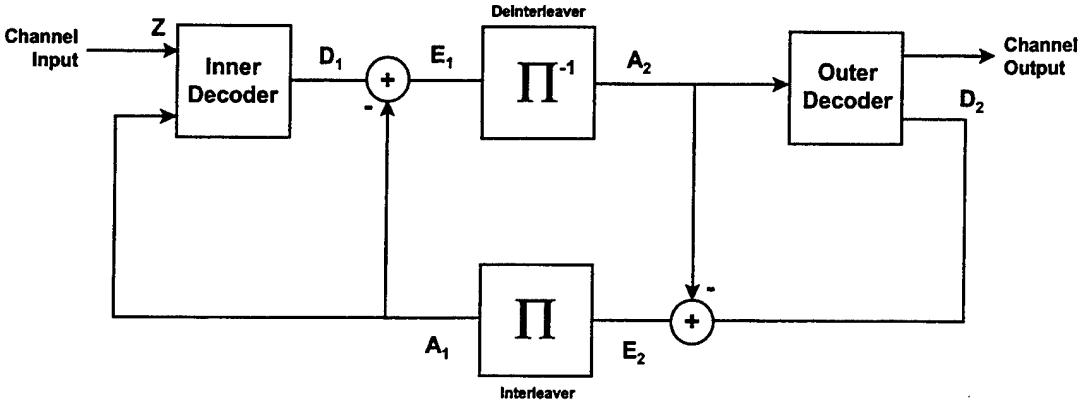


Figure 3-2: Iterative Decoder for Serial Concatenated Codes

Figure 3-2 shows the serial iterative decoder for which ten Brink developed his analysis [17]. From this figure, we can see that a deinterleaver and an interleaver separate the inner and outer convolutional decoders. These decoders perform maximum a posteriori (MAP) decoding according the the Bahl-Cocke-Jelinek-Raviv (BCJR) algo-

rithm [17]. The decoding process begins when the noisy channel bits, Z, are input to the decoder. These bits are decoded first by the inner decoder, giving the soft output D_1. The extrinsic information E of the individual decoder is the total soft output information, D, minus any *a priori* input information A such that $E_1 = D_1 - A_1$ [17, 18]. This subtraction is necessary to determine exactly how much information is gained as the bits pass through the individual decoder. E_1 gives the extrinsic information of the first decoder and is the *a priori* input for the second decoder, A_2. Similarly, $E_2 = D_2 - A_2$ is the extrinsic information of the second decoder and *a priori* input to the first decoder. The process of passing the extrinsic information from decoder to decoder continues until the decoding is completed, which is usually after a certain number of iterations or when the block is decoded to a codeword.

The variables $Z, E_1, D_1, A_1, E_2, D_2$, and A_2 are log-likelihood ratios (L-values) [17, 18]. For the AWGN channel, the received signal, z, is given as

$$z = x + n \tag{3.1}$$

where x are the transmitted bits and n is the additive noise. The noise n is also Gaussian distributed with zero mean and $\sigma_n^2 = N_o/2$. The L-values [17, 18] are then calculated according to

$$Z = \ln \frac{p(z|x = +1)}{p(z|x = -1)} \tag{3.2}$$

where the conditional PDF of the noisy channel bits, Z, given the transmitted bits X is

$$p(z|X = x) = \frac{e^{-((z-x)^2/2\sigma_n^2)}}{\sqrt{2\pi}\sigma_n}. \tag{3.3}$$

Using the conditional PDF and the Gaussian channel representation of Equation 3.1, the L-value representation can be simplified to

$$Z = \frac{2}{\sigma_n^2} \cdot z = \frac{2}{\sigma_n^2} \cdot (x + n). \tag{3.4}$$

This representation can also be re-written as

$$Z = \mu_Z \cdot x + n_Z \tag{3.5}$$

where

$$\mu_Z = 2/\sigma_n^2 \tag{3.6}$$

and the noise n_Z is Gaussian with zero mean and variance

$$\sigma_Z^2 = 4/\sigma_n^2. \tag{3.7}$$

Finally, we can show that the mean and variance of Z are related by

$$\mu_Z = \frac{\sigma_Z^2}{2}. \tag{3.8}$$

This development of the received signal Z will be useful for later derivations [17, 18].

3.1.2 Mutual Information Calculations

ten Brink chose to model the *a priori* input to the individual decoders as an independent Gaussian distributed random variable A with variance σ_A^2 and zero mean [17, 18]. As a result, it is possible to write the L-values A as

$$A = \mu_A \cdot x + n_A \tag{3.9}$$

in accordance with Equation 3.5, and where x are the transmitted bits and n_A is the additive Gaussian noise, with zero mean and variance σ_n^2. Following from modelling A as Gaussian distributed and with $\mu_A = \sigma_A^2/2$, the conditional probability density function of A given X becomes

$$p_A(\xi|X = x) = \frac{e^{-(\xi - x\cdot\sigma_A^2/2)^2/2\sigma_A^2}}{\sqrt{2\pi}\sigma_A}. \tag{3.10}$$

The mutual information is a measure of the gain in information due to the reception of a signal. This gain in information is found by determining the difference between the information uncertainty before, the *a priori* probabilities, and after the reception of the signal, the *a posteriori* probabilities [12]. A general representation of this calculation is given in Equation 1.2. The mutual information $I_A = I(X;A)$ between the transmitted bits X and the *a priori* information A is found using the following

$$I_A = \frac{1}{2}\sum_{x=-1,1}\int_{-\infty}^{+\infty} p_A(\xi|X=x) \times \log_2 \frac{2 \cdot p_A(\xi|X=x)}{p_A(\xi|X=-1)+p_A(\xi|X=1)} d\xi. \quad (3.11)$$

Using Equation 3.10 and Equation 3.11, the mutual information I_A becomes

$$I_A(\sigma_A) = \int_{-\infty}^{+\infty} \frac{e^{-(\xi-\sigma_A^2/2)^2/2\sigma_A^2}}{\sqrt{2\pi}\sigma_A} \cdot (1 - \log_2[1+e^{-\xi}])d\xi \quad (3.12)$$

The mutual information is now given as a function of the standard deviation of the *a priori* input to the decoder [17, 18]. The mutual information is a monotonically increasing value that exists over the range 0 to 1, assuming equally likely binary source symbols, and where $\sigma_A \geq 0$ always. We can also abbreviate Equation 3.12 as

$$J(\sigma) \equiv I_A(\sigma_A = \sigma). \quad (3.13)$$

Because I_A is monotonically increasing, it is also reversible according to

$$\sigma_A = J^{-1}(I_A). \quad (3.14)$$

We can quantify the mutual information of the extrinsic output $I_E = I(X;E)$ using a similar method. However, ten Brink did not assume that the extrinsic information is Gaussian distributed [18]. In his work, a Monte Carlo method was used to determine the distributions p_E, and the mutual information I_E is computed using equation 3.12 with the derived p_E in place a p_A. As with the mutual information I_A, I_E is monotonically increasing over the range 0 to 1 [17, 18].

3.1.3 Transfer Characteristics

The transfer characteristics of the constituent decoders are found by viewing the extrinsic output information I_E as a function of the mutual information of the *a priori* input I_A and the signal power to noise density, E_b/N_o, such that

$$I_E = T(I_A, E_b/N_o). \tag{3.15}$$

When the signal power to noise density is held constant, this relationship becomes $I_E = T(I_A)$. ten Brink was able to isolate his decoders so that he could independently plot the extrinsic information transfer characteristics of the inner and outer decoders for a given and constant E_b/N_o. Using the inverse relationship between σ_A and the mutual information I_A from Equation 3.14, ten Brink applied the independent Gaussian random variable of Equation 3.9 to the decoder of interest. By choosing the appropriate value of σ_A, the desired value of I_A could be achieved [17, 18].

In the plots of the transfer characteristics, the mutual information of the *a priori* input was plotted on the abscissa and the mutual information of the extrinsic output was plotted on the ordinate for the inner decoder. The axes are reversed for the second, or outer, decoder. These plots are the transfer characteristics of the individual decoders [17, 18]. Since the mutual information ranges from zero to one, the transfer characteristics will also range from zero to one. Finally, the transfer characteristics are monotonically increasing because the mutual information is also monotonically increasing.

3.2 The Extrinsic Information Transfer Chart

To plot the EXIT chart for a complete iterative decoding process, transfer characteristics for each of the decoders are used. The transfer characteristics for the first constituent decoder are plotted such that the input extrinsic information is on the abscissa and the output extrinsic information is on the ordinate. For the second decoder, the axes are reversed. By plotting the two curves on the same diagram, we

create the EXIT chart [17, 18]. This axis swapping is necessary such that the extrinsic information and channel output of the first decoder I_{E_1} becomes the *a priori* input I_{A_2} to the second decoder. Then, the extrinsic information and channel output of the second decoder I_{E_2} becomes the *a priori* input I_{A_1} to the first decoder, and so on. Please refer to Figure 3-1 on page 31 for an example of the EXIT chart.

The decoding trajectory can be added to the EXIT chart to show the evolution of the mutual information through the decoding process as a block is decoded. The trajectory is a stair-step shaped line connecting the two transfer characteristics. For a block to be decoded correctly, the decoding trajectory must span the range of the transfer characteristics, zero to one. We consider a decoding process to be complete and successful when the mutual information is monotonically increasing over the entire range $0 \leq I_E \leq 1$.

The key idea of the success of a decoding process is that the amount of mutual information must always increase from one half iteration to the next. As long as $I_{E_{2,n+1}} > I_{E_{2,n}}$, the iterations, and ultimately the decoding process, will proceed. If $I_{E_{2,n+1}} \leq I_{E_{2,n}}$, the decoding process will not be able to complete [17, 18]. As a result, the two transfer characteristics will intersect, and the corresponding decoding trajectory will terminate at the point of intersection. The current block will not be decoded correctly because the decoding trajectory did not reach one, which is needed for the block to be complete decoding.

3.3 Additional Results

ten Brink continued his examination of turbo codes by using the EXIT chart to predict the BER for a code structure after any number of iterations [18]. ten Brink also applied his methods to a coherently detected, fully interleaved Rayleigh channel, including the BER prediction [18]. Finally, he listed several important additional uses of the EXIT chart. These include using the chart as a design tool for iterative decoding schemes, including code searches, and using the chart to help gain insight into the convergence behavior of the iterative decoder [17, 18]. His methods for these

particular anyalyses will be discussed in the following chapters.

Chapter 4

Application of ten Brink's Methods to Turbo Product Codes

While ten Brink developed the EXIT chart for turbo codes, the main target of the examination was actually the iterative decoder. In chapter 2, it was shown that TPCs are decoded using an iterative decoder. As a result, TPCs are an ideal extension for the EXIT chart analysis. In this chapter, we determine changes necessary due to the differences in TPC decoder structures as well as in the information available from the TPC decoder. Then, ten Brink's assumptions will be extended to TPCs. Finally, differences in the mutual information analysis and methods used for plotting the EXIT chart and associated decoding trajectory will be presented.

4.1 Iterative Decoder

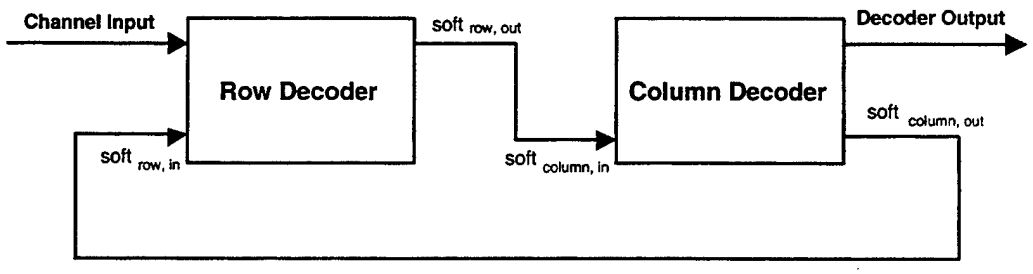


Figure 4-1: Iterative Decoder for Turbo Product Codes

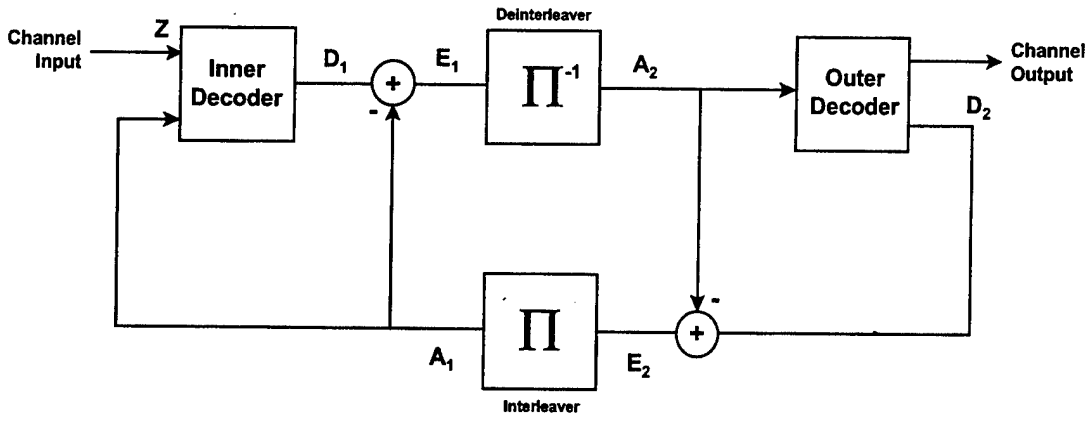


Figure 4-2: ten Brink's Iterative Decoder for Turbo Codes

Recall the TPC decoder and the turbo code decoder of sections 2.4 and 3.1.1, shown in Figures 4-1 and 4-2. These two decoders, while operating serially, have some key differences. First, in ten Brink's decoder, an interleaver separated the two constituent decoders. There is no interleaver in the TPC decoder, but this difference is not significant to this analysis. Also, the structure of ten Brink's decoder requires a subtraction, $E = D - A$, to determine the extrinsic output information of the previous iteration. However, for the TPC decoder, the extrinsic output information was supplied directly by the decoder, and these values were directly used as the extrinsic input information to the next iteration. This last difference is significant in that it will make this analysis easier.

Aside from these two main differences, the two decoders both operate with the same purpose and in a similar manner. Both decoders send the soft output and input information iteratively between the two decoders until the termination criterion is met. The termination criterion is either a maximum number of iterations, or until no further corrections can be made.

4.2 Turbo Code Assumptions

ten Brink used one key assumption in his calculation of the mutual information of the *a priori* inputs. This assumption is that these inputs A can be modelled as an independent Gaussian random variable, leasing to the mutual information given in equation 3.12.

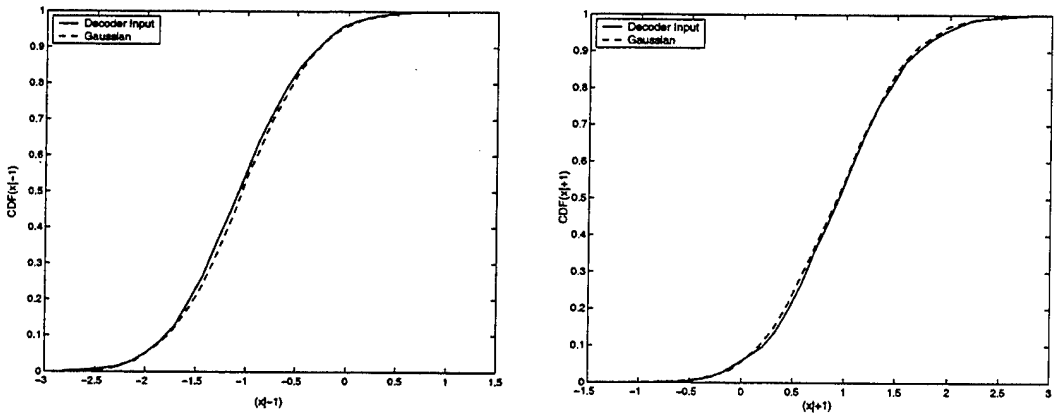
However, in contrast to the assumption on A, ten Brink did not assume the extrinsic output information E to be Gaussian distributed as well. Histogram measurements were used to compute the mutual information I_E according to equation 3.12, using the distributions for p_E [17, 18].

4.3 Extension from Turbo Codes to TPC

To calculate the mutual information of the extrinsic soft output and input information of the TPC decoder, several assumptions were made. First, we continued to follow ten Brink's Gaussian assumption on the *a priori* information A. However, as the TPC decoder setup dictated that $A = E$, we assumed the extrinsic output information E to be Gaussian distributed as well.

In [9], el Gamel and Hammons showed that a turbo decoder could be analyzed using a Gaussian approximation. They observed that if the inputs to a decoding algorithm were independent Gaussian random variables, then the outputs could be tightly approximated by a Gaussian random variable as well. This observation will be useful to show that a Gaussian assumption can be used on the extrinsic information here as well.

ten Brink developed his model using the conditional probability distribution of the *a priori* inputs A given the transmitted bits X [17, 18]. These same conditional distributions were also used in the TPC analysis. We will use data from the $(64, 57)^2$ TPC with BPSK modulation in AWGN at an $E_b/N_o = 2.5$ dB to illustrate our Gaussian assumption. The left-hand plot of Figure 4-3 shows the cumulative distribution function (CDF) of the TPC decoder input conditioned on the transmitted bits be-



(a) CDF $(x|-1)$ (b) CDF $(x|+1)$

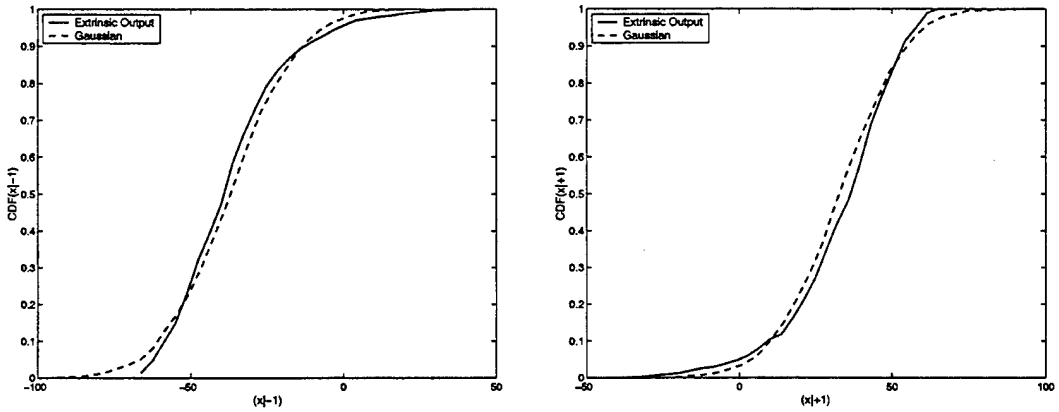
Figure 4-3: Decoder Input Bits Conditioned on $X = -1$ and $X = +1$ for BPSK in AWGN at $E_b/N_o = 2.5$ dB

ing negative and the right-hand plot shows the CDF for the case where the input is conditioned on the transmitted bits being positive.

As we can see in the two figures, the distributions are very closely approximated by the CDF of a Gaussian random variable with the same mean and variance. From [9], we know that the Gaussian approximation can be used for instances where the input to the decoder is a Gaussian variable. Since we know the decoder input can be approximated by a Gaussian random variable, we can proceed to use the Gaussian approximation on the extrinsic output information of the iterative TPC decoder.

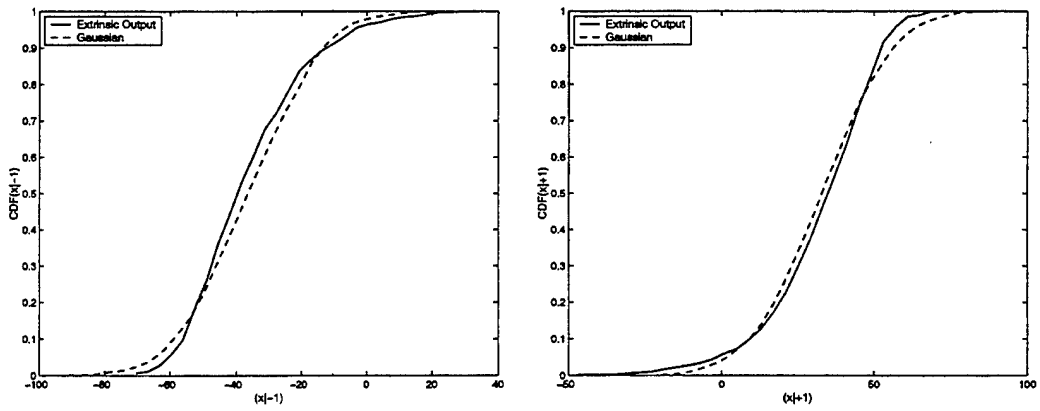
Figure 4-4 shows that the extrinsic output information after the first iteration may be approximated by a Gaussian of the same mean and variance as well. Figure 4-5 extends these ideas to GMSK modulation for the same $(64, 57)^2$ TPC. The conditional output information was also reasonably approximated by a Gaussian.

From the decoder setup, we know that this extrinsic output becomes the *a priori* input to the next iteration. Therefore, we will use a Gaussian assumption on both the extrinsic output information E of one iteration and the extrinsic, or *a priori* input information A of the next iteration of the TPC decoder. The mutual information is



(a) CDF $(x|-1)$ (b) CDF $(x|+1)$

Figure 4-4: Extrinsic Output Conditioned on $X = -1$ and $X = +1$ for BPSK in AWGN at $E_b/N_o = 2.5$ dB



(a) CDF $(x|-1)$ (b) CDF $(x|+1)$

Figure 4-5: Extrinsic Output Conditioned on $X = -1$ and $X = +1$ for GMSK in AWGN at $E_b/N_o = 3.0$ dB

then calculated according to

$$I(\sigma) = \int_{-\infty}^{+\infty} \frac{e^{-(\xi-\sigma^2/2)^2/2\sigma^2}}{\sqrt{2\pi}\sigma} \cdot (1 - \log_2[1 + e^{-\xi}])d\xi \tag{4.1}$$

for both E and A.

4.4 Differences in Analyses

In his analysis, ten Brink was able to isolate the constituent decoders. By isolating the decoders, he was able to independently plot the transfer characteristics for the decoders. Using the inverse relationship between the mutual information of A and σ_A, expressed as

$$\sigma_A = J^{-1}(I_A), \tag{4.2}$$

ten Brink selected the appropriate values of σ_A needed to generate these transfer characteristics. Then, by taking a transfer characteristic for a certain code from each of the two constituent decoders, ten Brink was able to combine these plots into the EXIT chart and the decoding trajectory could then be plotted between these two curves.

However, we did not have complete access to the TPC decoder. As a result, we treated the decoder, and its constituent decoders, as black boxes for the analysis. This impacted the generation of the EXIT charts and decoding trajectories because it did not allow for the constituent decoders to be isolated. Therefore, independent plots of the transfer characteristics of the TPC decoders could not be generated.

Treating the decoder as a black box also added another difference from ten Brink's analysis. The decoder set-up did not allow for a zero *a priori* input to the second, or column, decoder. As previously mentioned, the extrinsic output of the first decoder became the *a priori* input to the second decoder. The constituent decoders and the extrinsic data were not independent from iteration to iteration. Using this knowledge, EXIT charts and decoding trajectories could be plotted, even without generating independent transfer characteristics. To generate the transfer characteristics for the

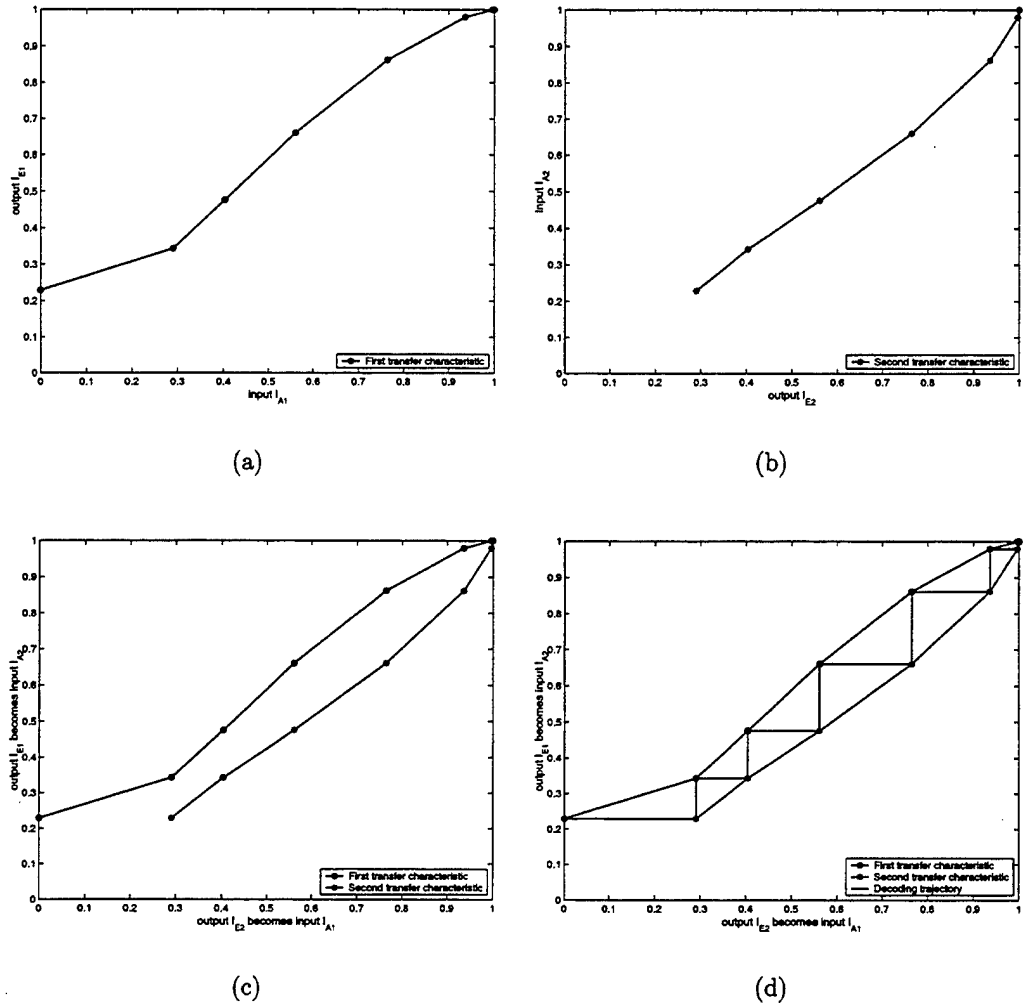


Figure 4-6: EXIT Chart Development for the $(64, 57)^2$ TPC with BPSK in AWGN at $E_b/N_o = 3.0$ dB

TPC decoder, a codeword was generated and sent through a communication channel. Then, the TPC decoder received this signal. Using the standard deviations, σ_A and σ_E, of the soft input and output extrinsic information from the decoding process the mutual information was computed. Then, the EXIT charts and decoding trajectories were generated from these mutual information values, I_A and I_E, derived from the simulation results. Figure 4-6 shows the development of an EXIT chart from the transfer characteristics, and then the decoding trajectory from the EXIT chart.

Chapter 5

Density Evolution Analysis Utilizing EXIT Charts: AWGN Channel

We begin our density evolution analysis of TPCs with the AWGN channel. The AWGN channel is a standard model and starting point for an analysis, and is a good model for many satellite and wireless channels as well. Also, we will examine the results of BPSK modulation first, and then examine the results of GMSK modulation. We look at BPSK first as a standard modulation scheme for comparison with other systems.

There are three codes that will be examined in depth throughout the remainder of this chapter. The first is the $(64, 57)^2$ TPC. With code rate 0.793 and block size 4096, this is a square code with a fairly high rate that makes it desirable for use in certain communication systems. The other codes that will be examined are the $(64, 57) \times (32, 26)$ and $(32, 26) \times (64, 57)$ non-square TPCs. While these codes both have the same rate, 0.724, and block size, 2048, several interesting differences exist in their performance results. These two codes are essentially the same, with only the encoding and decoding orders reversed. It should also be mentioned that the $(64, 57)$ and $(32, 26)$ constituent codes are both extended Hamming codes and have the same error correcting capability, in terms of minimum distance. Additionally, the

$(32, 26)^2$ TPC, with code rate 2/3 and block size 1024, will be used to help compare the performance of the non-square codes of interest. The d_{min} values, and error correcting capability, for these codes are not well defined. However, it is customary to use the product of the d_{min} values for the constituent codes as the d_{min} value for the TPC, which gives a d_{min} of 16 for the four codes listed here.

5.1 BPSK results

We begin with the results of the BPSK simulations to provide a basis for comparison with the GMSK simulations. All transfer characteristics and decoding trajectories shown were determined using averaged results from five consecutive blocks. The averaging was performed on the raw extrinsic information, and the resulting averaged deviations were used to compute the mutual information value for that five block set.

5.1.1 Square codes

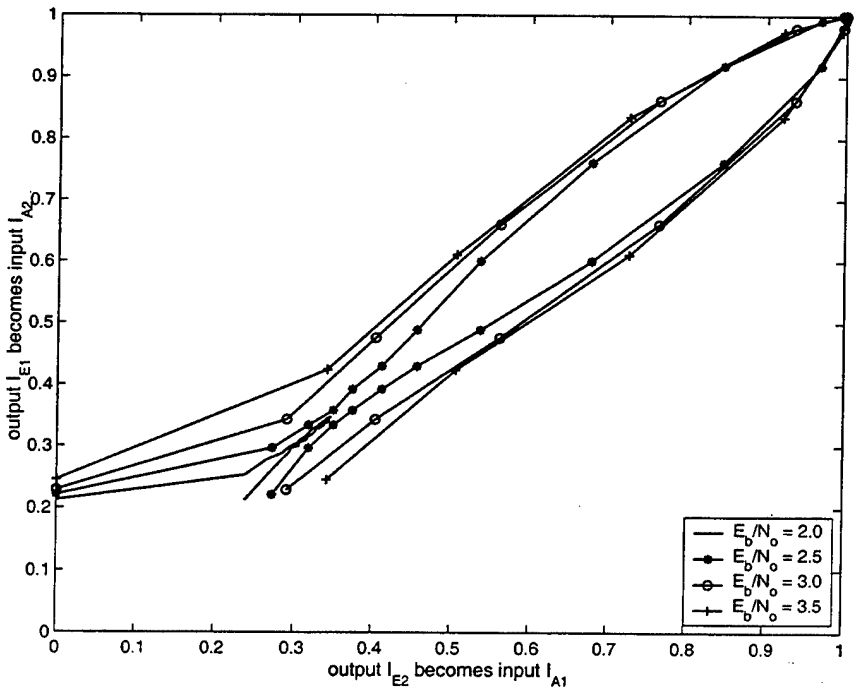


Figure 5-1: EXIT chart for a set of E_b/N_o values for the $(64, 57)^2$ TPC

Figure 5-1 gives the EXIT chart results over a set of increasing E_b/N_o values for the $(64, 57)^2$ TPC. As shown in the figure, the initial value of I_E increases as the E_b/N_o increases as expected. The transfer characteristics of both constituent decoders are monotonically increasing, as expected. The curves for $E_b/N_o = 2.0$ dB were not complete; that is they did not reach a mutual information value of one. However, the curves for the E_b/N_o values ≥ 2.5 dB were complete; that is they did reach a mutual information value of one. Figure 5-2 shows all of the curves over the mutual information values from zero to 0.5. We can clearly see in more detail how the curves for $E_b/N_o = 2.0$ dB end and intersect at a mutual information value of about 0.35. At some point between 2.0 dB and 2.5 dB, the code is said to have converged. For our purposes, we will define convergence as the E_b/N_o value for which the EXIT chart and associated decoding trajectory are first complete. Convergence will be discussed in more detail in chapter 8.

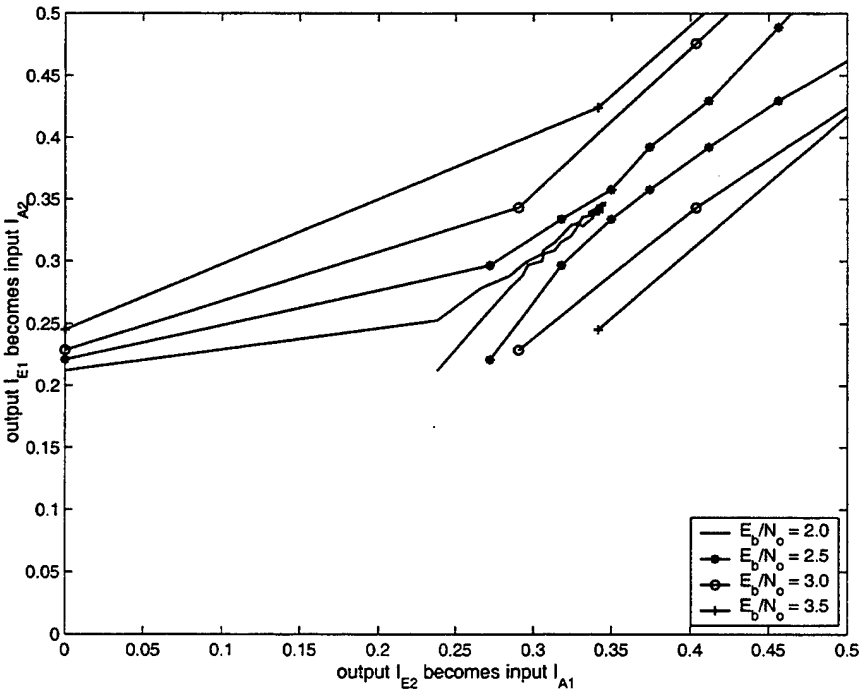


Figure 5-2: Close-up view of the transfer characteristics for a set of E_b/N_o values for the $(64, 57)^2$ TPC

ten Brink defined several regions of the EXIT chart as they relate to the BER

curve. The first region includes the E_b/N_o values where the transfer characteristics are not complete, or when they intersect at a low mutual information value. The resulting error rate of the TPC will fall in the region of high error rates, where negligible BER reduction occurs with decoding. ten Brink denoted this region as the pinch-off region [18]. The second region includes the E_b/N_o values which occur shortly after the convergence point. This bottleneck region of the EXIT chart relates to the waterfall region of the BER curve where the error rate improves with successive iterations [18]. From Figure 5-1, we notice that as the value of the E_b/N_o increases, the tunnel between the transfer characteristics widens for these curves. This widening of the tunnel gives a visual indication of the error rate improvement of the waterfall region. The final region of the BER curve is when fairly high E_b/N_o values result in a low BER after only a few iterations, and is referred to as the wide-open region [18]. We should remember that TPCs do not exhibit the same error floor as turbo codes. ten Brink linked the wide-open region to the error floor region of the BER curve, but this is not the case with TPCs where the wide-open region still exhibits BER improvement as the E_b/N_o increases.

We can observe these regions in Figure 5-1 and Figure 5-2. Clearly, the curves for $E_b/N_o = 2.0$ dB mark the pinch-off region. Then, we can see the bottleneck region in the curves of $E_b/N_o = 2.5$ dB. The decoding trajectory for this E_b/N_o can sneak through the narrow tunnel. The curves for $E_b/N_o = 3.0$ dB have a larger tunnel showing the improvement of the waterfall region. Finally, the transfer characteristics begin to follow a more linear path from mutual information values of zero to one in the wide-open region, as exhibited by the curves for $E_b/N_o = 3.5$ dB.

Figure 5-3 shows the EXIT charts and decoding trajectories for an E_b/N_o of 2.0 dB and 3.5 dB. As discussed previously, the curves for $E_b/N_o = 2.0$ dB intersect at about 0.35. Since the decoding trajectory did not reach a mutual information value of one, we can safely say the code will not be decoded correctly at this E_b/N_o value and that the block is in error. The resulting error rate for an E_b/N_o of 2.0 dB will be high. However, for the $E_b/N_o = 3.5$ dB curve, the tunnel is wide-open and the decoding trajectory quickly reaches a mutual information value of 1.0, after only about six full

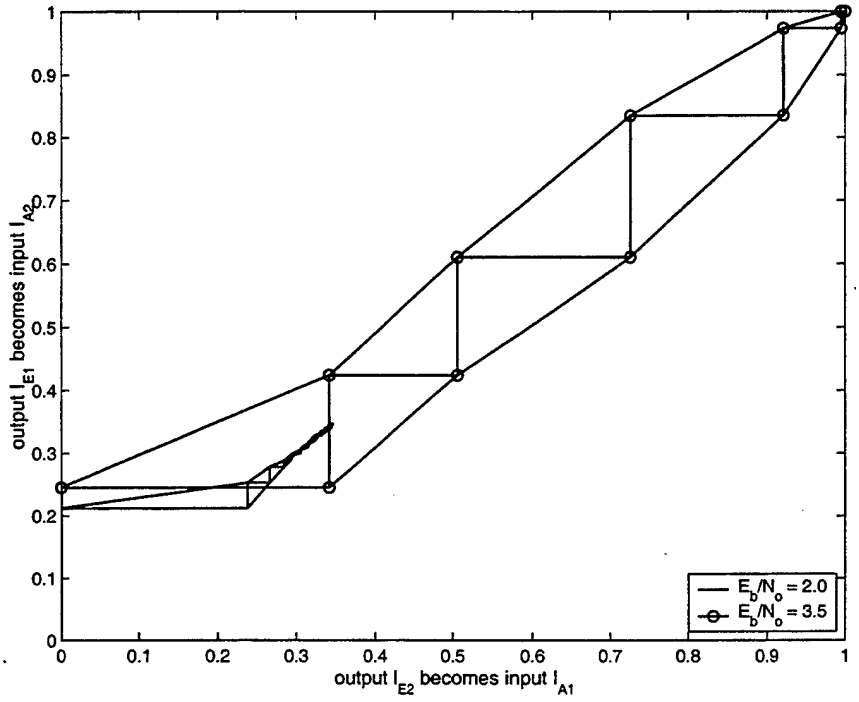


Figure 5-3: EXIT chart and decoding trajectories for the $(64, 57)^2$ TPC at $E_b/N_o =$ 2.0 and 3.5 dB

iterations. We can say that the code is decoded correctly at this E_b/N_o value and that the block will not be in error. The resulting error rate for the 3.5 dB curve will be in the region of low BER.

Finally, we compare two different square codes at the same E_b/N_o. Figure 5-4 shows EXIT charts for the $(64, 57)^2$ and $(32, 26)^2$ TPCs at $E_b/N_o = 2.5$ dB. As shown, the tunnel of the $(32, 26)^2$ TPC is much more open at this E_b/N_o than the $(64, 57)^2$ TPC. At this E_b/N_o, the $(64, 57)^2$ code is in the bottleneck region of the BER curve while the $(32, 26)^2$ code is in the wide-open region. However, when these two constituent codes are combined, several interesting changes occur, as will be discussed in the following section.

5.1.2 Non-Square codes

Square codes are often used because of their symmetry. However, by combining two different constituent codes into a TPC, we can develop codes of a much larger range

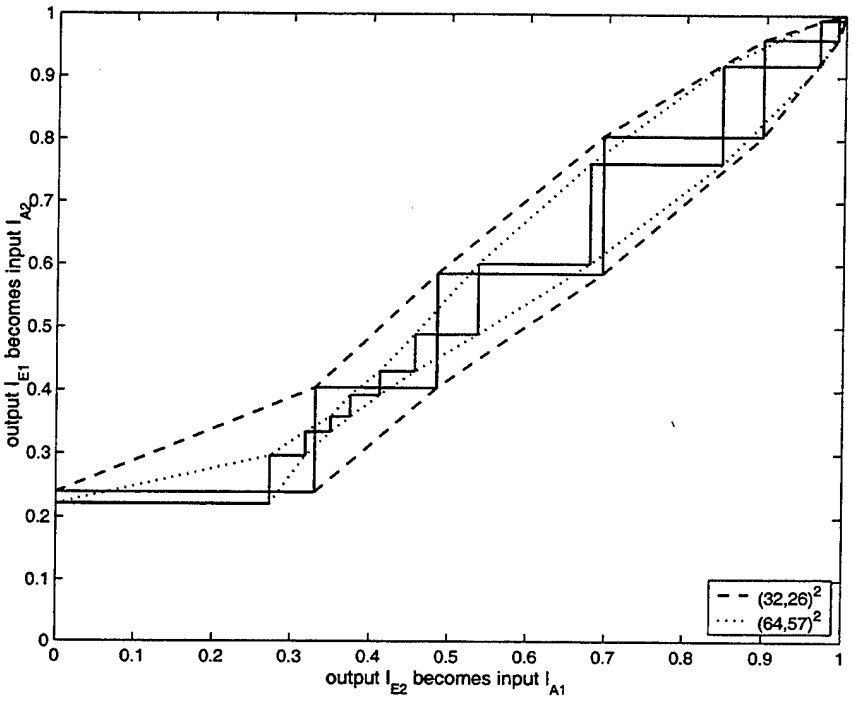
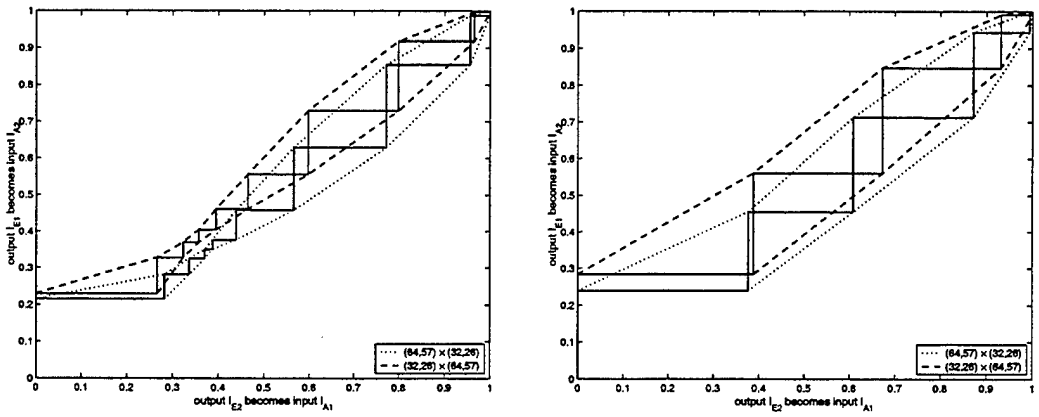


Figure 5-4: EXIT chart and decoding trajectories for the $(64,57)^2$ and $(32,26)^2$ TPCs at $E_b/N_o = 2.5$ dB

of block sizes and code rates. Non-square codes have several properties that will be discussed in the following section. Figure 5-5 shows the EXIT chart and decoding trajectories of two non-square codes at two different E_b/N_o values.

From the figure, we see that in both plots the $(32,26) \times (64,57)$ TPC has a larger initial value of I_{E_1}. This seems contrary to our thinking that the larger code, in terms of length, should be first, but when Figure 5-4 is examined, we find the result to be correct; the $(32,26)^2$ code has a larger initial value of I_{E_1} than the $(64,57)^2$ code. Also in both plots the $(64,57) \times (32,26)$ code has a larger initial value of I_{E_2} for the same reason. As the E_b/N_o value increases, the decoding rate varies as well. In the left-hand plot, at an $E_b/N_o = 2.0$ dB both codes decode at the same rate, an average of 10.5 full iterations. However, for larger values, including $E_b/N_o = 3.0$ dB shown in the right-hand plot, the $(32,26) \times (64,57)$ code decodes slightly faster than the $(64,57) \times (32,26)$ code, by about one half-iteration. It should be noted that at $E_b/N_o = 3.0$ dB, both these codes have a BER of about 10^{-5} after a full number



(a) $E_b/N_o = 2.0$ dB

(b) $E_b/N_o = 3.0$ dB

Figure 5-5: EXIT chart and decoding trajectories for the $(64, 57) \times (32, 26)$ and $(32, 26) \times (64, 57)$ TPCs

of decoding iterations. Finally, it must be mentioned that it is very interesting that these differences exist between these two codes which are in fact the same code, just ordered differently on the two axes.

Figure 5-6 shows the performance curves for these two non-square codes under varying conditions. First, we can see the curves for the two codes run where they have up to 32 full iterations to decode the block. We can also see the curves for these same two codes, except now they are limited to 5 full iterations. We can see that when there is no limit on the number of iterations that can be used, the codes perform essentially the same. However, when the number of iterations is greatly limited, the $(32, 26) \times (64, 57)$ performs slightly better, by about 0.05 dB, in BPSK as predicted by the EXIT chart analysis. We shall see if this observation and the others made previously in this section are further verified in the following section examining GMSK modulation.

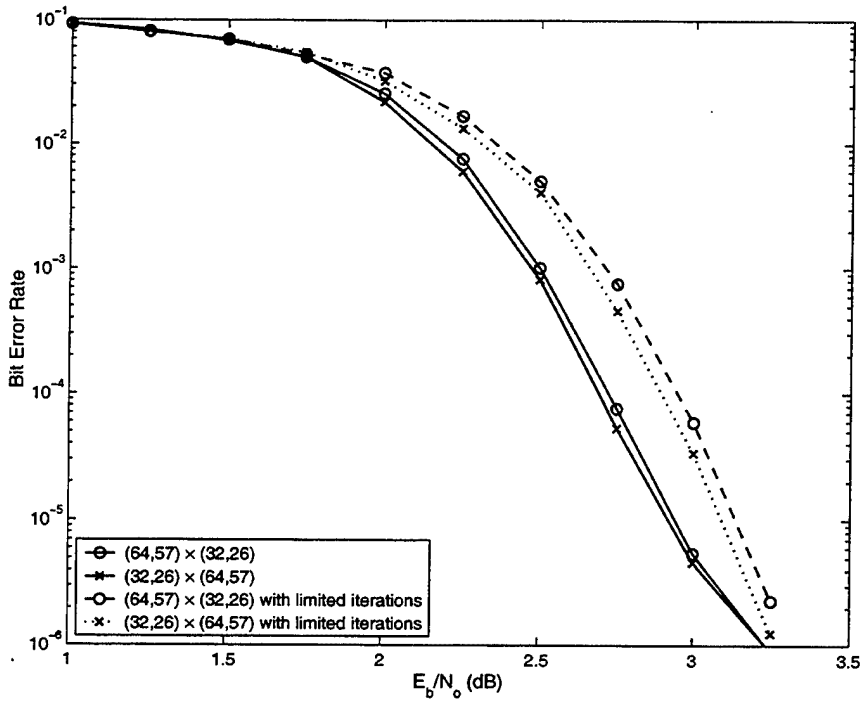


Figure 5-6: Performance Curves of the Non-square codes with BPSK in AWGN

5.2 GMSK results

From our analysis, we know that the decoder operates in the same manner regardless of the modulation scheme used on the code prior to the received signal entering the decoder. As a result, we expect similar EXIT chart results from the GMSK analysis as from the BPSK analysis given that the statistics of the input signal remain similar.

5.2.1 Square codes

We begin by examining the transfer characteristics of the $(64, 57)^2$ square TPC. Figure 5-7 gives the EXIT chart results over a set of increasing E_b/N_o values for the $(64, 57)^2$ TPC with GMSK modulation. As shown in the figure, we can see that as with the BPSK case, the mutual information I_E increases as the E_b/N_o value increases. These initial values for I_E are at approximately the same values for both the GMSK and BPSK cases. This indicates that the mutual information is in fact related to the standard deviation of the input to the decoder, which varies with the E_b/N_o

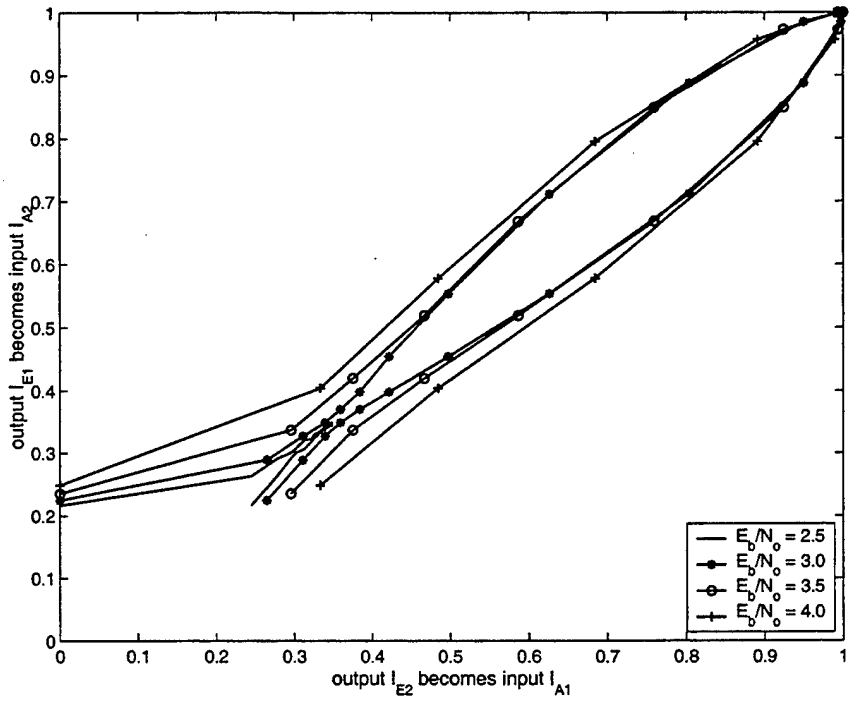


Figure 5-7: EXIT chart for a set of E_b/N_o values for the $(64, 57)^2$ TPC

value used, rather than the modulation scheme used.

We can also make observations about the various stages of the error rate curve as the curves on the EXIT chart transition from being incomplete at $E_b/N_o = 2.5$ dB to being complete at larger E_b/N_o values. Figure 5-8 shows the curves in GMSK from a mutual information value of zero to 0.5. In the figure, we can see that the curves for this E_b/N_o value intersect at ~ 0.35 dB, in the pinch-off region. We should note that this is very similar to the BPSK results shown in Figure 5-2. The curves for $E_b/N_o = 3.0$ dB are complete, but with a narrow tunnel, and fall in the bottleneck region of the EXIT chart, or in the waterfall region of the BER curve. Finally, the curves demonstrate the wide-open region at $E_b/N_o = 3.5$ and 4.0 dB. We can also see how the code moves through the convergence point in the chart. The convergence takes place somewhere between 2.5 and 3.0 dB. Overall, these results are very similar to the BPSK results of the previous section. The main difference is that these results occur approximately at an E_b/N_o value 0.5 dB larger for GMSK than BPSK.

Figure 5-9 shows the EXIT chart complete with decoding trajectories for two

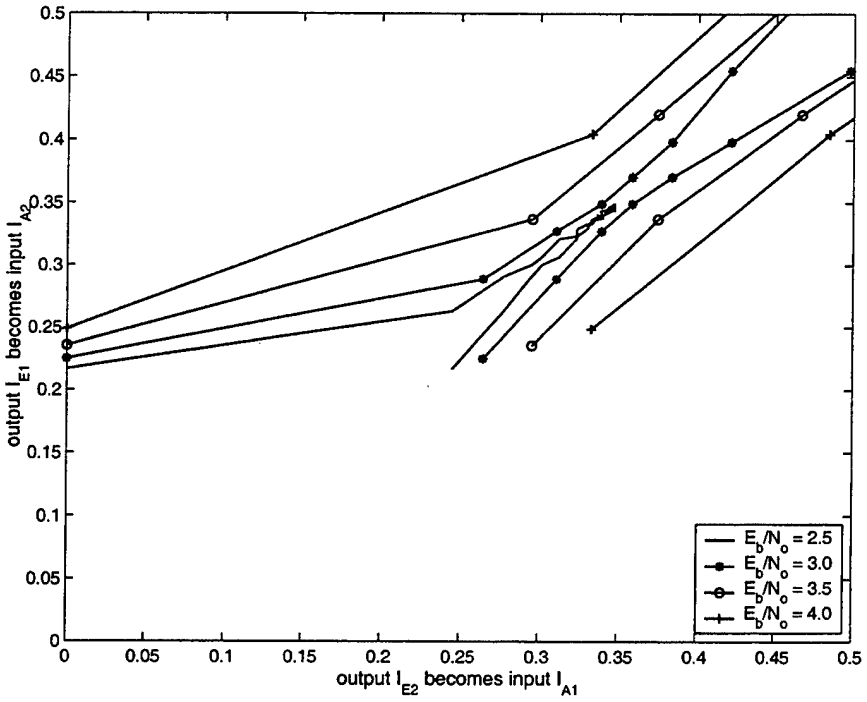


Figure 5-8: Close-up view of the transfer characteristics for a set of E_b/N_o values for the $(64, 57)^2$ TPC

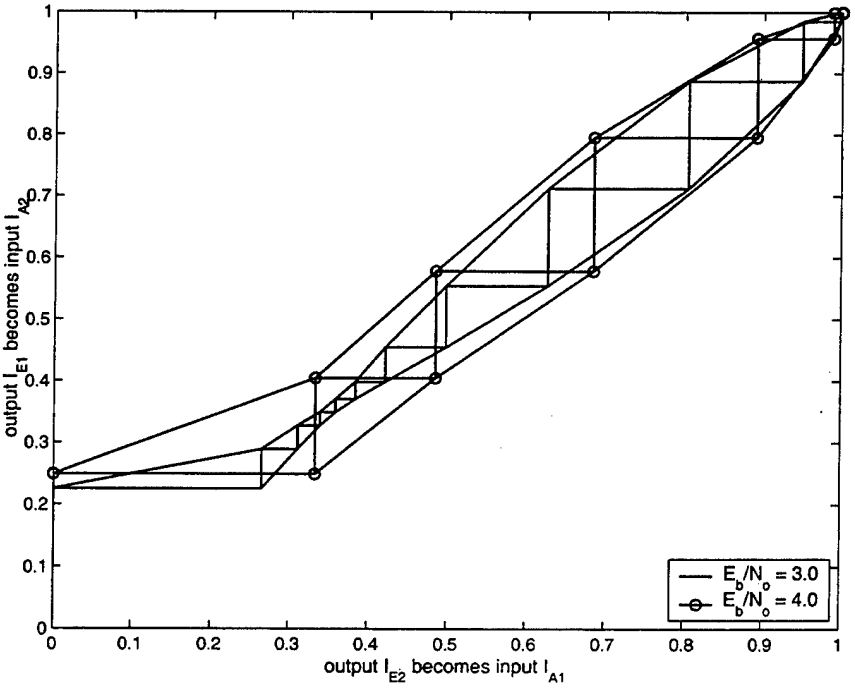


Figure 5-9: EXIT chart and decoding trajectories for the $(64, 57)^2$ TPC at $E_b/N_o =$ 3.0 and 4.0 dB

E_b/N_o values for the $(64,57)^2$ TPC. The $E_b/N_o = 3.0$ dB curve takes many more iterations to decode than the $E_b/N_o = 4.0$ dB curves.

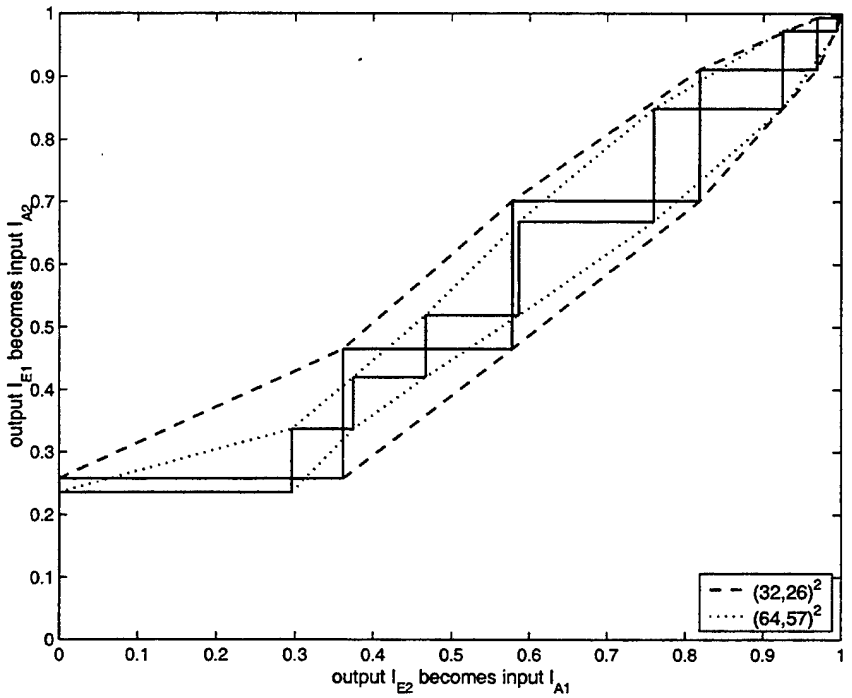


Figure 5-10: EXIT chart and decoding trajectories for the $(64,57)^2$ and $(32,26)^2$ TPCs at $E_b/N_o = 3.5$ dB

Again, we look at two different square TPCs at the same E_b/N_o value. Figure 5-10 shows the EXIT charts for the $(64,57)^2$ and $(32,26)^2$ square TPCs at $E_b/N_o = 3.5$ dB. As shown, the $(32,26)^2$ code has larger initial mutual information value. As a result, the $(32,26)^2$ TPC has a much more wide-open decoding trajectory than the $(64,57)^2$ TPC at this E_b/N_o value. In the following section we will examine results from the TPCS formed from combining the constituent codes used here.

5.2.2 Non-Square codes

Figure 5-11 shows the two non-square TPCs of interest. Again, the curves follow a pattern similar to the BPSK results. The $(32,26)\times(64,57)$ code has a larger initial I_{E_1} value and the $(64,57)\times(32,26)$ code has a larger initial I_{E_2} value. This is due to the fact that the $(32,26)^2$ code is more open at $E_b/N_o = 3.5$ dB than the $(64,57)^2$ TPC as

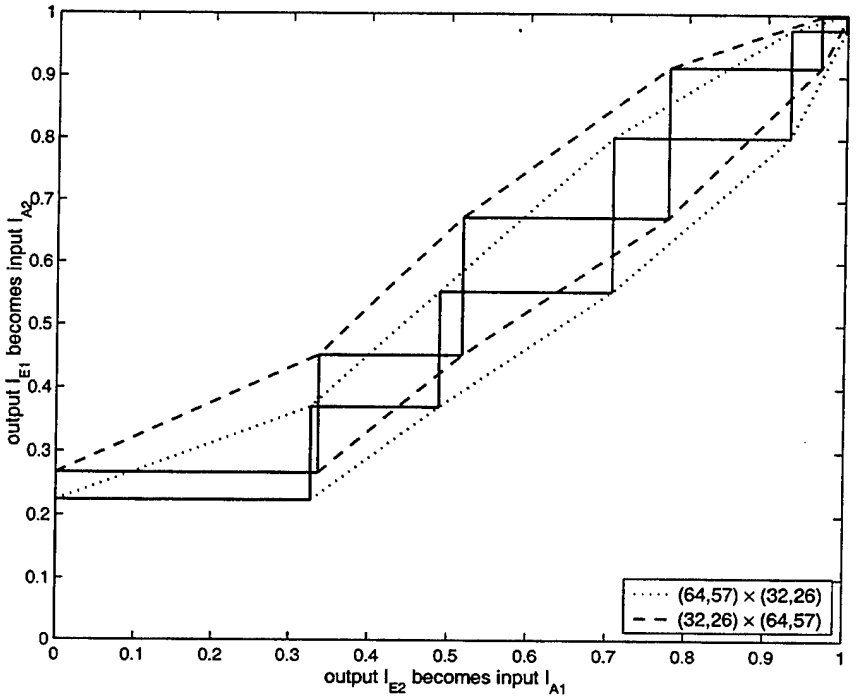


Figure 5-11: EXIT chart and decoding trajectories for the $(64, 57) \times (32, 26)$ and $(32, 26) \times (64, 57)$ TPCs at $E_b/N_o = 3.5$ dB

in Figure 5-10. At this E_b/N_o the decoding trajectories reach a mutual information value of one at almost the same rate; 6.5 full iterations for the $(32, 26) \times (64, 57)$ TPC versus 7 full iterations for the $(64, 57) \times (32, 26)$ TPC. Like the BPSK case, the $(32, 26) \times (64, 57)$ TPC decodes slightly faster than the $(64, 57) \times (32, 26)$ TPC. It should be noted that at this E_b/N_o, these codes have a BER of about 10^{-5}. As with the square codes, the results of the non-square codes are also at E_b/N_o values about 0.5 dB larger than those of the BPSK case. While these EXIT chart differences, observed for the non-square codes in both BPSK and GMSK, do not have much overall impact in the two-dimensional AWGN channel, we will see a larger impact of these differences in both the Rayleigh fading and multidimensional analysis.

Figure 5-12 shows the performance curves for these two non-square codes under varying conditions. First, we can see the curves for the two codes where they have up to 32 full iterations to decode the block. We can also see the curves for these same two codes, except now they are limited to 5 full iterations. We can see that there is no

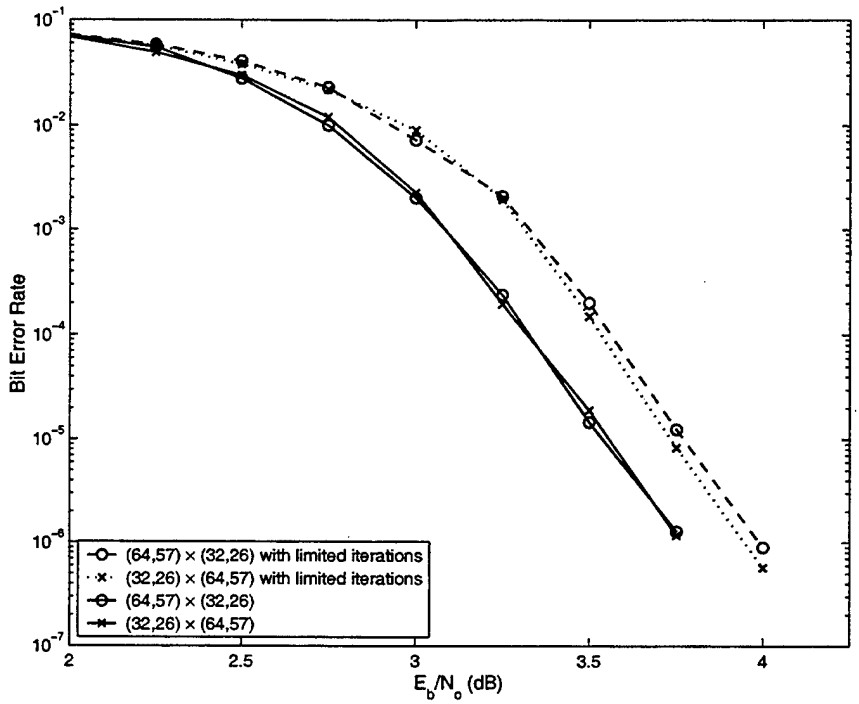


Figure 5-12: Performance Curves of the Non-square codes with GMSK in AWGN

significant performance difference between these two codes in either of the conditions shown at lower E_b/N_o values. However, at higher E_b/N_o values, the $(32, 26) \times (64, 57)$ code begins to perform slightly, by about 0.05 dB, better.

In summary, we have examined four two-dimensional TPCs in both BPSK and GMSK modulated AWGN channels. We have seen similar results in both modulations, differing only in that the GMSK curves occurred at E_b/N_o values about 0.5 dB larger. We have also seen how the TPCs move through their convergence points. Our examination of the two non-square TPCs brought forth some interesting points, such as the having the shorter constituent code on the x-axis increases the decoding speed of the block. We will keep these results in mind as we continue our analysis of the density evolution of TPCs by examining these same codes in a Rayleigh fading channel.

Chapter 6

Density Evolution Analysis Utilizing EXIT Charts: Fading Channels

In [18], ten Brink showed that the EXIT chart was not limited to the Gaussian channel. He applied his ideas to a coherently detected, fully interleaved Rayleigh channel with perfect channel state information at the receiver [18]. This chapter will show that the EXIT chart analysis for TPCs can also be extended to the Rayleigh channel. First, a brief overview of ten Brink's application to the Rayleigh channel will be described. Then, modifications for TPCs will be described and justified. Following the modifications, results will be given for both BPSK and GMSK modulation.

6.1 Turbo Code Assumptions: Rayleigh channel

ten Brink developed the following for his analysis involving Rayleigh fading channels. For the Rayleigh channel, the received signal is given in equation 2.11. The complex fading coefficient is $a_c = a_I + ja_Q$. The variance of the fading term is normalized to $\sigma_a^2 = 1/2$ and the magnitude $a = \sqrt{a_I^2 + a_Q^2}$ is Rayleigh distributed with mean 1. The

L-values are then calculated according to

$$Z = \ln \frac{p(z_c|a_c, x = +1)}{p(z_c|a_c, x = -1)} \tag{6.1}$$

where the conditional PDF of the noisy channel bits, Z, given the transmitted bits X and the fading coefficient a_c is

$$p(z_c|a_c, x) = \frac{1}{2\pi\sigma_n^2} \exp\left[-\frac{|z_c - a_c \cdot x|^2}{2\sigma_n^2}\right]. \tag{6.2}$$

Using the conditional PDF and the Rayleigh channel representation of equation 2.11, the L-value representation can be simplified to

$$Z = \frac{2}{\sigma_n^2} \cdot \mathrm{Re}\{a_c^* \cdot z_c\} = \frac{2}{\sigma_n^2} \cdot (a^2 \cdot x + a \cdot n). \tag{6.3}$$

The additive noise, n is Gaussian distributed with zero mean and variance σ_n^2.

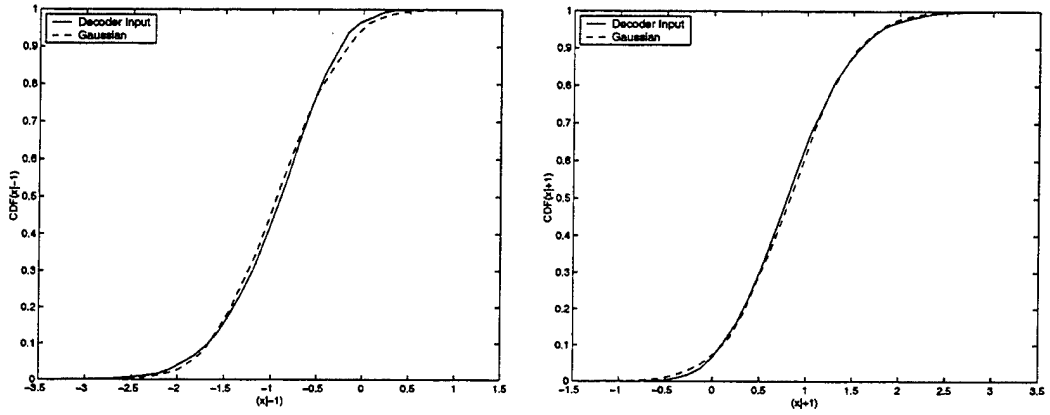
ten Brink followed the same method as used with the Gaussian channel to calculate the mutual information and plot the transfer characteristics. The *a priori* information was again assumed to be Gaussian distributed according to Equation 3.9. The extrinsic information transfer characteristics were computed by Monte Carlo simulation using the no longer Gaussian channel L-values Z [18].

6.2 TPC Assumptions: Rayleigh channel

Section 2.3.2 described how the Rayleigh faded channel would be applied in our analysis. This application and ten Brink's application were quite similar, and no adjustments need to be made regarding this area. However, the assumptions used on the *a priori* input and extrinsic output distributions were not as straight forward.

Initially, we did not think ten Brink's Gaussian assumption on the *a priori* inputs could be used. However, the distributions of the decoder input appeared to be Gaussian. Figure 6-1 shows the CDFs of the Rayleigh faded decoder input conditioned on the transmitted bits for the $(64, 57)^2$ TPC with BPSK at an $E_b/N_o = 7.5$ dB.

For the Rayleigh faded channel, we hoped to see that this perturbation composed of independent Rayleigh fades on each bit could be closely approximated by a Gaussian random variable in the cases examined here. From the figure we can see that a Gaussian random variable of the same mean and variance closely matches the conditioned decoder input.



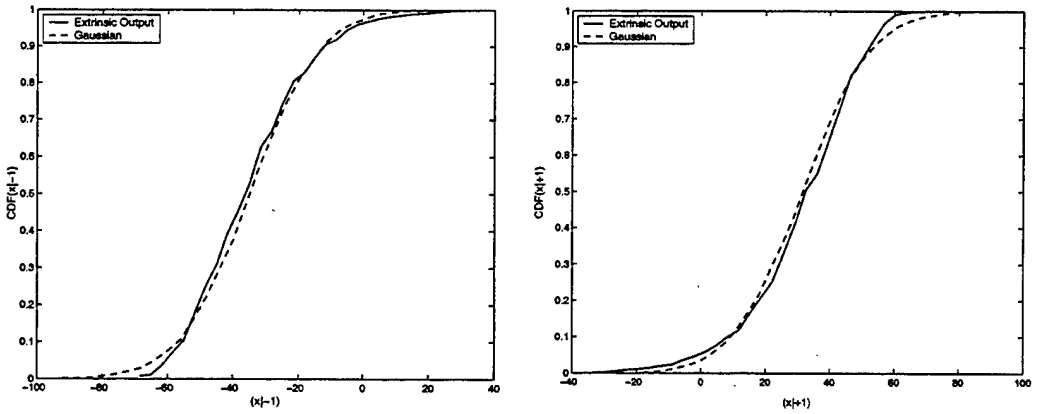
(a) CDF $(x|-1)$ (b) CDF $(x|+1)$

Figure 6-1: Decoder Input Bits for the Rayleigh Channel Conditioned on $X = -1$ and $X = +1$ at $E_b/N_o = 7.5$ dB

Figure 6-2 shows the CDFs of the extrinsic output of the first half iteration of the decoder for a Rayleigh faded input for the same $(64, 57)^2$ TPC. Again, we see that the extrinsic output can be reasonably approximated by a Gaussian random variable of the same mean and variance. From the decoder setup, we know that the extrinsic output becomes the *a priori* input to the next iteration. As a result from this and the observation of [9], we can conclude that a Gaussian assumption is also appropriate for the extrinsic output E of the decoder as well as for the *a priori* input A. The mutual information is calculated according to

$$I(\sigma) = \int_{-\infty}^{+\infty} \frac{e^{-(\xi-\sigma^2/2)^2/2\sigma^2}}{\sqrt{2\pi}\sigma} \cdot (1 - \log_2[1 + e^{-\xi}])d\xi \tag{6.4}$$

for the distributions of both E and A.



(a) CDF $(x|-1)$ (b) CDF $(x|+1)$

Figure 6-2: Extrinsic Output for the Rayleigh Channel Conditioned on $X = -1$ and $X = +1$ at $E_b/N_o = 7.5$ dB

6.3 BPSK Results

We will examine the same four two-dimensional TPCS as examined in the AWGN channel.

6.3.1 Square codes

Figure 6-3 shows the EXIT chart and decoding trajectories for the $(64, 57)^2$ TPC at $E_b/N_o = 7.5$ dB and the $(32, 26)^2$ TPC at $E_b/N_o = 6.5$ dB with both with Rayleigh fading. First, we notice that the $(32, 26)^2$ code has a much more open tunnel than the $(64, 57)^2$ code even though it is at a smaller E_b/N_o value. This tells us that the $(32, 26)^2$ TPC performs much better, that is it reached a BER of 10^{-5}, much sooner that the $(64, 57^2$ TPC in the same channel. We can see that the $(64, 57)^2$ code is near the convergence threshold for the code, in the bottleneck region of the EXIT chart. However, the $(32, 26)^2$ code is nearing the transition from the bottleneck region to the wide-open region. For both codes, we notice that these points occur at E_b/N_o values approximately 4.0 to 5.0 dB greater than the comparable curves in an AWGN channel, see Figure 5-4 on page 52 for the AWGN BPSK results.

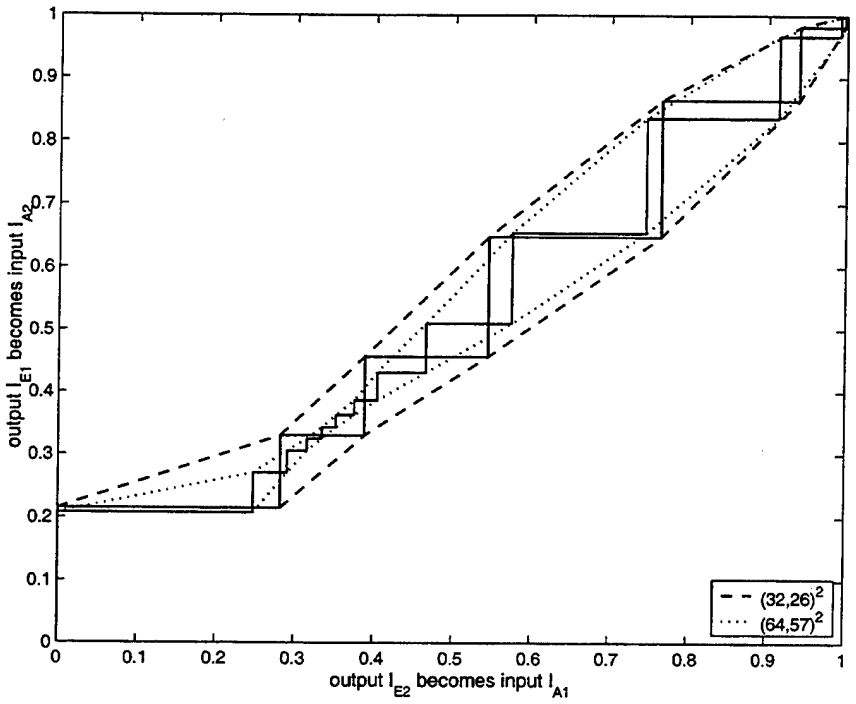


Figure 6-3: EXIT chart and decoding trajectories for the $(64,57)^2$ and $(32,26)^2$ TPCs

6.3.2 Non-Square codes

We continue our analysis by examining the two non-square codes of interest. Figure 6-4 shows EXIT charts for the $(64,57) \times (32,26)$ and $(32,26) \times (64,57)$ TPCs. In the plot, we see that both codes are in the bottleneck region of the EXIT chart, or the waterfall region of the BER curve, at an $E_b/N_o = 6.5$ dB. These non-square TPCs are at a point further along the BER curve than the $(64,57)^2$ TPC was at an E_b/N_o value 1 dB larger. Results such as these show one advantage of examining the many different code combinations available.

As with the AWGN channel results, the $(32,26) \times (64,57)$ TPC has a greater I_{E_1} value while the $(64,57) \times (32,26)$ TPC has a larger I_{E_2} value. The $(32,26) \times (64,57)$ decoded faster than the $(64,57) \times (32,26)$ TPC by more than one full iteration. This is an interesting observation which will be discussed in more detail in chapter 8. As with the square codes, the results for the Rayleigh channel occur at E_b/N_o values of about 4.0 to 5.0 dB greater than for the AWGN channel.

However, even though the $(32,26) \times (64,57)$ TPC decodes faster than the $(64,57) \times$

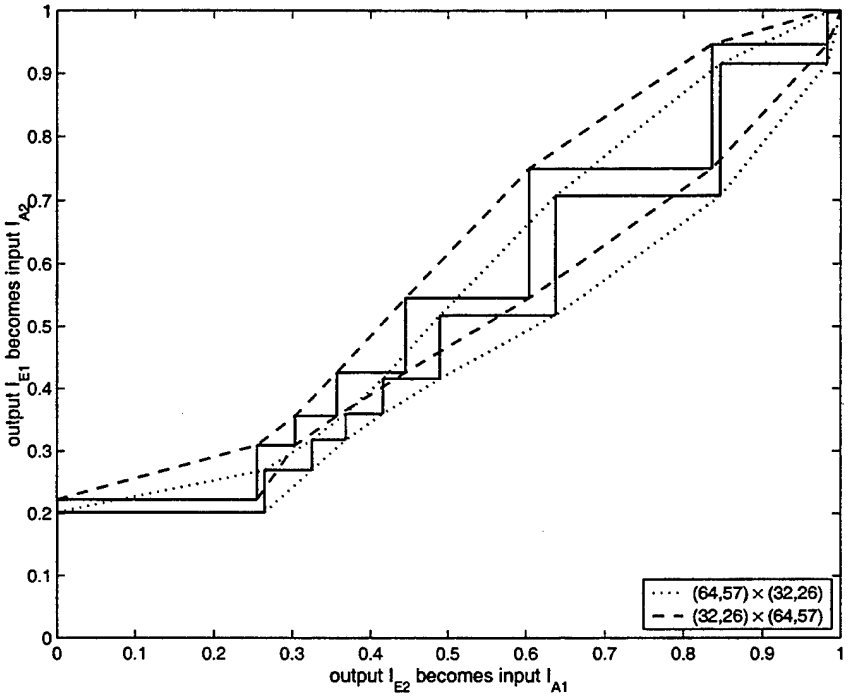


Figure 6-4: EXIT chart and decoding trajectories for the $(64, 57) \times (32, 26)$ and $(32, 26) \times (64, 57)$ TPCs at $E_b/N_o = 6.5$ dB

$(32, 26)$ TPC, there is no significant performance gain when the performance curves are examined. Figure 6-5 shows the performance curves for these two codes with 32 full iterations and 10 full iterations. As shown, no TPC has an advantage over the other under either condition. This differs from the results in AWGN where decoding the shorter dimension first had a slight performance gain with limited iterations.

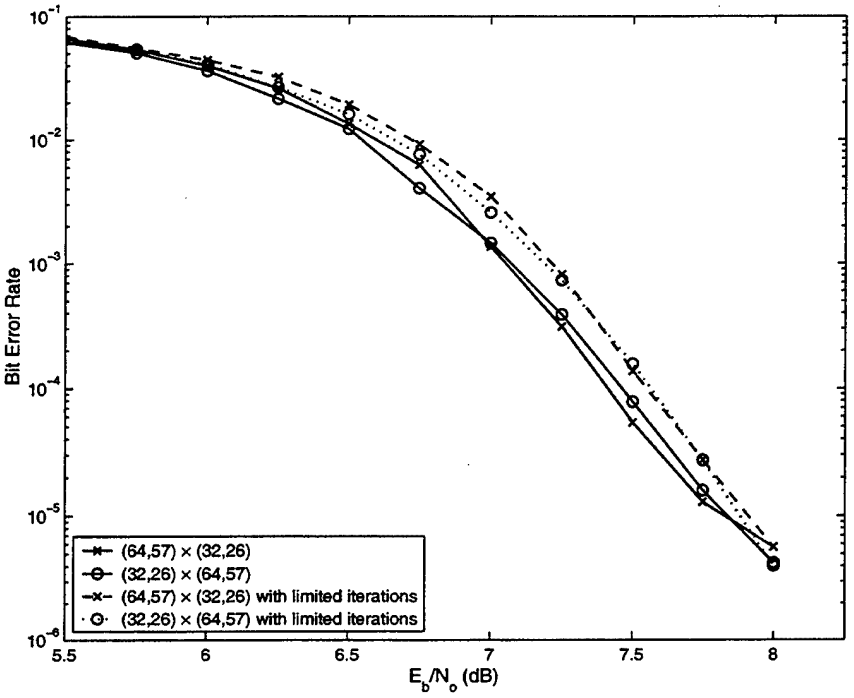


Figure 6-5: Performance Curves for the Non-square codes with BPSK in Rayleigh fading

6.4 GMSK Results

We now examine these same TPCs with GMSK modulation, and we will again see that the results are similar to the AWGN channel, except shifted by several dB.

6.4.1 Square codes

In Figure 6-6 we see EXIT charts and decoding trajectories for the $(64, 57)^2$ and $(32, 26)^2$ TPCs both at an $E_b/N_o = 6.0$ dB. We see once again that the $(64, 57)^2$ code is in the bottleneck region while the $(32, 26)^2$ code is even further into the wide-open region than the BPSK modulation results. However, this time we see these results for both codes at the same E_b/N_o value. It should be noted that these results occur at a point approximately 3.0 dB later than for the AWGN channel with GMSK modulation. Unlike the AWGN channel results where the performance with BPSK was better than GMSK, these codes perform better with GMSK modulation in Rayleigh fading than BPSK modulation.

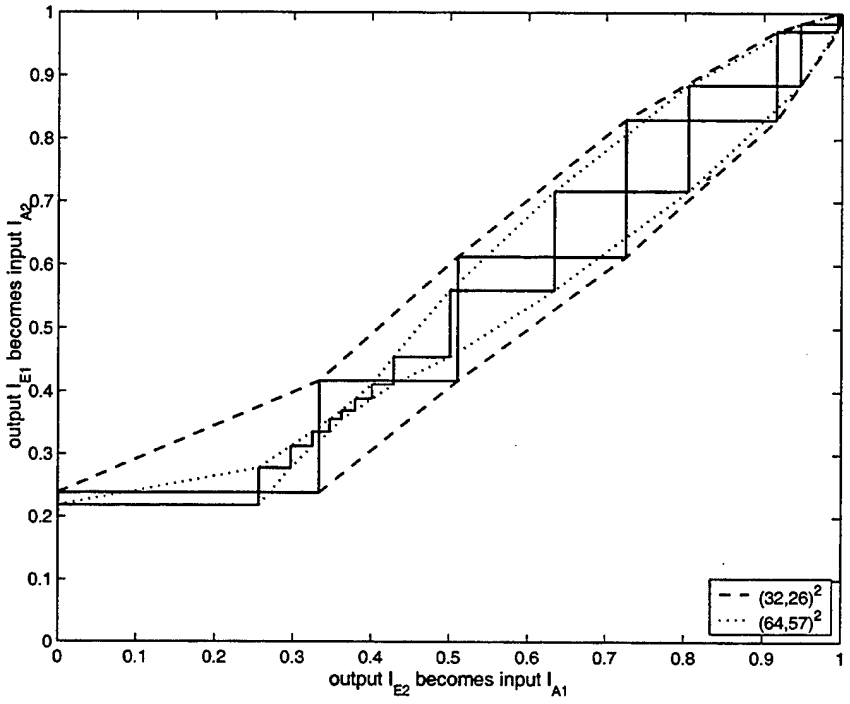


Figure 6-6: EXIT chart and decoding trajectories for the $(64,57)^2$ and $(32,26)^2$ TPCs at $E_b/N_o = 6.0$ dB

6.4.2 Non-Square codes

We continue by examining the non-square codes of interest with Rayleigh fading. Figure 6-7 shows the EXIT charts and decoding trajectories for the $(64,57) \times (32,26)$ and $(32,26) \times (64,57)$ TPCs at an $E_b/N_o = 5.0$ dB. As with the Rayleigh BPSK results, the $(32,26) \times (64,57)$ TPC has a greater I_{E_1} value while the $(64,57) \times (32,26)$ TPC has a larger I_{E_2} value. Also like the Rayleigh BPSK results, the $(32,26) \times (64,57)$ TPC decoded faster than the $(64,57) \times (32,26)$ TPC, by more than one full iteration. Finally, as with the square codes with GMSK, these results occur at a point approximately 3.0 dB later then for the same codes with AWGN GMSK modulation.

In summary, by applying TPCs these four TPCs to a fading environment, we have seen quite similar EXIT chart results as with the AWGN channel - for both BPSK and GMSK modulations. This is not surprising since the decoder operates in the same manner regardless of the modulation and channel used in the analysis. While the received input to the decoder differs, the decoder output remains quite similar.

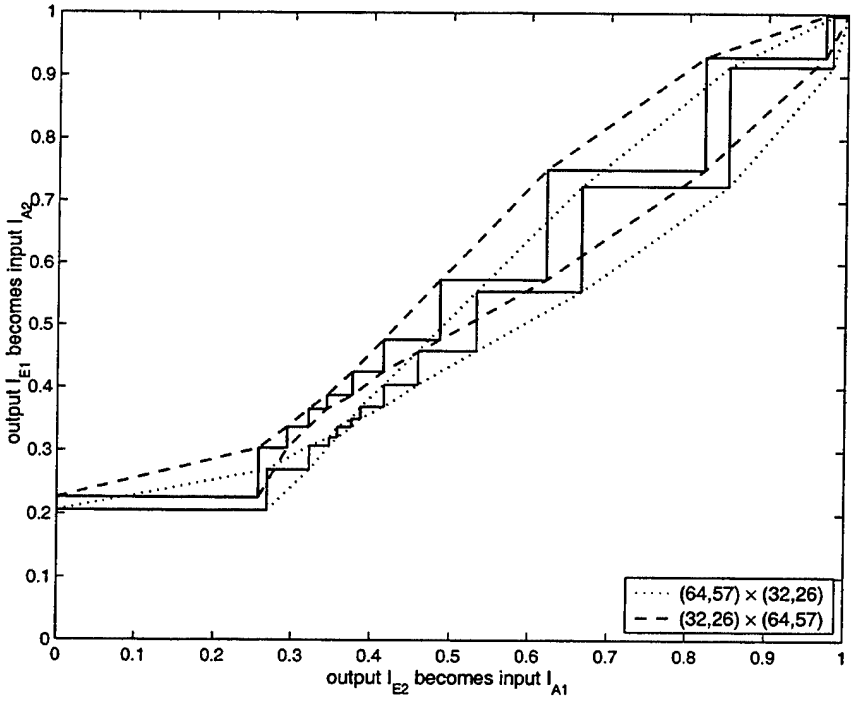


Figure 6-7: EXIT chart and decoding trajectories for the $(64, 57) \times (32, 26)$ and $(32, 26) \times (64, 57)$ TPCs at $E_b/N_o = 5.0$ dB

However, to achieve these similar results, the input E_b/N_o was several dB larger for the Rayleigh cases. Interestingly, these TPCs perform better in the Rayleigh environment with GMSK than BPSK. We can say this because the shift in input E_b/N_o to get similar results is only about 3 dB for the GMSK results, but 4 to 5 dB for the BPSK case. In AWGN, BPSK performs only about 0.5 dB better than GMSK.

In addition, we should note that in the Rayleigh channel results the non-square TPC with the smaller constituent code on the x-axis clearly decoded faster than the comparable TPC with the larger code on the x-axis. This was not necessarily the case in the AWGN channel results, and we will continue to examine this point in the remainder of this thesis. However, as with the GMSK AWGN channel results, no significant performance improvement resulted from the slightly faster decoding of the $(32, 26) \times (64, 57)$ TPC. In the next chapter we will further extend ten Brink's EXIT chart analysis by applying his methods to multidimensional TPCs.

Chapter 7

Density Evolution Analysis Utilizing EXIT Charts: Multidimensional Codes

One of the attractive qualities of TPCs is that they are not limited to two dimensions. In three dimensions, a much larger choice of code rates and block sizes become available for use. The decoding for these codes is performed using the same iterative decoder as in the two-dimensional analyses, but now a full iteration includes three axis iterations, one for each axis of the TPC. As with the two-dimensional TPCs, we wish to examine the density evolution of the three-dimensional TPCs using the EXIT chart.

We will use the $(16, 11)^3$ TPC to introduce the EXIT chart for multidimensional codes. This code has a block size of 4096, like the $(64, 57)^2$ TPC examined previously, and a code rate of 1/3. Since the decoder still operates the same, we continued to use the same mutual information calculations and principles for plotting the transfer characteristics and decoding trajectories as in the two dimensional case. The following will give some results from this code in both BPSK and GMSK modulation for the AWGN and Rayleigh faded channel.

7.1 Transfer Characteristics and EXIT Chart

In general, the EXIT charts are plotted with each axis representing one of the constituent decoders. Therefore for the three-dimensional TPCs, the EXIT charts must be plotted in three dimensions. Since the idea is to graphically show the flow of information through the decoder, a new ordering for the axes needed to be determined. The extension of the EXIT chart plotted I_A and I_E progressing through the x, y, and z axes in order as the decoding progresses through the dimensions of the code. This is illustrated in Figure 7-1.

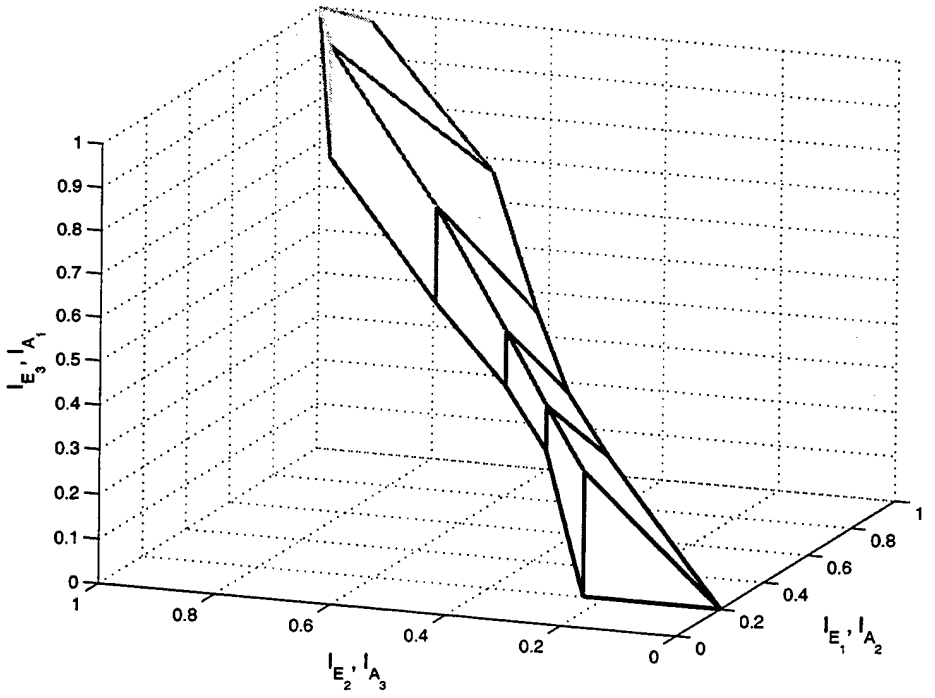


Figure 7-1: $(16, 11)^3$ TPC with BPSK in AWGN at $E_b/N_o = 1.0$ dB

Figure 7-1 and Figure 7-2 give results from the $(16, 11)^3$ code with BPSK modulation at an $E_b/N_o = 1.0$ dB in the AWGN channel. These figures depict the structure of the three-dimensional form of the EXIT chart for multidimensional codes in order to aid in viewing the open plots of the transfer characteristics with decoding trajectories that will be used later. Figure 7-1 shows a reverse view of the EXIT chart while Figure 7-2 shows a front view of the EXIT chart. In the second figure, the front face

has been made transparent so that the two faces which fall behind the front face will be visible.

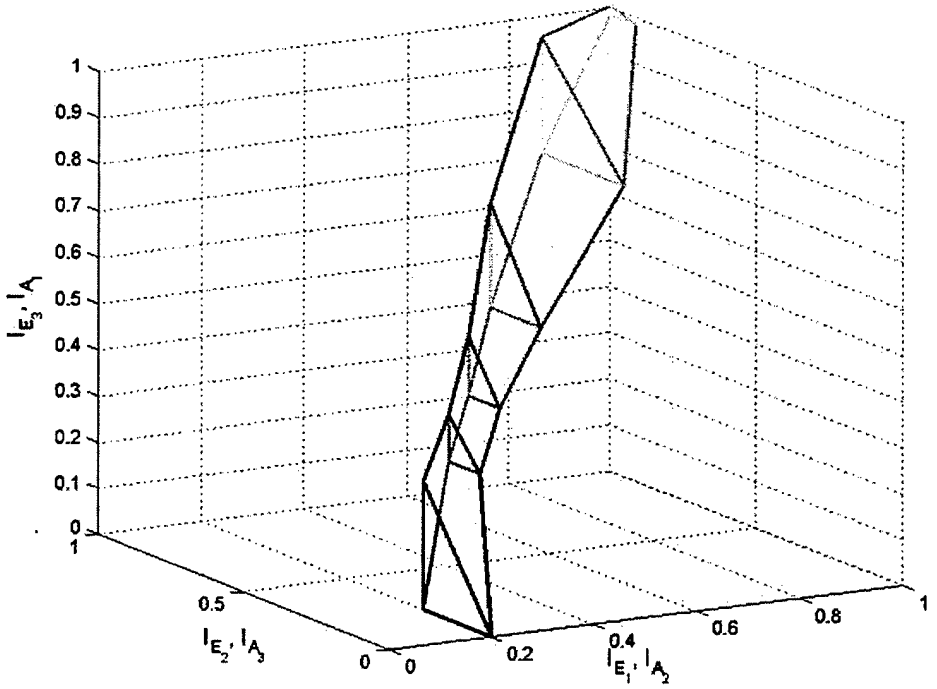


Figure 7-2: $(16, 11)^3$ TPC at $E_b/N_o = 1.0$ dB, front view

We can see from the two figures that the transfer characteristics exist over the range $0 \leq I_A \leq 1$ and that they are monotonically increasing on all three axes. A fairly open tunnel exists between the three faces of the structure, and the code depicted in these plots decoded correctly.

7.2 BPSK Results

As in the two-dimensional case, we begin with BPSK modulation. Figure 7-3 shows the EXIT chart and decoding trajectory for the same code as examined in the previous section, the $(16, 11)^3$ code with BPSK modulation in the AWGN channel at an $E_b/N_o = 1.0$ dB. We again see the EXIT chart plotted from the front view, which is the view that will be used in the remainder of the analysis. Figure 7-4 gives the EXIT chart for this same code at an $E_b/N_o = 2.0$ dB.

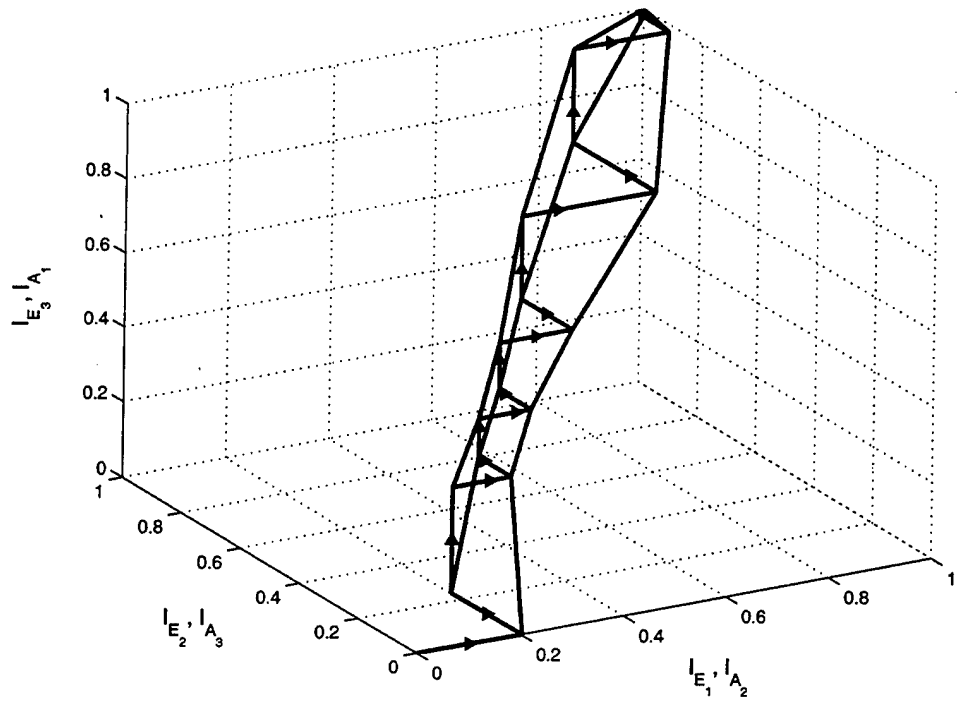


Figure 7-3: EXIT Chart for the $(16, 11)^3$ TPC with BPSK in AWGN at $E_b/N_o = 1.0$ dB

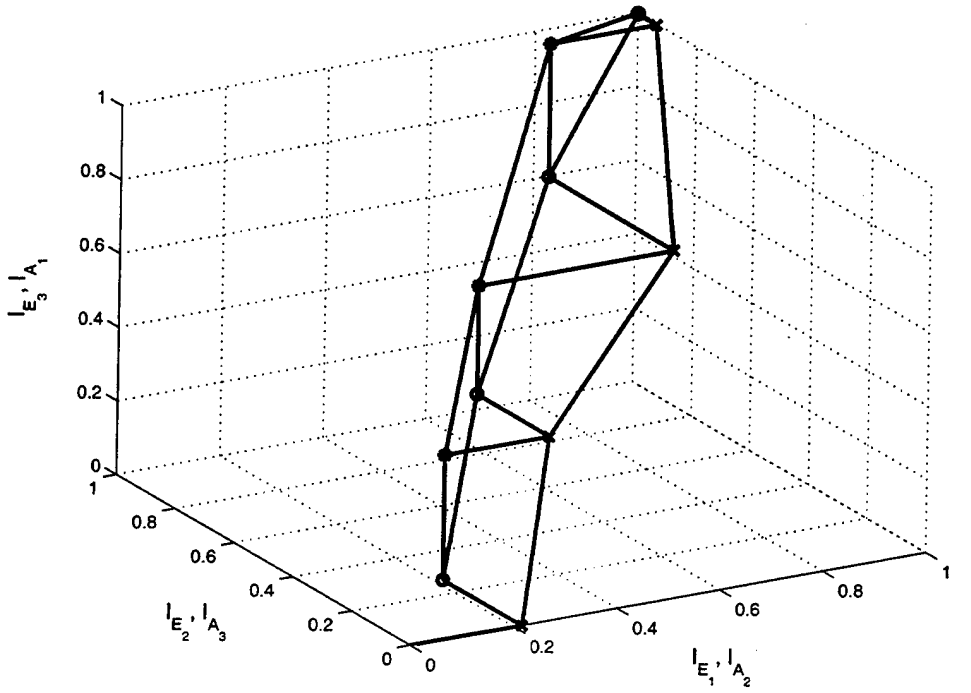


Figure 7-4: EXIT Chart for the $(16, 11)^3$ TPC with BPSK in AWGN at $E_b/N_o = 2.0$ dB

We can also see a couple of the stages of the error rate curve from these two plots. First, from Figure 7-3 we can see that this E_b/N_o is in the bottleneck region of the EXIT chart, or waterfall region of the BER curve. However, when the $E_b/N_o = 2.0$ dB, we can clearly see how the code has moved into the wide-open region of the BER curve. At this point, the block was decoded in four full, or 12 axis, iterations. This code is composed of an extended Hamming codes on each axis, and it performs quite well.

7.3 GMSK Results

We continue our discussion of multidimensional codes by examining the same code with GMSK modulation in the AWGN channel. Figure 7-5 shows the EXIT chart and decoding trajectory at $E_b/N_o = 1.0$ dB and Figure 7-6 shows the EXIT chart and decoding trajectory at $E_b/N_o = 2.0$ dB.

We can see the various stages of the BER curve in these plots as well. Figure 7-5 shows the pinch-off region of the curve, although this is not initially evident. In the two-dimensional case, the transfer characteristics would not reach a mutual information value of one when the block was not decoded correctly. However, due to the an effect from the additional pass through the decoder on the third axis, the mutual information values reach one even for the blocks that are in error. Also, the transfer characteristics do not intersect, as indicated by ten Brink in [17, 18] as a key sign that the decoding process was not complete. One possible reason for this difference is that the decoder found a codeword, as indicated by the mutual information values reaching one, but the codeword was not the correct codeword, and as a result, the block was in error. Further evidence that we are in the pinch-off region is that the convergence point for this code is at $E_b/N_o = 1.2$ dB, a larger dB value than shown here. Figure 7-6 shows the EXIT chart near the transition from the bottleneck region to the wide-open region.

The $(16, 11)^3$ TPC has clearly converged at a lower E_b/N_o value in BPSK modulation than GMSK. From Figures 7-3 and 7-5, we can see that in BPSK this TPC

performs much better than in GMSK at $E_b/N_o = 1.0$ dB. However, by the time the $E_b/N_o = 2.0$ dB, this difference in performance is not large; the code is decoded at BPSK in only about 1 full iteration faster than GMSK. Chapter 8 will continue this discussion.

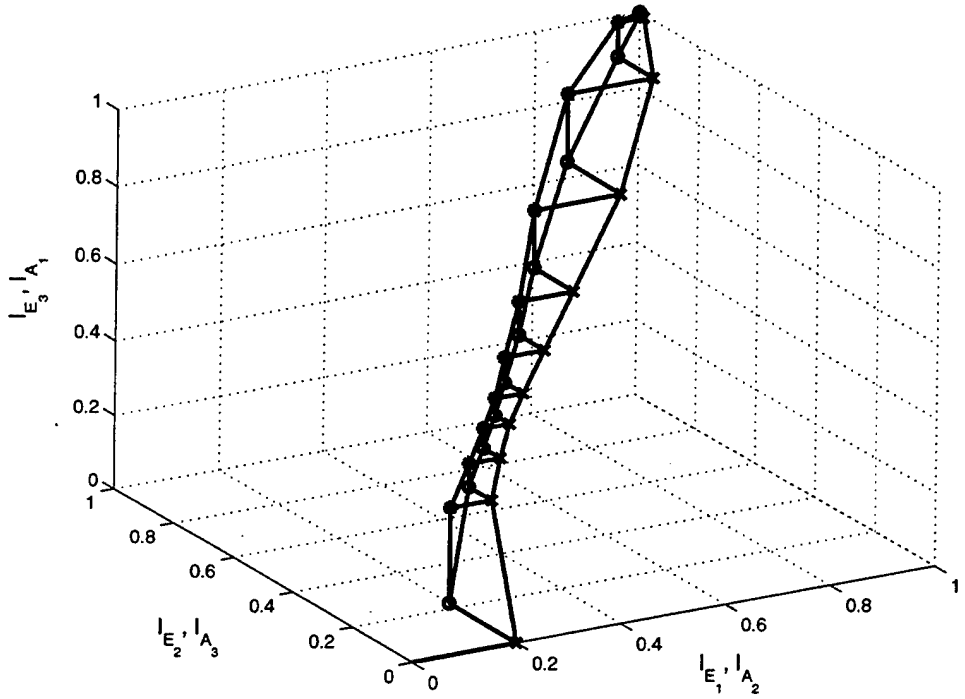


Figure 7-5: EXIT Chart with GMSK in AWGN at $E_b/N_o = 1.0$ dB

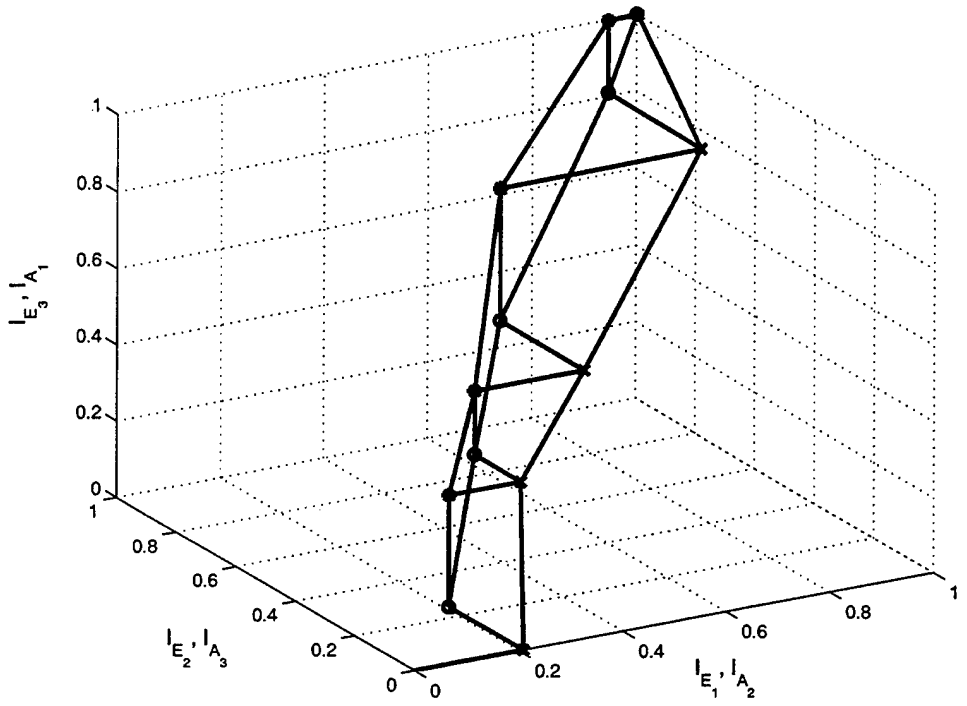


Figure 7-6: EXIT Chart with GMSK in AWGN at $E_b/N_o = 2.0$ dB

7.4 Fading Channels Results

It is possible to extend the analysis of the EXIT charts for multidimensional codes to fading channels as well. Figure 7-7 shows the EXIT charts and decoding trajectories for the $(16, 11)^3$ TPC in a Rayleigh fading channel with both BPSK and GMSK modulation at $E_b/N_o = 1.0$ dB. Interestingly, this code has an almost identical response in Rayleigh fading in the two different modulations. As seen previously, this was not the case for the AWGN channel, where BPSK performed better than GMSK. However, as highlighted in Figure 7-8, we can see that in BPSK, the EXIT chart opens up more beginning near the mutual information values of 0.5 to 0.6 dB and reaches a mutual information value of one faster than in the GMSK case. This faster decoding of the BPSK case is not reflected in the performance curves for this code in Rayleigh fading. The results for both BPSK and GMSK modulations are almost the same. These results will be shown and discussed in more detail in Chapter 8.

In summary, we have shown that the EXIT chart can be extended to multidimen-

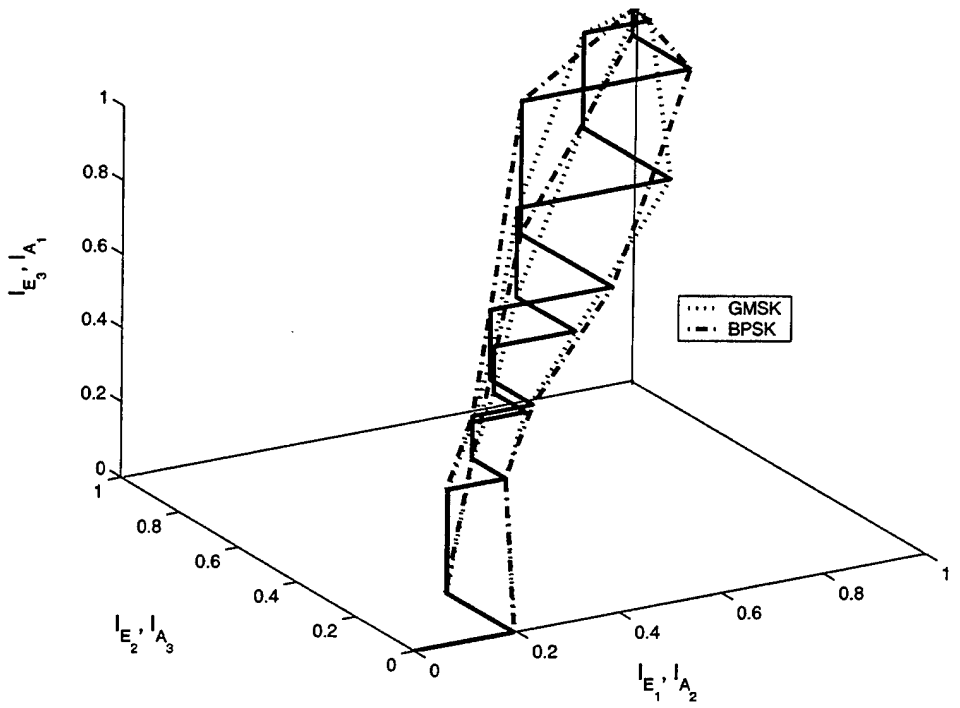


Figure 7-7: EXIT Chart in Rayleigh fading at $E_b/N_o = 1.0$ dB

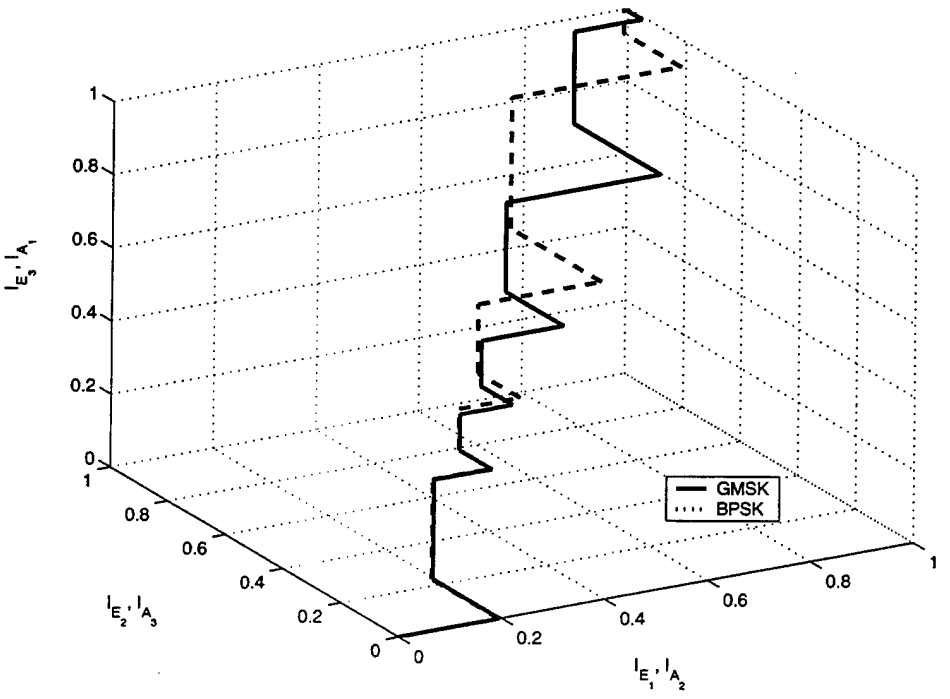


Figure 7-8: Decoding trajectories in Rayleigh fading at $E_b/N_o = 3.5$ dB

sional codes. We developed these methods and examined the results for the $(16, 11)^3$ TPC in both BPSK and GMSK modulation in the AWGN channel. We also showed that this analysis is not limited to the AWGN channel, but can be extended to the Rayleigh fading channel as well. We can also apply the EXIT charts to many other code combinations in three dimensions. We will accomplish this in a code design example in chapter 9. Next, the performance and convergence of TPCs will be examined utilizing EXIT charts.

Chapter 8

Performance and Convergence Analysis Utilizing EXIT Charts

In [18] ten Brink stated that the EXIT chart could be used to estimate the BER of a given code combination after an arbitrary number of iterations. This is useful in that a performance estimate can be made without running long simulations. This chapter will present ten Brink's development for predicting performance using the EXIT chart, in both the AWGN and Rayleigh faded channels. His methods will be extended to the EXIT charts for TPCs, and finally results will be given with BPSK and GMSK modulation in both AWGN and Rayleigh faded channels. In addition, we will give a summary of the convergence analysis completed for the codes examined throughout this thesis.

8.1 BER Analysis: AWGN

We will begin by presenting ten Brink's development for predicting the BER using an EXIT chart. ten Brink stated in [18] that we can estimate the BER using the total soft output information $D = Z + A + E$. To simplify the calculations, both the *a priori* knowledge, A, the extrinsic information, E, were assumed to be Gaussian distributed. As a result, the soft output D is also Gaussian distributed with variance

σ_D^2 and mean $\sigma_D^2/2$. The bit error probability is then

$$P_b \approx \frac{1}{2}\text{erfc}(\frac{1}{\sqrt{2}}\frac{\mu_D}{\sigma_D}) = \frac{1}{2}\text{erfc}(\frac{\sigma_D}{2\sqrt{2}}). \tag{8.1}$$

Assuming that the soft output information is independent, ten Brink then wrote σ_D^2 as

$$\sigma_D^2 = \sigma_Z^2 + \sigma_A^2 + \sigma_E^2. \tag{8.2}$$

Using the representation of Z from Equation 3.3 and

$$\frac{E_b}{N_o} = \frac{1}{2R\sigma_n^2} \tag{8.3}$$

we can write σ_Z^2 as

$$\sigma_Z^2 = (\frac{2}{\sigma_n^2} \cdot \sigma_n)^2 = \frac{4}{\sigma_n^2} = 8R \cdot \frac{E_b}{N_o}. \tag{8.4}$$

Finally, we can rewrite Equation 8.1 as

$$P_b \approx \frac{1}{2}\text{erfc}\left(\frac{\sqrt{8R \cdot \frac{E_b}{N_o} + \sigma_A^2 + \sigma_E^2}}{2\sqrt{2}}\right). \tag{8.5}$$

ten Brink showed that his Gaussian approximation for the BER achieved fairly reliable estimates down to 10^{-3} for the low E_b/N_o region [18].

8.1.1 Modifications for TPCs

A few modifications to ten Brink's development were necessary so that the BER values could be predicted for TPCs. First, through our analysis, we had access to the exact value of σ_n^2 used for each E_b/N_o. Also, from the TPC decoder, we know that we are given the extrinsic soft output information. The variances of this information can be used directly as σ_A^2 and σ_E^2 for each iteration. Therefore, the resulting formulation was used to predict the BER for TPCs.

$$P_b \approx \frac{1}{2}\text{erfc}\left(\frac{\sqrt{4/\sigma_n^2 + \sigma_A^2/2 + \sigma_E^2/2}}{2\sqrt{2}}\right) \tag{8.6}$$

However, one additional modification was also used. In our analysis, we have used averaged values to generate the transfer characteristics and decoding trajectories. Therefore, it was natural to extend the use of averaged values to the BER prediction as well. To incorporate this averaging, we first predicted the BERs for several consecutive blocks of data. Then, these BER values were averaged together and compared to the simulated results from the last block examined.

8.1.2 BPSK BER prediction

Figure 8-1 shows the predicted and simulated BER results as a function of iteration for the $(32, 26) \times (64, 57)$ TPC at $E_b/N_o = 2.5$ dB. The plot in the top half of the figure shows these results for five consecutive blocks of data. As shown, the predicted values do not provide a good fit for the simulated results on a block by block basis. However, when the predicted values for the consecutive blocks were averaged, the fit improves greatly. The plot in the lower half of the figure shows the averaged predicted value along with the simulated result from the fifth block. Even though the predicted values drop off after the 15^{th} half iteration, the curves match well overall. ten Brink's results also showed a close match between the simulated and predicted results [18]. The convergence value of this code is 1.5 dB. Table 8.1.2 shows the results for the first ten half iterations.

Figure 8-2 shows the EXIT chart and decoding trajectory for the $(32, 26) \times (64, 57)$ TPC at $E_b/N_o = 2.5$ dB as well as contours showing when certain BER values were reached. From this figure, you can see how the BER decreases as the mutual information values increase towards one and as the block is decoded correctly. The steady-state average BER value is about 10^{-3}. From this EXIT chart, we can see that this block decoded correctly. As a result, the simulated BER goes to zero when the block is not in error. However, this is not the case with the predicted BER values. The predicted values do not go to zero when the block is decoded correctly.

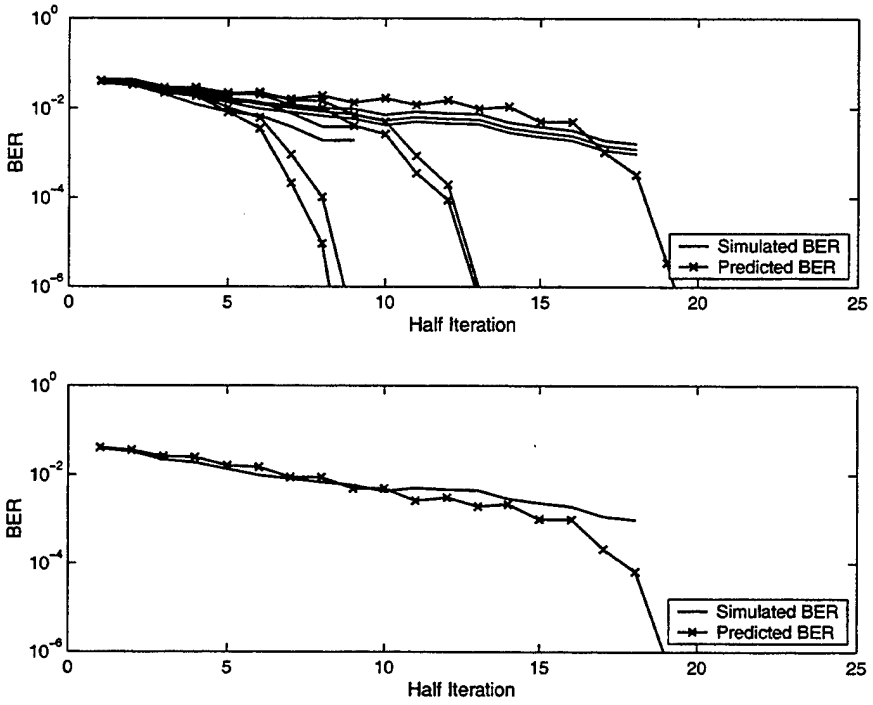


Figure 8-1: Predicted and Simulated BER curves for the $(32, 26) \times (64, 57)$ TPC at $E_b/N_o = 2.5$ dB

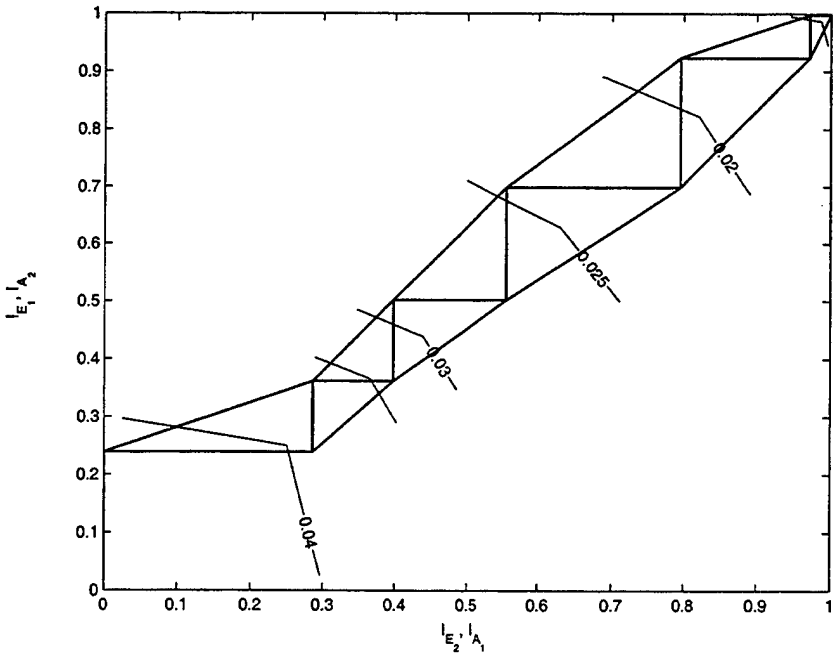


Figure 8-2: EXIT chart with BER contours for $(32, 26) \times (64, 57)$ TPC in AWGN

Table 8.1: Comparison of the predicted and simulated BER values for the $(32, 26) \times (64, 57)$ TPC at $E_b/N_o = 2.5$ dB

Half Iteration	Predicted BER	Simulated BER
1	$4.05e^{-2}$	$3.87e^{-2}$
2	$3.53e^{-2}$	$3.30e^{-2}$
3	$2.57e^{-2}$	$2.19e^{-2}$
4	$2.46e^{-2}$	$1.87e^{-2}$
5	$1.62e^{-2}$	$1.35e^{-2}$
6	$1.48e^{-2}$	$9.80e^{-3}$
7	$8.80e^{-3}$	$8.20e^{-3}$
8	$8.80e^{-3}$	$6.80e^{-3}$
9	$4.90e^{-3}$	$5.90e^{-3}$
10	$4.90e^{-3}$	$4.30e^{-3}$

8.1.3 GMSK BER prediction

To predict the BER with GMSK modulation we continued to the formulation of Equation 8.6. In GMSK modulation, we predicted the BER for the $(64, 57)^2$ TPC at $E_b/N_o = 3.0$ dB. Figure 8-3 shows the simulated and predicted BER curves for three consecutive blocks in the top plot of the figure. The first two blocks were not in error, but the third block was found to be in error, and this errored block dominated the BER results. The bottom plot shows the averaged BER prediction and the simulation results. As shown, there is a close match for the first 25 half iterations. We found that a better BER prediction could be made at E_b/N_o values close to the convergence value of the code, as demonstrated here. The convergence value for this code is at 2.9 dB.

These methods were also extended to the multidimensional codes. One modification for the three-dimensional codes was necessary, given as

$$P_b \approx 3 \times \frac{1}{2}\text{erfc}\left(\frac{\sqrt{4/\sigma_n^2 + \sigma_A^2/2 + \sigma_E^2/2}}{2\sqrt{2}}\right). \tag{8.7}$$

This multiplication accounts for the BER improvements that result from the third axis of the TPC. Figure 8-4 shows the simulated and predicted results for the $(16, 11) \times (32, 26) \times (8, 7)$ TPC with GMSK modulation at $E_b/N_o = 3.0$ dB. Once again, the

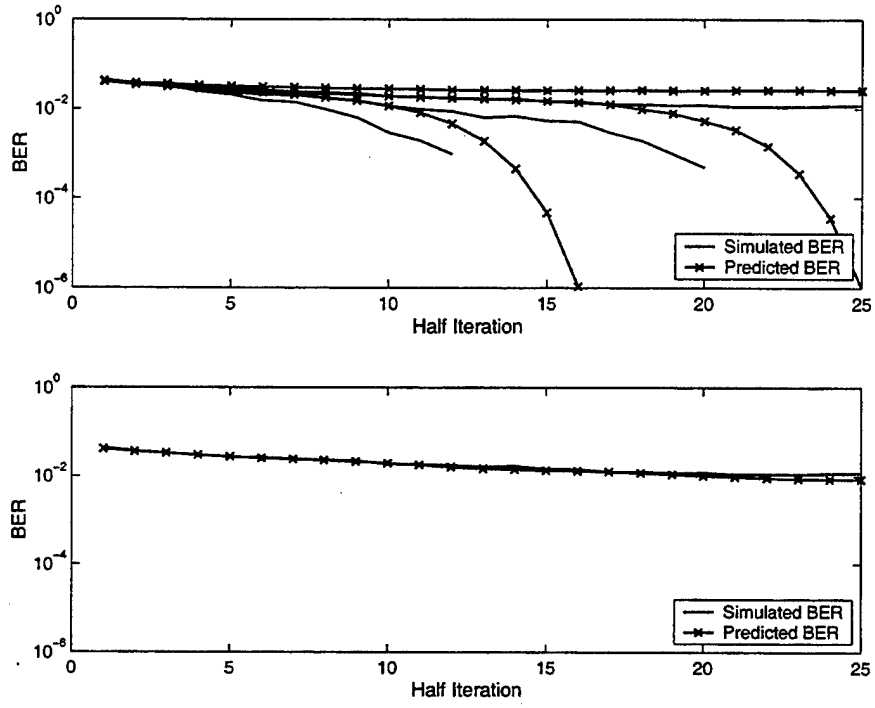


Figure 8-3: Predicted and Simulated BER curves for the $(64, 57)^2$ TPC at $E_b/N_o = 3.0$ dB

top plot shows the results for five consecutive blocks and the bottom curve gives the averaged result. An interesting result demonstrated in these curves is the jagged nature of the predicted BER curve. The unevenness of the curve shows how from one iteration to the next the BER either improves, on the axes with extended Hamming codes, or increases, on the third axis with a SPC code. This unevenness is not present in the predicted BER results from the $(64, 57)^2$ TPC as shown in Figure 8-3, but is present in the results for the $(32, 26) \times (64, 57)$ TPC as shown in Figure 8-1. In the results for the non-square code, we see how the BER improves at different rates for the two different constituent codes. This is an interesting result since the two constituent codes have the same theoretical error correcting capability, or d_{min}.

In the next section we will give ten Brink's development for predicting performance in Rayleigh fading channels.

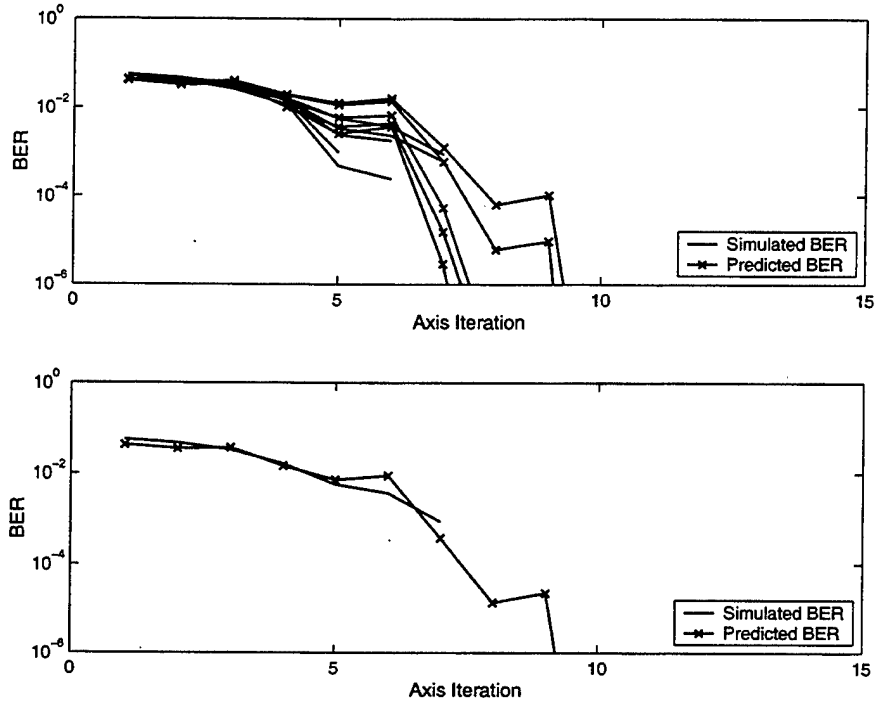


Figure 8-4: Predicted and Simulated BER curves for the $(16, 11) \times (32, 26) \times (8, 7)$ TPC at $E_b/N_o = 3.0$ dB

8.2 BER Analysis: Rayleigh channel

We continue by presenting ten Brink's analysis for estimating the BER in Rayleigh channels. From Equations 6.2 and 6.3, the PDF for the received signal Z was found to be

$$p_Z(t|x) = \frac{\sigma_n^2}{2\sqrt{1+2\sigma_n^2}} \cdot \exp\left[\frac{x \cdot t}{2} - \sqrt{1+2\sigma_n^2} \cdot \frac{|t|}{2}\right]. \tag{8.8}$$

Using the total soft output information $D = Z + A + E$, we can estimate the BER. As in the AWGN case, both the *a priori* knowledge A, the extrinsic information E, were assumed to be Gaussian distributed to simplify the calculating. As a result, the sum $A + E$ is also Gaussian distributed with variance $\sigma_{AE}^2 = \sigma_A^2 + \sigma_E^2$ and mean $\sigma_{AE}^2/2$ with the PDF as

$$p_{AE}(t|x) = \frac{1}{\sqrt{2\pi}\sigma_{AE}} \cdot \exp\left[\frac{1}{2\sigma_{AE}^2} \cdot \left(t - \frac{\sigma_{AE}^2}{2} \cdot x\right)^2\right]. \tag{8.9}$$

The PDF of the total soft output D is found through the following convolution

$$p_D(t|x) = p_Z(t|x) * p_{AE}(t|x). \tag{8.10}$$

Finally, integrating the PDF p_D gives the bit error probability

$$P_b = \int_{\infty}^{0} p_D(t|x=1)dt \tag{8.11}$$

with the closed-form result

$$P_b = \frac{1}{2}\text{erfc}\left(\frac{\sigma_{AE}}{2\sqrt{2}}\right) - \frac{1}{2}\text{erfc}\left(\frac{\sigma_{AE}\sqrt{1+2\sigma_n^2}}{2\sqrt{2}}\right) \times \frac{1}{\sqrt{1+2\sigma_n^2}} \cdot \exp\left[\frac{\sigma_n^2\sigma_{AE}^2}{4}\right]. \tag{8.12}$$

No modifications were necessary to the representation of Equation 8.12 to predicted the BER values for TPCs. However, in this analysis as in the AWGN channel case, we used averaged values to generate the transfer characteristics and decoding trajectories. Therefore, it was natural to extend the use of averaged values to the BER prediction as well. The methods used for averaging the predicted BER were the same as those described in section 8.1.1.

8.2.1 Rayleigh BER prediction

Figure 8-5 gives the predicted and simulated BER curves for the $(32, 26) \times (64, 57)$ TPC at $E_b/N_o = 7.0$ dB in Rayleigh fading with BPSK modulation. In the top plot, we see the curves for five consecutive blocks. The bottom plot gives the result averaged over those five blocks. We see a very close prediction over the first several half iterations. This code has a convergence value of 5.5 dB. Table 8.2.1 lists the results of both the predicted and simulated results.

As in the AWGN case, we can see the unevenness of the predicted BER showing how the BER improvement varies from axis to axis of the TPC as the constituent codes vary on each axis. Figure 8-6 shows the EXIT chart and decoding trajectory for this code as well as contours showing when certain BER values are reached. In the

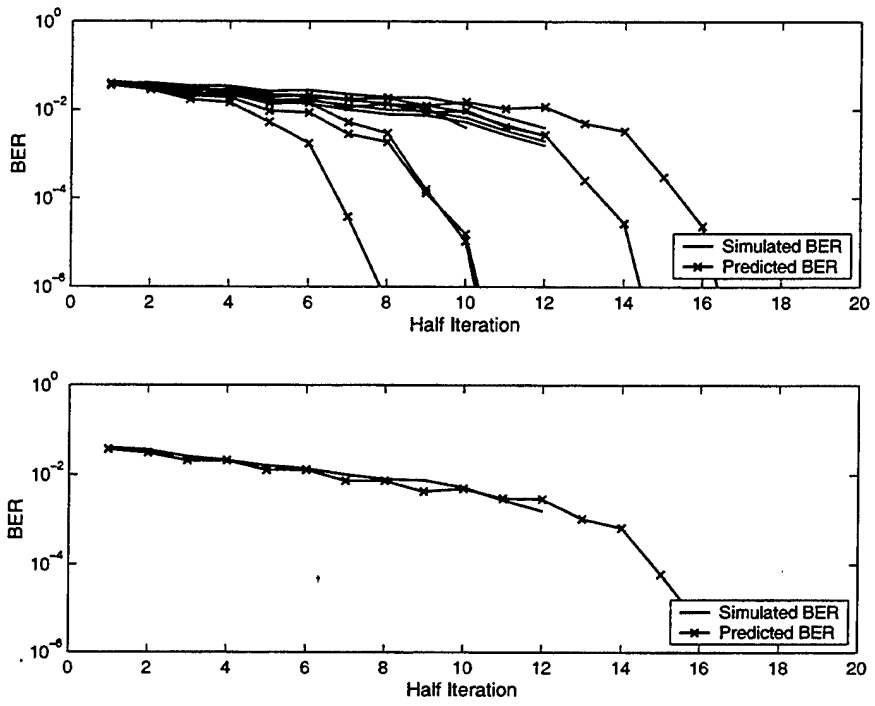


Figure 8-5: Predicted and Simulated BER curves for the $(32, 26) \times (64, 57)$ TPC at $E_b/N_o = 7.0$ dB

figure, we can see how the BER decreases as the mutual information values approach one as the block is decoded correctly. The steady-state average BER is about 10^{-3}.

We will now continue with our discussion of performance and convergence, by examining the convergence results for the various codes examined throughout this thesis.

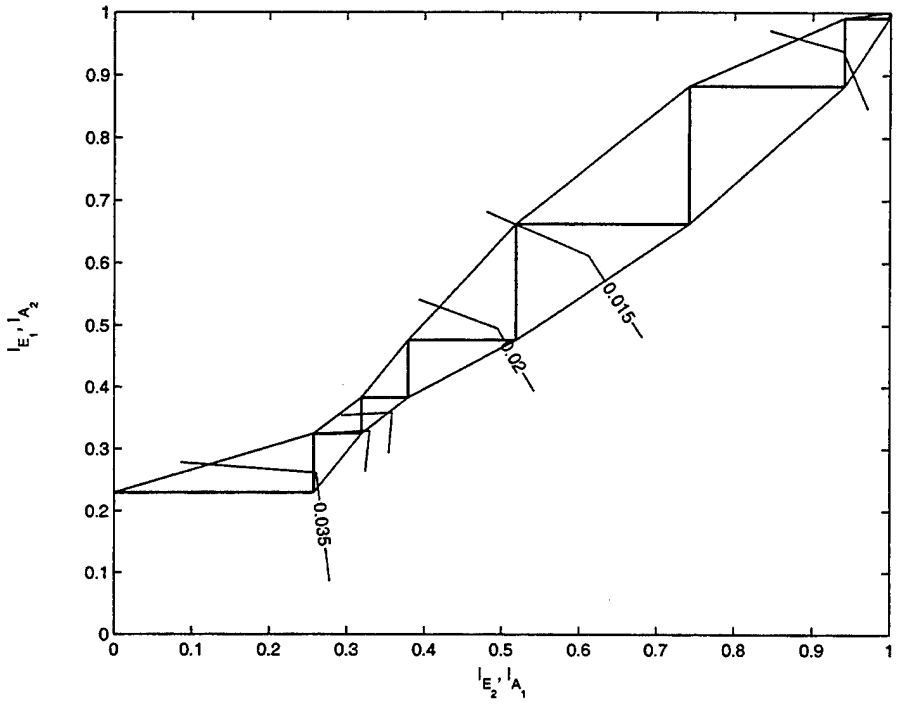


Figure 8-6: EXIT chart with BER contours for $(32, 26) \times (64, 57)$ TPC in Rayleigh Fading

Table 8.2: Comparison of the predicted and simulated BER values for the $(32, 26) \times (64, 57)$ TPC at $E_b/N_o = 7.0$ dB

Half Iteration	Predicted BER	Simulated BER
1	$3.71e^{-2}$	$4.02e^{-2}$
2	$3.08e^{-2}$	$3.57e^{-2}$
3	$2.08e^{-2}$	$2.56e^{-2}$
4	$2.08e^{-2}$	$2.11e^{-2}$
5	$1.28e^{-2}$	$1.60e^{-2}$
6	$1.29e^{-2}$	$1.37e^{-2}$
7	$7.40e^{-3}$	$1.02e^{-2}$
8	$7.40e^{-3}$	$8.00e^{-3}$
9	$4.30e^{-3}$	$7.60e^{-3}$
10	$4.30e^{-3}$	$5.30e^{-3}$

8.3 Convergence Analysis

Previously, we defined convergence as the E_b/N_o value for which the EXIT chart and associated decoding trajectory are first complete. This is when the transfer characteristics exist over the full range of mutual information values, from zero to one, are monotonically increasing over that range, and do not intersect. The convergence values were found for averaged values so that the results would be more general. For these three-dimensional codes, it is when the mutual information of every third axis iteration is monotonically increasing throughout the decoding process.

Table 8.3 gives convergence results for the two-dimensional codes examined throughout the thesis with both BPSK and GMSK modulation in an AWGN channel. Table 8.4 gives the results for those same two-dimensional codes with both BPSK and GMSK modulation in a Rayleigh fading channel. Finally Table 8.5 gives results for the $(16, 11)^3$ TPC with both BPSK and GMSK modulation in both AWGN and Rayleigh fading channels.

Table 8.3: Convergence results for Two-Dimensional TPCs in the AWGN channel

Code	Modulation	E_b/N_o	BER	Full Iterations
$(64, 57)^2$	BPSK	2.4 dB	$7.81e^{-3}$	18
	GMSK	2.9 dB	$2.57e^{-2}$	20
$(64, 57) \times (32, 26)$	BPSK	1.8 dB	$5.39e^{-2}$	17
	GMSK	2.2 dB	$5.58e^{-2}$	23
$(32, 26) \times (64, 57)$	BPSK	1.5 dB	$5.74e^{-2}$	16
	GMSK	2.2 dB	$5.02e^{-2}$	17
$(32, 26)^2$	BPSK	1.1 dB	$5.04e^{-2}$	17
	GMSK	1.7 dB	$6.54e^{-2}$	24

From the tables we can see that in the AWGN channel, convergence is reached first by the $(32, 26)^2$ TPC, and then the other codes as we move up the table from the bottom entry. In AWGN, convergence is reached at a lower E_b/N_o with BPSK modulation than GMSK for each code. However, in the Rayleigh fading channel, we see that the codes converge at a lower E_b/N_o with GMSK modulation than with BPSK. This is shown in Table 8.4. As with the AWGN results, we see the same pattern of convergence start with the $(32, 26)^2$ TPC and move up from the bottom of

Table 8.4: Convergence results for Two-Dimensional TPCs in the Rayleigh channel

Code	Modulation	E_b/N_o	BER	Full Iterations
$(64,57)^2$	BPSK	7.2 dB	$2.39e^{-2}$	21
	GMSK	5.9 dB	$1.89e^{-2}$	18
$(64,57) \times (32,26)$	BPSK	6.1 dB	$2.85e^{-2}$	17
	GMSK	4.8 dB	$3.71e^{-2}$	23
$(32,26) \times (64,57)$	BPSK	5.5 dB	$5.94e^{-2}$	17
	GMSK	4.9 dB	$3.56e^{-2}$	17
$(32,26)^2$	BPSK	4.3 dB	$7.27e^{-2}$	17
	GMSK	3.9 dB	$5.90e^{-2}$	17

Table 8.5: Convergence results for the $(16,11)^3$ TPC

Code	Modulation	Channel	E_b/N_o	BER	Full Iterations
$(16,11)^3$	BPSK	AWGN	0.8 dB	$1.05e^{-1}$	7
	GMSK	AWGN	1.2 dB	$1.82e^{-1}$	7
$(16,11)^3$	BPSK	Rayleigh	3.0 dB	$1.82e^{-2}$	10
	GMSK	Rayleigh	3.0 dB	$1.26e^{-1}$	7

the table in Rayleigh fading. Figure 8-7 shows the order of the convergence results. The convergence thresholds are plotted against the code rates. From the figure, we can see how first the BPSK AWGN convergence thresholds are reached, then the GMSK AWGN, GMSK Rayleigh, and finally the thresholds for BPSK Rayleigh are reached. Quickly, the $(32,26)^2$ TPC has rate 0.66, the $(64,57)^2$ TPC has rate 0.793, and the two non-square codes have rate 0.7236.

Figure 8-8 gives the performance curves for these two-dimensional TPCs in AWGN with BPSK in the left-hand plot and with GMSK in the right-hand plot. From the figure, we can see that the curves for the $(32,26) \times (64,57)$ and $(64,57) \times (32,26)$ TPCs have very similar BER curves within the conditions set for each plot. These curves reach a BER of 10^{-5} at an E_b/N_o of about 1/4 dB sooner than the $(64,57)^2$ TPC. Also, we can see that the convergence thresholds mark the beginning of the waterfall region of the BER curve.

In chapters 5 and 6, several interesting observations were made regarding the convergence and number of iterations required to decode the block correctly for the

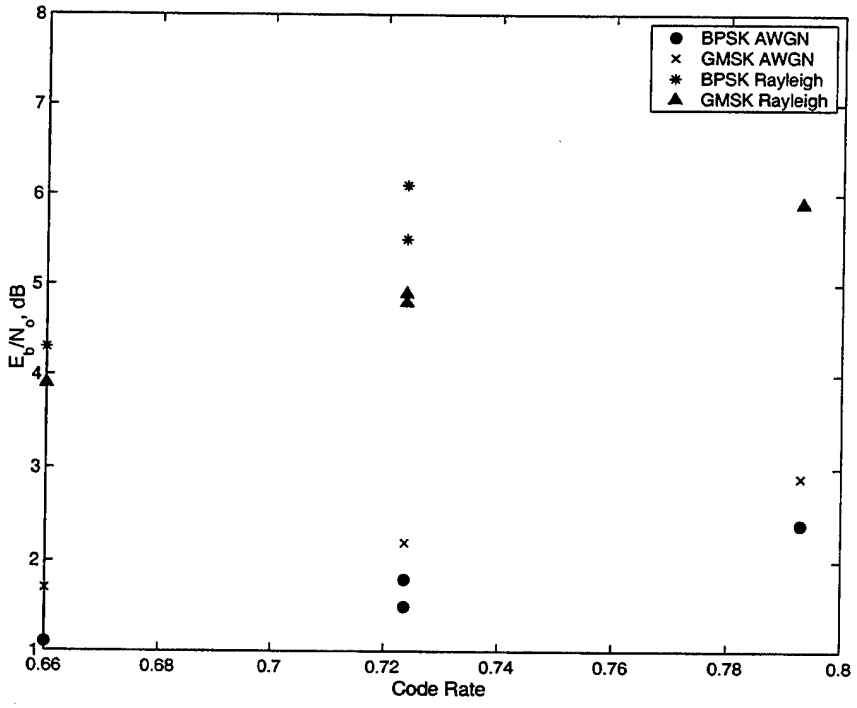


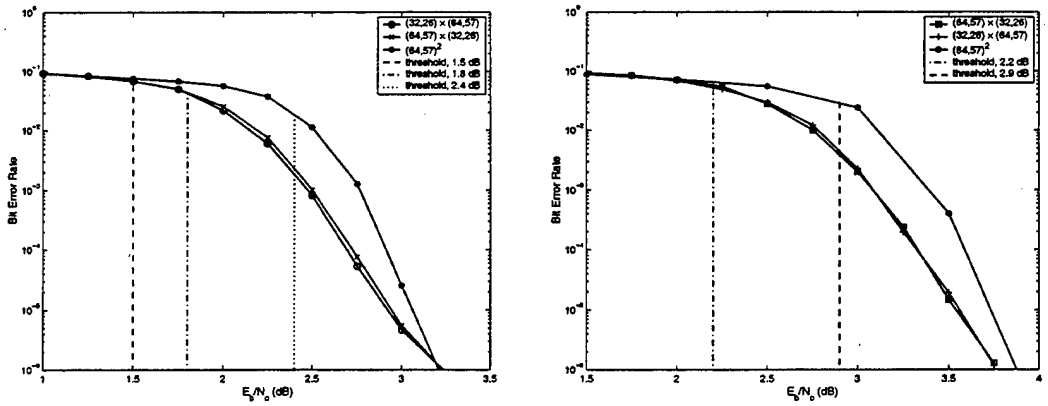
Figure 8-7: Convergence Thresholds versus Code Rate

two non-square codes, the $(32,26) \times (64,57)$ and $(64,57) \times (32,26)$ TPCs. In those two chapters, we found that the $(32,26) \times (64,57)$ TPC decoded from a half to a full iteration faster than the $(64,57) \times (32,26)$ TPC in all modulations and channel conditions examined. From the Table 8.3 and Figure 8-8, we can see that the $(32,26) \times (64,57)$ TPC converges earlier and in fewer iterations with BPSK than the $(64,57) \times (32,26)$ TPC in AWGN. However, both codes converge at the same E_b/N_o in GMSK, with the $(32,26) \times (64,57)$ decoding faster in AWGN. For the Rayleigh fading results, the $(32,26) \times (64,57)$ TPC converges much sooner than the $(64,57 \times (32,26)$ TPC with BPSK, but at the same rate. The two codes converge at almost the same E_b/N_o with GMSK, but at much different rates, as listed in Table 8.4. However, as shown in Figure 8-8, the performance curves for these two codes are quite similar, with no signifcant performance difference even when the number of iterations are limited. Therefore, we can only say that while the one code does tend to decode faster, our results are inconclusive as to recommending one code over the other.

Also noted in chapters 5 and 6 were the dB differences between the E_b/N_o needed

to get certain results. In AWGN, it was noted that the GMSK results occurred about 0.5 dB later than the corresponding BPSK results. From Table 8.3, we can indeed verify that there is about a 0.5 dB difference in the convergence values at BPSK versus GMSK. In chapter 6, it was noted that for GMSK modulation, the Rayleigh fading results occurred about 3 dB later than in AWGN. From the tables, we can see that there is about a 3 dB difference in the convergence values of GMSK. For BPSK modulation, a 4 to 5 dB difference was noted, and from the tables we can verify the about 4 dB difference in the non-square codes to the about 5 dB difference for the $(64, 57)^2$ TPC.

The $(16, 11)^3$ TPC converges at the earliest E_b/N_o value in all cases, but at the same value for both modulation schemes in Rayleigh fading. This is very interesting in that of all codes examined, this TPC is the only code to exhibit this property. This code also decodes the most quickly of the codes examined in this thesis. Figure 8-9 gives the performance curves for the $(16, 11)^3$ TPC with the modulations and channels examined perviously. We can see that in BPSK this code reaches a BER of 10^{-5} at about 1.7 dB, with the GMSK following about 0.5 dB later. With BPSK, this code does perform much better than in GMSK, as discussed and shown in EXIT charts in chapter 7.



(a) BPSK (b) GMSK

Figure 8-8: Performance Curves with AWGN

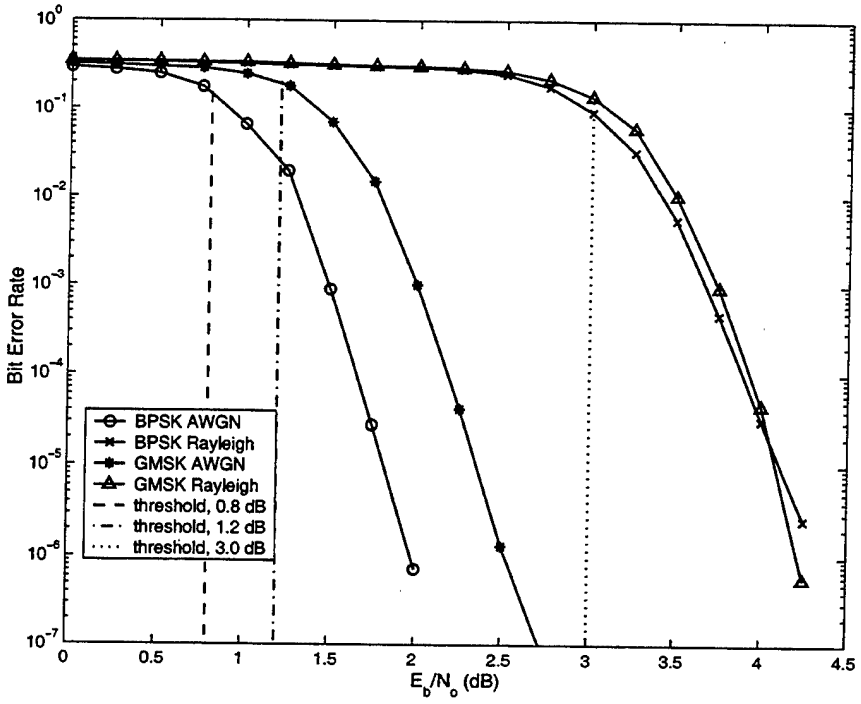


Figure 8-9: Performance Curves for the $(16, 11)^3$ TPC

In summary, we have presented the methods for BER prediction using EXIT charts and we have seen results of these predictions as well. The shape of the predicted curves closely matched the simulated results, which is useful so that we need not run long simulations to get an initial estimate on the BER performance. However, it should be noted that the BER predictions work best in the region of high BER and low E_b/N_o, which is usually near the convergence values. The convergence analysis presented here verified many of the results given in previous chapters. This analysis is also useful on its own for getting an estimate as to the shape of the overall performance curve.

The next chapter will use the methods developed and presented throughout this thesis in a code design example.

Chapter 9

Code Design with EXIT Charts

ten Brink listed code design as one of the ultimate uses for EXIT charts. Using the methods developed and discussed in this thesis, we will design a rate one-half TPC in this chapter. We chose to design a rate one-half code for a couple of reasons. First, with the given constituent codes, a square two-dimensional code does not exist at this rate. Codes at this rate are not only desirable for use in satellite communications, but are commonly used in a variety of other analysis as well. For example, ten Brink developed the EXIT chart using a rate one-half turbo code.

9.1 Constituent Codes

In our analysis, we have used the TPC decoder supplied by ECC and AHA. As a result, the constituent codes we have available for use is limited. For the x-axis and y-axis of the code, we are limited to the extended Hamming codes listed in the first two columns of Table 9.1. For the z-axis, any of the extended Hamming codes from the first two columns or the SPC codes listed in the third column of the table are available for use.

After combining the various constituent codes, several code combinations were found that would meet the code rate requirement. However, we were also concerned about the block size of the code. We chose to examine the codes of two different block sizes. First, the codes with block size 4096 because that is the block size of the

Table 9.1: Constituent Codes Available

X-axis codes	Y-axis codes	Additional Z-axis codes
$(128, 120)$	$(128, 120)$	$(128, 127)$
$(64, 57)$	$(64, 57)$	$(64, 63)$
$(32, 26)$	$(32, 26)$	$(32, 31)$
$(16, 11)$	$(16, 11)$	$(16, 15)$
$(8, 4)$	$(8, 4)$	$(8, 7)$
		$(4, 3)$

$(64, 57)^2$ TPC examined throughout this thesis. Also, there are three codes at this block size that we will be able to compare. In addition, we will examine codes with block size 8192. There are several codes to compare at this block size, and this will allow us to compare results of varying block sizes as well. See Table 9.2 for the code combinations with rate one-half.

Table 9.2: Rate One-Half Codes

Code	Rate	Block size
$(128, 120) \times (16, 11) \times (4, 3)$	0.483	8192
$(16, 11) \times (128, 120) \times (4, 3)$	0.483	8192
$(32, 26) \times (16, 11) \times (8, 7)$	0.488	4096
$(16, 11) \times (32, 26) \times (8, 7)$	0.488	4096
$(32, 26) \times (32, 26) \times (4, 3)$	0.495	4096
$(64, 57) \times (32, 26) \times (16, 11)$	0.497	32768
$(64, 57) \times (16, 11) \times (32, 26)$	0.497	32768
$(32, 26) \times (64, 57) \times (16, 11)$	0.497	32768
$(32, 26) \times (16, 11) \times (64, 57)$	0.497	32768
$(16, 11) \times (64, 57) \times (32, 26)$	0.497	32768
$(16, 11) \times (32, 26) \times (64, 57)$	0.497	32768
$(32, 26) \times (16, 11) \times (16, 15)$	0.524	8192
$(16, 11) \times (32, 26) \times (16, 15)$	0.524	8192
$(64, 57) \times (16, 11) \times (8, 7)$	0.536	8192
$(16, 11) \times (64, 57) \times (8, 7)$	0.536	8192

In total, there are three TPCs of interest with block size 4096 and six TPCs with block size 8192. These TPCs will be examined in more detail in the following.

9.2 Convergence Analysis

We will begin with an examination of the convergence points for the nine TPCs of interest. The convergence values were determined from the E_b/N_o value where the EXIT chart and decoding trajectory were first complete. For these three-dimensional codes, it is when the mutual information of every third axis iteration is monotonically increasing throughout the decoding process. Table 9.3 gives convergence results for codes of interest in BPSK and GMSK modulations with the AWGN channel and Table 9.4 gives the results in BPSK and GMSK modulation with the Rayleigh faded channel.

Table 9.3: Convergence results for the AWGN channel

Code	Modulation	E_b/N_o	BER	Full Iterations
$(16,11) \times (32,26) \times (8,7)$	BPSK	0.9 dB	$7.62e^{-2}$	13
	GMSK	1.5 dB	$6.55e^{-2}$	10
$(32,26) \times (16,11) \times (8,7)$	BPSK	0.7 dB	$8.93e^{-2}$	13
	GMSK	1.3 dB	$9.23e^{-2}$	12
$(32,26) \times (32,26) \times (4,3)$	BPSK	1.2 dB	$3.84e^{-2}$	21
	GMSK	1.4 dB	$8.16e^{-2}$	21
$(64,57) \times (16,11) \times (8,7)$	BPSK	1.5 dB	$1.15e^{-2}$	11
	GMSK	1.9 dB	$6.25e^{-2}$	15
$(16,11) \times (64,57) \times (8,7)$	BPSK	1.5 dB	$3.31e^{-2}$	9
	GMSK	1.9 dB	$3.75e^{-2}$	14
$(128,120) \times (16,11) \times (4,3)$	BPSK	2.1 dB	$5.51e^{-2}$	6
	GMSK	2.3 dB	$2.58e^{-2}$	9
$(16,11) \times (128,120) \times (4,3)$	BPSK	1.8 dB	$2.77e^{-2}$	9
	GMSK	2.4 dB	$2.91e^{-2}$	9
$(32,26) \times (16,11) \times (16,15)$	BPSK	1.3 dB	$2.144e^{-2}$	9
	GMSK	1.7 dB	$4.048e^{-2}$	10
$(16,11) \times (32,26) \times (16,15)$	BPSK	1.2 dB	$1.802e^{-2}$	9
	GMSK	1.4 dB	$7.422e^{-2}$	12

From the tables, we can see that the pair of TPCs composed from the $(16,11)$, $(32,26)$, and $(8,7)$ constituent codes for the 4096 block size TPCs, and the pair composed from the $(16,11)$, $(32,26)$, and $(16,15)$ constituent codes for the 8192 block size TPCs converge earlier, that is at the lowest E_b/N_o value, than the other codes listed in the two tables. The 4096 block size TPCs composed of $(16,11)$, $(32,26)$ and

Table 9.4: Convergence results for the Rayleigh channel

Code	Modulation	E_b/N_o	BER	Full Iterations
$(16,11) \times (32,26) \times (8,7)$	BPSK	3.4 dB	$8.82e^{-2}$	16
	GMSK	3.3 dB	$6.44e^{-2}$	12
$(32,26) \times (16,11) \times (8,7)$	BPSK	3.6 dB	$5.24e^{-2}$	13
	GMSK	3.3 dB	$6.78e^{-2}$	12
$(32,26) \times (32,26) \times (4,3)$	BPSK	4.1 dB	$5.11e^{-2}$	16
	GMSK	3.8 dB	$7.06e^{-2}$	15
$(64,57) \times (16,11) \times (8,7)$	BPSK	4.8 dB	$4.11e^{-2}$	13
	GMSK	3.9 dB	$6.75e^{-2}$	15
$(16,11) \times (64,57) \times (8,7)$	BPSK	4.5 dB	$2.79e^{-2}$	9
	GMSK	4.1 dB	$3.81e^{-2}$	10
$(128,120) \times (16,11) \times (4,3)$	BPSK	5.0 dB	$2.21e^{-2}$	9
	GMSK	4.7 dB	$2.75e^{-2}$	8
$(16,11) \times (128,120) \times (4,3)$	BPSK	5.0 dB	$2.66e^{-2}$	9
	GMSK	4.6 dB	$2.35e^{-2}$	8
$(32,26) \times (16,11) \times (16,15)$	BPSK	3.8 dB	$7.622e^{-2}$	11
	GMSK	3.9 dB	$3.447e^{-2}$	10
$(16,11) \times (32,26) \times (16,15)$	BPSK	4.1 dB	$3.140e^{-2}$	10
	GMSK	3.6 dB	$6.143e^{-2}$	11

$(8,7)$ converge the earliest when all nine TPCs are compared overall. For the 8192 block size codes, the TPCs composed of the $(16,11), (32,26)$ and $(16,15)$ constituent codes converge earliest. Therefore, we can reduce the number of TPCs of interest from nine to four. The four remaining TPCs are: $(16,11) \times (32,26) \times (8,7)$ and $(32,26) \times (16,11) \times (8,7)$ with block size 4096, and $(16,11) \times (32,26) \times (16,15)$ and $(32,26) \times (16,11) \times (16,15)$ with block size 8192. From Table 9.3 we can see that the two 4096 block size codes converge about 0.25 dB earlier in AWGN than the two 8192 block size codes. In the Rayleigh channel, we can see from Table 9.4 that the 4096 block size codes converge about 0.5 dB earlier than the 8192 block size codes.

Figure 9-1 shows the convergence thresholds for all the codes listed in the two tables. In most applications, we desire a code that converges in a small number of iterations with a low E_b/N_o value. This places our area of interest in the lower left-hand corner of the plot. The convergence points for the codes we chose to examine are represented by the filled in symbols. As shown, the codes we will examine approach

the area of interest for the respective channel conditions. These four codes will be examined in more detail by examining their EXIT chart performance with AWGN in the following.

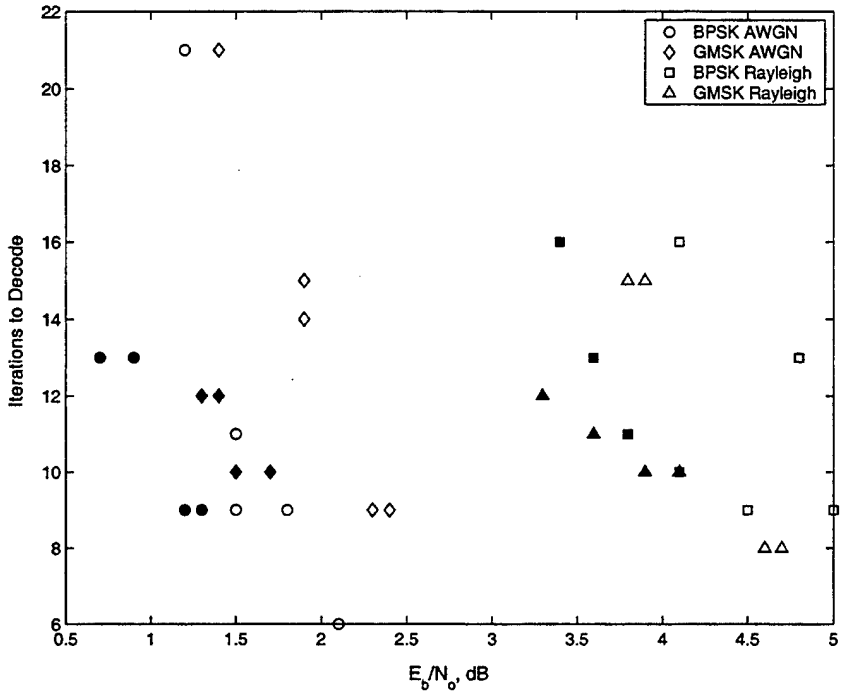


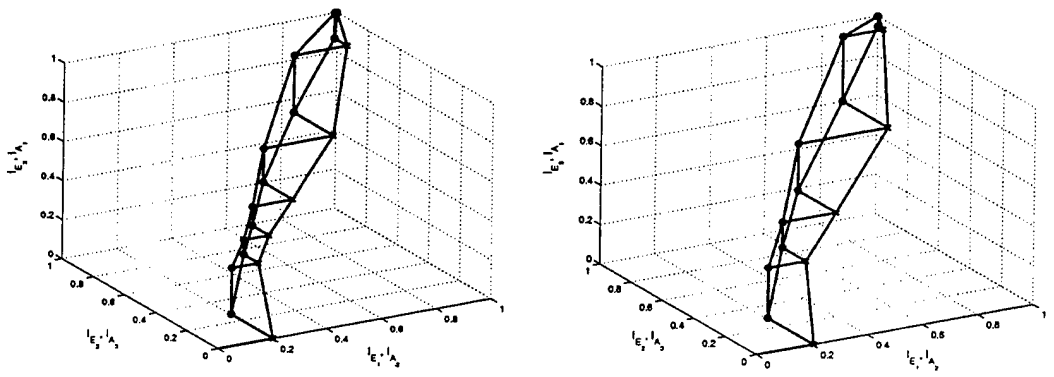
Figure 9-1: Convergence Thresholds versus Iterations to Decode for various codes, with codes for further examination highlighted

9.3 EXIT Chart Analysis in the AWGN Channel

Our code design will continue by examining the EXIT chart results for the four remaining codes with GMSK modulation in the AWGN channel. We will examine results at $E_b/N_o = 2.0$ and 2.5 dB for each of the four codes. At these E_b/N_o values, all of the codes have already converged, so we will primarily be looking at how many iterations are required for the decoding process to be completed and at the openness of the tunnel created between the transfer characteristics. By examining the two different E_b/N_o values we can see how the codes act first in the bottleneck region and then later in the waterfall region, which is the region where the codes will reach a

BER of 10^{-5}. We will not only compare the results within the same block size, but across the two different block sizes as well.

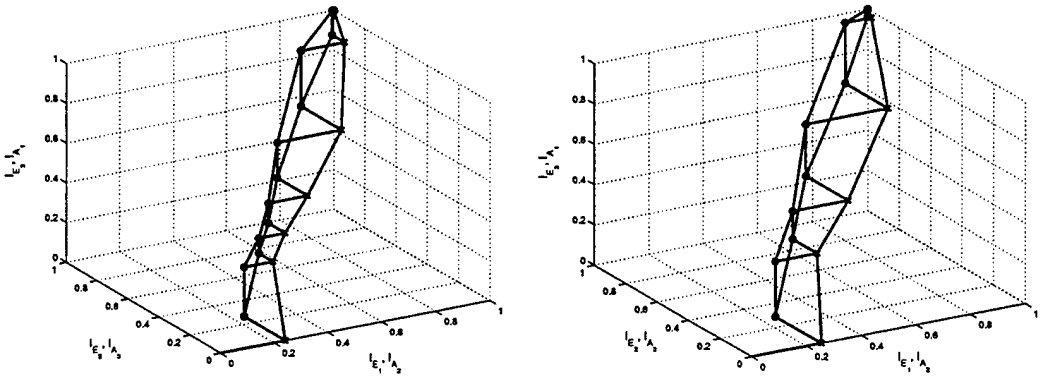
Figures 9-2 to 9-5 give the EXIT chart results for the four codes. In all cases, the left-hand plots, a, show an EXIT chart near the bottleneck region, while the right-hand plots, b, show an EXIT chart moving toward the waterfall region, the region of low BER. The tunnels created by the transfer functions of the left-hand plots are not very open, especially in the case of the $(32, 26 \times (16, 11) \times (16, 15)$ TPC as shown in Figure 9-4. The tunnels of the right-hand plots are much more open, and as a result the TPCs are decoded much faster at the greater E_b/N_o value. Figure 9-2 highlights this increase in the decoding rate, where at 2.0 dB 6 full , or 18 axis, iterations, are required to decode the block, but only 4 full, or 12 axis, iterations are required at 2.5 dB. Additionally, we should note that the TPCs with the $(16, 11)$ TPC on the x-axis have much larger initial I_E values than the other codes which have a $(32, 26)$ on the x-axis. While this larger initial I_E value does not speed up the decoding process in the 4096 block size codes, it does seem to have an impact on the 8192 block size codes. The $(16, 11) \times (32, 26) \times (16, 15)$ TPC decodes faster at both E_b/N_o values examined that the $(32, 26 \times (16, 11) \times (16, 15)$ TPC. Table 9.5 gives a summary of these results in the AWGN channel.



(a) $E_b/N_o = 2.0$ dB (b) $E_b/N_o = 2.5$ dB

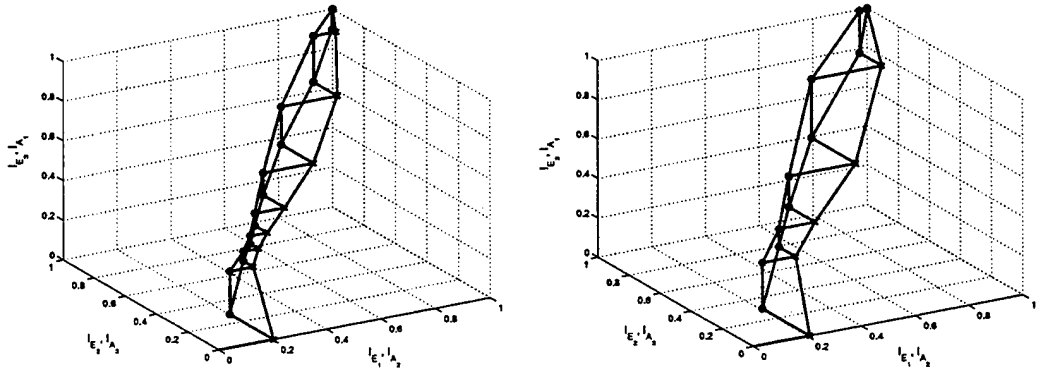
Figure 9-2: $(32, 26) \times (16, 11) \times (8, 7)$ with GMSK in AWGN

As we can see from these four figures and from Table 9.5, the $(32, 26) \times (16, 11) \times$



(a) $E_b/N_o = 2.0$ dB (b) $E_b/N_o = 2.5$ dB

Figure 9-3: $(16, 11) \times (32, 26) \times (8, 7)$ with GMSK in AWGN

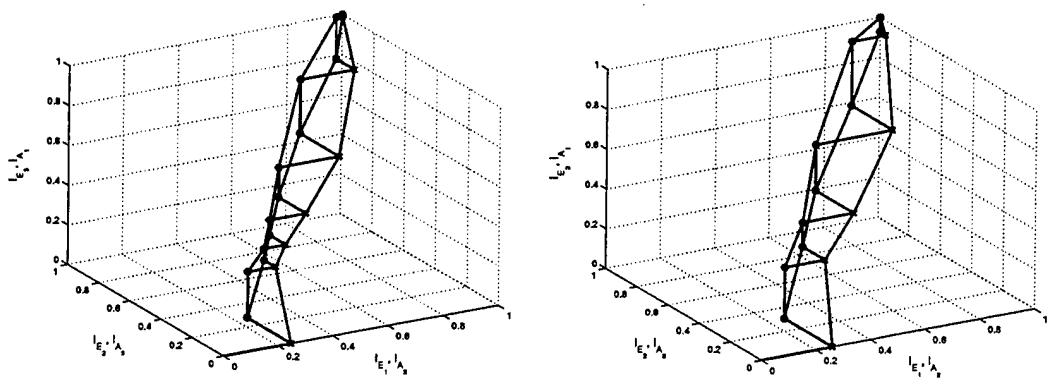


(a) $E_b/N_o = 2.0$ dB (b) $E_b/N_o = 2.5$ dB

Figure 9-4: $(32, 26) \times (16, 11) \times (16, 15)$ with GMSK in AWGN

Table 9.5: AWGN Results Summary: Number of Full Iterations to Decode

Code	$E_b/N_o = 2.0$ dB	$E_b/N_o = 2.5$ dB
$(32, 26) \times (16, 11) \times (8, 7)$	6	4
$(16, 11) \times (32, 26) \times (8, 7)$	6	4
$(32, 26) \times (16, 11) \times (16, 15)$	8	5
$(16, 11) \times (32, 26) \times (16, 15)$	6	4



(a) $E_b/N_o = 2.0$ dB

(b) $E_b/N_o = 2.5$ dB

Figure 9-5: $(16, 11) \times (32, 26) \times (16, 15)$ with GMSK in AWGN

$(8, 7)$, $(16, 11) \times (32, 26) \times (8, 7)$, and $(16, 11) \times (32, 26) \times (16, 15)$ TPCs are clearly performing quite well. Between the two different block sizes any of these three codes work equally well at this point. While neither of the 4096 block size codes has an advantage over the other, the $(16, 11) \times (32, 26) \times (16, 15)$ TPC has clearly decoded faster than the other 8192 block size code in the AWGN channel. We will next examine these four codes in a Rayleigh faded channel to see if they continue to perform equally well, or if one code performs better than the others in the new environment.

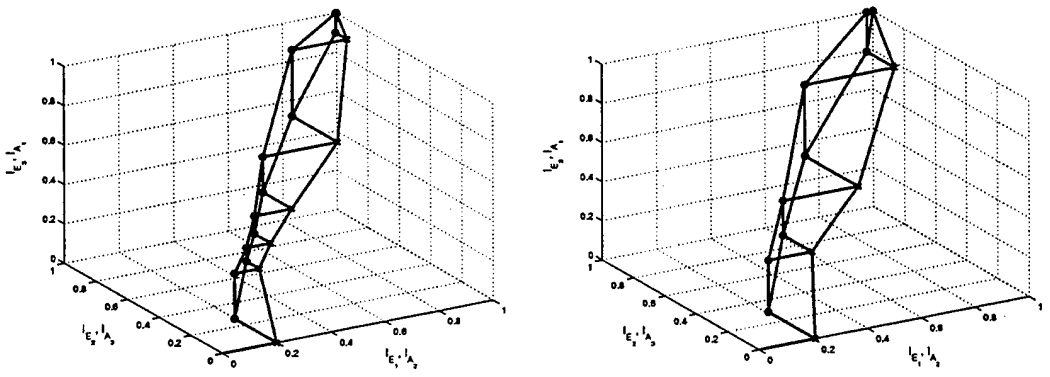
9.4 EXIT Chart Analysis with Rayleigh Fading

We will now examine these same four codes in the Rayleigh faded channel. We will examine results at $E_b/N_o = 4.0$ and 5.0 dB for each of the four codes with GMSK modulation. As with the AWGN channel analysis, these codes will have already converged, so we will primarily be looking at how many iterations are required for the decoding process to be completed and at the openness of the tunnel created between the transfer characteristics. Also, we will not only compare the results within the same block size, but across the two different block sizes as well.

Figures 9-6 through 9-9 give the results with Rayleigh fading. The results are quite similar to the AWGN channel results, except they were determined at E_b/N_o

values 2.0 to 2.5 dB larger than those previously used. For the 4096 block size codes, in Figures9-6 and 9-7, both the left-hand and right-hand plots of the figures are nearly in the waterfall region of the BER curve. These codes are easily decoded at these E_b/N_o values, in 6 and 4 full iterations. Once again, we can see that the EXIT charts for the $(16,11) \times (32,26) \times (8,7)$ TPC have a larger initial I_E value, however this does not see to have any effect on the decoding process.

Figures 9-8 and 9-9 show the results for the 8192 block size codes. The tunnels created in left-hand plot show a bottleneck region which was not present in the 4096 block size codes at the E_b/N_o. However, 1 dB later, in the right-hand plots, the tunnel is much more open and can be said to be in the waterfall region of the BER curve. This is especially evident in Figure 9-9 for the $(16,11) \times (32,26) \times (16,15)$ TPC where the number of full iterations required for the block to decode is reduced by half from the left-hand plot to the right.

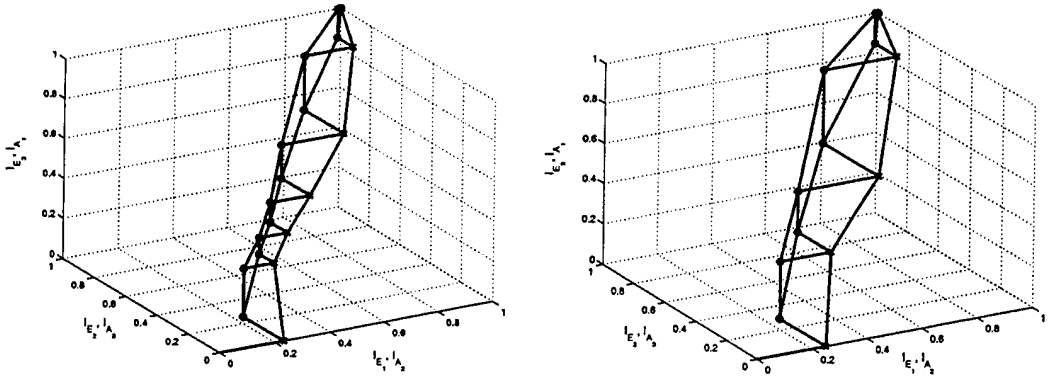


(a) $E_b/N_o = 4.0$ dB (b) $E_b/N_o = 5.0$ dB

Figure 9-6: $(32,26) \times (16,11) \times (8,7)$ with GMSK in Rayleigh fading

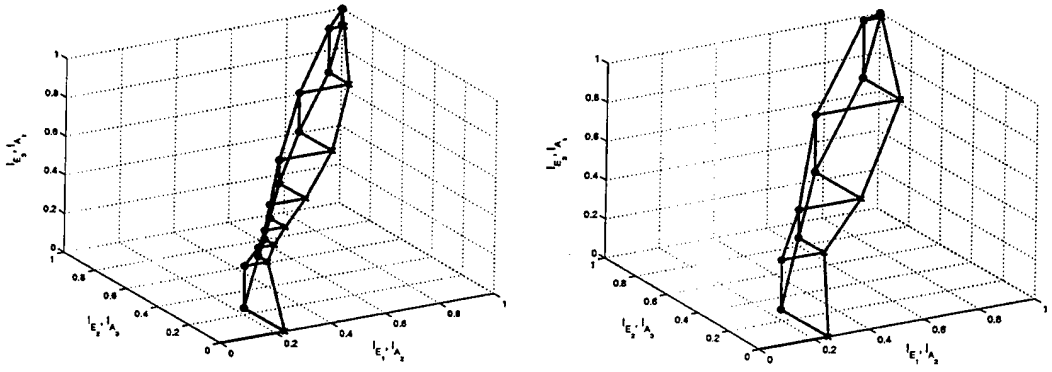
Table 9.6: Rayleigh Results Summary: Number of Full Iterations to Decode

Code	$E_b/N_o = 4.0$ dB	$E_b/N_o = 5.0$ dB
$(32,26) \times (16,11) \times (8,7)$	6	4
$(16,11) \times (32,26) \times (8,7)$	6	4
$(32,26) \times (16,11) \times (16,15)$	9	6
$(16,11) \times (32,26) \times (16,15)$	8	4



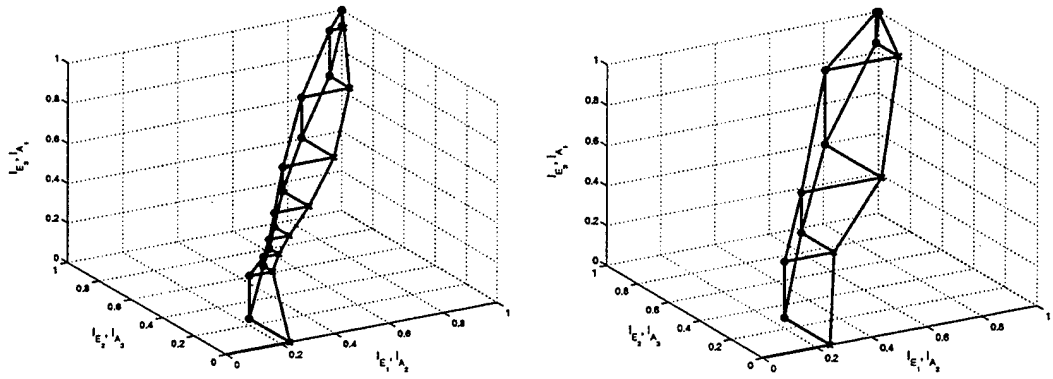
(a) $E_b/N_o = 4.0$ dB (b) $E_b/N_o = 5.0$ dB

Figure 9-7: $(16, 11) \times (32, 26) \times (8, 7)$ with GMSK in Rayleigh fading



(a) $E_b/N_o = 4.0$ dB (b) $E_b/N_o = 5.0$ dB

Figure 9-8: $(32, 26) \times (16, 11) \times (16, 15)$ with GMSK in Rayleigh fading



(a) $E_b/N_o = 4.0$ dB (b) $E_b/N_o = 5.0$ dB

Figure 9-9: $(16, 11) \times (32, 26) \times (16, 15)$ with GMSK in Rayleigh fading

Table 9.6 gives a summary of the decoding results for the Rayleigh channel. As we can see from these four figures and from table, the two 4096 block size codes continue to perform quite well. The decoding results have been equivalent for these two codes in both the AWGN channel and the Rayleigh channel. However, the $(16, 11) \times (32, 26) \times (16, 15)$ TPC has clearly decoded faster than the other 8192 block size code in the Rayleigh channel as well as the AWGN channel. Between the two different block sizes, the results from the Rayleigh channel show that the smaller 4096 block size code have a performance advantage over the larger code in terms of average number of iterations required for the block to decode.

9.5 Recommendations

One of the benefits of using EXIT charts for code design is that long simulations to determine the performance curves is not needed. Through the EXIT chart chart analysis performed in this chapter, we found several rate one-half codes which perform well with GMSK modulation in various channels. These codes are the two 4096 block size codes, the $(16, 11) \times (32, 26) \times (8, 7)$ TPC and the $(32, 26) \times (16, 11) \times (8, 7)$ TPC and the 8192 block size $(16, 11) \times (32, 26) \times (16, 15)$ TPC. These codes performed equally well in the AWGN channel and at high E_b/N_o in the Rayleigh channel. While

the analysis is not complete for these codes, they are recommended for further study.

As mentioned, the long simulations are not necessary when designing codes with the EXIT chart. However, to make this analysis more complete, the performance curves for the four TPCs examined in this chapter were determined for both the AWGN and the Rayleigh channel. Figures 9-10 and 9-11 give the performance curves for these TPCs. As shown, the $(32,26) \times (16,11) \times (8,7)$ TPC and the $(16,11) \times (32,26) \times (16,15)$ TPC reach a BER of 10^{-5} earliest in both the AWGN and Rayleigh channel. However, all four codes perform quite well, with all codes reaching a BER of 10^{-5} within a span of less than 0.25 dB.

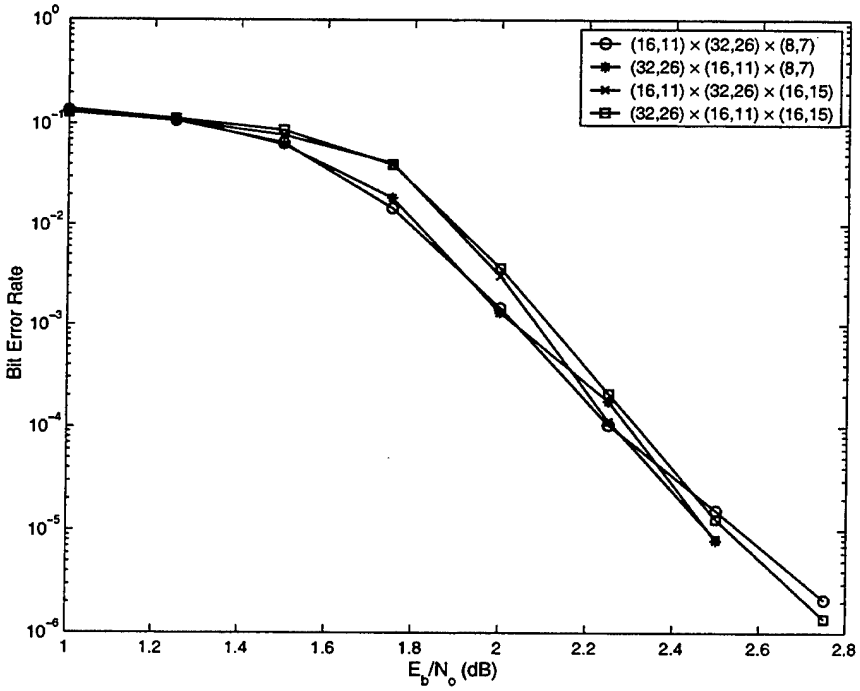


Figure 9-10: Performance Curves with GMSK in AWGN

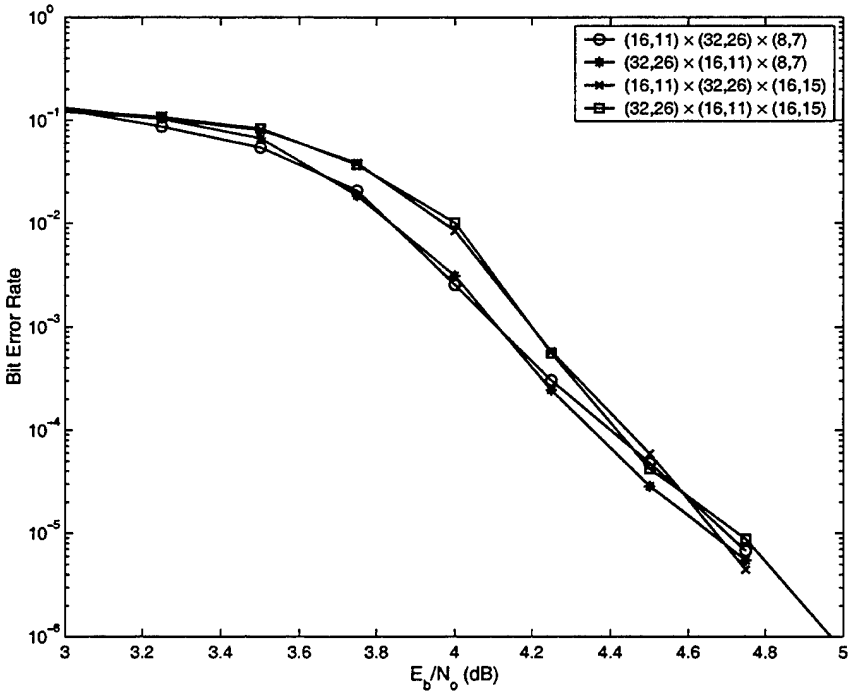


Figure 9-11: Performance Curves with GMSK in Rayleigh fading

Chapter 10

Summary and Conclusions

The need for power-efficient communications that not only meet the system constraints but are reliable as well is growing larger, particularly for satellite and terrestrial wireless systems. Turbo product codes (TPC) are one class of codes which have shown to be a promising approach to meeting these needs. TPCs are a product code formed from two or more constituent block codes. These codes use an iterative decoding method similar to turbo codes, but without the same high-degree of decoding complexity as turbo codes. TPCs have been shown to have a BER performance within a couple of dB of turbo codes without the error floor, and they also perform well at high code rates. However, other performance measures of TPCs are not well developed.

This thesis applied the Extrinsic Information Transfer (EXIT) Chart analysis, developed for turbo codes, to TPCs in order to examine the density evolution of the TPCs. The EXIT chart analysis allows for examination of the evolution of the probability densities of the information passed from iteration to iteration in the iterative decoder.

The analysis began with the examination of four two-dimensional TPCs in both binary phase-shift keying (BPSK) and Gaussian minimum-shift keying (GMSK) modulated additive white Gaussian noise (AWGN) channels using the EXIT chart. These codes were the $(64,57)^2$ TPC, $(32,26)^2$ TPC, $(32,26) \times (64,57)$ TPC, and $(64,57) \times (32,26)$ TPC. We have seen similar results with both modulations, differing only in

that the GMSK curves occurred at E_b/N_o values about 0.5 dB larger than the values required for similar results in BPSK. This correlates to the difference in BER performance found between GMSK and BPSK with these codes. We have also seen how the TPCs move through their convergence points as the decoding trajectories transition from being incomplete to being complete.

Then, the same four TPCs were examined in the Rayleigh fading environment. The EXIT chart results with Rayleigh fading were quite similar with the AWGN channel results - for both the BPSK and GMSK modulations. However, to achieve these similar results, the input E_b/N_o was several dB larger for Rayleigh channels that in AWGN. Interestingly, these TPCs perform better in the Rayleigh environment with GMSK than BPSK. This is indicated by a degradation of 3 dB for the GMSK results, but 4 to 5 dB for the BPSK results between AWGN and Rayleigh fading. The EXIT chart analysis was extended to multidimensional codes. We developed these methods and examined the results for the $(16, 11)^3$ TPC with both BPSK and GMSK modulation in both the AWGN and Rayleigh fading channel.

Results for BER prediction in the low E_b/N_o region were developed. The shape of the predicted curves closely matched the simulated results, which is useful so that we need not run long simulations to get an initial estimate on the BER performance. However, it should be noted that the BER predictions work best at E_b/N_o values near the convergence values for that code, which are fairly low compared to the E_b/N_o values usually required for communication links. Convergence, the E_b/N_o value for which the EXIT chart and associated decoding trajectory are first complete, verified many of the results given in previous chapters. This analysis is also useful on its own for estimating the shape of the overall performance curve as a function of iteration. Finally, all of the methods presented throughout the thesis were used to construct a new rate 1/2 TPC.

10.1 Key Results

Several key results were found from this analysis. From our examination of the two non-square TPCs, the $(32, 26) \times (64, 57)$ TPC and the $(64, 57) \times (32, 26)$ TPC, it was found that having the shorter constituent code on the x-axis decreases block decoding time for all modulations and channels examined. In addition, the convergence analysis verified this result. However, these two codes have an almost identical steady-state performance, and our results were inconclusive as to one coding order being preferred over the other.

Utilizing the techniques developed earlier, we constructed several rate one-half codes which perform well with GMSK modulation in various channels. These codes are the two 4096 block size codes, the $(16, 11) \times (32, 26) \times (8, 7)$ TPC and the $(32, 26) \times (16, 11) \times (8, 7)$ TPC and the 8192 block size $(16, 11) \times (32, 26) \times (16, 15)$ TPC. These codes performed equally well in the AWGN channel and at high E_b/N_o in the Rayleigh channel. While the analysis is not complete for these codes, they are recommended for further study.

In chapter 1, we compared several coding techniques that are under consideration for use in satellite communications. In Figure 10-1, we can see these same codes as well as several of the TPCs examined in this thesis. These codes include rate 1/2 and 2/3 turbo codes with QPSK modulation and CPM, and a rate 1/2 turbo code with binary GMSK modulation. The TPCs fall in the same general area as the other codes under consideration, and as a result, we feel that TPCs should be included in that process.

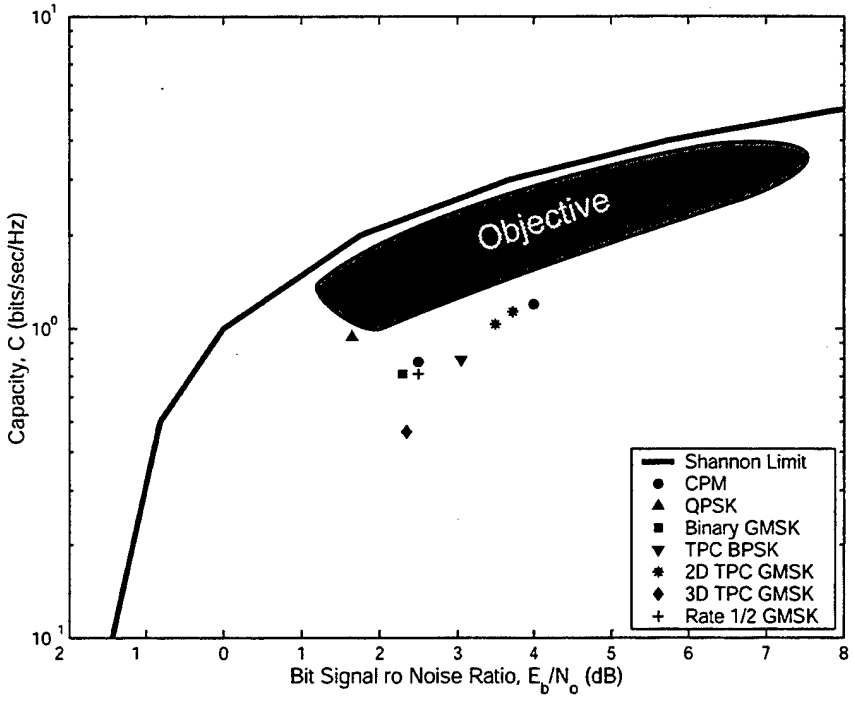


Figure 10-1: Comparison of Some Current Satcom Modulation and Coding Schemes with Shannon Capacity

10.2 Future Work

While several good results were found in this thesis, there are several questions we can leave open for future consideration. The first item comes from our result that it is better to place the shorter constituent code on the x-axis when working with non-square two-dimensional codes. Is this an anomaly based solely on the codes examined, or can it be said to be true in general? Also, there are many code combinations, especially in three-dimensions, that have not been examined. The three-dimensional TPCs are very powerful and should definitely be examined more thoroughly in the future. Finally, the TPC decoder designed by ECC and AHA utilizes feedback values in the decoding process. In our code design example, we did not attempt to optimize these feedback values. Such an optimization could prove quite useful, especially when three-dimensional codes are examined.

Appendix A

List of Acronyms

AHA	Advanced Hardware Architectures, Inc.
AWGN	Additive White Gaussian Noise
BCJR	Bahl-Cocke-Jelinek-Raviv algorithm
BER	Bit-Error Rate
BPSK	Binary Phase-Shift Keying
CDF	Cumulative Density function
CPM	Continuous Phase Modulation
E_b/N_o	Information bit energy to noise density ratio
ECC	Efficient Channel Coding, Inc.
EXIT Chart	Extrinsic Information Transfer Chart
FEC	Forward Error Correction
GMSK	Gaussian Minimum-Shift Keying
L-values	Log-likelihood ratios
LOS	Line of Sight

MAP	Maximum a posteriori
ML	Maximum likelihood
PAM	Pulse Amplitude Modulation
PDF	Probability Density function
QPSK	Quaternary Phase-Shift Keying
SNR	Signal-to-Noise Ratio
SPC	Single Parity Check code
SOVA	Soft-Output Viterbi algorithm
TPC	Turbo Product Code

Bibliography

[1] D.D. Abbey and R. R. Rhodes. Simple coherent receivers for frequency-hopped pulse driven GMSK. volume 1, pages 286–290. IEEE MILCOM, '99, 1999.

[2] Advanced Hardware Architectures, Inc. *Primer: Turbo Product Codes.*

[3] Advanced Hardware Architectures, Inc. *Product Specification AHA4501: 36 MBits/sec Turbo Product Code Endoder/Decoder.* Available at www.aha.com.

[4] C. Berrou, A. Glavieux, and P. Thitimajshima. Near Shannon limit error-correcting coding and decoding: turbo-codes. *Proceedings ICC*, pages 1064–1070, May 1993.

[5] D. Chase. A class of algorithms for decoding block codes with channel measurement information. *IEEE Transactions on Information Theory*, IT-18:170–182, January 1972.

[6] D. Divsalar, S. Dolinar, and F. Pollara. Iterative turbo decoder analysis based on density evolution. *IEEE Journal on Selected Areas in Communications*, 19(5):891–907, May 2001.

[7] Efficient Channel Coding, Inc. *Technical Description of Turbo Product Codes*, June 1999. Version 4.0.

[8] P. Elias. Error-free coding. *IRE Transactions on Information Theory*, PGIT-4:29–37, September 1954.

[9] H. El Gamal and A. R. Hammons Jr. Analyzing the turbo decoder using the Gaussian approximation. *IEEE Transactions on Information Theory*, 47(2):671–686, February 2001.

[10] A. Grant. Convergence of non-binary iterative decoding. volume 2, pages 1058–1062. IEEE GLOBECOM, '01, 2001.

[11] J. Hagenauer and P. Hocher. A Viterbi algorithm with soft-outputs and its applications. pages 1680–1685. IEEE GLOBECOM, '89, November 1989.

[12] R. W. Hamming. *Coding and Information Theory.* Prentice-Hall, Englewood Cliffs, NJ, Second edition, 1986.

[13] S. Lin and D. J. Costello Jr. *Error Control Coding: Fundamentals and Applications.* Prentice Hall, Inc., Englewood Cliffs, N.J., 1983.

[14] J. G. Proakis. *Digital Communications.* WCB McGraw-Hill, New York, New York, Third edition, 1995.

[15] R. Pyndiah. Near optimum decoding of product codes: block turbo codes. *IEEE Transactions on Information Theory*, 46(8), August 1998.

[16] T. Richardson and R. Urbanke. The capacity of low density parity check codes under message passing decoding. *IEEE Transactions on Information Theory*, 47(2):599–618, February 2001.

[17] S. ten Brink. Design of serially concatenated codes based on iterative decoding convergence. In *Proceedings International Symposium on Turbo Codes*, pages 319–322, Brest, France, September 2000.

[18] S. ten Brink. Convergence behavior of iteratively decoded parallel concatenated codes. *IEEE Transactions on Communications*, 49(10):1727–1737, October 2001.

[19] W. H. Thesling. *Efficient Block Decoding Algorithms Exploiting Channel Measurement Information.* Doctor of Engineering Dissertation. Cleveland State University. December 1995.

A Density Evolution Analysis of Turbo Product Codes*

by

Laura M. Durham

Submitted to the Department of Electrical Engineering and Computer Science
on May 22, 2002, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering and Computer Science

Abstract

Turbo product codes (TPC) are a promising approach for power-efficient communications, particularly in satellite and terrestrial wireless systems. These codes use an iterative decoding method similar to turbo codes. TPCs have been shown to have a bit error rate (BER) performance within a couple of dB of turbo codes without the error floor, however other performance measures of turbo product codes are not well developed. This thesis applies the Extrinsic Information Transfer (EXIT) chart analysis, developed for turbo codes, to turbo product codes. The EXIT chart analysis allows for examination of the evolution of the probability densities of the information passed from iteration to iteration of the decoder. The analysis begins with the EXIT chart analysis for two-dimensional TPCs, similar to the turbo code results, and then extends the analysis to three-dimensional TPCs. Binary phase-shift keying (BPSK) and Gaussian minimum-shift keying (GMSK) modulations are examined in both an unfaded additive white Gaussian noise (AWGN) as well as Rayleigh faded channel. In addition, BER results are predicted in the low E_b/N_o region, convergence thresholds determined, and lastly a new code construction for a rate 1/2 TPC is designed.

Thesis Supervisor: Lori L. Jeromin
Title: Technical Staff, MIT Lincoln Laboratory