ALGORITHMS FOR DECODING BLOCK CODES

by

Amer Aref Hassan

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering : Systems)
in the University of Michigan
1989

Doctoral Committee:

Associate Professor Wayne E. Stark, Chairman
Professor Daniel Burns
Professor David L. Neuhoff
Associate Professor Demosthenis Teneketzis

Correction on page 45:

The following sentence "It is possible for the parallel decoder to .... close to the maximum error correcting capability of the code" is incorrect and should read:

> "It is impossible for the parallel decoder to choose an erroneous concatenated codeword if t(v) is less than\ γ"

The rest of the sentence "however, Reed-Solomon ... capability of the code. " is redundant and should be omitted.

This is obvious since spheres centered around codedwords with radii less than or equal to half the minimum distance (Hamming or Euclidean) of the code do not intersect.

to my parents, Aref and Nadima
to my brothers and sisters
and to my lovely wife, Sara

ACKNOWLEDGEMENTS

The majority of my gratitude goes to professor Wayne E. Stark, my thesis advisor, for his continuing support and encouragement during the course of the study. I am happy to express my warmest thanks for his guidance.

I sincerely appreciate the effort invested by professors David L. Neuhoff, Demosthenis Teneketzis, and Daniel Burns in reading the dissertation. I am particularly grateful to professor David L. Neuhoff for his observations and thoughtful criticizm concerning my thesis.

This work has been supported by National Science Foundation under grant ECS 8451266, the Office of Naval Research under contract N00014-85-k-0545, and Hughes Aircraft. Their much needed financial support is greatly appreciated.

Outside the technical realm, I am deeply indebted to my parents, my family, and my wife, Sara. I owe much for their unending patience, encouragement, and understanding without which I could hardly have completed this work.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

LIST OF APPENDICES

Appendix

CHAPTER I

INTRODUCTION

1.1 The Essence of Coding

A communication system connects a data source to a data user (receiver) through a channel such as a microwave link, optical fiber, or a magnetic tape. A communication engineer designs a system to process the information to be sent to match the channel characteristics and processes the output of the channel to determine the transmitted information. Because the channel is subject to various types of noise, distortion, and interference, the channel output might be different than the channel input. One way to combat noise and interference is through the use of error control coding techniques. Data is first processed by the encoder which transforms a sequence of (information) data symbols into another (typically) longer sequence called the channel codeword. The set of all possible codewords form a code. The transformation consists of introducing redundant symbols into the original data sequence. Next a modulator converts each codeword symbol into a corresponding analog waveform, from a set of possible analog waveforms, which is transmitted through the physical channel. The demodulator converts each received channel output signal into another discrete-time analog symbol upon which a decision about the transmitted code symbol can be made. The decoder uses the redundancy in the transmitted codeword to correct as many errors as possible and puts out its best estimate. One very important measure of the reliability of communication is the probability of the decoder not putting out the transmitted codeword, referred to as the probability of error. Without coding it is well known that in the presence of Rayleigh fading or worst case finite power interference the bit error probability is an inverse linear function to the received signal energy-to-noise ratio, as opposed to an exponentially decreasing function for broadband additive noise. This can cause an increase of 30-40 dB in required bit energy-to-noise ratio when a probability of bit error of 10^{-5} is desired. Coding techniques for unintentional as well as hostile interference are discussed extensively in the literature.

Coding theory had effectively started in 1948 by Shannon [29]. Shannon showed, by the coding theorem, that associated with a channel there is a nonnegative number C, measured in bits per second, with the following significance. If the transmission rate R in bits per second is less than C, it is possible to design a communication system using error-control codes that results in as small an error probability as desired.

The prime motivation for coding research since 1948 has been Shannon's coding theorem. Although our understanding of the coding theorem has been refined, it still does not give satisfactory answers from a practical viewpoint. This is because the coding theorem is, from a practical viewpoint, an existence theorem. It demonstrates that a certain performance can be obtained by unstructured coding schemes, but fails to specify a particular code which achieves this performance and/or an encoding and decoding methods with reasonable complexity. The two main contributions to complexity are usually taken to be the maximum number of operations required and the number of memory registers needed to perform a certain decoding algorithm. The introduction of linear codes essentially solves the encoding problem for block codes; codes in this class can obtain the performance guaranteed by the coding theorem and can be encoded with algorithms that have low complexity (complexity linear in the length of the code). However, the decoding of linear block codes is, in general, very complex for reasons clarified below. We first, however, describe two classes of channel models which are used throughout the thesis. These channels model different modulation/detection schemes.

1.2 Channel Models

The channel models we use model the communication system from the input of the modulator to the output of the demodulator. These channels are discrete in time in the sense that the channel input and output are time sequences of letters selected from arbitrary alphabets. Denote the input sequence to the channel as $x_1, x_2, ...$ and the corresponding output as $y_1, y_2, ...$, where the input and output alphabets are $\mathcal{X}$ and $\mathcal{Y}$, respectively . A channel is memoryless if the output y_i at time i depends only on the input x_i at time i; i.e., the probabilities that $y_1, y_2, ..., y_n$ is the output given $x_1, x_2, ..., x_n$ is the input is the product $\Pi_{i=1}^{n} p(y_i|x_i)$, where $p(y|x)$ is the conditional probability the output of the channel is y given that the input to the channel is x; also the probabilities are independent of time (i.e., the position in the sequence). In the following models we restrict ourselves to memoryless channels that allow only a finite set of letters in the input alphabet. The output space will be infinite in two cases of interest (discussed in Chapters 3 and 4).

The first class of channel models discussed here are the (one-dimensional) binary discrete-time additive channels. Such a channel is depicted in Figure 1.1. The input alphabet $\mathcal{X}$ is equal to $\{0,1\}$ and the output alphabet $\mathcal{Y}$ is the set of real numbers. Moreover, if $X_1, X_2, \ldots$ are the inputs to the channel at times 1, 2, ... then the corresponding outputs $Y_1, Y_2, \ldots$ are given by $Y_i = h(X_i) + Z_i$ for all i, where

$$h(X) = \begin{cases} +\sqrt{E_s} \ , & X = 0 \\ -\sqrt{E_s} \ , & X = 1 \ , \end{cases}$$

where E_s is called the code symbol energy, and $Z_1, Z_2, \ldots$ are independent and identically distributed random variables. Also Z_i is independent of X_j for all i and j. The channel is called an additive white Gaussian noise (AWGN) channel if Z_i is a Gaussian random variable. This channel is used to model antipodal signaling in white Gaussian noise with coherent reception when no quantization is performed at the output of the demodulator. In this case when coding is used the decoder makes a decision about the input to the channel based on a vector of real-valued outputs from the channel. This is referred to as (pure) soft decision decoding.

The M-dimensional (vector) additive channel shown in Figure 1.2 is another additive channel of interest. The input to the channel is one of M symbols, $\mathcal{X} = \{0, 1, \ldots, M-1\}$. The vector channel disturbs the transmission and outputs a random vector

$$\boldsymbol{Y} = (Y_0, \ldots, Y_{M-1}) = F(X) + \boldsymbol{Z}$$

where $F(X)$ is a vector of length M in which only the X-th component is nonzero and has a value equal to $f_j(X)$; that is, $F(X) = (f_0(X), \ldots, f_{M-1}(X))$ such that $f_j(X) = \sqrt{E_s}$ when $j = X$ and 0 otherwise. Also, $\boldsymbol{Z} = (Z_0, \ldots, Z_{M-1})$, the error vector, is independent of X, and addition is component-wise. This channel serves

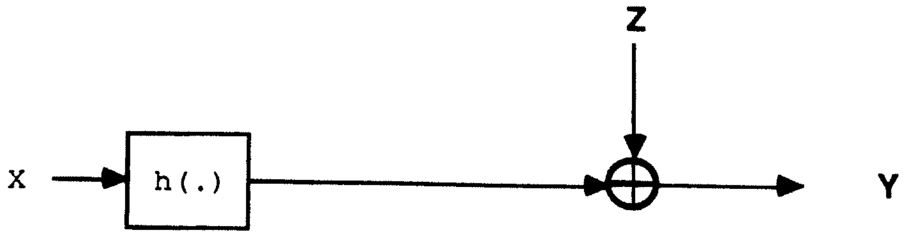

Figure 1.1: Discrete-Time Additive Channel.

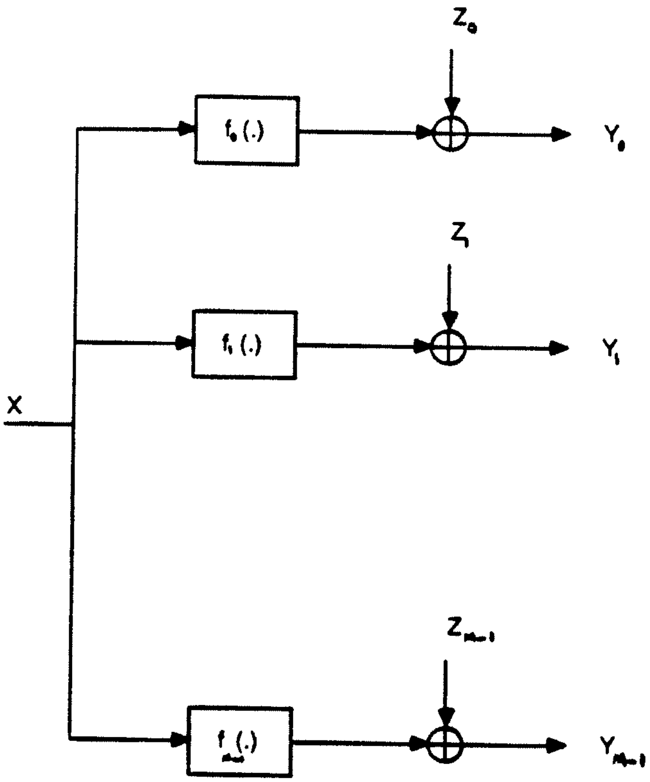

Figure 1.2: Vector Additive Channel.

as a model for an orthogonal signaling modulator with coherent demodulation. This channel, however, is not applicable to the noncoherent channel (i.e., the case when code symbols are noncoherently demodulated). For a noncoherent receiver the vector channel disturbs the signal in a nonlinear fashion.

The second class of channel models are the Discrete Memoryless Channels (DMC). Such a channel is characterized by a finite input alphabet $\mathcal{X}$ of , say, M symbols, finite output alphabet $\mathcal{Y}$, and a set of transition probabilities $p(y|x)$, defined for each $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ as the probability that the output of the channel is y if the input to the channel is x. When a finite level quantizer is placed at the output of the demodulator, e.g. the output space of the additive channel model described above is partitioned, the communication channel between the input to the modulator and the output of the quantizer is discrete, resulting in a DMC. Below we describe two special cases of the DMC.

In many of the applications, the sizes of the input and output alphabets are equal, say to M. Also, in many situations of interest, if a symbol is in error, i.e. the output symbol is not the same as the corresponding input symbol, it is equally likely to be any symbol excluding the transmitted one, and the probability the output of the channel is the same as the input to the channel is the same for all symbols in the alphabet. In such a situation the channel is said to be symmetric. Formally,

$$p(y \mid x) = \begin{cases} \frac{p}{M-1} \ , & y \neq x \\ 1 - p \ , & y = x \ , \end{cases}$$

for all $y \in \mathcal{Y}$ and $x \in \mathcal{X}$. These assumptions gives rise to the M-ary symmetric channel shown in Figure 1.3. Decoders for this channel are called hard decision decoders. Also it is convenient to represent both the input and the output alphabets

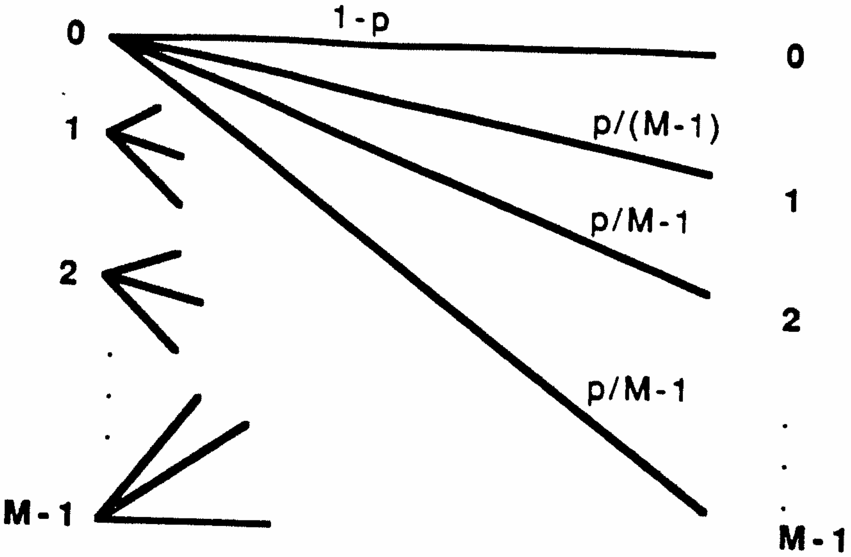

Figure 1.3: *M*-ary Symmetric Channel.

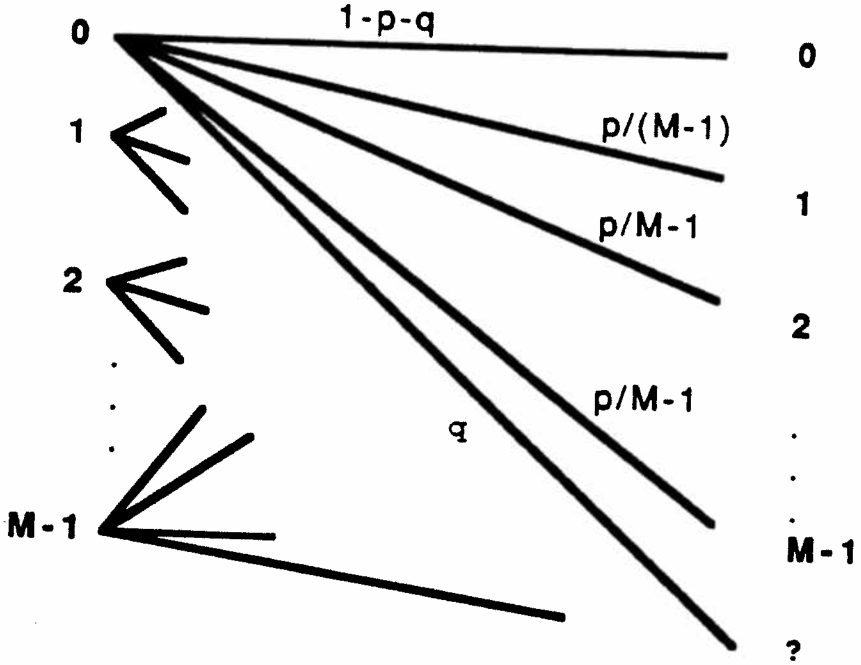

Figure 1.4: *M*-ary Symmetric-Erasure Channel.

by the integer values $\{0, 1, ..., M - 1\}$.

An M-ary symmetric channel can also be represented as an adder channel: If $c \in \{0, 1, ..., M-1\}$ is the transmitted code symbol, then the corresponding received symbol $y \in \{0, 1, ..., M - 1\}$ at the output of the channel is given by $y = c + e$, where $e \in \{0, 1, ..., M - 1\}$ is an error symbol added by the channel, and addition is performed modulo M. A code symbol is received in error if and only if $e \neq 0$ which occurs with probability p.

There is another DMC with one more output than that of an M-ary symmetric channel that is of considerable interest. This DMC is called an M-ary symmetric errors-and-erasures channel with M inputs and $M + 1$ outputs, and is shown in Figure 1.4 with q being the probability a received symbol will be erased. The input alphabet of this channel is $\mathcal{X} = \{0, 1, ..., M - 1\}$, and the output alphabet is $\mathcal{Y} = \{0, 1, ..., M - 1, ?\}$. This channel is characterized by the following transition probabilities:

$$p(w \mid x) = \begin{cases} \frac{p}{M-1} \ , & w \neq x \ w \neq ? \\ 1 - p - q \ , & w = x \\ q & w = ? \ . \end{cases}$$

The additional symbol, called an "erasure" (denoted "?"), may reflect symbols which the demodulator determines are very noisy, or those for which an estimate based on the received symbol is unreliable. This is often done using information, called side information, about the channel during the reception of a code symbol. Side information, has been used extensively in the literature when the channel is jammed [6]-[26] or or when hits occur in multiple access communication (see [25] and [18]), and it can be generated from a number of different sources, some of which are described in [24]; briefly these include predetection and postdetection methods. The predetection methods are based on power measurements applied to the received signal on a symbol-by-symbol basis. This can be accomplished via an automatic gain control (AGC) device. Usually predetection methods are the least reliable due to their high sensitivity to various fluctuations of the signal amplitude. Postdetection methods are based on certain statistics obtained from the output of the demodulator. Examples of this include Viterbi's ratio threshold technique (see [33] and [8]) which was proven to be very useful in a partial-band interference environment. An errors-and-erasures decoding algorithm takes this additional symbol into account. We shall demonstrate how converting an M-ary symmetric channel to a one that includes erasures can improve the performance of a communication system.

1.3 The Decoding Problem for Linear Block Codes

In this section we discuss block codes, particularly decoding for linear codes, for the q-ary symmetric channels and the additive channels. In particular we describe standard encoding and decoding algorithms for block codes and discuss the complexity of these algorithms on various channels. We begin with standard definitions.

For a q-ary (discrete-time) input channel, an $[M,n]$ block code C of length n over GF(q) (a field of q elements) is a collection of M vectors called codewords each of the form $(c_1, c_2, ..., c_n)$ with components in GF(q). An (n,k) linear code over GF(q) is a $[q^k, n]$ block code for which the codewords with symbols in GF(q) form a k-dimensional linear subspace of the n-dimensional space. Any such a code is uniquely defined by a set of k-linearly independent codewords, which form a generating set. Equivalently any (n,k) code can be defined to be the set of those n-tuples $(c_1, ..., c_n)$ satisfying a set of $n-k$ linearly independent linear equations

$$\sum_{i=1}^{n} c_i h_{i,j} = 0, \quad j = 1, 2, ..., n-k \tag{1.1}$$

where $h_{i,j} \in \mathrm{GF}(q)$ are entries in a $(n-k) \times n$ matrix called the parity check matrix of the code.

Encoding for a q-ary input, discrete time channel using a code over $\mathrm{GF}(q)$ consists of a one-to-one mapping $\mathcal{F}$ from the k-dimensional message space to the set $\mathcal{C}$ of codewords. That is

$$\mathcal{F} : \mathcal{X}^k \longrightarrow \mathcal{C},$$

where $\mathcal{X} = \{0, 1, ..., q-1\}$ and $\mathcal{C} \subset \mathcal{X}^n$. Decoding for any of the channels described earlier, with output alphabet $\mathcal{Y}$, consists of a composition of two functions $\mathcal{D}_1$ and $\mathcal{D}_2$ such that

$$\mathcal{D}_1 : \mathcal{Y}^{*n} \longrightarrow \mathcal{C}$$

$$\mathcal{D}_2 = \mathcal{F}^{-1} : \mathcal{C} \longrightarrow \mathcal{X}^k\,,$$

where $\mathcal{Y}^{*n} \subset \mathcal{Y}^n$. The reason for restricting the domain of $\mathcal{D}_1$ to be a subset of the output space will be clear later. When a q-ary code is used on a q-ary symmetric channel $\mathcal{Y} = \mathcal{X}$. When this code is used on a q-ary errors-and-erasures channel $\mathcal{Y} = \mathcal{X} \bigcup \{?\}$. When this code is used on an additive channel $\mathcal{Y} = \mathcal{R}$.

The Hamming distance $d(\boldsymbol{x}, \boldsymbol{y})$ between two q-ary sequences $\boldsymbol{x}$ and $\boldsymbol{y}$ of length n is the number of places in which they differ. The minimum distance of a code is defined to be the minimum Hamming distance between any distinct pair of codewords. An (n,k) linear code with minimum Hamming distance d is sometimes referred to as an (n, k, d) code. When used on a q-ary symmetric channel a code with minimum distance d can correct any pattern of e errors provided $2e + 1 \leq d$ [5] and e is said to be the error correcting capability of the code. When used on an q-ary symmetric errors-and-erasures channel the code can correct all patterns of e errors and τ erasures simultaneously if $2e + \tau \leq d - 1$.

When dealing with additive channels it will sometimes be useful to treat the codewords of a code as real valued vectors. The real valued vectors are obtained from vectors with symbols in $\{0, ..., M - 1\}$ via the transformations F and h discussed in the previous section. It will be clear to the reader which form is being used and should not cause any ambiguity. For example, in the simplest case of antipodal signaling (i.e., the additive channel), the one-dimensional transformation consists of the function $h(.)$ defined earlier. Then the Euclidean distance $d_E(x, y)$ between $x \in \mathcal{R}^n$ and $y \in \mathcal{R}^n$ is

$$d_E(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2},$$

where $\mathcal{R}$ is the real line. Similar definition holds for the M-ary vector additive channel using the transformation $F(.)$. The minimum Euclidean distance d_E of a code is the minimum Euclidean distance between any distinct pair of codewords with elements in $\mathcal{R}$. A maximum likelihood decoder will map a received vector to the closest codeword; i.e. $\mathcal{Y}^{*n} = \mathcal{Y}^n$. When used on an additive channel a (bounded distance) soft decision decoder will decode correctly only those received vectors with Euclidean distance $d_E/2$ from some codeword.

We assume that after transmission through the q-ary symmetric channel channel the received n-tuple $(y_1, ..., y_n)$ differs from the transmitted codeword by some error sequence $(e_1, e_2, ..., e_n)$; that is $y_j = c_j + e_j$ for $j = 1, ..., n$, where addition is performed over GF(q). Also, $e_1, e_2, ..., e_n$ are independent and identically distributed random variables with $\Pr(e_i \neq 0) = 1 - p$ and $\Pr(e_i = 0) = p$ for all i (as in Figure 1.3). Define $s_j,\ j = 1, ..., n - k$ as

$$s_j \triangleq \sum_{i=1}^{n} y_i h_{i,j} = \sum_{i=1}^{n} e_i h_{i,j}. \tag{1.2}$$

The $(n - k)$-tuple $(s_1, ..., s_{n-k})$ is known as the syndrome of the received sequence. The (optimal) decoding problem consists of finding the error sequence, producing a given syndrome, that is most probable for a given channel. For a symmetric memoryless channel the most probable error sequence is that one which has the minimum number of nonzero components. In that case the decoding problem reduces to finding the solution to equation (1.2) with the minimum weight.

It is generally not feasible to find the most probable solution of (1.2) for an arbitrary syndrome, simply because of the enormous number of possibilities. Furthermore, it has been shown by Berlekamp and McEliece [4] that the algorithm that solves for the most probable solution of (1.2) is in the class of NP-complete algorithms. This means that it is (currently) not possible to find an algorithm for (1.2) whose complexity does not grow exponentially but grows polynomially with n. Levitin and Hartman [19] based on a new concept of zero neighbors (a special set of codewords) found an algorithm for which the time complexity is polynomial but the space complexity is exponential in n. Optimal decoding algorithms for other channels are at least as complex as that of the DMC.

For an M-ary symmetric channel an efficient suboptimal solution to the decoding problem depends on finding a simple method for determining an approximation to the most probable solution of (1.2) for a high probability subset of the set of all error sequences. This shows why we assumed $\mathcal{Y}^{*n} \subset \mathcal{Y}^n$. For codes with some algebraic structure such as being linear or cyclic, algebraic decoding has led to a practical decoding algorithm for M-ary symmetric errors-and-erasures channels called bounded distance decoding. A bounded distance decoder decodes only those received vectors lying in a decoding sphere about a codeword (a decoding sphere is the set of errors and erasures pairs (e, τ) correctable by the decoder such that $2\tau + e \leq t$, where t is called the radius of the sphere). Other received vectors that have more than the number of errors and erasures correctable by the code are declared by the decoder as unrecognizable in which case the decoder is said to have *failed*. For instance, the Berlekamp algorithm for linear cyclic codes (codes for which cyclic shifts of a codeword is also a codeword) will correct e errors and τ erasures if $2e + \tau$ is less than the minimum Hamming distance d of the code. The complexity of implementation of this algorithm is proportional to d^2.

A decoding algorithm for an M-ary symmetric channel can be used for a discrete time additive channel by quantizing the output of the additive channel. However, there is an information loss caused by quantizing which will cause degradation in performance. Under most conditions soft decision decoding gives better performance than hard decision decoding or errors-and-erasures decoding mentioned earlier. For instance, for additive white Gaussian noise channel hard decision decoding results in an asymptotic 2 dB loss in signal-to-noise ratio over soft decision decoding in the limit as the code rate goes to zero, both for binary and nonbinary codes. One problem remains: soft decision decoding requires high complexity for codes with large block length. It is desired to recover the loss incurred by quantizing without implementing a highly complex decoding algorithm.

1.4 Outline and Summary of the Results of the Thesis

In Chapters 3 and 4 we investigate practical decoding methods for the additive channel. Then in Chapters 5 and 6 we propose techniques to create an M-ary errors-and-erasures channel for a slow time-varying channel. In the former case the proposed algorithms use the powerful class of codes known as concatenated codes introduced by Forney [12]. Very long block codes are possible with reasonable complexity. Decoding is done in two stages resulting in a significant reduction in decoding complexity over that which would be required to provide the same overall error rate with a single level of coding. The remainder of this Section gives an outline of the results of this thesis.

In Chapter 2 we briefly review concatenated codes as developed by Forney then describe a generalized concatenated coding scheme due to Zinov'ev and Zyablov [38]. We propose a parallel algorithm for the discrete-time additive channel similar to that introduced by Zyablov [40] for the M-ary symmetric channel. The proposed algorithm (described in detail in Chapter 2) uses several decoders in a parallel fashion. Each of the decoders consists of an inner and an outer decoder. The inner decoder is a soft decision decoder for a short block length code. The outer decoder is an errors-and-erasures Reed-Solomon decoder. The algorithm outputs several candidate codewords and a decision device chooses the most likely one, as the transmitted codeword. The main result is to optimize the inner decoders to maximize the minimum Euclidean distance correctable by the overall system.

Chapter 3 investigates the *parallel* decoding problem when the code symbols are coherently detected; in this case the performance measure is taken to be the Euclidean distance correctable by the concatenated code; this measure is referred to as the error correcting capability of the code when used on an additive channel. Then it is shown that the decoding algorithm has (close to) the maximum error correcting capability when only few decoding branches are used.

The second problem, investigated in Chapter 4, attempts to characterize the error correcting capability when the reception of code symbols is noncoherent; that is when the received waveforms are detected using a noncoherent matched filter. However, for noncoherent detection the noise effect is more complicated than in the coherent case; the noise does not affect the decision of the decoder in an additive fashion. Because of this, minimum Euclidean distance correctable is not an appropriate performance measure. Nevertheless, we motivate the idea of parallel decoding for this noncoherent channel by making certain assumptions about the noise. The inner code is taken to be a repetition code and the inner decoder is a square-law combiner which, for each codeword respectively, sums the (energy) outputs of the noncoherent matched filters. Thus the number of the sums at the output of the combiner is the same as the number of (inner) codewords. A decision devise then compares these sums and delivers an erasure if any two outputs are "close", otherwise it takes the codeword with the largest sum, as the transmitted codeword. We find the optimum set of thresholds that maximizes a certain error correcting capability when the inner decoder is similar to the Viterbi Ratio Thresholding described in [34].

The communications systems analyzed in Chapters 5 and 6 evaluate techniques that will enable the communication designer to create an errors-and-erasures channel and use coding for this channel. The particular application considered is a coded slow frequency hopped spread spectrum system where the channel suffers from Rayleigh fading. In a slow frequency hopped system a fixed number of symbols are transmitted in a given frequency band which is chosen pseudorandomly (from a set of preassigned frequency bands). Then the transmitter chooses another frequency in a pseudorandom fashion. Fading channels may produce such undesirable properties as a time-varying random amplitude response, random signal phase, the spreading of transmitted signals in the frequency domain (time-selectivity), and dispersion in the time (frequency-selectivity). The channel model that is used here and that accurately describes these examples is the wide sense stationary uncorrelated scatter (WSSUS) fading channel [3]-[17] briefly described in Chapter 5. This model is quite general and accurate in many applications. For instance, it includes the doubly selective (i.e., selective both in time and frequency) Rician channel as a special case.

In Chapter 5 we consider a system for which side information is generated at the output of the demodulator by transmitting a sequence of known test symbols in each hop and using the number of test symbols in error as a statistic upon which we decide whether a received symbol is reliable or not. If the number of test symbols in error is less than some threshold (chosen to minimize the symbol error probability), the decoder labels the received hop as unreliable and erases all the code symbols in that hop. Otherwise, the hop is declared as "good" and the symbol estimates are used by the decoder. The performance of such a system is evaluated for repetition coding with hard and soft decision combining and for Reed-Solomon codes. The improvement in performance for the Reed-Solomon coded system is substantial unlike the repetition coded system.

In Chapter 6 we consider a concatenated coded system. The inner code is used to correct and detect errors. If errors are detected one or more symbols are erased in the outer code. Then the outer code is used to correct these erasures and errors which are undetected by the first code. This technique has proven to be much more powerful than the previous technique (the test bits case), because redundancy is introduced in each hop in a more complicated fashion using coding. Final conclusions and remarks are made in Chapter 7.

CHAPTER II

CONCATENATED CODES

2.1 Introduction and Motivation

The discovery of cyclic BCH codes led to practical (low complexity) methods of designing the hardware or software for implementing for implementing the encoder and decoder. However, as the block length becomes larger the performance of BCH codes gets worse and the complexity for decoding, although polynomial, becomes substantial. Concatenation of codes, first investigated by Forney [12], is a way of constructing long block codes without requiring an impossibly complex decoder. The idea is that the channel is used with an *inner* encoder and decoder, and the combination can be viewed as a "super-channel." Then an outer code is designed for this discrete super channel as shown in Figure 2.1. The outer code is typically a Reed-Solomon code. Unfortunately, while Forney's theorem is practical, it is still not constructive in the sense that it does not tell us how to find the appropriate super channel (i.e., the inner code).

Since we are using two codes, one can design the inner decoder to learn about the channel and to help the outer code to correct as many errors as possible. If the channel statistics are fixed and known it is appropriate to design the inner code to

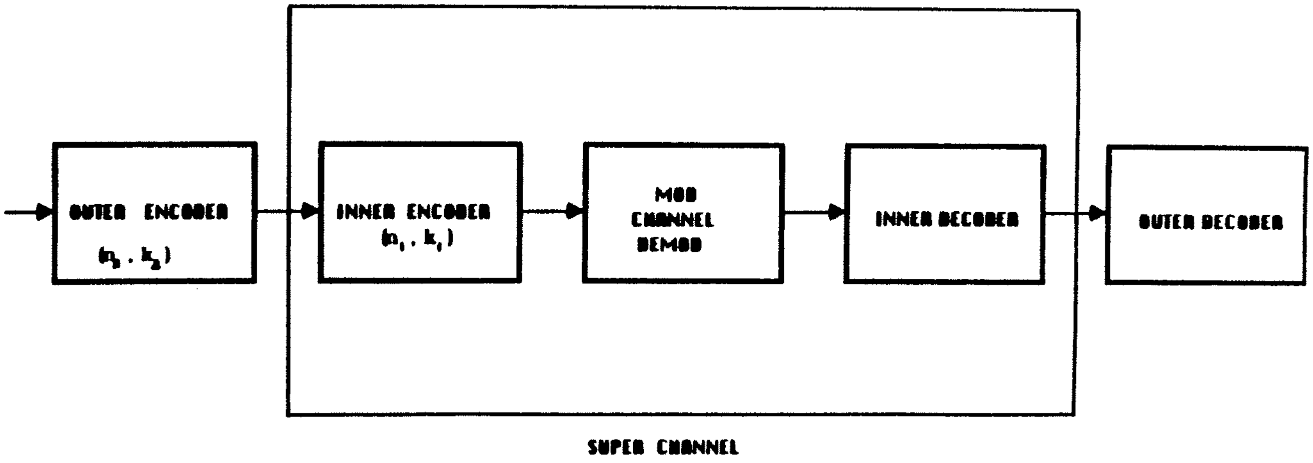

Figure 2.1: Concatenated Coding System

match the resulting super channel to the outer code which is a Reed-Solomon code. If, however, the channel parameters vary with time, or are unknown but belong to a class of channels, a technique which provides robust performance against a channel with varying statistics involves parallel decoding of the received vector by decoders which have inner decoders matched to different channel parameters and then using a selector to decide which super channel is the most probable for the duration of a codeword. More details about designing the inner decoder will be discussed in Section 2.4.

The Chapter is structured in the following way. In Section 2, we start by briefly discussing the basic concepts of first order (i.e., one outer code) concatenated coding. Section 3 describes a generalized class of codes called generalized concatenated codes. In Section 4 we investigate concatenated decoding schemes where we described a particularly attractive decoding structure called *parallel decoding*, and we discuss some of the earlier work that is relevant to our problem.

2.2 First Order Linear Concatenated Codes

In this section we describe a class of codes that are useful for all channel models introduced in Chapter 1. An (N, K, D) concatenated code with minimum distance D, block length N, and dimension K consists of two stages: an (n_2, k_2, d_{2H}) outer code C_2 with code symbols belonging to $X = GF(2^{m_2})$, $m_2 \geq 2$, and an (n_1, k_1, d_{1H}) inner code C_1 with code symbols over $U = GF(2^{m_1})$, where in general $m_2 > m_1 \geq 1$. Throughout the thesis, symbols in GF(2^m) are represented as binary m-tuples. With this in mind codewords are formed as follows. First k_2 information symbols from GF(2^{m_2}) are encoded using the outer code into n_2

symbols also in GF(2^{m_2}); the resulting m_2n_2 bits are considered to be a sequence of $\frac{m_2n_2}{m_1}$ (an integer, i.e. parameters are chosen such that m_1 divides m_2n_2) symbols in GF(2^{m_1}). Then each k_1 symbols in GF(2^{m_1}) are further encoded using the inner code into n_1 symbols in GF(2^{m_1}). If $m_1 = 1$ we have a binary concatenated code. In all cases of interest the outer code used is a Reed-Solomon code which belongs to the class of maximum distance separable codes. These are codes have the property that $d_{2H} = n_2 - k_2 + 1$, which is the maximum Hamming distance for any code with same block length and dimension.

The resultant linear concatenated code has block length $N = n_1n_2$, dimension $K = k_1k_2$, and minimum Hamming distance D that is lower bounded by $D_L = d_{2H}d_{1H}$. Moreover, $D \geq D_L$ with equality if the inner code is a code whose nonzero codewords are of constant weight.

For example the idea is illustrated below when an inner codeword corresponds to one outer code symbol:

Let the outer-encoding be characterized by a mapping

$$\mathcal{F} \, : \, \mathcal{X}^{k_2} \longrightarrow \mathcal{X}_2,$$

where $\mathcal{X}$ is the alphabet of the outer code. Let f be the i-th coordinate of $\mathcal{F}$ and let the inner-encoding be

$$g \, : \, X \longrightarrow \mathcal{U}^{n_1},$$

where U is the alphabet of the inner code. Then *concatenation* is defined by

$$F \triangleq g^{n_2}\mathcal{F} \, : \, \mathcal{X}^{k_2} \longrightarrow U^{n_1n_2} \quad .$$

where $g^{n_2}f$ is the "composition" of the two functions g^{n_2} and $\mathcal{F}$ such that $g^{n_2}\mathcal{F} \triangleq (gf_1, ..., gf_{n_2})^T$. The concatenated codeword is the $n_2 \times n_1$ matrix shown below.

$$m \xrightarrow{f} \begin{pmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ x_{n_2} \end{pmatrix} \begin{matrix} \xrightarrow{g} \\ \xrightarrow{g} \\ \cdot \\ \cdot \\ \xrightarrow{g} \end{matrix} \begin{pmatrix} u_{11} & u_{12} & \cdot & \cdot & u_{1n_1} \\ u_{21} & u_{22} & \cdot & \cdot & u_{2n_1} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ & & & & \\ u_{n_21} & u_{n_22} & \cdot & \cdot & u_{n_2n_1} \end{pmatrix}$$

where m represents a message ($\in \mathcal{X}^{k_2}$), x_i is an outer code symbol ($\in \mathcal{X}$), and u_{ij} is a channel symbol ($\in \mathcal{U}$).

Each row in the $n_2 \times n_1$ matrix is a codeword of the inner code. For each nonzero codeword of a concatenated code, there exists at least d_{2H} nonzero rows, each of the nonzero rows contains at least d_{1H} nonzero elements. Therefore, the weight of a nonzero concatenated code is at least $d_{1H}d_{2H}$.

Some binary concatenated codes of length 128 or less have been constructed in [36]. Some of these codes whose parameters are shown in Table 2.1, are superior to the best previously known linear codes with the same block length n_1n_2 and dimension k_1k_2. Specifically the constructed codes have a larger minimum distance (bound) D_L than the d_{min} of previously known codes (the actual improvement may be even better, because D_L is a lower bound to D). In the table, D_{pr} is the best known minimum Hamming distance of linear block codes, aside from concatenated codes.

2.3 Generalized Concatenated Codes

In the previous section we considered concatenated codes with only one outer code and one inner code. Now we describe a scheme due to Zinov́ev and Zyablov [38] where there are m outer codes and one inner code. This is called an m-th order

N	K	D_L	D_{pr}	n_2	k_2	d_{2H}	n_1	k_1	d_{1H}
70	9	32	30	10	3	8	7	3	4
88	12	36	34	11	3	9	8	4	4
98	12	40	36	12	3	10	8	4	4
98	16	33	32	14	4	11	7	4	3
104	12	44	40	13	3	11	8	4	4
104	24	32	30	13	6	8	8	4	4
105	20	33	32	15	5	11	7	4	3
112	12	48	44	14	3	12	8	4	4
112	24	36	33	14	6	9	8	4	4
120	12	52	51	15	3	13	8	4	4
128	16	52	48	16	4	13	8	4	4
128	32	36	32	16	8	9	8	4	4

Table 2.1: Minimum Distances of Concatenated Codes.

generalized concatenated codes. When $m = 1$ this is the first order concatenated code described in the previous Section. For simplicity, we only describe binary inner codes. Although the generalized codes will not be used in the remaining part of the thesis, they provide motivation for the parallel decoder discussed in Chapters 3 and 4.

Below we briefly describe the encoding method (shown in Figure 2.2). Consider encoding a message μ consisting of m non-overlapping smaller submessages $\{\mu_i : i = 1, \ldots, m\}$, that is $\mu = (\mu_1\mu_2 ... \mu_m)$. Each submessage μ_i is a sequence of symbols of some length b_i from GF(2^{a_i}); thus the total number of information bits to be encoded is $\sum_{i=1}^{m} a_i b_i$.

Denote by $C_i(q_i, n_2, d_{2,i}, M_{2,i})$ an outer block (possibly nonlinear) code with symbols in GF(q_i) of block length n_2 (symbols), minimum Hamming distance $d_{2,i}$, and code size $M_{2,i}$, for $i = 1, ..., m$. Let $\mathcal{F}_i$ denote the encoding for code C_i, i.e. a mapping from μ_i to a codeword in C_i. Then encoding the information μ with the outer encoder $\mathcal{F}(.)$ results in the outer codeword $\gamma = \mathcal{F}(\mu)$ which consists of m codewords $(\gamma_1, \ldots, \gamma_m)$ such that $\gamma_i \in C_i$, where the mapping $\mathcal{F}$ is defined as follows:

$$\begin{aligned} \gamma &= (\gamma_1, \ldots, \gamma_m)^T \\ &= \mathcal{F}(\mu_1, \ldots, \mu_m) \end{aligned}$$

where

$$\begin{aligned} \gamma_i = \mathcal{F}_i(\mu_i) \ &\in \ C_i(q_i, n_2, d_{2,i}, M_{2,i}) \\ &\quad i = 1, 2, \ldots, m \quad , \end{aligned}$$

and T denotes transpose. Outer encoding is equivalent to encoding with m (pos-

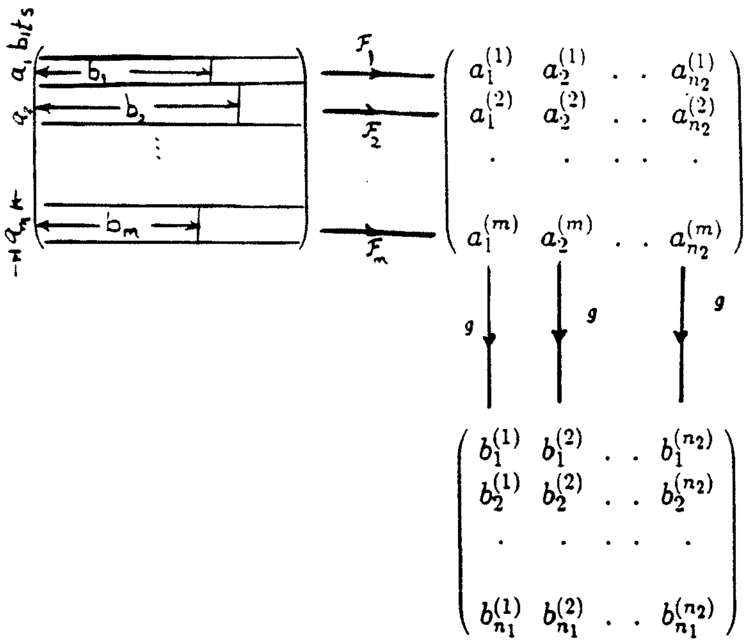
a_1 bits
b_1
a_2
b_2
a_m
b_m
k
$\mathcal{F}_1$
$\mathcal{F}_2$
$\mathcal{F}_m$
$a_1^{(1)}$ $a_2^{(1)}$ . . $a_{n_2}^{(1)}$
$a_1^{(2)}$ $a_2^{(2)}$ . . $a_{n_2}^{(2)}$
$a_1^{(m)}$ $a_2^{(m)}$ . . $a_{n_2}^{(m)}$
g
g
g
$b_1^{(1)}$ $b_1^{(2)}$ . . $b_1^{(n_2)}$
$b_2^{(1)}$ $b_2^{(2)}$ . . $b_2^{(n_2)}$
$b_{n_1}^{(1)}$ $b_{n_1}^{(2)}$ . . $b_{n_1}^{(n_2)}$

Figure 2.2: Generalized Concatenated Codes

sibly distinct) codes. That is, the outer codeword could be thought of as a matrix of size $m \times n_2$ with the i-th row containing an arbitrary codeword from code C_i, $i = 1, \ldots, m$; see Figure 2.2 for more clarification. Note that different rows, in general, have different field sizes.

Let $\gamma^{(j)}$ be the column vector, of length $\sum_{i=1}^{m} a_i$ bits, containing the j-th symbol of each outer-code codeword. That is $\gamma^{(j)}$ is the j-th column vector of the outer code in Figure 2.2. Also let $B(q, n_1, d_{1,1}, M_{1,1})$ denote an inner block code with symbols in GF(q) of block length n_1 (symbols), minimum Hamming distance $d_{1,1}$, and code size $M_{1,1}$. Then inner encoding $\mathcal{G}$ of γ results in a binary codeword c such that

$$\begin{aligned} c &= (c^{(1)}, \ldots, c^{(n_1)}) \\ &= \mathcal{G}(\gamma) \\ &= (g(\gamma^{(1)}), \ldots, g(\gamma^{(n_1)})) \end{aligned}$$

where g is an encoding using the binary code $B^1(2, n_1, d_{1,1}, M_{1,1})$. thus the column vector $c^{(j)} = g(\gamma^{(j)}) \in B^1(2, n_1, d_{1,1}, M_{1,1})$ and $M_{1,1} = q_1 q_2 \ldots q_m$. Hence, we have an $n_2 \times n_1$ (matrix) binary concatenated code of order m. The size of the resulting concatenated code is $\prod_{i=1}^{m} M_{2,i}$. The code distance is d is lower bounded below.

Let the inner code B^1 be decomposed (i.e. partitioned) into q_1 codes as follows $B^2_{i_1}(2, n_1, d_{1,2}, q_2 q_3 \ldots q_m)$, where $i_1 = 0, 1, \ldots, q_1 - 1$. This can be done by choosing all codewords that begin with symbols i_1 to be code $B^2_{i_1}$. Also, let $B^2_{i_1}$ be decomposed in a similar way into q_2 codes $B^3_{i_1,i_2}(2, n_1, d_{1,3}, q_3 \ldots q_m)$, $i_1 = 0, 1, \ldots, q_1 - 1$, $i_2 = 0, 1, \ldots, q_2 - 1$. Proceeding further we get m steps of decomposition. Codes $B^{(i)}$ are usually called inner codes. These inner codes are used in the decoding process which is not described in this thesis. Let

$$\begin{aligned} d_i &= d_{2,i}\, d_{1,i} \\ d^{(H)} &= \min_i\{d_i\}. \end{aligned}$$

Then it was shown in [38] that $d \geq d^{(H)}$.

The construction of generalized concatenated codes permits one to impose certain additional requirements on the weight spectrum of the codes and to construct codes with specified minimum and maximum distances. Moreover, these codes can be used when information symbols need to be unequally protected. That is, some of the information symbols are more valuable than others and need to be encoded with a lower rate code or with a code that has a large minimum Hamming distance.

In his survey, Sloane [30] offered a table of the best known codes. The use of generalized concatenated codes allows construction of codes which are closer to the optimal codes than those presented in Sloane's table. For lengths $n < 200$ around 60 codes were constructed in [38]. For instance we have the following example.

Example:

As inner codes we take the trivial binary codes $B^1(2,6,1,2^6)$; i.e., the set of all binary vectors of length 6. B^1 can be decomposed into

$$\begin{aligned} B_0^{(2)}(2,6,2,2^5) &= \text{all vectors with an even} \\ &\quad\ \text{number of 1's} \ , \\ B_1^{(2)}(2,6,2,2^5) &= \text{all vectors with an odd} \\ &\quad\ \text{number of 1's} \ . \end{aligned}$$

Let the external codes be the repetition code $C_1(2,30,30,2)$ and the Reed-Solomon code $C_2(2^5, 30, 15, 2^{80})$. The resultant generalized concatenated code has parameters $q = 2$, $n = 180$, $d = 30$, $k = 81$. If you delete any position of this code we get a code with parameters $n' = 179$, $d' = 29$, $k' = 81$. For specified $d' \geq 29$ and $r = n' - k' = 98$, this code has maximum length 179 from among the known codes of same distance and dimension.

2.4 Decoding of Concatenated Codes

Concatenated codes are usually considered to be effective for channels with burst errors as well as random errors. Consider a Reed-Solomon code over GF(2^m) with block length $n = 2^m - 1$; usually each symbol is represented by m bits. It is evident that such a code is not very effective on a channel with random errors, because one bit error in a symbol means the loss of the whole symbol of m bits. This is because the existing decoding algorithms has no way of taking into account the symbols that are "nearly right", and these decoders are for M-ary symmetric channels (or errors-and-erasures channels). However, the code is effective when errors and/or erasures come in bursts, since jm consecutive erroneous bits is equivalent to at most $j + 1$ symbols in error. Now if each symbol of the Reed-Solomon code is encoded into, for instance, an (n_1, m) code which corrects random errors, then the resultant concatenated code is effective when used over channels with burst as well as random errors.

In decoding concatenated codes we do decoding in two steps, i.e. one performs the inner-decoding and the outer-decoding separately: the inner-decoder processes the incoming data and uses all the available information (for instance soft decision statistics) to correct random errors and detect burst errors. The output of the inner-decoder is predominantly either correct data or stretches of burst-errors that may be flagged as being unreliable. This output then becomes the input to the Reed-Solomon decoder that corrects errors and erasures. One could think of the inner decoder as one that reduces poor quality data into medium quality data and the outer-decoder reduces medium quality data to very good quality data.

The challenging problem in constructing concatenated codes is to find the appropriate inner code and the inner decoder. The inner decoder is usually designed to "match" the super-channel to the outer code. Numerous configurations for the decoder have been considered in particular in spread-spectrum communications systems in the presence of unknown interference. Parallel decoding described below has proved to be very effective to combat such interference on channels with arbitrarily varying statistics. This was the initial motivation for our work on parallel decoding.

2.4.1 Parallel Decoding

In parallel decoding there is a family of decoders with decoding rules $\mathcal{D}_1, ..., \mathcal{D}_z$. Each decoding rule is applied to the received vector, say $\boldsymbol{y}$. Then the output of the parallel decoder is $\mathcal{D}_i(\boldsymbol{y})$ where i is chosen such that $d(\mathcal{D}_i, \boldsymbol{y})$ is the smallest for all i. The ***cost function d*** is the appropriate channel distance function (d is the Hamming distance for a q-ary symmetric channel or symmetric errors-and-erasures channel, and d is the Euclidean distance for an additive channel. Or the cost function can be the probability of error for an arbitrary channel).

For instance consider a communication system with a noise source that pulses between off and on (or a spread-spectrum communication system with partial-band interference [?]) with fixed total power, where the fraction of the band the noise is "on" is constant for a whole codeword but may vary from codeword to codeword. For these types of systems it is desirable to have reliable communication irrespective of the fraction of time (or band) the interference affects the transmitted symbol. Furthermore, assume the decoder knows if a code symbol has been subject to interference. Then one useful decoding algorithm for small pulsing times is to erase symbols that are subject to interference, then use an erasure correcting decoder to correct these erasures. If the erasure correcting capability of the code is larger than the expected number of erasures, this algorithm should perform well. For large duty times, erasing such symbols will cause too many erasures for an erasure correction decoder. In this case it is often better to perform error correction since many of the erased symbols would not have caused an error. A parallel approach using an erasure correcting decoder in parallel with an error correcting decoder will perform well for all values of the duty cycle . Thus parallel decoding with different decoding algorithms "matched" to different channels is a useful way of combatting channels with unknown interference. This idea was the original motivation behind the work presented here and in Chapters 3 and 4. Previous studies on parallel decoding in partial band interference begin with Pursley and Stark [26] where perfect side information about the interference was assumed to be available at the receiver. Castor and Stark [6] - [7] analyzed parallel decoding for partial band jamming with no side information. Also, Kim [18] analyzed parallel decoding with no side information for a multiple access environment.

Parallel decoding for concatenated codes has each decoder D_i, $i = 1, ..., z$ consisting of an inner decoder and an outer decoder. The problem then is to design the z inner decoders appropriately to optimize for some performance measure (such as

$d(.)$ above). Consider a channel as seen by the inner encoder-decoder pair. Let x be the transmitted inner codeword and w be the corresponding output of the channel. If the channel statistics are time-invariant and known then one can design the inner decoder such that the resulting super channel matches the outer code, for instance, to minimize the probability of bit error, or to maximize the capacity of the super channel. However, in many occasions the channel statistics are slowly time-varying or unknown; one could think of the channel at a given time interval (such as the duration of a codeword) as belonging to a class of channels. Different channels have different inner decoders which yield the best performance.

In parallel decoding of concatenated codes the channel output w is processed by several (say z) distinct branches; each branch consists of an inner decoder connected to an outer decoder as shown in Figure 2.3. The i-th inner decoder is characterized by a threshold Δ_i for deciding whether to erase or to output its best estimate to the outer decoder. The output of an inner decoder is either an erasure, a correct estimate, or an erroneous (inner) codeword. Then z identical bounded distance outer decoders (one for each inner decoder) are used to correct the maximum number of errors and erasures. The decoders produce z candidate estimates of the transmitted concatenated codeword. The final decoding step, performed by the decision device, is to choose the "closest" (i.e., the most likely) concatenated codeword to the received vector for a given channel, as the transmitted one. When designed properly each inner decoder is at least nearly optimal for a subclass of (super) channels. The problem then is that of finding the set of thresholds Δ_i, $i = 1, ..., z$ which optimizes some performance criteria. The algorithm for erasing inner codewords (which uses the above thresholds for an erasure criteria) depends on the channel model and will be described in Chapters 3 and 4.

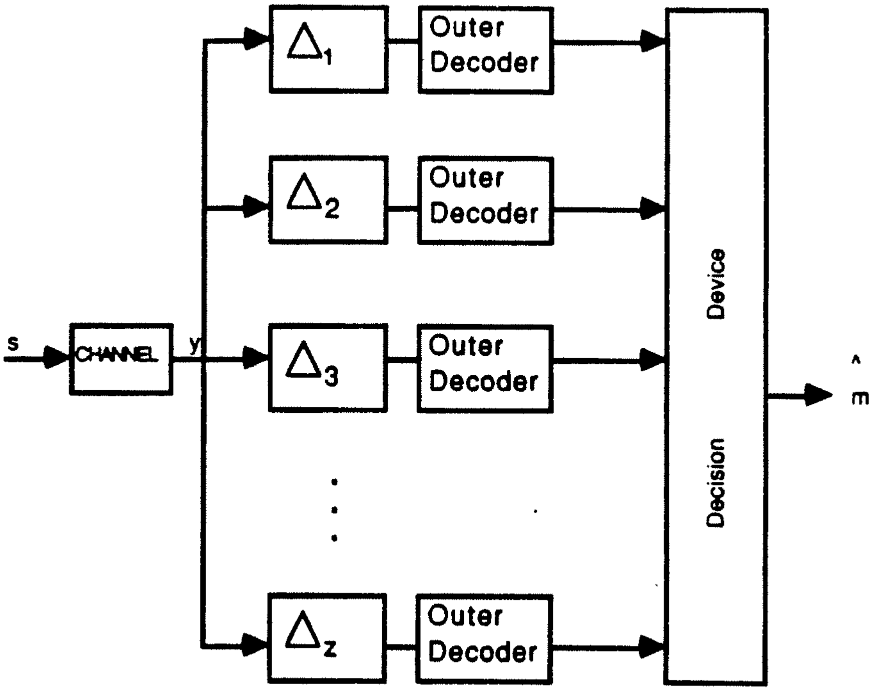

Figure 2.3: Parallel Concatenated Decoding.

In the next two Chapters we use parallel decoding for concatenated codes with the above description. As noted earlier, the performance measure of the concatenated decoder is the number or the size of errors that can be corrected and will be referred to as the "error correcting capability" of the code (using some decoding algorithm and a choice of thresholds). We define in Chapters 3 and 4 the error correcting capability in different ways depending on the channel model and the applications in mind.

2.4.2 Previous Work

We start by reviewing the work done by Forney. He considered an inner decoder which passes to the outer decoder its best estimate of the inner codeword along with a real number which indicates how reliable it supposes its estimate to be. He showed how such information can be efficiently used by the outer decoder in a method called Generalized Minimum Distance (GMD) decoding; this type of decoding allows the use of likelihood information in algebraic decoding algorithms. If y is the output of the demodulater, $p(y \mid x)$ is the probability density of the output of the demodulator given the input to the modulator is x, and $L(y) = \ln\frac{p(y|0)}{p(y|1)}$ is the bit log likelihood ratio, then the input to the outer decoder is a vector $\underline{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_n)$, ($n$ being the inner code length) such that, for some threshold T, $\alpha_i = q(y)$ such that

$$q(y) = \begin{cases} +1 & L(y) \geq T; \\ \frac{L(y)}{T} & -T \leq L(y) \leq T; \\ -1 & L(y) \leq -T. \end{cases}$$

Thus the channel considered is that of an additive channel with a soft limiter at the output. The outer decoder processes the received vector $\underline{\alpha}$ by making hard decisions on each component of the vector $\underline{\alpha}$ and then attempting to decode with a standard bounded distance error correcting method. If the (bounded distance) error correcting decoding fails, the least reliable symbol is erased and an attempt is made to decode using an errors and erasures method. If the decoding fails the two least reliable symbols are erased and decoding is attempted using an errors and erasures decoding. This continues until a codeword is decoded with $(c, r) > n - d_{min}$, or there are more erasures than can be corrected by the code, where (c, r) is the scalar product of a codeword c and the received vector r and d_{min} is the minimum Hamming distance of the inner code. Note that there is a vector successfully decoded by the inner decoder with i erasures, where i can take values in $\{0, 1, 2, ..., d_{min} - 1\}$; however, it can be shown that only $\lfloor \frac{d_{min}+1}{2} \rfloor$ attempts for decoding are necessary and sufficient for decoding. It is easy to show that errors only decoding and errors-and-erasures decoding are special cases of this algorithm.

The other parallel algorithms that have been considered in the literature are described below.

1. Parallel decoding for spread spectrum communications with jamming.

   Castor and Stark [6] - [7] and Pursley and Stark [26] have considered the use of parallel decoding schemes to mitigate the effects of partial band Gaussian jamming for spread spectrum, M-ary orthogonal, frequency hopped communication systems. Results were presented for the probability of error when the decoders performed hard decision decoding on the received vectors. Their results demonstrate that good performance is achievable in the partial band jamming environment independent of the percentage of bandwidth jammed.

2. First order concatenated codes transmitted over an M-ary symmetric channel.

The logical error correcting capability is the number of errors correctable by the decoder. It is desirable to have a decoding algorithm that corrects all error patterns with Hamming weight $\lfloor \frac{d_H-1}{2} \rfloor$ or less, where d_H is the minimum Hamming distance of the concatenated code.

Knowing that the Hamming distance of the code is lower bounded by $d_{1H}d_{2H}$, the maximum error correcting capability of the code is at least $\lfloor \frac{d_{1H}d_{2H}-1}{2} \rfloor$. Designing a decoder which employs the maximum possible capability is not a trivial task. For instance, if the inner code is used to correct $\lfloor \frac{d_{2H}-1}{2} \rfloor$ errors and the outer code to correct $\lfloor \frac{d_{1H}-1}{2} \rfloor$ errors, then the decoder $f^{-1}(g^{-1})^{n_2}(u^{n_1 \times n_2})$ can correct only about $\frac{d_{1H}d_{2H}}{4}$ errors. For this case Zyablov [40] found a parallel decoding algorithm which has the maximum possible error-correcting capability, at the expense of increasing the complexity of the decoder, and Ericson [10] has a simplified description of the algorithm. The algorithm was developed for an M-ary input and output channel and depends on errors and erasures decoding and the use of several branches with different tentative decisions. Altogether there are $\lfloor \frac{d_{1H}+1}{2} \rfloor$ branches. The inner code of the i^{th} branch corrects all error patterns with $i-1$ or fewer errors for $i = 1, 2, ..., \lfloor \frac{d_{1H}+1}{2} \rfloor$, and the outer code corrects all e_i errors and r_i erasures if $2e_i + r_i < d_{2H}$. Of the $\lfloor \frac{d_{1H}+1}{2} \rfloor$ branches, the one that has the smallest Hamming distance from the received vector is taken as the final result. Then the decoder described above corrects all error patterns with weight $\lfloor \frac{d_{1H}d_{2H}-1}{2} \rfloor$ or less. A more general treatment of concatenated codes for the M-ary input-output channel can be found in [41].

3. m-th order concatenated codes transmitted over an M-ary symmetric channel.

   The same argument, as that of the previous case, holds except d_H is the minimum Hamming distance of the generalized concatenated code. Zinov'ev [38] presented a cascaded decoding algorithm that realizes the maximum error correcting capability of a generalized concatenated code, where the distance measure is the minimum Hamming distance of the code. The algorithm is very similar to the third case summarized below.

4. m-th order binary concatenated codes transmitted over an additive channel.

   In this case the output of the channel is soft limited to between -1 and +1. Generalized Minimum Distance decoding algorithm is used where the error correcting capability is taken to be the correlation between the received vector and the inner codewords. For the above continuous-output channel, an algorithm was proposed by Dumer et. al. [9] for concatenated decoding of binary codes with respect to the generalized minimum distance introduced by Forney. This distance measure uses the correlation between the received vector and the codewords as a performance measure. A brief summary of their results follows. Consider a binary concatenated code C of order m. Then $c \in C$ is a $n_2 \times n_1$ matrix with symbols in $GF(2) = \{0,1\}$, where n_2 is the block length of the outer codes, and n_1 is the block length of the inner code(s). Denote by $d^{(H)}$ as the minimum Hamming distance of the concatenated code C. Moreover, the likelihood detector output is soft limited using $q(.)$ defined earlier.

Let z be the resulting matrix at the output of the soft limiter. Then there exists at most one codeword c such that (Theorem 3.1, Forney [12])

$$(c,z) > n_1 n_2 - d^{(H)}. \tag{2.1}$$

where (c, z) is the scalar product of the codeword c and z. Note that c is obtained by at most $\lfloor \frac{d^{(H)}+1}{2} \rfloor$ attempts using errors and erasures decoding (Theorem 3.2, Forney [12]). Decoding using 2.1 is referred to as Minimum Generalized Distance (MGD) decoding. For typically large values of $d^{(H)}$ this way of finding c is complex. Dumer (et. al.) breaks the problem of decoding into m steps by MGD decoding of inner codes then decoding (errors and erasures) with outer codes. This is called cascaded decoding which is less complex than decoding using 2.1 directly. The algorithm proposed in [9] uses parallel decoding and is characterized by a set of thresholds $T_i = \{t_1, t_2, ..., t_{M_i}\}$ for the i-th step of decoding. The two main results in [9] are:

- Only by appropriately choosing $\{\{T_i(z\} : i = 1, ..., m\}$, which is a function of the received vector z, and upon which a set of candidate (outer code) vectors are derived, the proposed algorithm outputs the correct codeword if and only if (c,z) satisfies 2.1.

  The number of thresholds (that are optimal in the above sense) could be as large as n_a. However, only $\lfloor \frac{d_{2,i}+1}{2} \rfloor$ are sufficient when determining the i-th row (i.e., the i-th step), where $d_{2,i}$ is the minimum Hamming distance of the i-th outer code $(i = 1, 2, ..., m)$.

- Restricting the decoder to $M < \min_i \lfloor \frac{d_{a,i}+1}{2} \rfloor$ fixed thresholds will result in a some loss in the error correcting capability of the code based on

(2.1). This loss is minimized by choosing *uniform* thresholds. In this case the decoding algorithm will output c if and only if

$$(c,z) > n_1 n_2 - d^{(H)} \frac{2M}{2M+1}. \tag{2.2}$$

The loss incured in using the uniform fixed thresholds is not being able to decode correctly if $\exists\ c \in C$ such that

$$n_1 n_2 - d^{(H)} \frac{2M}{2M+1} > (c,z) > n_1 n_2 - d^{(H)}. \tag{2.3}$$

In the proof of [?] a stronger result was obtained which is a lower bound on the realizable *distance* for any system of M fixed thresholds. The bound depends only on two parameters δ and Δ where

$$\begin{aligned} \delta &= d_{1,i} - t_1 \\ \Delta &= \text{maximum separation of adjacent thresholds} \\ &= \max_k (t_k - t_{k+1}). \end{aligned}$$

Next we formulate the problem for the additive channel case. With soft decision inner decoding we combine the advantage of likelihood decoding with the power of algebraic outer decoding.

Notice that the algorithm we consider is for an additive channel (as described later), and it is slightly different than Forney's decoding algorithm in that erasures are declared by comparing the reliability of each symbol to a threshold (that increases from one decoder to another). Thus we may declare three erasures for one decoding and then 5 erasures for the next without considering the case of 4 erasures. Furthermore, we test all candidate codewords to find the code closest to the received vector.

In the next two Chapters we consider soft decision decoding of first order concatenated codes with the channel being the additive or the vector additive discussed in Chapter 1. That is, no quantization or limiting is performed on the output of the demodulator. Error correcting capability is taken to be the norm of the additive noise correctable by the decoder, to be explained later in more detail. We analyze a parallel decoding structure when the reception of the code symbols is noncoherent. In this case the channel models described in the previous Chapter do not apply.

CHAPTER III

PARALLEL DECODING OF CONCATENATED CODES: COHERENT RECEPTION

In this chapter we analyze and optimize the performance of a parallel decoder for concatenated codes on additive channels. Additive channels are motivated by coherent demodulation and by the fact that quantization loses performance. Hence, we consider soft decision inner decoders, which process the received (real) vector and output either an estimate or an erasure. Each of the z inner decoders, say the i-th branch, has a different threshold Δ_i for deciding when to erase. If the received vector is within Euclidean distance Δ_i of some inner codeword, the i-th inner decoder will output the information symbols of that codeword; otherwise it will erase the inner codeword and pass the erasure to the outer decoder. Of the z concatenated codewords produced by the parallel branches, the decision device selects the one which is closest in Euclidean distance to the received vector, as the transmitted codeword.

The soft decision decoding described above assumes, implicitly, coherent detection of the received signals. Thus the notion of a correctable Euclidean distance (of the code) is easy to define and is a reasonable performance criteria to consider.

This is particularly true when considering additive white gaussian noise channel such as in deep space or satellite communications.

The principal contribution of this chapter is to show how to determine the optimal choice of $\Delta_i, i = 1, 2, ..., z$, in order to maximize the guaranteed maximum Euclidean distance correctable by the concatenated code, which we referred to in Chapter 1 as the error correcting capability of the code. Furthermore, we show that for moderate values of z (3 or 4) this decoding algorithm can correct errors up to something close to half the minimum Euclidean distance of the concatenated code.

The error correcting capability of the code is evaluated by analyzing a game situation with the channel and the decoder as opponents. The strategy of the channel consists of choosing a noise vector. The strategy for the decoder consists of choosing the thresholds for declaring erasures in each decoding branch. The objective of the channel is to minimize the Euclidean length of the noise it must add to the transmitted signal in order to cause a decoding error. The objective of the decoder is to choose the thresholds to maximize the noise length necessary to cause an error (i.e., to maximize the error correcting capability of the code).

3.1 Game Theoretic Formulation

Consider the parallel decoding algorithm for a concatenated code and an additive channel, such that the inner decoder of the i-th branch decodes all vectors within Euclidean distance Δ_i of some codeword, $i = 1, 2, ..., z$, (note that $\Delta_i \in (0, \frac{d_{1E}}{2})$ as in Figure 2.1), where z is the number of decoder branches, and d_{1E} is the Euclidean distance of the inner code. Errors are detected in the i-th branch if a received vector (corresponding to an inner codeword) does not fall in any sphere of radius Δ_i, and centered around an inner codeword. In this case the inner decoder outputs an erasure to the outer decoder. The decision device in this case chooses among the z candidate codewords the one which is closest in Euclidean distance to the received vector, as the transmitted one.

The Euclidean distance d_{1E} of the inner code depends on the modulation being employed. For example, if we are using a binary inner code and antipodal signaling (with unit energy) then $d_{1E} = 2\sqrt{d_{1H}}$, d_{1H} being the Hamming distance of the inner code, whereas for an M-ary inner code and M-ary orthogonal signal set $d_{1E} = \sqrt{2d_{1H}}$. The exact relation between Euclidean distance and Hamming distance is not important in this thesis since we are dealing with correcting capability with respect to Euclidean distance.

The forthcoming analysis to evaluate the error correcting capability of the parallel decoder is valid for outer codes defined over GF(q), for all $q = 2^m$, $m \geq 1$, and inner codes over GF(2^l) (usually $l < m$). The outer code is a Reed-Solomon code that corrects any set of e errors and τ erasures if $2e + \tau < d_{2H}$. As mentioned earlier, of the at most z possible decoding results, the one that has the smallest Euclidean distance from the received combination is taken as the final result.

We now discuss the additive channel model in more detail (refer to Figures 1.1

and 1.2). Let the transmitted vector be given by

$$\underline{s} = \begin{pmatrix} s_{11} & s_{12} & \cdots & s_{1n_1} \\ s_{21} & s_{22} & \cdots & s_{2n_1} \\ . & . & . & . \\ . & . & . & . \\ s_{n_2 1} & .. & . & s_{n_2 n_1} \end{pmatrix},$$

(for binary codes with antipodal modulation $s_{ij} \in \{-1,+1\}$; for M-ary orthogonal signaling $s_{ij} \in \{-1,+1\}^M$) then the received vector is

$$\underline{y} = \underline{s} + \underline{\nu}$$

where

$$\underline{\nu} = \begin{pmatrix} \nu_{11} & \nu_{12} & \cdots & \nu_{1n_1} \\ \nu_{21} & \nu_{22} & \cdots & \nu_{2n_1} \\ . & . & . & . \\ . & . & . & . \\ \nu_{n_2 1} & .. & . & \nu_{n_2 n_1} \end{pmatrix}$$

is the additive noise matrix; that is $y_{ij} = s_{ij} + \nu_{ij}$, where ν_{ij} is a noise vector. The square of the Euclidean distance between $\underline{s}$ and $\underline{y}$ is

$$t(\underline{\nu}) \triangleq d_E^2(\underline{s}, \underline{y}) = \sum_{i=1}^{n_2} \sum_{j=1}^{n_1} \nu_{ij} \nu_{ij}^T \tag{3.1}$$

where T denotes transpose. We call $t(\underline{\nu})$ the channel loss. If d_E and d_H are, respectively, the Euclidean distance and Hamming distance of the concatenated code, then for any transmitted concatenated codewords $\underline{s}$ and $\underline{s}'$ we have

$$d_E^2 \geq \sum_i \sum_j (s_{ij} - s'_{ij})(s_{ij} - s'_{ij})^T$$

$$d_E \;\geq\; d_{1E}\sqrt{d_{2H}}. \tag{3.2}$$

We will be interested in bounded-distance decoders which decode correctly any vector within Euclidean distance $\frac{d_{1E}\sqrt{d_2H}}{2}$ from the transmitted one.

Recall that we are interested in evaluating and maximizing the error correcting capability of concatenated codes using the Euclidean distance as a measure of performance. For any code this capability depends on the particular choice of concatenated decoding algorithm. To find the best decoding thresholds, it is appropriate to treat the problem as a game with two players. The strategies of Player 1, called the decoder, consist of all vectors $\Lambda_z = \{\Delta_1, \ldots, \Delta_z\}$, $0 \leq \Delta_1 < \ldots < \Delta_z < \frac{d_{1E}}{2}$. The strategies of Player 2, called the channel, consist of all noise vectors $\underline{\nu}$. Player 1 chooses the set Λ_z to maximize the error correcting capability of the code; that is to maximize the minimum noise vector length that will cause an error. Player 2 chooses $\underline{\nu}$ to minimize the (squared) noise length $t(\underline{\nu})$ to guarantee each decoding branch is not decoded correctly. By convention we let $\Delta_0 = 0$ and $\Delta_{z+1} = d_1 - \Delta_z$. Hence, associated with the game are two programs:

<u>Program I (Decoder's Program)</u>

$$\gamma = \max_{\Lambda_z} \;\min_{\underline{\nu}}\; t(\underline{\nu})$$

where the minimum over $\underline{\nu}$ is such that the errors in the received vector <u>cannot</u> be corrected by the parallel decoder (i.e., the output of each decoder branch is incorrect).

<u>Program II (Channel's Program)</u>

$$\gamma\prime = \min_{\underline{\nu}} \;\max_{\Lambda_z}\; t(\underline{\nu})$$

where the minimum over $\underline{\nu}$ is also conditioned on the decoder output being incorrect.

It is clear from the above programs that $\gamma \leq \gamma\prime$; therefore, for a pessimistic analysis (observed by the decoder) we are interested in investigating program I which we do in several steps; however, we need some additional definitions that are convenient in finding optimal solutions to our program. The solution to program II is trivial and is shown in Appendix B. It is possible for the parallel decoder to choose an erroneous concatenated codeword in some cases when $t(\underline{\nu})$ is less than γ; however, Reed-Solomon decoders are more likely to fail than being in error; thus γ is very close to the maximum error correcting capability of the code.

Now define for any $m \geq 0$ and any $\underline{\nu}$

$$x_m = \text{ number of rows of } \underline{\nu} \text{ such that} \sum_{j=1}^{n_1} \nu_{ij}\nu_{ij}^T = m.$$

Let $N(m)$ be a counting measure with $N(m) = 1$ if $x_m > 0$ and $N(m) = 0$, otherwise; then

$$t(\underline{\nu}) = \sum_{m \in [0,\infty)} m x_m \equiv \int_0^\infty m x_m dN(m). \tag{3.3}$$

The above summation is well defined because $x_m = 0$ except for a finite number of m's.

In solving for γ, we are interested in evaluating, for a fixed threshold set Λ_z, $\min_{\underline{\nu}} t(\underline{\nu})$, where the minimum over $\underline{\nu}$ is such that the received vector cannot be corrected by the parallel decoder. For a given $\underline{\nu}$, let τ_k be the number of induced erasures in the outer code for the k-th decoding branch. Also let e_k be the number of induced errors in the outer code for the k-th branch. Further let $\tau = (\tau_1, \tau_2, ..., \tau_z)$ and $e \; = (e_1, ..., e_z)$. Then $\underline{\nu}$ causes the parallel decoder to output an incorrect codeword if (τ_k, e_k) for all $k = 1, ..., z$ causes the k-th outer decoder to output an incorrect codeword, i.e. $2e_k + \tau_k \geq d_{2H}$ (recall that incorrect codeword refers to decoding failure or erroneous decoding). Thus we have the following

$$\min_{\underline{\nu}} t(\underline{\nu}) \; = \; \min_{\tau, e} \; \min_{\underline{\nu}:\tau, e} t(\underline{\nu}),$$

where the minimization over $\underline{\nu}$ is such that the channel causes τ erasures and e errors, and the minimization over τ and e is such that $2\tau_k + e_k \geq d_{2H}, \; k = 1, ..., z$.

Define $\gamma(\Lambda_z, \tau, e)$ as the optimum (i.e. minimum) channel loss given the erasures τ, errors e, and fixed thresholds Λ_z. That is

$$\gamma(\Lambda_z, \; \tau, e) \; \triangleq \; \min_{\underline{\nu}:\tau, e} t(\underline{\nu})$$

For the channel to cause an erasure in the j-th symbol (inner codeword) at the output of the i-th inner decoder, the row vector $\underline{\nu}_j = (\nu_{j1}, \nu_{j2}, ..., \nu_{jn_1})$ need only have length just greater than Δ_i. For the channel to cause an error in the k-th symbol (inner codeword) at the output of the i-th inner decoder, $\underline{\nu}_k$ must have length no smaller than $d_{1E} - \Delta_i$ and this is sufficient if the vector is directed towards a codeword that is at a distance d_{1E} (there is always such a codeword). Since the channel should always choose the direction to be that towards the minimum Euclidean distance codeword, the above minimization can be expressed in terms of $\underline{x} = \{x_m : m \in [0, \infty)\}$; that is the number of vectors $\underline{\nu}_i, \; 1 \leq i \leq n_2$ of a length m. Thus we have

$$\begin{aligned}\gamma(\Lambda_z,\ \tau,e) &= \min_{\underline{\nu}:\tau,\,e} t(\underline{\nu}) \\ &= \min_{\underline{x}:\tau,\,e} t(\underline{\nu})\end{aligned}$$

We proceed for solving Program I in steps as following.

- First we obtain a general form for $\gamma(\Lambda_z,\ \tau,e)$ for the optimal channel strategy, given Λ_z, τ, and e.
- We then minimize $\gamma(\Lambda_z,\tau,e)$ over all vectors e such that the output of each decoder branch is in error, for a given τ; i.e., find

  $$\gamma(\Lambda_z,\tau) = \min_{e} \gamma(\Lambda_z,\tau,e),$$

  where the minimum over e is such that $2e_k + \tau_k \ \geq d_{2H},\ k = 1,2,...,z$; this condition assures outer decoding failure.
- Next we find

  $$\gamma(\Lambda_z) = \min_{\tau} \gamma(\Lambda_z,\tau);$$

  where the minimum over τ is such that $\tau_k \geq \tau_{k+1},\ \ k = 1,...,z$. This constraint follows from the fact that the erasure region gets smaller for larger Δ_i.
- To find the optimal decoder strategy, we solve for

  $$\gamma = \max_{\Lambda_z} \gamma(\Lambda_z).$$

The values of τ_k and e_k are related to the coordinates of $\underline{x}$ by the expressions

$$\tau_k + e_k = \int_{\Delta_k^2}^{\infty} x_m dN(m) \tag{3.4}$$

$$k = 1, 2, ..., z$$

For a Δ_k distance correcting code, error detection will certainly occur when $\Delta_k^2 \leq \sum_{j=1}^{n_1} \nu_{ij}^2 \leq (d_{1E} - \Delta_k)^2$ (see Figure 3.1). Either error detection or incorrect decoding will occur if $\sum_{j=1}^{n_1} \nu_{ij}^2 > (d_{1E} - \Delta_k)^2$. Therefore we obtain the following bounds

$$r_k \geq \int_{\Delta_k^2}^{(d_{1E}-\Delta_k)^2} x_m dN(m) \tag{3.5}$$

$$e_k \leq \int_{(d_{1E}-\Delta_k^+)^2}^{\infty} x_m dN(m) \tag{3.6}$$

$$k = 1, 2, ..., z.$$

From (3.3) we can write the channel loss as

$$\begin{aligned} t &= \int_0^{(d_{1E}-\Delta_z)^2} m x_m dN(m) + ... + \int_{(d_{1E}-\Delta_2^+)^2}^{(d_{1E}-\Delta_1)^2} m x_m dN(m) + \\ &\quad \int_{(d_{1E}-\Delta_1^+)^2}^{\infty} m x_m dN(m) \\ &\geq \int_0^{(d_{1E}-\Delta_z)^2} m x_m dN(m) + ... + \int_{(d_{1E}-\Delta_2^+)^2}^{(d_{1E}-\Delta_1)^2} m x_m dN(m) \\ &\quad + (d_{1E} - \Delta_1^+)^2 \int_{(d_{1E}-\Delta_1^+)^2}^{\infty} x_m dN(m) \\ &\geq \int_0^{(d_{1E}-\Delta_z)^2} m x_m dN(m) + ... + \int_{(d_{1E}-\Delta_2^+)^2}^{(d_{1E}-\Delta_1)^2} m x_m dN(m) + (d_{1E} - \Delta_1^+)^2 e_1 \end{aligned} \tag{3.7}$$

with equality if $e_1 = \int_{(d_{1E}-\Delta_1^+)^2}^{\infty} x_m dN(m)$; this occurs when $x_m = e_1$ for $m = (d_{1E} - \Delta_1^+)^2$ and $x_m = 0$ for $m > (d_{1E} - \Delta_1^+)^2$. Similarly, going backward in equation (3.7) we conclude that the channel loss is minimum when equality is attained in (3.6) for $k = 1, 2, ..., z$. With equality achieved in (3.6) equality is also achieved in (3.5), as implied by (3.4). Then in the case of optimal channel strategy, we can rewrite (7) and (8) in a form more convenient for later use, as follows:

$$e_k = \int_{(d_{1E}-\Delta_k^+)^2}^{(d_{1E}-\Delta_{k-1})^2} x_m dN(m) + \int_{(d_{1E}-\Delta_{k-1}^+)^2}^{\infty} x_m dN(m)$$

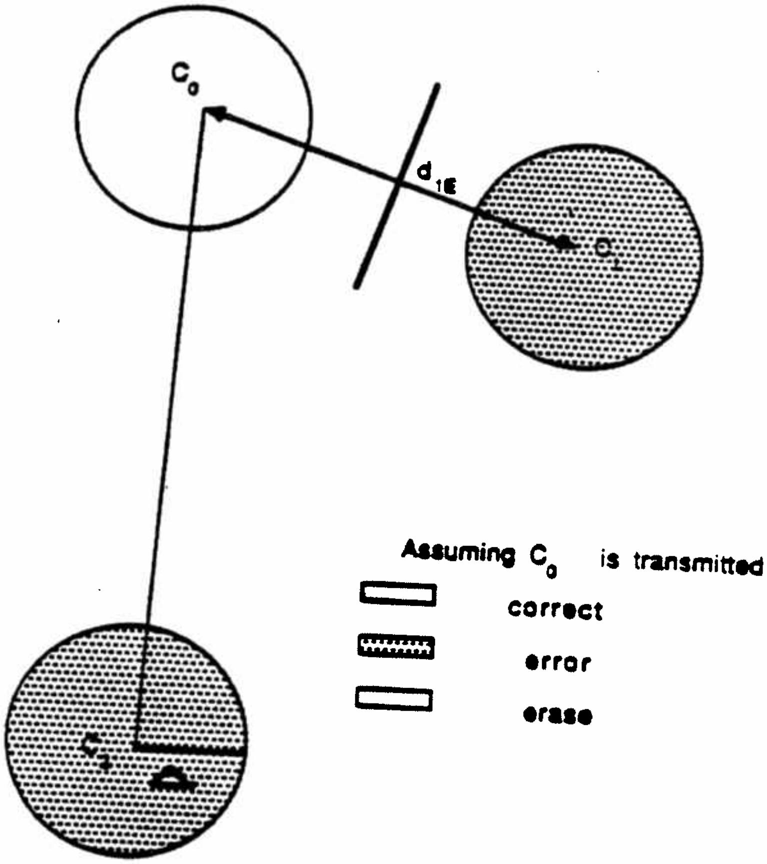

Figure 3.1: Decoding Regions For the *i*-th Branch

$$= \int_{(d_{1E}-\Delta_k^+)^2}^{(d_{1E}-\Delta_{k-1})^2} x_m dN(m) + e_{k-1} \quad k = 2, 3, ..., z \tag{3.8}$$

$$\begin{aligned}\tau_k &= \int_{(\Delta_k)^2}^{(\Delta_{k+1}^-)^2} x_m dN(m) \\ &\quad + \int_{\Delta_{k+1}^2}^{(d_{1E}-\Delta_{k+1})^2} x_m dN(m) + \int_{(d_{1E}-\Delta_{k+1}^+)^2}^{(d_{1E}-\Delta_k)^2} x_m dN(m) \\ &= \int_{\Delta_k^2}^{(\Delta_{k+1}^-)^2} x_m dN(m) + \tau_{k+1} + e_{k+1} - e_k \quad k = 1, 2, ..., z-1 \end{aligned}\tag{3.9}$$

Also,

$$e_1 = \int_{(d_{1E}-\Delta_1^+)^2}^{\infty} x_m dN(m) \tag{3.10}$$

$$\tau_z = \int_{\Delta_z^2}^{(d_{1E}-\Delta_z)^2} x_m dN(m) \tag{3.11}$$

It follows from (3.8) and (3.9) that

$$e_k \leq e_{k+1} \quad k = 1, 2, ..., z-1$$

$$\tau_k \geq \tau_{k+1} \quad k = 1, 2, ..., z-1$$

The above inequalities will be used as optimization constraints later in this section. We estimate the conditional (conditioned on $\boldsymbol{\Delta}_z$,$\boldsymbol{\tau}$, and $\boldsymbol{e}$) channel loss $\gamma(\boldsymbol{\Delta}_z$,$\boldsymbol{\tau}$,$\boldsymbol{e})$ in the case of its optimal strategy under the condition that there are τ_k erasures and e_k errors in the outer code in the k-th decoding branch; that is,

$$\gamma(\boldsymbol{\Delta}_z, \boldsymbol{\tau}, \boldsymbol{e}) = \min_{\underline{x}} \int_0^{\infty} m x_m dN(m) \tag{3.12}$$

where the minimum is over $\underline{x}$ satisfying (3.8)-(3.11). Before we proceed, we note that

$$\min \int_{j_1}^{j_2} m x_m dN(m) = j_1 a \tag{3.13}$$

where the minimum is over all x_m , $j_1 \leq m \leq j_2$ such that

$$\int_{j_1}^{j_2} x_m dN(m) = a \ .$$

Furthermore, this occurs when $x_{j_1} = a$ and $x_j = 0$ for $j_1 < j \leq j_2$. Then (3.12) can be specified as follows:

$$\begin{aligned}
\gamma(\Delta_z, \tau, e) &= \min_{\mathbf{x}} \int_0^{\infty} m x_m dN(m) \\
&= \min_{\mathbf{x}} \{ \int_0^{(\Delta_1^-)^2} m x_m dN(m) + \int_{\Delta_1^2}^{(\Delta_2^-)^2} m x_m dN(m) \\
&\quad + \int_{\Delta_2^2}^{(\Delta_3^-)^2} m x_m dN(m) + ... + \int_{\Delta_{z-1}^2}^{(\Delta_z^-)^2} m x_m dN(m) \\
&\quad + \int_{\Delta_z^2}^{(\Delta_{z+1}^-)^2} m x_m dN(m) + \int_{(d_{1E}-\Delta_z^+)^2}^{(d_{1E}-\Delta_{z-1})^2} m x_m dN(m) + ... \\
&\quad + \int_{(d_{1E}-\Delta_2^+)^2}^{(d_{1E}-\Delta_1)^2} m x_m dN(m) + \int_{(d_{1E}-\Delta_1^+)^2}^{\infty} m x_m dN(m) \}.
\end{aligned}$$

Using (3.13) and (3.8)-(3.11) we get

$$\begin{aligned}
\gamma(\Delta_z, \tau, e) &= \Delta_1^2(\tau_1 - \tau_2 + e_1 - e_2) + ... + \Delta_{z-1}^2(\tau_{z-1} - \tau_z + e_{z-1} - e_z) \\
&\quad + \Delta_z^2 \tau_z + (d_{1E} - \Delta_z)^2 (e_z - e_{z-1}) + ... \\
&\quad + (d_{1E} - \Delta_2)^2 (e_2 - e_1) + (d_{1E} - \Delta_1)^2 e_1 \\
&= \sum_{k=1}^{z-1} \Delta_k^2 (\tau_k - \tau_{k+1} + e_k - e_{k+1}) + \Delta_z^2 \tau_z \\
&\quad + \sum_{k=2}^{z} (d_{1E} - \Delta_k)^2 (e_k - e_{k-1}) + (d_{1E} - \Delta_1)^2 e_1 \\
&= \sum_{k=1}^{z} \{ (\Delta_k^2 - \Delta_{k-1}^2) \tau_k + (2\Delta_k^2 - \Delta_{k-1}^2 - \Delta_{k+1}^2 \\
&\quad + 2 d_{1E} \Delta_{k+1} - 2 d_{1E} \Delta_k) e_k \} \\
&\equiv \sum_{k=1}^{z} \gamma^{(k)}(\tau_k, e_k),
\end{aligned} \tag{3.14}$$

where

$$\gamma^{(k)}(\tau_k, e_k) = (\Delta_k^2 - \Delta_{k-1}^2)\tau_k + (2\Delta_k^2 - \Delta_{k-1}^2 - \Delta_{k+1}^2 + 2d_{1E}\Delta_{k+1} - 2d_{1E}\Delta_k)e_k$$

From (3.14) we can determine the minimum channel loss in the case of a channel optimal strategy under the condition that the output of the $k-$th inner decoder has τ_k erasures and e_k errors. Next we use this result to determine the channel loss $\gamma(\Lambda_z,\tau)$ in the case of the optimal channel strategy under the condition that τ_k erasures occur in the $k-th$ decoding trial and decoding is unsuccessful, i.e. the k-th outer decoder fail to decode or outputs an incorrect codeword. For all the decoder branches to be unsuccessful, it is necessary and sufficient that the following z inequalities hold:

$$e_k \geq \max(\frac{d_{2H} - \tau_k}{2}, 0) \quad k = 1, 2, ..., z. \tag{3.15}$$

from this, we obtain

$$\gamma(\Lambda_z, \tau) = \min_e \gamma(\Lambda_z, \tau, e) \tag{3.16}$$

where the minimum is taken over e satisfying (3.15). From (3.15) and (3.16) we have

$$\begin{aligned}\gamma(\Lambda_z, \tau) &= \min_e \gamma(\Lambda_z, \tau, e) \\ &= \sum_{k=1}^{z} \min_{e_k} \gamma^{(k)}(\tau_k, e_k) \\ &= \sum_{k=1}^{z} \gamma^{(k)}(\tau_k);\end{aligned} \tag{3.17}$$

where

$$\gamma^{(k)}(\tau_k) = \begin{cases} (\Delta_k^2 - \Delta_{k-1}^2)\tau_k + b(k)\lfloor\frac{d_{2H}-\tau_k+1}{2}\rfloor & 0 \leq \tau_k \leq d_{2H} \\ (\Delta_k^2 - \Delta_{k-1}^2)\tau_k & \tau_k \geq d_{2H} \end{cases}, \tag{3.18}$$

where $b(k) = 2\Delta_k^2 - \Delta_{k-1}^2 - \Delta_{k+1}^2 + 2d_{1E}\Delta_{k+1} - 2d_{1E}\Delta_k$. We use (3.18) to minimize over all vectors $\boldsymbol{\tau}$ such that

$$\tau_k \geq \tau_{k+1}.$$

Then we have

$$\gamma(\Lambda_z) = \min_{\boldsymbol{\tau}} \gamma(\Lambda_z, \boldsymbol{\tau}) = \min_{d_{2H} \geq \tau_1 \geq 0} \{\gamma^{(1)}(\tau_1) + \min_{\tau_1 \geq \tau_2} \{\gamma^{(2)}(\tau_2) + \ldots + \min_{\tau_{z-1} \geq \tau_z} \gamma^{(z)}(\tau_z)\} \ldots\}\}. \tag{3.19}$$

We now evaluate the above minimization using the principle of dynamic programming, sometimes called Bellman's optimality (suppose one has a system characterized by a state, and some decision is performed which depends on that state, then an optimal strategy has the property that regardless of the initial state and initial decision the next decision must determine an optimal strategy with respect to the state obtained as a result of the initial decision.) The following lemma is also needed to find the solution to (3.20).

Lemma: Let $\Upsilon = \{\tau : \tau \equiv d_{2H} \pmod 2\}$, and $0 < a < b$, we have

$$\begin{aligned} \min_{y \leq \tau \leq z} \{a\tau + b\lfloor (d - \tau + 1)/2 \rfloor\} &= \frac{db}{2} + \min_{y \leq \tau^* \leq z} \tau^*(2a - b)/2 \\ &= db/2 + \frac{1}{2}\min\{x^*(2a-b), y^*(2a-b)\} \end{aligned} \tag{3.20}$$

where $\tau^* \in \Upsilon$, $x^* = \max\{\tau \in \Upsilon : x \geq \tau\}$, and $y^* = \min\{\tau \in \Upsilon : y \leq \tau\}$.

Proof: Let $f(\tau) \triangleq a\tau + b\lfloor (d - \tau + 1)/2 \rfloor$. Then

$$f(\tau^* - 1) = a(\tau^* - 1) + b\lfloor (d - \tau^* + 1 + 1)/2 \rfloor = a\tau^* + b(d - \tau^*)/2 + b - a;$$

$$f(\tau^*) = a\tau^* + b\lfloor (d - \tau^* + 1)/2 \rfloor = a\tau^* + b(d - \tau^*)/2;$$

$$f(\tau^* + 1) = a(\tau^* + 1) + b\lfloor (d - \tau^* - 1 + 1)/2 \rfloor = a\tau^* + b(d - \tau^*)/2 + a.$$

The lemma follows from the relations $f(\tau^* - 1) > f(\tau^*)$ and $f(\tau^* + 1) > f(\tau^*)$.

According to the lemma, the minimum in (3.19) occurs at point τ whose elements satisfy

$$\tau_1 \equiv \tau_2 \equiv \ldots \equiv \tau_z \equiv d_{2H} \pmod 2 \tag{3.21}$$

Using (3.20) and (3.21) permits us to write (3.19) as

$$\gamma(\Lambda_z) = \sum_{k=1}^{z} \frac{d_{2H}}{2} b(k) + \min_{\tau^*} \sum_{m=1}^{z} \tau^*_m (\Delta^2_{k+1} - \Delta^2_{k-1} + 2d_{1E}\Delta_k - 2d_{1E}\Delta_{k+1})/2$$

where

$$\begin{aligned}
\sum_{k=1}^{z} b(k) &= \sum_{k=1}^{z} (2\Delta^2_k - \Delta^2_{k-1} - \Delta^2_{k+1} + 2d_{1E}\Delta_{k+1} - 2d_{1E}\Delta_k) \\
&= \sum_{k=1}^{z} (\Delta^2_k - \Delta^2_{k-1}) + \sum_{k=1}^{z} (\Delta^2_k - \Delta^2_{k+1}) + 2d_{1E} \sum_{k=1}^{z} (\Delta_{k+1} - \Delta_k) \\
&= \Delta^2_z + (\Delta^2_1 - \Delta^2_{z+1}) + 2d_{1E}(\Delta_{z+1} - \Delta_1) \\
&= \Delta^2_z + \Delta^2_1 - d^2_{1E} - \Delta^2_z + 2d_{1E}\Delta_z + 2d_{1E}(d_{1E} - \Delta_z - \Delta_1) \\
&= \Delta^2_1 + d^2_{1E} - 2d_{1E}\Delta_1 \\
&= (d_{1E} - \Delta_1)^2 .
\end{aligned}$$

since $\Delta_{z+1} = d_{1E} - \Delta_z$. Hence,

$$\gamma(\Lambda_z) = \frac{d_{2H}(d_{1E} - \Delta_1)^2}{2} + \min_{\tau^*} \sum_{m=1}^{z} \tau^*_m (\Delta^2_{k+1} - \Delta^2_{k-1} + 2d_{1E}\Delta_k - 2d_{1E}\Delta_{k+1})/2 \ . \tag{3.22}$$

Two cases for d_{2H} are considered.

CASE I: $d_{2H} \equiv 0 \pmod 2$, then according to the lemma above the minimum is attained at one of the $z + 1$ points

$$\tau \in S = \{(0,0,\ldots,0), (d_{2H}, 0, \ldots, 0), (d_{2H}, d_{2H}, 0, \ldots, 0), \ldots, (d_{2H}, \ldots, d_{2H})\}$$

then

$$\gamma(\Lambda_z) = \frac{d_{2H}(d_{1E} - \Delta_1)^2}{2} + \min_{\tau^*} \sum_{m=1}^{z} \tau^*_m (\Delta^2_{k+1} - \Delta^2_{k-1} + 2d_{1E}\Delta_k - 2d_{1E}\Delta_{k+1})/2$$

$$
\begin{aligned}
&= \frac{d_{2H}(d_{1E}-\Delta_1)^2}{2} + \min_k \sum_{m=1}^{k} d_{2H}(\Delta_{m+1}^2 - \Delta_{m-1}^2 + 2d_{1E}\Delta_m - 2d_{1E}\Delta_{m+1})/2 + 0 \\
&= d_{2H}(d_{1E}-\Delta_1)^2/2 + d_{2H}/2 \min_k((d_{1E}-\Delta_k)^2 - (d_{1E}-\Delta_1)^2 + \Delta_{k-1}^2) \\
&= \min_k \frac{d_{2H}}{2}((d_{1E}-\Delta_k)^2 + \Delta_{k-1}^2) \quad ,k = 1,2,...,z+1 \qquad (3.23)
\end{aligned}
$$

<u>CASE II:</u> $d_{2H} \equiv 1 \pmod 2$; this implies that

$$\tau \in S' = \{(1,1,...,1),(d_{2H},1,...,1),(d_{2H},d_{2H},1,...,1),...,(d_{2H},...,d_{2H})\}$$

and the channel loss is given by

$$\gamma(\boldsymbol{\Delta}_z) = \sum_{k=1}^{z} \frac{d_{2H}}{2} b(k) + \min_k \left(\sum_{m=1}^{k-1} d_{2H}(\Delta_{m+1}^2 - \Delta_{m-1}^2 + 2d_{1E}\Delta_m - 2d_{1E}\Delta_{m+1})/2 + \right.$$

$$\sum_{m=k}^{z} (\Delta_{m+1}^2 - \Delta_{m-1}^2 + 2d_{1E}\Delta_m - 2d_{1E}\Delta_{m+1})/2;$$

the first two sums have been calculated above, and it is easy to show that

$$
\begin{aligned}
\sum_{m=k}^{z} (\Delta_{m+1}^2 - \Delta_{m-1}^2 + 2d_{1E}\Delta_m - 2d_{1E}\Delta_{m+1}) &= \\
(\Delta_{z+1}^2 - \Delta_k^2) + (\Delta_z^2 - \Delta_{k-1}^2) - 2d_{1E}(\Delta_{z+1} - \Delta_k) &= \\
2\Delta_z^2 - (d_{1E} - \Delta_k)^2 - \Delta_{k-1}^2;
\end{aligned}
$$

hence,

$$\gamma(\boldsymbol{\Delta}_z) = \min_k \frac{(d_{2H}-1)}{2}((d_{1E}-\Delta_k)^2 + \Delta_{k-1}^2) + \Delta_z^2 \quad ,k = 1,2,...,z+1 \qquad (3.24)$$

The minimum channel loss in the case of its optimal strategy under the condition of decoding failure is given in (3.23) and (3.24), which also shows the dependance of the error correcting capability of the concatenated code as a function of the parameters $\boldsymbol{\Delta}_z$, d_{2H}, d_{1E}.

3.2 Optimal Decoder Strategy

The optimal decoder strategy maximizes the channel loss for a fixed z; i.e., it is the solution of the equation

$$\Lambda_z^* = \arg \max_{\Lambda_z} \gamma(\Lambda_z)$$

For d_{2H} even the resulting channel loss is given by

$$\gamma = \max_{\Lambda_z} \min_{k} \frac{d_{2H}}{2}[(d_{1E} - \Delta_k)^2 + \Delta_{k-1}^2] \quad k = 1, 2, ..., z + 1.$$

Let $\delta_k = \frac{\Delta_k}{d_{1E}}$ and $f(\delta_k, \delta_{k-1}) = (1 - \delta_k)^2 + \delta_{k-1}^2$. It is shown in Appendix A that

$$\max_{\Lambda_z} \min_{k} f(\delta_k, \delta_{k-1})$$

is obtained when $f(\delta_k, \delta_{k-1}) = \alpha_z$, $k = 1, 2, ..., z + 1$; where α_z is some unique constant that depends on z. Given the conditions $\delta_0 = 0$ and $\delta_{z+1} = 1 - \delta_z$; the $z + 1$ equations for $\delta_1, ..., \delta_z, \alpha_z$ can be solved numerically.

We now can solve for the optimal value of the thersholds Λ_z; that is we can determine the decoder strategy that maximizes the error correcting capability of concatenated codes when used on q-ary additive channels (recall that $\Delta_i = \delta_i \, d_{1E}$). These thresholds are strictly a function of the number of decoder branches used in the parallel decoder. One expects that the error correcting capability of the code improves with larger z. This behaviour is analyzed below.

Similar argument as above holds for d_{2H} odd. In this case the channel loss is given by

$$\gamma = \max_{\Delta_z} \min_{k} \frac{(d_{2H}-1)}{2}[(d_{1E}-\Delta_k)^2 + \Delta_{k-1}^2] + \Delta_z^2, \qquad k = 1, 2, ..., z+1.$$

It is shown in Appendix C that the solution for the above game is the same as for the even case. That is, the strategy is the same and the resulting losses are equal regardless if d_{2H} is odd or even.

The error-correcting capability (squared) of the code when the decoder uses its best strategy is

$$\begin{aligned} \gamma &= \frac{d_{2H} d_{1E}^2}{2} \alpha_z \\ \text{or } \sqrt{\gamma} &= d_{1E}\sqrt{d_{2H}}\sqrt{\frac{\alpha_z}{2}} \\ &= \frac{d_{1E}\sqrt{d_{2H}}}{2}\sqrt{2\alpha_z} \\ &= \frac{d_{1E}\sqrt{d_{2H}}}{2}\beta_z \, , \end{aligned} \qquad (3.25)$$

where $\beta_z < 1$; also β_z goes to one as z becomes larger, as proved in the following proposition. **Proposition:**

$$\lim_{z \to \infty} \beta_z = 1$$

Proof: α_z is given by

$$\alpha_z = (1 - \delta_k)^2 + \delta_{k-1}^2$$

where δ_k satisfies the conditions:

1. δ_k is a strictly increasing function of k ;
2. δ_k is bounded above by 0.5.

The second property uses the fact that $\alpha_z < 0.5$ (see Appendix), also $\delta_z = \sqrt{\alpha_z/2} < 0.5$. Now assume $\delta_{k+1} < 0.5$, then we have $\delta_k = \sqrt{\alpha_z - (1-\delta_{k+1})^2} < 0.5$; therefore, $\delta_k < 0.5$ for all $k = 1, 2, ..., z$, by induction. The above two facts imply that δ_k converges as $k, z \rightarrow \infty$; moreover,

$$\begin{aligned} \alpha_\infty &= (1-\delta_\infty)^2 + \delta_\infty, \\ \alpha_\infty &= 2\delta_\infty. \end{aligned}$$

Solving these two equations for α_∞ we get

$$\alpha_\infty = 0.5,$$

and the proposition follows.

The last proposition implies that the full error correcting capability of the concatenated code is achieved asymptotically. However, Table 3.1 shows that over 95 % of the full error correcting capability is realized for $z = 4$, with values: $\delta_1 = 0.317, \delta_2 = 0.395, \delta_3 = 0.443, \delta_4 = 0.481$. Thus we only need few branches to get a good error correcting capability.

The following are more examples of thresholds settings:

$z = 2 \rightarrow \delta_1 = 0.35288,\ \delta_2 = 0.45757.$

$z = 3 \rightarrow \delta_1 = 0.32940,\ \delta_2 = 0.41600,\ \delta_3 = 0.47410.$

3.3 Conclusions

The error correcting capability of concatenated codes was evaluated for soft decision decoding. The code symbols can belong to a general alphabet with values in GF(2^m), and could be transmitted over the physical channel using an-

z	α_z	β_z
1	0.3431	0.828
2	0.4187	0.915
3	0.4500	0.948
4	0.4655	0.965
5	0.4750	0.974
6	0.4800	0.979
7	0.4850	0.984
8	0.4877	0.988
9	0.4900	0.990
10	0.4915	0.991
15	0.4957	0.995
30	0.4982	0.998

Table 3.1: Error-Correcting capability for various number of branches z.

tipodal signaling (for m=2) or orthogonal signaling (for $m \geq 2$). We developed a decoding algorithm that maximizes the ability of the concatenated code to correct more errors. The code symbols were assumed to be coherently demodulated. This made the Euclidean distance (from the transmitted codeword) correctable by the concatenated code, an appropriate performance measure of the error correcting capability of the code.

The decoding algorithm proposed uses errors-and-erasures decoding and makes use of several branches with different tentative decisions giving rise to parallel decoding. The set of thresholds for each algorithm is chosen to optimize for the error correcting capability of the code. Moreover, the algorithm can be used without modification for a more general case and with no loss of optimality. For instance, it applies when k_2 symbols of same coordinates, each symbol from a possibly different codeword of the outer code, are inner encoded.

The error correcting capability of the code improves with increasing z. We showed that the full error correcting capability is attained asymptotically with z. However, the numerical results shows that z = 4 gaurantees more than 95 % of this capability.

In case we need to communicate over a bursty channel with long burst lengths, one can add an outer-outer code to pick the correct outer codeword. Then a decoding algorithm is needed to make best use of the resulting concatenated code.

The inner codes need not be block codes. Convolutional or trellis codes for which the Viterbi algorithm algorithm provides a good algorithm for maximum likelihood decoding can be used. We leave such treatment for future research.

CHAPTER IV

PARALLEL DECODING OF CONCATENATED CODES: NONCOHERENT RECEPTION

4.1 Preliminaries and Motivation

In the previous chapter we investigated the error correcting capability of an additive channel appropriate for when the received signals are coherently detected. In this Section we attempt to characterize the error correcting capability for a model appropriate to the case of noncoherent reception. The concept of a distance that characterizes some performance measure is not clear in this case, and for any reasonable distance measure taken the problem is extremely complex to evaluate. Here we specialize the treatment by making additional assumptions on the inner code being a repetition code and develop an appropriate distance measure. The outer code is a bounded distance Reed-Solomon decoder, which is capable of correcting any combination of e errors and τ erasures such that $2e + \tau \leq d_{2H} - 1$.

Consider an M-ary additive channel that models with noncoherent reception of M-ary code symbols transmitted over a continuous additive white Guassian channel using Frequency Shift Keying (FSK) shown in Figure 4.1.

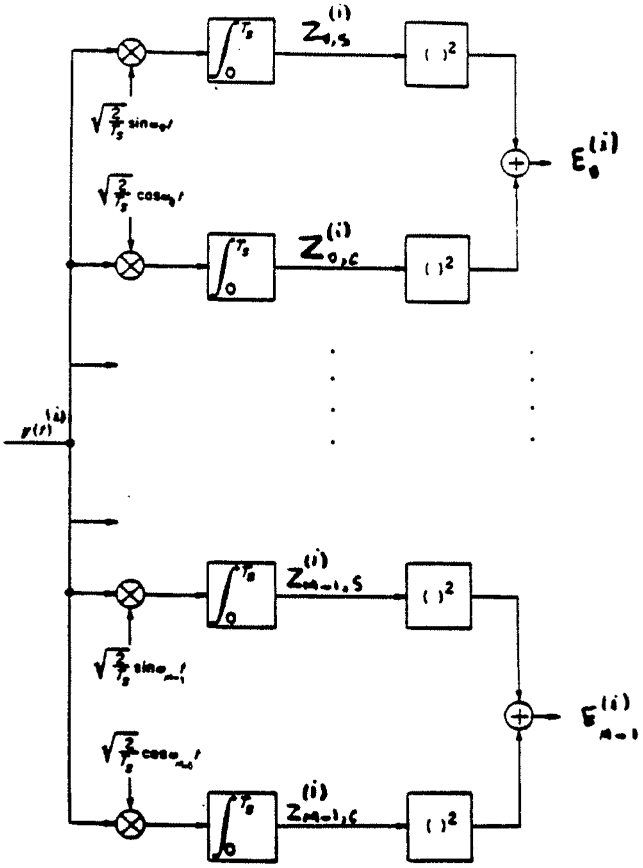
$r(t)^{(i)}$
$\sqrt{\frac{2}{T_s}}\sin\omega_0 t$
$\sqrt{\frac{2}{T_s}}\cos\omega_0 t$
$\sqrt{\frac{2}{T_s}}\sin\omega_{M-1} t$
$\sqrt{\frac{2}{T_s}}\cos\omega_{M-1} t$
$\int_0^{T_s}$
$Z_{0,s}^{(i)}$
$Z_{0,c}^{(i)}$
$Z_{M-1,s}^{(i)}$
$Z_{M-1,c}^{(i)}$
$(\ )^2$
$E_0^{(i)}$
$E_{M-1}^{(i)}$

Figure 4.1: Noncoherent Reception for M-ary Orthogonal FSK Signals.

In what follows we assume that the all zeroes inner codeword is transmitted, and the sum of the outputs of the energy detectors matched to symbol 1 is the largest among the $M-1$ nonzero symbols. In this case the analysis takes into consideration only the outputs of the 0 and 1 energy detectors.

Referring to Figure 4.1, let

$$\begin{aligned} Z_0 &= (Z_{0,c} \;\; Z_{0,s}) \\ Z_1 &= (Z_{1,c} \;\; Z_{1,s}) \end{aligned}$$

where

$$\begin{aligned} Z_{i,c} &= \int_0^T r(t)\cos w_i t \mathrm{dt} \\ Z_{i,s} &= \int_0^T r(t)\sin w_i t \mathrm{dt} \end{aligned}$$

$$i = 1,2 \quad ;$$

$r(t)$ being the received signal consisting of the transmitted signal plus additive noise process $n(t)$, and T is the signal duration.

Given 0 is transmitted we have (assuming the energy per symbol $E_s = 1$)

$$\begin{aligned} Z_{0,c} &= \cos\theta + n_{0,c} \\ Z_{0,s} &= \sin\theta + n_{0,s} \\ Z_{1,c} &= n_{1,c} \\ Z_{1,s} &= n_{1,s} \end{aligned}$$

where

$$n_{m,s} = \sqrt{2}\int_0^T n(t)\sin w_m t \; dt$$

$$\begin{aligned} n_{m,c} &= \sqrt{2}\int_0^T n(t)\cos w_m t \; dt \quad , \\ m &= 0, 1. \end{aligned}$$

Consider a repetition inner code of order L (that is the transmission of L replicas of the same symbol; see Section 5.3) and an outer decoder that corrects errors and erasures. The inner decoder, referred to as the diversity combiner, is taken to be a square law combiner which consists of the sum of squares of the output of a matched filter (see Chapter 5 for more details). Moreover, the output of the square law combiner with maximum value is compared with that of next largest value. If the output of these two energy detectors are "close" the inner decoder decides the corresponding symbol is unreliable and declares an erasure to the outer decoder. Formally, the diversity decoder outputs symbol 0 if and only if

$$\sum_{k=1}^{L}(Z_{0,c}^2(k) + Z_{0,s}^2(k)) \geq \eta \sum_{k=1}^{L}(Z_{1,c}^2(k) + Z_{1,s}^2(k))$$

where η is a threshold chosen to optimize some performance criteria. Assuming C_0 is transmitted, the above equation reduces to

$$L + \mathcal{F}(\underline{n}_0, \underline{\theta}) \geq \eta \parallel n_1 \parallel^2$$

where

$$\begin{aligned} \parallel n_1 \parallel^2 &= \sum_k (n_{1,c}^2(k) + n_{1,s}^2(k)) \\ \mathcal{F}(\underline{n}_0, \underline{\theta}) &= \sum_k (n_{0,c}^2(k) + n_{0,s}^2(k)) \\ &\quad + 2\sum_k (\cos\theta(k) n_{0,c}(k) + \sin\theta(k) n_{0,s}(k)) \\ &= \sum_k (n_{0,c}^2(k) + n_{0,s}^2(k)) \\ &\quad + 2\sqrt{n_{0,c}^2(k) + n_{0,s}^2(k)}\,\beta(k) \quad . \end{aligned}$$

In the above equation

$$\begin{aligned}\beta(k) &= \cos(\phi(k)+\theta(k)) \\ \phi(k) &= \arccos\frac{n_{0,c}(k)}{\sqrt{n_{0,c}^2(k)+n_{0,s}^2(k)}}.\end{aligned}$$

The inner decoder output is an erasure if and only if

$$\frac{1}{\eta} \parallel n_1 \parallel^2 < L + \mathcal{F}(\underline{n}_0, \underline{\theta}) < \eta \parallel n_1 \parallel^2,$$

and an error will occur when

$$L + \mathcal{F}(\underline{n}_0, \underline{\theta}) \leq \eta \parallel n_1 \parallel^2 .$$

We would like to choose η such that the concatenated code has its maximum error correcting capability for that single branch. Ideally we would like to use $\parallel n_1 \parallel^2 + \parallel n_2 \parallel^2$ correctable by the code, as the error correcting capability of the decoder. However, this performance measure is very hard to analyze, and we use a slightly different, but appropriate, performance measure. Before describing the performance measure of interest, define the following.

Let $x_m \triangleq$ number of rows such that

$$\parallel n_1 \parallel^2 = m \quad ,$$

then

$$\gamma = \int_0^\infty m x_m dN(m);$$

where $N(m)$ is a counting measure with $N(m) = 1$ if $x_m > 0$ and $N(m) = 0$, otherwise. Hence, γ is the total noise added to the energy detector with next largest value. Then choose $\eta = \eta^*$ to maximize γ, i.e.

$$\eta^* = \arg\max_{\eta} \gamma$$

where the maximum is over all η such that the decoder outputs the correct codeword.

If we have more than one decoder branch with different thresholds, it is reasonable to expect that with optimal settings of these thresholds the parallel decoder will perform better than the single branch decoder, as is the case for the coherent reception. In this Section we consider such a parallel decoder scheme. We start by formulating the problem and proceed like we did in Chapter 3.

4.2 Problem Formulation

In the following analysis we rely on some results from Section 3.5. We consider a parallel decoding scheme with z branches. Branch i is characterized by threshold η_i used by the inner decoder to declare erasures for the outer decoder. Thus the decoder is determined by the set of thresholds:

$$H_z = \{\eta_1, \ldots, \eta_z\}$$

where $\eta_1 < \eta_2 < \ldots < \eta_z$. The set H_z is to be chosen to maximize the error correcting capability of the code.

Also, by convention we let

$$\eta_0 = \frac{1}{\eta_1} \; , \quad \eta_{z+1} = \infty \quad .$$

Let $\mathcal{L} \triangleq L + \mathcal{F}(\underline{n_0}, \underline{\theta})$. Then the inner decoder outputs an error if and only if

$\mathcal{L} < \frac{1}{\eta_k} \parallel n_1 \parallel^2$, and it outputs an erasure if and only if

$$\frac{1}{\eta_k} \parallel n_1 \parallel^2 < \mathcal{L} < \eta_k \parallel n_1 \parallel^2 .$$

Therefore, we can write an exact expression for the number of errors e_k and number of erasures τ_k, for the k-th branch, at the input of the outer decoder. These are

$$e_k = \int_{\eta_k \mathcal{L}}^{\infty} x_m dN(m) \tag{4.1}$$

and

$$\tau_k = \int_{\frac{\mathcal{L}}{\eta_k}}^{\eta_k \mathcal{L}} x_m dN(m) \quad , \quad k = 1, 2, \ldots z \quad . \tag{4.2}$$

To find recursive relations for e_k and τ_k, notice that we can rewrite (4.1) and (4.2) as follows

$$\begin{aligned}
e_k &= \int_{\eta_k \mathcal{L}}^{\infty} x_m dN(m) \\
&= \int_{\eta_k \mathcal{L}}^{\eta_{k+1} \mathcal{L}} x_m dN(m) + \int_{\eta_{k+1} \mathcal{L}}^{\infty} x_m dN(m) \\
&= \int_{\eta_k \mathcal{L}}^{\eta_{k+1} \mathcal{L}} x_m dN(m) + e_{k+1} \quad , \quad k = 1, \ldots, z-1 \\
e_z &= \int_{\eta_z \mathcal{L}}^{\infty} x_m dN(m) \quad .
\end{aligned}$$

Similarly,

$$\begin{aligned}
\tau_k &= \int_{\frac{\mathcal{L}}{\eta_k}}^{\eta_k \mathcal{L}} x_m dN(m) \\
&= \int_{\frac{\mathcal{L}}{\eta_k}}^{\frac{\mathcal{L}}{\eta_{k-1}}} x_m dN(m) + \int_{\frac{\mathcal{L}}{\eta_{k-1}}}^{\eta_{k-1} \mathcal{L}} x_m dN(m) + \int_{\eta_{k-1}^{\mathcal{L}}}^{\eta_k \mathcal{L}} x_m dN(m) \\
&= \int_{\frac{\mathcal{L}}{\eta_k}}^{\frac{\mathcal{L}}{\eta_{k-1}}} x_m dN(m) + \tau_{k-1} + e_{k-1} - e_k \quad , \qquad k = 2, \ldots, z \\
\tau_1 &= \int_{\frac{\mathcal{L}}{\eta_1}}^{\eta_1 \mathcal{L}} x_m dN(m) \quad .
\end{aligned}$$

From the above relations we note that

$$\tau_k \leq \tau_{k+1} \quad , \quad e_k \geq e_{k+1}$$

which is reasonable, since for a larger value of threshold we expect a larger number of erasures and therefore, less errors.

Now we find an expression for the minimum channel loss such that the number of errors and erasures in the branches are given by the vectors $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_z)$ and $\boldsymbol{e} = (e_1, \ldots, e_z)$, respectively. We make use again of the following property:

$$\min \int_{j_1}^{j_2} m x_m dN(m) = a j_1$$

where the minimum is over $\underline{x}$ such that $\int_{j_1}^{j_2} x_m dN(m) = a$. The equality holds if and only if

$$x_m = a \;\text{ for }\; m = j_1, \;\; x_m = 0 \;\text{ for }\; j_1 < m \leq j_2 \;\; .$$

For a given $H_z, \boldsymbol{e}$, and $\boldsymbol{\tau}$ we denote γ by $\gamma(H_z, \boldsymbol{\tau}, \boldsymbol{e})$ which is rewritten as

$$\begin{aligned}
\gamma(H_z, \tau, e) &= \min_{\underline{x}} \int_0^\infty m x_m dN(m) \\
&= \min_{\underline{x}} \left\{ \int_0^{\frac{1}{\eta_z}\mathcal{L}} m x_m dN(m) + \int_{\frac{\mathcal{L}}{\eta_z}}^{\frac{1}{\eta_{z-1}}\mathcal{L}} m x_m dN(m) + \ldots \right. \\
&\quad + \int_{\frac{\mathcal{L}}{\eta_2}}^{\frac{\mathcal{L}}{\eta_1}} m x_m dN(m) + \int_{\frac{\mathcal{L}}{\eta_1}}^{\eta_1 \mathcal{L}} m x_m dN(m) + \int_{\eta_1 \mathcal{L}}^{\eta_2 \mathcal{L}} m x_m dN(m) \\
&\quad + \ldots + \int_{\eta_{z-1}\mathcal{L}}^{z\mathcal{L}} m x_m dN(m) + \int_{\eta_z \mathcal{L}}^{\infty} m x_m dN(m) \\
&= 0 + \frac{\mathcal{L}}{\eta_z}(\tau_z - \tau_{z-1} - e_{z-1} + e_z) + \frac{\mathcal{L}}{\eta_{z-1}}(\tau_{z-1} - \tau_{z-2} - e_{z-2} + e_{z-1}) \\
&\quad + \ldots + \frac{\mathcal{L}}{\eta_2}(\tau_2 - \tau_1 - e_1 + e_2) + \tau_1 \frac{\mathcal{L}}{\eta_1} \\
&\quad + \mathcal{L}\eta_1(e_1 - e_2) + \mathcal{L}\eta_2(e_2 - e_3) + \ldots + \mathcal{L}\eta_{z-1}(e_{z-1} - e_z) \\
&\quad + \mathcal{L}\eta_z e_z
\end{aligned}$$

$$\begin{aligned} &= \mathcal{L} \sum_{k=2}^{z} \frac{1}{\eta_k}(\tau_k - \tau_{k-1} - e_{k-1} + e_k) + \frac{\mathcal{L}}{\eta_1}\,\tau_1 \\ &\quad + \mathcal{L} \sum_{k=1}^{z-1} \eta_k (e_k - e_{k+1}) + \mathcal{L}\eta_z e_z \end{aligned}$$

where the minimum above is over all noise strategy such that the decoder outputs are incorrect. By rearranging terms (see Appendix D) we get

$$\begin{aligned} \gamma(H_z, \tau, e) &= \mathcal{L} \sum_{k=1}^{z} \left\{ \left(\frac{1}{\eta_k} - \frac{1}{\eta_{k+1}} \right) \tau_k + \left(\frac{1}{\eta_k} - \frac{1}{\eta_{k+1}} + \eta_k - \eta_{k-1} \right) e_k \right\} \\ &\equiv \sum_{k=1}^{z} \gamma^{(k)}(\tau_k, e_k) \quad . \end{aligned} \tag{4.3}$$

As in the coherent case, minimizing over e is simple

$$\begin{aligned} \frac{1}{\mathcal{L}} \gamma(H_z, \tau) &= \min_{\substack{e: 2e_k + \tau_k \geq d_{2H} \\ k=1,\ldots,z}} \sum_{k=1}^{z} \gamma^{(k)}(\tau_k, e_k) \\ &= \sum_{k=1}^{z} \min_{e_k : 2e_k + \tau_k \geq d_{2H}} \gamma^{(k)}(\tau_k, e_k) \\ &\equiv \sum_{k=1}^{z} \gamma^{(k)}(\tau_k) \quad , \end{aligned}$$

where

$$\gamma^{(k)}(\tau_k) = \begin{cases} \left(\frac{1}{\eta_k} - \frac{1}{\eta_{k+1}} \right) \tau_k + \left(\frac{1}{\eta_k} - \frac{1}{\eta_{k+1}} + \eta_k - \eta_{k-1} \right) \lfloor \frac{d_{2H} - \tau_k - 1}{2} \rfloor \quad , \\ \qquad\qquad\qquad\qquad \text{if} \quad 0 \leq \tau_k \leq d_{2H} \\ \left(\frac{1}{\eta_k} - \frac{1}{\eta_{k+1}} \right) \tau_k \quad , \quad \text{if} \quad \tau_k \geq d_{2H}. \end{cases}$$

Then,

$$\begin{aligned} \frac{1}{\mathcal{L}}\, \gamma(H_z) &= \min_{\tau : \tau_k \leq \tau_{k+1}} \gamma(H_z, \tau) \\ &= \min_{\tau_z \leq d_2} \left\{ \gamma^{(z)}(\tau_z) + \min_{\tau_{z-1} \leq \tau_z} \left\{ \gamma^{(z-1)}(\tau_{z-1}) \right.\right. \\ &\quad \left.\left. + \quad \ldots \quad + \min_{\tau_1 \leq \tau_2} \gamma^{(1)}(\tau_1) \right\} \right\} \ldots \Big\} \quad . \end{aligned}$$

According to Lemma, the minimum occurs when

$$\tau_1 \equiv \tau_2 \equiv \ldots \equiv d_{2H} \quad .$$

This implies

$$\begin{aligned}\frac{1}{\mathcal{L}}\,\gamma(H_z) &= \sum_{k=1}^{z}\frac{d_{2H}}{2}\left(\frac{1}{\eta_k}-\frac{1}{\eta_{k+1}}+\eta_k-\eta_{k-1}\right)+ \\ &\quad \min_{\tau^*}\sum_{k=1}^{z}\frac{\tau_k^*}{2}\left[2\left(\frac{1}{\eta_k}-\frac{1}{\eta_{k+1}}\right)-\left(\frac{1}{\eta_k}-\frac{1}{\eta_{k+1}}+\eta_k-\eta_{k-1}\right)\right] \\ &= \eta_z\frac{d_{2H}}{2}+\min_{\tau^*}\sum_{k=1}^{z}\frac{\tau_k^*}{2}\left(\frac{1}{\eta_k}-\frac{1}{\eta_{k+1}}-\eta_k+\eta_{k-1}\right) \quad .\end{aligned}$$

Two cases are considered

<u>Case 1:</u> $d_{2H}\equiv 0 \mod 2$; this implies that

$$\tau\in\{(0,0,\ldots,0)\ ,\ (0,\ldots,0,d_{zH}),\ldots,(d_{2H},\ldots,d_{2H})\} \quad .$$

In this case

$$\begin{aligned}\frac{1}{\mathcal{L}}\gamma(H_z) &= \frac{d_{2H}}{2}\,\eta_z+\min_k\sum_{m=1}^{z}\frac{d_{2H}}{2}\left(\frac{1}{\eta_m}-\frac{1}{\eta_{m+1}}-\eta_m+\eta_{m-1}\right) \\ &= \frac{d_{2H}}{2}\eta_z+\min_k\frac{d_{2H}}{2}\left(\frac{1}{\eta_k}+\eta_{k-1}-\eta_z\right) \\ &= \frac{d_{2H}}{2}\min_{1\le k\le z+1}\left(\frac{1}{\eta_k}+\eta_{k-1}\right) \quad .\end{aligned}$$

<u>Case 2:</u> $d_{2H}\equiv 1 \mod 2$; which implies that

$$\tau\in\{(1,1,\ldots,1),(1,\ldots,1,d_{2H}),\ldots,(d_{2H},\ldots,d_{2H})\} \quad .$$

Therefore

$$\begin{aligned}\frac{1}{\mathcal{L}}\gamma(H_z) &= \frac{d_{2H}}{2}\,\eta_z+\min_{k\in\{1,2,\ldots,z+1\}}\left\{\sum_{m=1}^{k-1}\frac{1}{2}\left(\frac{1}{\eta_k}-\frac{1}{\eta_{k+1}}-\eta_k+\eta_{k-1}+\right)\right. \\ &\quad \left.+\sum_{m=k}^{z}\frac{d_{2H}}{2}\left(\frac{1}{\eta_k}-\frac{1}{\eta_{k+1}}-\eta_k+\eta_{k-1}\right)\right\} \\ &= \frac{1}{\eta_1}+\frac{d_{2H}-1}{2}\min_{1\le k\le z+1}\left(\frac{1}{\eta_k}+\eta_{k-1}\right)\end{aligned}$$

The optimal decoder strategy maximizes the channel loss $\gamma(H_z)$ for a fixed number of branches z. For d_{2H} even, it is the solution of the game,

$$\begin{aligned}\gamma &= \max_{H_z} \min_{1\leq k\leq z+1} \gamma(H_z) \\ &= \frac{d_{2H}\mathcal{L}}{2} \max_{H_z} \min_{1\leq k\leq z+1} \left(\frac{1}{\eta_k} + \eta_{k-1}\right);\end{aligned}$$

for d_{2H} odd

$$\gamma = \max_{H_z} \left\{\frac{1}{\eta_1} + \frac{d_{2H}-1}{2} \min_{1\leq k\leq z+1} \left(\frac{1}{\eta_k} + \eta_{k-1}\right)\right\} \mathcal{L} \ .$$

The solution to the above game follows exactly the steps we used for the coherent case, namely

$$\max_{H_z} \min_{1\leq k\leq z+1} \left(\frac{1}{\eta_k} + \eta_{k-1}\right)$$

is obtained when

$$\frac{1}{\eta_k} + \eta_{k-1} = \alpha_z \quad , \quad k = 1, 2, \ldots z, z+1 \ . \tag{4.4}$$

Moreover, $H_z^* = \arg\max_{H_z} \gamma(H_z)$ is a unique decoder strategy and there is only one constant α_z such that (4.3) holds. Similar arguments hold for d_{2H} odd.

Property:

$$1 < \alpha_z < 2 \quad , \quad \text{forall } z$$

$$\lim_{z\to\infty} \alpha_z = 2$$

Proof:

$$\text{Since } \eta_k > 1 \quad , \quad k = 1, \ldots, z \Rightarrow$$

$$\alpha_z = \tfrac{1}{\eta_k} + \eta_{k-1} > 1 \ .$$

z	α_z	H_z
1	$\sqrt{2}$	$\eta_1 = \sqrt{2}$
2	1.7316	$\eta_1 = 1.155 \quad \eta_2 = 1.7316$
3	1.8500	$\eta_1 = 1.0824, \eta_2 = 1.3066, \eta_3 = 1.8500$

Table 4.1: Error-Correcting capability for various number of branches z (noncoherent case).

Also, for $k = 1$ we have

$$\frac{1}{\eta_1} + \eta_0 = \frac{2}{\eta_1} = \alpha \Rightarrow$$

$$\eta_1 = \frac{2}{\alpha} \;;$$

but $\eta_1 > 1$ which implies that $\frac{2}{\alpha} > 1$ which in tern implies that $\alpha < 2$.

To show the second property, notice that

$$1 < \eta_k < \eta_{k+1} < \ldots < \eta_z,$$

and η_k is upper bounded by 2. This implies η_k converges as $k, z \to \infty$ to η_∞. In this case we have,

$$\begin{aligned} \frac{1}{\eta_\infty} + \eta_\infty &= \alpha_\infty, \\ \eta_\infty &= \frac{2}{\alpha_\infty}. \end{aligned}$$

Solving these two equations we get $\alpha_\infty = 2$. Several values are tabulated below.

4.3 Conclusions

In this Chapter we evaluated the error correcting capability for a particular concatenated code: the inner code being a repetition code, and the outer code is a Reed-Solomon code. The code symbols were transmitted through the channel using orthogonal signaling and were noncoherently detected by the receiver. We proposed a parallel decoding algorithm when the diversity symbols are combined by a soft decision decoder, mainly square-law combining. This algorithm made use of parallel decoding with different branches characterized by different thresholds, the later chosen to maximize a certain performance measure which reflects the error correcting capability of the concatenated code.

The error correcting capability $\gamma = \gamma(\underline{n_0}, \underline{\theta})$ is random variable. It would be of more interest to find the expected value of γ with respect to the random vectors $\underline{n_0}$ and $\underline{\theta}$, which depend on the channel statistics.

Due to noncoherent demodulation of the code symbols, the Euclidean distance measure used in the previous Chapter is not an appropriate performance criteria for error correcting capability. We use a criteria for the Viterbi Ratio Threshold decoder. The reason for restricting the inner code to be a repetition code is the difficulty in defining an appropriate performance measure for the error correcting capability of general code used in the noncoherent communication system. This remains an open problem.

CHAPTER V

PERFORMANCE OF A FHSS SYSTEM IN RAYLEIGH FADING WITH NOISY SIDE INFORMATION

5.1 Introduction

The engineering importance of communication media that exhibit fading has increased markedly in recent years. Common examples of channels where fading is encountered in practice are the ionospheric high frequency (HF) channel and the tropospheric scatter channel [23]-[28]. For instance when using cellular mobile radio communications [1] and indoor radio communications [27] the reception usually suffers from severe multipath fading. In this chapter we are interested in problems which arise when considering slow-frequency hopped (SFH) spread spectrum communications systems over selective fading channels [13]-[16], because in many applications of SFH systems (such as in SFH multiple access communications) the channel cannot be adequately modeled as a non-dispersive additive white Gaussian noise channel. Moreover, frequency hopping spread spectrum modulation is an effective way to combat fading. In slow frequency-hopped spread spectrum modulation the hopping rate is smaller than the data rate. During transmission the spread spectrum signal encounters on such channels severe fading (i.e. reduced signal strength) and may produce intersymbol-interference or other dispersive effects.

For most fading channels it is impractical to obtain accurate phase estimates and incorporate these estimates in the detection process; thus the random character of the fading channel prohibits the use of coherent demodulation. Therefore, noncoherent demodulation is considered for faded channels. Binary differential phase-shift-keying (DPSK) and frequency-shift keying (FSK) are of particular interest for applications of SFH systems in selective fading channels [23], since these forms do not require the receiver to establish phase coherence at the beginning of each hop. The form of modulation considered in this Chapter is orthogonal BFSK.

As a result of fading, reliable communication over such channels requires a large bit energy to noise ratio $\frac{E_b}{N_0}$. It is known [37] that when communicating over a fading channel the uncoded bit error rate (BER) decreases linearly rather than exponentially with $\frac{E_b}{N_0}$. As a result, to achieve an error probability of 10^{-5}, which requires only 13.4 dB for a noncoherent channel with no fading and when using binary orthogonal signaling, requires approximately 50 dB for a fading channel [34]. Also, the loss in capacity and cutoff rate of the channel due to Rician fading was investigated by Stark in [31].

To compensate for such a tremendous loss, most communication systems use some forms of error-correction coding. For fading channels we recover most of the loss incurred from fading with the use of diversity (repetition coding) with optimum rate. For example with repetition coding we need 22.4 dB as compared to 50 dB to achieve error probability of 10^{-5}. The use of further smart coding techniques reduces the loss in performance to approximately 5 dB [34], assuming codes of equal complexity and equal error probability requirements.

In a SFH communication system more than one data symbol is transmitted per dwell interval. If the system uses some form of coding it is desirable to obtain information concerning the reliability of the symbols in a particular dwell interval which enables us to erase unreliable symbols.

As mentioned in Chapter 1, postdetection is one way to generate side information. Included in the postdetection methods is the technique used in this chapter and which is described as follows. We include in each transmitted hop a known sequence of symbols called test symbols. The number of such symbols that are received correctly during a given dwell interval is used as a statistic upon which to base an estimate of the reliability of the data symbols in that dwell interval. This method was suggested by McEliece and Stark in [21], and it was used by Pursley in [25] for a frequency-hopped multiple access channel to detect the presence of a hit in a given dwell interval. The estimate regarding the reliability of the symbols in a given dwell interval can be made arbitrarily reliable by increasing the number of symbols in the known "test" sequence at the expense of reduced data rate. The implementation of such a side information generation is simple; however, this side information is noisy in the sense that correct data symbols will occasionally be classified as unreliable or incorrect data symbols as reliable.

In Chapter 6 we develop another method in Chapter 6 for generating side information about a received hop. This method is based on concatenated coding. The idea is to introduce redundancy (i.e., parity check) symbols in each dwell interval by inner encoding the interleaved outer-code symbols. The inner-code is used for error correction and error detection to judge whether a received hop is reliable; if the received inner-code lies in the detection region, then it is claimed unreliable and the entire hop is erased.

The goal of this chapter is to investigate the performance of a coded slow-frequency-hopped spread spectrum communication system, when using the above form of side information, in the presence of Rayleigh fading and additive white Guassion noise. The system considered assumes noncoherent reception with orthogonal BFSK as a signaling scheme. We also assume that the hopping patterns are sequences of independent random variables each of which is uniformly distributed on the set of q available frequencies, and assume the channel is memoryless from hop to hop; that is, fading is independant from hop to hop. Full interleaving is employed such that no two code symbols from the same codeword will be transmitted in the same dwell interval. This will ensure that errors and erasures within a codeword are independent.

We give a careful introduction on the system model. Section 3 investigates the performance of the system when repetition coding is employed with hard decision combining and soft decision combining. The performance when using Reed-Solomon codes is analyzed in Section 4 in which we consider two configurations for decoding. The first configuration consists of a single Reed-Solomon decoder that corrects errors and erasures. The second configuration analyzed is a parallel decoder scheme with two Reed-Solomon decoders: the first decoder corrects errors and erasures and the second one is used only as an error correcting decoder.

We show how using this technique for generating side information, significantly improves the performance of Reed-Solomon coded SFH spread spectrum communications in a fading channel. The performance measure taken is the probability of symbol error. When symbol error probablity of 10^{-4} is desired, the improvement in symbol signal-to-noise ratio required to achieve this performance is approximately 2 dB. This is done by transmitting only 3 test bits in each dwell interval and using a (32,5) Reed-Solomon code.

5.2 System Model

As mentioned earlier, we consider orthogonal BFSK signaling; the modulated signal is frequency hopped to produce the transmitted signal. The received signal is dehopped then noncoherently demodulated to produce the channel output. The received signal is corrupted by multiplicative noise with Rayleigh statistics, which is referred to as fading. We treat the frequency hopper and dehopper as performing inverse operations on the modulated signal. Below we elaborate more on the system model used throughout this chapter and Chapter 6.

The channel considered is assumed to be a slow Rayleigh faded channel. This means that fading is assumed to be slow enough that the amplitude of the faded signal is nearly constant over the duration of a frequency hop. This is usually referred to as a uniform channel. In frequency hopped spread spectrum communications the channel spectrum utilized is divided into frequency slots where the communicator uses one slot during a time interval called the dwell interval during which J symbols are transmitted. The communicator then uses another frequency slot during the next dwell interval to transmit the next J symbols, and so on. The frequency slots are chosen during each dwell interval in a pseudorandom fashion. Moreover, it is assumed that the fading at different frequency slots are independent.

For a more formalized treatment of the frequency hopped spread spectrum system we use the model adopted by Stark in [32]. Consider frequency hopping with J symbols transmitted in each dwell interval. Assume that the particular hop begins at time $t = 0$ and is of duration JT where T is the duration of one BFSK signal. When the input to the channel is X_i, $i = 0, 1$, the data modulated signal $s_i'(t-jT)$ is the input to the frequency hopper during the interval $jT \leq t < (j+1)T$. The signal $s_i'(t)$ is given by

$$s_i'(t) = \sqrt{2P}\cos(\omega_i t + \theta_i)p_T(t) \ , \quad (j-1)T \leq t < jT \tag{5.1}$$

where ω_i, $i = 0, 1$, is the radian frequency of the signal, P its power, and θ_i the phase of the i-th signal. Also, $p_T(t)$ is a basic pulse shaping function usually designed to reduce intersymbol interference (ISI) and used for spectrum shaping. Since this is beyond the scope of this chapter, $p_T(t)$ is taken to be

$$p_T(t) = \begin{cases} 1 \ , & 0 \leq t < T \\ 0 \ , & \text{otherwise.} \end{cases}$$

The frequency hopper changes the center frequency of the modulated signal in different hops to one of q different center frequencies according to a specified hopping pattern to produce the transmitted signal $s_i(t - jT)$. Let T_h be the hop dwell interval. The transmitted signal for the j-th symbol of the l-th hop during the interval $jT + lT_h \leq t < (j+1)T + lT_h$, $0 \leq j < J$, given by

$$s_i(t) = \sqrt{2P}p_T(t)cos((\omega_l^{(H)} + \omega_i)t + \phi_i) \tag{5.2}$$

where $\{\omega_l^{(H)} : -\infty < l < \infty\}$ is a sequence of independent and uniformly distributed random variables that take values from a set of q available frequencies. The received signal $y(t)$ when $s_i(t - jT)$, $0 \leq j < J$, is transmitted consists of the transmitted signal with random amplitude and additive white Gaussian noise:

$$y(t) = R_l \ s_i(t - jT) + n(t), \quad jT + lT_h \leq t < (j+1)T + lT_h \ , \tag{5.3}$$

where $\{R_l : -\infty < l < \infty\}$ is a sequence of independent and identically Rayleigh distributed random variables; i.e. with probability density function

$$p_{R_l}(r) = \begin{cases} 2re^{-r^2} \ , & r > 0 \\ 0 \ , & \text{otherwise.} \end{cases} \tag{5.4}$$

We have normalized $\mathrm{E}(R^2)$ to be equal to 1 so that the average received signal energy is $\overline{E_b}=\mathrm{E}(R^2)E_b = E_b$ and is the mean bit energy, E_b being the unfaded signal energy per bit at the receiver. Also $n(t)$ is additive white Gaussian noise with double sided power spectral density $\frac{N_0}{2}$.

The frequency dehopper changes the center frequency of the received signal according to the hopping pattern of the transmitter. The signal $y'(t)$ at the output of the frequency dehopper is then given by

$$y'(t) = R_l\ \sqrt{2P}cos(\omega_i t + \phi_i)p_T(t) + n'(t), \quad (j-1)T + lT_h \leq t < jT + lT_h, \tag{5.5}$$

where $n'(t)$ is the term due to the additive white Gaussian noise, which is also additive and white Gaussian. In (5.5) ϕ_i is a random phase which accounts for the phase introduced by the frequency hopper, dehopper, transmission delays, and more seriously random phase due to fading, since degradation due to uncertainty in phase is more severe than that of amplitude. The received signals are first demodulated by a noncoherent matched filter followed by an envelope detector; this is equivalent to the square root of the sum of the squares of the outputs of an inphase-quadrature (I-Q) square-law demodulator. Equivalent block diagrams for the demodulator are shown in Figures 5.1 and 5.2.

For our case (BFSK) the detector consists of two matched filters-envelope detectors (matched to $s_0(t)$ and $s_1(t)$, respectively), or equivalently inphase and quadrature correlators whose two outputs are squared and summed. Then a comparator

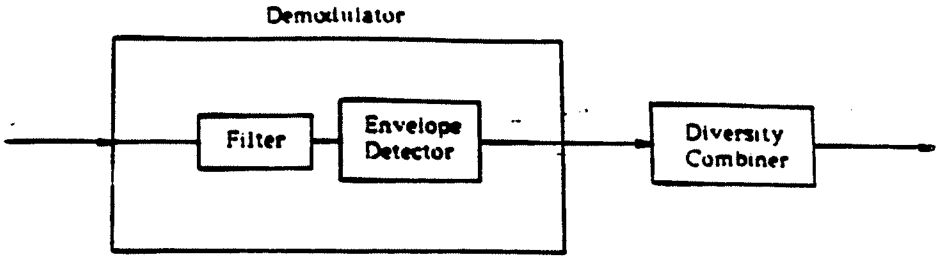

Figure 5.1: Block diagram of a noncoherent receiver.

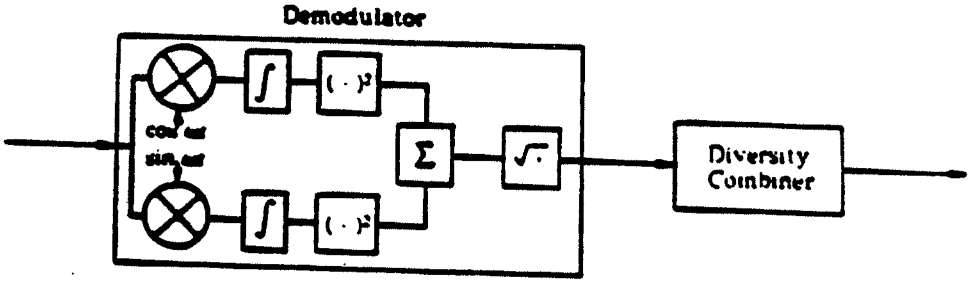

Figure 5.2: Equivalent block diagram of a noncoherent receiver.

determines the largest detector output and decides in favor of the corresponding signal.

Without any form of coding, the bit error rate $p_b(r)$ conditioned on the fade r is given by

$$p_b(r) = \frac{1}{2}e^{-\frac{r^2 E_b}{2N_0}}.$$

Averaging the above expression with respect to the probability density function in (5.4) we get

$$p_b = \frac{1}{2 + \frac{E_b}{N_0}}$$

which shows the detrimental effect of Rayleigh fading, when compared to the exponential dependence of p_b on E_b/N_0 for the additive Gaussian channel.

A discrete-time channel model that incorporates the above SFH system is as follows. The data stream is divided into blocks of length J in which each block being transmmitted in one hop. Since fading is independent from hop to hop we can view the channel considered as a block interference, which is a class of channel models with memory investigated in [21]. In this case the channel is characterized by occasional severe error bursts of constant length J for a time period equal to a dwell interval. The block interference channel we consider does not assume that the decoder knows the state of the channel (i.e. the severity of the fading) for any hop. Consider the channel from the input of the modulator (with inputs 0 or 1) to the output of the demodulator/quantizer (with output alphabet $\{0, 1\}$). For a given fade level $R = r$ the channel is a binary symmetric channel with crossover probability $s = p_b(r)$. Let $S = p_b(R)$ be a random variable that depends on the fading random variable R. Then for orthogonal BFSK signaling the probability distribution of S on $[0, \frac{1}{2}]$ has a density function given by

$$p(s) = \beta 2^{\beta} s^{\beta - 1} \qquad 0 \leq s < \frac{1}{2},$$

where $\beta = 2/\frac{E}{N_0}$. The channel from the output of the frequency hopper to the input of the frequency dehopper can then be modelled as a 2^J-ary DMC channel with input and output alphabets $\{0,1\}^J$. The probability that $\boldsymbol{y}=(y_1, ..., y_J)$ is received, given that $\boldsymbol{x}=(x_1, ..., x_J)$ is transmitted, is given by

$$p(\boldsymbol{y} \mid \boldsymbol{x}) = \int_0^{\frac{1}{2}} s^d (1-s)^{J-d} p(s) \mathrm{d}s$$

where d is the Hamming distance between $\boldsymbol{x}$ and $\boldsymbol{y}$.

The capacity for this type of channels with no side information was shown to be monotonically increasing in J, and as J tends to infinity, the capacity approaches that with perfect side information about the state of the channel available at the receiver. Also, it was argued that interleaving when side information is absent degrades the theoretical performance (where performance is the *information theoretic capacity*). Since we interleave the code symbols we are trading in capacity. However, if J is large enough, the receiver can make a reliable statistical estimate of the noise severity even if the channel does not provide side information. If this is done, interleaving is close to the optimal coding strategy. This was the primary motivation for using a test pattern in each frequency hop.

Now consider the first method for generating the side information, that of transmitting a known test pattern of bits in each dwell interval. A second system that is based on concatenated coding will be presented separately in Chapter 6. For the former case, three types of symbols are transmitted in each dwell interval: information symbols, redundant symbols, and Λ known test symbols. Collectively, the information symbols and the redundant symbols are referred to as the data symbols and they are from an alphabet of size M. Each information symbol conveys $\log_2 M$ bits of information. In our application M is a power of 2 (say $M = 2^m$) and a data symbol is represented as a sequence of m binary symbols. The test symbols are assumed to be binary symbols. The Λ test bits provide the decoder with information about the channel, and the decoder uses this information to improve the correcting capability of the code. A block diagram of the communication system considered is shown in Figure 5.3.

To decide about which symbol to erase, the receiver performs the following additional function. Hard decisions are made on the Λ test bits. If λ or more of these are in error, for some threshold λ, then the receiver labels all the other symbols transmitted during that hop as "bad" and erases all the symbols in that hop. If fewer than λ are in error then the receiver labels all other bits transmitted during that hop as "good" and delivers the corresponding estimates to the decoder.

The measure of performance considered is the probability of symbol error, and we choose the threshold λ which will minimize this probability.

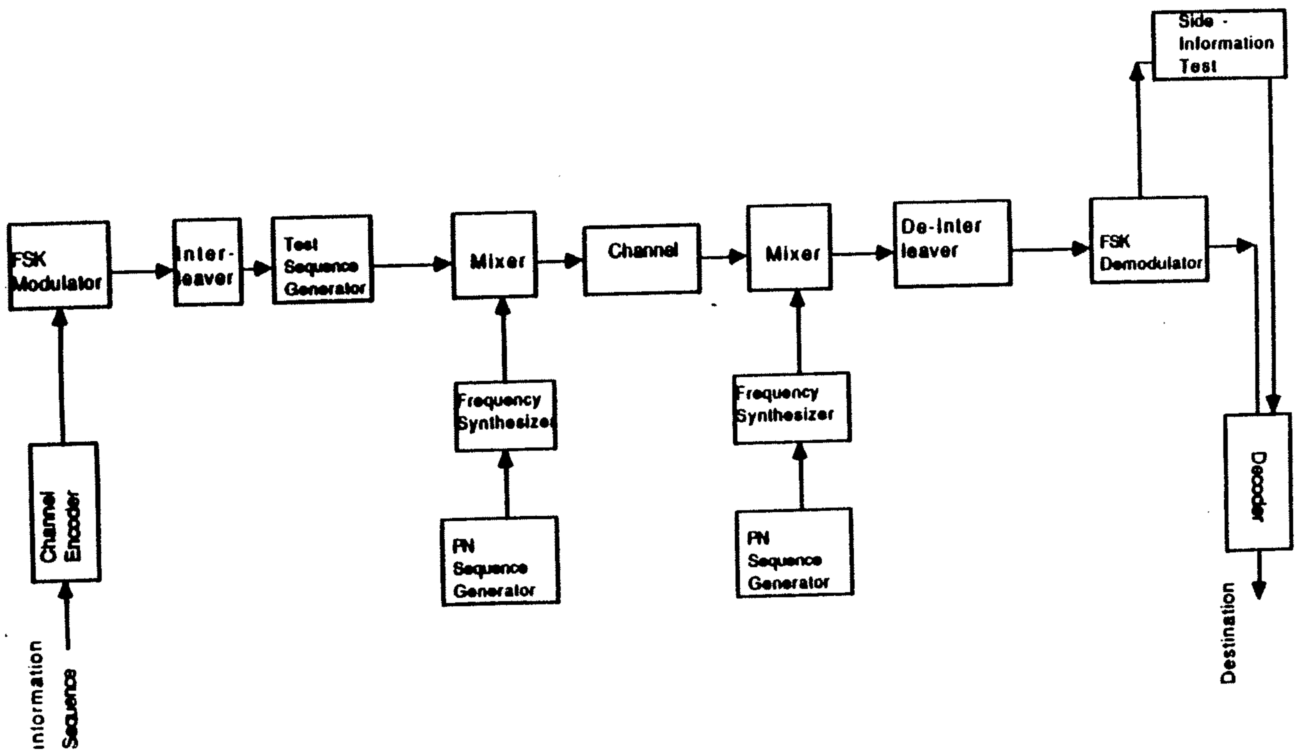

Figure 5.3: Block diagram of a frequency-hopped spread spectrum system with side information.

5.3 Performance of a Repetition Coded System

5.3.1 Hard Decision Decoding

The simplest type of block codes allowing a variable amount of redundancy is the repetition code; this approach is often called diversity transmission. With this code a single information symbol is encoded into a block of L identical symbols producing an $(L, 1)$ code. In this section we evaluate the average probability of symbol error for the $(L, 1)$ repetition code on an M-ary symmetric errors-and-erasures channel. Decoding for such a channel is referred to (in Chapter 1) as hard decision decoding. Diversity transmission is often employed to provide reliable communication in the presence of fading or other forms of interference such as partial-band Gaussian interference. Here we first assume $m = 1$ which means that we have a binary-symmetric-erasure channel as seen by the encoder decoder pair. Because of interleaving, each code symbol is transmitted on a separate hop. As mentioned earlier, to ensure statistical independence among successive code symbols the fading is assumed to be frequency selective. If fading is time selective, independence is provided using time diversity which requires interleaving the diversity symbols so that any two symbols are separated in transmission time by an amount which is longer than the inverse bandwidth of the fading phenomena.

The performance of the diversity transmission system depends on the way in which the received code symbols, called diversity receptions, are combined. The diversity combiner considered first is a hard decision decoder which works as follows. If at least one of the diversity transmissions is on a hop that is declared "good" by the receiver then the diversity decoder is a majority logic decoder which combines the "good" diversity receptions. That is, among the good diversity receptions, the diversity decoder counts the number of times each symbol is received and chooses the one that have the largest count as the transmitted symbol. If all the diversity receptions are declared "bad" by the receiver, then the majority logic decoder combines all the "bad" symbols.

Our goal is to calculate the symbol error probability when using the above described side information. Also for $\Lambda > 1$, we are to choose the threshold λ which minimizes this error probability. To do this calculate the performance of the above system let $\alpha_{m,k}$ be the probability of a particular k errors in a hop of m symbols. Then

$$\alpha_{m,k} = \mathrm{E}\left[p_b^k(r)(1 - p_b(r))^{(m-k)}\right] \tag{5.6}$$

where the expectation is with respect to a Rayleigh distributed random variable r with probability density function given by (5.4). Also $p_b(r)$ is the error probability of the channel conditioned on r. For BFSK signaling and noncoherent reception this is given by

$$p_b(r) = \frac{1}{2}e^{-\frac{E_s r^2}{2N_0}}, \tag{5.7}$$

where the energy per symbol $E_s = \frac{E_b}{L}$, E_b being the energy per bit. Substituting (5.7) in (5.6) we get an expression for $\alpha_{m,k}$ as follows

$$\begin{aligned} \alpha_{m,k} &= \mathrm{E}[p_b^k(r)(1 - p_b(r))^{(m-k)}] \\ &= (\frac{1}{2})^k \int_0^\infty e^{-\frac{kE_b r^2}{2N_0 L}}(1 - \frac{1}{2}e^{-\frac{E_b r^2}{2N_0 L}})2re^{-r^2} \; dr \\ &= \sum_{i=0}^{m-k} \begin{pmatrix} m-k \\ i \end{pmatrix} \frac{(\frac{1}{2})^{k+i}(-1)^i}{(k+i)\frac{E_s}{2N_0} + 1}. \end{aligned} \tag{5.8}$$

We now define the following events concerning a diversity symbol which are of interest.

E_G = The event a symbol is in error and the corresponding hop is declared "good";

C_G = The event a symbol is correct and the corresponding hop is declared "good";

E_B = The event a symbol is in error and the corresponding hop is declared "bad";

C_B = The event a symbol is correct and the corresponding hop is declared "bad";

B = The event the hop is declared bad which is the same as $E_B \bigcup C_B$.

Then, from the decoding process, the corresponding probabilities are given by

$$\begin{aligned}
P(E_G) &= \mathrm{E}\left[p_b(R)\sum_{i=0}^{\lambda-1}\begin{pmatrix}\Lambda\\ i\end{pmatrix}p_b(R)^i(1-p_b(R))^{\Lambda-i}\right]\\
&= \sum_{i=0}^{\lambda-1}\begin{pmatrix}\Lambda\\ i\end{pmatrix}\mathrm{E}\left[p_b(R)^{i+1}(1-p_b(R))^{\Lambda-i}\right]\\
&= \sum_{i=0}^{\lambda-1}\begin{pmatrix}\Lambda\\ i\end{pmatrix}\alpha_{\Lambda+1,i+1}\\
P(C_G) &= \mathrm{E}\left[(1-p_b(R))\sum_{i=0}^{\lambda-1}\begin{pmatrix}\Lambda\\ i\end{pmatrix}p_b(R)^i(1-p_b(R))^{\Lambda-i}\right]\\
&= \sum_{i=0}^{\lambda-1}\begin{pmatrix}\Lambda\\ i\end{pmatrix}\mathrm{E}\left[p_b(R)^i(1-p_b(R))^{\Lambda-i+1}\right]
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{i=0}^{\lambda-1} \begin{pmatrix} \Lambda \\ i \end{pmatrix} \alpha_{\Lambda+1,i} \\
P(E_B) &= \mathrm{E}\left[p_b(R) \sum_{i=\lambda}^{\Lambda} \begin{pmatrix} \Lambda \\ i \end{pmatrix} p_b(R)^i (1-p_b(R))^{\Lambda-i} \right] \\
&= \sum_{i=\lambda}^{\Lambda} \begin{pmatrix} \Lambda \\ i \end{pmatrix} \alpha_{\Lambda+1,i+1} \\
P(C_B) &= \mathrm{E}\left[(1-p_b(R)) \sum_{i=\lambda}^{\Lambda} \begin{pmatrix} \Lambda \\ i \end{pmatrix} p_b(R)^i (1-p_b(R))^{\Lambda-i} \right] \\
&= \sum_{i=\lambda}^{\Lambda} \begin{pmatrix} \Lambda \\ i \end{pmatrix} \alpha_{\Lambda+1,i} \\
\text{and, } P(B) &= \mathrm{E}\left[\sum_{i=\lambda}^{\Lambda} \begin{pmatrix} \Lambda \\ i \end{pmatrix} p_b(R)^i (1-p_b(R))^{\Lambda-i} \right] \\
&= \sum_{\lambda}^{\Lambda} \begin{pmatrix} \Lambda \\ i \end{pmatrix} \alpha_{\Lambda,i} \ . \qquad (5.9)
\end{aligned}
$$

To calculate the probability of bit error P_b we proceed as follows. First we calculate the probability of error and j hops are received as good. We then sum over values of j from 0 to L. Thus

$$P_b = \sum_{j=0}^{L} \Pr\{\text{error and } j \text{ hops "good"}\} \qquad (5.10)$$

where for j odd

$$\Pr\ \{\text{ error and } j \text{ hops good}\} = \sum_{k=\lfloor j/2 \rfloor+1}^{j} \begin{pmatrix} j \\ k \end{pmatrix} P(E_G)^k P(C_G)^{j-k} \begin{pmatrix} L \\ j \end{pmatrix} P(B)^{L-j};$$

and for j even

$$\Pr\{\text{ error and } j \text{ hops good }\} = \left[\sum_{k=j/2+1}^{j} \begin{pmatrix} j \\ k \end{pmatrix} P(E_G)^k P(C_G)^{j-k} + \frac{1}{2} \begin{pmatrix} j \\ j/2 \end{pmatrix} P(E_G)^{j/2} P(C_G)^{j/2} \right] \begin{pmatrix} L \\ j \end{pmatrix} P(B)^{L-j}.$$

We can now evaluate the performance of such a system and compare it to the case when $\Lambda = 0$; i.e., with no side information case. In this case for L odd

$$P_b = \sum_{k=\lfloor L/2 \rfloor + 1}^{L} \begin{pmatrix} L \\ k \end{pmatrix} \mathrm{E}\left[p(R)\right]^k \mathrm{E}\left[(1 - p(R))\right]^{L-k},$$

and for L even

$$P_b = \sum_{k=L/2+1}^{L} \begin{pmatrix} L \\ k \end{pmatrix} \mathrm{E}\left[p(R)\right]^k \mathrm{E}\left[(1 - p(R))\right]^{L-k} + \frac{1}{2} \begin{pmatrix} L \\ L/2 \end{pmatrix} \mathrm{E}\left[p(R)\right]^{L/2} \mathrm{E}\left[(1 - p(R))\right]^{L/2}.$$

Thus we have derived a formula for the symbol error probability. However, cares must be taken into consideration when calculating the probability of bit error in two matters. First, for the uninteresting case $L = 1$, the performance is the same for any choice of Λ. This is obvious since we only have one symbol to decide which signal was transmitted. In this case the probability of bit error is

$$\begin{aligned} P_b &= \mathrm{E}(p(R)) = \frac{1}{2 + \frac{E_b}{N_0}} \\ &= \alpha_{1,1}. \end{aligned}$$

Second when computing the $j = 0$ term in (5.10) we need to calculate Pr{error and "0" hops are good }. Since in the case when all the hops are declared as "bad" we combine all the symbols prior to the "side information test" which assumes the availability of a buffer storage equal to the diversity level. Thus,

$$
\begin{aligned}
\Pr\{\text{error and 0 hops are good}\} &= \\
\Pr\{\text{error and } L \text{ hops are bad}\} &= \\
\Pr\{\text{error} \mid L \text{ hops are bad}\}\Pr\{L \text{ hops are bad}\} &= \\
\sum_{k=\lfloor L/2\rfloor+1}^{L} \begin{pmatrix} L \\ k \end{pmatrix} P(E_B)^k P(CB)^{L-k} & \qquad \text{for } L \text{ odd, and} \\
= \sum_{k=L/2+1}^{L} \begin{pmatrix} L \\ k \end{pmatrix} P(E_B)^k P(C_B)^{L-k} & \\
+\frac{1}{2} \begin{pmatrix} L \\ L/2 \end{pmatrix} P(E_B)^{L/2} P(C_B)^{L/2} & \qquad \text{for } L \text{ even.}
\end{aligned}
$$

Figure 5.4 shows the performance of the system for $\Lambda = 1$ and for various values of $\frac{E_b}{N_0}$ by plotting the bit-error-rate (BER) versus the diversity level. It is noticed that for a big range of bit energy-to-noise ratio there is a degradation in performance due to the use of test bits. This holds for all values of diversity levels.

In Figure 5.5- Figure 5.7 we show the BER versus diversity for higher values of Λ, and for a fixed value of Λ we show the behaviour for each threshold λ less than or equal to Λ. Initially, this is done for a small value of $\frac{E_b}{N_0}$, namely 10 dB. In some cases it is apparent that the optimum threshold λ_{opt} depends on the diversity level used, the bit energy-to-noise ratio, and Λ; i.e.,

$$\lambda_{opt} = \lambda_{opt}(\frac{E_b}{N_0}, \Lambda, L)$$

For instance we have a slight improvement when using the test bits with $\lambda_{opt} = \lambda_{opt}(10dB, 4, 3) = 1$; that is we erase if one or more of the received test pattern is in error.

Similarly, Figure 5.8 - Figure 5.13 exhibits the BER for larger values of $\frac{E_b}{N_0}$, mainly $\frac{E_b}{N_0}$=16 dB and 19 dB. For most cases of interest we either have no improvement when using the side information, or the improvement is not substantial and might not be worth implementing this type of side information generation. This improvement becomes better, however, for large $\frac{E_b}{N_0}$ with large Λ as apparent from Figure 5.14. This is not true if Λ is small (e.g. see Figure 5.8).

5.3.2 Soft Decision Decoding

Now consider the case of soft decision decoding of diversity transmissions. That is we want to evaluate the probability of bit error for a repetition code used on a binary additive channel. In this case we consider the performance of two combining methods for the case $m = 1$: linear combining and square-law combining of symbols which are declared "good" by the receiver. In case all diversity receptions of a given symbol are declared "bad" all the diversity receptions are combined. These types of combining have been studied extensively for use in noncoherent systems. In particular square-law combining is the optimum noncoherent combining technique for Rayleigh fading. However, square-law combining may not be optimum when using side information about the received symbols. Linear and square-law combining are believed to be suboptimal for the side information case. We first analyze the performance of a linear combiner.

Linear Combining: In linear combining, the decision statistics are the sums of linear terms. If a diversity reception is declared as unreliable, it is excluded from the sum and all the emphasis is given to the "good" receptions. A block diagram which

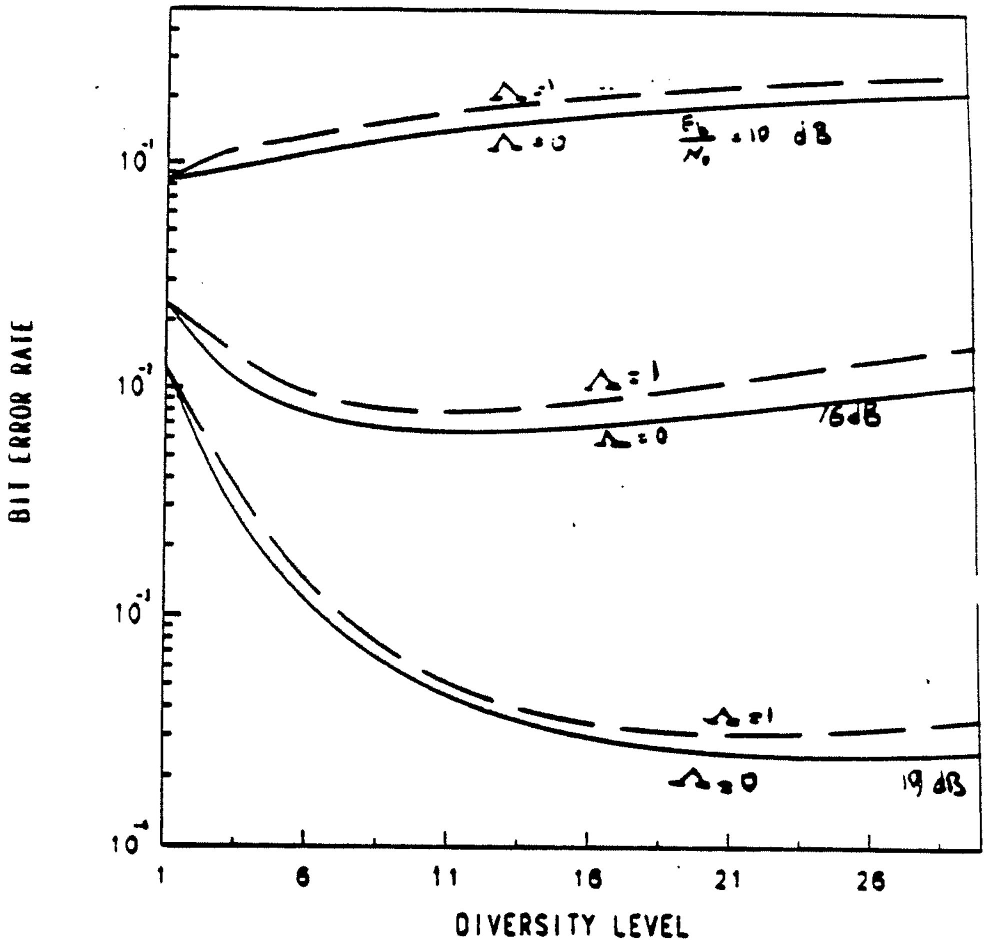

Figure 5.4: Plot of bit-error-rate versus diversity level for various values of $\frac{E_b}{N_0}$ ($\Lambda = 0, 1$).

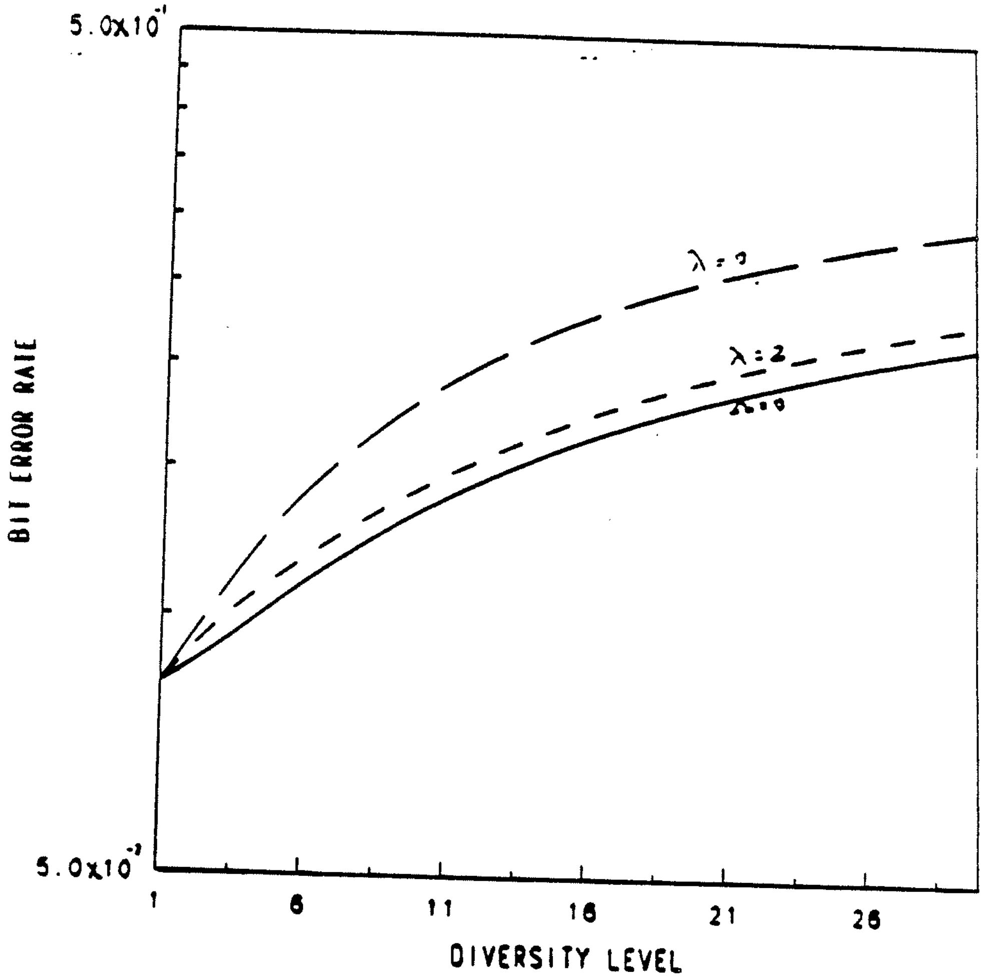

Figure 5.5: Plot of bit-error-rate versus diversity level for various values of λ ($\Lambda =$ 0, 2 and $\frac{E_b}{N_0}$ = 10 dB).

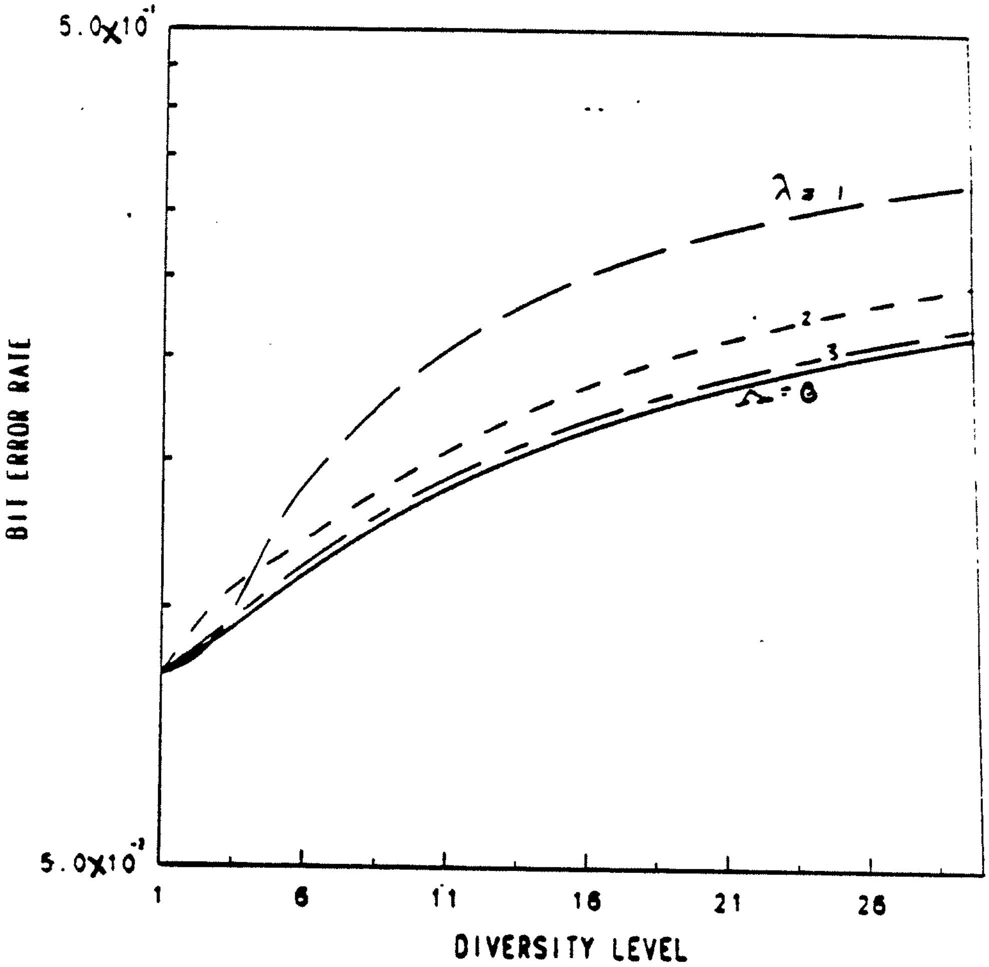

Figure 5.6: Plot of bit-error-rate versus diversity level for various values of λ ($\Lambda = 0, 3$ and $\frac{E_b}{N_0} = 10$ dB).

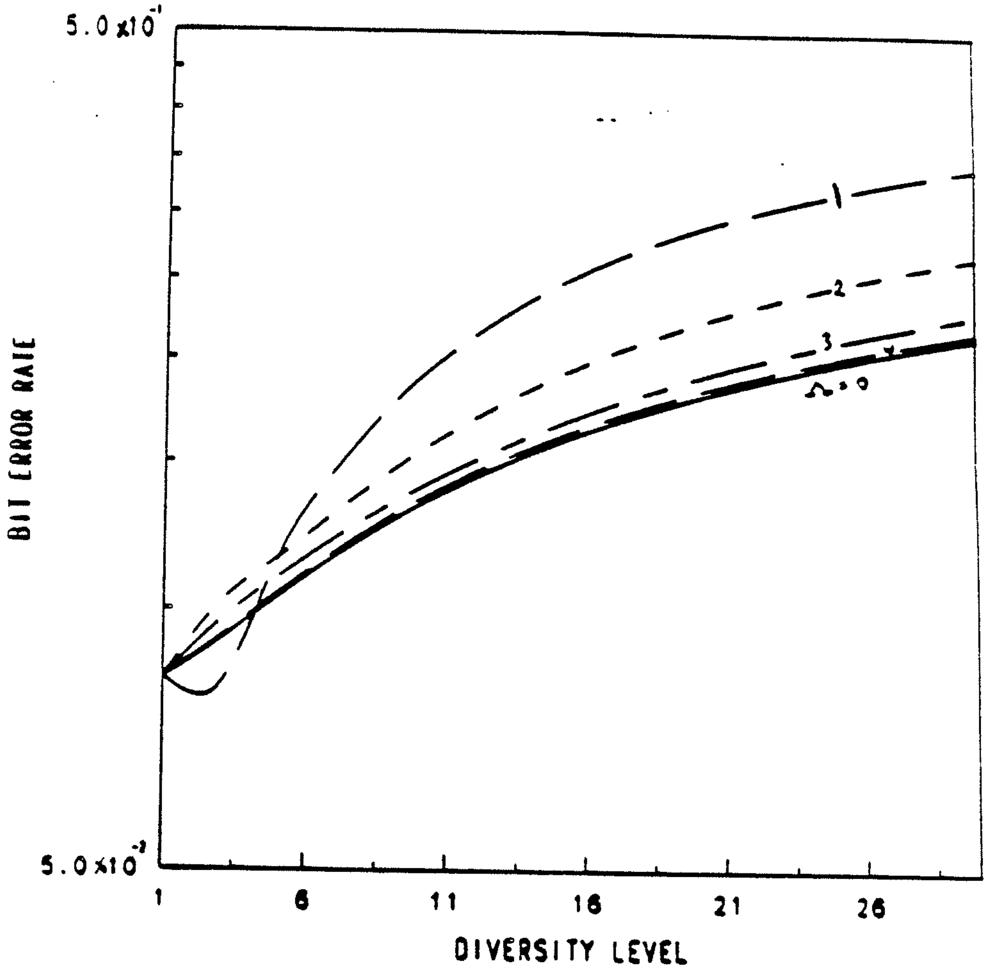

Figure 5.7: Plot of bit-error-rate versus diversity level for various values of λ ($\Lambda =$ $0, 4$ and $\frac{E_b}{N_0} = 10$ dB).

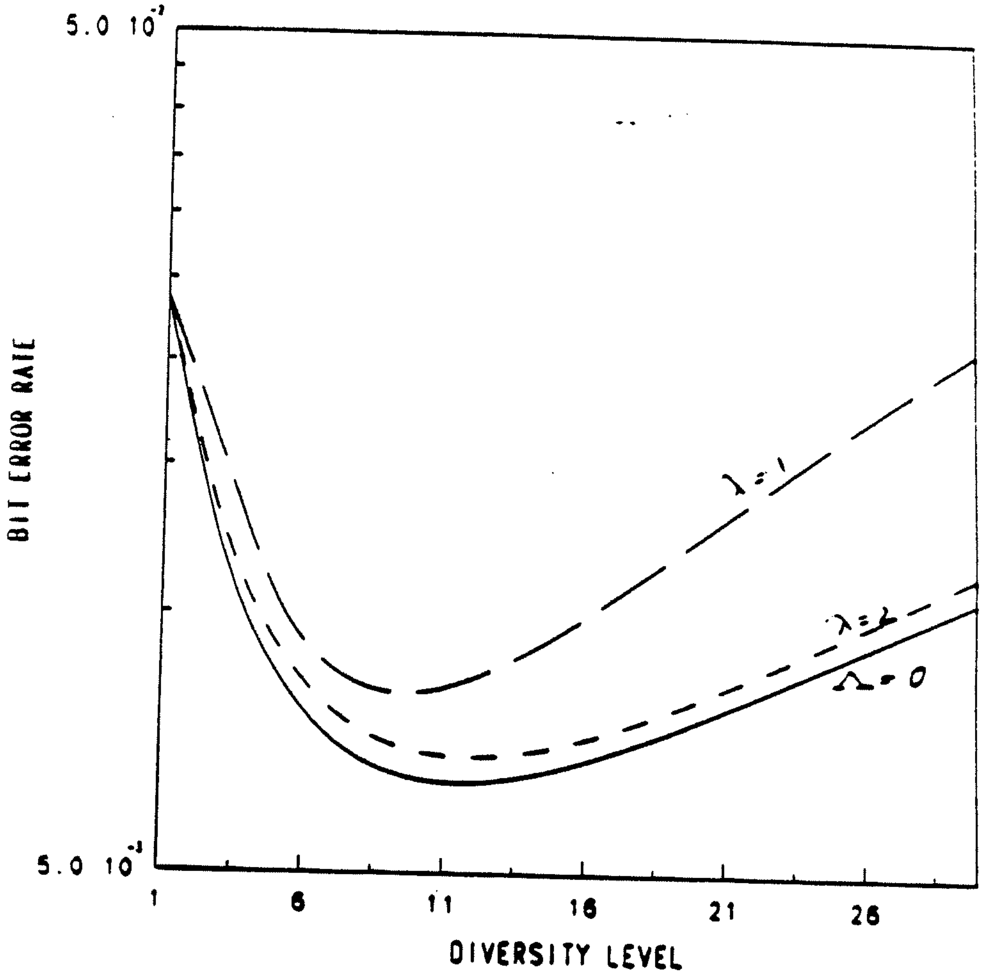

Figure 5.8: Plot of bit-error-rate versus diversity level for various values of λ ($\Lambda =$ 0, 2 and $\frac{E_b}{N_0} = 16$ dB).

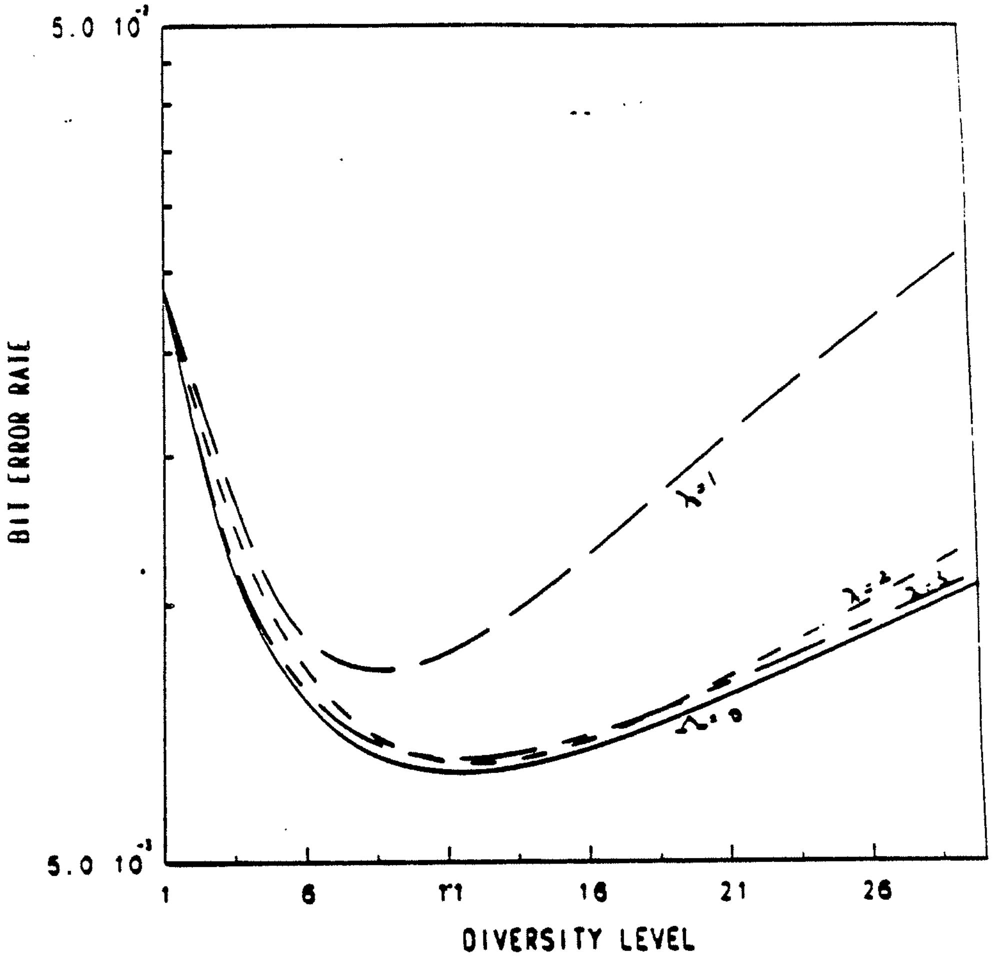

Figure 5.9: Plot of bit-error-rate versus diversity level for various values of λ ($\Lambda = 0,3$ and $\frac{E_b}{N_0} = 16$ dB).

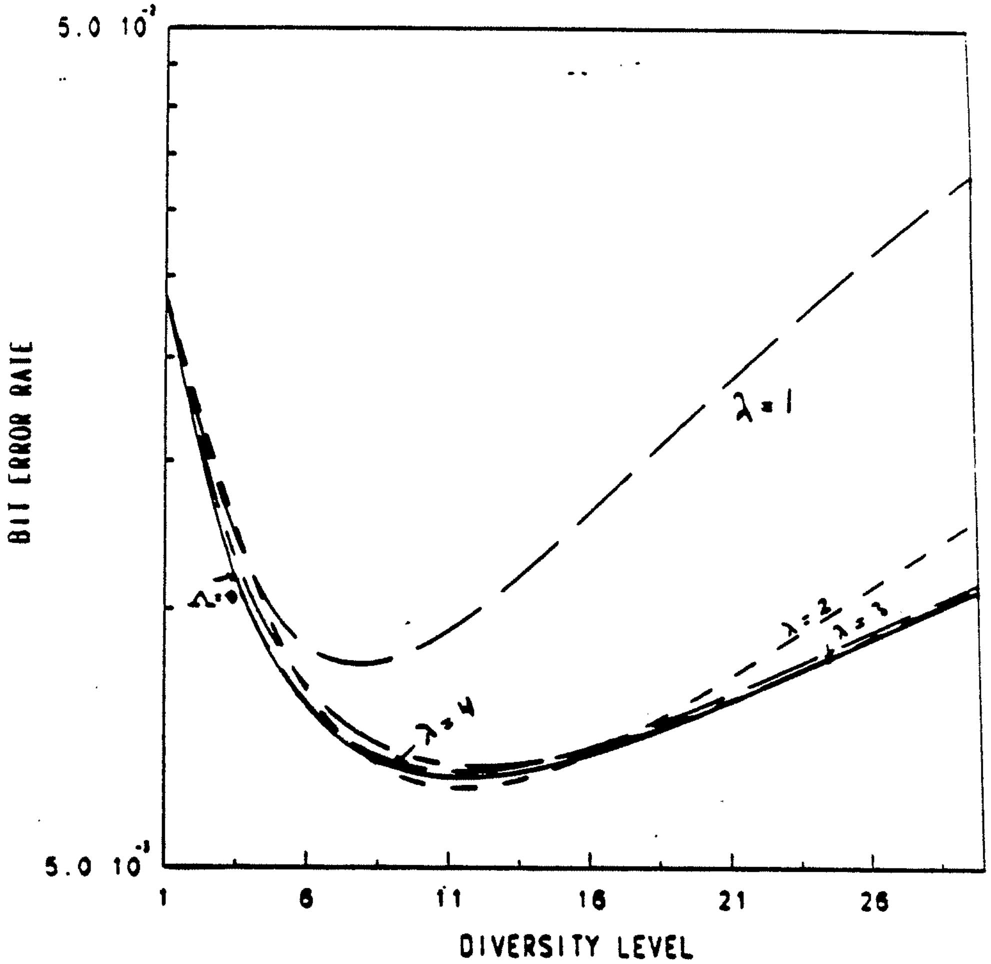

Figure 5.10: Plot of bit-error-rate versus diversity level for various values of λ ($\Lambda =$ 0, 4 and $\frac{E_b}{N_0}$ = 16 dB).

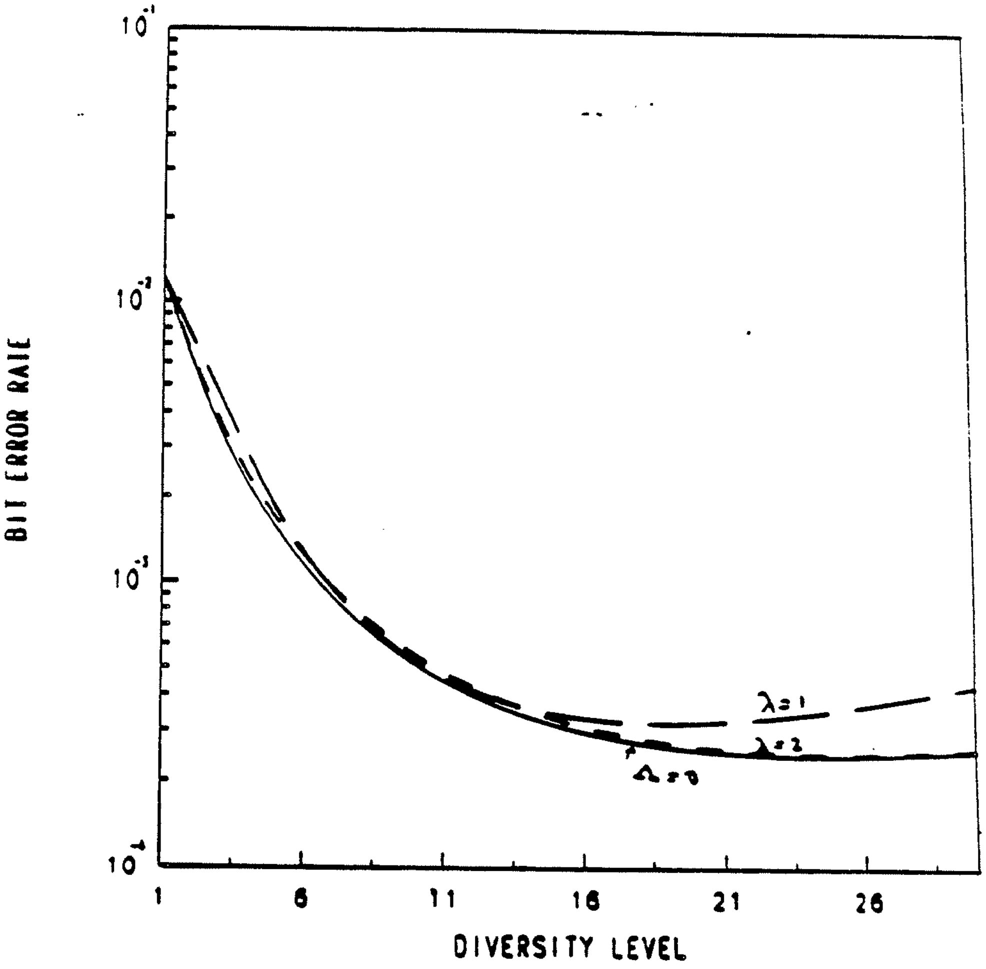

Figure 5.11: Plot of bit-error-rate versus diversity level for various values of λ ($\Lambda =$ 0, 2 and $\frac{E_b}{N_0} = 19$ dB).

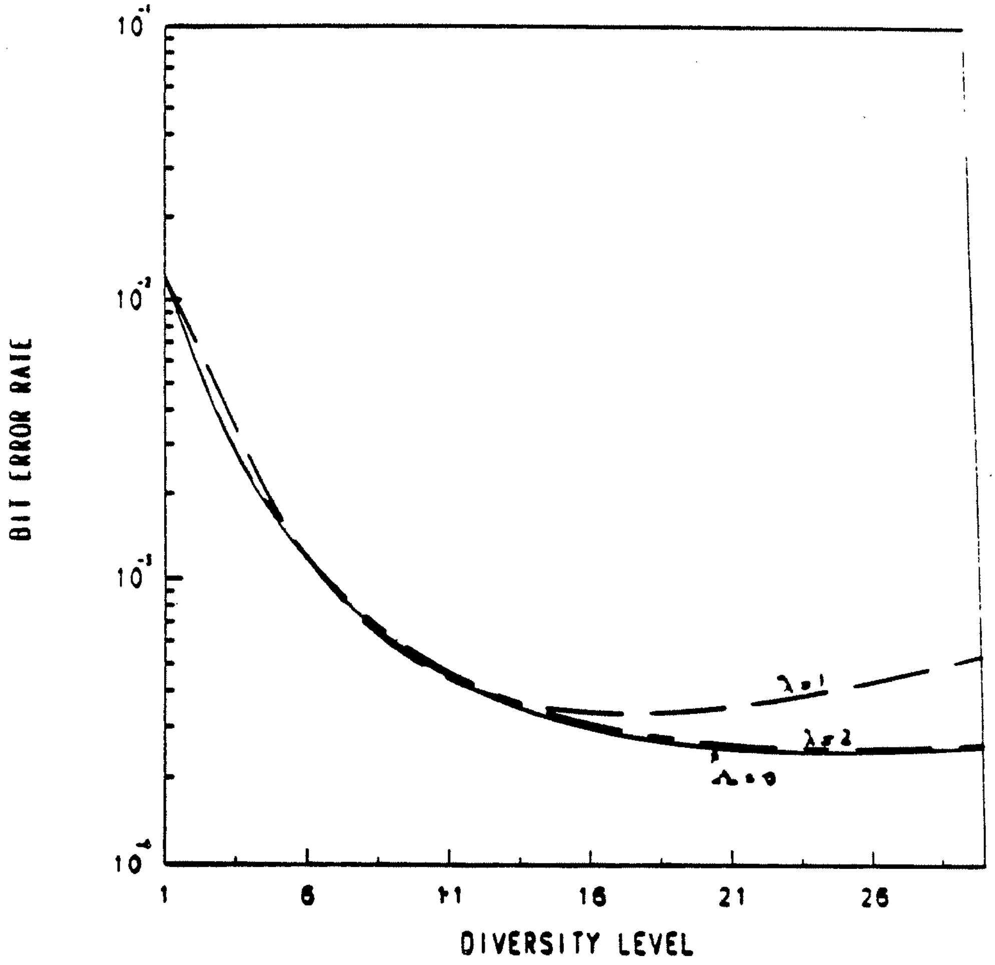

Figure 5.12: Plot of bit-error-rate versus diversity level for various values of λ (Λ = 0, 3 and $\frac{E_b}{N_0}$ = 19 dB).

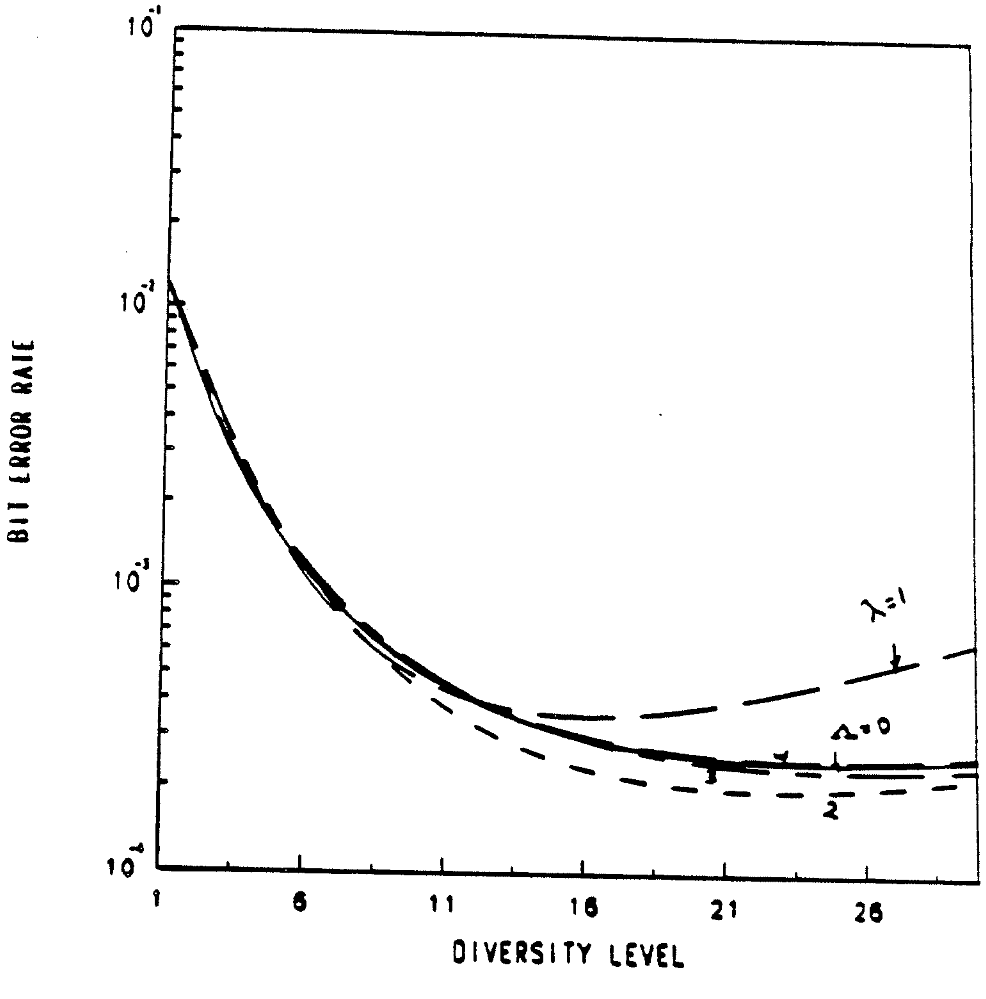

Figure 5.13: Plot of bit-error-rate versus diversity level for various values of λ ($\Lambda = 0, 4$ and $\frac{E_b}{N_0} = 19$ dB).

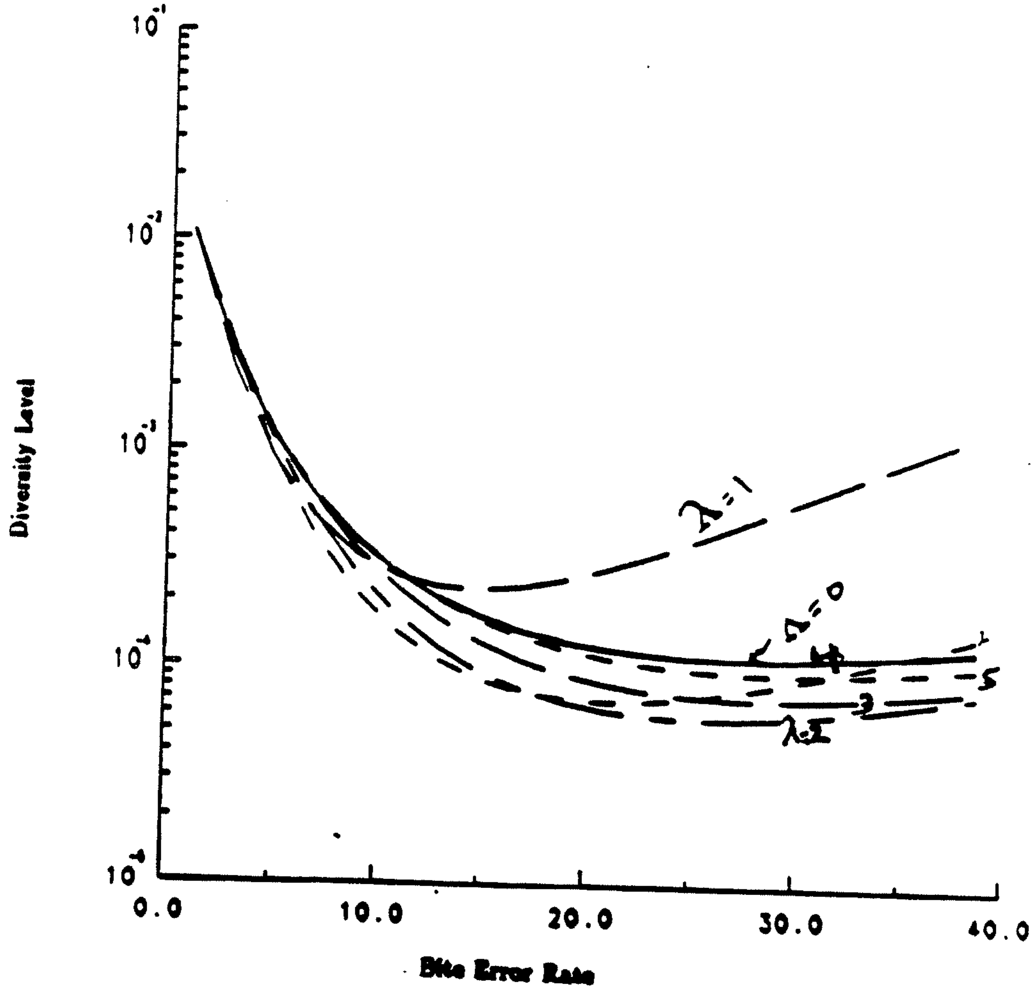

Figure 5.14: Plot of bit-error-rate versus diversity level for various values of λ ($\Lambda =$ 0,8 and $\frac{E_b}{N_0} = 19$ dB).

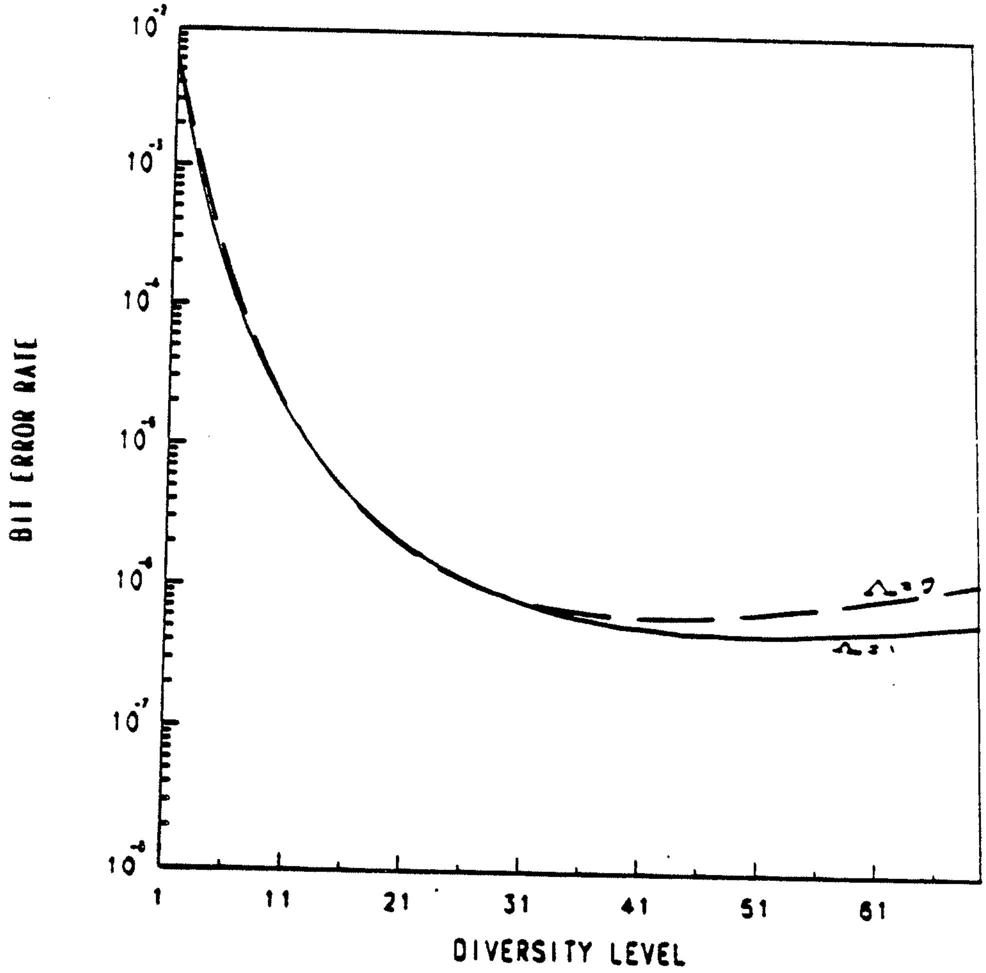

Figure 5.15: Plot of bit-error-rate versus diversity level for $\frac{E_b}{N_0} = 22$ dB.

shows linear combining is shown in Figure 5.15. The decision statistics are the sums of the outputs of the envelope detectors. Given the symbol 0 is sent, and given j hops are declared good, the decision statistics are

$$Z_0(r_1,\ldots,r_j) = \sum_{l=1}^{j} \sqrt{(X_{0,l}+\nu r_l^2)^2 + Y_{0,l}^2} \equiv \sum_{l=1}^{j} W_{0,l}(r_l) \tag{5.11}$$

$$Z_1(r_1,\ldots,r_j) = \sum_{l=1}^{j} \sqrt{X_{1,l}^2 + Y_{1,l}^2} \equiv \sum_{l=1}^{j} W_{1,l}(r_l) \tag{5.12}$$

where

$$\nu = \sqrt{\frac{2E_b}{N_0 L}}$$

and given the random vector $\underline{r} = (r_1, r_2, ..., r_j)$ the random variables $\{X_{k,l},\ Y_{k,l}\ :\ k = 0,1,\ 1 \le l \le j\}$ are mutually independent zero-mean, unit-variance Gaussian random variables. In (5.11) and (5.12) we explicity show the dependence on the L Rayleigh distributed random variables $r_1 \ldots r_j$ which are independent and identically distributed.

Now we calculate the probability of bit error for this type of soft decision decoding. To calculate the probability of bit error $P\{$ error $\}$, first we calculate the probability of error and j hops are received as "good", then we sum over all values of j from 0 to L. That is,

$$\begin{aligned} P\{\text{error}\} &= \sum_{j=0}^{L} P\{\text{error and } j \text{ hops good}\} \\ &= \sum_{j=0}^{L} \binom{L}{j} E\left\{ P\left\{ \sum_{l=1}^{j} W_{0,l}(r_l) \le \sum_{l=1}^{j} W_{1,l}(r_l), \right.\right. \\ &\qquad \left.\left. I_1 = I_2 = \ldots = I_j = 1, I_{j+1} = \ldots = I_L = 0 \right\} \middle| \underline{r} \right\} \end{aligned} \tag{5.13}$$

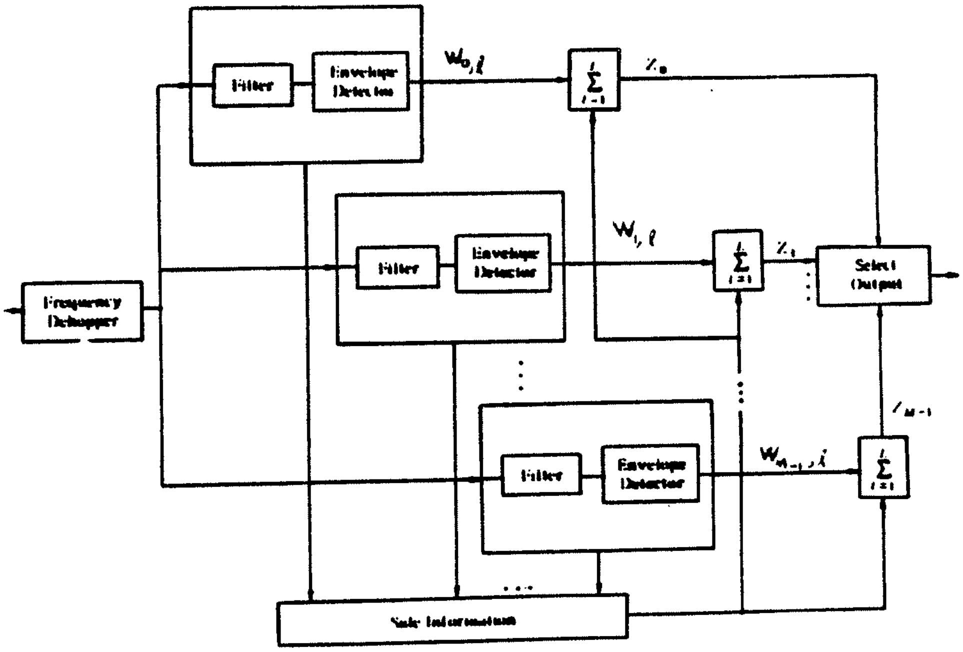

Figure 5.16: Linear Combining of M-ary Diversity Symbols.

where I_k is defined as

$$I_k = \begin{cases} 1 \ , & \text{if test pattern checks for hop k} \\ 0 \ , & \text{if test pattern fails for hop k} \ . \end{cases}$$

it can be shown that

$$\begin{aligned} P\{\text{error}\} \ &= \ \sum_{j=0}^{L} \binom{L}{j} P\Bigg\{ \sum_{l=1}^{j} W_{0,l} \leq \sum_{l=1}^{j} W_{1,l} \ , \\ &\qquad\qquad I_1 = \ldots = I_j = 1, I_{j+1} = \ldots = I_L = 0 \Bigg\} \end{aligned}$$

where $W_{0,l} = \mathrm{E}\ W_{0,l}(r_l)$ and $W_{1,l} = \mathrm{E}\ W_{1,l}(r_l)$, expectation being with respect to r_l.

In order to perform the above calculation it is useful to have a density function for $W_{i,l}$ and $I_l = 1$, i.e. determine

$$\mathcal{F}_{W_{i,l}, I_l=1}(w_{i,l}) \triangleq \frac{d \ \ P\{W_{i,l} \leq w_{i,l} \cap I_d = 1\}}{d \ \ w_{i,l}}$$

To do this note that

$$\begin{aligned} P\{W_{i,l} \leq w_{i,l} \cap I = 1\} \ &= \ E[P\{W_{i,l} \leq w_{i,l} \cap i_l = 1 | r_l\}] \\ &= \ E[P\{w_{i,l} \leq w_{i,l} | r_l\} P\{I_l = 1 | r_l\}] \ \ . \end{aligned}$$

Since differentiation is a linear operation we have

$$d \ \frac{P\{W_{i,l} \leq w_{i,l} \cap I_l = 1\}}{d \ \ w_{i,l}} = E\left[\frac{d \ \ P\{W_{i.l} \leq w_{i,l} | r_l\}}{d \ \ w_{i,l}} P\{I_l = 1 | r_l\} \right] \ \ .$$

This can be computed knowing that given zero is sent $W_{o,l}$ and $W_{1,l}$ are independent random variables. Furthermore,

$$P\{I_l = 1 | r_l\} \ \triangleq \ \mathcal{F}(r_l)$$

$$
\begin{aligned}
&= \sum_{i=0}^{\lambda-1} \binom{\Lambda}{i} p_b(r)^i (1 - p_b(r))^{\Lambda-i} \\
&= \sum_{i=0}^{\lambda-1} \binom{\Lambda}{i} \sum_{k=0}^{\Lambda-i} \binom{\Lambda^i}{k} (-1)^k p_b(r)^{i+k} \quad .
\end{aligned}
$$

Substituting for the value of $p_b(r)$ given in (7) we get

$$
\mathcal{F}(r) = \sum_{i=0}^{\lambda-1} \sum_{k=0}^{\Lambda-i} \binom{\Lambda}{i} \binom{\Lambda - i}{k} (-1)^k \left(\frac{1}{2}\right)^{i+k} e^{-(i+k)\frac{r^2\nu^2}{4}} \quad .
$$

Also,

$$
\begin{aligned}
\mathcal{F}_{w_{o,l}|R}(w_{o,l}|r) &= \frac{d \;\; P\{W_{o,l} \leq w_{o,l}|r_l = r\}}{d \;\; w_{o,j}} \\
&= w_{o,l} e^{-\frac{w_{o,l}^2}{2} - r^2 \frac{\nu^2}{2}} I_0(\nu r w_{o,l})
\end{aligned}
$$

where $I_0(x)$ is the zeroth order modified Bessel's Function. The density function of r is given by

$$
\mathcal{F}_R(r) = \begin{cases} 2r^{-r^2} & , \quad r \geq 0 \\ 0 \;, & \quad r < 0 \;. \end{cases}
$$

Thus the density function we are interested in is

$$
\begin{aligned}
\mathcal{F}_{W_{i,l},I_{l=1}}(w) &= \int_0^\infty \mathcal{F}_R(r) \mathcal{F}_{W_{i,l}|R_j}(w|r) \mathcal{F}(r) dr \\
&= \sum_{i=0}^{\lambda-1} \sum_{k=0}^{\Lambda-i} \binom{\Lambda}{i} \binom{\Lambda - i}{k} (-1)^k \left(\frac{1}{2}\right)^{i+k-1} \\
&\qquad \int_0^\infty r e^{-r^2} w e^{-\frac{w^2}{2} - r^2 \frac{\nu^2}{2}} I_0(\nu r w) e^{-(i+k)\frac{\nu^2 r^2}{4}} dr
\end{aligned}
$$

which can be simplified to

$$
\begin{aligned}
\mathcal{F}_{W_{0,l,I_l}} = 1^{(w)} &= \sum_{i=0}^{\lambda-1} \sum_{k=0}^{\Lambda-i} \binom{\Lambda}{i} \binom{\Lambda - i}{k} (-1)^k (\frac{1}{2})^{i+k-1} w e^{-\frac{w^2}{2}} \\
&\qquad \int_0^\infty r e^{-r^2} e^{-r^2 \frac{\nu^2}{2}} e^{-(i+k)\frac{\nu^2 r^2}{4}} I_0(\nu r w) dr \\
&= \sum_{i=0}^{\lambda-1} \sum_{k=0}^{\Lambda-i} \binom{\Lambda}{i} \binom{\Lambda - i}{k} (-1)^k \left(\frac{1}{2}\right)^{i+k-1}
\end{aligned}
$$

$$\int_0^\infty r e^{-\left(1+\frac{\nu^2}{2}+(i+k)\frac{\nu^2}{4}\right)r^2} \; I_0(\nu r w)dr$$
$$= \sum_{i=0}^{\lambda-1} \sum_{k=0}^{\Lambda-i} \binom{\Lambda}{i}\binom{\Lambda-i}{k}(-1)^k \frac{1}{2}^{i+k-1} w e^{-\frac{w^2}{2}}$$
$$\frac{1}{2+\nu^2+(i+k)\frac{\nu^2}{2}} \; e^{\frac{\frac{\nu^2}{2}w^2}{2+\nu^2+(i+k)\frac{\nu^2}{2}}}$$
$$\equiv \begin{cases} \sum_{i=0}^{\lambda-1} \sum_{k=0}^{\Lambda-1} C_{i,k} \; w \; e^{-\mathcal{A}_{i,k}w^2} & , \quad w \leq 0 \\ 0 & , \quad w < 0 \;, \end{cases} \tag{5.14}$$

where

$$\mathcal{A}_{i,k} \triangleq \frac{1}{2} - \frac{\nu^2/2}{2+(2+i+k)\frac{\nu^2}{2}} \quad ;$$

and

$$C_{i,k} \triangleq \frac{\binom{\Lambda}{i}\binom{\Lambda-i}{k}(-1)^k \frac{(1}{2)}^{i+k-1}}{2+(2+i+k)\frac{\nu^2}{2}}.$$

In arriving at (5.14) we made use of the identity

$$\int_0^\infty r \; e^{-\mathcal{A}r^2} \; I_0(br)dr = \frac{1}{2\mathcal{A}}e^{\frac{b^2}{4\mathcal{A}}} \quad .$$

On the other hand to calculate $\mathcal{F}_{W_{1,l}I_l=1}(w)$ we use

$$\mathcal{F}_{W_{1,l|R_l}}(w|r) = \begin{cases} we^{-\frac{w^2}{2}} & , \quad w \geq 0 \\ 0 & , \quad w < 0 \end{cases}$$

which is independent of the fade r. Therefore, we have

$$\mathcal{F}_{W_{1,l;I_l=1}}(w) = \begin{cases} we^{-\frac{w^2}{2}} \int_0^\infty \mathcal{F}_R(r)\mathcal{F}(r)dr, & w \geq 0 \\ 0 & , \quad w < 0 \end{cases}$$

Certainly the integral above is equal to one.

Thus we have expressions for $\mathcal{F}_{W_{0,l;I_l=1}}(w)$ and $\mathcal{F}_{W_{1,l;I_l=1}}$. In order to calculate the probability of bit error we need to find the probability density functions of Z_0 and Z_1 where

$$\begin{aligned} Z_0 &= \sum_{l=1}^{j} W_{0,l;I_0=1} \quad , \qquad \text{and} \\ Z_1 &= \sum_{l=1}^{j} W_{1,l;I_l=1}. \end{aligned}$$

These are the j-th fold convolution of the probability density functions calculated earlier. Unfortunately, closed form analytical expressions for $\mathcal{F}_{Z_0}(z)$ and $\mathcal{F}_{Z_1}(z)$ are not possible. However, when using the Chernoff bound it is easy to show that

$$P_b \leq \sum_{j=0}^{L} \binom{L}{j} \min_{0<s_j<1} [A(s_j)B(s_j, a_{i,k})]^j \quad ,$$

where

$$\begin{aligned} A(s) &= 1 + s\sqrt{\frac{\pi}{2}} \; e \; \frac{s^2}{2} \left[1 - \Phi\left(\frac{-s}{\sqrt{2}}\right)\right] \quad , \\ B(s, \mathcal{A}_{i,k}) &= \frac{1}{2\mathcal{A}_{i,k}} - \frac{s}{4\mathcal{A}_{i,k}}\sqrt{\frac{\pi}{A_{i,k}}} \; e\frac{s^2}{4\mathcal{A}_{i,k}} \left[1 - \Phi\left(\frac{s/2}{\sqrt{a_{i,k}}}\right)\right] \\ \text{and} \qquad \Phi(x) &= \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \end{aligned}$$

is the error function.

<u>**Square-Law Combining:**</u> In square-law combining, the decision statistics are the sums of linear terms: the squares of the outputs of an I-Q square-law detector are added, as shown in Figure 5.16.

For this case the decision statistics are

$$\begin{aligned} Z_0(r_1, \ldots, r_j) &= \sum_{l=1}^{j} \left[(X_{0,l} + \nu r_l^2)^2 + Y_{0,l}^2\right] \equiv \sum_{l=1}^{j} W_{0,l}(r_l) \\ Z_1(r_1, \ldots, r_j) &= \sum_{l=1}^{j} (X_{1,l}^2 + Y_{1,l}^2) \equiv \sum_{l=1}^{j} W_{1,l}(r_l) \end{aligned}$$

The main difference in calculating P_b for this case concerns the conditional

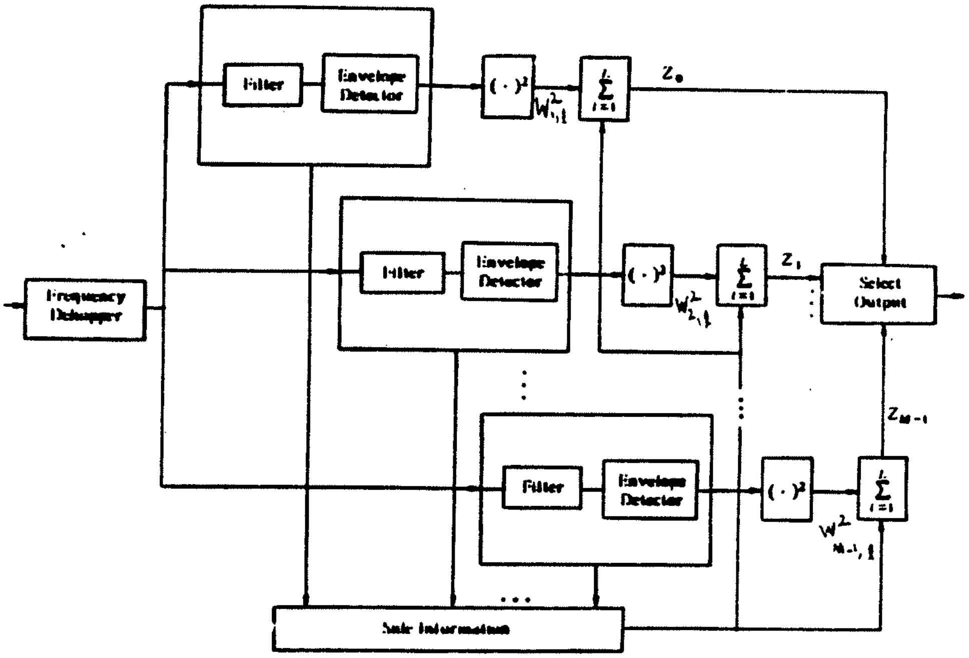

Figure 5.17: Square-Law Combining of M-ary Diversity Symbols.

probability density functions $\mathcal{F}_{W_{0,l}|R}(w|r)$ and $\mathcal{F}_{W_{1,l}|R}(w|r)$. Here we have:

$$\begin{aligned} \mathcal{F}_{W_{0,l}|R}(w|r) &= \frac{1}{2}e^{-\frac{w}{2}-r^2\frac{\nu^2}{2}} I_0\left(\sqrt{w}\nu r\right) \quad , \quad y \geq 0 \\ \mathcal{F}_{W_{1,l}|R}(w|r) &= \frac{1}{2}\, e^{-\frac{w}{2}} \qquad\qquad\qquad , y \geq 0 \end{aligned}$$

and both equal to 0 for $y < 0$. Performing computations similar to the previous case gives

$$\mathcal{F}_{W_{0,l;I_j}=1}(w) = \sum_{i=0}^{\lambda-1} \sum_{k=0}^{\Lambda-i} C'_{i,k}\; e^{-\mathcal{A}'_{i,k} w} \quad , \quad w \geq 0$$

where

$$\begin{aligned} C'_{i,k} &= \binom{\Lambda}{i}\binom{\Lambda-i}{k}(-1)^k\left(\frac{1}{2}\right)^{i+k-2} = 2C_{i,k} \\ \mathcal{A}'_{i,k} &= \mathcal{A}_{i,k} \quad , \end{aligned} \tag{5.15}$$

and

$$\mathcal{F}_{W_{1,l;I_j}=1}(w) = \frac{1}{2}\; e^{-\frac{w}{2}} \quad , \quad w \geq 0 \ .$$

It is easy in this case to calculate an exact expression for P_b. Using the characteristic functions method it is shown in Appendix E that the bit error rate is given by the following expression:

$$P_b = \sum_{i=0}^{L} \binom{L}{i} P(i) \quad ,$$

where

$$P(i) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \left(1 - \frac{1}{(1+2js)^i}\right) \frac{1}{js} \left(\sum_i \sum_k \frac{C_{i,k}}{\mathcal{A}_{i,k} - js}\right)^i ds$$

j being the complex root of -1. Thus we have an expression for P_b which is numerically much faster to compute than the convolution method.

The performance of soft decision combining with test bits is compared to that of no test bits. In the later case the probability of bit error is given by [35]

$$P_b = (\frac{1-\mu}{2})^L \sum_{k=0}^{L-1} \binom{L-k+1}{k} \ (\frac{1+\mu}{2})^k$$

where

$$\mu = \frac{\frac{E_s}{N_0}}{2+\frac{E_s}{N_0}}.$$

In Figures 5.17-5.19 we present the results of using these new formulae. Notice that the performance when transmitting test bits improves only for high signal to noise ratio, as is the case in hard decision combining. The results for hard decision combining and soft decision combining show that the method proposed to generate side information about each hop is unattractive for repetition coding. Effectively we are reducing the diversity gain due to erasing. The case treated in the next section makes the method more interesting, however.

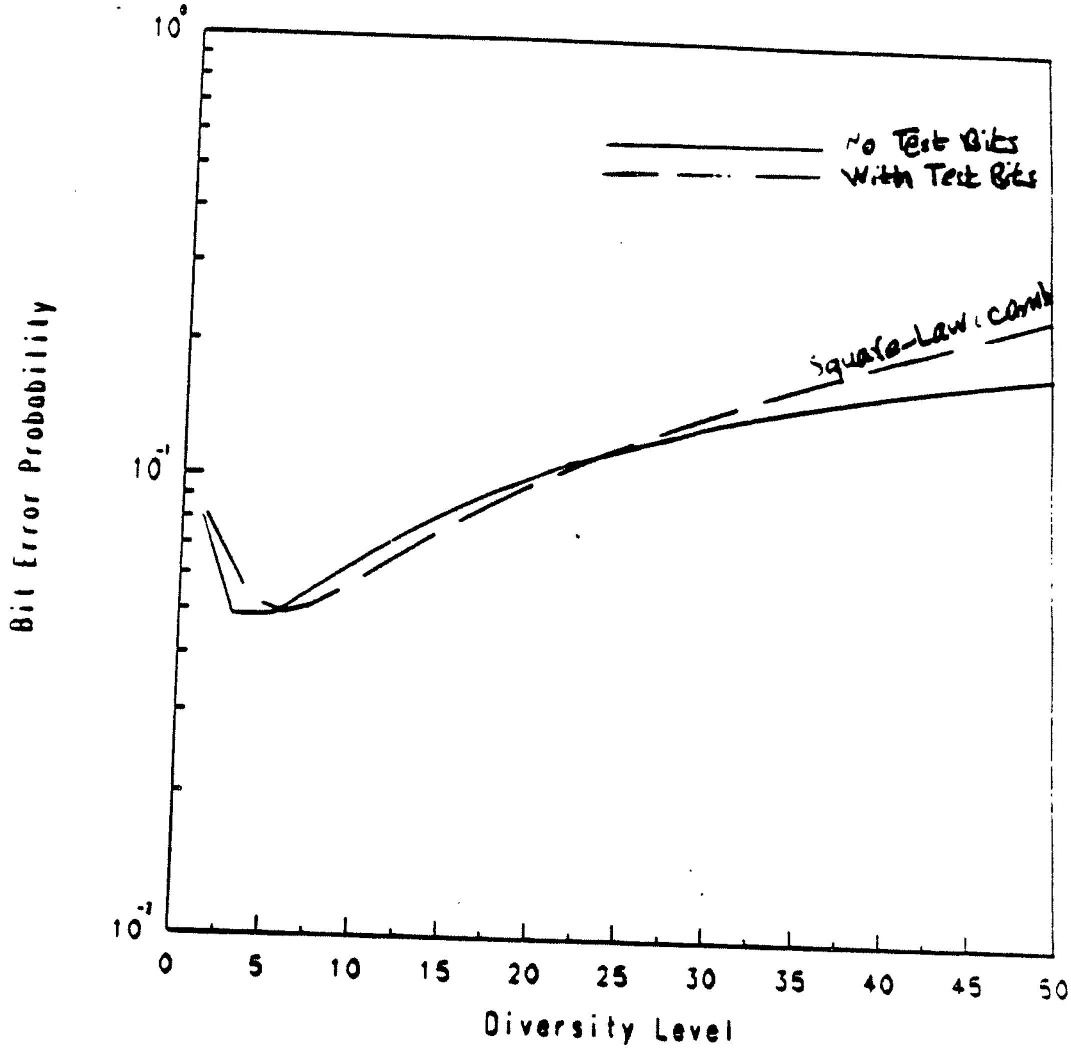

Figure 5.18: Bit error rate versus diversity level for soft decision combining ($\frac{E_b}{N_0}$ = 10 dB).

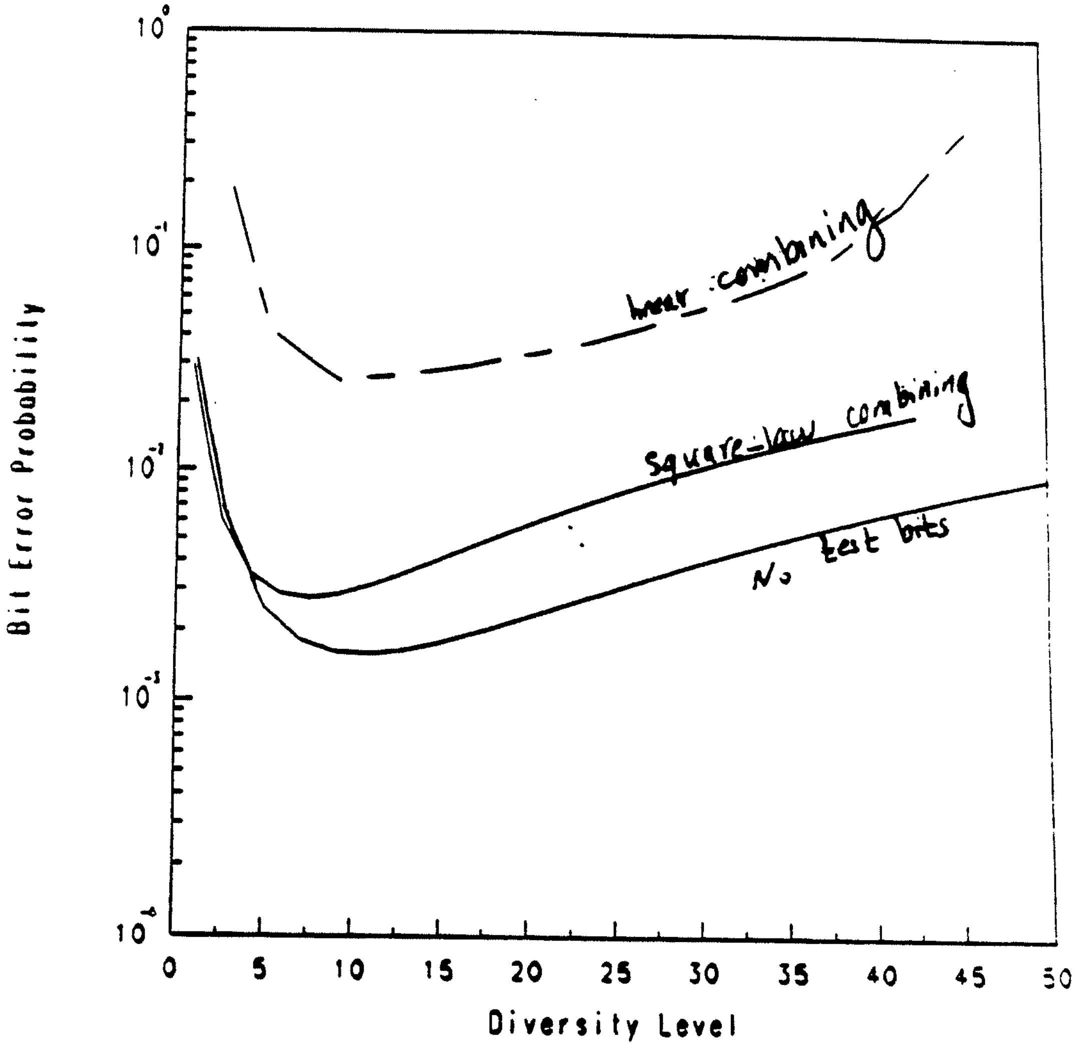

Figure 5.19: Bit error rate versus diversity level for soft decision combining ($\frac{E_b}{N_0}$ = 14 dB).

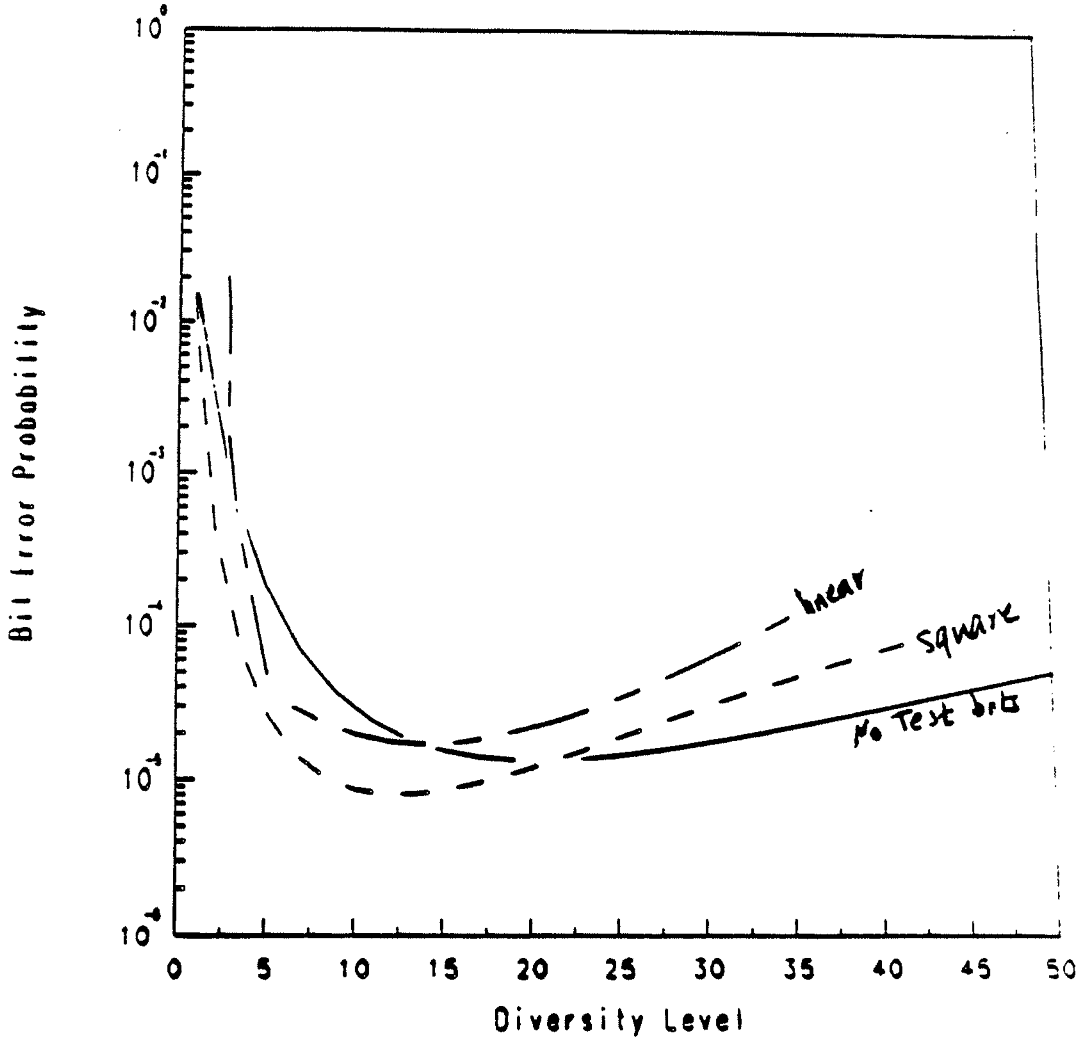

Figure 5.20: Bit error rate versus diversity level for soft decision combining ($\frac{E_b}{N_0}$ = 18 dB).

5.4 Performance of a Reed-Solomon Coded System

We found earlier that in most cases test patterns with repetition coding does not improve performance when compared to the no side information case, and when the performance does improve it does so by a small amount. In this section we analyze a coded SFH spread spectrum communication system with extended Reed-Solomon codes. In this case the block length n of the code is equal to the alphabet size M of the code alphabet. Reed-Solomon codes are desirable for their maximal distance separable (MDS) property in which, for an (n, k) Reed-Solomon code, the minimum distance of the code $d_{min} = n - k + 1$, which is the best distance value that could be achieved for any code with the same parameters. As pointed out before, each code symbol is represented as a sequence of m bits, and thus a symbol is in error if any of the m bits are estimated incorrectly. Therefore, we have a discrete memoryless channel with channel input alphabet $\{0, 1, ..., 2^m - 1\}$ and output alphabet $\{0, 1, ..., 2^m - 1, ?\}$ which is an M-ary symmetric-erasure channel described in Chapter 1.

All the events defined in the last section are still of interest; however, a symbol is in error if one or more of the m bits is in error; hence, the corresponding probabilities are modified as follows (supressing the dependance on the random variable r).

$$P(C_G) \quad = \quad \mathrm{E}\left[(1 - p_b)^m \sum_{i=0}^{\lambda-1} \begin{pmatrix} \Lambda \\ i \end{pmatrix} p_b^i (1 - p_b)^{\Lambda - i}\right]$$

$$\begin{aligned}
&= \sum_{i=0}^{\lambda-1} \begin{pmatrix} \Lambda \\ i \end{pmatrix} \mathrm{E}[p_b^i(1-p_b)^{\Lambda-i+m}] \\
&= \sum_{i=0}^{\lambda-1} \begin{pmatrix} \Lambda \\ i \end{pmatrix} \alpha_{\Lambda+m,i} \\
P(B) &= \mathrm{E}\left[\sum_{i=\lambda}^{\Lambda} \begin{pmatrix} \Lambda \\ i \end{pmatrix} p_b^i(1-p_b)^{\Lambda-i}\right] \\
&= \sum_{\lambda}^{\Lambda} \begin{pmatrix} \Lambda \\ i \end{pmatrix} \alpha_{\Lambda,i} \\
P(E_G) &= 1 - P(C_G) - P(B) \\
P(C_B) &= \mathrm{E}\left[(1-p_b)^m \sum_{i=\lambda}^{\Lambda} \begin{pmatrix} \Lambda \\ i \end{pmatrix} p_b^i(1-p_b)^{\Lambda-i}\right] \\
&= \sum_{i=\lambda}^{\Lambda} \begin{pmatrix} \Lambda \\ i \end{pmatrix} \mathrm{E}[p_b^i(1-p_b)^{\Lambda-i+m}] \\
&= \sum_{i=\lambda}^{\Lambda} \begin{pmatrix} \Lambda \\ i \end{pmatrix} \alpha_{\Lambda+m,i} \\
P(E_B) &= 1 - P(C_G) - P(B). \qquad (5.16)
\end{aligned}$$

The input to the Reed-Solomon decoder is one of three events: correct symbol, symbol in error, or an erasure. We will always assume that the Reed-Solomon decoder will either fail to decode or will output the correct codeword.

Thus in order to calculate symbol error probability, we need to specify the decoding procedure in the case that the error-and-erasure correcting capability of the code is exceeded. In this section we analyze two different configurations for the decoder shown in Figure 5.20 and Figure 5.21. The first configuration (Decoder A) consists of one Reed-Solomon decoder which corrects errors and erasures and a

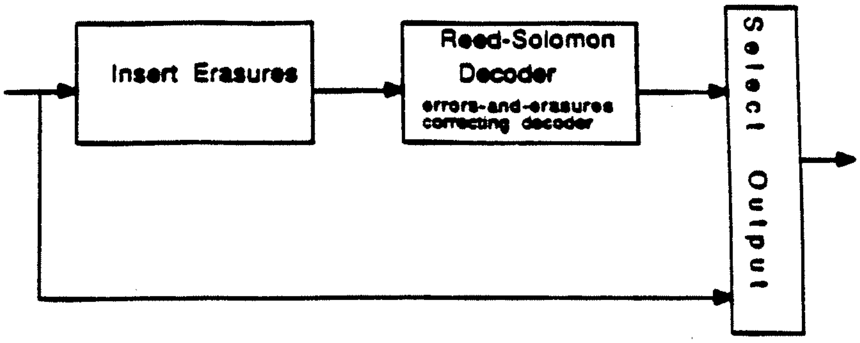

Figure 5.21: Decoder configuration (A).

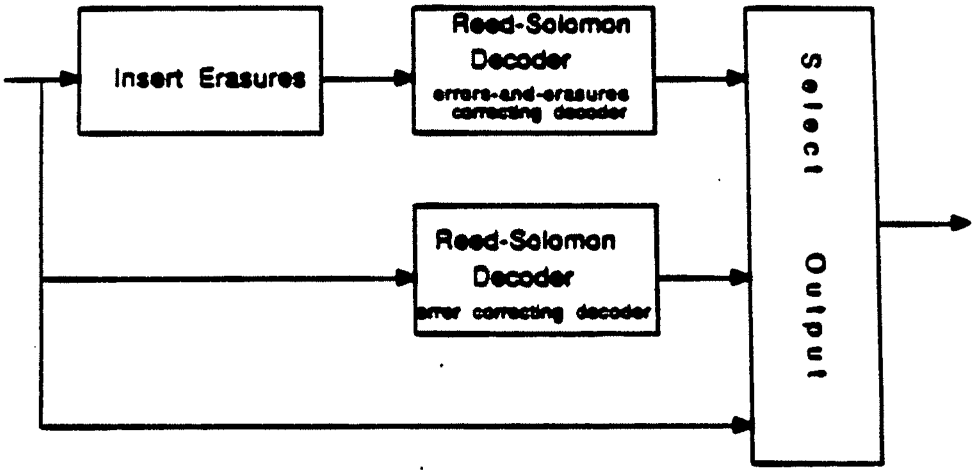

Figure 5.22: Decoder configuration (B).

selector to choose the information portion of the codeword if the decoder fails.

If the error-and-erasure correcting capability of the code is exceeded the decoder fails, and the receiver will output the information symbols of the received vector (we assume systematic encoding). The second configuration (Decoder B) consists of two Reed-Solomon decoders in parallel and a selector to choose the output of a decoder except when the two decoders fail. The first one is used for errors-and-erasures correcting decoding. The second decoder is used to correct errors only. There will be occasions when the first decoder fails to decode and the second one outputs the correct codeword. In case both outputs codewords, they have to be the same one, by assumption. If both decoders fail to decode, the receiver will output the information symbols of the received vector. Obviously we expect the second configuration to perform better than the first one, the second being more complex.

We now calculate exact expressions for the probability of symbol error for the two decoders described above when using bounded-distance decoding. The baseline performance measure that we use for comparisons is the symbol error probability with no test bits (i.e., $\Lambda = 0$) used for side information extraction, denoted as P_0. The output symbol error probability for an (n,k) code error correcting decoder when the input average symbol error probability is P_s code is given by

$$P_0 = \sum_{k=t+1}^{n} \binom{n}{k} \frac{k}{n} P_s^k (1-P_s)^{n-k} \quad .$$

where $t = \lfloor \frac{n-k}{2} \rfloor$ is the error correcting capability of the code and where it is assumed that the decoder fails only when it does not decode the correct codeword and outputs the information symbols of the received vector. Using our terminology

P_s is given by

$$\begin{aligned} P_s &= 1 - \mathrm{E}\{(1 - p_b(r))^m\} \\ &= 1 - \alpha_{m,0} \end{aligned} \tag{5.17}$$

where $p_b(r) = \frac{1}{2} e^{-\frac{E_b r^2 r_c}{2N_0}}$, and $r_c = \frac{k}{n}$ is the dimensionless code rate.

Performance of decoder A

Our goal now is to evaluate the probability of symbol error for the first decoder configuration. First we define several events which will facilitate calculating the probability of symbol error.

$E_{s,o}$ = Event the first output symbol of the decision device is in error.

$E_{s,i}$ = Event the first input symbol to the Reed-Solomon decoder is in error.

$E_{s,ic}$ = Event the first input symbol to the Reed-Solomon decoder is correct.

$E_{s,i?}$= Event the first input symbol to the decoder has been declared bad (i.e., erased).

E_h = Event there are h erasures in the codeword received.

E_e = Event e errors have occurred in the received vector.

Then due to symmetry, the probability of symbol error is the same as $P(E_{s,o})$, which is calculated as follows.

$$P(E_{s,o}) \doteq \sum_{h} \sum_{e:2e+h \geq d_{min}} P(E_{s,o} \mid E_h, E_e) P(E_h, E_e) \tag{5.18}$$

where $2e + h \geq d_{min}$ is the condition for the decoder to fail. In (5.18) $P(E_h, E_e)$ is the probability of having h erasures and e errors in a received vector of length n.

Since errors and erasures are independent this is given by

$$P(E_h, E_e) \doteq \begin{pmatrix} n \\ h, e \end{pmatrix} P(EG)^e P(B)^h (1 - P(EG) - P(B))^{n-e-h} \tag{5.19}$$

where

$$\begin{pmatrix} n \\ h, e \end{pmatrix} = \begin{pmatrix} n \\ h \end{pmatrix} \begin{pmatrix} n-h \\ e \end{pmatrix}.$$

$P(E_{s,o} \mid E_h, E_e)$ can be calculated as shown below.

$$\begin{aligned} P(E_{s,o} \mid E_h, E_e) &= P(E_{s,o} \mid E_h, E_e, E_{s,i}) P(E_{s,i} \mid E_h, E_e) \\ &\quad + P(E_{s,o} \mid E_h, E_e, E_{s,i?}) P(E_{s,i?} \mid E_h, E_e) \\ &\quad + P(E_{s,o} \mid E_h, E_e, E_{s,ic}) P(E_{s,ic} \mid E_h, E_e). \end{aligned} \tag{5.20}$$

The last term on the right hand side is obviously equal to 0. Moreover, $P(E_{s,o} \mid E_h, E_e, E_{s,i}) = 1$ for $2e + h \geq d_{min}$. It can be shown that the following hold

$$P(E_{s,i} \mid E_h, E_e) = \frac{e}{n},$$

$$P(E_{s,i?} \mid E_h, E_e) = \frac{h}{n},$$

and

$$\begin{aligned} P(E_{s,o} \mid E_h, E_e, E_{s,i?}) &= P(E_{s,o} \mid E_{s,i?} \bigcap \text{decoder fails}) \\ &= \frac{P(E_{s,o} \bigcap E_{s,i?} \mid \text{decoder fails})}{P(E_{s,i?} \mid \text{decoder fails})} \\ &= \frac{P(EB)}{P(B)} \end{aligned} \tag{5.21}$$

Combining all the above we get

$$P(E_{s,o} \mid E_h, E_e) = \frac{e + h\frac{P(EB)}{P(B)}}{n}. \tag{5.22}$$

Substituting (5.19)-(5.22) in (5.18) we get an expression for the symbol error probability for decoder (A)

$$P(E_{s,o}) = \sum_{h} \sum_{e:2e+h\geq d_{min}} \left(\frac{e + h\frac{P(EB)}{P(B)}}{n}\right) \left(\begin{array}{c} n \\ h,e \end{array}\right)$$
$$P(EG)^{e}P(B)^{h}(1-P(EG)-P(B))^{n-e-h}. \qquad (5.23)$$

For this decoder with Reed-Solomon coding we have found that λ_{opt} is equal to 1 for all cases of interest. That is, the best strategy for deciding whether a symbol is reliable or not is to decide in favor of the received symbol only if the received test pattern is correct. A typical behaviour for different thresholds λ (and fixed $\Lambda = 3$) is shown in Figure 5.22.

When using the optimal threshold value λ_{opt} there is a substantial improvement in performance due to the use of test bits in the decoding process. This is shown for different rates in Figure 5.23-Figure 5.26. One first thing to notice is that for all $\frac{E_b}{N_0}$ the performance for any $\Lambda > 0$ is better than the $\Lambda = 0$, no test bits, case. This is unlike the repetition coded system discussed earlier. For instance, for (32,5) Reed-Solomon code and $\Lambda = 3$, there is more than 1.8 dB improvement in $\frac{E_b}{N_0}$ for probability of symbol error less than 10^{-4}. For a larger rate the improvement in performance, although substantial, is less; e.g. for (32,10) Reed-Solomon code the improvement is approximately 1.3 dB for symbol error probability 10^{-4}. However, the overall performance for the $r_c = \frac{10}{32}$ case is better than that of $r_c = \frac{5}{32}$. It seems that some rate close in value to $r = \frac{10}{32}$ is the best rate to use in the communication system considered when it is desired to have $P_s = 10^{-4}$ or less. The value for an optimum rate was expected and close to the value predicted by Stark in [31]. The performance with the same rates described above but with a larger minimum

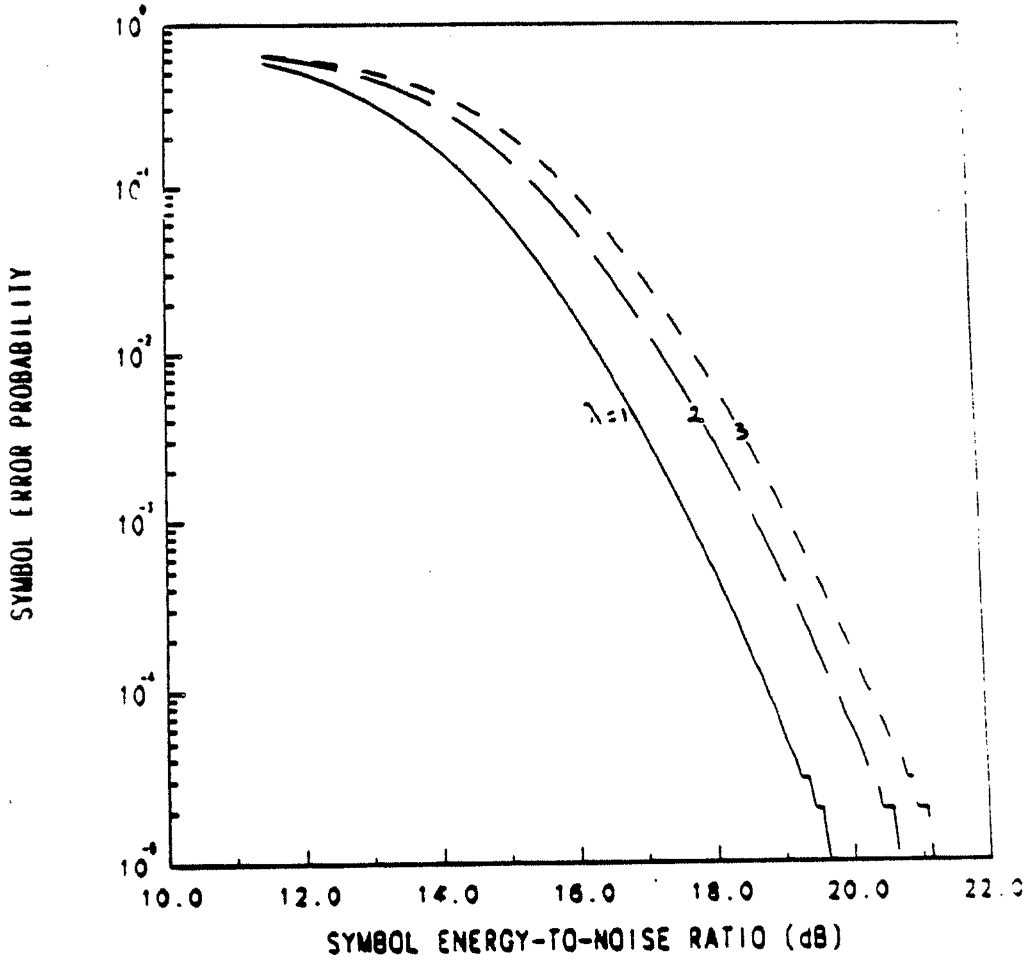

Figure 5.23: Performance of (32,5) Reed-Solomon code for $\Lambda = 3$ and for different thresholds λ.

Hamming distance d_{min} is shown in Figures 5.27-5.30. The improvement when using side information is slightly less in this case than the previous ones and there is approximately 1 dB gain for $\Lambda = 2$ and $P_s = 10^{-4}$. Obviously, since we are using a larger minimum distance we have a better overall performance in the latter case.

We point out another important conclusion from the figures. The improvement in performance for each increment of Λ is monotonically decreasing. For a fixed rate and required symbol error probability, the required $\frac{E_b}{N_0}$ approaches an asymptotic value for large Λ. This is demonstrated in Figure 5.31.

Performance of decoder B

For this system we have two decoders in parallel: one performs errors-and-erasures correcting decoding, and the second performs error correction decoding alone. Pursley and Stark have considered a similar system in [26] but with parallel error-correction and erasure correction decoding. That system was proven to have good performance in partial-band Gaussian interference and for most values of fraction jammed. Also Castor and Stark proposed similar systems [6] - [7] with various methods for erasure criterion.

We calculate an expression for the symbol error probability which is the same as the probability of the first output symbol being in error, $P(E_{so})$. The first output will be in error if and only if both decoders fail to decode and the first input symbol to the "insert erasure" block is in error; i.e.,

$$\begin{aligned} P(E_{so}) &= P\{\text{decoder1 fails} \bigcap \text{decoder2 fails} \bigcap (E_{si} \bigcup E_{si,i})\} \\ &= P\{\text{decoder1 fails} \bigcap \text{decoder2 fails} \mid E_{si}\} P\{E_{si}\} \\ &\quad + P\{\text{decoder1 fails} \bigcap \text{decoder2 fails} \mid E_{si,i}\} P\{E_{si,i}\} \end{aligned}$$

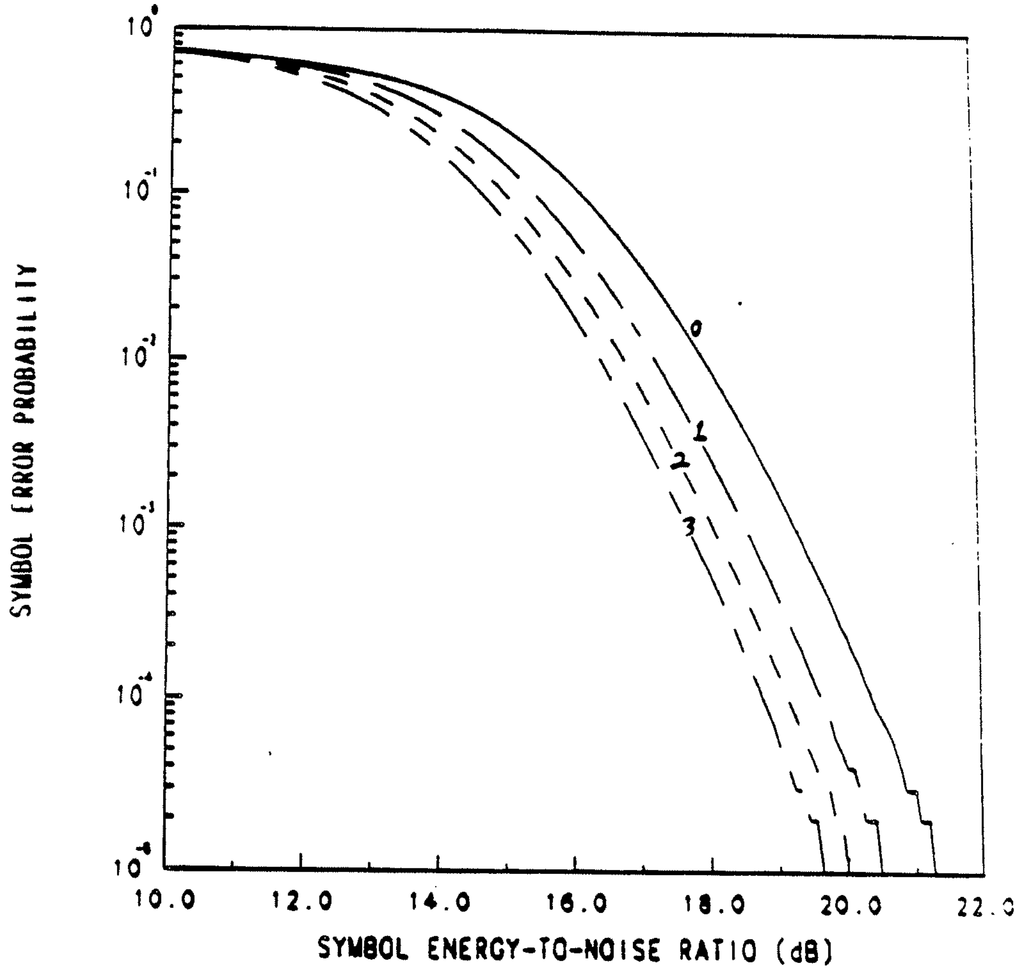

Figure 5.24: Performance of (32,5) Reed-Solomon code for $\Lambda = 3$ and for different thresholds λ.

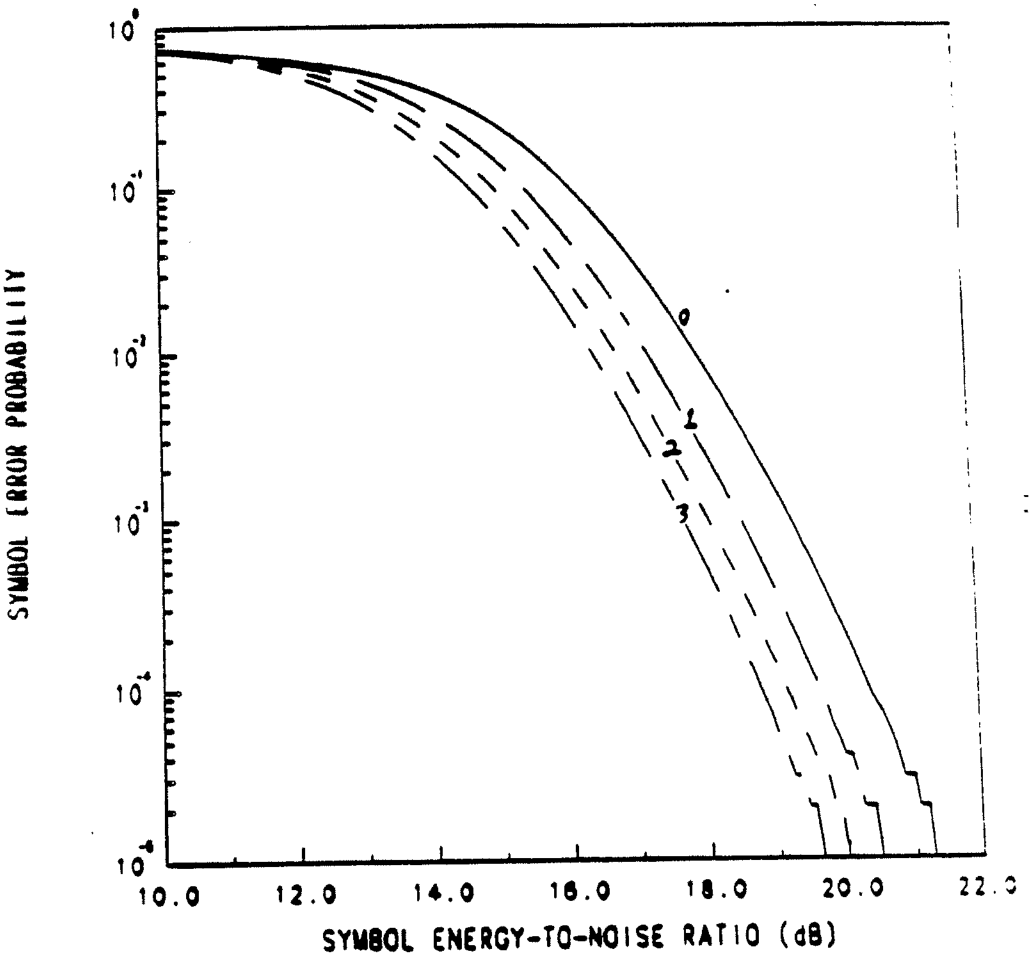

Figure 5.25: Performance of (32,5) Reed-Solomon code for $\Lambda = 0, 1, 2, 3$ and for optimal λ.

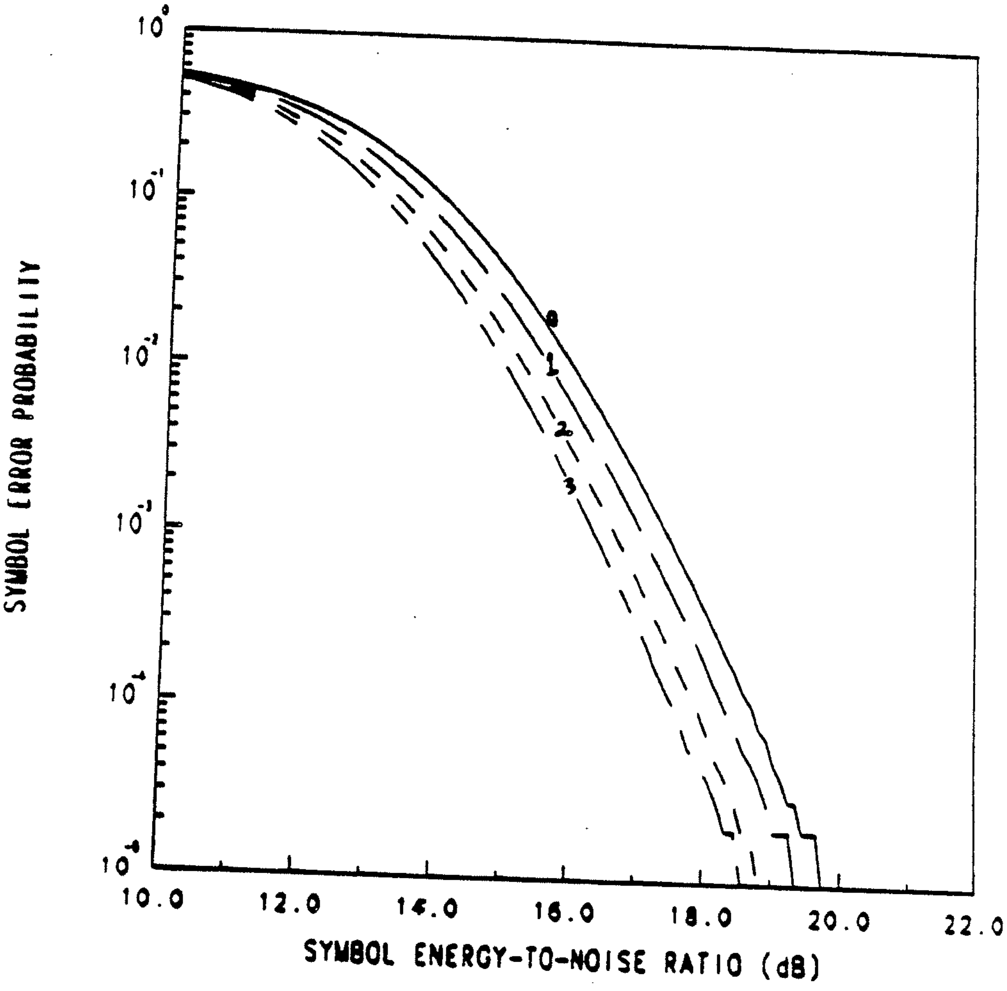

Figure 5.26: Performance of (32,10) Reed-Solomon code for $\Lambda = 0, 1, 2, 3$ and for optimal λ.

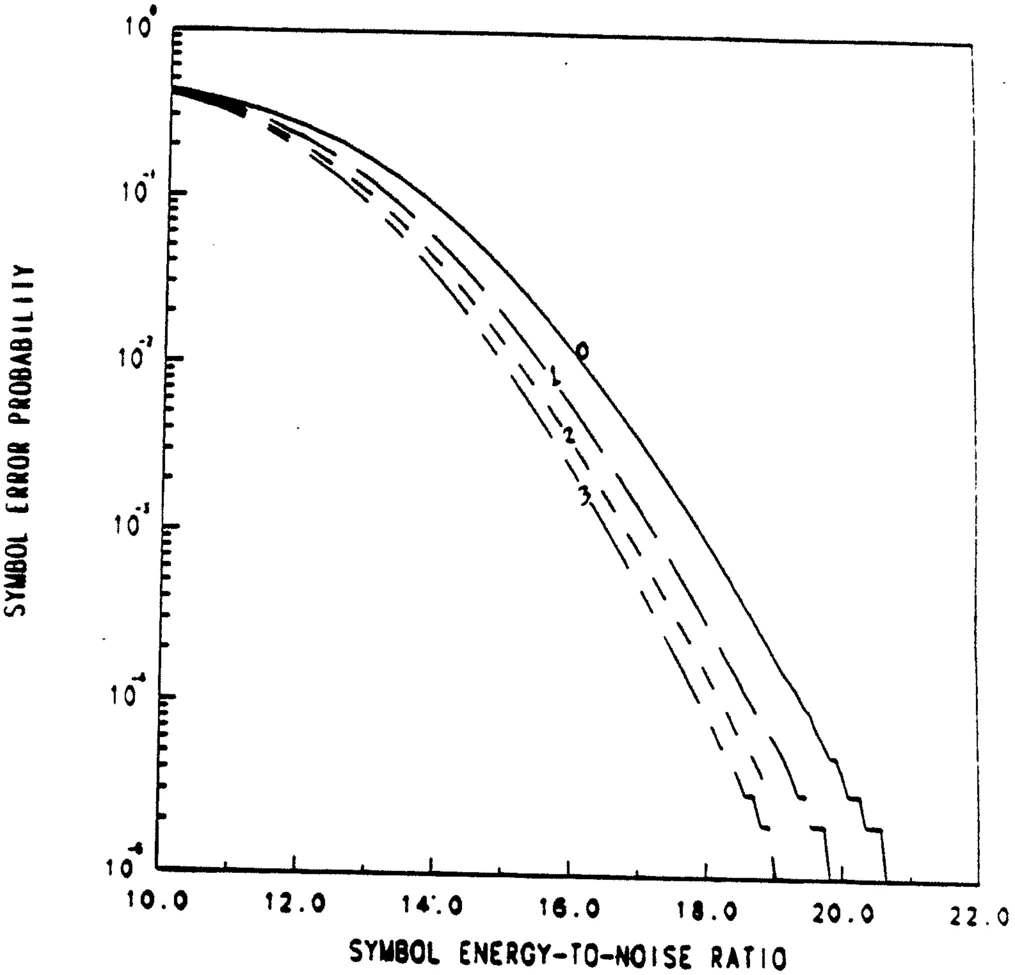

Figure 5.27: Performance of (32,15) Reed-Solomon code for $\Lambda = 0, 1, 2, 3$ and for optimal λ.

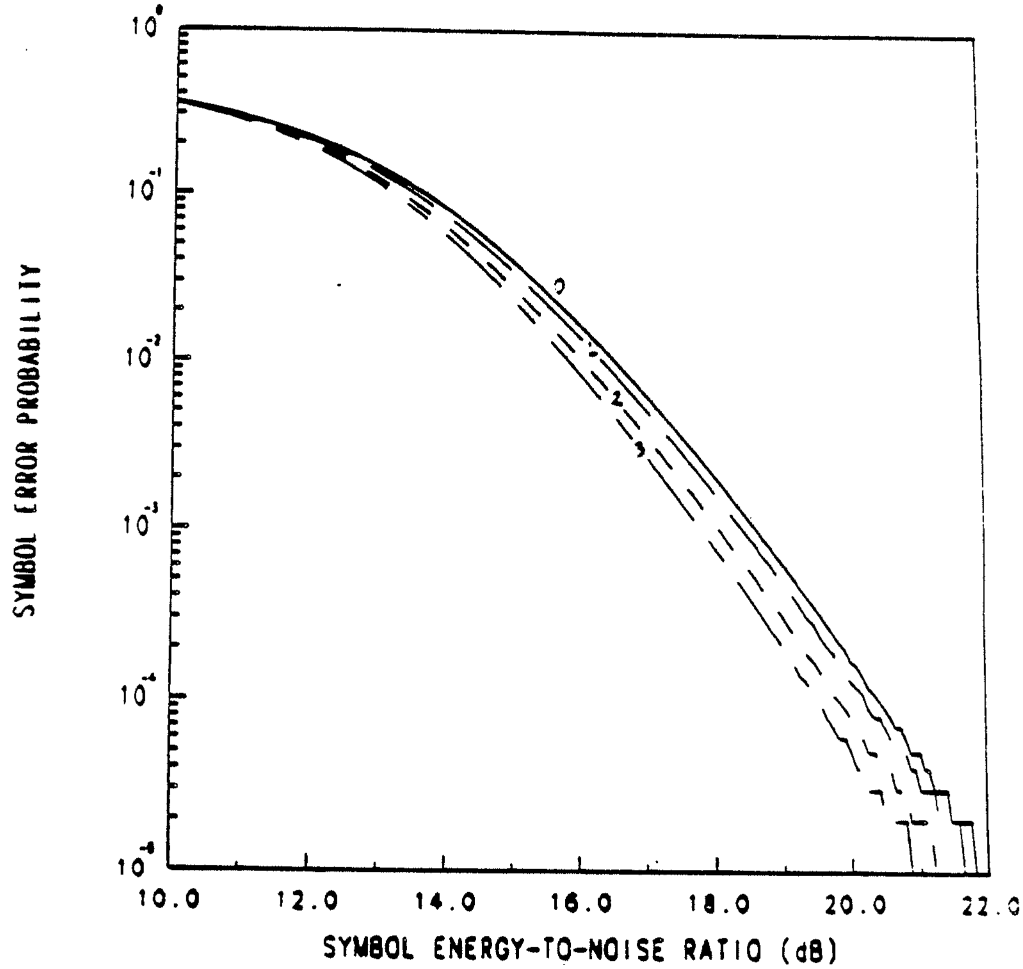

Figure 5.28: Performance of (32,20) Reed-Solomon code for $\Lambda = 0, 1, 2, 3$ and for optimal λ.

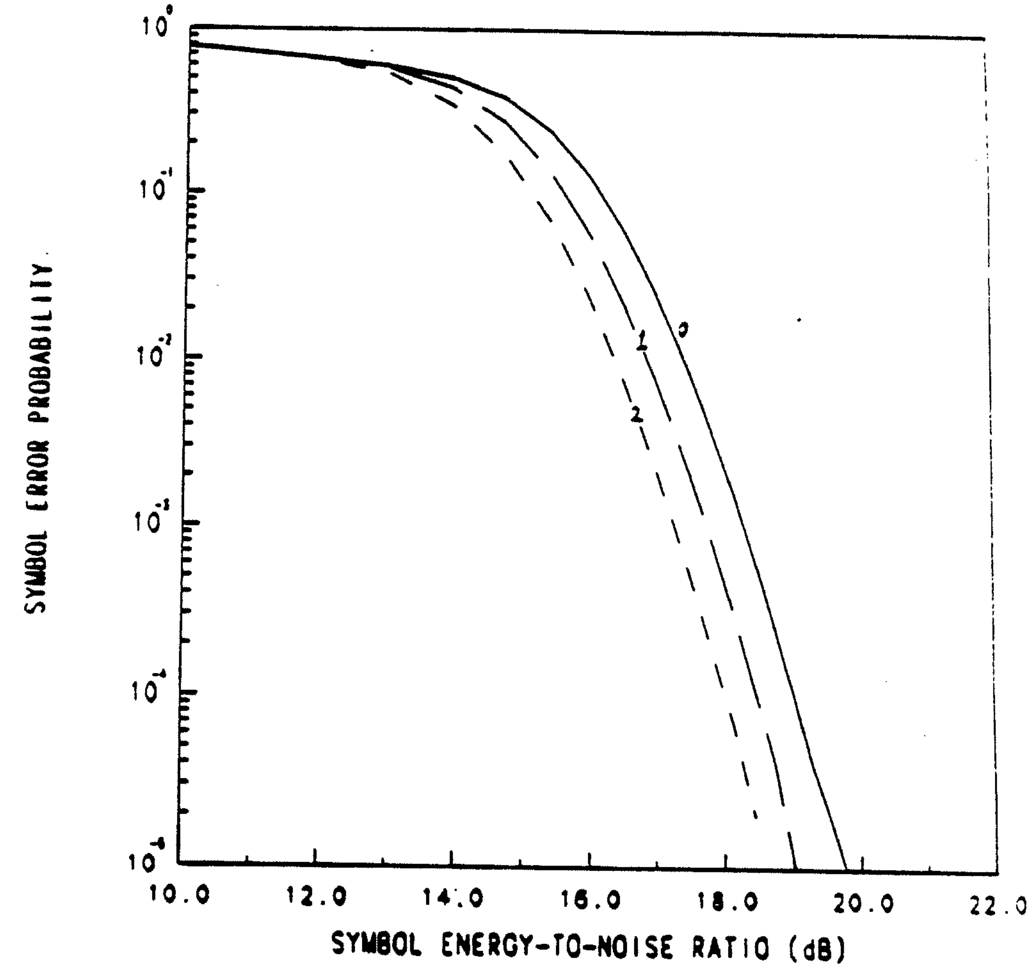

Figure 5.29: Performance of (64,10) Reed-Solomon code for $\Lambda = 0, 1, 2$ and for optimal λ.

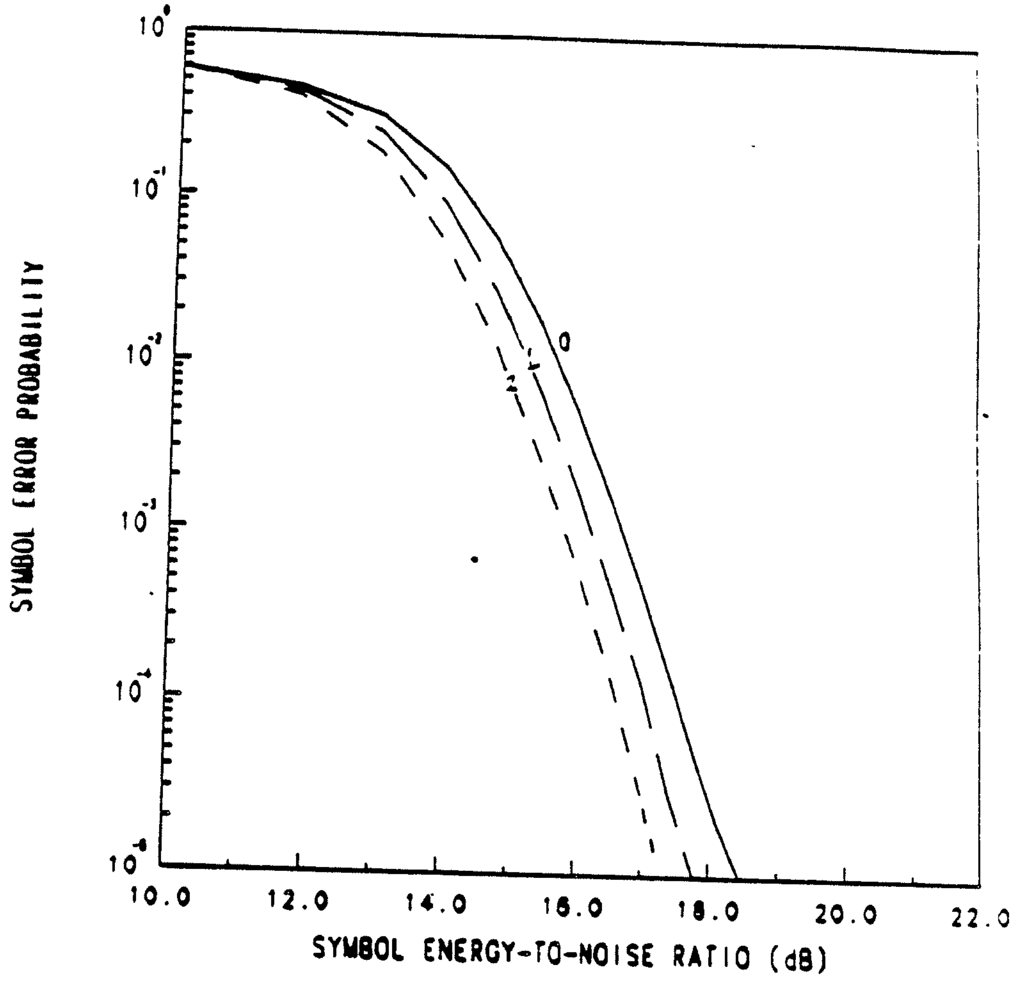

Figure 5.30: Performance of (64,20) Reed-Solomon code for $\Lambda = 0, 1, 2$ and for optimal λ.

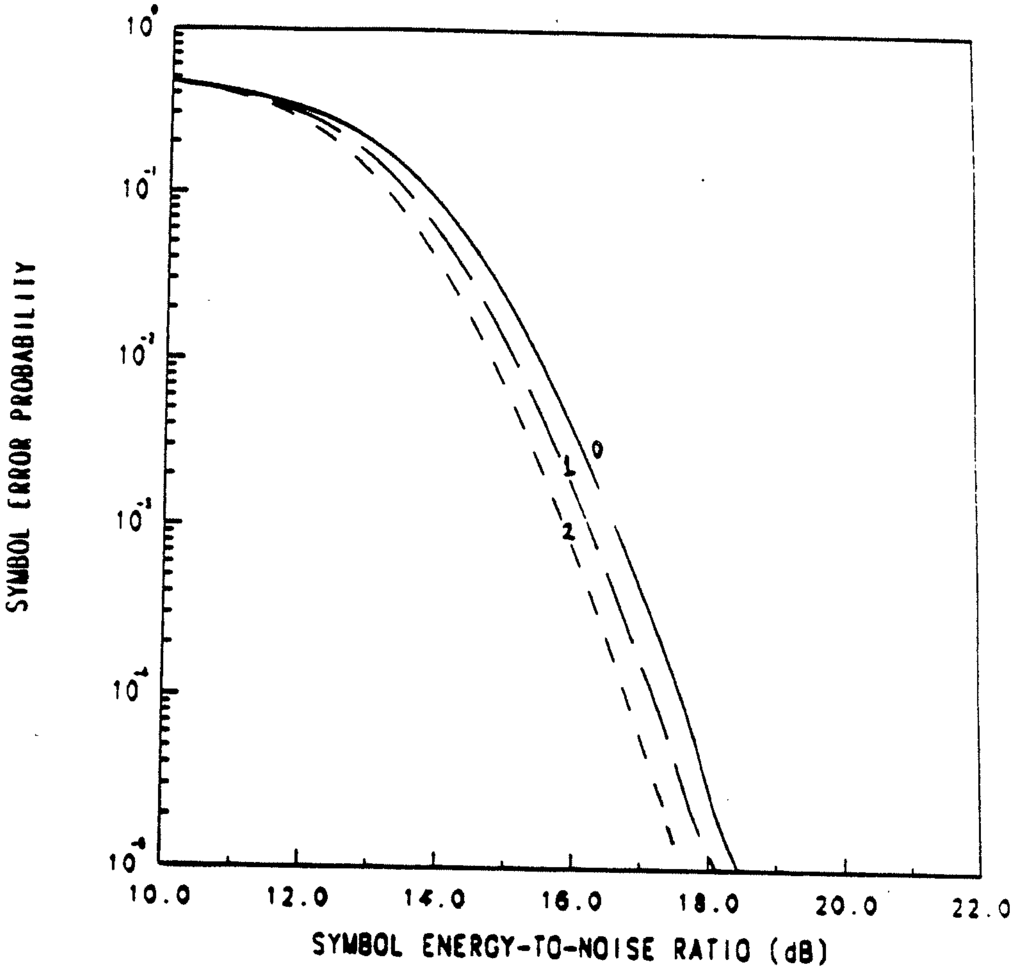

Figure 5.31: Performance of (64,30) Reed-Solomon code for $\Lambda = 0, 1, 2$ and for optimal λ.

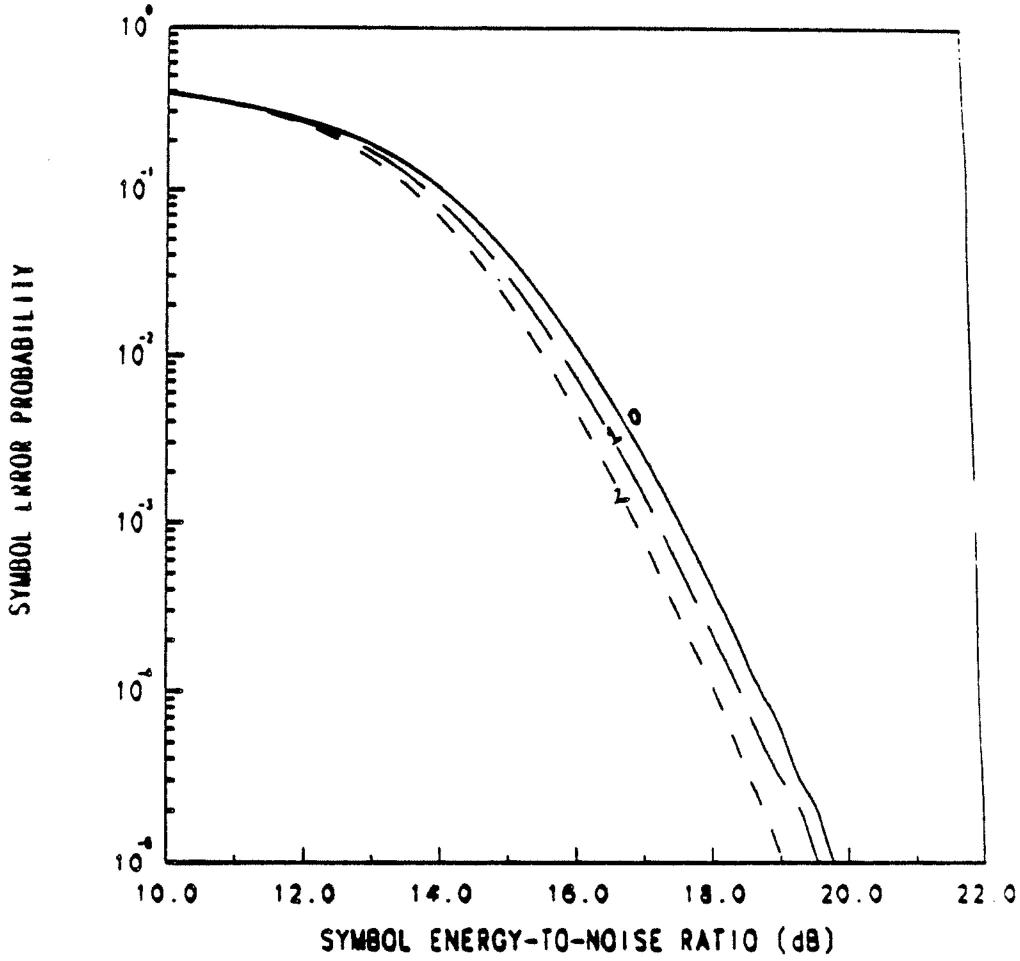

Figure 5.32: Performance of (64,40) Reed-Solomon code for $\Lambda = 0, 1, 2$ and for optimal λ.

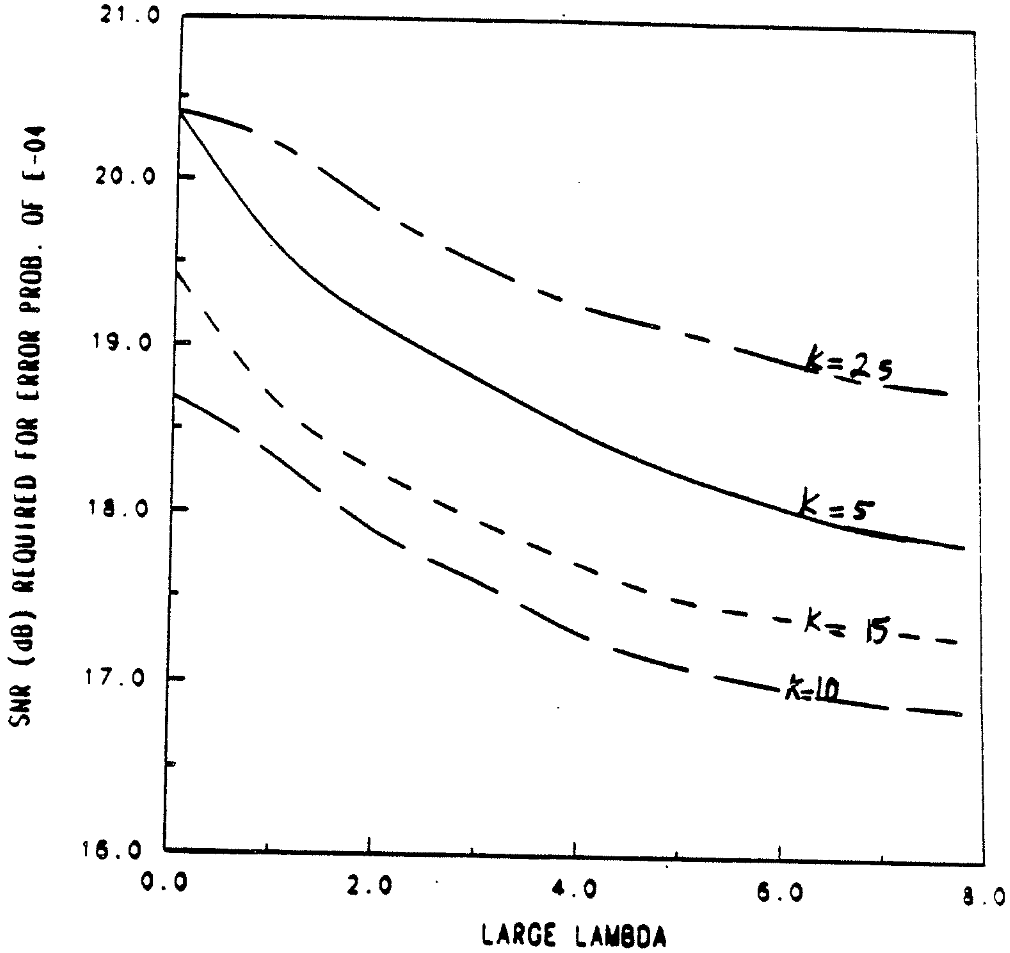

Figure 5.33: The signal to noise ratio required to achieve $P_s = 10^{-4}$ as a function of Λ (for (32,5) Reed-Solomon code).

Event	Input to Decoder1	Input to Decoder2	Event Probability
E_1	correct	correct	$P(CG)$
E_2	correct	erased	$P(CB)$
E_3	error	error	$P(EG)$
E_4	error	erased	$P(EB)$

Table 5.1: Event description and the corresponding probabilities.

$$= P\{2\#(\text{errors+1})+\#\text{erasures} \geq d_{min} \bigcap 2\#(\text{errors+1}) \geq d_{min} \mid E_{si}\}P(EG)$$
$$+P\{2\#(\text{errors+1})+\#\text{erasures} \geq d_{min} \bigcap 2\#(\text{errors+1}) \geq d_{min} \mid E_{si,i}\}P(EB).$$

To evaluate the above probabilities it is helpful to make use of the notation in Table 5.1.

Also, let S_n = # of symbols in the last $n-1$ symbols satisfying E_n. Then

$$P\{2(\#(\text{errors+1})+\#\text{erasures} \geq d_{min} \bigcap 2\#(\text{errors+1}) \geq d_{min} \mid E_{si}\} =$$
$$= P\{2(S_3+1)+S_2+S_4 \geq d_{min}, 2(S_3+S_4+1) \geq d_{min}\}$$
$$= \sum_i \sum_j \sum_k \begin{pmatrix} n-1 \\ i,j,k \end{pmatrix} P(CB)^i P(EG)^j P(EB)^k P(CG)^{n-1-i-j-k}$$
$$\text{such that } 2(j+1)+i+k \geq d_{min} \text{ and } 2(j+k+1) \geq d_{min}. \tag{5.24}$$

Similarly,

$$P\{2\#\text{errors}+\#\text{erasures} \geq d_{min} \bigcap 2\#(\text{errors+1}) \geq d_{min} \mid E_{si,i}\} =$$
$$= P\{2S_3+S_2+S_4+1 \geq d_{min}, 2(S_3+S_4) \geq d_{min}\} \tag{5.25}$$

which is the same as (5.24) but with the constraints

$$2j + i + k \geq d_{min}$$

$$2(j + k + 1) \geq d_{min}.$$

Thus we have an expression for the probability of symbol error.

The symbol error probability performance of decoder (B) is compared with that of decoder (A). The comparisons are shown in Figure 32-Figure 36. The performance of the two decoders was found to be very close. The improvement when using decoder (B) as compared with that of (A) is about 0.1 dB, which is very small.

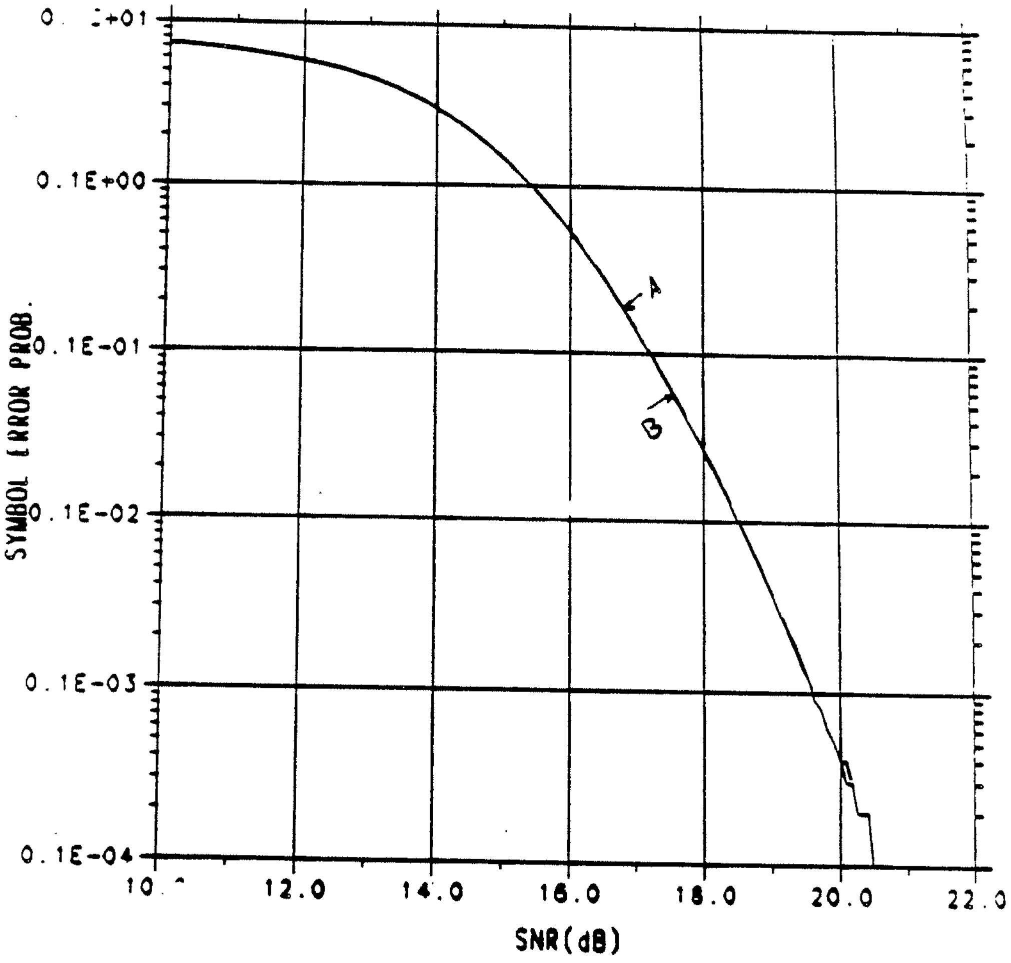

Figure 5.34: Comparing Decoders (A) and (B) for (32,5) Reed-Solomon code and $\Lambda = 1$.

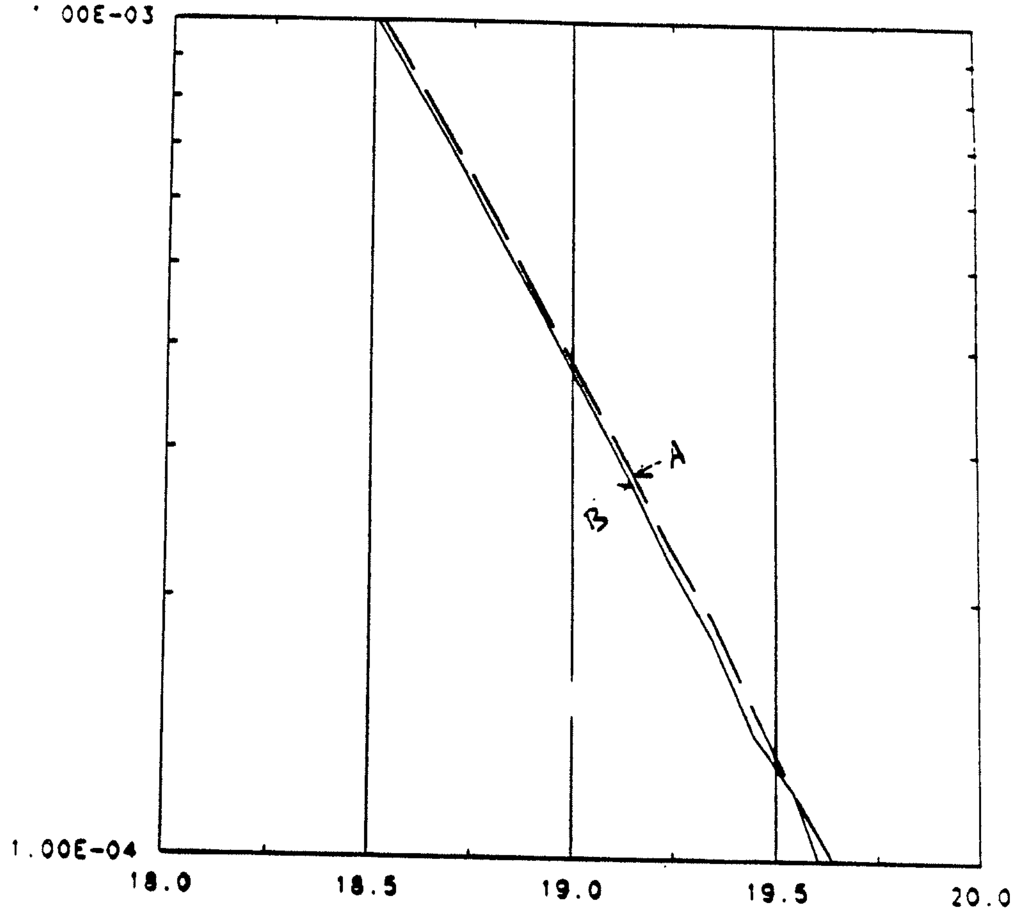

Figure 5.35: Enlarged version for Figure 29.

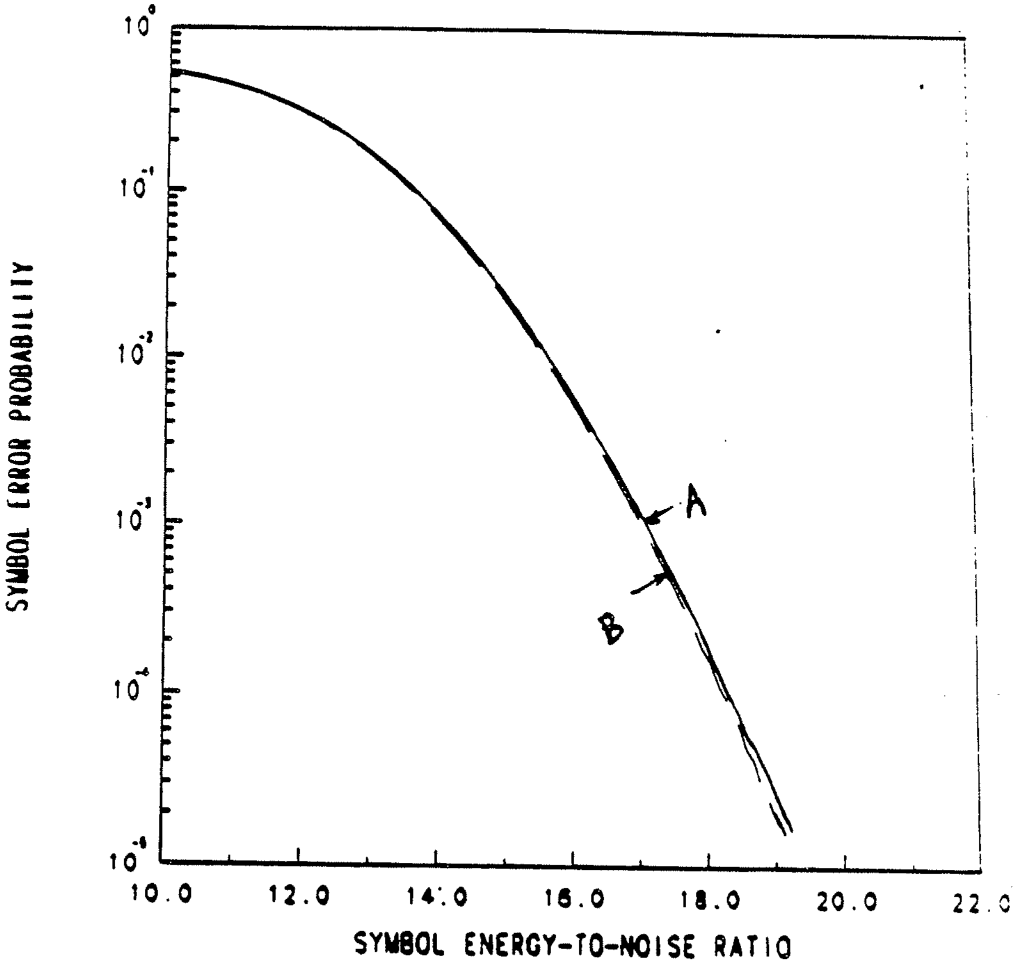

Figure 5.36: Comparing Decoders (A) and (B) for (64,10) Reed-Solomon code and $\Lambda = 1$.

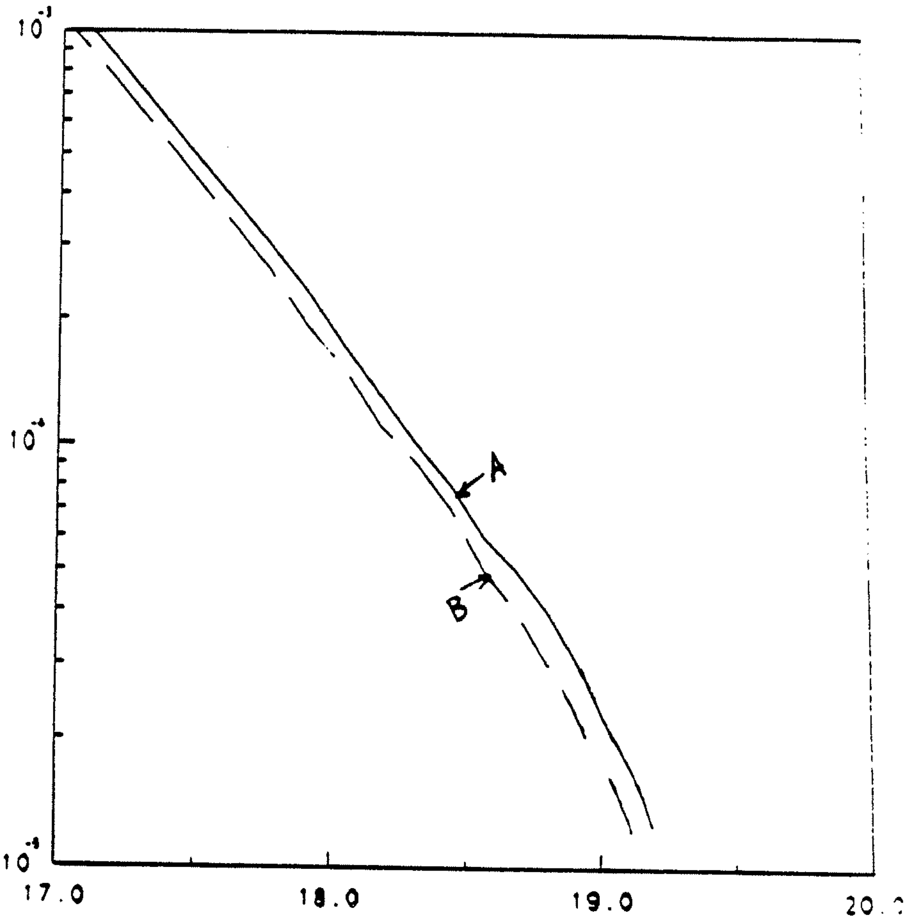

Figure 5.37: Enlarged version for Figure 31.

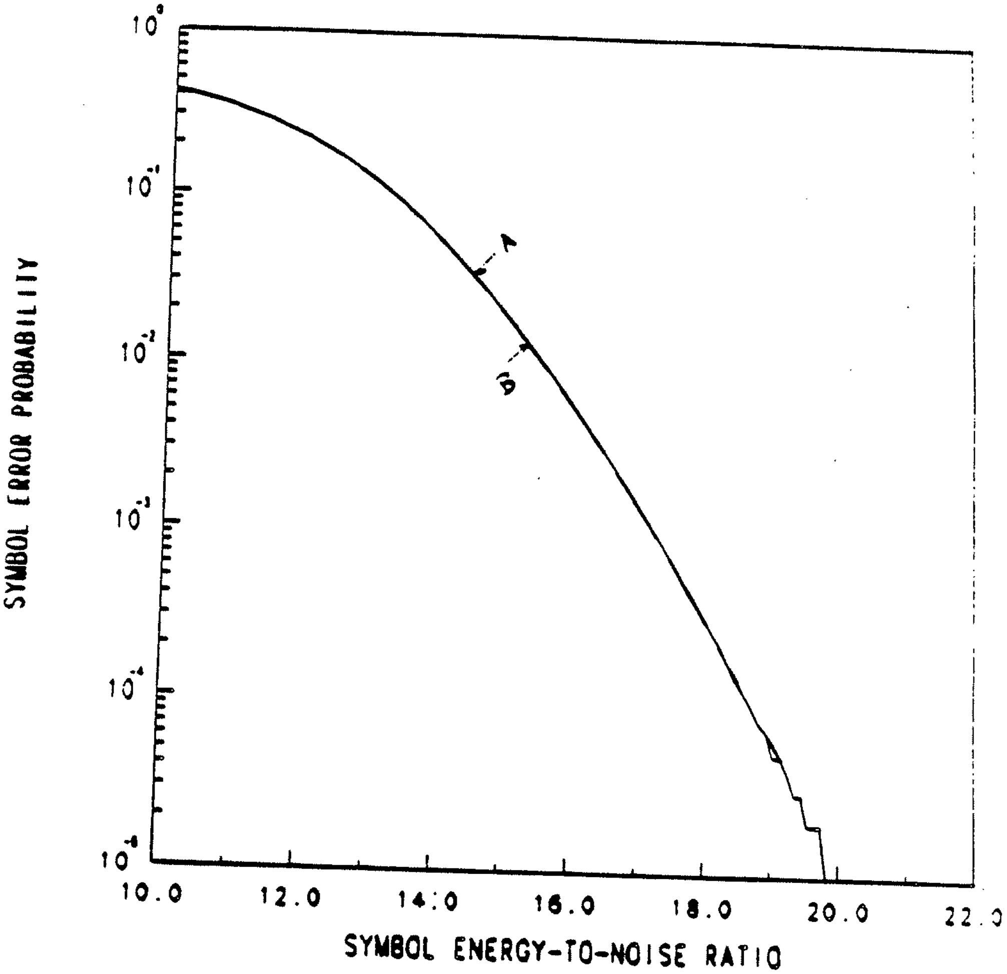

Figure 5.38: Comparing Decoders (A) and (B) for (64,15) Reed-Solomon code and $\Lambda = 1$.

CHAPTER VI

A CONCATENATED CODED SYSTEM IN RAYLEIGH FADING

6.1 Introduction

In the previous Chapter we demonstrated how transmitting a known pattern of test bits can be used at the decoder to significantly enhance the performance of a Reed-Solomon coded system in the presence of fading. In this chapter the idea of using such redundant bits to learn about the state of each hop is used to develop a smarter way of introducing redundancy into each hop. The idea is to further encode the interleaved Reed-Solomon code symbols in each hop. Thus we have a concatenated coded system with the outer-code being interleaved, and each inner-codeword is transmitted over a fixed channel (we assumed that the fade is slow enough and is constant over one dwell interval). The concatenated coded system considered uses the inner code to correct and detect errors. That is the inner code can correct e errors and detect f errors ($e \leq f$) provided $e + f < d_{1H}$, where again d_{1H} is the minimum Hamming distance of the inner code. When errors are only detected, every symbol of the inner code is erased. There are, however, errors that are not detected nor corrected by the inner code. Thus the outer Reed-Solomon code correct errors and erasures.

For the system considered, the performance criteria is taken to be the packet error probability where each packet consists of a number of codewords. This performance criteria is motivated by the application of the system considered in frequency-hop packet radio networks [25] - [24].

The channel model and system model is the same as in Section 5.2 except for the following. Consider transmitting a packet that consists of L extended Reed-Solomon (N, K) codewords with code symbols belonging to $GF(2^m)$. The transmission of each symbol is done by m uses of a BSC channel (i.e., 1 symbol $\equiv m$ bits) with cross over probability $p_b(r)$ given by (5.7), where r is the fading random variable. The outer code is interleaved to depth N; thus we have L symbols from different codewords that need to be transmitted over one hop.

6.2 Performance Analysis

We first consider the case when the inner encoder is binary. Then we investigate a specific class of nonbinary inner codes.

The L symbols consist of mL bits which are encoded using a binary (n, mL) code with a minimum Hamming distance d. We make the assumption that if the inner codeword is decoded incorrectly, then all the corresponding Reed-Solomon Code symbols are incorrect. This implies that the packet error probability P_L is the same as the outer codeword error probability P_w (obviously $P_L < P_W$).

To calculate P_w we need to calculate the following probabilities of events that are relevant to the inner code.

$$P_{CD} \triangleq P_r \; \{\text{correct decoding}\} \quad ;$$

$$P_{ICD} \triangleq P_r \text{ \{incorrect decoding\}} \quad ;$$

$$P_{ED} \triangleq P_r \text{ \{error detection\}} \quad .$$

The inner decoder is assumed to be a bounded-distance decoder that corrects all error patterns of Hamming weight t or less, where $t \leq \lfloor \frac{d-1}{2} \rfloor$. Also, error detection results in erasing the entire hop. Then

$$P_w = 1 - P_c \quad ,$$

where

$$\begin{aligned} P_c &= \sum_{e} \sum_{\tau:2e+\tau \leq N-K} \binom{N}{e,\tau} P_{ICD}^{e} \, P_{ED}^{\tau} \, P_{CD}^{N-e-\tau} \\ &= \sum_{\tau=0}^{N-K} \sum_{e=0}^{\lfloor \frac{N-K-\tau}{2} \rfloor} \binom{N}{e,\tau} P_{ICD}^{e} \, P_{ED}^{\tau} \, P_{CD}^{N-e-\tau} \quad . \end{aligned}$$

Conditioned on the fade r, the inner code symbol errors are independent. Understanding that all the expectations below are with respect to the Rayleigh distributed random variable r, we have the following:

$$\begin{aligned} P_{CD} &= E\left\{ \sum_{i=0}^{t} \binom{n}{i} p_b(r)^i (1 - p_b(r))^{n-i} \right\} \\ &= \sum_{i=0}^{t} \binom{n}{i} E[p_b(r)^i (1 - p_b(r))^{n-i}] \\ &= \sum_{i=0}^{t} \binom{n}{i} \alpha_{n,i} \quad , \end{aligned}$$

where $\alpha_{n,k}$ is given by (5.8).

Clearly some weight-$(m-t)$ or less error patterns might be decoded into weight-m error patterns with a t-error-correcting decoder. Denoting the probability of the above event by $P_m(r)$ we have:

$$P_{ICD} = E\left\{ \sum_{m=d}^{n} A_m P_m(r) \right\}$$

$$= \sum_{m=d}^{n} A_m \;\; E[P_m(r)] \qquad (6.1)$$

where A_m = number of codewords with weight m. Also from Michelson and Levesque [22]

$$P_m(r) = \sum_{v=0}^{t} \sum_{r'=0}^{v} \binom{m}{v-r'} \binom{n-m}{r'} P(r)^{m-v+2r'}(1-P(r))(n-m+v-2r') \;\; ;$$

substituting for $P_m(r)$ in (6.1) we get an expression for P_{ICD}:

$$P_{ICD} = \sum_{m=d}^{n} A_m \sum_{v=0}^{t} \sum_{r'=0}^{v} \binom{m}{v-r'} \binom{n-m}{r'} \alpha_{n,m-v+2r'} \;\; .$$

Finally, $P_{ED} = 1 - P_{ICD} - P_{CD}$.

Thus, calculating the probabilities of interest for a linear binary inner-code can be carried out exactly if the weight distribution A_m is known. Often the weight distribution of a code is not readily determined, but if the weight distribution of the dual code is known, exact performance results can be obtained using the MacWilliams identity

$$B(z) = 2^{-mL}(1+z)^n A\left(\frac{1-2}{1+2}\right)$$

$A(z), B(z)$ are the weight-enumerator polynomial of the code and its dual, respectively. The weight distribution of most binary (and nonbinary) codes or their dual is not known, and exhaustive calculation laborious. However, very good approximations for A_m can be obtained based on the assumption that codeword weights follow a binomial distribution over their nonzero range. For an (n, mL) code for which the all-ones vector is a codeword, we have the following approximation

$$A_o \;\; = \;\; A_n = 1$$

$$\begin{aligned} A_i &= 0 \ , \qquad n-d+1 \leq i \leq n-1 \\ A_i &\simeq \binom{n}{i} \frac{2^{mL}-2}{\sum_{j=d}^{n-d} \binom{n}{j}} \approx 2^{-r} \binom{n}{i} \ , \ d \leq i \leq n-d \ , \end{aligned}$$

where r is the number of redundant bits $(n-mL)$. The approximation considered has greatest accuracy for high-rate codes, except possibly for weights in the tails of the distribution (near d and $n-d$). This approximation is used later to obtain numerical results for P_{ICD}.

The second case is that of using nonbinary concatenated codes. The procedure outlined above can be generalized to obtain results for linear nonbinary inner codes used with bounded-distance decoding. The codes considered are defined on b-ary alphabets where each inner code symbol is represented by b bits (i.e. we have $(n, \left(\frac{mL}{b}\right))$ code over $GF(2^b)$). Then, given r, the symbol error probability is

$$P_s(r) = 1 - (1 - p_b(r))^b \ .$$

The analytical expression for the probability of correct decoding is the same as before with substituting $P_s(r)$ for $p_b(r)$. That is

$$\begin{aligned} P_{CD} &= \sum_{i=0}^{t} \binom{n}{i} \ E[P_s(r)^i (1 - P_s(r))^{n-i}] \\ &= \sum_{i=0}^{t} \binom{n}{i} \sum_{j=0}^{i} \binom{i}{j} (-1)^{i-j} \alpha_{b(n-j),0} \ . \end{aligned}$$

However, the expression for P_{ICD} is more complicated since there are more involved error patterns. From Michelson and Levesque [22] we have

$$P_{ICD} = \sum_{m=d}^{n} A_m \sum_{s=0}^{t} \sum_{k=m-s}^{m+s} \sum_{r'=r_1}^{r_2} \binom{m}{m-s+r'} \binom{s-r'}{k-m+s-2r'}$$

$$\binom{n-m}{r'} (b-2)^{k-m+s-2r'} (b-1)^{r'} E[H(k;r)]$$

where

$$\begin{aligned} r_1 &= \max\{0, k-m\} \\ r_2 &= \left\lfloor \frac{k-m+s}{2} \right\rfloor \end{aligned}$$

and

$$H(k,r) = \left[\frac{P_s(r)}{b-1}\right]^k (1-P_s(r))^{n-k} \quad .$$

An approximation for A_m that is similar to the binomial approximation in the binary case is not known. The difficulty in knowing A_m for a nonbinary code can be avoided if we are able to use Reed-Solomon inner codes since the weight distributions of maximum distance separable (MDS) codes are known. To be able to use Reed-Solomon codes for inner codes we assume that $\frac{mL}{b} < 2^b$. In this case (see [5])

$$A_m = \binom{n}{m}(q-1)\sum_{j=0}^{m-d}(-1)^j\binom{m-1}{j}q^{m-d-j}$$

where $d = n - mL + 1$. Thus, in this case we can calculate P_w.

The formulae derived in this chapter for the packet error probability are evaluated numerically. The performance of the proposed concatenated coded system proved to be much better than using test bits in each dwell interval. For instance, Figure 6.1 compares the system analyzed in this chapter with the system analyzed in Chapter 5 (i.e. the test bits case). The outer code is a (32,10) Reed-Solomon code over $GF(2_6)$, and the inner code used is a (63,45) BCH binary code with minimum Hamming distance 8. Thus we have a packet which consists of 9 codewords. The packet error probability is calculated when the inner code is used to correct all error patterns of Hamming weight t or less and detect other error patterns. In the example above, $t = 2$ results in the best performance possible for all signal to

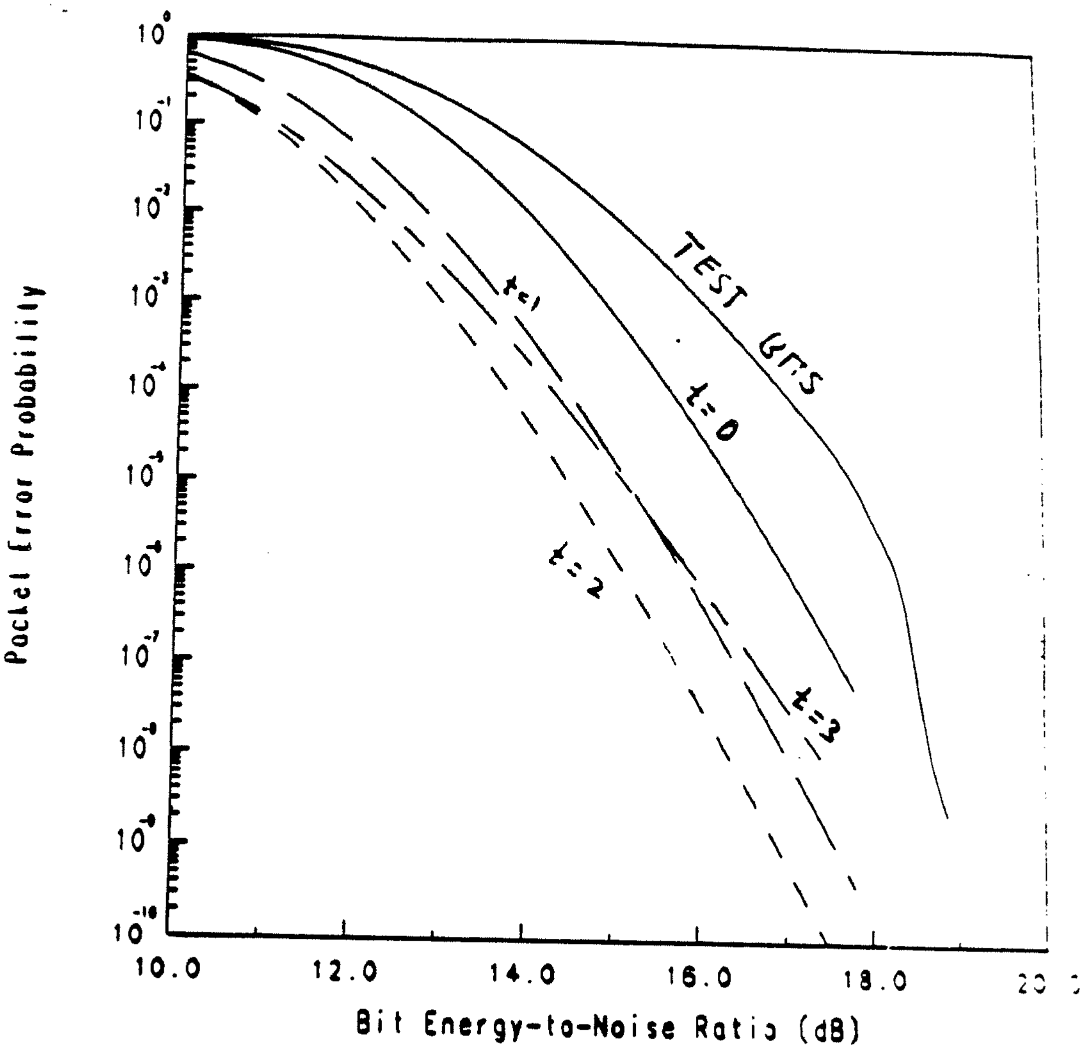

Figure 6.1: Packet Error Probability versus $\frac{E_b}{N_0}$ for a (63,45,8) BCH binary inner code.

noise ratios. Furthermore, the curve showing the performance for the test bits case is a lower bound, because in this example we have not normalized for the total energy used during a hop, for using the test bits. Notice that for probability of packet error 10^{-6} there is more than 3 dB improvement in $\frac{E_b}{N_0}$ over the test bits case. Table 6.1 shows the accuracy of the binomial approximation used to find the weight ditribution of the (63,45) code. The plot showing the behaviour of the system with test bits uses the same number of redundant bits (i.e., 18) as the BCH code.

Figure 6.2 shows the performance for (31,15) BCH binary code with Hamming distance 8. In this case we penalize the energy per code symbol for using the 16 test bits. The exact weight distribution is used in this case as shown in Table 6.2. In Figure 6.3 we do not normalize for using the test bits and the Figure shows similar results for (127,105) BCH inner code. In all the above examples $t = 2$ is the optimal value.

Finally, Figure 6.4 shows the performance of the concatenated coded system when the inner code is a (32,9) Reed-Solomon code. That is $L = 9$ in this case. For small values of $\frac{E_b}{N_0}$ the symbol error probability is high and compared to the (63,45) BCH code its performance is inferior. However, the performance of the nonbinary case becomes better for high signal to noise ratio.

6.3 Conclusions

In Chapters 5 and 6 we have investigated the performance of a coded slow-frequency-hopped spread spectrum communication system with the channel being dispersive with Rayleigh statistics. In particular we considered two methods for

i	$A_i(appr.)$	A_i
8	14774.	23877.
10	487544.7	423360.
12	1.0179370×10^7	1.0350×10^7
14	1.4262210×10^8	1.4255×10^8
16	1.3976890×10^9	1.3972×10^9
18	9.8751410×10^9	9.8751×10^9
.		
.		
.		
30	3.2836120×10^{12}	3.2832×10^{12}
32	3.4954400×10^{12}	3.4959×10^{12}
34	2.8973130×10^{12}	2.8969×10^{12}

Table 6.1: A Portion of the Weight Distribution and the Corresponding Approximation ((63,45) BCH Code).

i	A_i
8	465
12	8680
16	18259
20	5208
24	155

Table 6.2: The Weight Distribution of the (31,15) BCH code.

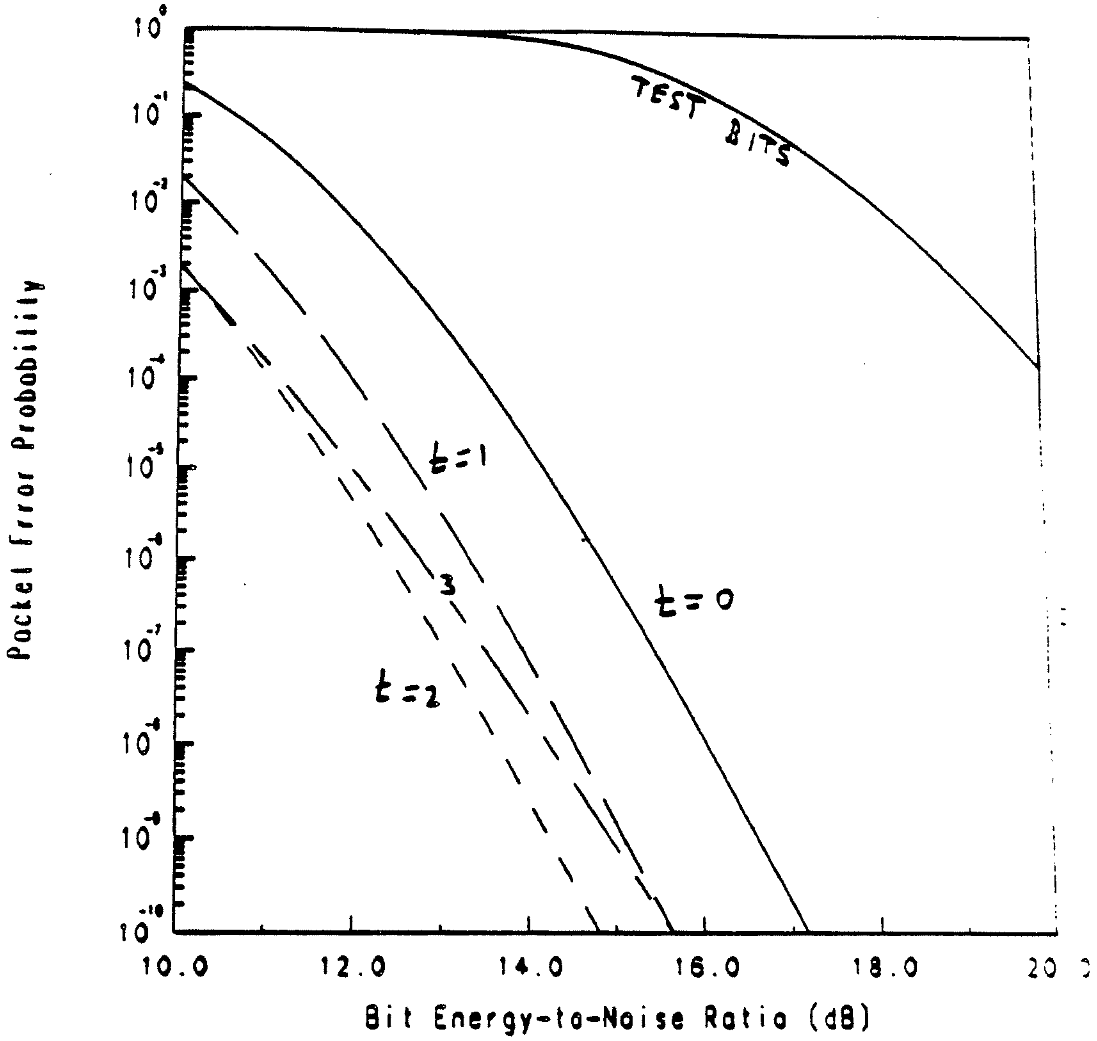

Figure 6.2: Packet Error Probability versus $\frac{E_b}{N_0}$ for a (31,15,8) BCH binary inner code.

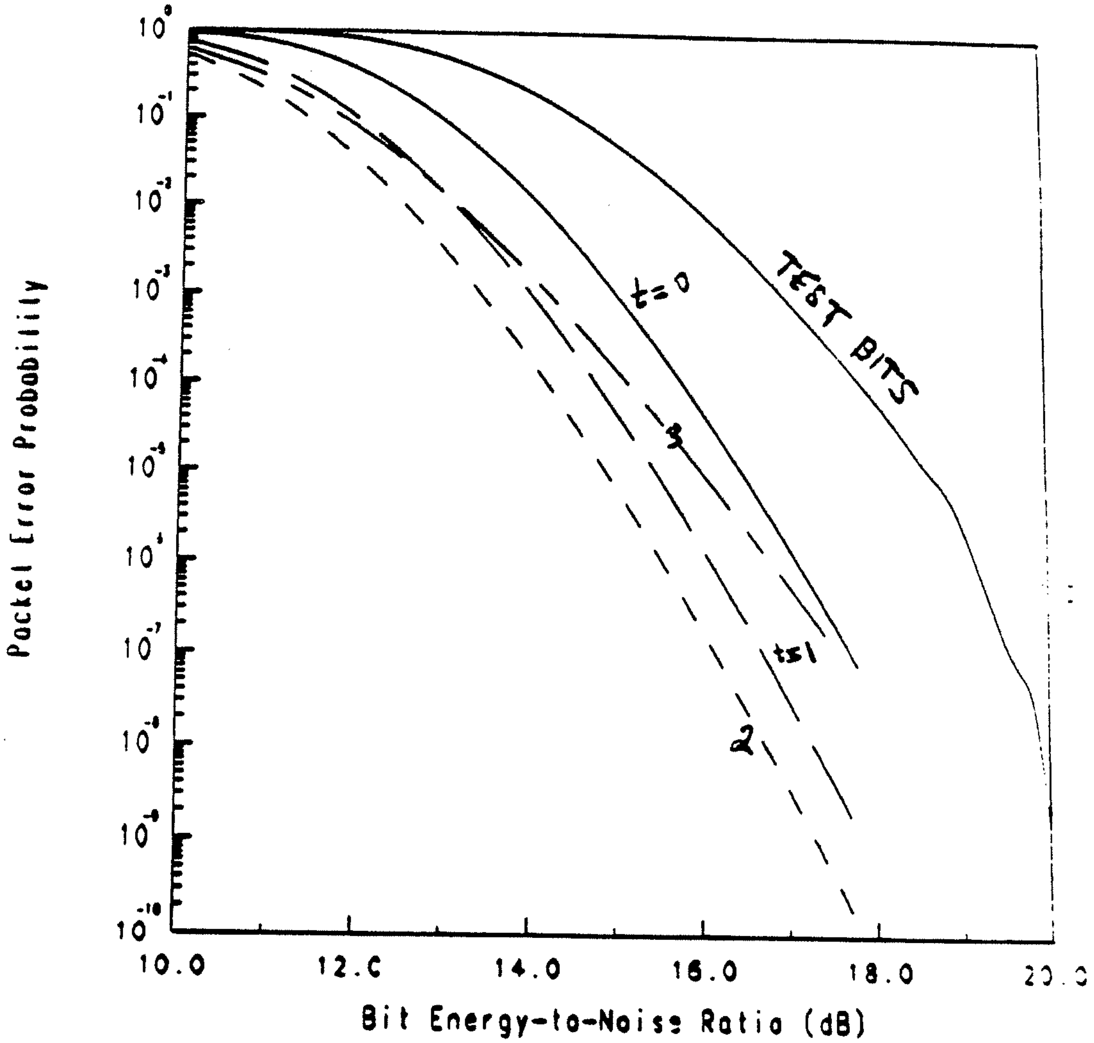

Figure 6.3: Packet Error Probability versus $\frac{E_b}{N_0}$ for a (127,105,7) BCH binary inner code.

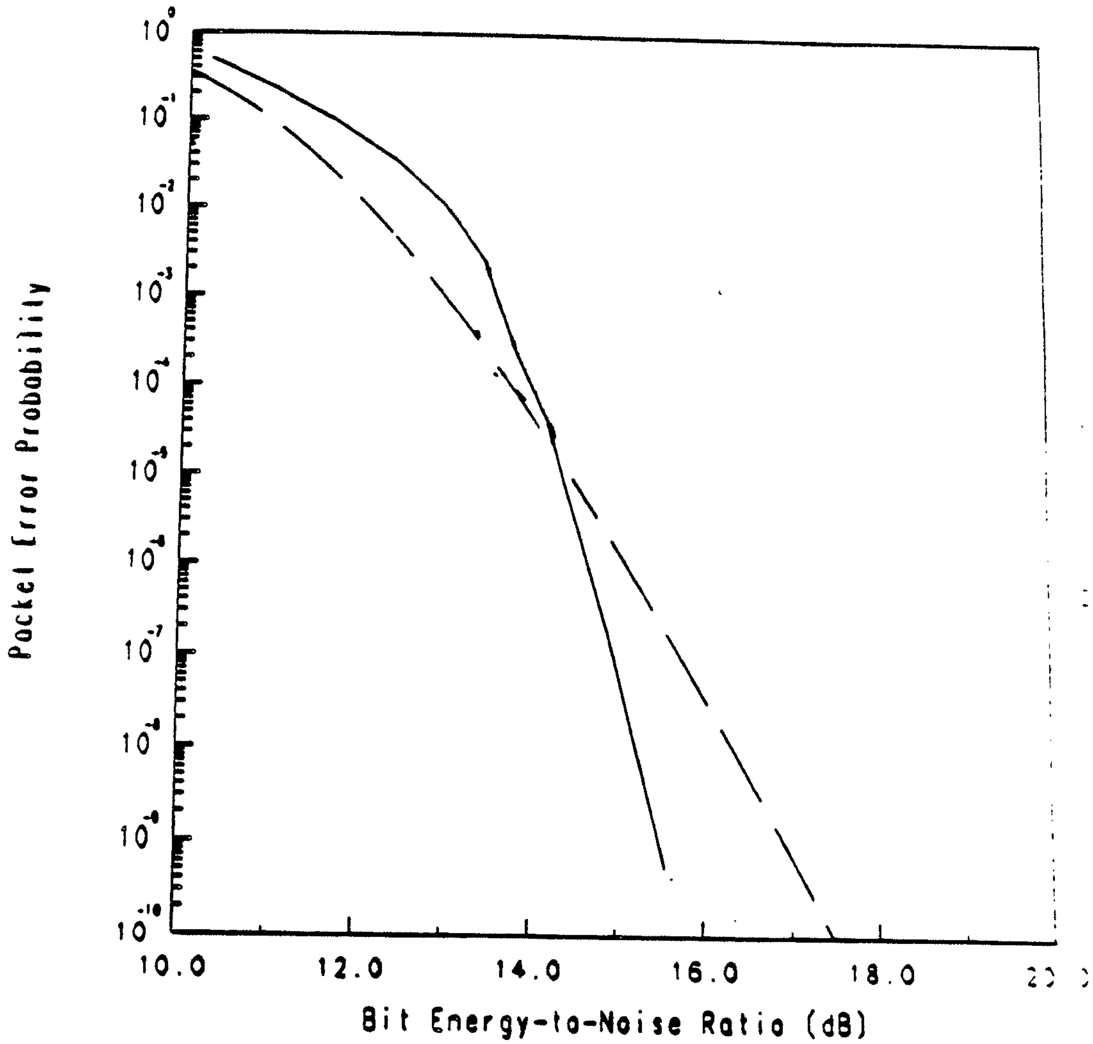

Figure 6.4: Packet Error Probability versus $\frac{E_b}{N_0}$ for a (32,9) Reed-Solomon inner code.

evaluating whether a received hop is reliable or if that hop is "bad" and, therefore, the corresponding code symbols should be erased.

The first system proposed in Chapter 5 uses a fixed number of known test bits which are inserted in each hop. The number of erroneous bits received are compared to a threshold for an erasure criteria. For repetition coding this method is not attractive. Improvement in performance is only possible for large values of $\Lambda(> 4)$. This is not true for the Reed-Solomon coded system. Even for $\Lambda = 1$ the improvement is still substantial. For both systems the increase in the number of test bits in each hop improves the reliability of the side information.

In the second system we have a concatenated coded system with the inner code used for error detection and error correction. As compared to the previous technique, the concatenated coded system is superior in performance. The tradeoff is that of complexity versus performance. In the first case the generation of side information is very simple. In the second system we need an additional decoder thus increasing hardware and time complexity.

CHAPTER VII

CONCLUSIONS

One of the advances in decoding to arise since Shannon's work in 1948 is algebraic decoding, where the decoding problem is that of two computational problems in finite fields:

- determine the coefficients of the error-locator polynomial, and
- given the coefficients of the error-locator polynomial find its roots.

To be able to use algebraic decoding one imposes restrictions on the receiver such as information loss quantization of amplitude at the output of the demodulator, which forces the code to be suboptimal, and imposing algebraic structure on the code such as linearity and cyclicity. Moreover, if we provide the decoder errors and erasures, then we generally reduce the number of decoding errors at the cost of introducing decoding erasures. Algebraic errors-and-erasures decoding algorithms can be used to correct the resulting errors and erasures. The resulting overall error probability is lower than that attainable with error correction only.

In this thesis we proposed several techniques to enhance the performance of coded digital communications systems by creating an error-and-erasure channel seen by an encoder-decoder pair. We started by discussing the general decoding problem of linear block codes. Then we summarized some techniques that could simplify the decoding process on the expence of trading some performance. One technique is concatenated coding which is very attractive to achieve long block codes with low complexity. The error correcting properties of concatenated codes was evaluated in Chapters 3 and 4, for soft decision decoding. We developed two algorithms for coherent reception and noncoherent reception, respectively. These algorithms use errors-and-erasures decoding and make use of several branches with different tentative decisions giving rise to parallel decoding. The set of thresholds Λ_z and H_z (for the coherent and noncoherent cases, respectively) for each algorithm is chosen to optimize the error correcting capability of the code. These algorithms combine the power of soft decision decoding with low complexity, and algebraic decoding. Thus we are able to use soft decision decoding of a long code, something which is prohibitively complex to perform with one decoder. Therefore, we recover most of the information loss which result in quantization needed to use algebraic decoding. We found an expression for the error correcting capability of the code that depends on z (the number of branches), and we showed for the coherent case that full error correcting capability is achied asymptotically with z. However, the numerical results indicated that only four decoders are sufficient to achieve over 95 % of the error correcting capability.

In Chapter 5 we let the transmitter help in establishing an error-and-erasure channel for a slow-frequency-hopped spread spectrum comminication system in a Rayleigh faded channel. We transmit a known sequence of test bits in each dwell interval. At the receiver we use the number of errors in these test bits and compare it to a threshold upon which we decide whether to erase the symbols in the corresponding dwell interval or to declare the hop as reliable. The symbols of each codeword are fully interleaved such that no two symbols from the same codeword are transmitted in the same hop. For repetition coding this method is not attractive. Improvement in performance is only possible when transmitting many test symbols. However, for a Reed-Solomon coded system, transmitting even one test symobl improves performance substantially. For instance, for (32,5) Reed-Solomon code and $\Lambda = 3$, there is more than 1.8 dB improvement in the required $\frac{E_b}{N_0}$ to achieve probability of symbol error less than 10^{-4}.

In Chapter 6 we use a concatenated coded system with the inner decoder used to help the outer decoder in whether the received hop is reliable or not. The inner decoder is used for error detection and error correction. As compared to the previous technique, the concatenated coded system is superior in performance. For probability of packet error 10^{-6} there is more than 3 dB improvement in the required $\frac{E_b}{N_0}$ over the test bits case (assuming equal redundancy in each hop and same outer code). The tradeoff between the two systems proposed in Chapters 5 and 6 is that of complexity versus performance. In the first case the generation of side information is very simple. In the second system we need an additional decoder thus increasing hardware and time complexity.

APPENDICES

APPENDIX A

Solving for the Optimal Decoder Strategy

Let $f(\delta_k, \delta_{k-1}) = (1 - \delta_k)^2 + \delta_{k-1}^2$ be a sequence of functions defined for $k = 1, 2, ..., z + 1$ with δ_k satisfying the following properties:

1. $\delta_k \in (0, \frac{1}{2})$ for $k = 1, 2, ..., z$
2. $\delta_0 = 0$
3. $\delta_{z+1} = 1 - \delta_z$

Also, define $\Lambda_z = \{\delta_1, \delta_2, ..., \delta_z\}$. We need to find

$$\max_{\Lambda_z} \min_{k} f(\delta_k, \delta_{k-1}). \tag{A.1}$$

We will show first that it is necessary to have $f(\delta_k, \delta_{k-1}) = \alpha, k = 1, ..., z+1$, α being some constant, then prove α is unique for a given integer z.

Necessary condition:

Assume $\exists$ some strategy Λ_z such that for some α (that depends on z)

$$f(\delta_k, \delta_{k-1}) = \alpha,$$

$$k = 1, 2, ..., z + 1.$$

To prove the necessary condition, we need to show that for any $\Lambda'_z \neq \Lambda_z$ (i.e., $\delta'_k \neq \delta_k$, for some k.)

$$\max_{\Lambda'_z} \min_k f(\delta_k, \delta_{k-1}) \ < \ \alpha.$$

For any strategy $\Lambda_z \neq \Lambda'_z$ we have $\delta_j > \delta'_j$ or $\delta_j < \delta'_j$, for at least one j. Now let

$$j = \arg\min_k \{\delta_k : \delta_k < \delta'_k\};$$

if such j exists, then it is obvious that

$$(1 - \delta'_j)^2 + \delta'^2_{j-1} < (1 - \delta_j)^2 + \delta^2_{j-1} = \alpha;$$

if such j does not exist, let

$$l = \arg\max_k \{\delta_k : \delta_k > \delta'_k\}.$$

Then we have, $(1 - \delta'_{l+1})^2 + \delta'^2_l < (1 - \delta_{l+1})^2 + \delta^2_l = \alpha$; we know that l or j, or both, exist. Hence, the optimal strategy is Λ_z.

Existence:

We have assumed earlier the existence of some strategy Λ_z and α such that

$$f(\delta_k, \delta_{k-1}) = \alpha, \tag{A.2}$$

$$k = 1, 2, ..., z + 1.$$

Now we show that a positive solution for the above $z+1$ (nonlinear) equations with $z+1$ unknowns always exist. We prove this by showing a computational algorithm converges to a solution of (27). Using $\delta_0 = 0$ and $\delta_{z+1} = 1 - \delta_z$ we rewrite (27) as

$$\begin{aligned} (1-\delta_1)^2 &= \alpha \\ (1-\delta_2)^2 + \delta_1^2 &= \alpha \\ &\vdots \\ (1-\delta_z)^2 + \delta_{z-1}^2 &= \alpha \\ 2\delta_z^2 &= \alpha. \end{aligned} \tag{A.3}$$

Begin by choosing an initial value for $\alpha = \alpha^{(0)} < \frac{1}{2}$, (since from Appendix B we know that $\alpha < \frac{1}{2}$). Using (A.3) we can solve for $\delta_z^{(0)}$, $\delta_{z-1}^{(0)}, \ldots \delta_1^{(0)}$ as follows

$$\begin{aligned} \delta_z^{(0)} &= \sqrt{\frac{\alpha^{(0)}}{2}} \\ \delta_{z-1}^{(0)} &= \sqrt{\alpha^{(0)} - (1-\delta_z^{(0)})^2} \\ &\vdots \\ \delta_1^{(0)} &= \sqrt{\alpha^{(0)} - (1-\delta_2^{(0)})^2} \\ A^{(0)} &= (1-\delta_1^{(0)})^2. \end{aligned}$$

Then it is clear that (A.2) will be satisfied if $A^{(0)} = \alpha^{(0)}$. Notice that one can always choose $\alpha^{(0)}$ such that $\alpha^{(0)} - (1-\delta_k^{(0)})^2 > 0, \quad \forall k$ and, therefore, the square roots have real positive values. For instance, choose $\alpha^{(0)} = 0.5^-$ since from

Appendix B $(1 - \delta_k)^2 + \delta_{k-1}^2 < \frac{1}{2} \Rightarrow 0.5 - (1 - \delta_k)^2 > \delta_{k-1}^2 > 0$. If $A^{(0)} = \alpha^{(0)}$ and a solution for (26) exists. If the above condition is not satisfied then consider the two cases: $A^{(0)} < \alpha^{(0)}$ or $A^{(0)} > \alpha^{(0)}$. In either case we are to show that it is possible to update $\alpha^{(0)}$, (say) n times, until the the resulting $\alpha^{(n)}$ and $A^{(n)}$ are as close as desired; thus proving the existence of a solution for (A.2).

<u>Case I</u> $A^{(0)} < \alpha^{(0)}$.

In this case use $\alpha = \alpha^{(1)}$ by updating $\alpha^{(0)}$ such that

$$\alpha^{(1)} = \alpha^{(0)} - \epsilon_1$$

where ϵ_1 is an arbitrary small positive number. Then the following holds:

$$\begin{aligned}
\alpha^{(1)} &< \alpha^{(0)} \Rightarrow \\
\delta_z^{(1)} &< \delta_z^{(0)} \\
\delta_{z-1}^{(1)} &< \delta_{z-1}^{(0)} \\
&\vdots \\
\delta_1^{(1)} &< \delta_1^{(0)} \\
A^{(1)} &> A^{(0)}.
\end{aligned}$$

The last equation follows because all functions considered are continuous; thus we can always choose ϵ_1 to satisfy the last equation and have $A^{(1)} < \alpha^{(1)}$. If we continue the process above, we have at stage n

$$\alpha^{(n)} = \alpha^{(0)} - \sum_{i=1}^{n} \epsilon_i.$$

The algorithm stops when

$$\mid \alpha^{(n)} - A^{(n)} \mid < \epsilon. \tag{A.4}$$

Due to the continuity of the functions, and

$$A^{(n+1)} > A^{(n)}$$

$$\alpha^{(n+1)} < \alpha^{(n)}$$

$\exists n$ and $\{\epsilon_1, ..., \epsilon_n\}$ such that (A.3) is satisfied. Thus the algorithm converges to a solution of (27).

Case II $A^{(0)} > \alpha^{(0)}$.

The same steps hold as in the previous case except updating $\alpha^{(0)}$

$$\alpha^{(n)} = \alpha^{(0)} + \sum_{i=1}^{n} \epsilon_i.$$

Uniqueness:

Assume there exists strategies Λ and Λ' such that (for $k = 1, 2, ..., z + 1$)

$$f(\delta_k, \delta_{k-1}) = \alpha \ , \ f(\delta'_k, \delta'_{k-1}) = \alpha', \ \ \alpha < \alpha'.$$

Solving $(1 - \delta_k)^2 + \delta^2_{k-1} = \alpha$ for δ_k we get $\delta_k = 1 - \sqrt{\alpha - \delta^2_{k-1}}$. Furthermore, $\delta_1 = 1 - \sqrt{\alpha} > \delta'_1 = 1 - \sqrt{\alpha'}$; assume $\delta_k > \delta'_k$, then $1 - \sqrt{\alpha - \delta^2_k} = \delta_{k+1} > 1 - \sqrt{\alpha' - \delta'^2_k} = \delta'_{k+1}$; hence, by induction,

$$\delta_k > \delta'_k \quad \forall k; \tag{A.5}$$

in particular, for $k = z$, $\delta_z > \delta'_z$. We can find δ_z and δ_z' in terms of α, since for $k = z + 1$ we have $(1 - \delta_{z+1})^2 + \delta^2_z = \alpha$ or $\delta_z = \sqrt{\frac{\alpha}{2}}$. Therefore, $\sqrt{\alpha/2} > \sqrt{\alpha'/2} \Rightarrow \alpha > \alpha'$, which contradicts our assumption that $\alpha < \alpha'$.

APPENDIX B

Solving for a bound on α

Notice that

$$\begin{aligned}\inf_k \sup_{\Lambda_x}\{(1-\delta_k)^2+\delta_{k-1}^2\} &= \inf\{\sup_{\delta_1}(1-\delta_1)^2, \sup_{\delta_1<\delta_2}(1-\delta_2)^2+\delta_1^2, \ldots, \sup_{\delta_{x-1}<\delta_x} 2\delta_x^2\} \\ &= \inf\{1,1,\ldots,1,\frac{1}{2}\} \\ &= \frac{1}{2}\ ; \qquad \text{(B.1)}\end{aligned}$$

(This asserts the fact that $\alpha \leq \frac{1}{2}$.)

The second equality follows from the fact that

$$\sup_{0<y<x<.5}\left\{(1-x)^2+y^2\right\}=1.$$

APPENDIX C

Equivalence of the Odd Case and Even Case

Let the optimum decoder strategy for d_{2H} even be Λ_z, and that for d_{2H} odd be Λ'_z. Then for d_{2H} even

$$\frac{d_{2H}}{2}((d_{1E} - \Delta_k)^2 + \Delta_{k-1}^2) = l,$$

and for d_{2H}

$$\frac{d_{2H} - 1}{2}((d_{1E} - \Delta'_k)^2 + \Delta'^2_{k-1}) + \Delta'^2_z = l'.$$

Furthermore, for a given z Λ_z and l are unique, and similarly Λ'_z and l'. Then if we show that Λ_z realizes l' in the odd case we are done. Using Λ_z in the odd case we have:

$$\begin{aligned}
\frac{d_{2H} - 1}{2}((d_{1E} - \Delta_k)^2 + \Delta_{k-1}^2{}') + \Delta_z^2 &= \frac{d_{2H} - 1}{2}(\frac{2l}{d_{2H}} + \Delta_z^2) \\
&= \frac{d_{2H} - 1}{2}(\frac{2l}{d_{2H}} + \frac{l}{d_{2H}}) \\
&= l - \frac{l}{d_{2H}} + \frac{l}{d_{2H}},
\end{aligned}$$

where we used the fact that $\Delta_z^2 = l/d_{2H}$.

APPENDIX D

The Proof of (4.3)

We have

$$\gamma(H_z, \tau, e) = \sum_{k=2}^{z} \left\{ \frac{1}{\eta_k}(\tau_k - \tau_{k-1} - e_{k-1} + e_k) + \frac{1}{\eta_1}\,\tau_1 \right.$$
$$\left. + \sum_{k=1}^{z-1} \eta_k(e_k - e_{k+1}) \right\} .$$

We break this sum into three sums.

$$\mathrm{I}: \frac{1}{\eta_1}\,\tau_1 + \sum_{k=2}^{z} \frac{1}{\eta_k}\,\tau_k - \sum_{k=2}^{z} \frac{1}{\eta_k}\,\tau_{k-1} =$$
$$\sum_{k=2}^{z} \frac{1}{\eta_k}\,\tau_k - \sum_{k=1}^{z-1} \frac{1}{\eta_{k+1}} \qquad \sum_{k=1}^{z} \frac{1}{\eta_k}\,\tau_k - \sum_{k=1}^{z-1} \frac{1}{\eta_{k+1}}\,\tau_k \qquad =$$
$$\sum_{k=1}^{z} \left(\frac{1}{\eta_k} - \frac{1}{\eta_{k+1}} \right) \tau_k \ ,$$

where we used the assumption $\eta_{z+1} = \infty$ in the last step.

$$\mathrm{II}: \quad \sum_{k=2}^{z} \frac{1}{\eta_k}\,e_k - \sum_{k=2}^{z} \frac{1}{\eta_k}\,e_{k-1} = \sum_{k=2}^{z} \frac{1}{\eta_k}\,e_k - \sum_{k=1}^{z-1} \frac{1}{\eta_{k+1}}\,e_k =$$
$$\sum_{k=1}^{z} \frac{1}{\eta_k}\,e_k - \sum_{k=1}^{z} \frac{1}{\eta_{k+1}}\frac{1}{\eta_{k+1}} \qquad e_k \qquad - \qquad \frac{1}{\eta_1} \qquad e_1$$

$$(D.2)$$

$$\mathrm{III}: \quad \sum_{k=1}^{z-1} \eta_k e_k - \sum_{k=1}^{z-1} \eta_k e_{k+1} + \eta_z e_z = \sum_{k=1}^{z-1} \eta_k e_k - \sum_{k=2}^{z} \eta_{k-1}\, e_k + \eta_z e_z =$$

$$\sum_{k=1}^{z} \eta_k e_z - \sum_{k=1}^{z} \eta_{k-1} e_k + \eta_0 e_1 = \sum_{k=1}^{z} (\eta_k - \eta_{k-1})\, e_k + \frac{1}{\eta_1}\, e_1 \quad ,$$

where we used $\eta_0 = \frac{1}{\eta_1}$.

I + II + III yields the desired expression.

APPENDIX E

Calculating $Pr\{W_0 < W_1\}$

Consider

$$W_0 = \sum_{l=1}^{n} W_{0,l} \quad , \quad W_1 = \sum_{l=1}^{n} W_{1,l}$$

where $\{W_{0,l}\}_{l=1}^{n}$ are i.i.d. with

$$\boxed{\mathcal{F}_{W_{0,l}}(w) = \sum_i \sum_k C_{i,k}\, e^{-A_{i,k}\, w}} \quad , \quad w \geq 0 \quad (= 0 \ \text{ otherwise}) \ .$$

Similarly $\{W_{i,l}\}_{l=1}^{j}$ are i.i.d. with

$$\boxed{\mathcal{F}_{W_{1,l}}(w) = \frac{1}{2}\, e^{-\frac{w}{2}} \quad , \quad w \geq 0} \quad (= 0 \quad \text{otherwise}).$$

Then the characteristic functions of $W_{0,l}$ and $W_{1,l}$ are $\Psi_{W_{0,l}}(js)$ and $\Psi_{W_{1,l}}(js)$, respectively, and given by:

$$\begin{aligned}\Psi_{W_{0,l}}(s) &= \int_0^\infty \mathcal{F}_{W_{0,l}}(w)\, e^{jsw} dw = \sum_i \sum_k C_{i,k} \int_0^\infty e^{-(a_{i,k}-js)w} dw \\ &= \sum_i \sum_k C_{i,k} - \frac{1}{A_{i,k} - js},\end{aligned}$$

and $\Psi_{W_{1,l}}(s) = \dfrac{1}{1-2js}$.

Since all r.v.'s are independent and identically distributed ⇒

$$\Psi_{W_0} = \left(\sum_i \sum_k \frac{C_{i,k}}{A_{i,k} - js} \right)^n$$

and $\Psi_{W_1} = \left(\frac{1}{1-2js}\right)^n$

We have from Feller [11],

$$\boxed{\mathrm{F}_{W_1}(w) = \tfrac{1}{2}\ \frac{\frac{(w}{2)}^{n-1}}{(n-1)!}\ e^{-\frac{w}{2}} \quad , \quad w > 0;}$$

also,

$$\mathcal{F}_{W_0} = \frac{1}{2\pi}\int_{-\infty}^{+\infty} e^{-jsw} \left(\sum_i \sum_k \frac{C_{i,k}}{\mathcal{A}_{i,k} - js}\right)^n ds \ .$$

We are interested in

$$\begin{aligned}
Pr\{W_0 < W_1\} &= \int_0^\infty \mathcal{F}_{W_1}(\alpha)\left[\int_0^\infty \mathcal{F}_{W_0}(w)dw\right]d\alpha \\
&= \int_0^\alpha \frac{\frac{1}{2}\frac{\alpha}{2}^{n-1}}{(n-1)!}e^{-\frac{\alpha}{2}}\left[\frac{1}{2\pi}\int_0^\alpha \int_{-\infty}^\infty\right]d\alpha \\
&= \int_0^\infty \frac{\frac{1}{2}\frac{\alpha}{2}^{n-1}}{(n-1)!}e^{-\frac{\alpha}{2}}\left[\frac{1}{2\pi}\int_{-\infty}^\infty \underbrace{\int_0^\alpha e^{-jsw}\ dw}_{\frac{i}{js}(1-e^{-\alpha js})}\left(\sum\sum\frac{C_{i,k}}{\mathcal{A}_{i,k}-js}\right)^n ds\right]d\alpha \\
&= \frac{1}{2\pi}\int_{-\infty}^\infty\left\{\int_0^\infty \frac{\frac{1}{2}\left(\frac{\alpha}{2}\right)^{n-1}}{(n-1)!}e^{-\frac{\infty}{2}}(1-e^{-s\alpha j})d\alpha\right\}\frac{1}{js}\left(\sum\sum\frac{C_{i,k}}{\mathcal{A}_{i,k}-js}\right)^n ds \\
&= \frac{1}{2\pi}\int_{-\infty}^\infty\left(1-\frac{1}{(1+js^2)^n}\right)\frac{1}{js}\left(\sum_i\sum_k\frac{C_{i,k}}{\mathcal{A}_{i,k}-js}\right)^n ds.
\end{aligned}$$

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] H.W. Arnold, W.F. Bodtmann, "Switched-diversity FSK in frequency-selective Rayleigh fading," *IEEE J. Select. Areas Commun.*, vol. SAC-2, pp. 540-548, July 1984.

[2] E.F. Assmus, J.M. Goethals, H.F. Mattson, "Generalized *t*-designs and majority logic decoding of linear codes," *Inform. Contr.*, vol. 32, pp. 43-60, 1976.

[3] P.A. Bello, "Characterization of randomly time variant linear channels," *IEEE Transactions on Commun. Systems*, vol. CS-11, pp. 360-394, Dec. 1963.

[4] E.R. Berlekamp, R.J. McEliece, "On the inherent intractability of certain coding problems," *IEEE Trans. Inform. Theory*, vol. IT-24, pp. 384-386, May 1978.

[5] R.E. Blahut, *Theory and Practice of Error Control Codes.* California: Addison Wesley, 1983.

[6] K.G. Castor, W.E. Stark, "Parallel decoding of diversity/Reed-Solomon coded SSFH communication with repetition thresholding," *Proc. 20th. Conf. Information Sciences and Systems*, Princeton University, 1986.

[7] K.G. Castor, W.E. Stark, "Performance of diversity/Reed-Solomon coded SSFH communications with errors/erasures via ratio thresholding," *Proceeding of the Allerton Conferences*, 1986.

[8] L.F. Chang, R.B. McEliece, "A study of Viterbi's ratio-threshold anti-jam technique," *Proceedings IEEE Military Communications Conference*, pp. 11.2.1-5, October 1984.

[9] I.I. Dumer, V.A. Zinovev, and V.V. Zyablov, "Cascaded decoding with respect to minimal generalized distance," *Problems of Control and Information Theory*, Vol. 10, No. 1, 1982, pp. 1-17.

[10] T. Ericson, "Analyses of the Blokh-Zyablov decoding algorithm," Internal Report (1986).

[11] W. Feller, *An Introduction to Probability Theory and Its Applications.* New York: Wiley, 1968.

[12] G.D. Forney, *Concatenated Codes*, MIT research monograph No. 37, The MIT press, Cambridge, Mass. 1966.

[13] F.D. Garber, M.B. Pursley, "Effects of frequency-selective fading on slow-frequency-hopped DPSK spread-spectrum multiple-access communications," *Proceedings of the 1982 IEEE Military Commun. Conference*, vol. 2, pp. 35.2.1-6.

[14] F.D. Garber, M.B. Pursley, "Effects of time-selective fading on slow-frequency-hopped DPSK spread-spectrum multiple-access communications," *Conference Record, IEEE National Telecommun. Conference*, vol. 4, pp. G8.1.1-5.

[15] E.A. Geraniotis, "Error probabilities for coherent hybrid SFH/DS spread-spectrum multiple-access communications," *Conference Record, IEEE National Telecommun. Conference*, vol. 4, pp. G8.6.1-5.

[16] E.A. Geraniotis, M.B. Pursley, "Error probabilities for SFH spread-spectrum multiple-access communications over fading channels," *IEEE Transactions on Commun.*, vol. Com-30, no. 5, pp. 986-1009, May 1982.

[17] T. Kailath, "Channel characterization: time-variant dispersive channels," *Lectures on Commun. System Theory*, McGraw-Hill, New York, 1961.

[18] B.G. Kim, W.E. Stark, "Coding for spread-spectrum communication networks," *IEEE Trans. Commun.*, to be published.

[19] L.B. Levitin, C.R. Hartmann, "A new approach to the general minimum distance decoding problem: the zero-neighbors algorithm," *IEEE Trans. Inform. Theory*, vol. IT-31, pp. 378-385, May 1985.

[20] J.L. Massey, *Threshold decoding.* Cambridge, MA: MIT Press, 1963.

[21] R.J. McEliece, W.E. Stark, "Channels with block interference," *IEEE Transactions on Inform. Theory*, vol. IT-30, no. 1, January 1984.

[22] A.M. Michelson and A.H. Levesque, *Error-Control Techniques for Digital Communication.* New York: Wiley, 1985.

[23] P. Monsen, "Fading channel communications, " *IEEE Communications Magazine*, vol. 18, no. 1, pp. 16-25, Jan. 1980.

[24] M.B. Pursley, "Packet error probabilities in frequency-hop radio networks-coping with statistical dependence and noisy side information," *IEEE Global Telecommun. Conference Record*, vol. 1, pp. 165-170, Dec. 1986.

[25] M.B. Pursley, "Tradeoffs between side information and code-rate in slow-frequency-hop packet radio networks," *Conference Record, IEEE International Conference on Commun.*, June 1987.

[26] M.B. Pursley, W.E.Stark, "Performance of Reed-Solomon coded FH spread spectrum communication in partial-band interference," *IEEE Trans. on Commun.*, vol. Comm-33, No. 8, Aug 1985.

[27] A.M. Saleh, R.A. Valenzuela, "A statistical model for indoor multipath propagation," *IEEE J. Select. Areas Commun.*, vol. SAC-5, pp. 128-137, Feb. 1987.

[28] M. Schwartz, W.R. Benett, and S. Stein, *Communication Systems and Techniques*, McGraw-Hill, New York, 1966, Part III.

[29] C.E. Shannon, "A mathematical theory of communication," *Bell System Tech. Jour.*, vol. 38, pp. 61-656.

[30] N.J. Sloane, "A survey of constructive coding theory and a table of binary codes of highest known rate," *Discrete Mathematics*, vol. 3, pp. 265-294, 1972.

[31] W.E. Stark, "Coding for frequency-hopped spread-spectrum communication with partial-band interference-part I: capacity and cutoff rate," *IEEE Trans. Commun.*, vol. COM-33, pp. 1036-1044, Oct. 1985.

[32] W.E. Stark, "Capacity and cutoff rate of noncoherent FSK with nonselective Rician fading," *IEEE Trans. Commun.*, vol. COM-33, pp.1036-1044, Sept. 1985.

[33] A.J. Viterbi, "A robust ratio threshold technique to mitigate tone and partial band jammer in coded MFSK systems," *Proceedings of the 1982 IEEE Military Communications Conference*, October 1982.

[34] A.J. Viterbi, I.M. Jacobs, "Advances in coding and modulation for noncoherent channels affected by fading, partial-band, and multiple-access interference," *Advances in Communications Systems*, vol. 4. pp. 279-308.

[35] A.J. Viterbi and J.K. Omura, *Principles of Digital Communication and Coding.* New York: McGraw Hill, 1979.

[36] L. Weng, "Concatenated codes with larger minimum distance," *IEEE Trans. Inform. Theory*, vol. IT-23, no.5,pp. 613-615, Sept. 1977.

[37] J.M. Wozencraft, I.M. Jacobs, *Principles of Communication Engineering.* John Wiley and Sons, 1965.

[38] V.A. Zinov'ev, "Generalized cascade codes," *Problems of Information Transmission*, vol. 12, 1976.

[39] V.A. Zinov'ev, V.V. Zyablov, "Decoding of nonlinear generalized cascade codes," *Problems of Information Transmission*, no. 14, 1978.

[40] V.V. Zyablov, "Optimization of concatenated decoding algorithms," *Probl. Peredachi Inf.* vol. 9, No.1, pp. 26-32, 1973.

[41] V.V. Zyablov, *Linear Concatented Codes*, Moscow 1982 (in Russian).