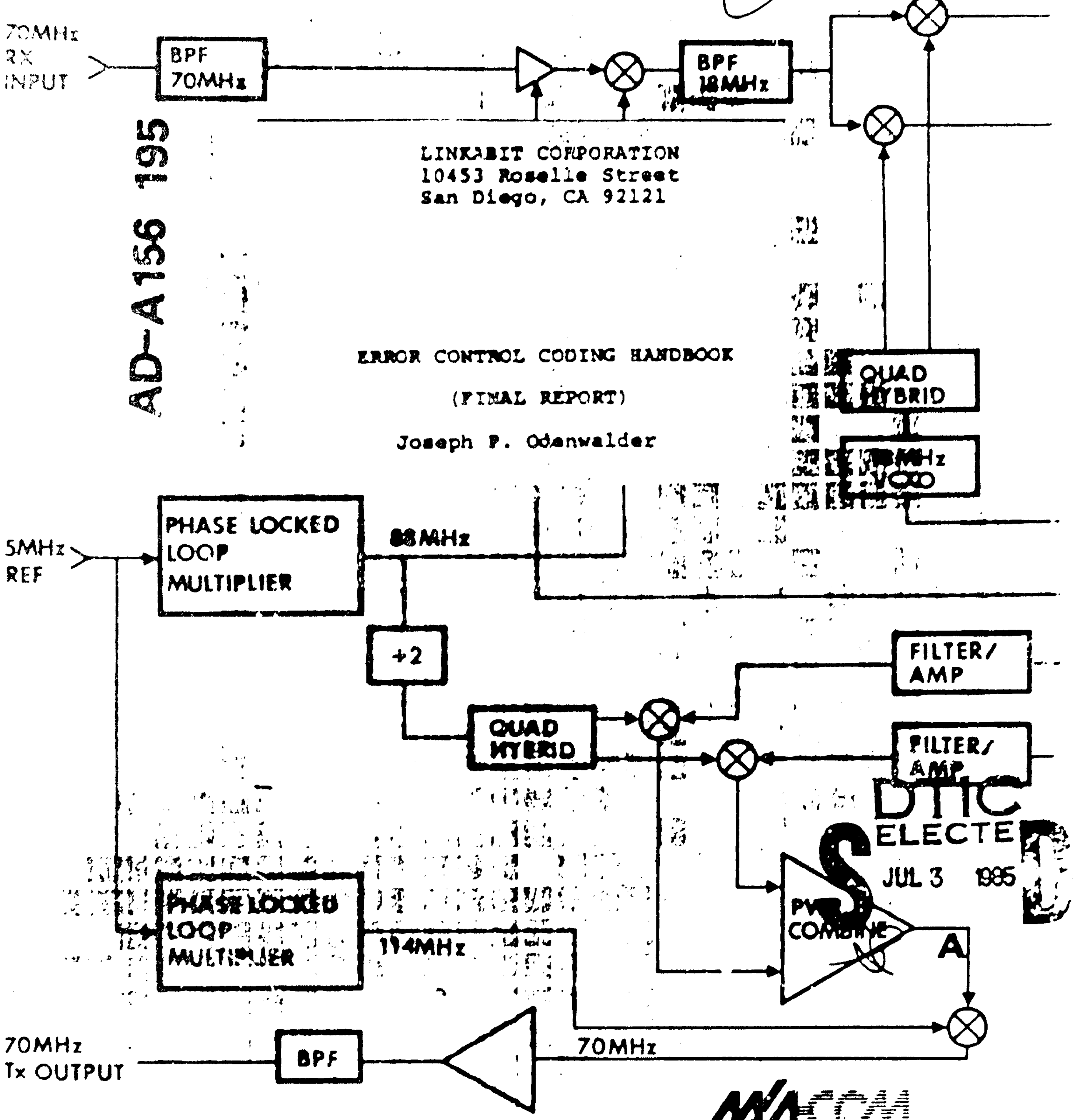


LINKABIT CORPORATION
10453 Roselle Street
San Diego, CA 92121

ERROR CONTROL CODING HANDBOOK

(FINAL REPORT)

Joseph P. Odenwalder

M/A-COM

M/A-COM LINKABIT, INC.
3033 Science Park Road
San Diego, CA 92121
619/457 2340
TWX 910 337 1211

2

LINKABIT CORPORATION
10453 Roselle Street
San Diego, CA 92121

ERROR CONTROL CODING HANDBOOK

(FINAL REPORT)

Joseph P. Odenwalder

15 July 1976

Prepared Under Contract No. F44620-76-C-0056

for

Assistant Chief of Staff, Studies and Analysis
Command, Control and Reconnaissance Division
Strategic Directorate
Hq. United States Air Force

A revised and updated version of much of the material in this report is available in J.P. Odenwalder, "Error Control," in Data Communications, Networks, and Systems, T.C. Bartee, Ed. Indianapolis, IN: Howard Sams, 1985, Chapter 10.

TABLE OF CONTENTS

Section Page

TABLE OF FIGURES

Figure Page

Figure Page

Figure | Page

Figure Page

TABLE OF TABLES

TABLE OF FIGURES AND TABLES THAT PROVIDE PERFORMANCE RESULTS FOR UNCODED SYSTEMS

Figure — Page

Table — Page

TABLE OF FIGURES AND TABLE THAT PROVIDE PERFORMANCE RESULTS FOR HAMMING CODING SYSTEMS

TABLE OF FIGURES AND TABLES THAT PROVIDE PERFORMANCE RESULTS FOR GOLAY CODING SYSTEMS

Figure **Page**

Table **Page**

TABLE OF FIGURES AND TABLES THAT PROVIDE PERFORMANCE RESULTS FOR BCH CODING SYSTEMS

TABLE OF FIGURES AND TABLES THAT PROVIDE PERFORMANCE RESULTS FOR REED-SOLOMON CODING SYSTEMS

TABLE OF FIGURES AND TABLES THAT PROVIDE PERFORMANCE RESULTS FOR VITERBI-DECODED CONVOLUTIONAL CODING SYSTEMS

Figure Page

Table Page

TABLE OF FIGURES AND TABLES THAT PROVIDE PERFORMANCE RESULTS FOR SEQUENTIAL-DECODED CONVOLUTIONAL CODING SYSTEMS

TABLE OF FIGURES AND TABLES THAT PROVIDE PERFORMANCE RESULTS FOR FEEDBACK-DECODED CONVOLUTIONAL CODING SYSTEMS

Figure Page

TABLE OF FIGURES AND TABLES THAT PROVIDE PERFORMANCE RESULTS FOR NONBINARY (DUAL-K) CONVOLUTIONAL CODING SYSTEMS

TABLE OF FIGURES AND TABLES THAT PROVIDE PERFORMANCE RESULTS FOR CONCATENATED CODING SYSTEMS

1.0 Introduction

With the continued improvement in coding techniques and the implementation of these techniques, and the growing acceptance of error control coding, increasingly many systems engineers are incorporating error control codes into communication systems. However, due to the rapid changes in this field and the fact that much of the information needed to decide whether error control coding should be used is in widely scattered or unpublished sources, it has been difficult for the systems engineer to weigh the advantages versus the costs of various coding systems and to specify the parameters of a coding system when error control coding is selected. The purpose of this report is to provide a reference which can be used by systems engineers to aid in selecting and specifying error control codes.

The effort described here emphasizes the coding techniques most likely to be used in applications. The methods of evaluating the performance of various coding techniques and numerous performance curves and tables are presented. In addition other system considerations such as synchronization, automatic gain control (AGC), and implementation complexity, are discussed.

Chapter 2 introduces the advantages and costs of error control coding and presents a brief summary of the performance that can be achieved with several representative coding techniques and of other factors that should be considered in selecting and specifying error control codes.

Additional errors: Phase locked loop's multipliers; frequency multipliers (the idea); Synchronization (electronics); automatic gain control.

The remaining chapters present a more detailed description of error control coding. Chapter 3 begins with a description of the performance which can be achieved without coding and with some theoretical results on the limits of coding. The uncoded performance is included to acquaint the reader, who may be unfamiliar with error control systems, with the usual ways of specifying the error rate performance of a system and to provide a convenient reference for determining the coding gain of coded communication systems.

The two fundamental coding limits discussed are the channel capacity and the computational cutoff rate. The absolute upper limit on the rate of a code (defined as the ratio of the number of encoder input bits to the number of encoder output bits) is the channel capacity and the upper limit for practically implementable systems is the computational cutoff rate. These limits are presented in a form which shows the minimum signal-to-noise ratio for which coding is useful versus the code bandwidth expansion (defined as the inverse of the code rate) for several modulation and channel types. If these results show that the signal-to-noise ratio for a particular modulation and channel is insufficient for any code rate, then what is required is a better modulation technique or system changes that will increase the received signal-to-noise;

there is no need to hopelessly search for a coding technique to achieve some impossible goal.

Chapter 4 through 7 discuss and give the performance of specific coding techniques. Chapter 4 covers block codes and Chapter 5 convolutional codes which are decoded using Viterbi, sequential, and feedback decoding algorithms.

Chapter 6 describes nonbinary symbol convolutional codes and, in particular, the dual-k convolutional coding system which is useful for fading and non-Gaussian noise channels with 2^k-signal MFSK modulation.

Chapter 7 describes and gives performance results for several concatenated coding systems.

A glossary of coding terminology is provided in Appendix A.

2.0 Summary of the Procedures for Specifying Error Control Codes

The digital communication system engineer, who must weigh the advantages of error control coding against its costs, will form a decision based on the nature and quality of the channel and other terminal equipment already available. But with the dramatic improvements in error control techniques in recent years and the greater reliance on satellite and terrestrial microwave links for broadband data transmission, decisions in favor of error control are becoming ever more frequent.

For satellite communication channels the most effective forward error correction techniques can reduce the received signal-to-noise required for a given desired bit error rate by 5 to 6 dB, or more, compared to a system without error control. This translates directly into an equivalent reduction in required satellite effective radiated power, with consequently reduced satellite weight and potentially remarkable reductions in satellite booster costs. For a satellite system with many ground stations an even greater cost savings may be possible by reducing the receiving antenna area by a factor of 4 which is compensated for by the error control savings of 6 dB. The cost of error control is two-fold: the equipment which may be more than compensated by savings which it makes possible in other terminal equipment; and the redundancy required by the error control code. This redundancy need not, however, reduce throughput if additional bandwidth is available on the channel. Satellite channels, in particular, are often not nearly as much bandwidth limited as they are power limited. An error control technique which employs a rate 1/2 code (100% redundancy) will require double the bandwidth of an uncoded system; on the other hand if a rate 3/4 code is used, the redundancy is 33% and the bandwidth expansion only 4/3.

Terrestrial channels such as microwave links, HF and tropospheric propagation links can also be improved by error control techniques. For these channels which are subject to fading and multipath phenomena, the errors tend to occur in bursts, and thus corrupt long strings of data, rather than as single randomly distributed bit errors. A very effective error control technique for these channels is forward error correction coupled with data interleaving before transmission and after reception, which causes the bursts of channel errors to be spread out and thus to be handled by the decoder as if they were random errors.

In the remainder of this section the main factors which must be considered in specifying error control codes are summarized.

2.1. Error Correction Versus Error Detection

One question that must be addressed in weighing the advantages of error control coding against its costs is whether error correction or error detection coding is best for the application.

Error detection techniques are much simpler than forward error correction (FEC). Considerably less redundancy is required to detect up to a given number of errors than to correct the same errors. The weaknesses of error detection, however, are several. First, error detection presupposes the existence of an automatic repeat request (ARQ) feature which provides for the retransmission of those blocks, segments or packets in which errors have been detected. This assumes some protocol for reserving time for the retransmission of such erroneous blocks and for reinserting the corrected version in proper sequence. It also assumes sufficient overall delay and corresponding buffering that will permit such reinsertion. The latter becomes particularly difficult in synchronous satellite communication where the transmission delay in each direction is already a quarter second.

A further drawback of error detection with ARQ is its inefficiency at or near the system noise threshold. For, as the error rate approaches the inverse block (or packet) length, the majority of blocks will contain detected errors and hence require retransmission, even several times, reducing the throughput drastically. In such cases, forward error correction, in addition to error detection with ARQ, may considerably improve throughput. One technique, convolutional coding with sequential decoding is basically a forward error correcting approach but with an inherent error detecting feature provided without additional complexity.

In summary, forward error correction may be desirable in place of, or in addition to, error detection for any of the following reasons:

(1) When a reverse channel is not available or the delay with ARQ would be excessive;

(2) The retransmission strategy is not conveniently implemented;

(3) The expected number of errors, without corrections, would require excessive retransmission.

2.2 Block Versus Convolutional Codes

The two basic types of error control codes are block and convolutional.

Early attempts at designing error control techniques were based on block codes. In the binary case for every block of k information bits, n-k redundant parity-check bits are generated as linear (modulo-2) combinations of the information bits and transmitted along with the information bits as a code of rate k/n bits/ binary channel symbol. The code rate is the ratio of information bits to total bits transmitted - this is also the inverse of the bandwidth expansion factor. The more successful block coding techniques have centered about finite-field algebraic concepts, culminating in various classes of codes which can be generated by means of a linear feedback shift register encoder.

Emphasis in the last decade has turned to convolutional codes, for which the encoder may be viewed as a digital filter, whose output is the convolution* of the input data and the filter impulse response. Several decoding techniques have been developed, which unlike those generally used with block codes, reply more on channel characteristics than on the algebraic structure of the code.

In almost every appplication, convolutional codes outperform block codes for the same implementation complexity of the encoder-decoder. In addition, convolutional codes have several other advantageous features which further tip the scales in their favor:

(a) Code synchronization is much simpler; for rate 1/2 codes only a 2-fold ambiguity needs to be resolved, and for rate 3/4 a 4-fold ambiguity; in contrast for block codes the ambiguity is n-fold where n is the total number of data plus redundant bits in a block.

(b) Channel quality information can easily be utilized with two of the three main convolutional decoding algorithms - on a channel

* Performed with binary field arithmetic, rather than real numbers as for ordinary digital filters.

with BPSK or QPSK modulation disturbed primarily by wideband (e.g., thermal) noise, soft decision decoding, as this is generally called permits the same performance at a signal-to-noise ratio of approximately 2 dB less than hard decision decoding, in which the information fed to the decoder is only the demodulator decision on each bit. Similar improvements are possible with other types of modulation and channel interference (see Section 3.).

(c) Associa^ed with (b) is the decoder's ability to monitor channel quality and to display or output this information in real time while decoding data.

Applications where block codes may be preferable are for error detection or in a few cases for a system with a blocked data format and in which only hard quantized demodulator outputs are available.

2.3 Summary of the Performance of Forward Error Correcting Coding Systems

The efficiency of a communication system in the presence of wideband noise with a single sided noise spectral density of N_o is commonly measured by the received information bit energy-to-noise ratio (E_b/N_o) required to achieve a specified error rate. This ratio can be expressed in terms of the received modulated signal power (P) by

$$\frac{E_b}{N_o} = \frac{P}{N_o} \frac{1}{R_{bps}} \qquad (2.1)$$

where R_{bps} is the information data rate in bits per second (bps). So for a specified error rate a system that requires a smaller E_b/N_o could have a higher data rate or a smaller received power. Note that for a rate k/n code (i.e., n channel bits/k information bits) the channel symbol energy-to-noise ratio is k/n less than the information bit energy-to-noise ratio.

With or without coding an efficient modulation technique should be chosen. For example, a coherent biphase (0° or 180°) BPSK system requires an E_b/N_o of 9.6 dB for a bit error probability of 10^{-5} whereas a DBPSK system requires 10.3 dB.

Table 2.1 shows the coding gain that can be achieved with several coding systems with coherent BPSK or QPSK modulation on a channel with wideband Gaussian noise. With the perfect phase coherence assumed, QPSK performs the same as BPSK.

Table 2.2 shows the required E_b/N_o and the coding gain which can be achieved with a constraint length K=7, (see definition in Table 2.1) rate R= 1/2 Viterbi-decoded convolutional coding system with several

Coding Type	Section where coding type is described	Number of bits receiver quantization	Coding* gain in dB
K=7, R=1/2 Viterbi-decoded Convolutional	5.1	1	3.1
K=7, R=1/2 Viterbi-decoded Convolutional	5.1	3	5.2
K=7, R=1/3 Viterbi-decoded Convolutional	5.1	1	3.5
K=7, R=1/3 Viterbi-decoded Convolutional	5.1	3	5.5
K=9, R=3/4 Viterbi-decoded Convolutional	5.1	1	2.4
K=9, R=3/4 Viterbi-decoded Convolutional	5.1	3	4.3
K=24,R=1/2 Sequential-decoded Convolutional 20Kbps†, 1000-bit blocks	5.2	1	4.2
K=24,R=1/2 Sequential-decoded Convolutional 20Kbps†, 1000-bit blocks	5.2	3	6.2
K=10,L=11,R=1/2 Feedback-decoded Convolutional	5.3	1	2.1
K=8,L=8,R=2/3 Feedback-decoded Convolutional	5.3	1	1.9
K=8,L=9,R=3/4 Feedback-decoded Convolutional	5.3	1	2.0
K=3,L=3,R=3/4 Feedback-decoded Convolutional	5.3	1	1.1
(24,12) Golay	4.2	3	4.0
(24,12) Golay	4.2	1	2.1
(127,92) BCH	4.3	1	3.3
(127,64) BCH	4.3	1	3.5
(127,36) BCH	4.3	1	2.3
(7,4), Hamming	4.1	1	0.6
(15,11) Hamming	4.1	1	1.3
(31,26) Hamming	4.1	1	1.6

* 9. 6 dB required for uncoded system

† The same system at a data rate of 100 Kbps has .5 dB less coding gain.

<u>Notation</u>

K = Constraint length of a convolutional code defined as the number of binary register stages in the encoder for such a code. With the Viterbi decoding algorithm, increasing the constraint length increases the coding gain but also the implementation complexity of the system. To a much lesser extent the same is also true with sequential and feedback decoding algorithms.

L = Look-ahead length of a feedback-decoded convolutional coding system defined as the number of received symbols, expressed in terms of the corresponding number of encoder input bits, that are used to decode an information bit. Increasing the look-ahead length increases the coding gain but also the decoder implementation complexity.

(n, k) denotes a block code (Golay, BCH or Hamming here) with n decoder output bits for each block of k encoder input bits.

Receiver Quantization describes the degree of quantization of the demodulator outputs. Without coding and biphase (0° or 180°) modulation the demodulator output (or intermediate output if the quantizer is considered as part of the demodulator) is quantized to one bit (i.e., the sign is provided). With coding, a decoding decision is based on several demodulator outputs and the performance can be improved if in addition to the sign the demodulator provides some magnitude information.

TABLE 2.1 Summary of the E_b/N_o requirements of several coded communication systems for a bit error rate of 10^{-5} with BPSK modulation.

Modulation	Number of bits of receiver quantization per binary channel symbol (see Table 2.1 note)	E_b/N_o in dB required for $P_b = 10^{-5}$	Coding Gain in dB
Coherent biphase BPSK or QPSK	3	4.4	5.2
BPSK or QPSK	2	4.8	4.8
BPSK or QPSK	1	6.5	3.1
Octal - PSK*	1	9.3	3.7
DBPSK*	3	6.7	3.6
DBPSK*	1	8.2	2.1
Differentially * Coherent QPSK	1	9.0	3.0
Noncoherently Demodulated Binary FSK	1	11.2	2.1

*Interleaving/deinterleaving assumed .

TABLE 2.2 Summary of the E_b/N_o requirements and coding gains of K=7, R=1/2 Viterbi-decoded convolutional coding systems with several modulation types at a bit error rate of 10^{-5}.

types of modulation on a channel with wideband Gaussian noise. Here coding gain is defined as the reduction in the E_b/N_o ratio required to achieve a specified error rate (10^{-5} bit error rate here) of the coded system over an uncoded system with the same modulation.

More extensive performance results are given in later sections. The results in these later sections are presented in several formats for the more common modulation and channel types. Descriptions of the coding techniques are also given.

2.4 Code Specification

The following factors should be considered in selecting and specifying error control codes.

(1) Performance required for a particular modulation/demodulation, channel, and, if known, coding technique. For example, the error probability (bit, block, etc.) for several E_b/N_o ratios could be specified.

(2) Modem interface requirements.

(3) Synchronization requirements. That is, the method of determining the start of a block or other code grouping.

(4) Data rate requirements.

(5) Modem phase ambiguity requirements. Some decoders can internally compensate for the effects of a 90 or 180 degree phase ambiguity

present in BPSK or QPSK modems which obtain a carrier phase referenced from the data modulated signals.

(6) Encoder-decoder delay requirements. That is, the delay in bit times from the time an information bit is first put into the encoder to the time it is provided as an output from the decoder.

(7) Decoder start-up delay.

(8) Built-in test requirements.

(9) Package requirements. The decoder could be on a card for insertion in an existing modem or a separate decoder could be provided. Power and thermal requirements should also be specified.

3.0 Potential Advantages of Coding

Before proceeding to the analysis and error rate performance evaluation of specific error control codes, it is helpful to briefly review the performance of several common uncoded communication systems and to determine the maximum possible coding gain with several modulation techniques and channel types.

3.1 Uncoded System Error Rate Performance

Many types of channels, modulation types, and performance criteria have been studied. Here, for reference, we present the bit error probability performance of several uncoded communications systems on an additive white Gaussian noise channel and on a Rayleigh fading channel.

3.1.1 Additive White Gaussian Noise Channel

The additive white Gaussian noise channel model is a widely used channel model which is valid for channels where the primary disturbance is due to receiver thermal noise or wideband noise jamming. This model is a good representation of the disturbance in many space and satellite communication links.

3.1.1.1 Coherent Phase-Shift Keyed Systems

With a phase-shift keyed (PSK) system one of M (usually $M = 2^m$) different phases is transmitted on each channel use. Figure 3.1 gives the bit error rate versus E_b/N_o performance of BPSK, QPSK, and octal-PSK [18] systems. The QPSK and octal-PSK results are based on a Gray coding for the m-bit modem input symbol-to-M-ary channel phase mapping as shown in Figure 3.2. This mapping guarantees that when the received signal is hard-decision demodulated to a phase next to the correct phase (the most common type of error), only one bit error results.

Figure 3.1 shows that with the perfect phase reference assumed here the BPSK and QPSK bit error probability performances are identical. This is because the in-phase and quadrature demodulator components with QPSK are independent Gaussian random variables and therefore they can be treated separately. The bit error probability of the BPSK and QPSK systems are

$$\left(P_b\right)_{BPSK} = Q\left(\sqrt{\frac{2E_s}{N_o}}\right) \qquad (3.1)$$

and

$$\left(P_b\right)_{QPSK} = Q\left(\sqrt{\frac{E_s}{N_o}}\right) \qquad (3.2)$$

where

$$Q\left(y\right) = \int_y^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) dx \qquad (3.3)$$

and E_s/N_o is the channel symbol energy-to-noise ratio.

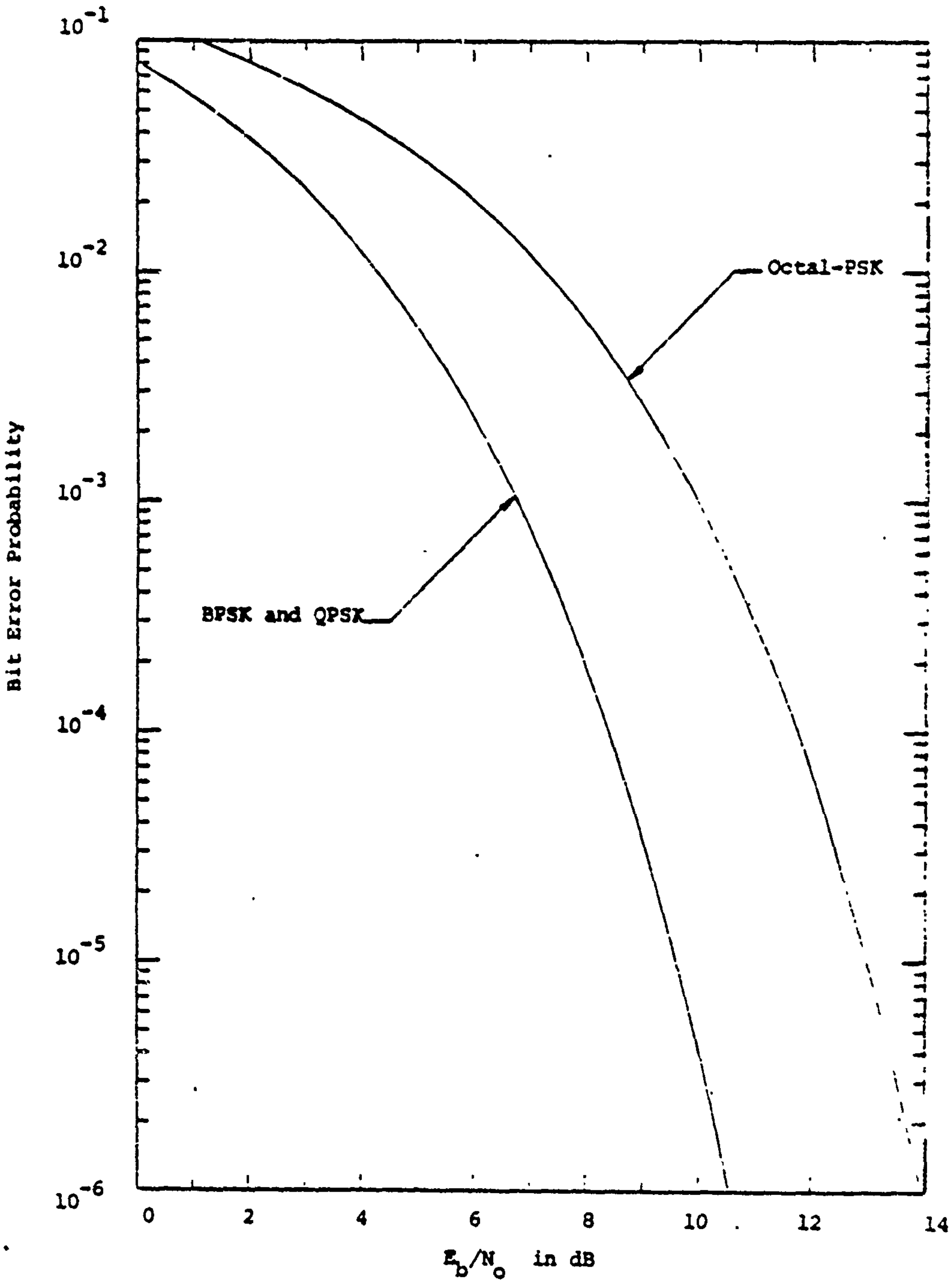
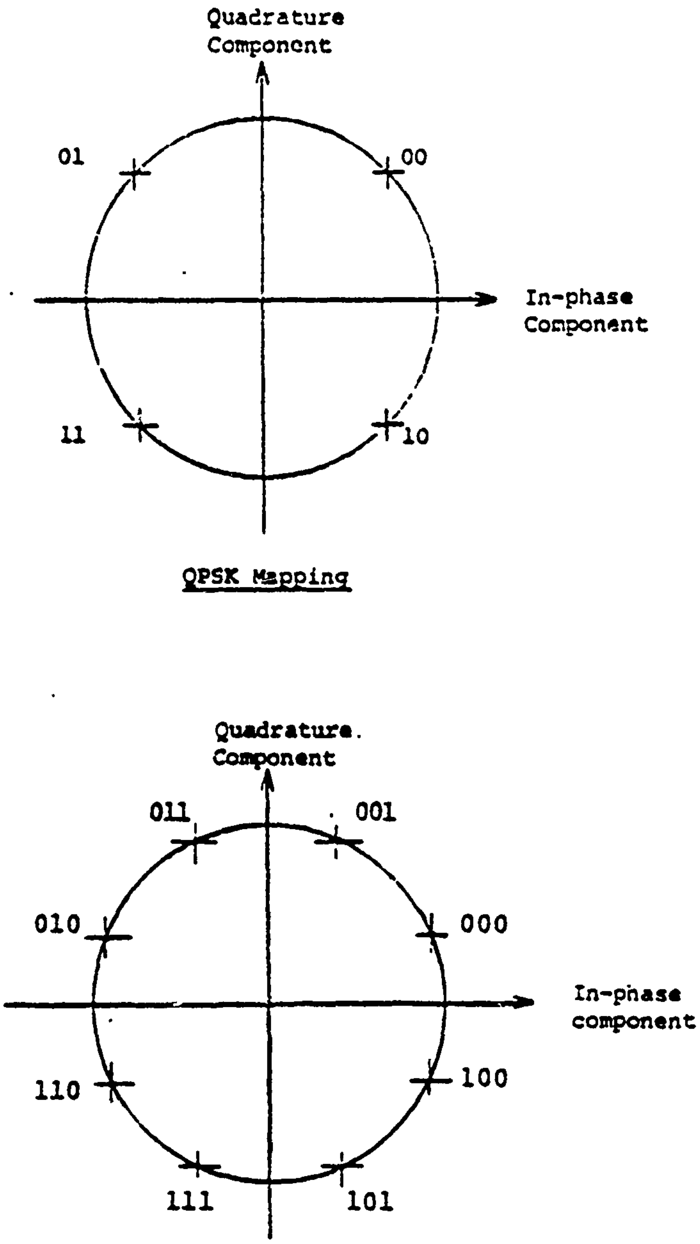


Figure 3.1 Bit error probability versus E_b/N_o performance of coherent BPSK, QPSK, and octal-PSK.



QPSK Mapping

Octal-PSK Mapping

Figure 3.2 Modem input symbol-to-channel phase mapping for QPSK and octal-PSK.

Since with QPSK each channel represents two information bits whereas with BPSK a channel symbol only represents one information bit, (3.1) and (3.2) reduce to

$$\left(P_b\right)_{BPSK} = \left(P_b\right)_{QPSK} = Q\left(\sqrt{\frac{2E_b}{N_o}}\right) \qquad (3.4)$$

where E_b/N_o is 'he information bit energy-to-noise ratio.

The octal-PSK bit error probability expression [18] is more complicated than (3.4) and is not given here. In this report error probability expressions are given only when they are particularly simple or when they provide insight into certain analysis techniques. Otherwise, graphs which more readily show the error probability versus system parameter relationships are given.

The main advantage of using a phase-shift keyed system with a larger number of phases is that the bandwidth which is required for a given data rate is reduced. The main disadvantages are the degraded performance (for more than 4 phases) and the increased sensitivity to phase errors.

3.1.1.2 Differentially Coherent Phase-Shift Keyed Systems

Differentially coherent phase-shift keying is a method of obtaining a phase reference by using the previously received channel symbol. A reference channel symbol is sent first. Then the remaining channel symbols are based on the bit-by-bit modulo-2 sum of the previous and present modem input symbols. Again for 2^m-phases a Gray coding is used to map the m-bit symbol differences to channel phases. The demodulator makes its decision based on the change in phase from the previous to the present received channel symbol.

The bit error probability of a binary differentially coherent phase-shift keying system (DBPSK) is given by [1]

$$P_b = \frac{1}{2} \exp\left(-\frac{E_b}{N_o}\right) \tag{3.5}$$

Figure 3.3 gives a graph of the performance of this binary system and that of a 4-phase (DQPSK) system [17].

The primary advantage of this type of system is the ease with which a phase reference can be obtained. However, comparing the coherently demodulated results of Figure 3.1 with the corresponding results in Figure 3.3 shows that for the higher error rates the differentially coherent systems require a significantly larger energy-to-noise ratio to achieve a specified error rate. For small error rates the energy-to-noise ratio required for DBPSK

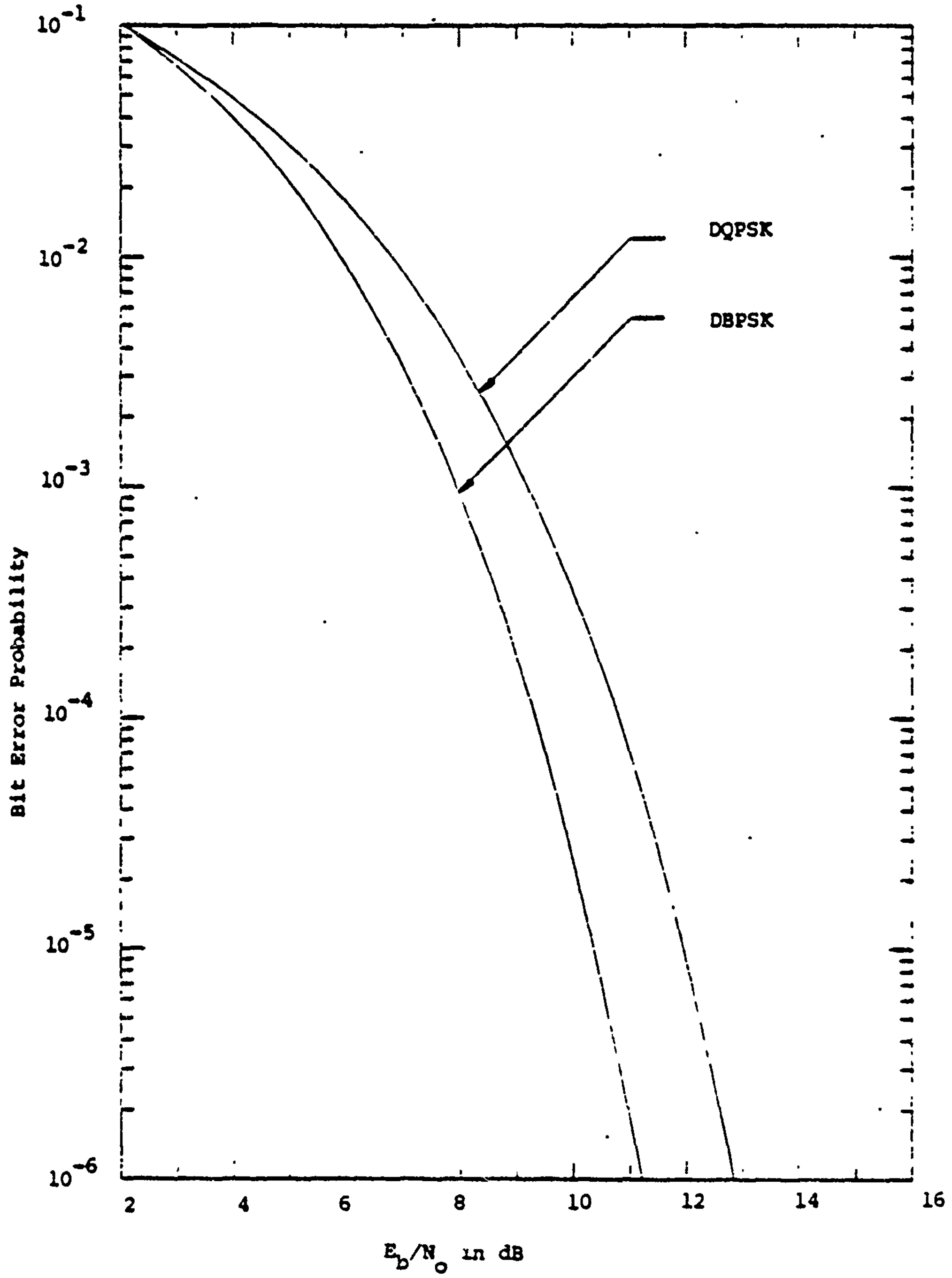


Figure 3.3 Bit error probability versus E_b/N_o performance of DBPSK and DQPSK.

approaches that required for coherent BPSK. Another characteristic of differentially coherent systems is that symbol errors tend to occur in pairs, since an error in one symbol decision indicates a high probability of a bad phase reference, and thus an error, for the next symbol decision.

3.1.1.3 Noncoherently Demodulated Orthogonal Signal Modulated (MFSK) Systems

Another class of modulation systems employs a set of orthogonal signals. For example, for every m modulator input bits one of 2^m frequencies could be sent, with spacing chosen such as to make the signals orthogonal [1]. This type of modulation is referred to as frequency-shift keying (FSK) when only two frequencies are used, or MFSK when more than two frequencies are used.

This type of demodulation is used when the initial phase reference for each channel symbol is unknown or difficult to determine. A common application of MFSK is in jamming environments where the modulator output is frequency hopped.

Orthogonal signal modulation can be viewed as a type of error correcting coding with a bandwidth expansion of $2^m/m$. That is, each set of m information bits could be encoded (mapped) into one of m, 2^m-bit orthogonal sequences. As the bandwidth expansion approaches infinity this modulation/ demodulation technique achieves the maximum possible coding gain on an additive white Gaussian noise channel [1]. However, the large bandwidth expansion required by this technique makes it impractial for large m.

The m-bit symbol error probability for this modulation/demodulation technique is [1]

$$P_s = \frac{1}{M} \exp\left(- \frac{E_s}{N_o}\right) \sum_{j=2}^{M} \left(-1\right)^j \binom{M}{j} \exp\left(\frac{E_s}{jN_o}\right) \qquad (3.6)$$

where $M = 2^m$ and E_s/N_o is the channel symbol energy-to-noise ratio which is related to the information bit energy-to-noise ratio by

$$\frac{E_s}{N_o} = m \frac{E_b}{N_o}$$

The bit error probability is related to the symbol error probability of (3.6) by [2]

$$P_b = \frac{2^{m-1}}{2^m-1} P_s \qquad (3.7)$$

Note that for m = 1, the bit error probability expression reduces to

$$P_b = \frac{1}{2} \exp\left(- \frac{E_b}{2N_o}\right) \qquad (3.8)$$

Figure 3.4 gives the bit error probability performance of this modulation/demodulation technique for m = 1 and 3.

3.1.2 Independent Rayleigh Fading Channel

In some applications fading due to ionospheric variations causes phase and amplitude fluctuations from channel symbol to channel symbol that can severely degrade the error rate performance. The received amplitude c. such a channel can many times be accurately modeled by the Rayleigh probability distribution.

The performance of a system with this type of a channel can be greatly improved by providing some type of diversity; that is, by providing several independent transmissions for each information symbol. Time, spatial, and frequency diversity have been used. Here we will restrict our attention to time diversity which can be achieved by repeating each information symbol several times and using interleaving/deinterleaving for the channel symbols. The result is a channel for which the amplitude and phase of the received channel symbols can be treated as independent random variables with Rayleigh and uniform distributions, respectively. Such a channel is called an independent Rayleigh fading channel.

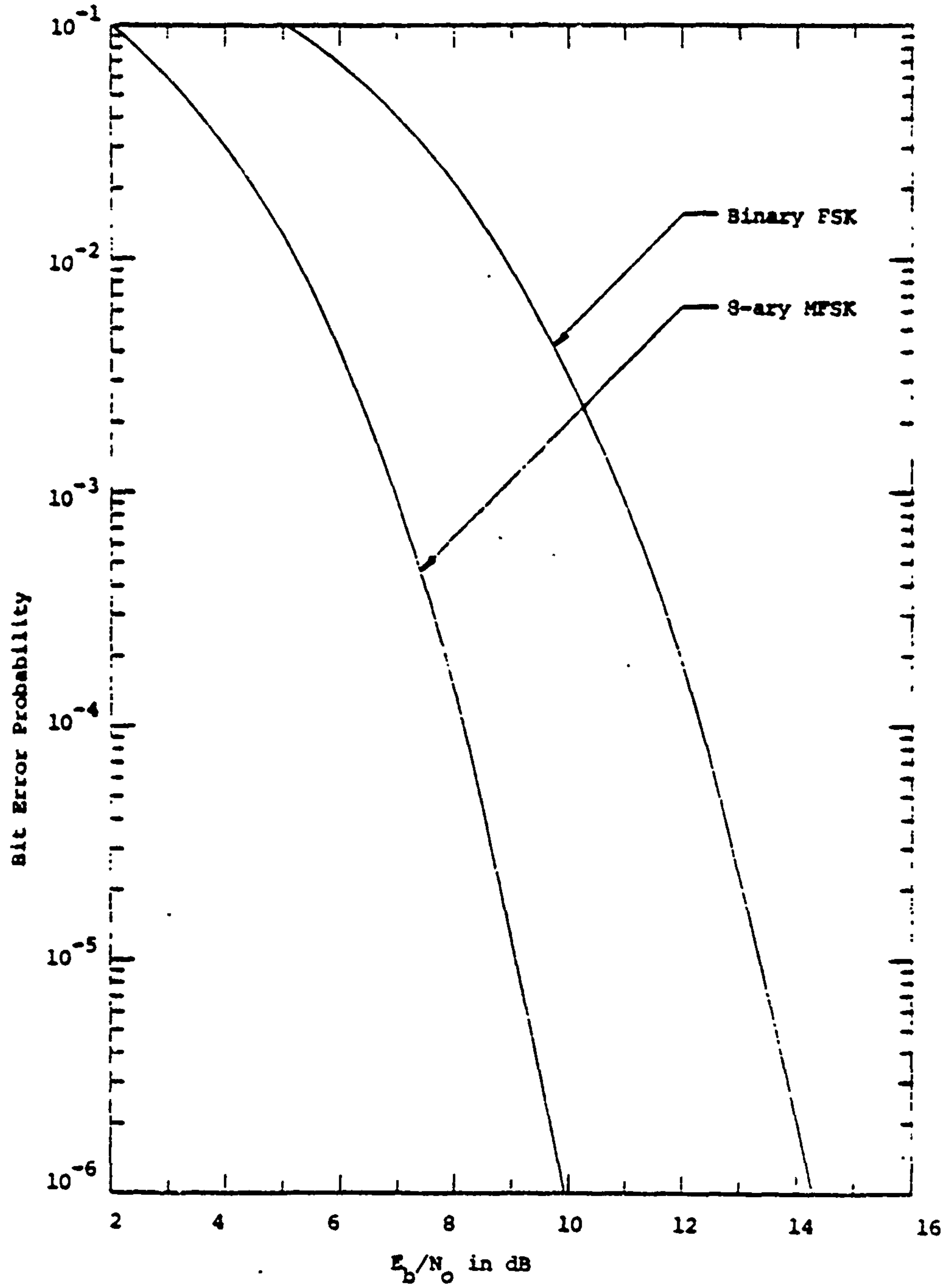


Figure 3.4 Bit error probability versus E_b/N_o performance of binary and 8-ary MFSK.

Since the received amplitude and phase are random variables, we will only consider square-law noncoherently demodulated orthogonal signal modulated MFSK. The closed form expression for the bit error probability of binary FSK on a Rayleigh fading channel is [4]

$$P_b = p_{FSK}^{L} \sum_{i=0}^{L-1} \binom{L-1+i}{i} \left(1-p_{FSK}\right)^{i} \qquad (3.9)$$

where

$$p_{FSK} = \frac{1}{2+\frac{1}{L}\frac{\overline{E_b}}{N_o}} \qquad (3.10)$$

$\overline{E_b}/N_o$ is the mean bit energy-to-noise ratio, and L is the order of the diversity. That is, L channel symbols are transmitted for each information symbol. The order of the diversity (L) corresponds to the bandwidth expansion of a coded system.

Figure 3.5 gives this binary bit error probability for several orders of diversity (L). This figure shows that for a particular error rate, there is an optimum amount of diversity.

For noncoherently demodulated 2^k -signal MFSK a union upper bound on the bit error probability can be

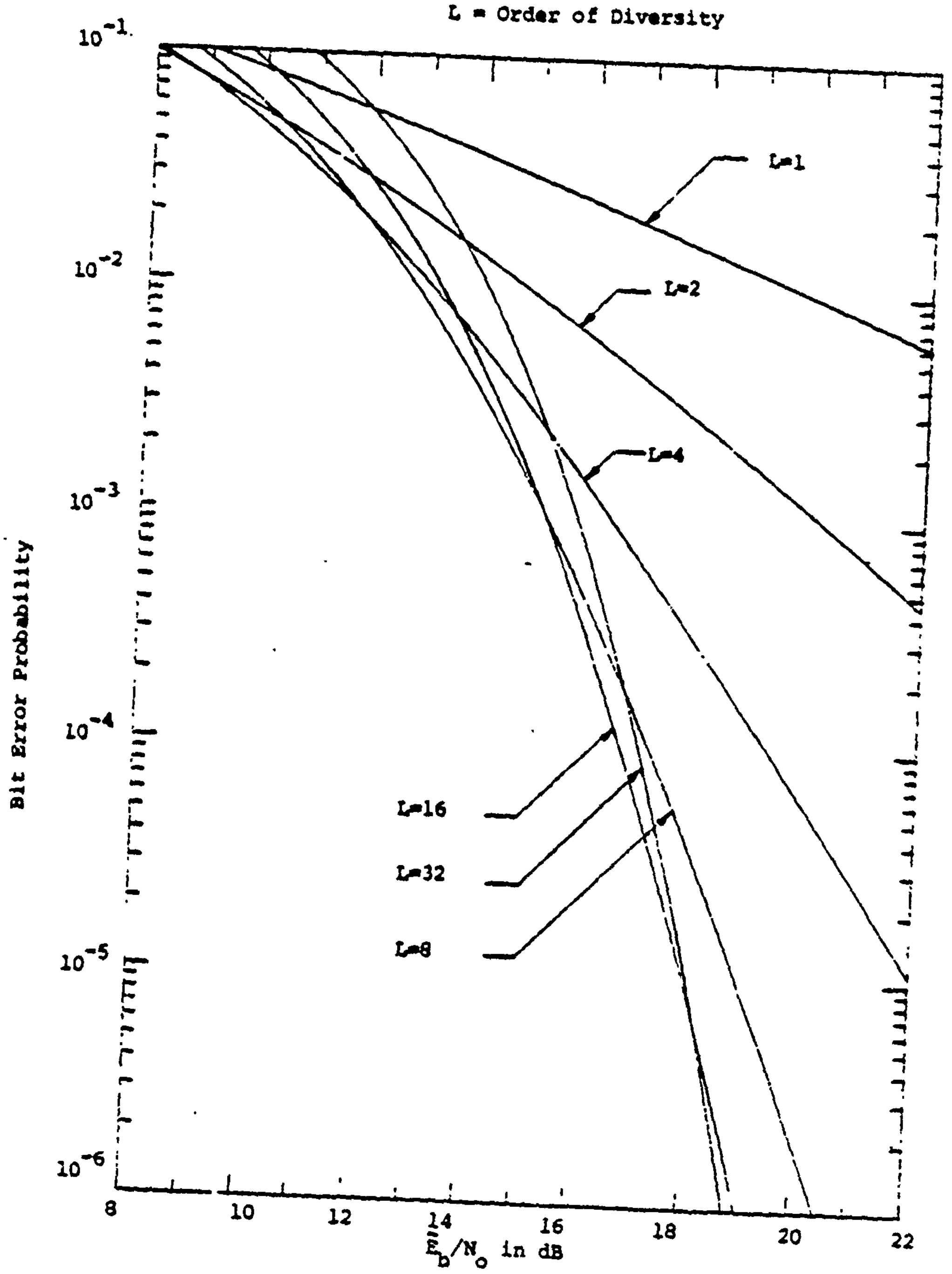


Figure 3.5 Bit error probability versus $\bar{E}_b/N_o$ performance of binary FSK on a Rayleigh fading channel for several orders of diversity.

obtained [19] as 2^{k-1} times the binary error probability of (3.9) with the channel symbol energy-to-noise ratio increased by the factor k. The result is

$$\left(P_b\right)_{\substack{2^k\text{-signal}\\ \text{MFSK}}} \leq 2^{k-1} \; p_{MFSK}^{L} \sum_{i=0}^{L-1} \binom{L-1+i}{i} \left(1-p_{MFSK}\right)^{i} \tag{3.11}$$

where

$$p_{MFSK} = \frac{1}{2+\frac{k}{L}\,\frac{\overline{E_b}}{N_o}} \tag{3.12}$$

In (3.11) and (3.12) the diversity, L, is the number of 2^k-ary channel symbols per k-bit information symbol.

3.1.3 Summary of Uncoded System Performances

Table 3.1 summarizes the E_b/N_o ratios required to obtain a 10^{-5} bit error probability with the modulation and and channel types discussed in the previous sections.

Channel	Modulation / Demodulation	E_b/N_o in dB required for given bit error rate						
		10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}
Additive white Gaussian noise	BPSK and QPSK	-.8	4.3	6.8	8.4	9.6	10.5	11.3
"	Octal-PSK	1.0	7.3	10.0	11.7	13.0	13.9	14.7
"	DBPSK	2.1	5.9	7.9	9.3	10.3	11.2	11.9
"	DQPSK	2.1	6.8	9.2	10.8	12.0	12.9	13.6
"	Noncoherently demodulated binary FSK	5.1	8.9	10.9	12.3	13.4	14.2	14.9
"	Noncoherently demodulated 8-ary MFSK	2.0	5.2	7.0	8.2	9.1	9.9	10.5
Independent Rayleigh fading	Binary FSK, L = 1	9.0	19.9	30.0	40.0	50.0	60.0	70.0
"	Binary FSK, L = 2	7.9	14.8	20.2	25.3	30.4	35.4	40.4
"	Binary FSK, L = 4	8.1	13.0	16.5	19.4	22.1	24.8	27.3
"	Binary FSK, L = 8	8.7	12.8	15.2	17.2	18.9	20.5	22.0
"	Binary FSK, L = 16	9.7	13.2	15.3	16.7	18.0	19.1	20.0
"	Binary FSK, L = 32	10.9	14.1	15.8	17.1	18.1	18.9	19.7

Table 3.1 Summary of uncoded system performances.

3.2 Channel Capacity and Other Fundamental Limits to Coding Performance

Error control coding is a means of adding redundancy to the transmitted symbol stream in such a manner that at the decoder the redundancy can be used to provide a more reliable information transfer. Generally speaking, Shannon [3] has shown that for any input discrete, finite memory channel it is possible to find a code which achieves any arbitrarily small probability of error if the rate of the code is less than the channel capacity (C) and conversely it is not possible to find such a code when the rate is greater than the channel capacity. Unfortunately this result is based on considering the ensemble of all possible codes and thus is only an existence theorem. Systems engineers are faced with the task of finding a code with a reasonable implementation complexity that satisfies their error probability requirements. While Shannon's result is an existence theorem, it is helpful to compare the coding gain of particular coding techniques with the maximum possible coding gain that could be achieved for that code rate.

Another quantity which frequently arises in describing the performance of coded communication systems is the computational cutoff rate R_o. Sequentially decoded convolutional codes are only useful at rates less than R_o. Moreover, for $R<R_o$ most good convolutional codes exhibit a bit error rate proportional to $2^{-K R_o/R}$ [29] where K is the constraint length and R the code rate. Of course, R_o is less than the channel capacity (C).

In general, closed-form expression for R_o and C are difficult to obtain, but numerical evaluation is straightforward. Discussions on the computation and interpretation of these quantities are given in [4] and [5]. In the remainder of this section we present some of these results.

3.2.1 Binary Symmetric Channels

The simplest type of channel is that of the binary symmetric channel (BSC). Such a channel has two inputs and two outputs and the probability of the channel causing an error is the same regardless of which channel symbol was sent. This channel is commonly represented by the channel transition diagram of Figure 3.6. The transitions in this diagram represent the probabilities of receiving the output symbol given the indicated input was transmitted.

The computational cutoff rate and capacity for this channel are [4].

$$R_o = -\log_2\left[\frac{1}{2} + \sqrt{p(1-p)}\right] \qquad (3.13)$$

and

$$C = 1 + p\log_2 p + \left(1-p\right)\log_2\left(1-p\right) \qquad (3.14)$$

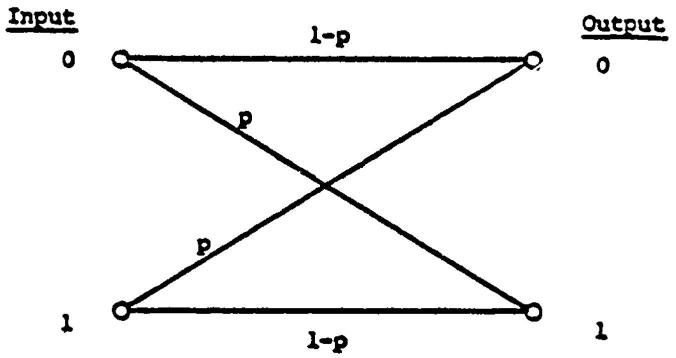


Figure 3.6 Binary symmetric channel transition diagram.

where p is the probability of an error in either channel input symbol and the units of both are bits per channel use.

In later sections where nonbinary channel inputs are considered, the R_o and C quantities will be appropriately defined and computed. With these units and defining the error control code rate R to be the ratio of the number of encoder input bits to the number of encoder output bits, we have that channel coding will be of no help unless $R < C$ and for practical operation $R \leq R_o$ will usually be required.

Figure 3.7 gives curves of the channel error probability, p, required to operate at rates of R_o and C versus the code bandwidth expansion. The bandwidth expansion is defined as one over the code rate. For example, Figure 3.7 shows that a rate 1/2 (bandwidth expansion 2) code is only useful when the channel error probability is less than .11 (i.e., $R < C$) and most coding techniques would require $p \leq .045$ (i.e., $R \leq R_o$) for small output error rates.

Any memoryless channel is converted into a BSC if hard decisions are performed on each received symbol. The channel error rate can be determined from the results of Section 3.1 or from similar error probability curves for other channels. If the channel is not memoryless, inter-

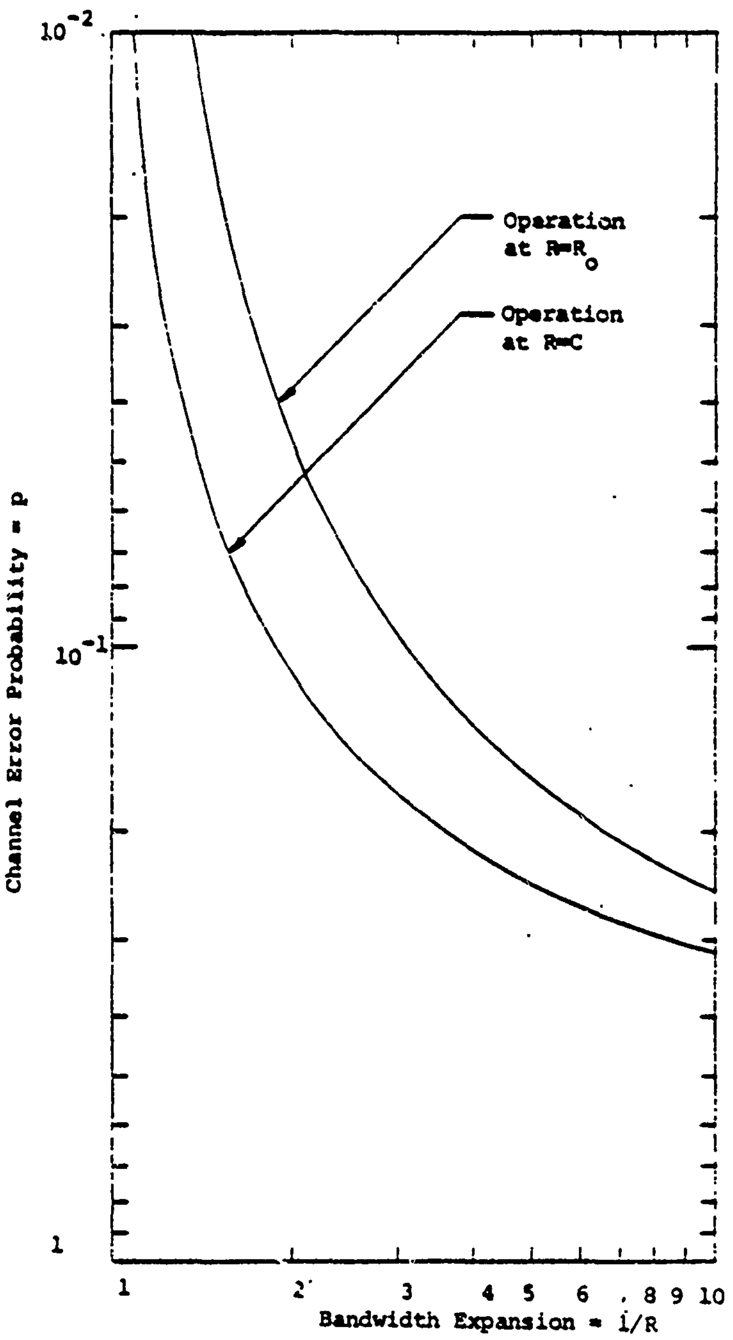


Figure 3.7 Coding limits for a binary symmetric channel.

leaving can be used to make the channel appear to be memoryless to the encoder/decoder. The block diagram of Figure 3.8 shows where the interleaving would be added. At the receiver, deinterleaving is used prior to decoding to recover the sequence corresponding to the encoder output.

When the channel error rate with this binary symmetric channel is not low enough to make coding useful, coding may still be helpful if the modem output is not hard quantized, i.e., not quantized to 1-bit. The demodulator outputs used in this report are defined on a continuum. Before these demodulator outputs can be processed with digital circuits some form of amplitude quantization must be introduced. In fact, such a quantizer is many times considered as part of the demodulator. The demodulator output quantization, receiver quantization, or just quantization discussed in this report all refer to this process of converting a demodulator output defined on a continuum to one of a set of discrete numbers.

With biphase (0° or 180°) modulation and no coding the demodulator produces one output defined over a continuum, for each information bit that was transmitted. A hard (irrevocable) decision as to which information bit was transmitted is made by determining the polarity of the demodulator output. That is, a one bit quantizer is used. This 1-bit quantization is also referred to as hard quantization. Without coding, providing additional amplitude information about the demodulator

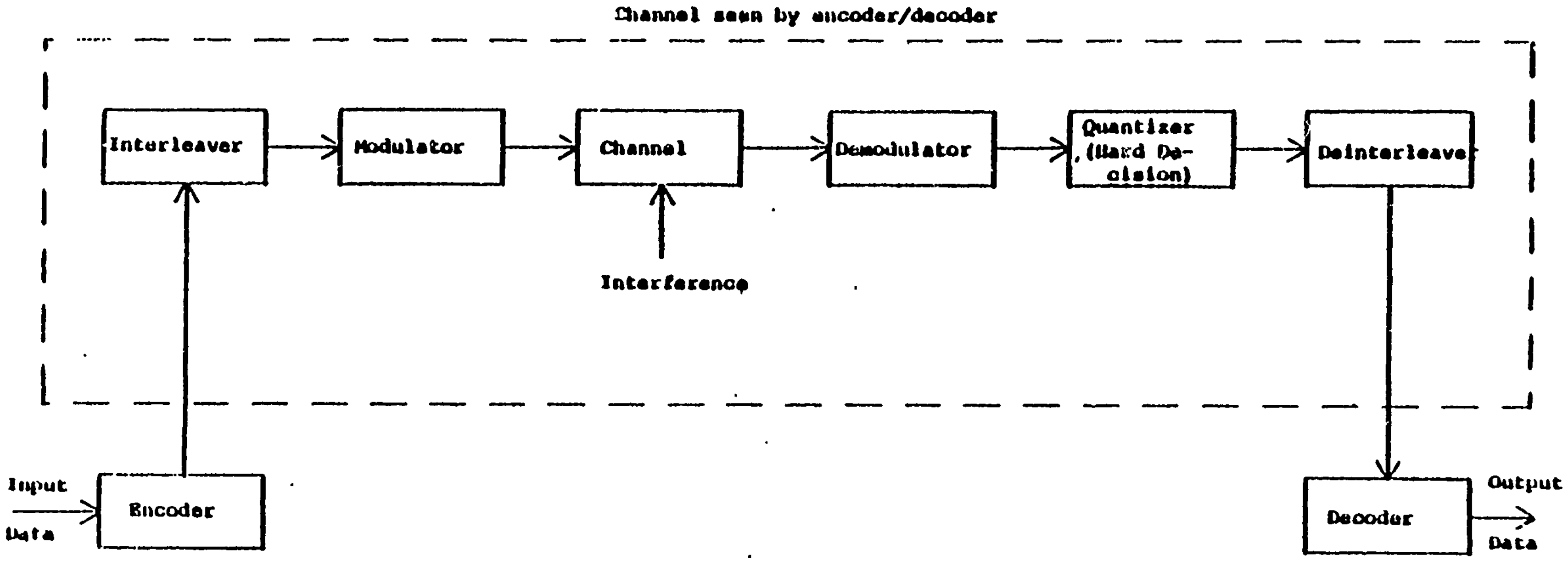


Figure 3.8 Block diagram of a coding system with interleaving/deinterleaving

output is of no help (other than for tracking loop purposes) in determining the phase (0° or 180°) of the transmitted signal.

With coding a decoding decision on a particular information bit is based on several demodular outputs and retaining some amplitude information, rather than just the sign of the demodulator outputs, is helpful. For example, if a particular demodulator output is very large, we can be confident that a polarity decision on that demodulator output is correct; whereas if the demodulator output is almost zero there is a high probability that a polarity decision on that demodulator output would be in error. A decoder that uses this amplitude information to, in effect, weigh the contributions of the demodulator outputs to the decoding decisions can perform better than a similar decoder that only uses the polarity information. A quantizer that retains some amplitude information (i.e., more than one bit is retained) is called a soft quantizer.

No quantization refers to the ideal situation where no quantizer is used at the demodulator output. That is, all of the amplitude information is retained.

In the remainder of this section the effects of demodulator output quantization on coded systems with several modulation/demodulation techniques are discussed.

3.2.2 Additive White Gaussian Noise Channel

3.2.2.1 BPSK or QPSK Modulation

As mentioned in the previous section, using a finer quantization for the demodulator outputs can improve the performance of coded systems. The potential gain in using soft versus hard quantized demodulator outputs can be determined by comparing the E_b/N_o ratios required to operate at $R = R_o$ or $R = C$ for the channels with and without fine quantization.

In the limiting case with no quantization of the demodulator outputs, R_o and C for BPSK modulation on an additive white Gaussian noise channel are [4]

$$R_o = 1-\log_2\left[1 + \exp\left(-R\,\frac{E_b}{N_o}\right)\right] \qquad (3.15)$$

and

$$C = \frac{1}{2}\log_2\left(1+2R\,\frac{E_b}{N_o}\right) \qquad (3.16)$$

where E_b/N_o is the information bit energy-to-noise ratio and the units of R_o and C are bits per binary channel use. The restrictions $R < C$ and $R \leq R_o$ correspond to

$$\frac{E_b}{N_o} > \frac{2^{2R}-1}{2R} \quad \text{for } R < C \qquad (3.17)$$

and

$$\frac{E_b}{N_o} \geq -\frac{1}{R}\,\ln\left(2^{1-R}-1\right) \quad \text{for } R \leq R_o \qquad (3.18)$$

In the limit as the rate approaches zero, the restrictions of (3.17) and (3.18) become

$$\frac{E_b}{N_o} > \ln 2 = -1.59 \text{ dB for } R < C \qquad (3.19)$$

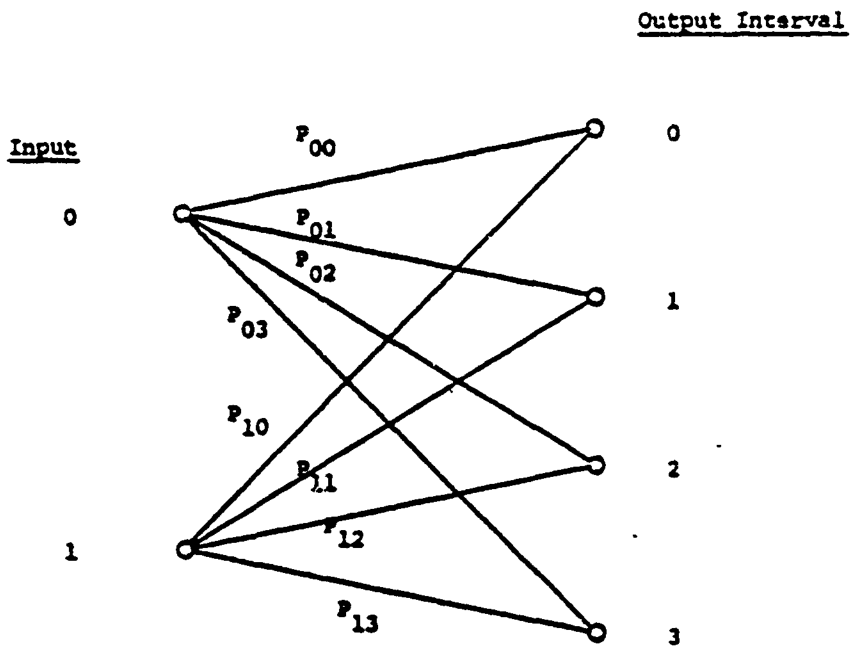
and

$$\frac{E_b}{N_o} \geq 2 \ln 2 = 1.42 \text{ dB for } R \leq R_o \qquad (3.20)$$

So on an additive white Gaussian noise channel any coding technique will require an E_b/N_o of greater than -1.59 dB and for small error rates and a reasonable implementation complexity 1.42 dB will be required regardless of the code rate or of how fine a quantization is used on the demodulator outputs.

Now consider the more realistic channel where the demodulator output is quantized to several bits. Consider an N-bit linear quantizer which has levels of quantization at $0, \pm T, \pm 2T, \ldots, \pm (2^{N-1} - 1) T$ where T is a quantization parameter to be chosen. Such a channel can be represented by a channel transition probability diagram similar to that for the binary symmetric channel of Figure 3.6. Figure 3.9 shows such a diagram for the 2-bit quantized channel.

The R_o and C values for this symmetric N-bit quantized channel are [5].



$$P_{1j} = P_{0,3-j} \quad , j = 0, 1, 2, 3$$

Figure 3.9 Channel transition diagram for a 2-bit output quantized BPSK modulated channel.

$$R_o = - \log_2 \frac{1}{2} \left[1 + \sum_{j=0}^{2^N-1} \sqrt{P_{0j} \; P_{1j}} \right] \qquad (3.21)$$

and

$$C = \sum_{j=0}^{2^N-1} P_{0j} \log_2 \frac{2 \, P_{0j}}{P_{0j}+P_{1j}} \qquad (3.22)$$

where P_{kj} is the probability of the output being in bin j given the input was k. Of course, for $N = 1$, (3.21) and (3.22) reduce to the hard quantized values of (3.13) and (3.14).

For the hard quantized case the probability of a channel error is

$$p = P_{01} = P_{10} = Q \left(\sqrt{2R \frac{E_b}{N_o}} \right) \qquad (3.23)$$

The usual procedure for selecting the quantization parameter T is to choose it to minimize the E_b/N_o required to operate at a code rate of R_o. The justification for this is that by this choice we are, in some sense, maximizing the possible coding gain for codes that operate near R_o. When computer simulations of the coding system are possible, this parameter can be determined based on minimizing the E_b/N_o required for the desired output error rate. Such simulations have sh[illegible] xcellent agreement with the theoretically chosen va[illegible].

For 3-bit quantization and channel symbol energy-to-noise ratios around 1.5 dB this theoretical criterion yields a quantization parameter, T, of .5 to .6 times the standard deviation of the unquantized demodulator outputs. Larger energy-to-noise ratios yield slightly larger T values and smaller energy-to-noise ratios yield slightly smaller T values. In practice, a fixed quantization parameter is usually used for all E_b/N_o ratios. However, an automatic gain control (AGC) is required to estimate the noise variance. Fortunately, we will show that good coding systems exist that are insensitive to small fluctuations in this AGC output.

Figure 3.10 gives curves of the E_b/N_o ratio required to operate at capacity, C, on this BPSK modulated additive-white-Gaussian-noise channel with 1-, 2-, and 3- bits of quantization and no quantization of the demodulator outputs versus the code bandwidth expansion (i.e., one over the rate). Figure 3.11 gives corresponding curves for operation at $R = R_o$. These curves show that 3-bit soft quantization is almost equivalent to no quantization and hard quantization is about 2 dB inferior to no quantization. Comparing the two figures, it can be seen that for the small symbol energy-to-noise ratios, which correspond to the larger bandwidth expansions, about 3 dB more is required to operate at $R = R_o$ than is required to operate at $R = C$.

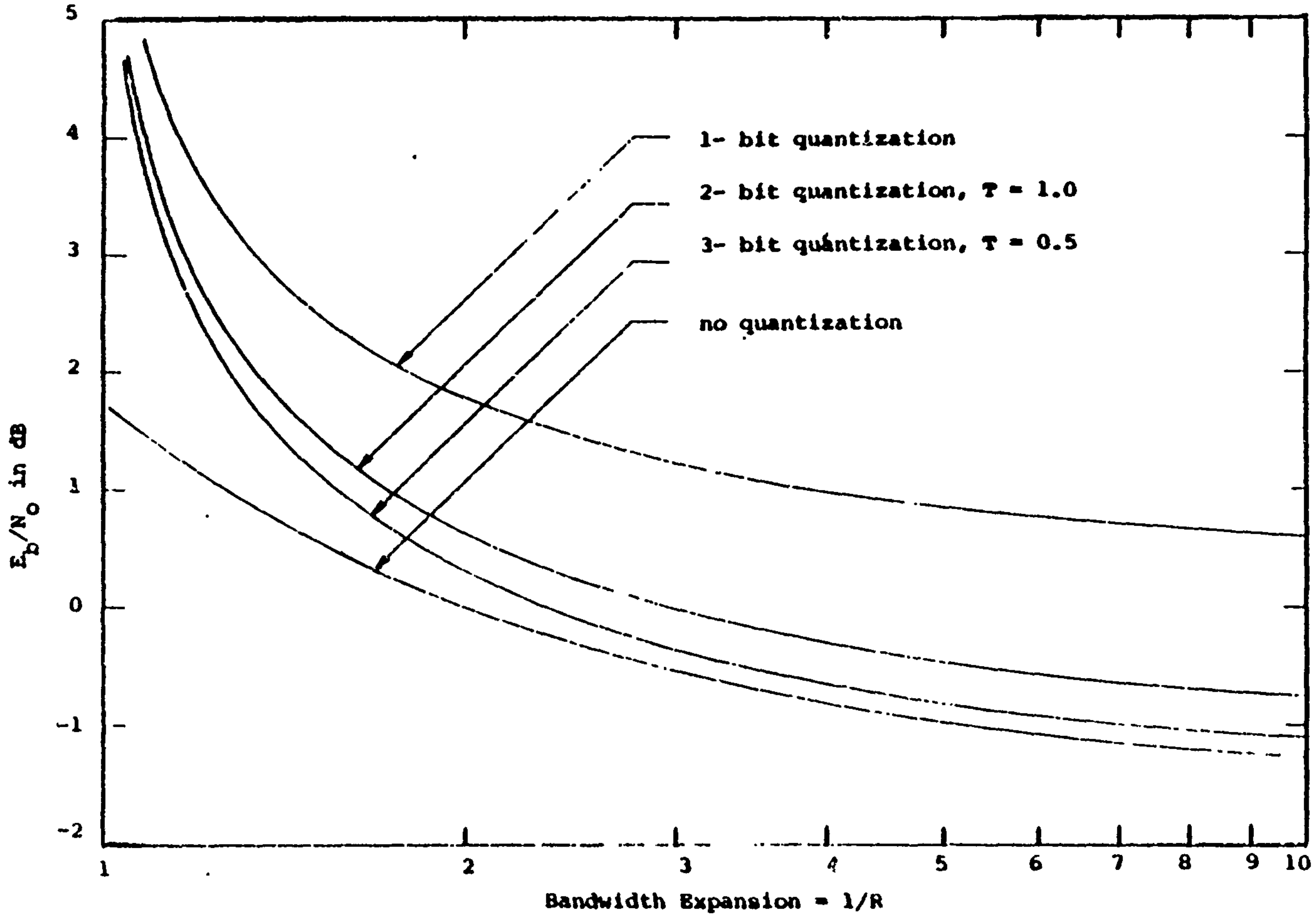


Figure 3.10 E_b/N_o required to operate at channel capacity for a BPSK modulated additive white Gaussian noise channel.

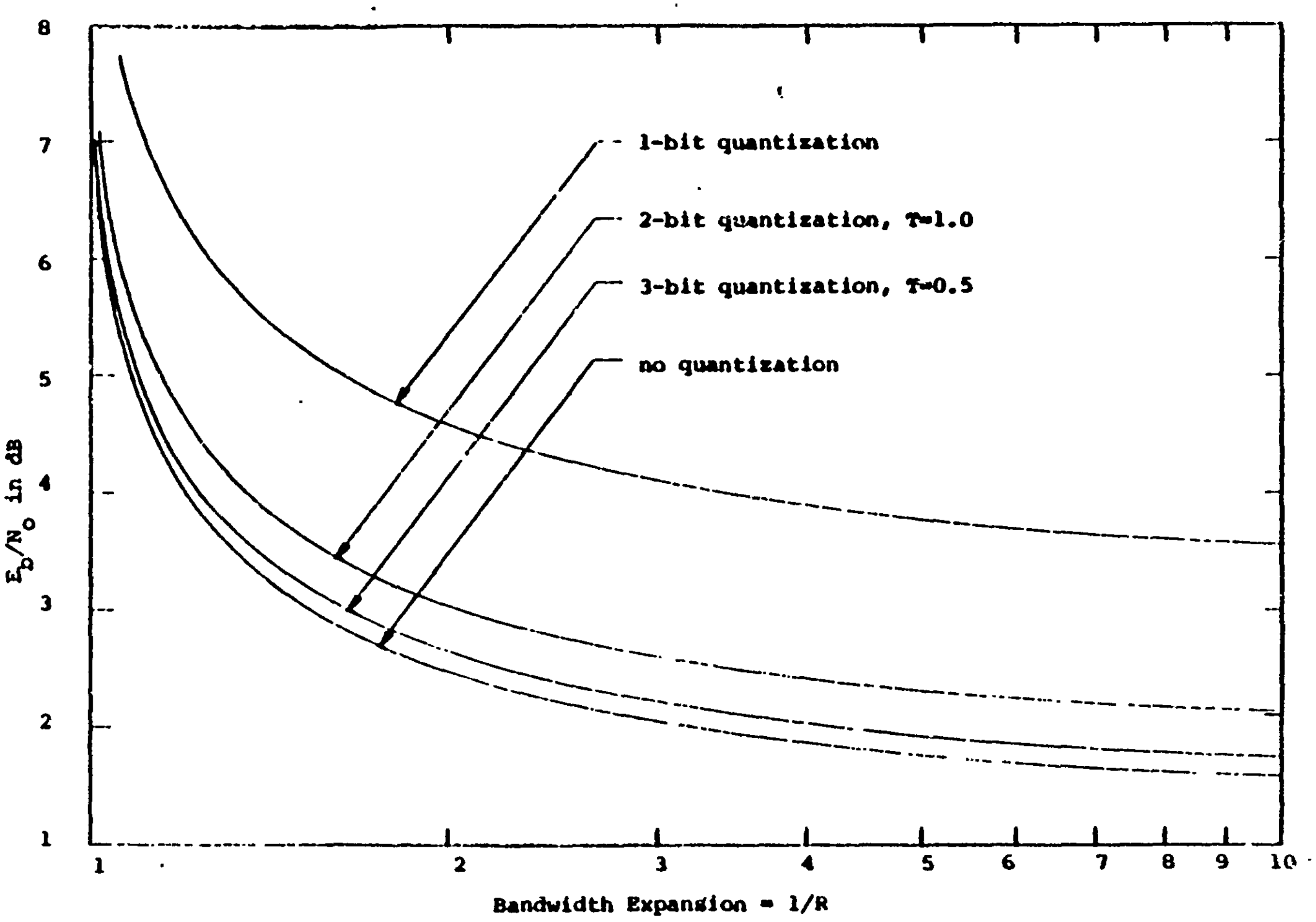


Figure 3.11 E_b/N_o required to operate at $R=R_o$ for a BPSK modulated additive white Gaussian noise channel.

The E_b/N_o required to achieve a 10^{-5} bit error probability with no coding is 9.6 dB. The difference between this value and the curves represents the potential or maximum possible coding gain for that set of conditions.

As with the uncoded perfect phase and time reference case, the QPSK modulated system can be treated as two independent BPSK modulated channels. Thus all the results of this section also apply with perfect reference QPSK modulation.

3.2.2.2 M-ary PSK Modulation

M-ary PSK modulation ($M > 4$) is sometimes used to reduce the bandwidth required for a given data rate relative to the bandwidth required with QPSK modulation. At first glance it may seem a contradiction to consider bandwidth expanding error control coding in such a situation, but we will show that for a small bandwidth expansion relative to the bandwidth required for uncoded QPSK, the E_b/N_o required to operate at $R = R_o$ for a M-ary PSK system can be less than that required for a QPSK system.

With no quantization R_o in units of bits per binary channel use is given by [4]

$$R_o = -\frac{1}{m} \log_2 \frac{1}{M} \sum_{k=1}^{M} \exp\left[- mR \frac{E_b}{N_o} \sin^2 \frac{k\pi}{M} \right] \qquad (3.24)$$

where $M = 2^m$. Since the bandwidth of a PSK system is approximately equal to one over the channel symbol period, the octal- PSK system only requires 2/3 the bandwidth of a QPSK system for the same data rate and a 16-ary PSK system only requires $\frac{1}{2}$ the bandwidth of a QPSK system. Figure 3.12 compares the E_b/N_o ratios required to operate at $R = R_o$ versus the bandwidth expansion relative to an uncoded QPSK system for M = 4, 8, and 16 PSK systems. The larger alphabet sizo systems are seen to have an E_b/N_o advantage for small bandwidth expansions.

Several methods of quantization have been used for the octal-PSK demodulator outputs. One method is to quantize the in-phase and quadrature outputs so that the signal space, consisting of signal components every 45° on a circle of radius $\sqrt{E_s}$, is divided into small squares (see Figure 3.13a). Another method is to divide the received signal space into pie-shaped wedges depending on the angle of the received signal component (see Figure 3.13b). The particular quantization technique will depend on implementation considerations.

The same comparisons presented here for operation at $R = R_o$ can be performed to compare the minimum possible E_b/N_o ratios based on operation at channel capacity.

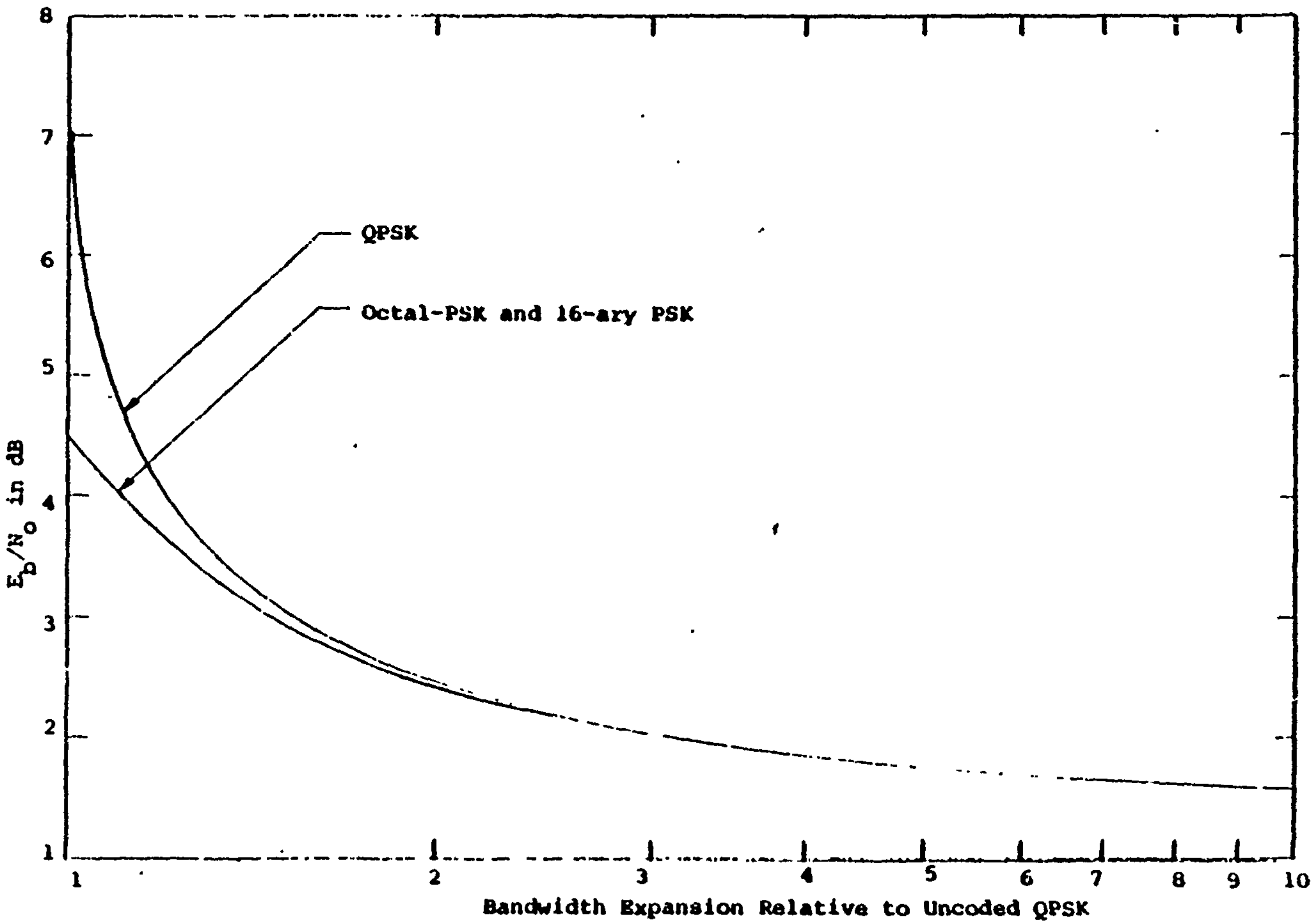
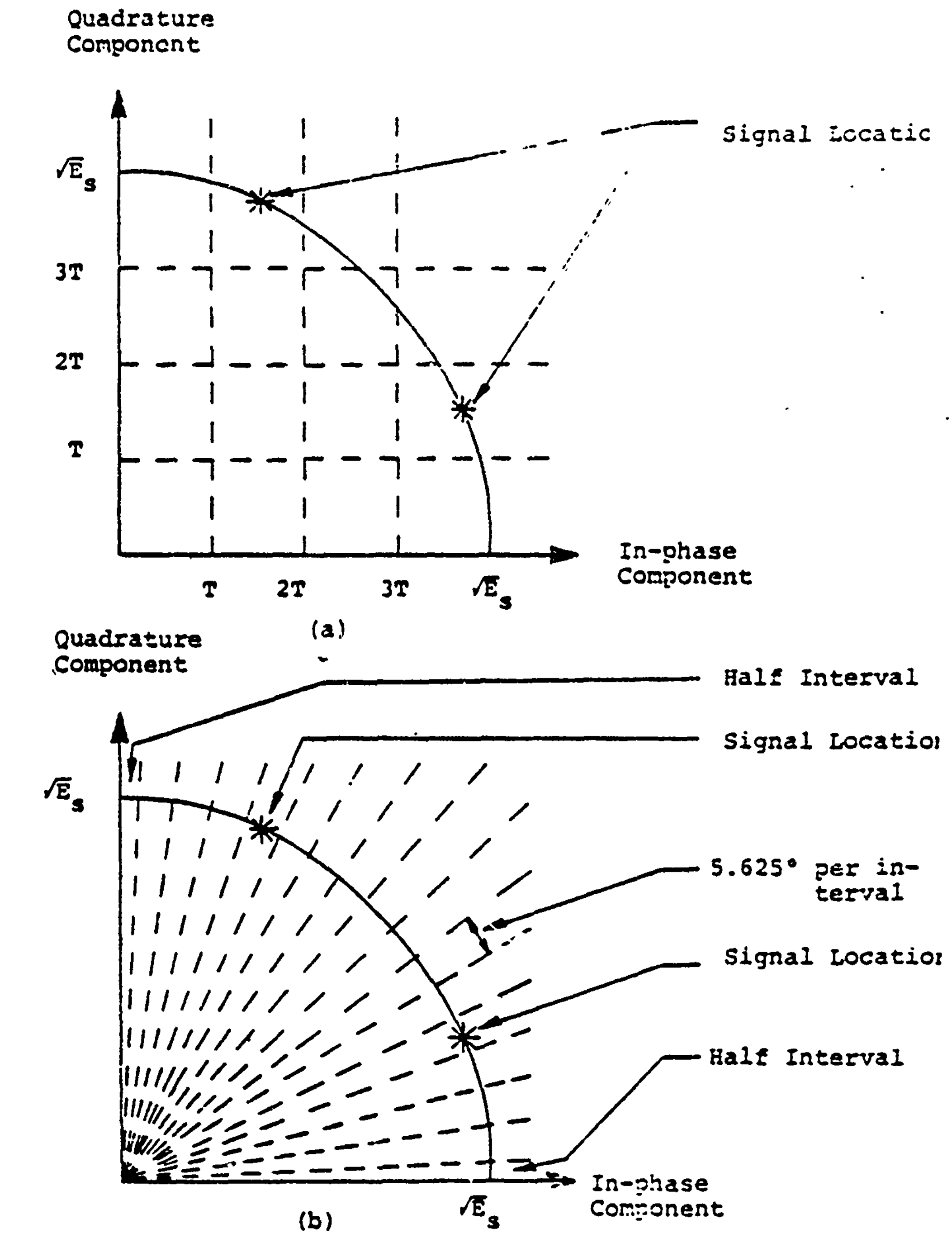


Figure 3.12 E_b/N_o required to operate at $R=R_o$ versus the uncoded QPSK bandwidth expansion for octal-PSK and QPSK with no quantization.



T = quantization parameter

E_s = received symbol energy

Figure 3.13 Diagrams of the first quadrant signal space quantization intervals for two possibl 6-bit quantization techniques for octal-PSK.

However, it is usually more difficult to obtain closed form expressions for C than for R_o. Also for small channel symbol energy-to-noise ratios we have [5]

$$R_o \approx \frac{C}{2} \tag{3.25}$$

So the comparisons based on operation at channel capacity produce approximately the same results as those based on operation at $R = R_o$.

3.2.2.3 DBPSK Modulation

As mentioned previously, differentially coherent phase-shift keying produces a channel with memory. While some codes have been designed aspecially for channels with memory [7-10], the performance of most of the more powerful coding systems are degraded when they are used on such channels. To remedy this situation (i.e., to make the channel appear to be memoryless) simple interleavers can be used as illustrated in Figure 3.8. The potential coding gain of such an interleaved/deinterleaved DBPSK channel is discussed here. The size of the interleavers will depend on the type of coding and will be discussed in later sections.

With no quantization the computational cutoff rate for this channel is [5]

$$R_o = -\log_2 \frac{1}{2}\left\{1 + \int_{-\infty}^{\infty} \sqrt{P(x/\theta = 0)\; P(x/\theta = \pi)}\,dx\right\} \tag{3.26}$$

where $P(x/\theta = \theta_o)$ is the probability density function of the demodulator output random variable given that the phase change of the transmitted symbol from the last to the present symbol is θ_o. For DBPSK the transmitted symbol phase changes are 0 or π radians. These density functions are given in Reference 6. Substituting these density functions in (3.26) yields.

$$R_o = - \log_2 \frac{1}{2} \left\{ 1 + \frac{1}{2} \exp\left(-R \frac{E_b}{N_o}\right) f\left(R \frac{E_b}{N_o}\right) \right\} \qquad (3.27)$$

where

$$f(a) = \int_0^{\infty} \sqrt{e^{-a} \sum_{m=0}^{\infty} \frac{a^m}{m!} e^{-x} \sum_{k=0}^{m} \frac{x^k}{k!}} \, dx \qquad (3.28)$$

The E_b/N_o ratio required to operate at $R = R_o$ can be determined by numeri-ally integrating the integral in (3.28). Of more interest is this E_b/N_o value with implementable quantization techniques.

Figure 3.14 shows this E_b/N_o value versus bandwidth expansion for 1- and 3- bit receiver quantization.

The hard quantization results were obtained using (3.13) with a channel symbol error probability of

$$p = \frac{1}{2} \exp\left(-R \frac{E_b}{N_o}\right) \qquad (3.29)$$

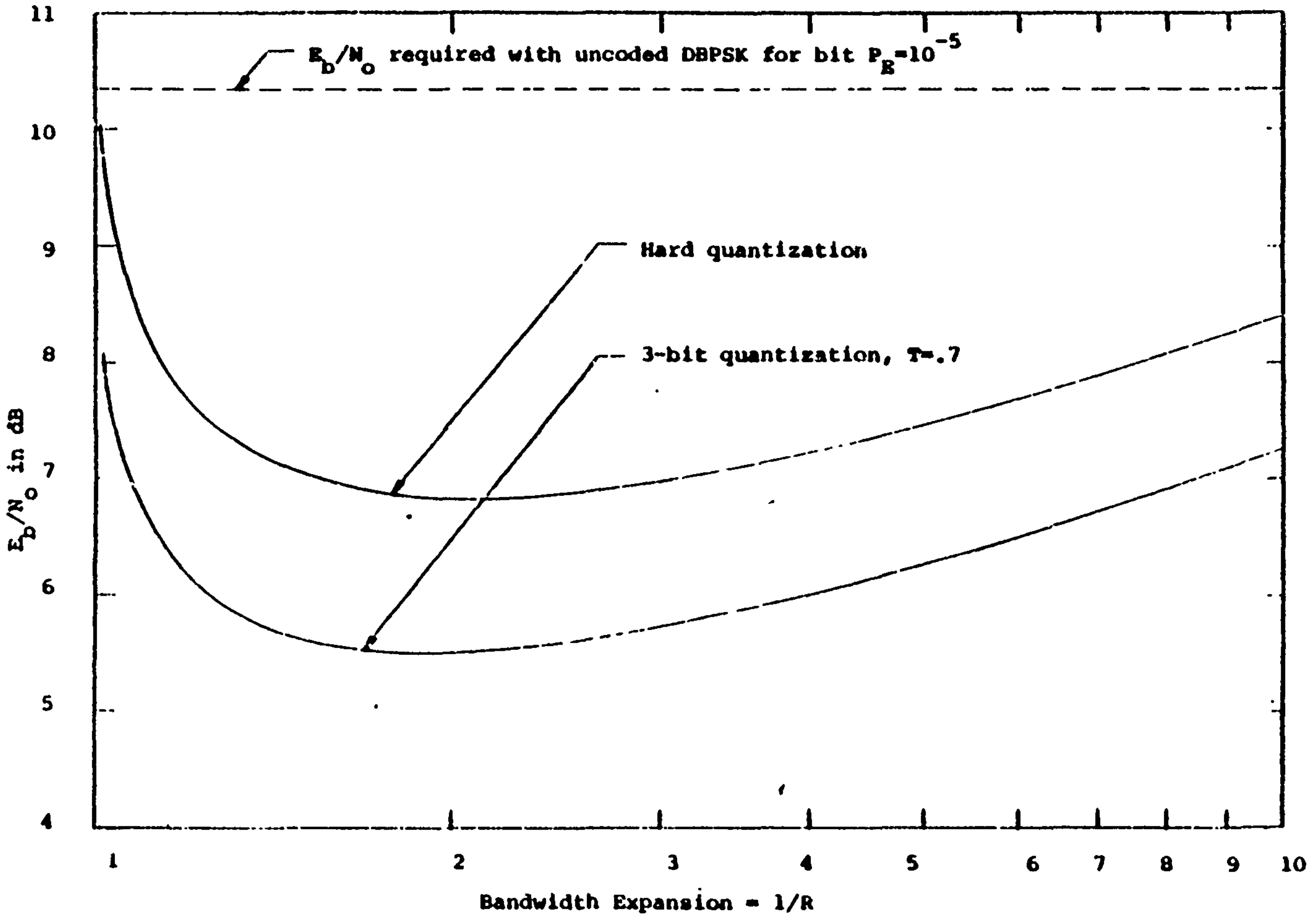


Figure 3.14 E_b/N_o required to operate at $R=R_o$ for an interleaved DBPSK Channel.

The soft quantization results were obtained using (3.21) with $N = 3$ and $P_{ij} = P_{0,\ 7-j}$. The P_{0j} transition probabilities are the probabilities of being in the different quantization intervals given the transmitted symbol phase change is zero. These can be determined from the probability distribution function defined by

$P_0\ (x) =$ Probability that the demodulator output random variable is less than or equal to x given that the transmitted symbol phase change from the last to the present symbol is 0.

Integrating the density function of [6] gives this distribution function

$$P_0\left(x\right) = \begin{cases} 1 - \exp\left(-2R\frac{E_b}{N_o} - 2x\right) \sum\limits_{m=0}^{\infty} \sum\limits_{j=0}^{m} \frac{\left(2R\frac{E_b}{N_o}\right)^m}{m!} \frac{(2x)^j}{j!} \\ + \frac{1}{2} \exp\left(-2R\frac{E_b}{N_o} - 2x\right) \sum\limits_{m=0}^{\infty} \sum\limits_{j=0}^{m} \frac{\left(R\frac{E_b}{N_o}\right)^m}{m!} \frac{(4x)^j}{j!}, & \text{for } x > 0 \\ \frac{1}{2} \exp\left(-R\frac{E_b}{N_o} + 2x\right), & \text{for } x \leq 0 \end{cases}$$

(3.30)

Then the probability of the demodulator output random variable being in the quantization interval betwen T_1 and T_2 ($T_1 < T_2$) is just $P_0(T_2) - P_0(T_1)$. The quantization parameter T was varied to determine the optimum (i.e., the E_b/N_o required to operate at $R = R_o$ was minimized) value and in the regions of primary interest $T \approx .7$ was best. As with the coherent PSK case, R_o is relatively insensitive to small changes in this parameter.

Figure 3.14 shows that lower rate ($R < \frac{1}{2}$) codes will not necessarily improve the coding performance with this type of channel. This somewhat unexpected result can be explained by noting that as the code rate is decreased the channel symbol energy is decreased and thus the phase reference becomes noisier. With the coherent PSK channel a perfect phase reference was assumed. In practice, the non-ideal phase reference will contribute an E_b/N_o loss which will increase as the code rate is decreased.

Figure 3.14 again illustrates the gain which can be achieved with soft quantization of the demodulator outputs. For a rate 1/2 code hard quantization is 1.3 dB inferior to 3-bit soft quantization.

Comparing the DBPSK results of Figure 3.14 with the coherent BPSK results of Figure 3.11 it can be seen th-: the potential coding gain of the DBPSK system is significantly less than that of the BPSK system. Figure 3.15 shows the potential coding gain based on operation at $R = R_o = 1/2$ for BPSK and DBPSK systems.

The minimum possible E_b/N_o ratio determined based on $R < C$ can also be obtained for the hard and 3-bit soft quantization cases using (3.14) and (3.22), respectively. The results are plotted in Figure 3.16.

3.2.2.4 Noncoherently Demodulated MFSK

The most common application of this type of modulation/demodulation is in anti-jamming frequency-hopped systems and time-diversity Rayleigh fading channels in which one of 2^m different, properly spaced, frequencies are used.

With a demodulator consisting of $M = 2^m$ square-law envelope detectors and no quantization the computational cutoff rate for this modulation/demodulation with additive white Gaussian noise can be computed. The result shows that the E_b/N_o ratio required to operate at $R = R_o$ is a monotonically increasing function of the code bandwidth expansion. So such a channel is not an attractive can-

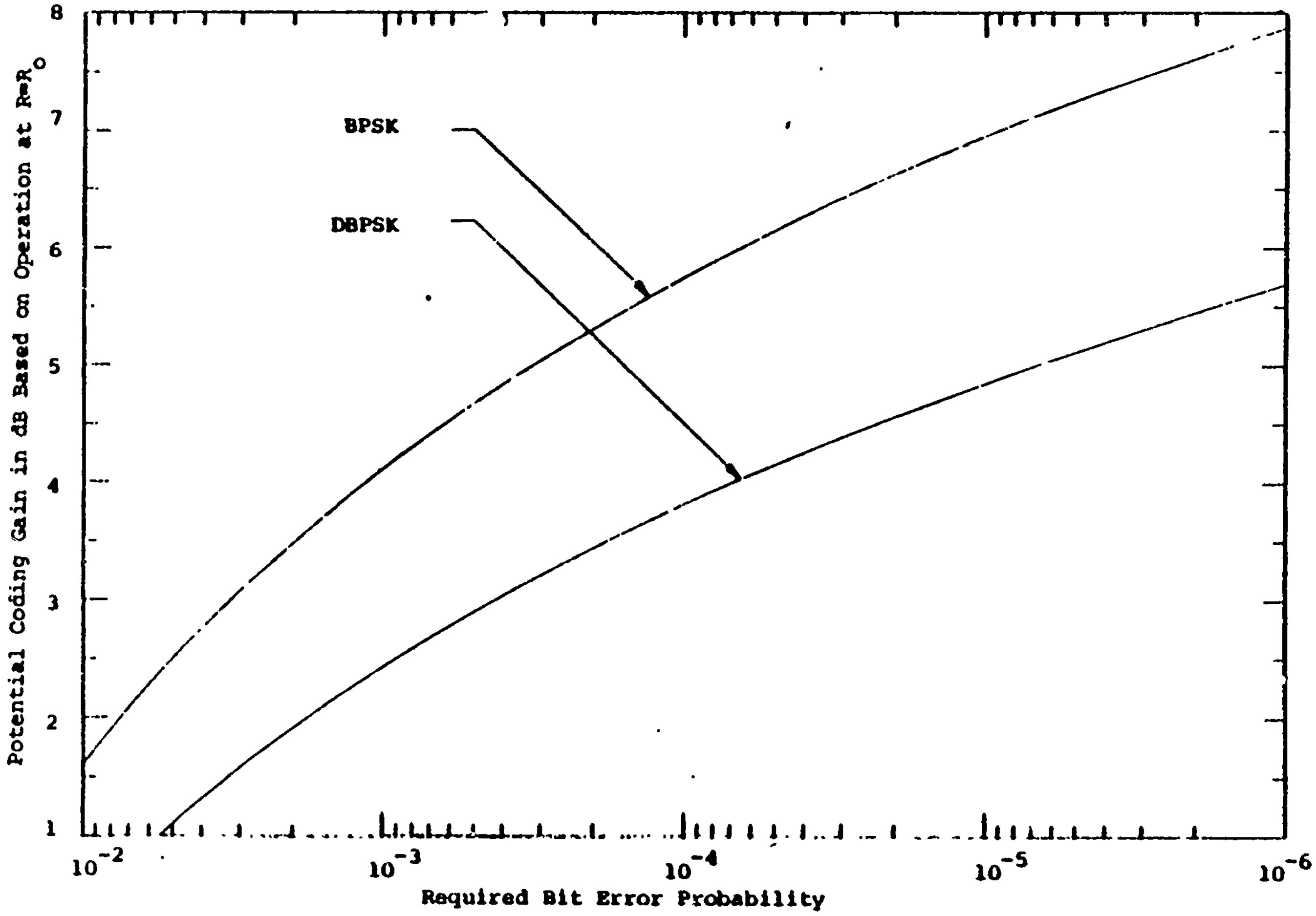


Figure 3.15 Potential coding gain of rate 1/2 coding with BPSK and DBPSK modulation and 3-bit quantization.

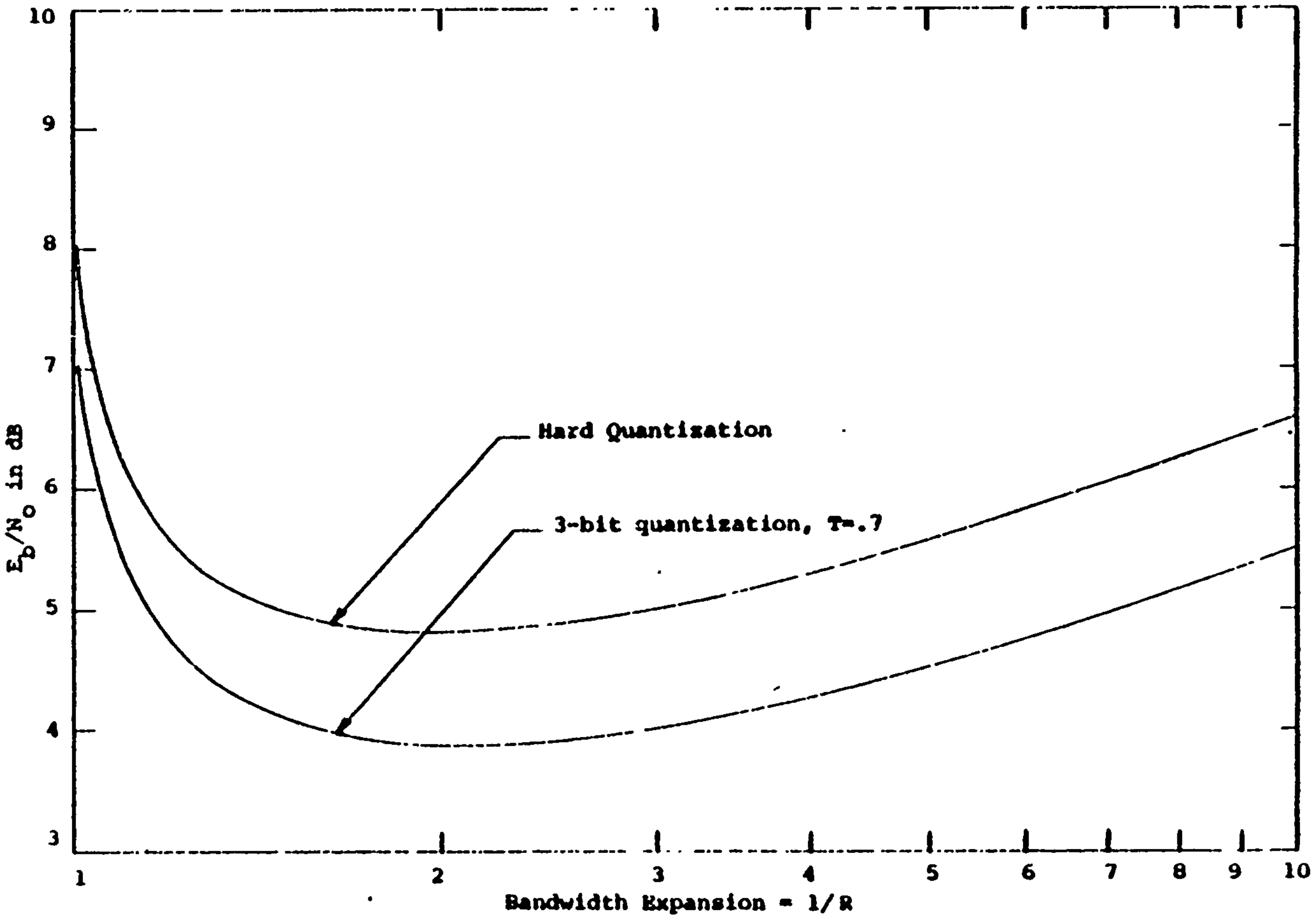


Figure 3.16 E_b/N_o required to operate at R=C for an interleaved DBPSK channel.

didate for error control coding with additive white Gaussian noise. However, this system is very useful in jamming or fading environments.

To implement such a system the matched filter outputs would be quantized. If each matched filter output is quantized to N bits, a total of $N2^m$ bits per received symbol are required. For large m such a system becomes difficult to implement. However, systems with 8 matched filter outputs and 2 bits of quantization per filter have been used effectively in non-Gaussian noise environments with dual-3 convolutional codes. These results will be discused in Section 6.

3.2.3 Independent Rayleigh Fading Channel

As noted in Section 3.1.2, diversity can greatly improve the performance of communication systems on a Rayleigh fading channel. Coding can reduce the diversity requirements (i.e., the order of the diversity) and provide an energy-to-noise ratio coding gain.

The computational cutoff rate for this channel with noncoherently demodulated 2^m-ary MFSK and no quantization is [4]

$$R_o = \frac{1}{m} \log_2 \left[\frac{2^m}{1+4\ (2^m-1)\ p\ (1-p)} \right] \tag{3.31}$$

where

$$p = \frac{1}{2+mR\,\frac{\overline{E_b}}{N_o}} \qquad (3.32)$$

The units of R_o are bits per binary channel use and $\overline{E_b}/N_o$ is the mean information bit energy-to-noise ratio.

Curves of the $\overline{E_b}/N_o$ ratio required to operate at $R = R_o$ versus the bandwidth expansion are given in Figure 3.17 for binary and octal MFSK. This figure also gives this $\overline{E_b}/N_o$ ratio for a hard-quantized binary FSK system. This ratio was obtained using (3.13) with a channel error probability given by (3.32). This figure shows that the difference between the potential performances of soft and hard quantized coding systems on this channel is even larger than on the additive white Gaussian noise (AWGN) channel. On the AWGN channel fo:r a code rate of 1/2 the energy-to-noise ratio difference was 2 dB while here it is 4.9 dB.

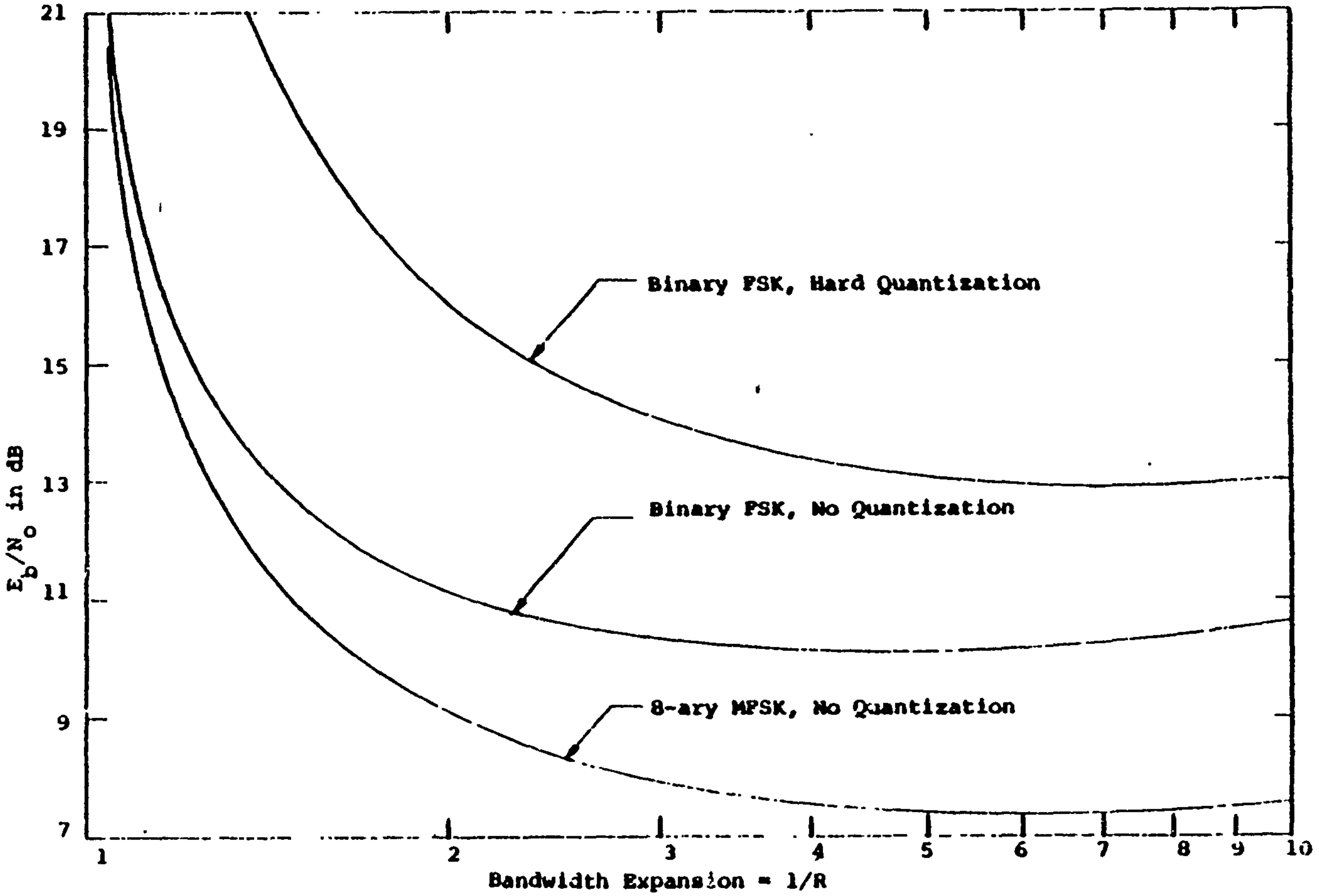


Figure 3.17 $\bar{E}_b/N_o$ required to operate at $R=R_o$ for an independent Rayleigh fading channel with MFSK.

4.0. Block Codes

For this class of codes the data is transmitted in blocks of symbols. For every k encoder input symbols, n-k parity-check symbols are added to produce a total of n symbols to transmit. The code rate is k/n. The more successful block coding techniques have centered about finite-field algebraic concepts.

Linear block codes can be described by a k x n generator matrix $\underline{G}$. If the k-symbol encoder input is represented as a k-dimensional column vector, $\underline{x}$, and the encoder output by an n-dimensional column vector, y, the encoder input-output relationship is given by

$$\underline{y} = \underline{x}\,\underline{G} \tag{4.1}$$

So the n-symbol encoder output blocks are linear algebraic combinations of the rows of the generator matrix. In the binary symbol case, the output blocks are bit-by-bit modulo-2 sums of the appropriate rows of $\underline{G}$.

Usually block codes are decoded using algebraic techniques which require the demodulator to make a hard decision on each received symbol. In Section 3 it was shown that such hard quantization unnecessarily reduces the potential performance of the coding system. For the additive white Gaussian noise channel with BPSK or QPSK modulation the potential coding gain of a finely quantized coding system is about 2 dB more than that of hard quantized system. Recently several soft decision coding techniques have been proposed for block codes [11,12] which, at least for some particular codes, seem to recover most of this 2 dB loss. However, the implementation complexity of such systems is usually much greater than that of the corresponding hard quantized system. In general, when soft decisions are available a convolutional coding technique which easily adapts to soft decisions is preferable.

Another disadvantage of block codes as compared to convolutional codes is that with block codes the receiver must resolve an n-way ambiguity to determine the start of a block whereas with Viterbi-or feedback-decoded convolutional codes a much smaller ambiguity needs to be resolved (see Section 5.1.12).

In spite of these disadvantages, block codes are sometimes useful on channels where only hard decisions are available and the data is presented in blocked format.

Another common application of block codes is for error detection. That is, instead of trying to correct errors the decoder performs the simpler task of just detecting if one or more errors have occurred in the block.

Such error detection systems have been used in applications where a feedback channel is available to tell the transmitter to retransmit the blocks where errors have been detected.

The selection and estimated performance of block codes are usually based on the block distance properties of the code. The distance (sometimes called Hamming distance) between two code words or sequences with an equal number of symbols is defined as the number of positions in which the symbols differ. The minimum distance of the code is defined as the minimum distance between any two different encoder output words (or sequences or blocks). Also the performance and distance properties of linear block codes are independent of the encoder input sequence, so for analysis purpose without loss of generality the all zero sequence is usually assumed to have been transmitted.

For a fixed code rate and block length the goal is to choose a code with a large minimum distance. Then the decoder can more reliably detect or correct channel errors in the received block. A block code with a minimum distance of d_{min} is capable of correcting any combination of $\left[(d_{min}-1)/2\right]_I$ or fewer channel errors or detecting any combination of d-1 or fewer channel errors, where

$[x]_I$ is the integer part of x. However, while the minimum distance of the code may be sufficient to guarantee the detection or correction of a certain number of errors, the particular decoding algorithm may not be capable of such operation.

The performance of block codes with hard receiver quantization is usually determined by assuming that the decoder can correct any combinination of E, $E \leq \left[(d-1)/2\right]_I$, or fewer channel errors and no combination of more than E errors. Such a decoder is called a "bounded distance decoder". Then on a binary symmetric channel the block error probability, P_{block}, is the probability that more than E errors occured. Since there are $\binom{n}{i}$ different ways of having i errors in n symbols, the block error probability is

$$P_{block} = \sum_{i=E+1}^{n} \binom{n}{i} p^i (1-p)^{n-i} \qquad (4.2)$$

The bit error probability depends on the particular code and decoder. Usually block codes are selected to have given block weight properties and codes are called equivalent if they have the same set of block distances (or weights, i.e., number of nonzero encoded block symbols when the all zero input is assumed). However, the bit error probabilities of these so-called equivalent codes may vary. To determine this error probability, assume that the decoder can correct up to E channel errors. These errors are then corrected in the received sequence. The final step is to determine the encoder input block corresponding to the corrected received sequence. This step can be simplified by using a systematic code. Such a code has the property that all the k information symbols are sent unchanged along with n-k parity symbols. In general, every output could depend on every input. It has been shown [5] that for every linear block code there exists a linear systematic block code with the same distance properties. Therefore, systematic block codes are commonly used. We will assume systematic block codes in the remainder of this report.

The bit error probability for this type of decoder with a systematic code can be estimated by assuming that the error rate of the corrected received sequence is equal to the error rate of the encoder input information symbol sequence. Then the bit error probability can be expressed as

$$P_b = \frac{1}{n} \sum_{i=E+1}^{n} \beta_i \binom{n}{i} p^i (1-p)^{n-i} \qquad (4.3)$$

where β_i is the average number of symbol errors remaining in the corrected received sequence given that the channel caused i symbol errors. Of course, $\beta_i = 0$ for $i \leq E$. When $i > E$, β_i can be bounded by noting that when more than E errors occur a decoder which can correct at most E errors would at best correct E of the errors and at worst add E errors. So

$$i - E \leq \beta_i \leq i + E \quad ,i > E \qquad (4.4)$$

The decoder performance can be slightly improved by passing the received sequence unchanged when the corrected received sequence is not a valid code word. In either case for the majority of codes for which β_i has not been determined, $\beta_i = i$ is a good approximation.

When a block code is used for error detection only, the decoder fails to detect an error in the block only when the error sequence transforms the encoded sequence into another valid encoded sequence. By the linearity of the code this implies that the error sequence is equal to a valid code word. This probability of an undetected error can be expressed as

$$P_u = \sum_{i=E+1}^{n} A_i \, p^i (1-p)^{n-i} \qquad (4.5)$$

where A_i is the number of encoded words of weight i (i.e., the number of encoded words with i nonzero symbols). Sometimes it is also of inte.est to determine the probability of not detecting an error under any channel conditions. On a binary symmetric channel the worst channel is when the channel error probability is 1/2. Substituting p= 1/2 in (4.5) and using the fact that there are 2^k-1 codewords of weight E+1 to n and one codeword of weight zero gives

$$P_u \bigg|_{p = 1/2} = (2^k - 1)2^{-n} < 2^{-(n-k)}$$

This bound is sometimes used as an upper bound on the undetected error probability for any channel error rate. While this is valid for Hamming codes, it is not true in general [13]. Nevertheless, the $P_u < 2^{-(n-k)}$ bound gives a first approximation to this worst case undetected error probability.

When the information and encoded symbols of a block code are from some nonbinary alphabet and the probability of any channel input symbol being changed to any other symbol is the same for any nonidentical input-output symbol combination and p is taken to be the channel symbol error probability, then (4.2) and (4.5) still apply and the bit error probability of (4.3) becomes the symbol error probability.

The block code error probability formulas presented thus far have been for hard receiver quantization. Decoders capable of using soft quantized inputs are possible, but they are more difficult to implement. The simplest type of soft quantization is one in which three, rather than two, quantization intervals are used. This additional interval could be used to erase unreliable symbols. Forney [16] shows how the hard quantization block decoding techniques can be extended to take advantage of such a receiver quantization scheme. This type of decoding is sometimes called erasure-and-errors decoding.

If the decoder does erasure-and-errors decoding, the code minimum distance is d, and the maximum number of errors that the decoder can correct is E, then a decoded error occurs when the number of errors t and the number of erasures s satisfy $2t+s \geq d$ or $t \geq E+1$. So if the probability of an erasure is p_x and the probability of a channel error is p_E, then the block error probability is [16]

$$P_{block} = \sum_{t=0}^{E} \sum_{s=d-2t}^{n-t} \binom{n}{s,t} p_E^t \; p_x^s \left(1-p_E-p_x\right)^{n-s-t} + \sum_{t=E+1}^{n} \binom{n}{t} p_E^t \left(1-p_E\right)^{n-t} \qquad (4.6)$$

where

$$\binom{n}{s,t} = \frac{n!}{s!\,t!\,(n-s-t)!}$$

In the following subsections the structure and performance of several specific block codes are examined.

4.1. Hamming Codes

Hamming codes are the simplest nontrival class of codes with $n = 2^m - 1$ ($m = 2, 3, \ldots$) encoder output symbols for each block of $k = n-m$ input symbols. These codes have a minimum distance of 3 and thus are capable of correcting all single errors or detecting all combinations of 2 or fewer errors. Although Hamming codes are not very powerful they belong to a very limited class of block codes called perfect codes. An e-error-correcting, $e = [(d-1)/2]_I$, code is called a perfect code if every n-symbol sequence is at a distance of at most e from some n-symbol encoder output sequence.

Hamming codes are usually described in terms of an $n \times (n-k)$ dimensional parity check matrix [5], $\underline{H}$, with the property that for each n-dimensional encoded output word $\underline{y}$

$$\underline{y}\,\underline{H} = \underline{0} \qquad (4.7)$$

For Hamming codes the n rows of the parity check matrix are equal to all positive nonzero m-bit sequences. Given a parity check matrix, a generator matrix can be determined [5].

If the binary additive noise sequence is represented by an n-dimensional vector $\underline{z}$, then the received signal is

$$\underline{y} = x\,\underline{G} \oplus \underline{z} \qquad (4.8)$$

where ● denotes bit-by-bit modulo-2 addition.

Decoding is accomplished by multiplying this binary vector by the parity chack matrix to form an n-k = m dimensional syndrome vector $\underline{S}$. Using (4.7) we have

$$\underline{S} = \underline{y}\,\underline{H} = \underline{x}\,\underline{G}\,\underline{H} \bullet \underline{z}\,\underline{H} = \underline{z}\,\underline{H} \qquad (4.9)$$

Because of the form of H, this m-bit syndrome specifies the locations of any single error which can then be corrected. If the syndrome is zero, the decoder assumes no errors occurred.

The weight distribution of Hamming codes have been determined. Expressed as a polynomial this distribution for the binary case is [14]

$$\begin{aligned} A\left(z\right) &= \sum_{i=0}^{n} A_i\, z^i \\ &= \frac{1}{n+1}\left[\left(1+z\right)^n + n\left(1+z\right)^{\frac{n-1}{2}}\left(1-z\right)^{\frac{n+1}{2}}\right] \qquad (4.10) \end{aligned}$$

where A_i is the number of code words of weight i. This weight enumerator polynomial makes the computation of the undetected error probability possible.

From (4.2), (4.3) and (4.5) the block, bit, and undetected error probabilities are

$$P_{block} = \sum_{i=2}^{n} \binom{n}{i} p^{i} \left(1-p\right)^{n-i}$$

$$= 1-\left(1-p\right)^{n} - n\, p \left(1-p\right)^{n-1} \qquad (4.11)$$

$$P_{b} = \frac{1}{n} \sum_{i=2}^{n} i \binom{n}{i} p^{i} \left(1-p\right)^{n-i}$$

$$= p - p\left(1-p\right)^{n-1} \qquad (4.12)$$

and

$$P_{u} = \sum_{i=1}^{n} A_{i}\, p^{i} \left(1-p\right)^{n-i}$$

$$= \left(1-p\right)^{n} \left[A\left(\frac{p}{1-p}\right) - 1\right] \qquad (4.13)$$

where (4.12) uses the $\beta_i = i$ approximation for $i > 1$.

Figures 4.1, 4.2, and 4.3 show these probabilities versus the channel error rate for $m = 3, 4$, and 5. The channel error rates can be determined from Section 3.1 for the binary memoryless channel case. With interleaving the bit error probability versus E_b/N_o results for fading channels or for differentially coherent or nonbinary modulation can also be used. Just interpret the bit error probability versus E_b/N_o results of Section 3.1 as the channel error rate versus the channel symbol energy-to-noise ratio,

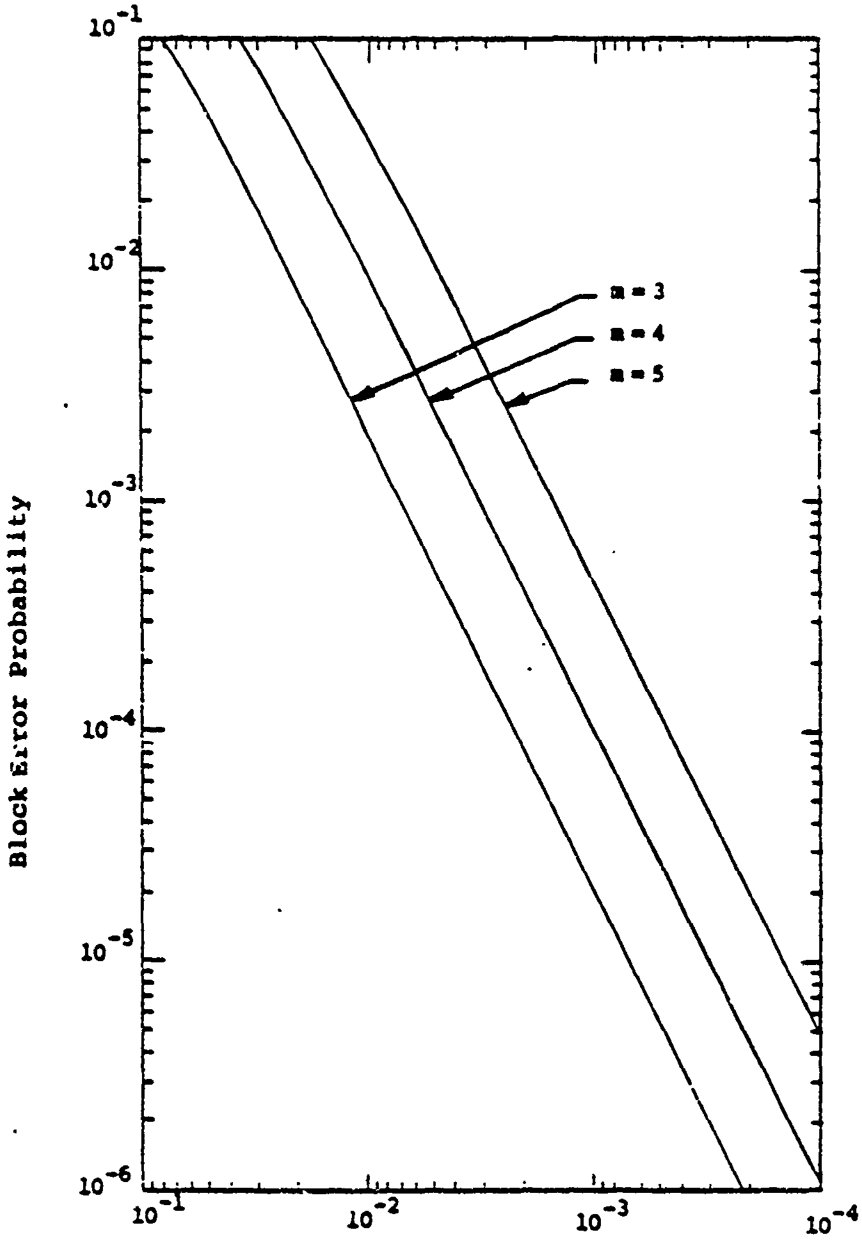


Figure 4.1 Block error probability versus channel error probability for block length $n = 2^m-1$ Hamming codes with $m = 3$, 4, and 5.

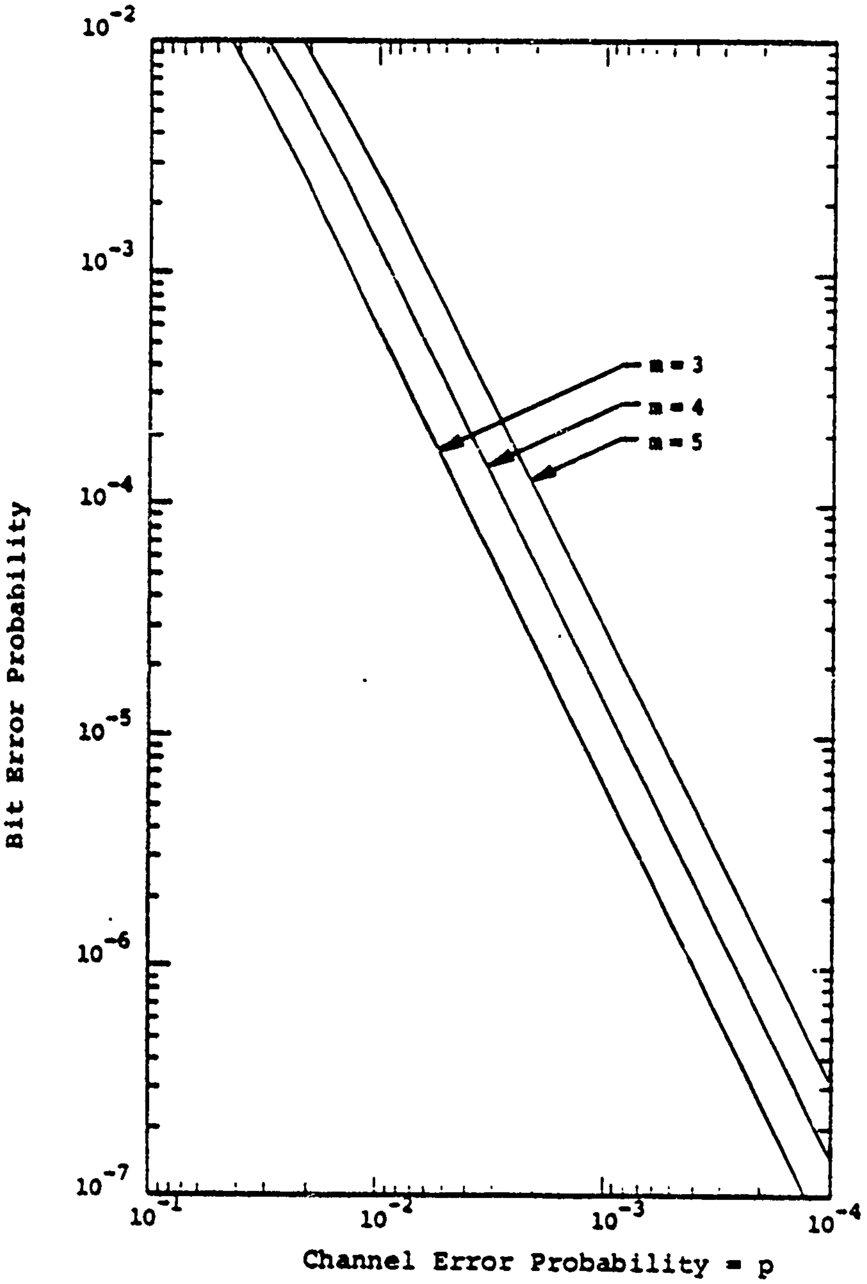


Figure 4.2 Bit error probability versus channel error probability for block length $n = 2^m - 1$ Hamming codes with $m = 3, 4$, and 5.

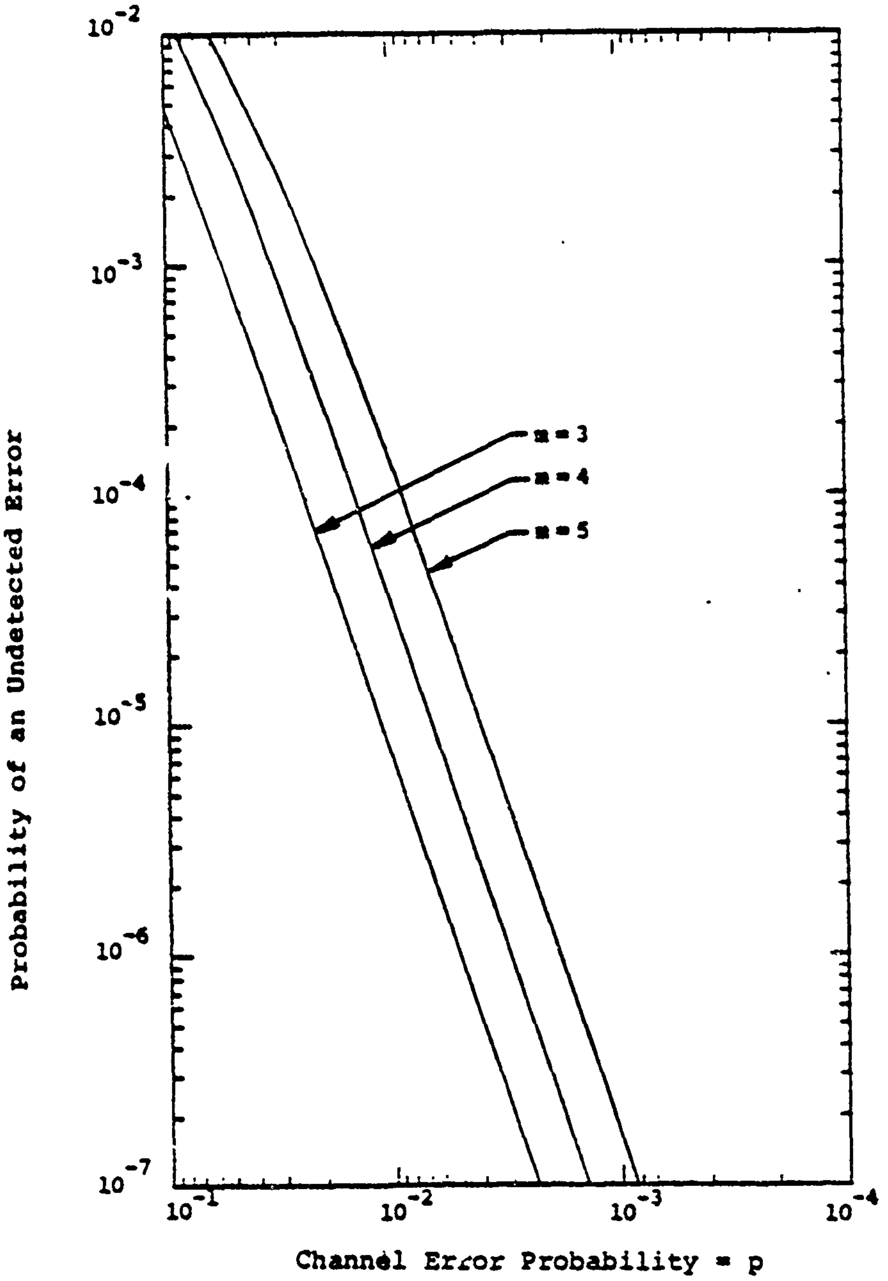


Figure 4.3 Probability of an undetected error versus channel error probability for block length $n = 2^m - 1$ Hamming codes with $m = 3$, 4, and 5.

E_s/N_o. To obtain the coded system information bit energy-to-noise ratio use

$$\frac{E_b}{N_o} = \frac{1}{R} \frac{E_s}{N_o} \qquad (4.14)$$

For Hamming codes (4.14) becomes

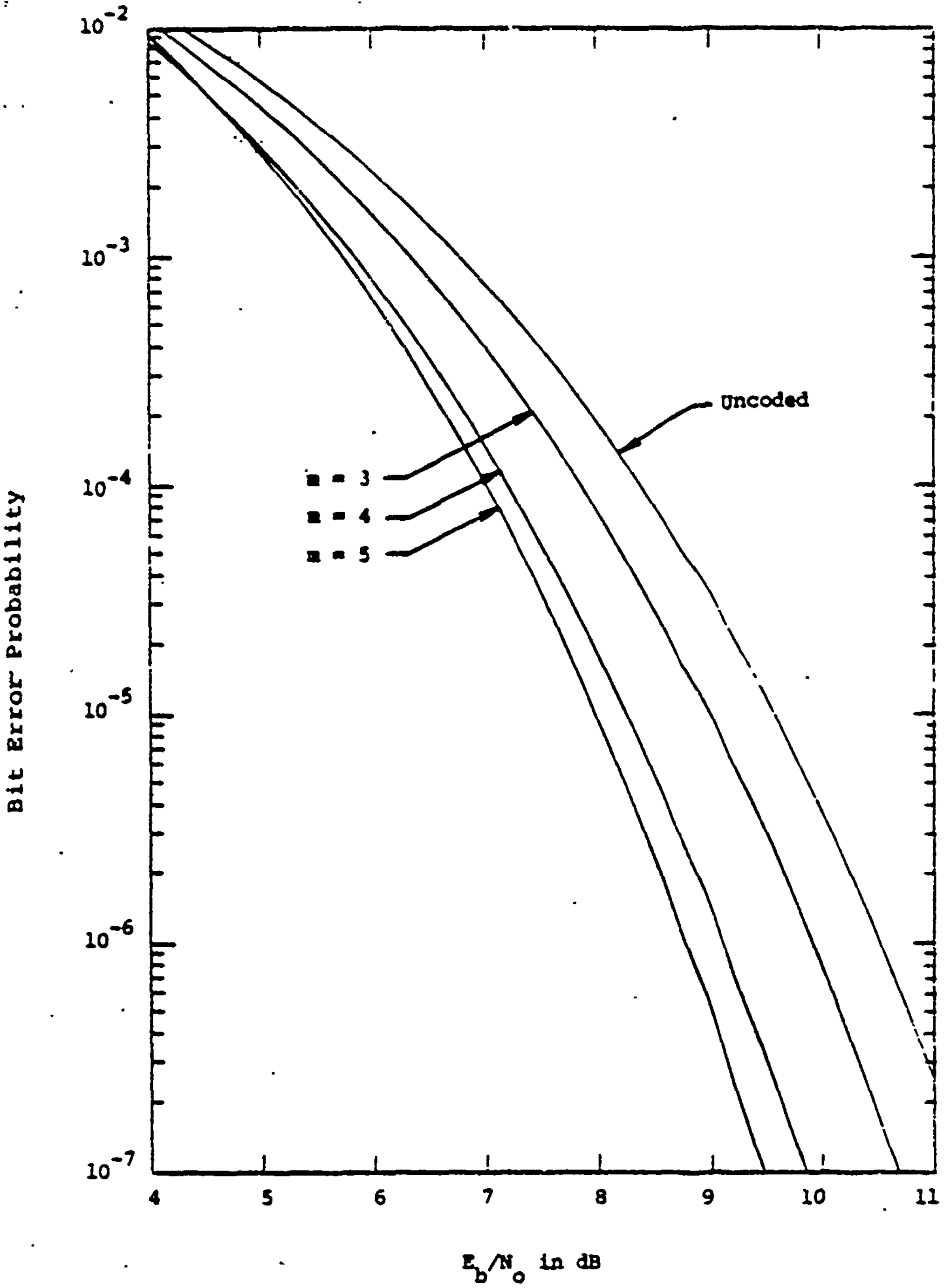
$$\frac{E_b}{N_o} = \frac{2^m-1}{2^m-1-m} \frac{E_s}{N_o} \qquad (4.15)$$

With these changes the probabilities of Figures 4.1, 4.2, and 4.3 can be plotted versus E_b/N_o. The bit error probability result for an additive white Gaussian noise channel is given in Figure 4.4. The coding gain can be determined as the reduction in the E_b/N_o required to achieve a specified error probability with the coded system as compared to the uncoded system.

Table 4.1 summarizes the E_b/N_o ratios required to achieve a 10^{-5} bit error probability with Hamming coding for several modulation/demodulation techniques with additive white Gaussian noise (AWGN) and Rayleigh fading.

4.2 Extended Golay Code

One of the more useful block codes is the binary n = 24, k = 12, i.e, (24,12) extended Golay code formed by adding an overall parity bit to the perfect (23,12) Golay code. This parity bit increases the minimum distance



E_b/N_o in dB

Figure 4.4 Bit error probability versus E_b/N_o for block length $n = 2^m - 1$ Hamming codes with $m = 3$, 4, and 5 on an AWGN channel.

Type of Interference	Modulation/demodulation technique	E_b/N_o in dB required to achieve a 10^{-5} bit error probability		
		m = 3	m = 4	m = 5
AWGN	BPSK or QPSK	9.0	8.3	8.0
AWGN	Octal-PSK with interleaving	12.2	11.5	11.2
AWGN	DBPSK with interleaving	10.2	9.4	9.1
AWGN	DQPSK with Interleaving	11.4	10.7	10.4
Independent Rayleigh Fading	Noncoherently demodulated binary FSK with optimum diversity (see Figure 3.5)	17.5 (L=8 diversity)	16.7 (L=16 diversity)	16.4 (L=16 diversity)

Table 4.1 Summary of the E_b/N_o ratios required to achieve a 10^{-5} bit error probability with Hamming coding for several modulation/demodulation techniques.

of the code from 7 to 8 and produces a rate 1/2 code which is easier to work with than the rate 12/23 of the (23,12) code.

Extended Golay codes are considerably more powerful than the Hamming codes of the previous section. The price for the improved performance is a more complex decoder and a lower rate, and hence a larger bandwidth expansion. Decoding algorithms which make use of soft decision demodulator outputs have also been proposed for these codes [11,12]. When such soft decision decoding algorithms are used the performance of the extended Golay code is similar to that of a simple Viterbi-decoded convolutional coding system of constraint length about 5 (see Section 5.1.10). While it is difficult to compare the implementation complexity of two different coding systems, it can be concluded that when only hard decision demodulator outputs are available, extended Golay coding systems are of the same approximate complexity as similar performance convolutional coding systems. However, when soft decisions are available convolutional coding is superior.

The hard-decision block and bit error probability expressions of Section 4.0 assumed that the decoder was capable of correcting any combination of E or fewer errors and no combination of more than E errors. With perfect codes, such as the Hamming codes, with $E = [(d-1)/2]_I$

this is always the case. However, with an extended Golay code the decoder could be designed to correct some but not all 4-error patterns. Usually, in order to simplify the decoder implementation the decoder is implemented in such a way that these 4-error patterns cannot be corrected. Since for extended Golay codes only 19% of the 4-error patterns can be corrected we will assume the decoder cannot correct these 4-error patterns. Then the block, bit, and undetected error probabilities for hard-decision decoding can be determined from (4.2), (4.3) and (4.5). The results are

$$P_{block} = \sum_{i=4}^{24} \binom{24}{i} p^i (1-p)^{24-i} \tag{4.16}$$

$$P_b = \frac{1}{24} \sum_{i=4}^{24} \beta_i \binom{24}{i} p^i (1-p)^{24-i} \tag{4.17}$$

and

$$P_u = (1-p)^{24} \left[A\left(\frac{p}{1-p}\right) - 1 \right] \tag{4.18}$$

where the β_i coefficients and the coefficients of the weight enumerator polynomial, A, are given in Table 4.2.

Figure 4.5 gives these probabilities versus the channel error rate. As in the previous section the channel

i	A_i = Number of Code Words of Weight i	β_i
0	1	0
1-3	0	0
4	0	4
5	0	8
6	0	120/19
7	0	8
8	759	8
9	0	2637/323
10	0	3256/323
11	0	3656/323
12	2576	12
13	0	4096/323
14	0	4496/323
15	0	5115/323
15	759	16
17	0	16
18	0	336/19
19	0	16
20	0	20
21-23	0	24
24	1	24

Table 4.2 Weight enumerator and β_i coefficients for the extended Golay code [12].

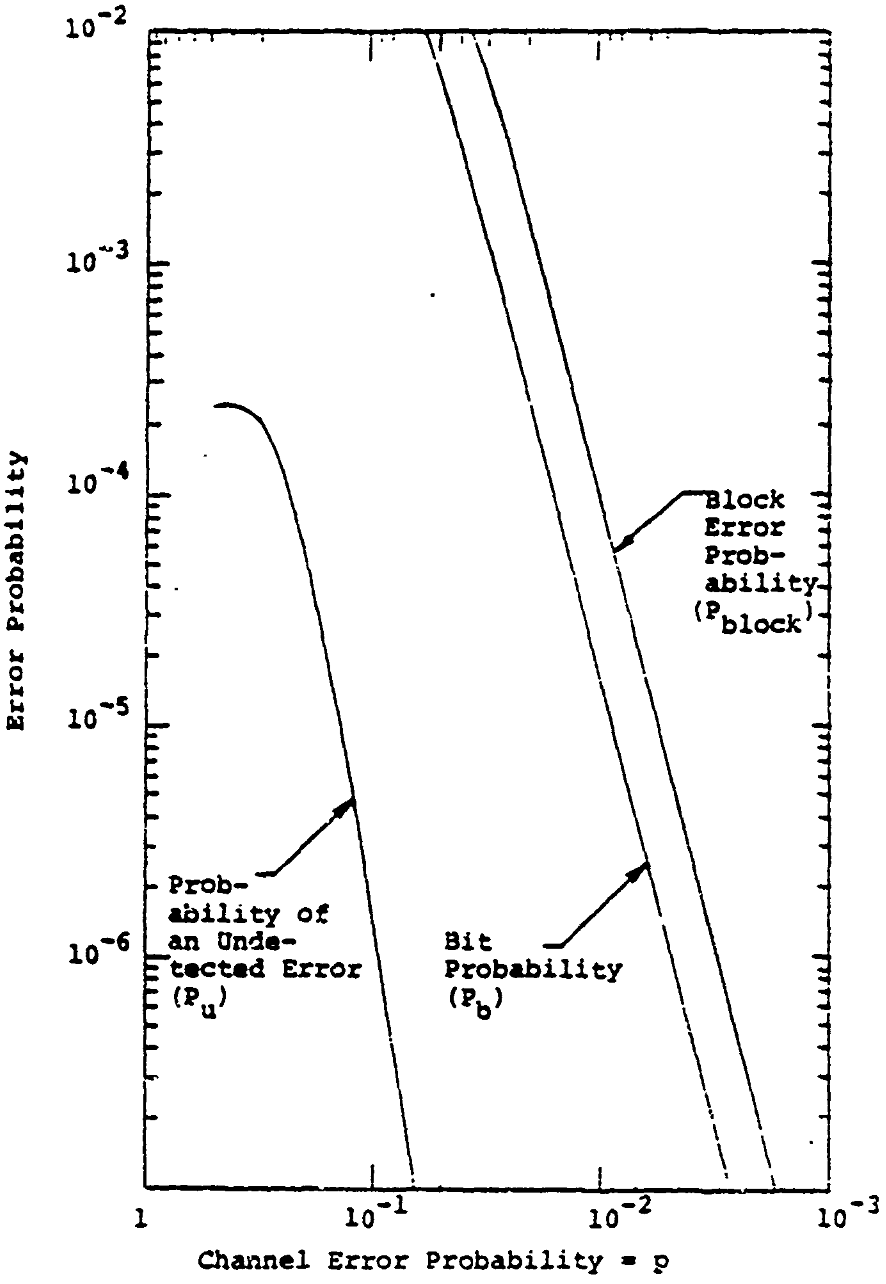


Figure 4.5 Block, bit, and undetected error probabilities versus channel error rate with extended Golay coding.

error rate can be determined from the results of Section 3.1. Using these results the hard-decision coding performance for coherently demodulated BPSK or QPSK on an AWGN channel are given in Figure 4.6. Figure 4.6 also shows the bit error probability obtained with a soft-decision decoding algorithm proposed in [11]. The soft-decision algorithm is seen to recover most of the 2 dB hard quantization loss determined in Section 3.2.2.

Table 4.3 gives the E_b/N_o ratios required to obtain a 10^{-5} bit error rate with the extended Golay coding and several different modulation/demodulation techniques for AWGN and Rayleigh fading channels.

4.3 BCH Codes

Bose-Chaudhuri-Hocquenghem (BCH) codes are a powerful class of codes which have well defined decoding algorithms. A large selection of block lengths, code rates, alphabet sizes, and code minimum distances are possible. The most common codes use a binary alphabet, an encoder output block length of $n = 2^m-1$ (m a positive integer), and, of course, the largest possible code minimum distance.

A detailed description of BCH codes requires elaborate algebraic developments and is beyond the scope of this report. The main point is that while a description of these codes and their decoding algorithms is somewhat

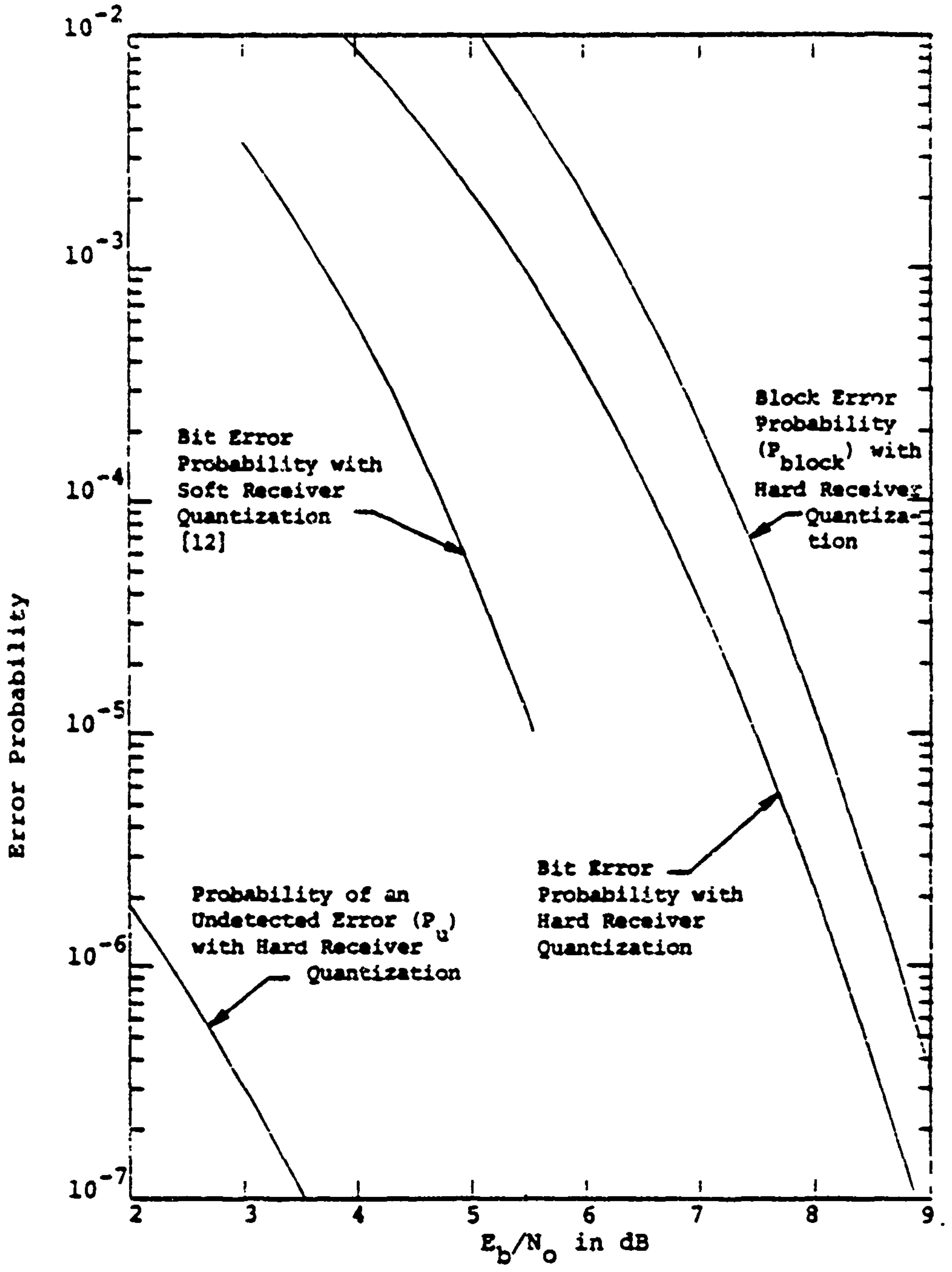


Figure 4.6 Block, bit, and undetected error probabilities versus E_b/N_o for BPSK or QPSK modulation, an AWGN channel, and extended Golay coding.

Type of Interference	Modulation/demodulation technique	$\frac{E_b}{N_o}$ in dB required to achieve a 10^{-5} bit error rate
AWGN	BPSK or QPSK	7.5 (hard decision), 5.6 (soft decision)
AWGN	Octal-PSK with inter-leaving	10.4 (hard decision)
AWGN	DBPSK with interleaving	9.0 (hard decision)
AWGN	DQPSK witn interleaving	10.0 (hard decision)
Independent Rayleigh fading	Noncoherent binary FSK with optimum diversity (i.e., L=8 or 16 channel bits per information bit with rate 1/2 coding)	15.9 (hard decision)

Table 4.3 E_b/N_o required to achieve a 10^{-5} bit error rate with extended Golay coding and several modulation/demodulation techniques.

complicated, the actual decoder can be readily implemented. Here we will just outline the decoding procedure and indicate the techniques for determining their error rate performance.

Reference 14 gives tables of the BCH code minimum distance, d_{BCH}, for a wide varity of encoder input and output block lengths. The actual minimum distance of the code may be slightly larger than the BCH minimum distance, but the algebraic decoding algorithms treat the code as if it had the BCH minimum distance.

The block, bit, and undetected error probabilities can be determined from (4.2), (4.3), and (4.5) with

$$E_{BCH} = \left[\left(d_{BCH} - 1\right) / 2\right]_I \qquad (4.19)$$

For most codes the weight enumerator polynomial coefficients of (4.5) are not known. So small channel error rate approximations to it are usually obtained using only the first one or two terms of the summation. The weight enumerator coefficients for these first few terms can usually be determined or estimated.

The decoding of these codes involves basically four steps [5].

(1) Calculate d_{BCH-1} syndromes. These syndromes are computed using the same general approach as described in Section 4.1.

(2) Find the coefficients for an e-degree error locator polynomial where e, $e \leq E_{BCH}$, is the number of channel errors. The technique for doing this is referred to as the Berlekamp Algorithm. This polynomial has the significance that its roots give the locations of the channel errors in the received block of symbols.

(3) Find the roots, and thus the locations of the errors, of the error locator polynomial. The usual technique for doing this is referred to as the Chien Search [5]. It involves checking each of the n code symbol locations to see if that location corresponds to a root of the error locator polynomial.

(4) Find the values of the errors. With binary codes the errors can be corrected by complementing the present symbol. With nonbinary symbols a simple formula is available [5].

This algebraic decoding procedure uses hard quantized demodulator outputs and thus gives up some potential coding gain on channels where soft decisions are available. While extension to soft decision is possible, with the same techniques used for Golay codes, the complexity increases substantially.

Figure 4.7 illustrates the bit error probability versus channel bit error rate performances that can be achieved with block length n = 127 codes capable of correcting 5, 10, and 15 channel errors. The results were obtained using (4.3) with $\beta_i = i$. The largest possible number of information bits per block for the 5, 10, and 15 error-correcting BCH codes of Figure 4.7 are k = 92, 64, and 36, respectively [41].

One special type of BCH code worthy of further note is the class of Reed-Solomon codes discussed in the next section.

4.4. Reed-Solomon Codes

Reed-Solomon Codes are a particularly interesting and useful class of nonbinary BCH codes which achieve the largest possible code minimum distance for any linear code with the same encoder input and output block lengths. For nonbinary codes the distance between two code words is defined as the number of nonbinary symbols in which the sequences differ. For Reed-Solomon codes the code minimum distance is given by [5]

$$d = n+1-k \qquad (4.20)$$

An E-error-correcting Reed-Solomon code with an alphabet of 2^m symbols has

$$n = 2^m-1 \qquad (4.21)$$

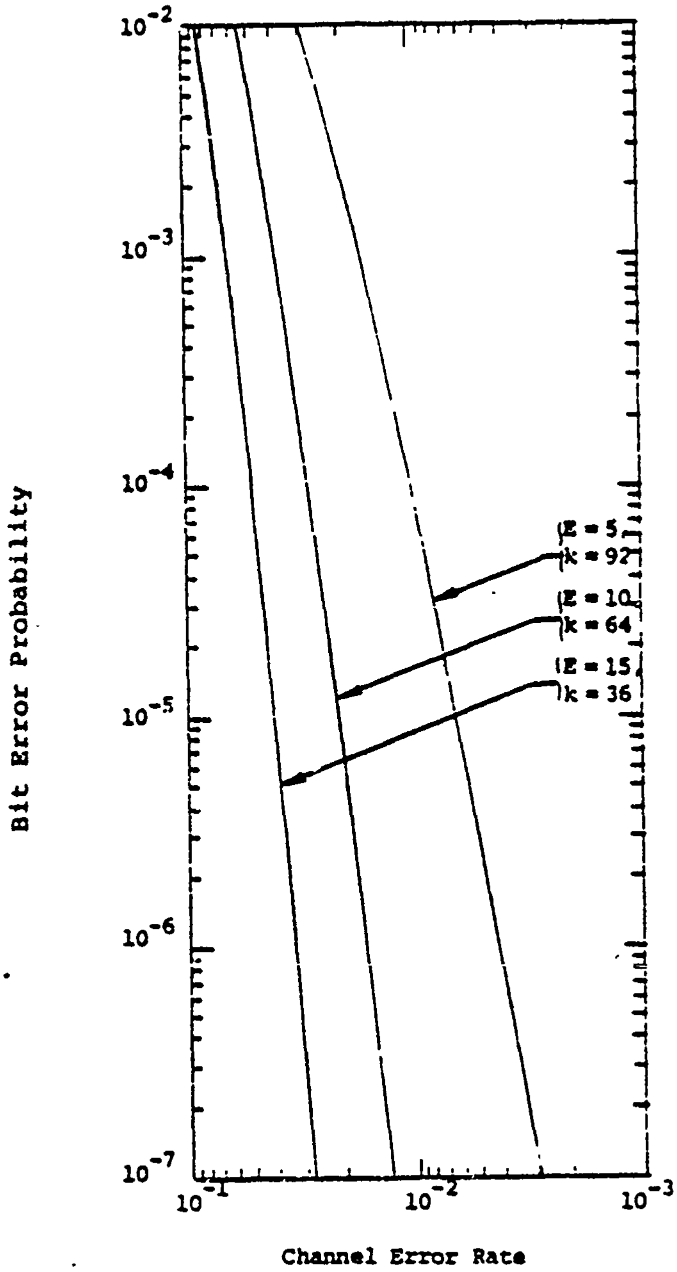


Figure 4.7 Bit error probability versus channel error rate performance of several block length 127, BCH codes.

and

$$k = 2^m - 1 - 2E \qquad (4.22)$$

These codes are particularly good as outer codes in concatenated coding systems (see Section 7.0). In such a system the inner code provides some error control by operating on soft-decision demodulator outputs and then presents hard-decision data to the outer decoder which reduces the error rate to the desired level. Binary inner code symbols are grouped to form the 2^m-ary Reed-Solomon code symbols. These codes are also sometimes used on jamming or fading channels with noncoherent demodulation and 2^m - orthogonal-signal modulation.

The performance of a system with this type of coding on a memoryless channel can be specified in terms of the channel symbol error probability p_s. If the channel is not memoryless, it is usually best to provide some interleaving to break up any bursts. In general, the performance of a coding system not specificially designed for channels with memory is degraded by channel memory. Even channels specifically designed for a channel with memory will be degraded if the memory is different than expected. Usually since the characteristics of channels with memory are difficult to measure, interleaving is a wise approach. Only a rough idea of the channel memory length and any periodic properties of the channel are required to build the interleaver.

Also a system with interleaving is very effective with random errors.

A code which achieves the minimum distance of (4.20) is called a maximum distance spearable code [15] and the weight enumerator polynomial coefficients for these codes have been determined [16]. The result is

$$A_k = \binom{2^m-1}{k}\left(2^m - 1\right)\sum_{i=0}^{k-2E-1}\left(-1\right)^i\binom{k-1}{i}2^{m(k-i-2E-1)}$$

$$\text{for } k \geq 2E+1 \qquad (4.23)$$

Of course, $A_0 = 1$ and $A_k = 0$ for $1 \leq k \leq 2E$.

From (4.2), (4.3) and (4.5) the block, symbol, and undetected error probabilities are

$$P_{block} = \sum_{i=E+1}^{2^m-1}\binom{2^m-1}{i} p_s^i \left(1-p_s\right)^{2^m-1-i} \qquad (4.24)$$

$$P_{sym} = \frac{1}{2^m-1}\sum_{i=E+1}^{2^m-1} i\binom{2^m-1}{i} p_s^i \left(1-p_s\right)^{2^m-1-i} \qquad (4.25)$$

and

$$P_u = \sum_{i=E+1}^{2^m-1} A_i \; p_s^i \left(1-p_s\right)^{2^m-1-i} \qquad (4.26)$$

where (4.25) uses the $\beta_i = i$ approximation for $i > E$.

The bit error probability can be upper bounded by the symbol error probability or for specific channels expressions relating the two probabilities can be obtained. For 2^m- orthogonal-signal modulation, the relationship is [1]

$$P_b = \frac{2^{m-1}}{2^m-1} P_{sym} \qquad (4.27)$$

Figure 4.8 shows the bit error probability versus channel symbol error probability obtained using (4.25) and (4.27) for a n = 31 code capable of correcting various numbers of channel errors. Figure 4.9 shows the bit error probability performances of the same codes versus E_b/N_o for a system with 32-ary MFSK modulation and noncoherent demodulation. Results on the performance of concatenated coding systems which use a Reed-Solomon outer code are given in Section 7.0.

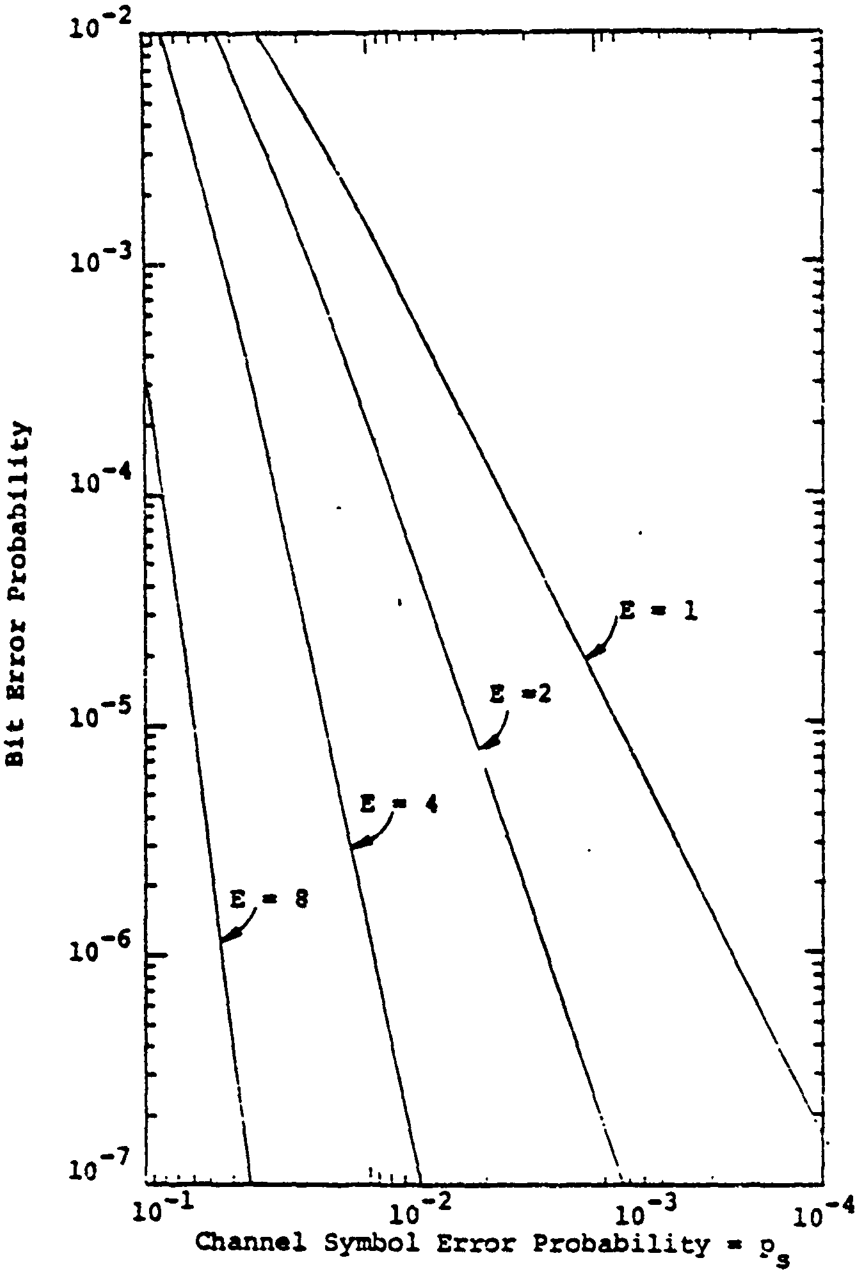


Figure 4.8 Bit error probability versus channel symbol error probability fcr 32-crthogonal-signal modulation and n=31, E-error-correcting Reed-Solomon coding.

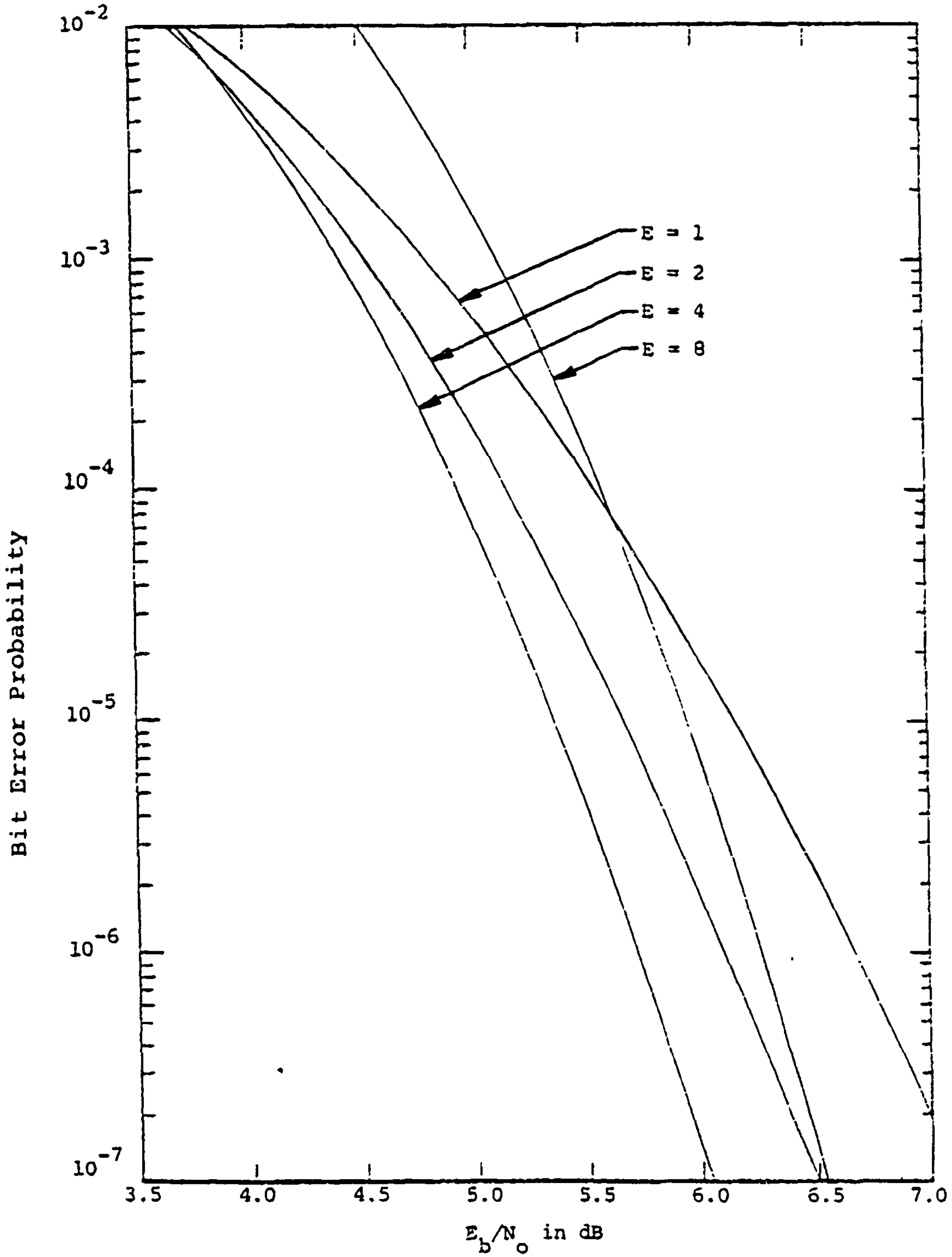


Figure 4.9 Bit error probability versus E_b/N_o performance of several $n = 31$, E-error-correcting Reed-Solcmon coding systems with 32-ary MFSK modulation on an AWGN channel.

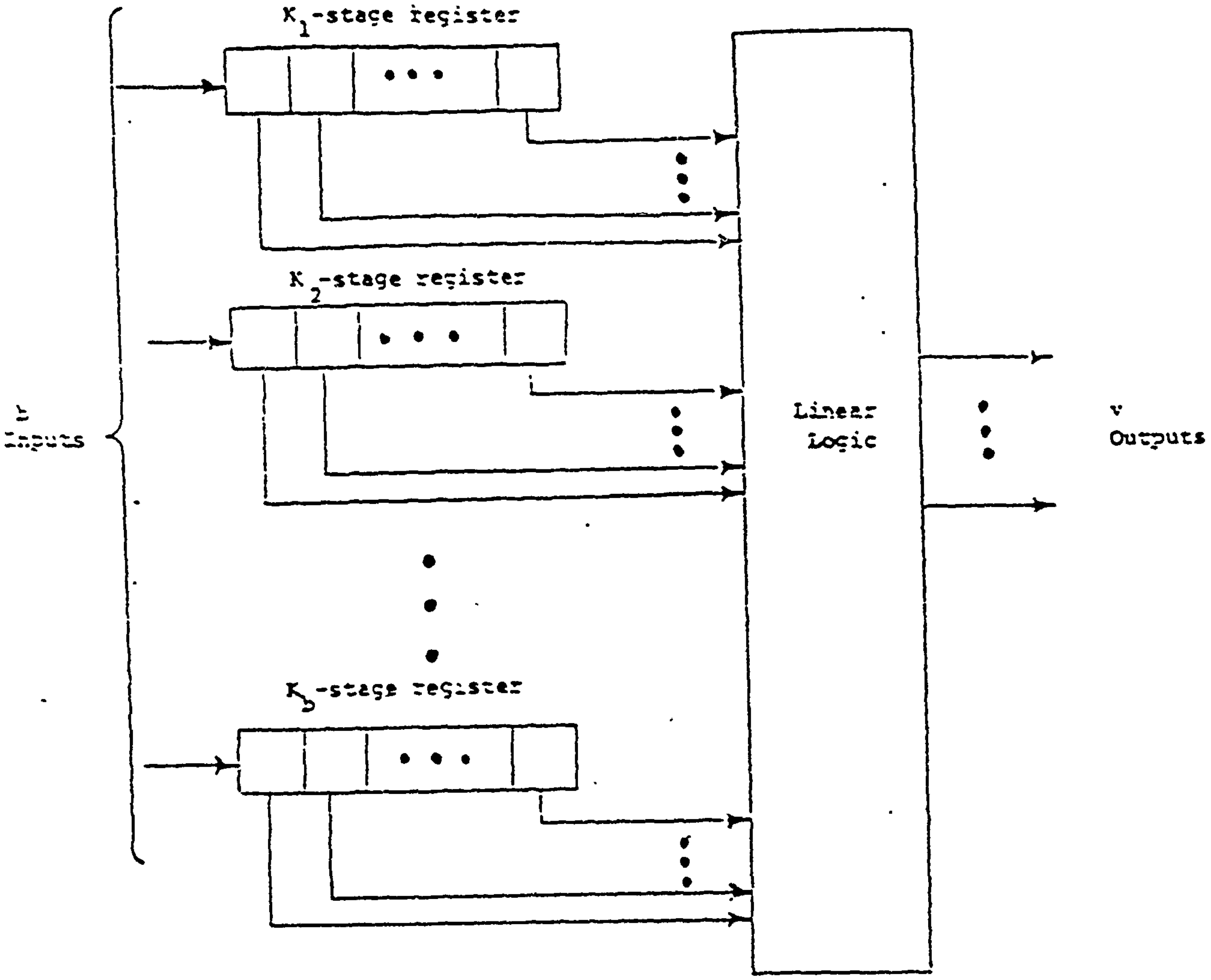
5.0. Binary Convolutional Codes

A rate b/v, constraint length K, binary convolutional encoder is a b-input v-output linear finite-state device which can be implemented with K binary register stages and linear logic as shown in Figure 5.1. Each set of v outputs depends on K variables of which b are the current inputs and K-b are state variables. So there are 2^{K-b} different states. The constraint length K is defined as the total number of binary register stages in the encoder. Sometimes the constraint length is also defined as the number of state variable ν where

$$\nu = K-b \qquad (5.1)$$

Here the first constraint length K definition will be used.

To make some of the convolutional coding concepts easier to understand we will describe some of their properties for the rate R=1/2 constraint length K=3 encoder of Figure 5.2 and then extend the results to the more general encoder of Figure 5.1. Figure 5.2 indicates the outputs for a particular binary input sequence assuming the state (i.e., the previous two data bits into the shift register) were zero. Modulo-2 addition (i.e., $0 \oplus 0 = 0$, $0 \oplus 1 = 1$, $1 \oplus 0 = 1$, $1 \oplus 1 = 0$) is used. With the input and output sequences defined from right-to-left the first three input bits 0, 1, and 1, generate the code outputs 00, 11, and 01, respectively.



Constraint Length = $K = K_1 + K_2 + \ldots + K_b$

Figure 5.1 Rate b/v, constraint length K convolutional encoder.

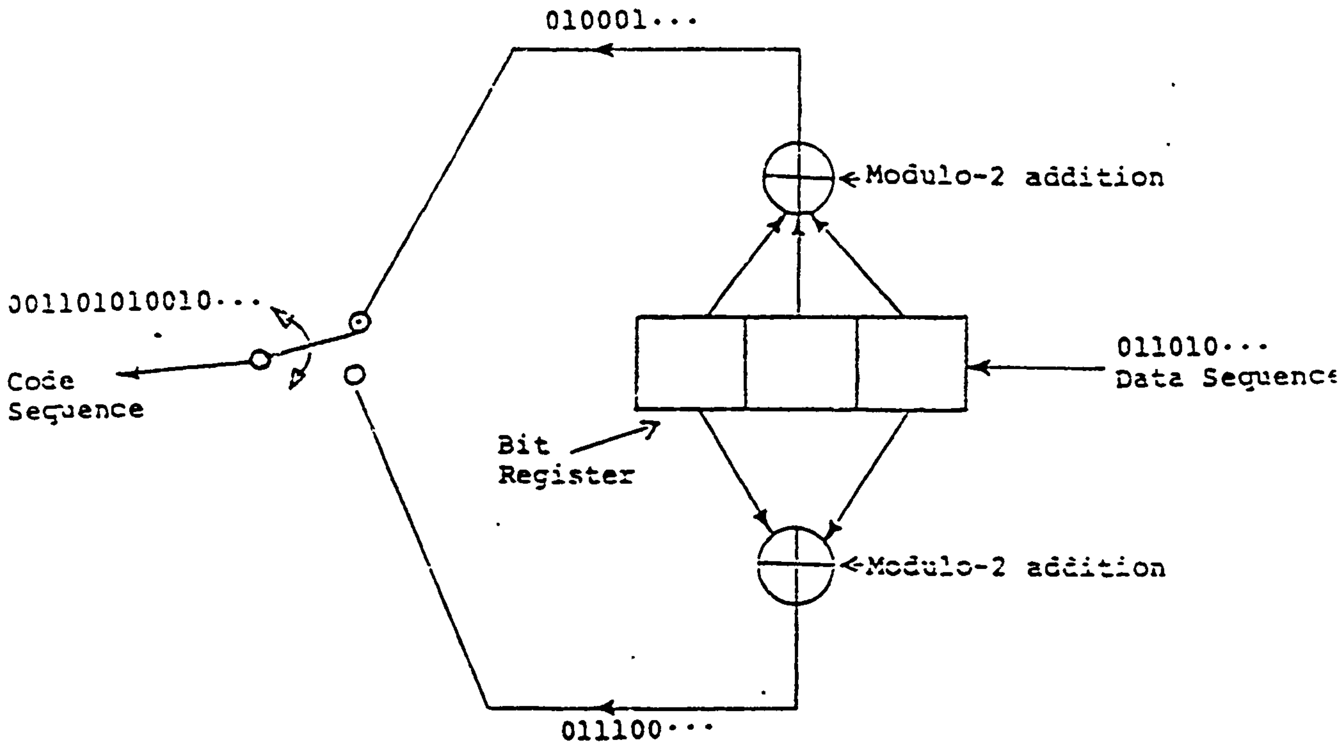


Figure 5.2 Rate 1/2 constraint length 3 convolutional encoder.

The outputs are shown demultiplexed into a single code sequence. Of course, the code sequence has twice the bit rate as the data sequence. We shall pursue this example to develop various representations of convolutional codes and their properties. The techniques thus developed will then be shown to generalize directly to any convolutional code.

It is traditional and instructive to exhibit a convolutional code by means of a tree diagram as shown in Figure 5.3.

If the first input bit is a zero,the code symbols are those shown on the first upper branch, while if it is a one, the output code symbols are those shown on the first lower branch. Similarly, if the second input bit is a zero, we trace the tree diagram to the next upper branch, while if it is a one, we trace the diagram downward. In this manner all thirty-two possible outputs for the first five inputs may be traced.

From the diagram it also becomes clear that after the first three branches the structure becomes repetitive. In fact, we readily recognize that beyond the third branch the code symbols on branches emanating from the two nodes labelled "a" are identical, and similarly for all the similarly labeled pairs of nodes The reason for this is obvious from examination of the encoder. As the fourth input bit enters the coder at the right, the first data bit

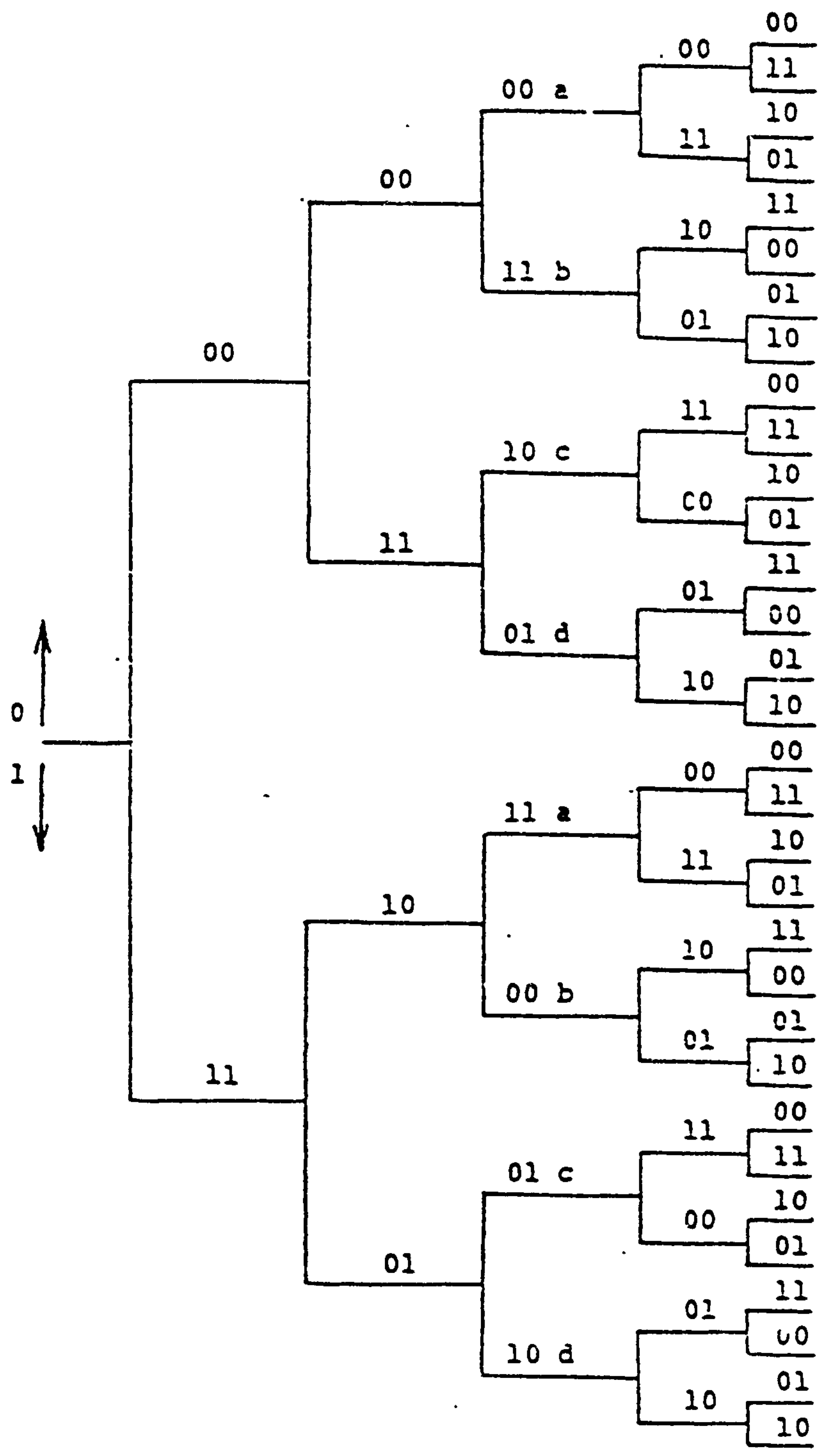


Figure 5.3 Tree code representation for coder of Figure 5.2.

falls off on the left end and no longer influences the output code symbols. Consequently, the data sequences 100xy... and 000xy... generate the same code symbols after the third branch and, as is shown in the tree diagram, both nodes labeled "a" can be joined together.

This leads to redrawing the tree diagram as shown in Figure 5.4. This has been called a trellis diagram since a trellis is a tree-like structure with remerging branches. We adopt the convention here that code branches produced by a "zero" input bit are shown as solid lines and code bra: nes produced by a "one" input bit are shown dashed.

The completely repetitive structure of the trellis diagram suggests a further reduction in the representation of the code to the state diagram of Figure 5.5. The "states" of the state diagram are labeled according to the nodes of the trellis diagram. However, since the states correspond merely to the last two input bits to the coder we may use these bits to denote the nodes or states of this diagram.

We observe finally that the state diagram can be drawn directly observing the finite-state machine properties of the encoder and particularly the fact that a four-state directed graph can be used to represent uniquely the input-output relation of the finite-state machine. For the nodes

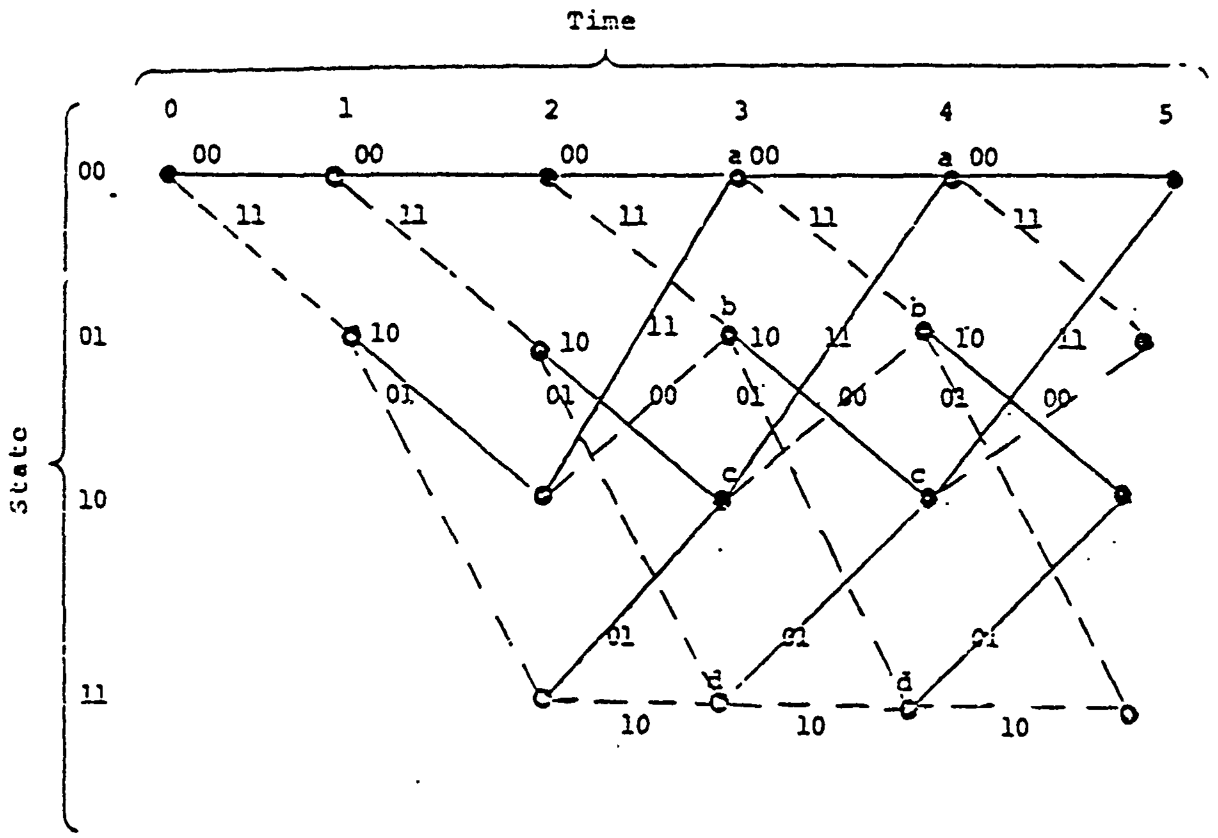


Figure 5.4 Trellis code representation for coder of Figure 5.2

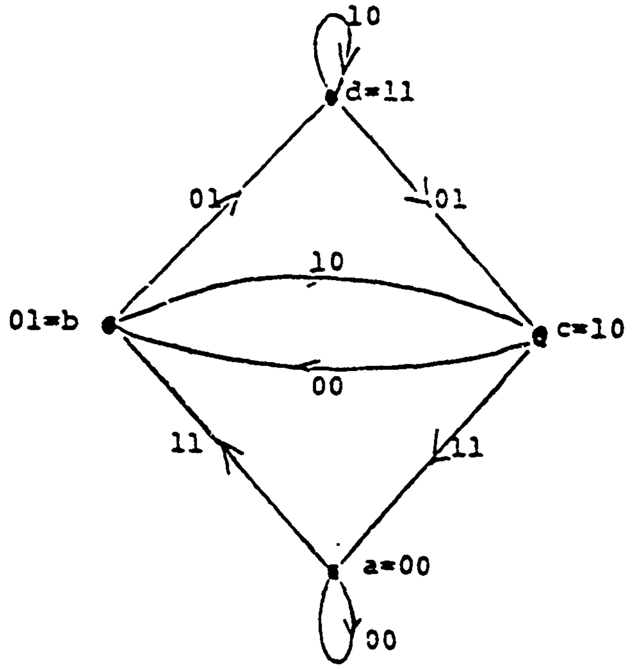


Figure 5.5 State-diagram representation for coder of Figure 5.2

represent the previous two bits while the present bit is indicated by the transition branch; for example, if the encoder (machine) contains 011, this is represented in the diagram by the transition from state b=01 to state d=11 and the corresponding branch indicates the code symbol outputs 01.

In the following sections we will use these representations to describe the three main types of decoders for convolutional codes: Viterbi, sequential and feedback.

5.1 Viterbi Decoded Convolutional Codes

The Viterbi decoding algorithm [20] is a path maximum-likelihood decoding algorithm which takes advantage of the remerging path structure (see Figure 5.4) of convolutional codes. By path maximum-likelihood decoding algorithm we mean that of all the possible paths through the trellis, a Viterbi decoder chooses the path, or one of the paths, most likely in the probabilistic sense to have been transmitted. To simplify the Viterbi decoder description we will describe it first for a hard quantized channel and then generalize the description to a soft-quantized channel.

5.1.1 The Viterbi Decoding Algorithm for the Binary Symmetric Channel

On a binary symmetric channel, errors which trans-form a channel code symbol 0 to 1 or 1 to 0 are assumed to occur independently from symbol to symbol with probability p. If all input (message) sequences are equally likely, the decoder which minimizes the overall path error probability for any code, block or convolutional, is one which examines the error-corrupted received sequence $y_1\ y_2 \ldots y_j \ldots$ and chooses the data sequence corresponding to the transmitted code sequence $x_1\ x_2 \ldots x_j \ldots$ which is closest to the received sequence in the sense of Hamming distance; that is the transmitted sequence which differs from the received sequence in the minimum number of symbols.

Referring first to the tree diagram, this implies that we should choose that path in the tree whose code sequence differs in the minimum number of symbols from the received sequence. However, recognizing that the transmitted code branches remerge continually, we may equally limit our choice to the possible paths in the trellis diagram of Figure 5.4. Examination of this diagram indicates that it is unnecessary to consider the entire received sequence (which conceivably could be thousands or millions of symbols in length) at one time in deciding upon the most likely (minimum distance) transmitted sequence. In particular, immediately after the third branch we may determine which of the two paths leading to node or state "a" is more likely to have been sent. For example, if 010001 is received, it is clear that this is at distance 2 from 000000 while it is at distance 3 from 11101i and consequently we may exclude the lower path into node "a". For, no matter what the subsequent received symbols will be, they will affect the distances only over subsequent branches after these two paths have remerged and consequently in exactly the same way. The same can be said for pairs of paths merging at the other three nodes after the third branch. We shall refer to the minimum distance path of the two paths merging at a given node as the "survivor". Thus it is necessary only to remember which was the minimum distance path from the received sequence (or survivor) at each node, as well as the value of that minimum distance. This is necessary because at the next node level we must compare the two branches merging at each node level, which were survivors at the previous level for different nodes; e.g., the comparison at node "a" after the fourth branch is among the survivors of comparison at nodes "a" and "c" after the third branch. For example, if the received sequence over the first four branches is 01000111, the survivor at the third node level for node "a" is 000000 with distance 2 and at node "c" it is 110101, also with distance 2. In going from the third node level to the fourth the received sequence agrees precisely with the survivor from "c" but has distance 2 from the survivor from "a". Hence the survivor at node "a" of the fourth level is the data sequence 1100 which produced the code sequence 11010111 which is at (minimum) distance 2 from the received sequence.

In this way, we may proceed through the received sequence and at each step preserve one surviving path and its distance from the received sequence, which is more generally called metric. The only difficulty which may arise is the possibility that in a given comparison between merging paths, the distances or metrics are identical. Then we may simply flip a coin as is done for block code words at equal distances from the received sequence. For even if we preserved both of the equally valid contenders, further received symbols would affect both metrics in exactly the same way and thus not further influence our choice.

This decoding algorithm was first proposed by Viterbi [20] in the more general context of arbitrary memoryless channels. Another description of the algorithm can be obtained from the state diagram representation of Figure 5.5. Suppose we sought that path around the directed state diagram, arriving at node "a" after the kth transition, whose code symbols are at a minimum distance from the received sequence. But clearly this minimum distance path to node "a" at time k can be only one of two candidates: the minimum distance path to node "a" at time k-1 and the minimum distance path to node "c" at time k-1. The comparison is performed by adding the new distance accumulated in the kth transition by each of these paths to their minimum distances (metrics) at time k-1.

It appears thus th-t the state-diagram also represents a system diagram for this decoder. With each node or state, we associate a storage register which remembers the minimum distance path into the state after each transition as well as a metric register which remembers its (minimum) distance from the received sequence. Furthermore, comparisons are made at each step between the two paths which lead into each node. Thus four comparators must also be provided.

We defer tho question of truncating the trellis and thereby making a final decision on all bits beyond L branches prior to the given branch until we have some additional properties of convolutional codes.

5.1.2 Distance Properties of Convolutional Codes

We continue to pursue the example of Figure 5.2 for the sake of clarity; in the next section, we shall easily generalize results. As with linear block codes there is no loss in generality in computing the distance from the all zeros code word to all the other code words, for this set of distances is the same as the set of distances from any specific codeword to all the others.

For this purpose, we may again use either the trellis diagram or the state diagram. We first of all redraw the trellis diagram in Figure 5.4 labelling the branches according to their distances from the all zeros path. Now consider all the paths that merge with the all zeros path for the first time at some arbitrary node "j". From the diagram of Figure 5.6 it can be seen that of these paths, there will be just one path at distance 5 from the all zeros path and this diverged from it three branches back. Similarly there are two at distance 6 from it; one which diverged 4 branches back and the other which diverged 5 branches back, and so forth. We note also that the input bits for the distance 5 path are 00...01000 and thus differ in only one input bit from the all zero path. The minimum distance, sometimes called the minimum "free" distance, among all paths is thus seen to be 5. This implies that any pair of channel errors can be corrected, for two errors will cause the received sequence to be at distance 2 from the transmitted (correct) sequence but it will be at least at distance 3 from any other possible code sequence. In this matter the distances of all paths from the all zeros (or any arbitrary) path can be determined from the trellis diagram.

5.1.3 Generalization of Viterbi Decoding to Arbitrary Rate Convolutional Codes

The generalization of these techniques to arbitrary rate 1/v convolutional codes is immediate. That is, an encoder with a K stage shift register and linear logic will produce a trellis or state diagram with 2^{K-1} nodes or states

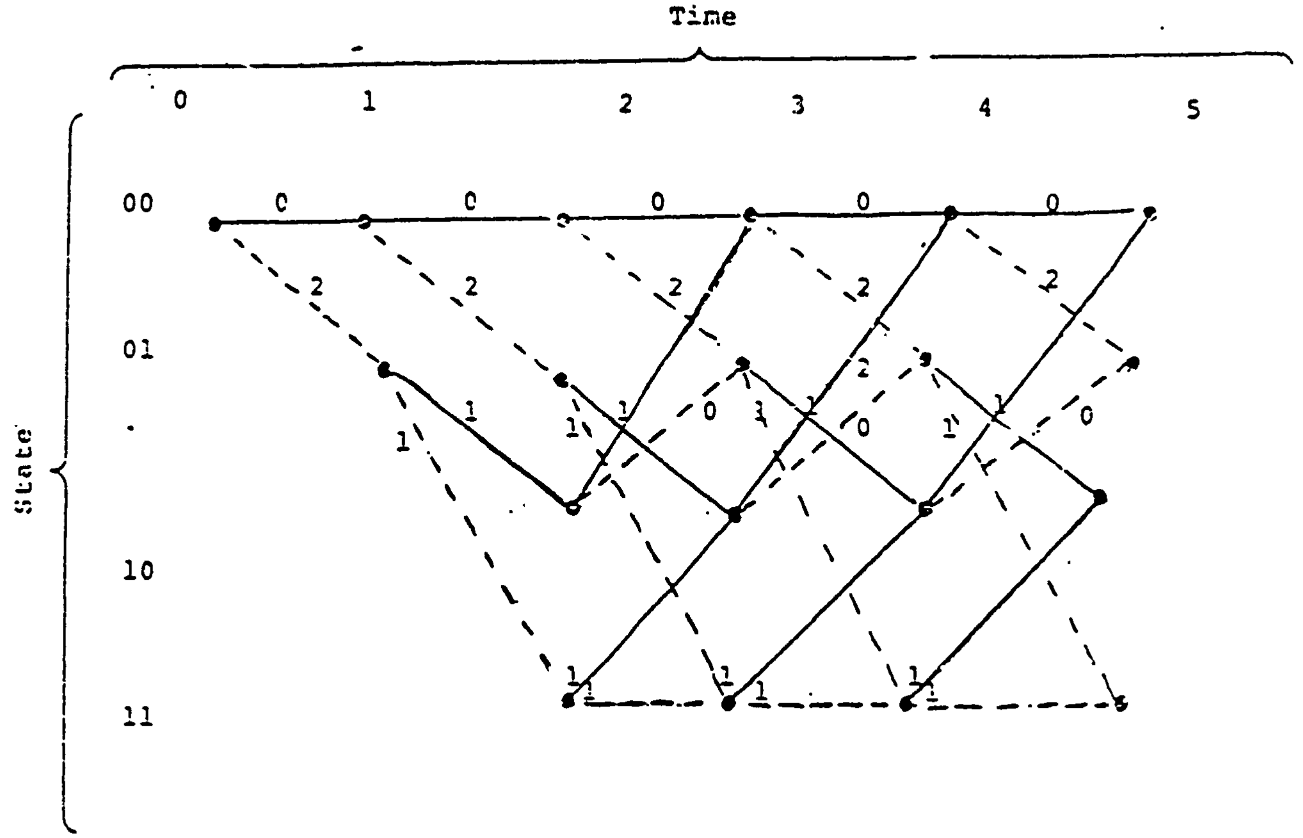


Figure 5.6 Trellis diagram labelled with distances from the all zeros path.

and each branch will contain v code symbols. The rate of this code is then

$$R = \frac{1}{v} \ \frac{\text{bits}}{\text{code symbol}}$$

The example pursued in the previous sections had rate R=1/2. The primary characteristic of rate 1/v codes is that only two branches exit from and enter each node.

If rates other than 1/v are desired, we must make b greater than 1 where b is the number of bits shifted into the encoder at one time. An example for K=4 and rate R=2/3 is shown in Figure 5.7. Its state diagram is shown in Figure 5.8.

It differs from the binary-tree (b=1) representation only in that each node is connected to four other nodes, and for general "b", it will be connected to 2^b nodes. Still all the preceding techniques including the trellis and state diagram analysis are still applicable. It must be noted, however, that the minimum distance decoder must make comparisons among all the paths entering each node at each level of the trellis and select one survivor out of 2^b.

An interesting class of, in general, nonbinary-tree (b≠1) convolutional codes is the unit-memory codes of Lee [21]. The memory of a convolutional code is defined as the number of b-bit input groups from the last input group to the oldest group that contributes to the present v outputs.

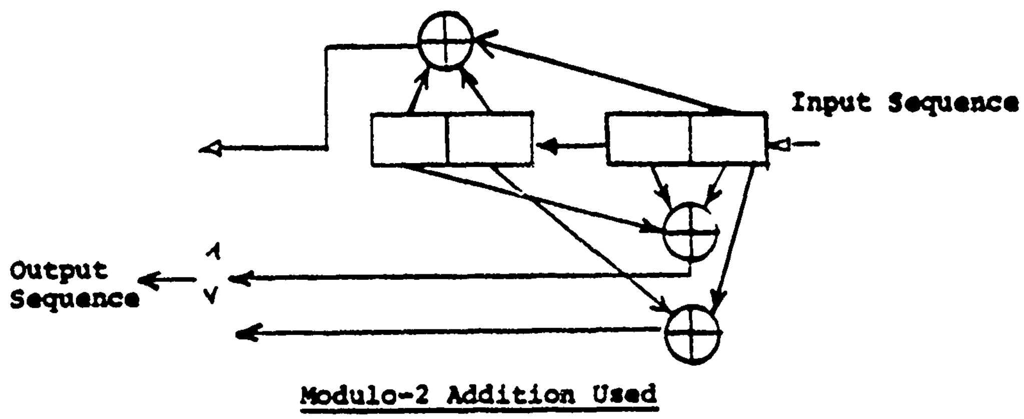


Figure 5.7 K=4, R=2/3 encoder.

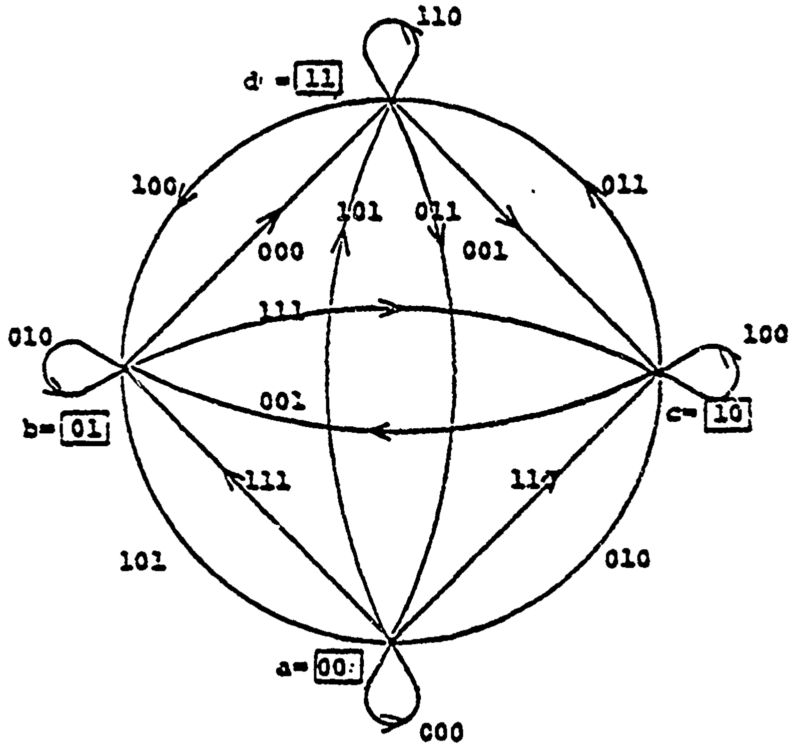


Figure 5.8 State diagram for code of Figure 5.7

With unit-memory codes the v outputs only depend on the present and the previous sets of b inputs. Any memory M, rate b/v convolutional code can be converted to this form by grouping symbols to form a rate (Mb)/(Mv) code. The only problem with this form is that the Viterbi decoder would have to make 2^{Mb}-way rather than 2^{b}-way comparisons. However, for a fixed number of binary states and rate, the additional linear logic possibilities of the unit-memory codes compared to non-unit-memory codes makes it possible to slightly improve the distance properties of such a code.

5.1.4 Systematic and Nonsystematic Convolutional Codes

The term *systematic* convolutional code refers to a code on each of whose branches the uncoded information bits are included in the encoder output bits generated by that branch. Figure 5.9 shows an R=1/2 systematic coder for K=3.

For linear block codes, any nonsystematic code can be transformed into a systematic code with the same block distance properties. This is not the case for convolutional codes. The reason for this is that the per-performance of a code on any channel depends largely upon the relative distance between codewords and particularly on the minimum free distance. Making the code systematic, in general, reduces the maximum possible free distance for a given constraint length and rate. For example, the maximum minimum free distance systematic code for K=3

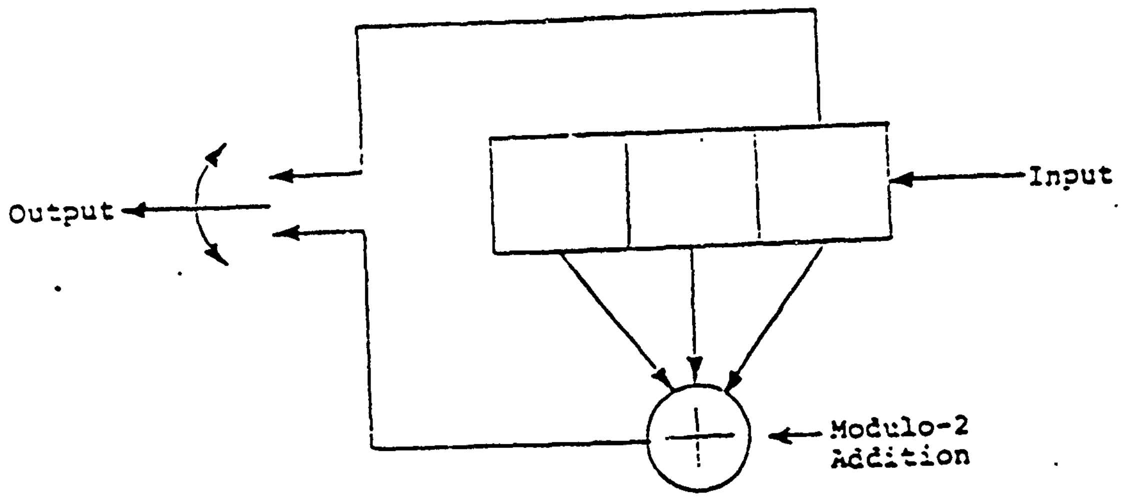


Figure 5.9 Systematic convolutional encoder for K = 3, R = 1/2.

is that of Figure 5.9 and this has d=4, while the ronsystematic K=3 code of Figure 5.2 has minimum free distance d=5. Table 5.1 shows the maximum,minimum free distance for R=1/2 systematic and nonsystematic codes for K=2 through 5.

For large constraint lengths the results are even more widely separated.

5.1.5 Catastrophic Error Propagation

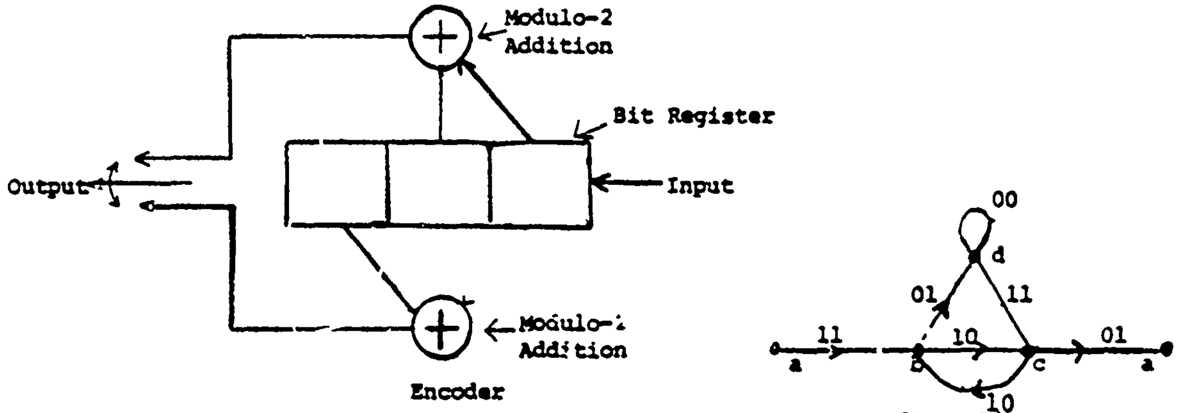
A catastrophic error is defined as the event that a finite number of channel symbol errors causes an infinite number of data bit errors to be decoded. Massey and Sain [22] have derived a necessary and sufficient condition for convolutional codes to display catastrophic error propagation. For rate 1/v codes with the bit register tap multipliers (0 or 1) represented as polynomials in a delay operator D, this condition reduces to the statement that a convolutional code can display catastrophic error propagation if, and only if, the bit register tap multiplier polynomials (sometimes called subgenerator polynomials) have a common factor with modulo-2 arithmetic.

In terms of the state diagram for any rate code, catastrophic errors can occur if, and only if, any closed loop path in the diagram has a zero weight. To illustrate this, consider the example of Figure 5.10. Assuming that the all zeros is the correct path, the incorrect path

Maximum, Minimum Free Distance

K	Systematic	Nonsystematic
2	3	3
3	4	5
4	4	6
5	5	7

Table 5.1. Comparison of systematic and nonsystematic R=1/2 code distances.



Encoder

State Diagram

Figure 5.10 Coder dirplaying catastrophic error propagation.

a b d d ... d c a has exactly 6 ones, no matter how many times we go around the self loop d. Thus, for a binary symmetric channel, for example, four channel errors may cause us to choose this incorrect path or consequently make an arbitrarily large number of bit errors (equal to two plus the number of times the self loop is traversed).

The necessary and sufficient condition of Massey and Sain can also be used to show that the code of Figure 5.10 displays catastrophic error propagation. The subgenerator polynomials for this code are D+1 and D^2+1. Since D^2+1 = (D+1) (D+1) with modulo-2 arithmetic, both subgenerator polynomials have a common factor of D+1. Therefore the code displays catastrophic error propagation.

We observe also that for binary-tree (R=1/v) codes, if each adder of the coder has an even number of connections, then the self loop corresponding to the all ones (data) state will have zero weight and consequently the code will be catastrophic.

The only advantage of a systematic code is that it can never be catastrophic, since each closed loop must contain at least one branch generated by a nonzero data bit and thus have a nonzero code symbol. Still, it can be shown that only a small fraction of nonsystematic codes are catastrophic [23]. We note further that if catastrophic errors are ignored, nonsystematic codes with even larger free distance than those of Table 5.1 exist.

5.1.6 Generalization of Viterbi Decoding to Soft Quantized Channels.

To describe how the Viterbi decoding algorithm operates with soft quantization consider the biphase (0° or 180°) modulated additive white Gaussian noise channel. Then in addition to the sign of the demodulator output an indication of its magnitude is provided. The first step is to assign metric values to each of the possible output intervals under the hypothesis that the 0° phase was used and that the 180° phase was used. A common choice is to use integer metrics which for a positive (0°) hypothesis assigns a "0" symbol metric value to the most negative demodulator output interval and unity increasing metric values to the progressively more positive demodulator intervals. For 3-bit quantization the metrics would go from 0 to 7. With the negative (180°) hypothesis metrics decreasing from 7 to 0 are used.

The metrics for any branch or path are computed by summing the corresponding symbol metrics with the set of metrics to use (positive or negative hypothesis) determined by the polarity of the test channel symbol. Then the metrics of remerging paths are compared and the path with the smallest metric is eliminated.

Note that the path metrics as described he-e would continually increase. However, since the metrics are used in comparison, only their relative differences are required. So some amount can be occasionally subtracted from all of the path metrics to keep them within a certain range.

Computer simulations and measurements of hardware systems for a wide variety of codes and channels have shown that the differences in the E_b/N_o ratios required to achieve a given error rate for various numbers of quantization intervals with the integer metrics described here (and in fact most reasonable metric choices) are almost exactly as estimated in Section 3.2 based on operation at $R=R_o$.

5.1.7 Path Memory Truncation

Another problem which arises in the implementation of a Viterbi decoder is the length of the path history which must be stored. In our previous discussion we ignored this important point and therefore implicitly assumed that all past data would be stored. A final decision can be made by forcing the coder into a known (all zeros) state, but this is totally impractical for long data sequences, for it requires storage of the entire trellis memory for each state. Suppose we truncate the path memories after L bits (branches) have been accumulated, by comparing all 2^K metrics for a maximum and deciding on the bit corresponding to that path (out of 2^K) with the highest metric L branches forward. If L is several times as large as K, the additional bit errors introduced in this way are very few. It can be shown that the additional error probability due to path truncation, based on the largest path metric L branches beyond where the decision is to be made, is of the order of a block coding error for a code of block length L bits. Both theory and simulation have shown that by making L four to five times as large as the code constraint length K, we can ensure that such additional errors have only a slight affect on the overall bit error probability.

Of course, basing the decision upon the maximum metric L branches forward may require a costly implementation to compare all 2^K state metrics. Other decision techniques, based on majority polling and metric overflow monitoring, are much less costly and yield the same or better performance when L is increased slightly.

5.1.8 Code Selection

The linear logic for a convolutional code is usually selected based on the code distance properties as discussed in Section 5.1.2. The first criterion is to select a code (linear logic) that does not have catastrophic error propagation (see Section 5.1.5) and that has the maximum possible free distance for the given rate and constraint length. Then the number of paths or dversaries at the free distance or if the bit error probability is the performance measure, the total number of information bit errors represented by the adversaries at the free distance should be minimized. This selection procedure can be further refined by considering the number of adversaries or information bit errors at the free distance plus 1, plus 2, etc. until only one code or class of codes remains. A listing of R=1/2 K=3 to 9 and R=1/3 K=3, to 8 codes selected based on this criterion is given in Table 5.2, (Reference 24, but note K=7, R= 1/3 correction).. The R=1/v constraint length K codes in this table are specified in terms of v K-digit sequences. The i th digit (0 or 1) in the j th sequence specifies the tap multiplier in determining the contribution to the j th branch output due to the symbol in the i th encoder register stage. The total j th output is the modulo-2 sum of the v individual contributions.

Rate	Constraint Length	Code
1/2	3	111 101
1/2	4	1111 1101
1/2	5	11101 10011
1/2	6	111101 101011
1/2	7	1111001 1011011
1/2	8	11111001 10100111
1/2	9	111101011 101110001
1/3	3	111 111 101
1/3	4	1111 1101 1011
1/3	5	11111 11011 10101
1/3	6	111101 101011 100111
1/3	7	1111001 1110101 1011011
1/3	8	11110111 11011001 10010101

Table 5.2 Optimum short constraint length R=1/2 and 1/3 convolutional codes.

Other codes which achieve the maximum free distance but do not necessarily have the minimum number of bit errors as described above, are given in Reference 25 for R=1/2, 1/3 and 1/4 codes and in Reference 26 for R=2/3 and 3/4 codes.

5.1.9 Computer Simulation Performance Results

One of the main methods of determining the performance of convolutional coding systems is by computer simulation. Such simulations are especially helpful in determining the error rate performance at higher error rates where analytical bounding techniques are not very tight and where the error probabilities can be estimated with a reasonable amount of computer time.

Sometimes the all zero information input sequence is assumed, but when only a few quantization intervals are used it is best to use random data to avoid biasing the results due to the method of resolving metric comparison ties. The advantage of using the all zero sequence is that no encoder is necessary since the encoded sequence will still be all zeros and determining the error rate reduces to determining the fraction of nonzero decoder outputs.

The quantized received data sequence is generated by modifying the encoded sequence according to the channel transition probability diagram (see Figure 3.6 and 3.9). Once the quantized received sequence is generated a Viterbi decoder identical to a hardware implementation can be programed. The error rate is determined by comparing the Viterbi decoder output with the delayed information sequence.

Figures 5.11, 5.12, and 5.13 show the bit error probability performance of K=7 R= 1/2, K=7 R= 1/3, and K=9 R= 3/4 convolutional coding systems on an additive white Gaussian noise channel with hard and 3-bit soft quantization. These simulation results have also been verified by measurements on hardware systems. The upper bounds shown in these figures are discussed in the next two sections. Figures 5.11 through 5.13 again illustrate the advantages of soft quantization discussed in Section 3.2. The K=7 R= 1/2 code used for Figure 5.11 is the optimum code given in Table 5.2 and the K=7 R= 1/3 code used for Figure 5.12, while not optimum in the distance sense used in Table 5.2, achieves a bit error probability virtually equivalent to (in fact slightly better than) the code of [25] in the range of error probabilities shown.

The R= 3/4 code used for Figure 5.13 is not the best possible code. At a bit error probability of 10^{-5} other codes superior (i.e., with smaller bit error probability) to this code and the code of [26] can achieve about .4 dB E_b/N_o improvement over that shown in Figure 5.13. The

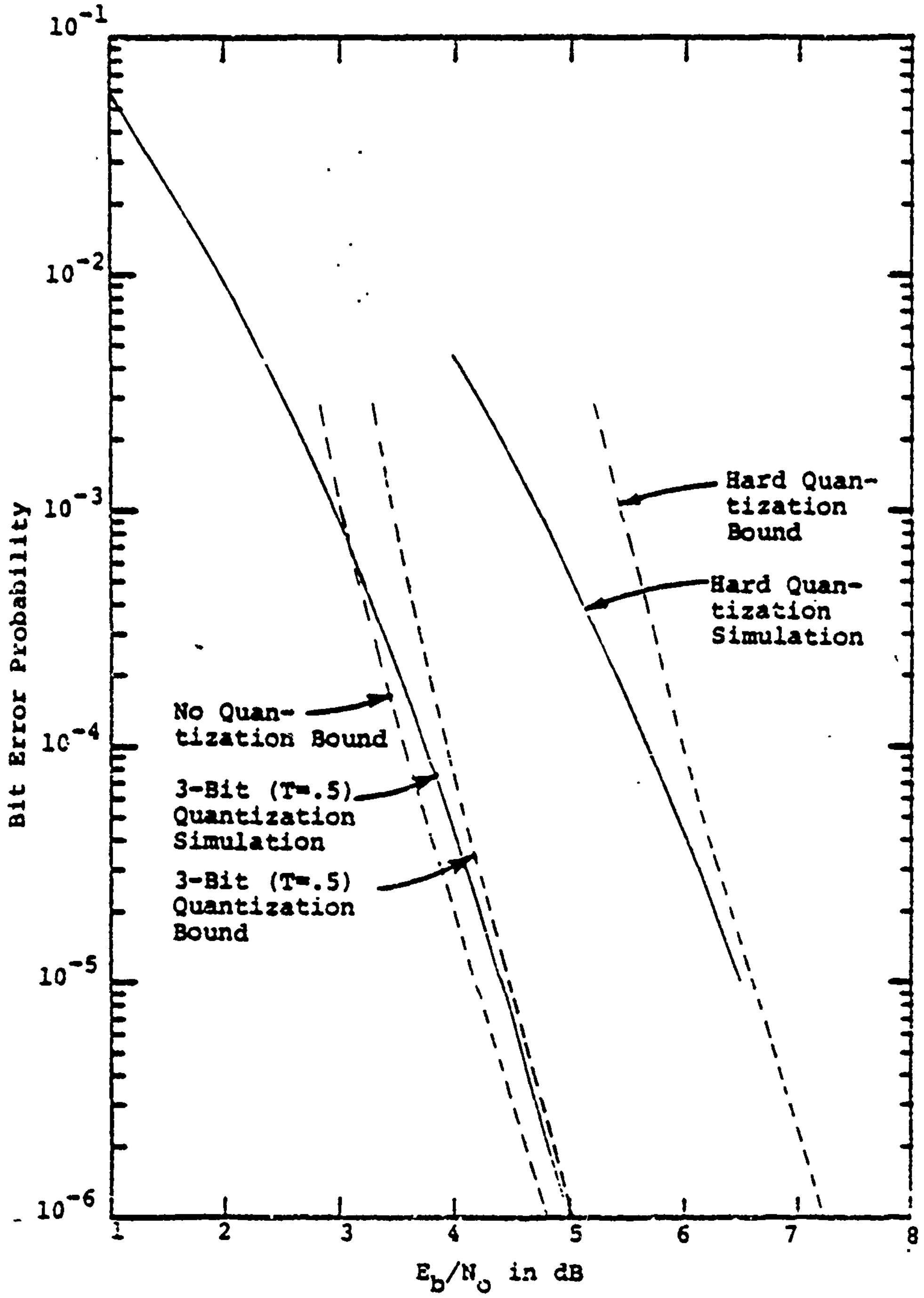


Figure 5.11 Bit error probability versus E_b/N_o performance of a K=7, R= 1/2 convolutional coding system with BPSK modulation and an AWGN channel.

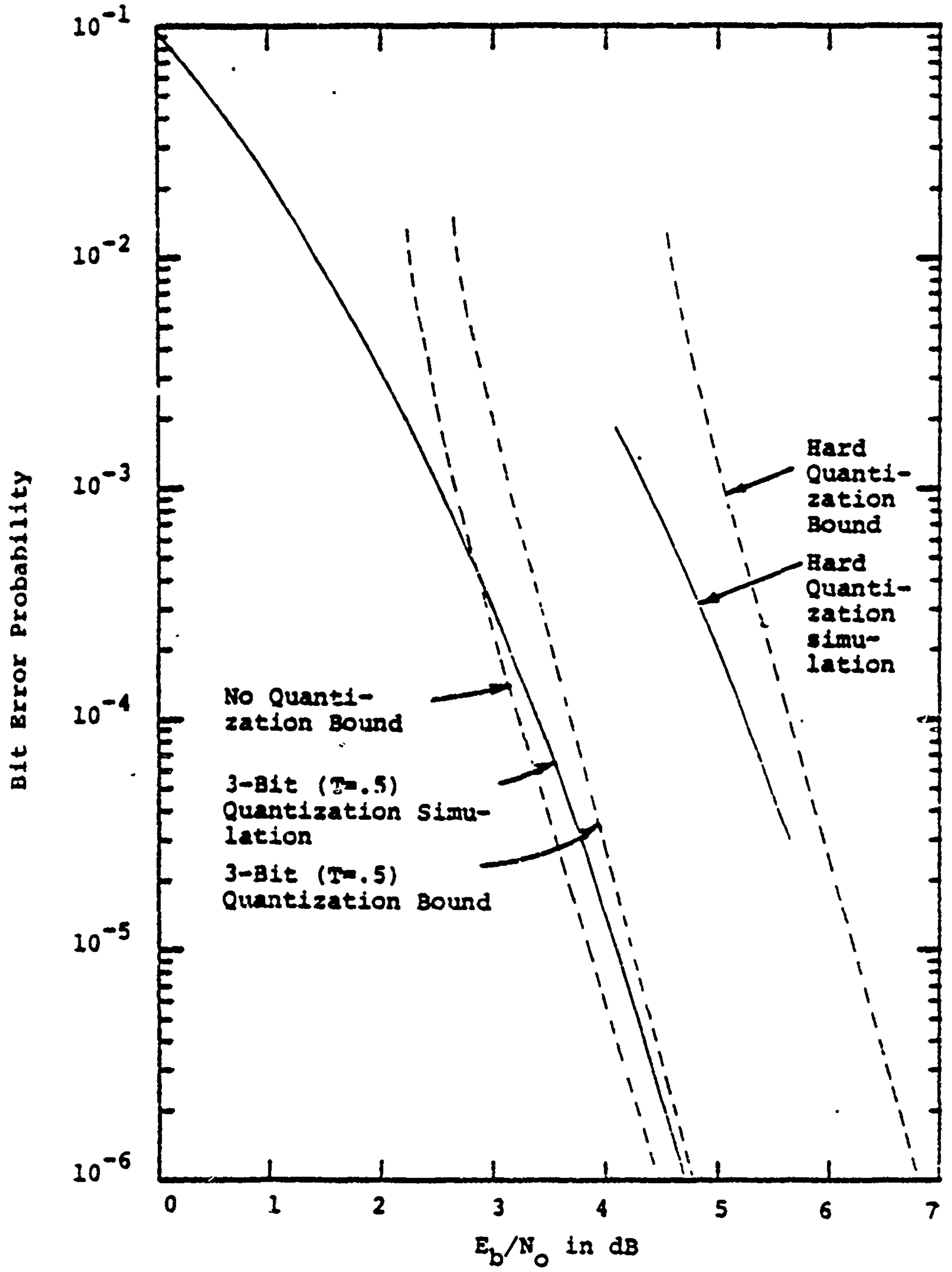


Figure 5.12 Bit error probability versus E_b/N_o performance of a K=7, R= 1/3 convolutional coding system with BPSK modulation and an AWGN channel.

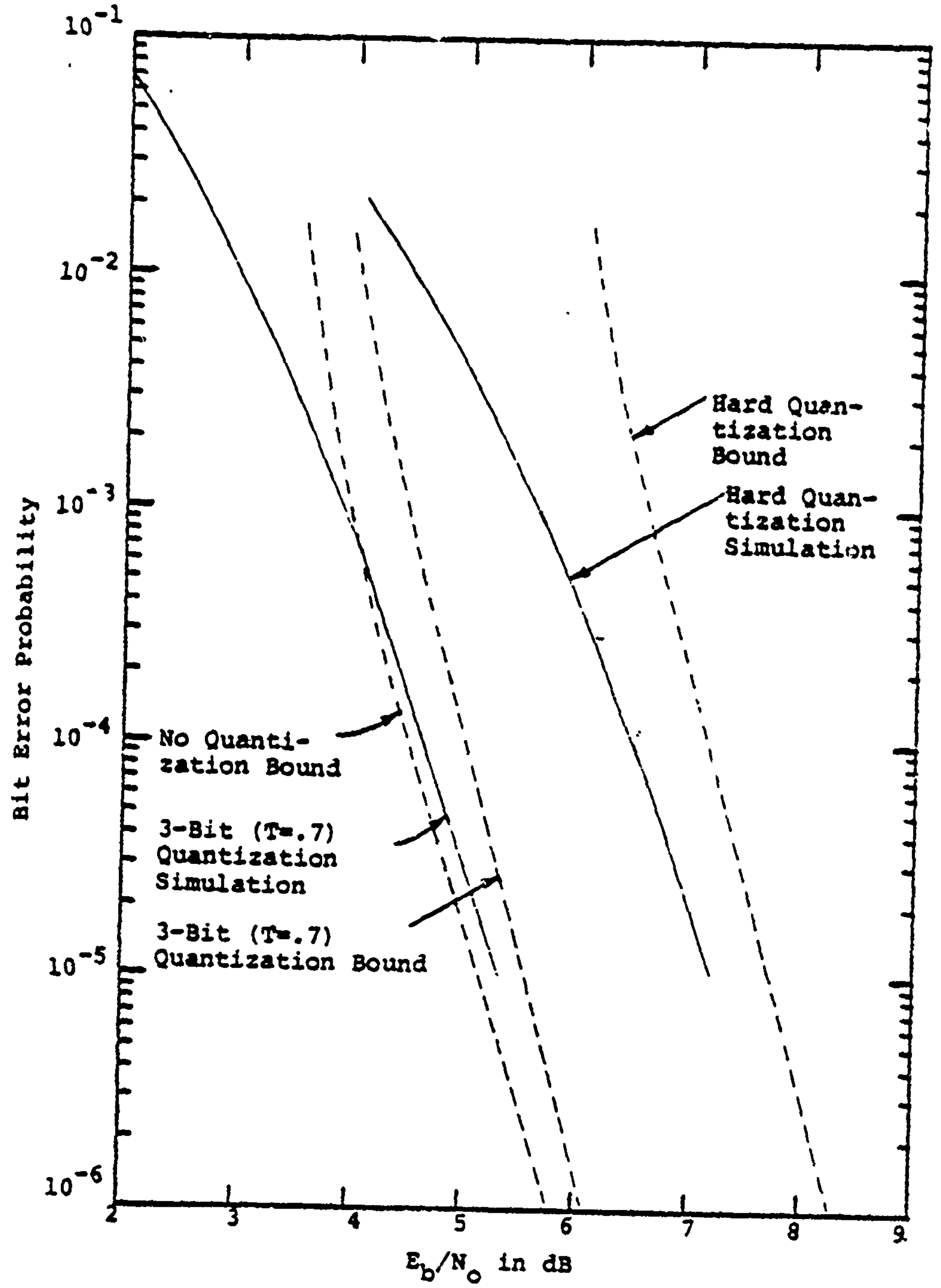


Figure 5.13 Bit error probability versus E_b/N_o performance of a K=9, R= 3/4 convolutional coding system with BPSK modulation and an AWGN channel.

reason for giving the performance of this particular code is that the encoder/decoder for this K=9 R= 3/4 code and the K=7 R= 1/2 code have been implemented as a switch-selectable option in a single unit with only a few more standard integrated circuit chips than are required for the single K=7 R= 1/2 encoder/decoder.

Simulation results have also been obtained many other codes on the additive white Gaussian noise channel [27]. The results show that for rate 1/2 codes each increment increase in the constraint length in the range K=3 to 8 provides an approximate .4 to .5 dB E_b/N_o improvement at a bit error rate of 10^{-5}.

The coding gain is just the difference between the E_b/N_o required for a particular error rate without coding and with coding. Figure 5.14 shows the 3-bit quantization coding gain for the codes of Figure 5.11. 5.12, and 5.13.

The hard quantization curves of Figure 5.11, 5.12, and 5.13 can also be expressed in terms of the channel error rate. The results are given in Figure 5.15. This figure can be used to obtain the performance of these hard quantized coding systems for different memoryless modulation and channel types using the bit error probability curves of Section 3.1. Just treat the curves of Section

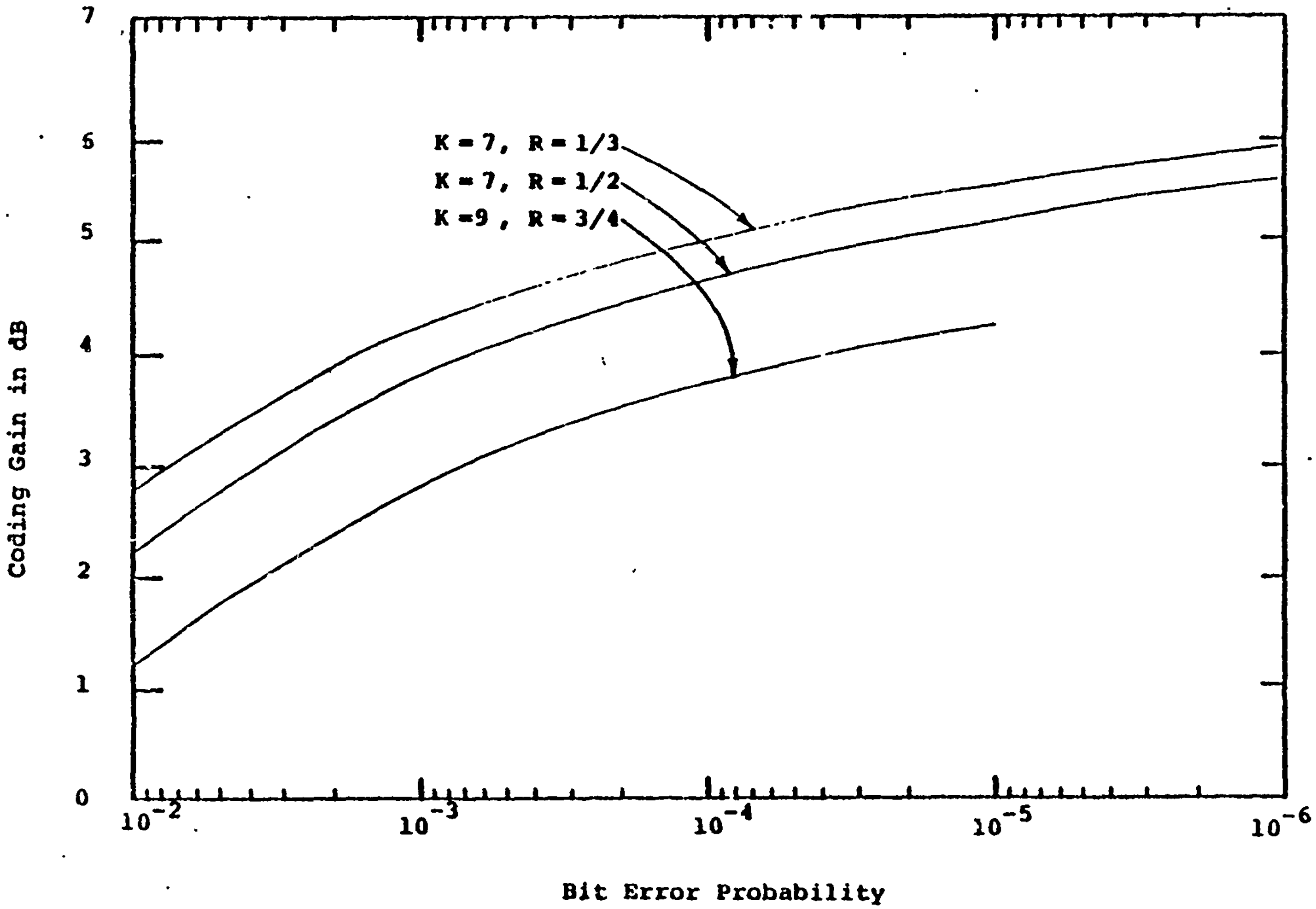


Figure 5.14 Coding gain for several convolutional codes with BPSK modulation, AWGN, and 3-bit receiver quantization.

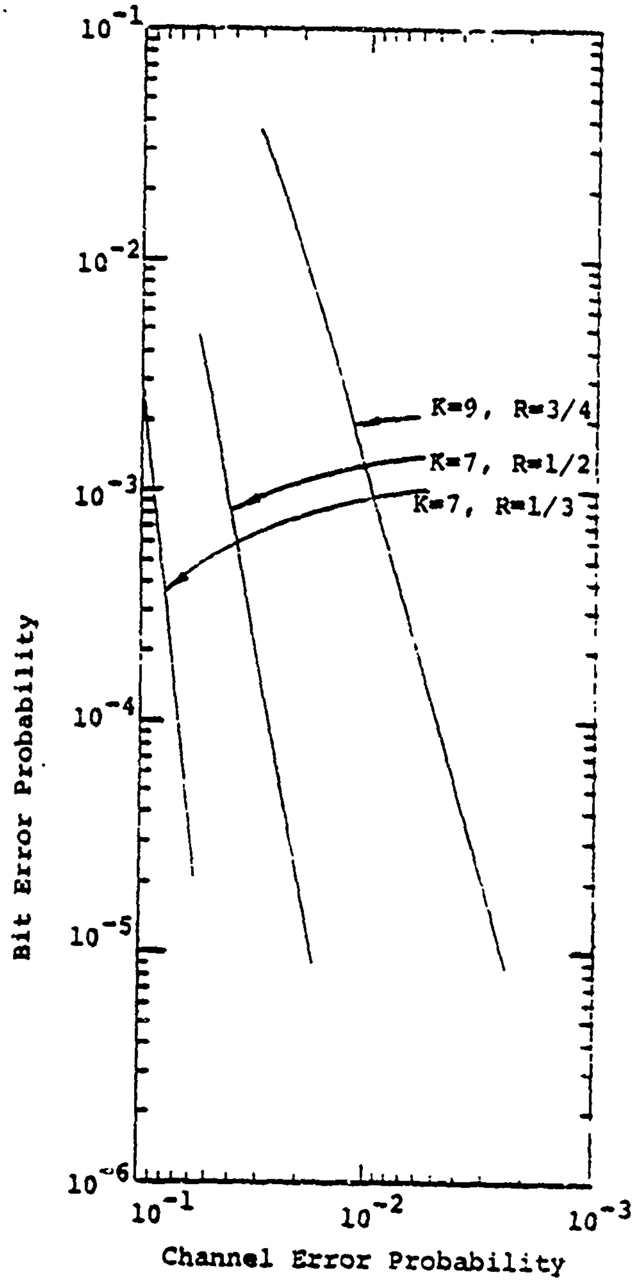


Figure 5.15 Bit error probability versus channel error rate performance of several convolutional coding systems.

3.1 as the channel symbol error probability versus the channel symbol energy-to-noise ratio. The coded system information bit energy-to-noise ratio is then

$$\frac{E_b}{N_o} = \frac{1}{R}\,\frac{E_s}{N_o} \qquad (5.2)$$

As mentioned previously, interleaving can be used to make the channel appear to be memoryless.

For example, on an independent Rayleigh fading channel with binary FSK a K=7, R= 1/2, L=4 diversity system with hard quantization requires $\bar{E}_b/N_o$ = 15.1 dB for $P_b = 10^{-5}$. This is 2.8 dB better than the optimum diversity (L=16) uncoded system (see Figure 3.5) for this channel and 3.8 dB better than an uncoded systen with the same 8 channel bits per information bit.

Figure 5.16 shows the simulated additive white Gaussian noise channel performance of a Viterbi-decoded convolutional coding system ideally suited for bandlimited situations [36]. This system consists of a K=8, R= 2/3 convolutional code with an octal-PSK modem. This system has the same bandwidth requirements for a given data rate as an uncoded QPSK system. Such a system is sometimes referred to as a "unity bandwidth expansion coding system". Figure 5.16 also shows the effects of Viterbi decoder path memory truncation for this system.

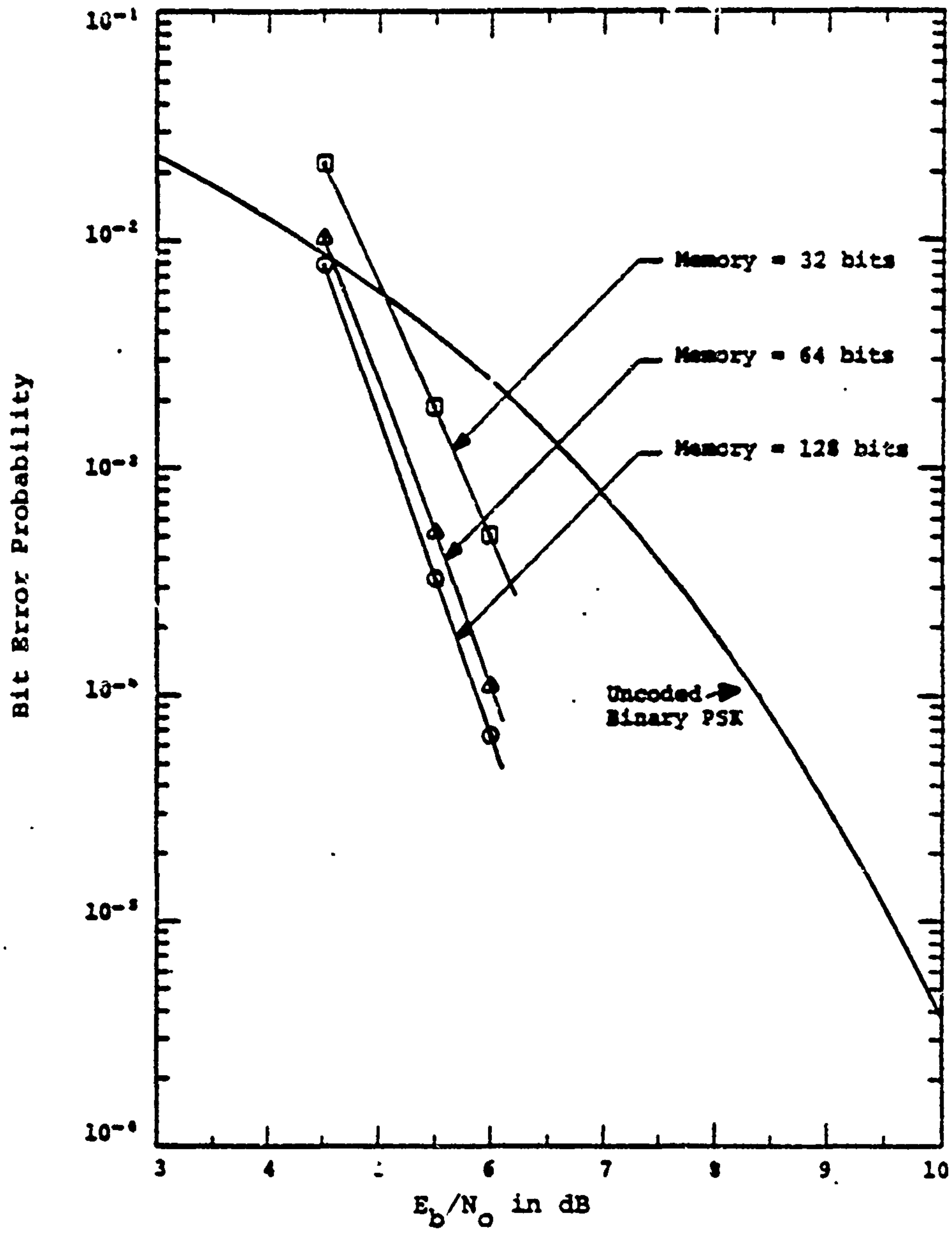


Figure 5.16 Performance of the modulation and coding system with a rate 2/3, constraint length 8, Viterbi-decoded convolutional code and an octal-PSK modem for several path length memories.

Figure 5.17 gives the simulation bit error probability performance of a K=7 R= 1/2 convolutional coding system with DBPSK modulation on an additive white Gaussian noise channel. As expected from the results of Section 3.2.2.3 the performance of this system is considerably inferior to the same coding system and channel with BPSK modulation.

Sometimes the message rather than the bit error probability is the performance measure. A simple upper bound on the message error probability for an M-bit message is just M times the bit error probability. However, since the output errors in an Viterbi-decoded convolutional coding system tend to occur in bursts, this bound is somewhat pessimistic. To characterize the bursts out of a Viterbi-decoder, define an error burst to be the sequence of information symbols from the first error to the last error during which the path choosen by the Viterbi decoder through the trellis is not merged with the correct path. During this burst some of the symbols may be correct, but for R= 1/v codes the correct subsequences in the burst are less than K-1 bits in length because longer sequence would cause the path to remerge with the correct path. Note that since the last K-1 bits of the unmerged span must be correct for the path to remerge with the correct path, the burst

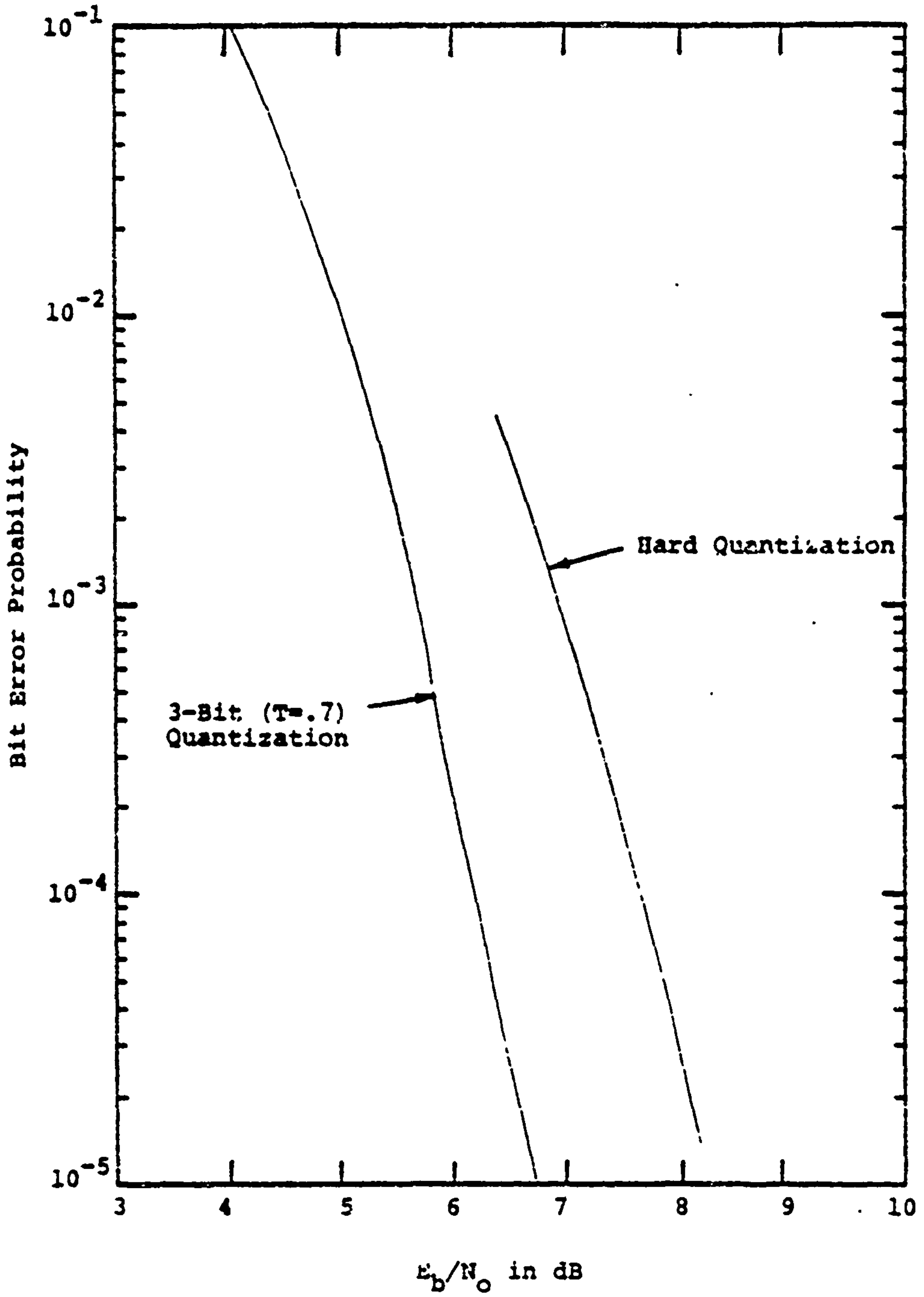


Figure 5.17 Bit error probability versus E_b/N_o performance of a a K=7, R=1/2 convolutional coding system with DBPSK modulation and an AWGN channel.

length is K-1 less than the length of the unmerged span. Tables 5.3 and 5.4 give error burst statistics for the K=7 R= 1/2 system with 3-bit and hard quantization, respectively. The event error probability (i.e., the event of the start of an error burst) is the bit error probability divided by the average number of bit errors per burst. Then a better upper bound on the M-bit message error probability is M times the event error probability

5.1.10 Analytical Performance Techniques with No Quantization

The basic method of analytically determining the performance of noncatastrophic Viterbi-decoded convolutional coding systems is with the generating function approach of Viterbi [28]. With this technique the first step is to determine a generating function T(D,N,L) which describes all the different paths which could be compared with the correct path assuming the all zeros message is used. In the infinite expansion of T(D,N,L) the power of D in the terms represents the number of channel symbols in which the path differs from the correct (all zero) path, the power of N represents the number of information bit errors in the path,

E_b/N_o	Average Error Burst Length in Bits	Average Number of Errors per Burst
1.0	17.3	12.1
2.0	10.9	5.9
3.0	7.6	4.3
4.0	6.2	3.8

Table 5.3 Error burst statistics for K=7 R= 1/2 system with 3-bit quantization.

E_b/N_o	Average Error Burst Length in Bits	Average Number of Errors per Burst
4.0	13.3	7.8
5.0	9.9	5.5
6.0	7.9	4.5

Table 5.4 Error burst statistics for a K=7 R= 1/2 sy. :em with hard quantization.

and the power of L represents the length in branches (b-bit) information segments for R= b/v) of the path. Let

$$T(D,N,L)\Big|_{\substack{N=1\\L=1}} = \sum_{i=d_f}^{\infty} a_i D^i \qquad (5.3)$$

$$\frac{dT(D,N,L)}{dN}\Big|_{\substack{N=1\\L=1}} = \sum_{i=d_f}^{\infty} b_i D^i \qquad (5.4)$$

and

P_i = Probability of an error in comparing two paths that differ in i positions (channel symbols) (5.5)

where d_f is the free distance of the code. The summations in (5.3) and (5.4) could also have been taken from 0 to ∞ since a_i and b_i ara, by the definition of free distance, zero for $i < d_f$. Then for a rate R= b/v code and an M-bit message,the message and bit error probabilities are bounded by [28]

$$P_{message} < \frac{M}{b} \sum_{i=d_f}^{\infty} a_i P_i \qquad (5.6)$$

and

$$P_b < \frac{1}{b} \sum_{i=d_f}^{\infty} b_i P_i \qquad (5.7)$$

To illustrate this technique and to provide some rational for (5.6) and (5.7) consider the K=3, R= 1/2 code of Figure 5.2. The first step is to determine the generating function. To do this refer to the modified state diagram of Figure 5.18. This modified state diagram was obtained from the state diagram of Figure 5.5 with the all zero state split into an initial and final state, the all zero state self loop omitted, and the branches marked with the branch generating functions. The path generating functions of all the paths that can be compared with the correct (zero) path are represented in this diagram by all the possible paths from the initial all zero state to the final all zero state. These paths can be exprssed as the transfer function of the diagram. For this example the result is

$$T\left(D,N,L\right) = \frac{D^5NL^3}{1-DNL-DNL^2} \tag{5.8}$$

$$= D^5NL^3+D^6N^2\left(L^4+L^5\right) +D^7N^3\left(L^5+2L^6+L^7\right) +..$$

Equation 5.8 shows that among the adversaries to the correct (all zero) path there is one path of weight 5, and it is three branches long and results in one bit error. There are two paths of weight 6, one of length 4 and one of length 5, and both result in two bit errors, etc. The message error probability can then be bounded by the probbaility of an error in any set of comparisons times the

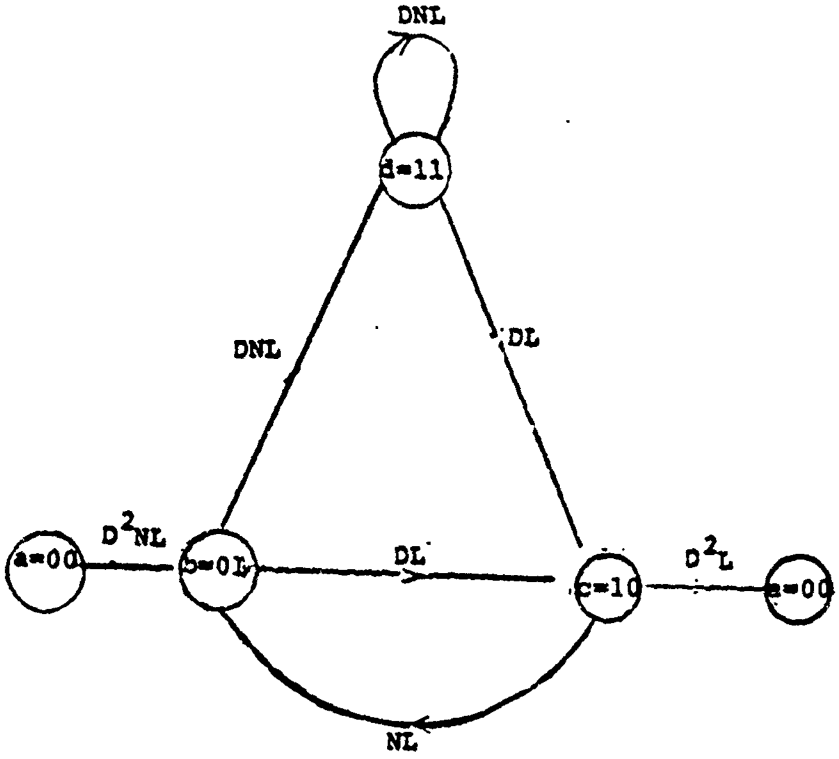


Figure 5.18 Modified state diagram for the K=3, R=1/2 convolutional code of Figure 5.2.

number of comparisons per message. For a R= b/v code comparisons are made every b information bits (i.e., every branch). So for an M-bit message there are M/b comparisons. The union bound of (5.6) follows.

To determine the bit error probability the number of information bit errors in an incorrect path must be accounted for. In the example, an error in comparisons with either one of the distance 6 paths produces 2 bit errors while an error in the comparison with the distance 5 path only produces one error. The number of bit errors per incorrect path can be accounted for by taking the derivative of (5.8) with respect to N. The bound of (5.7) results. Again the 1/b factor accounts for the fact that comparisons are only made every branch (i.e., every b information bits). For the example, the transfer function derivative of (5.4) is

$$\left.\frac{dT\left(D,N,\ L\right)}{dN}\right|_{\substack{N=1\\L=1}} = \frac{D^5}{\left(1-2D\right)^2} \tag{5.9}$$

The P_i probabilities depend on the particular modulation, channel, and quantization. For BPSK modulation, additive white Gaussian noise, and no quantization

$$\begin{aligned} P_i &= Q\left(\sqrt{2iR\,\frac{E_b}{N_o}}\right) \\ &= Q\left(\sqrt{i\,\frac{E_b}{N_o}}\right) \end{aligned} \tag{5.10}$$

for this R= 1/2 example where

$$Q\left(x\right) = \int_{x}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left(- \frac{y^2}{2}\right) dy \qquad (5.11)$$

Using the bound

$$P_i \leq Q\left(\sqrt{2d_f R \frac{E_b}{N_o}}\right) \exp\left\{-\left(i-d_f\right) R \frac{E_b}{N_o}\right\}, \; i \geq d_f \qquad (5.12)$$

the message and bit error probability bounds for this example become

$$P_{message} < \frac{M\, Q\left(\sqrt{5 \frac{E_b}{N_o}}\right)}{1-2 \exp\left(- \frac{E_b}{2N_o}\right)} \qquad (5.13)$$

and

$$P_b < \frac{Q\left(\sqrt{5 \frac{E_b}{N_o}}\right)}{\left[1-2 \exp\left(- \frac{E_b}{2N_o}\right)\right]^2} \qquad (5.14)$$

for E_b/N_o ratios large enough so that the denominator of (5.13) is positive.

This example illustrates the two main problems with this technique: that of determining the generating function and that of computing or bounding P_i. For small error rates only the first few terms in the summations of (5.6) and (5.7) contribute significantly to the bounds. So a good method of using this technique is to use exact expressions or tight bounds for the P_i of the first few terms and then to bound the remaining P_i by bounds of the form

$$P_i < C_o D_o^{\,i} \qquad (5.15)$$

where C_o and D_o are quantities which do not depend on i. In the example above we used

$$C_o = Q\left(\sqrt{2d_f R \frac{E_b}{N_o}}\right) \exp\left(d_f R \frac{E_b}{N_o}\right) \qquad (5.16)$$

and

$$D_o = \exp\left(- R \frac{E_b}{N_o}\right) \qquad (5.17)$$

for $i \geq d_f + 1$.

With bounds of the form of (5.15) the transfer function only has to be evaluated for a particular $D=D_o$. This can be accomplished with a computer using the state equations [29].

$$\underline{S} = \underline{A}\,\underline{S} + \underline{B} \qquad (5.18)$$

$$T\left(D,N,L\right) = \underline{C}\,\underline{S} \qquad (5.19)$$

where $\underline{S}$ is a column vector whose components are equal to the branch transfer functions from the all zero state to each of the nonzero states, $\underline{A}$ is a matrix whose components a_{ij} are the branch transfer functions from the j th to the i th nonzero state, $\underline{B}$ is a column vector whose components are the branch transfer functions from the initial zero state to the i th nonzero state, and $\underline{C}$ is a row vector whose components are the branch transfer functions from the i th nonzero state to the final zero state.

A computer program has been written [29] to compute (5.7) for specific D values. This no-quantization bound is compared with simulation results in Figures 5.11, 5.12, and 5.13.

Figures 5.19, 5.20, 5.21, 5.22 and 5.23 give this bit error probability bound versus E_b/N_o for R= 1/2, 1/3, 1/4, 2/3 and 3/4 codes, respectively, of [25], and [26] with biphase BPSK modulation, additive white Gaussian noise and no quantization. With 3-bit quantization an additional .25 dB is required and with hard quantization an additional 2 dB is required.

The somewhat poor peformance of the K=7, R=1/4 code of Figure 5.21 is due to the large leading b_i coefficient (i.e., b_{d_f} in Equation 5.4) of that code. A K=7, R=1/4 code with a smaller leading coefficient and the same maximum free distance can probably be found.

5.1.11 Analytical Performance Techniques with Quantization

The bounding technique of Section 5.1.10 can also be used on quantized channels. Let D(Z) be a polynomial describing the metric differences that could occur in comparing an incorrect path with the correct (zero) path.

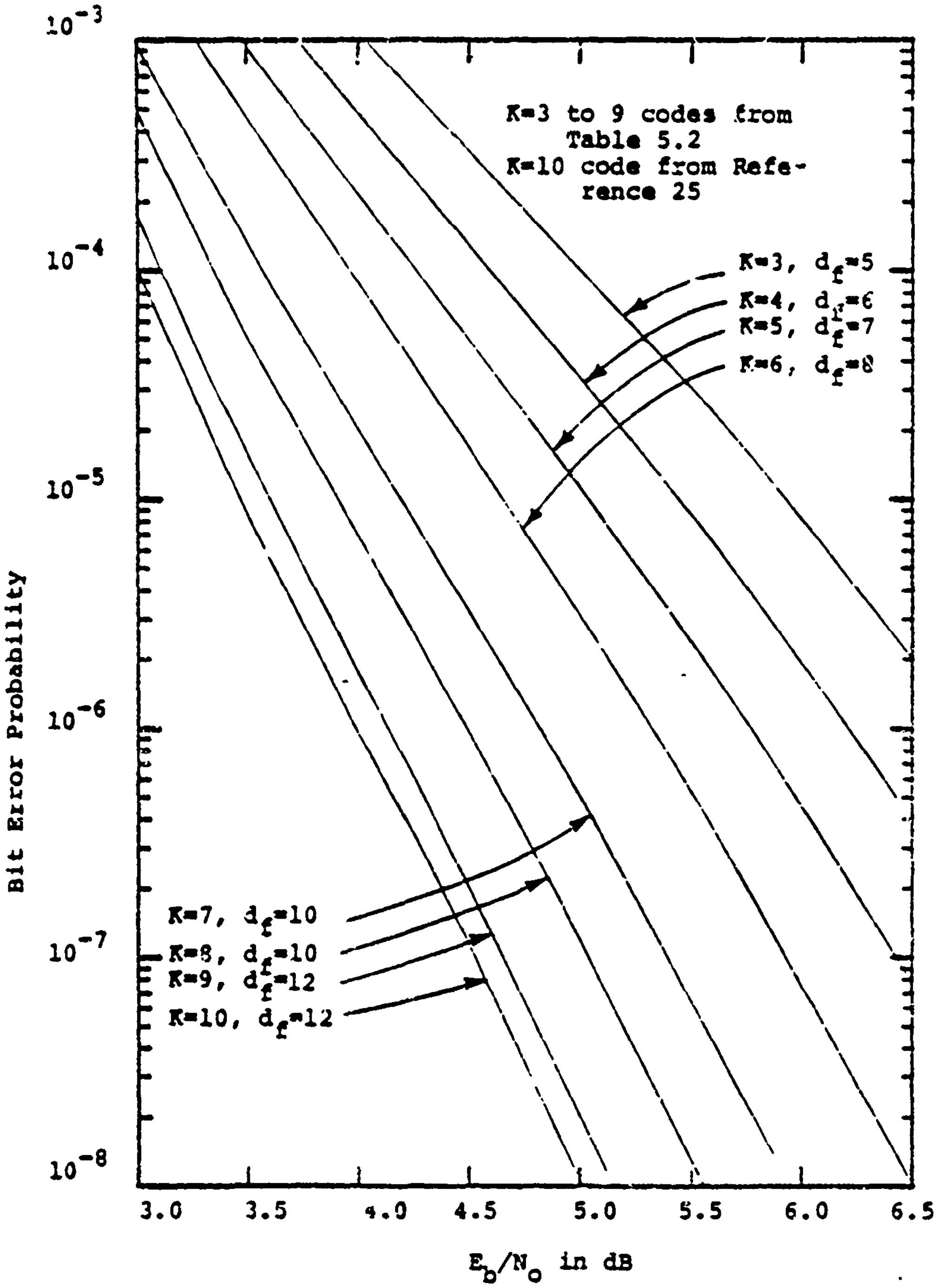


Figure 5.19 Bit error probability versus E_b/N_o performance bounds for several R=1/2 Viterbi-decoded convolutional coding systems with no quantization.

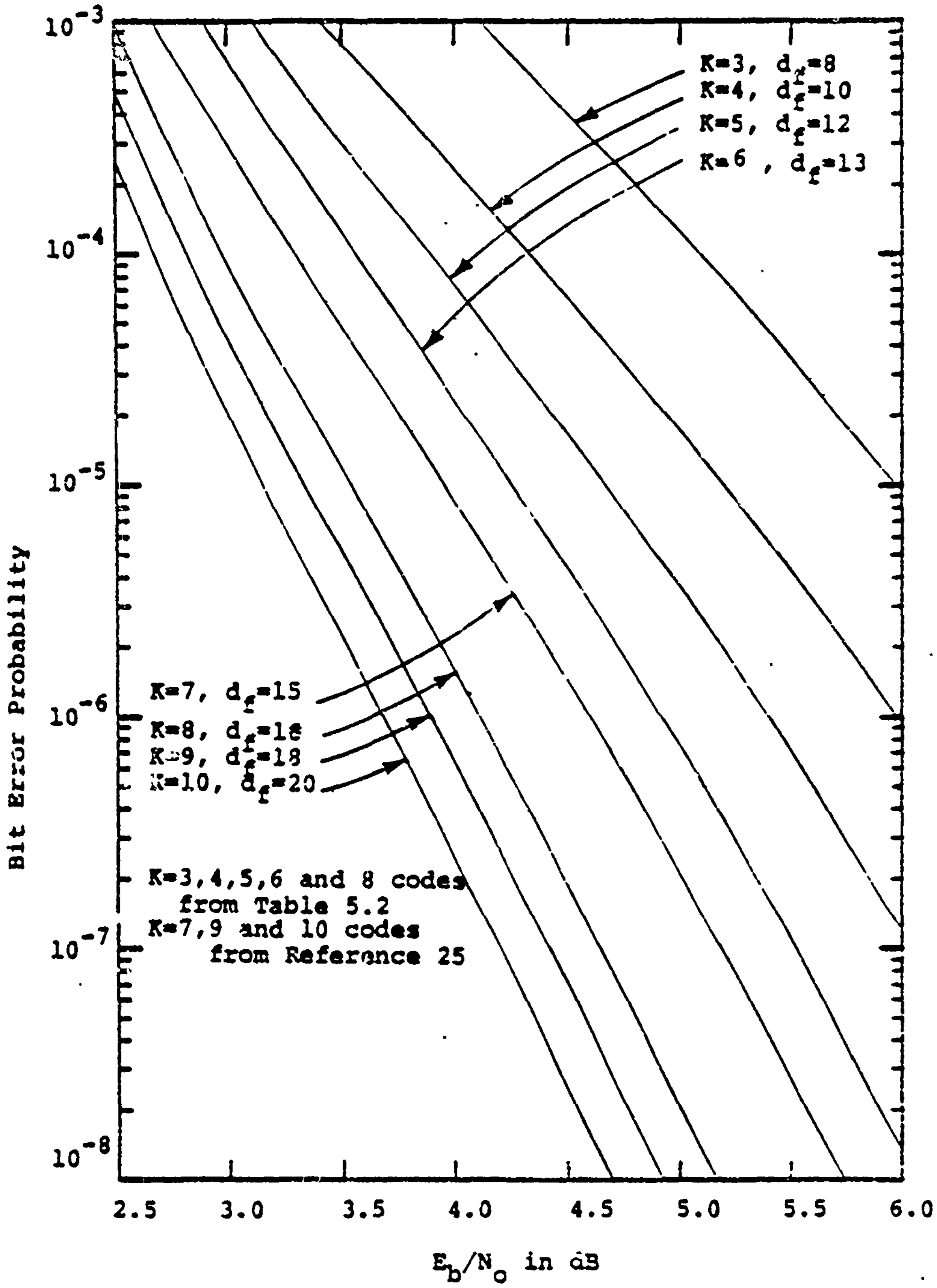


Figure 5.20 Bit error probability versus E_b/N_o performance bounds for several R=1/3 Viterbi-decoded convolutional coding systems with no quantization.

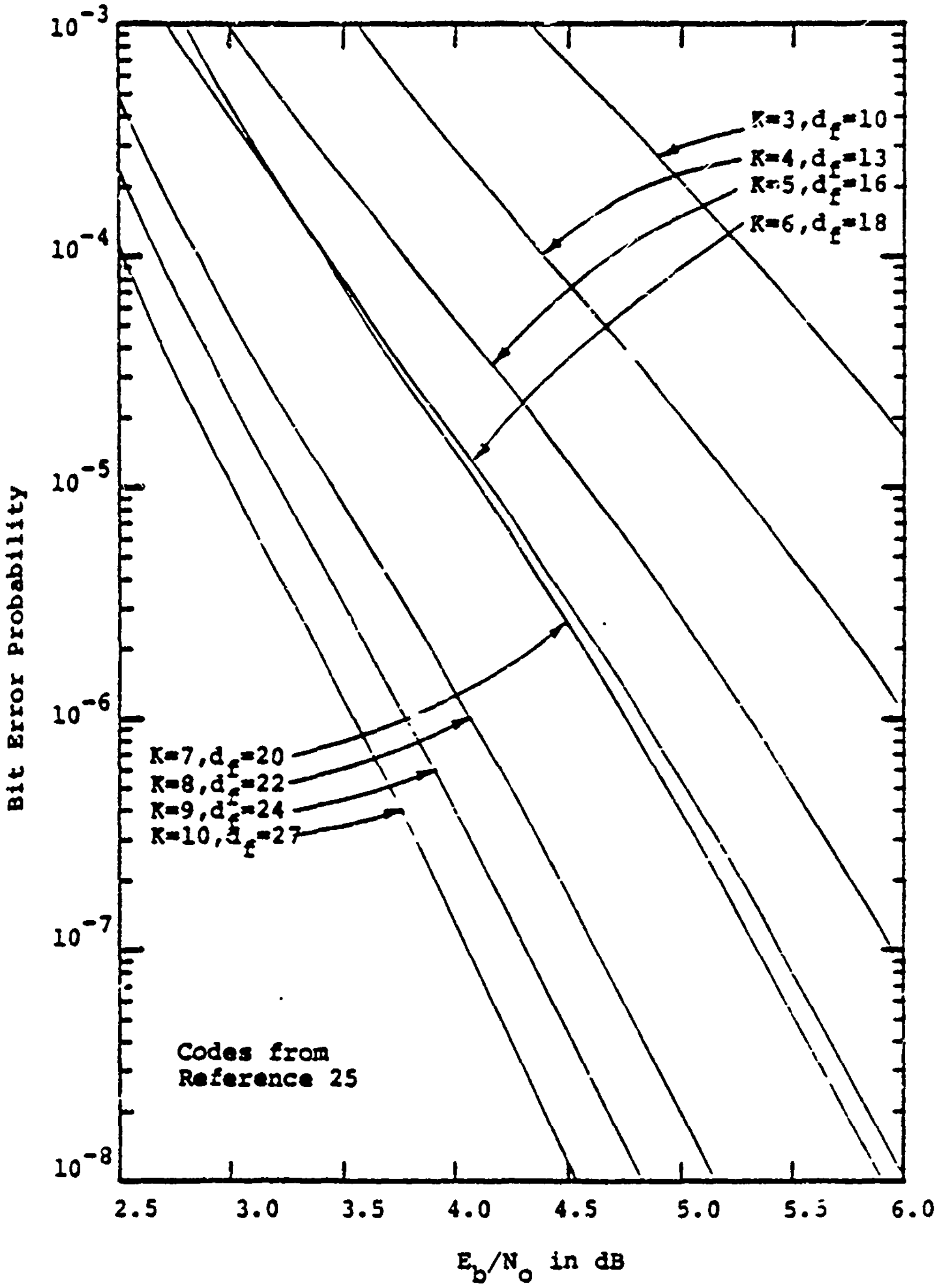


Figure 5.21 Bit error probability versus E_b/N_o performance bounds for several R=1/4 Viterbi-decoded convolutional coding systems with no quantization.

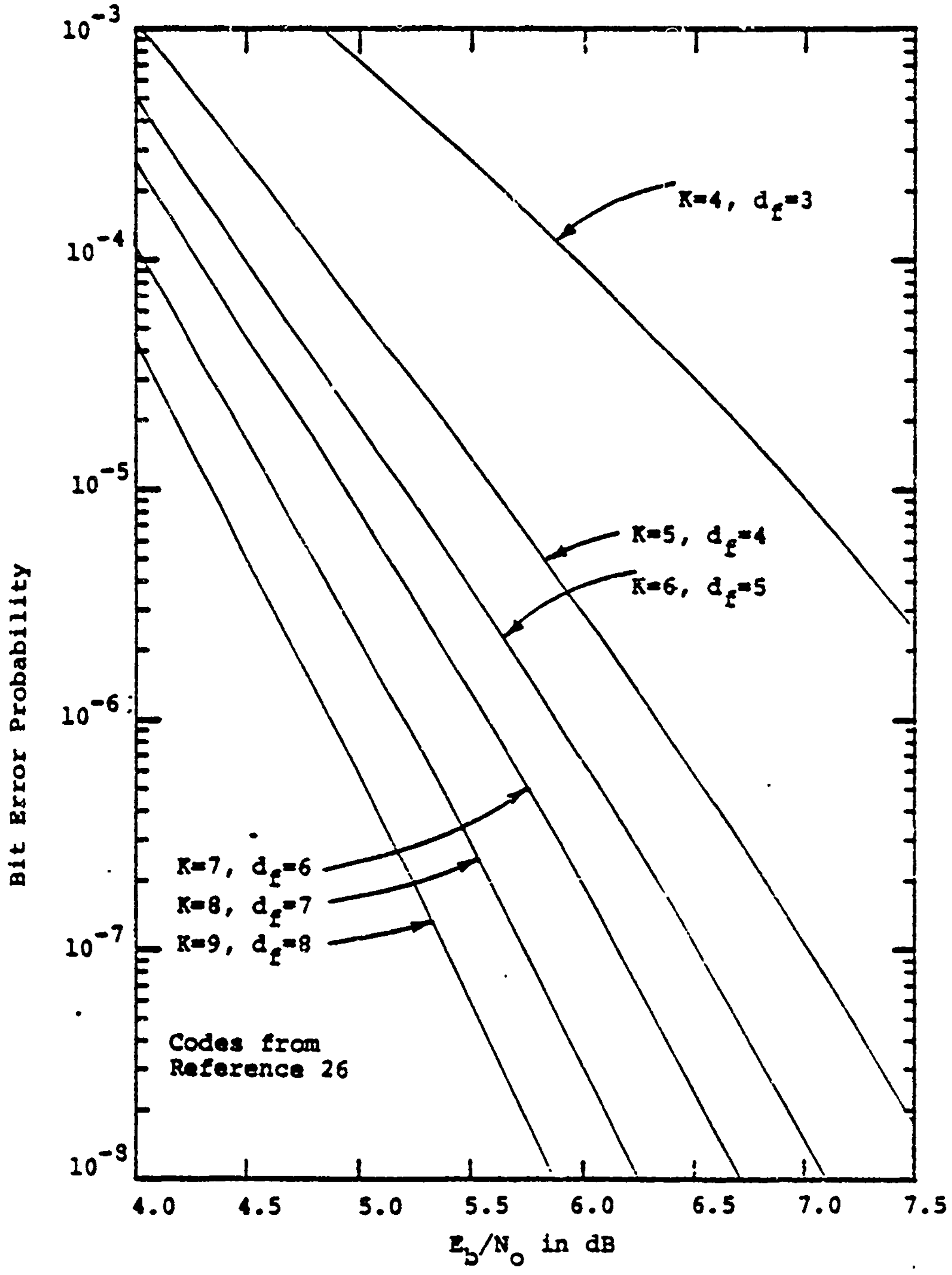


Figure 5.22 Bit error probability versus E_b/N_o performance bounds for several R=2/3 Viterbi-decoded convolutional coding systems with no quantization.

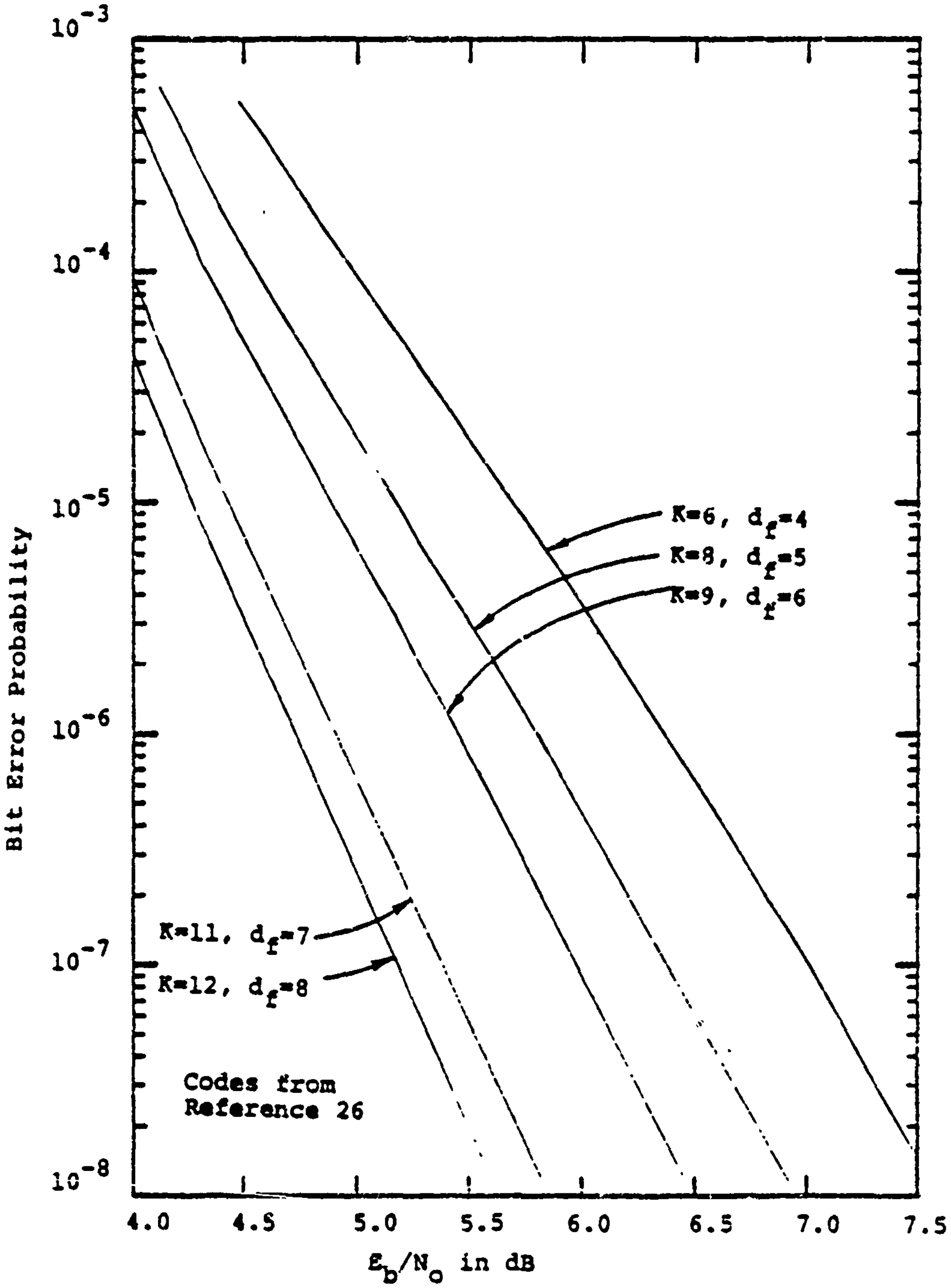


Figure 5.23 Bit error probability versus E_b/N_o performance bounds for several R=3/4 Viterbi-decoded convolutional coding systems with no quantization.

In particular let the powers of Z represent the possible branch metric differences between the incorrect and the correct branch metrics and the coefficient of that term the probability of that metric difference. With n quantization intervals

$$D\left(Z\right) = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} p_{ij} \; Z^{M_i^{(1)} - M_j^{(0)}} \qquad (5.20)$$

where p_{ij} is the probability that the incorrect channel symbol metric is $M_i^{(1)}$ and the correct channel symbol metric is $M_j^{(0)}$

If two paths differed in only one channel symbol, a Viterbi decoder would make an error in comparing the two if the incorrect path metric exceed the correct path metric. So

$$P_1 = \left\{D\left(Z\right)\right\}_+ \qquad (5.21)$$

where

$$\left\{D\left(Z\right)\right\}_+ = \left(\begin{array}{l}\text{Sum of the coefficients of the}\\ \text{positive power terms of } D(Z)\end{array}\right) + \frac{1}{2}\left(\begin{array}{c}\text{Coefficient of the zero power}\\ \text{term of } D(Z)\end{array}\right) \qquad (5.22)$$

The 1/2 factor for the coefficient of the zero power term is to resolve ties randomly. If the paths being compared by the Viterbi decoder differ in i channel symbols, it chooses the wrong path when the sum of the differences of the incorrect and correct metrics corresponding to the channel symbols where the paths differ is positive. For the memoryless channel we have assumed here this probability can be expressed as

$$P_i = \left\{ D^i\left(Z\right)\right\}_+ \qquad (5.23)$$

where the definition of (5.22) applies here with $D^i(Z)$ instead of D(Z). For moderate values of i, (5.23) can easily be computed especially for the integer metric case which is used in practice.

To bound the tail terms (i.e., the infinite sequence of terms remaining in (5.6) and (5.7) after the first few terms have been factored out) a Chernoff bound of the form of (5.15) is used [29]

$$P_i \leq \frac{1}{2} \left[\min_{1 \leq Z} D\left(Z\right)\right]^i = C_o D_o^i \qquad (5.24)$$

With hard receiver quantization and branch metrics of 0 and 1 for the 0 hypothesis and the complement metrics for the corresponding intervals with the 1 hypothesis, D(Z) is simply

$$D\left(Z\right) = \left(1-p\right) Z^{-1} + pZ \qquad (5.25)$$

where p is the probability of a channel symbol error. The bound of (5.24) becomes

$$P_i \leq \frac{1}{2} \left[\sqrt{4p\left(1-p\right)}\right]^i \qquad (5.26)$$

This bound is compared with simulated results with 3-bit quantization in Figures 5.11, 5.12, and 5.13. These figures show that, as expected, at high error rates this union bounding technique is not useful, but for small error rates it is very tight. Similar results have been observed for other codes.

The first few coefficients of the bounds of (5.6) and (5.7) for the codes of Figures 5.11, 5.12 and 5.13 are given in Table 5.5 [24].

5.1.12 Node Synchronization and Phase Ambiguity Resolution

Because of the inherent continuity involved in convolutional coding, code synchronization at the receiver is usually much simpler than in the case of block codes. For convolutional decoding techniques involving a fixed number of computations per bit decoded, such as Viterbi decoding, the decoder initially makes an arbitrary guess of the encoder state to start decoding. If the guess is incorrect, the decoder will output several bits or, at most, tens of bits of unreliable data before assuming

Distance	K=7, R= 1/2 d_f= 10 code of Figure 5.11		K=7, R= 1/3 d_f= 14 code of Figure 5.12		K=9, R= 3/4 d_f=5 code of Figure 5.13	
	a_i	b_i	a_i	b_i	a_i	b_i
5	0	0	0	0	8	42
6	0	0	0	0		201
7	0	0	0	0		
8	0	0	0	0		
9	0	0	0	0		
10	11	36	0	0		
11	0	0	0	0		
12	38	211	0	0		
13	0	0	0	0		
14	193	1404	1	1		
15	0	0	0	0		
16	1331	11633	7	20		
17	0	0	0	0		
18			11	53		
19			0	0		
20			35	184		
21			0	0		

Table 5.5 Upper bound coefficients of (5.6) and (5.7).

steady state reliable operation. Thus, the block synchronization problem does not really exist. There remains the problem of node synchronization and, depending upon the modulation-demodulation technique used, the problem of phase ambiguity resolution. For a rate b/v code, there are v code symbols on each branch in the code tree. Node synchronization is obtained when the decoder has knowledge of which sets of v symbols in the received symbol stream belonq to the same branch. In a purely serial received stream, this is a 1 in v ambiguity.

In addition, modems using biphase or quadriphase PSK with suppressed carriers derive a phase reference for coherent demodulation from a squaring or fourth power phase lock loop or its equivalent. This introduces ambiguities in that the squaring loop is stable in the in-phase and 180° out of phase positions, and the 4th power loop is, in addition, stable at ± 90° from the in-phase position.

Viterbi decoders have been implemented which maintain node and biphase or quadriphase PSK phase synchronization completely within the decoder. One way to resolve 180° phase ambiguities is to use a code which is transparent to 180° phase flips, precode the data differentially and use differential decoding. A transparent code has the property that the bit-by-bit complement of a codeword is also a codeword. Such a code must have an odd number of taps on each of its encoder mod-2 adders. This insures that if a given data sequence generates a certain codeword, its complement will generate the complementary code word.

If the received data is complemented due to a 180° phase reversal, it will still look like a codeword to the decoder, and will likely be decoded into the complement of the correct data sequence. Now decoding to the complement of the sequence input to the encoder is no problem if the data was precoded differentially. This means that information is contained in the occurrence or non-occurrence of transitions in the encoded output sequence rather than the absolute sequence itself. These transitions occur in the same places even if the decoded sequence is complemented.

The major fault with this scheme is that when an isolated bit error occurs in the decoder output, it causes two differentially decoded errors, since two transitions are changed. At first glance, this would seem to indicate a doubling of the output bit error rate. In fact, this doubling does not occur because errors typically occur in short bursts. Two adjacent bit errors, for instance cause only two differentially decoded bit errors. This indicates the possibility of only a small increase in bit error rate with differential encoding-decoding.

In practice this is the case. For the K=7, R=1/2 transparent code of Figure 5.11, using differential encoding-decoding causes an E_b/N_o loss of less than .1 dB for bit error probabilities in the range from 10^{-2} to 10^{-6}.

Another method of resolving node or phase ambiguities is to monitor the metrics and to change the node or phase reference when unsatisfactory (very noisy channel) operation is detected. This and the preceeding technique have been implemented in hardware systems.

5.1.13 Quantization Threshold Levels

With soft receiver quantization the receiver must have an automatic gain control (AGC) circuit to maintain the best quantization threshold levels. Throughout this report we use a uniform quantization. With 0° or 180° biphase modulation and N-bit quantization the quantization threshold levels are at $0, \pm T, \pm 2T, \ldots, \pm (2^{N-1}-1)T$. With additive white Gaussian noise, rate 1/2 coding, and 3-bit quantization (N=3) the best choice of T is about .5 times the standard deviation of the random variable to be quantized, i.e., $.5\sqrt{N_o/2}$. So an estimate of the noise level is required.

Figure 5.24 shows the degradation resulting from an error in the measurement of N_c for the K=7, R= 1/2 convolutional code at a bit error rate of $2X10^{-5}$. This figure shows that this system is not every sensitive to small AGC variations. For bit error rates of from 10^{-1} to 10^{-5} less than a .1 dB larger E_b/N_o ratio is required to maintain the error rate performance due to up to ± 3 dB errors in the measurement of N_o for this code. Similar results have also been obtained for other Viterbi-decoded convolutional coding systems.

5.1.14 Implementation Considerations

The main factors governing the implementation complexity of a Viterbi decoder are the number of state variables (i.e., K-b for R= b/v) and the speed. The K=7, R= 1/2 encoder/decoder with internal node and phase ambiguity synchronization has been implemented with 55 TTL IC chips. This implementation performs the Viterbi decoder comparisons mostly in serial and is capable of operating at any information bit rate up to 100 Kbps. Higher data rates can be obtained by performing the comparisons in parallel. Using such a parallel implementation the same K=7, R= 1/2 encoder/decoder with synchronization capable of operating at information bit rates of up to 10 Mbps has been implemented with about 250 IC chips.

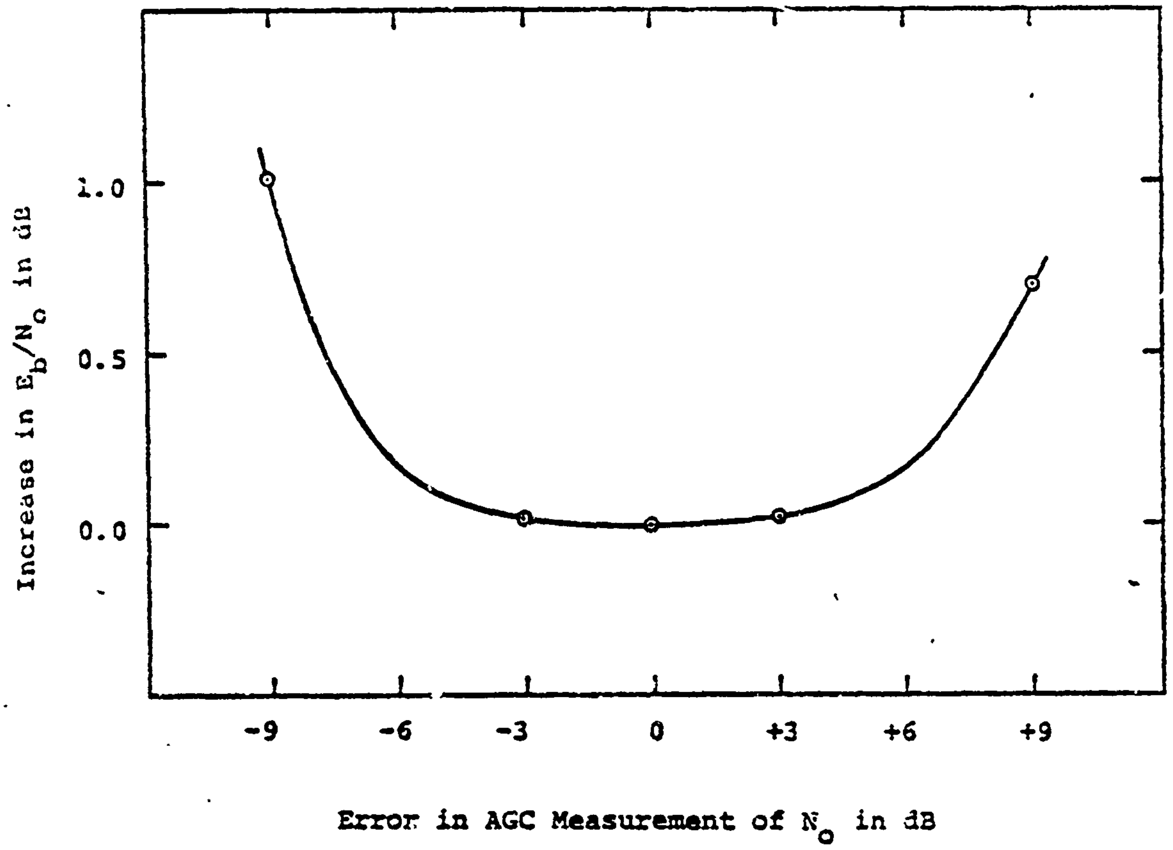


Figure 5.24 Increase in E_b/N_o required to maintain a 2×10^{-5} bit error rate versus error in AGC measurement of E_b/N_o for a k=7, R=1/2 code.

In general, increasing the number of state variables by one approximately doubles the implementation complexity of a Viterbi decoder that performs comparisons in parallel and increases the implementation complexity of a serial type decoder by somewhat less than a factor of two.

At low data rates Viterbi decoders can be implemented with microprocessors. However, unless the microprocessor is also required for other functions, a single decoder of the complexity of a K=7, R= 1/2 code can presently be implemented more economically in hardware. One application where a microprocessor implementation may be preferable to a hardware implementation is where several slow speed, short constraint length decoders are required, such as in some multiple access systems.

Other factors that effect Viterbi decoder implementation complexity are:

(1) The choice of metrics

(2) The method of storing state metrics

(3) The design of the path memory and the selection of the output bit

(4) The method of sharing the state metric calculation

(5) The choice of logic family

(6) The code rate

Comparing the implementation complexity of diffetent coding techniques is difficult. However, when soft decisions are available the implementation complexity of a convolutional coding system is, in general, less than that of a block coding system that achieves the same error rate performance. The main advantages of Viterbi-decoded convolutional coding systems are that they can easily take advantage of soft decision data and that node and phase ambiguity resolution can be resolved internal to the decoder.

The performance of a 3-bit soft decision extended Golay code (see Figure 4.8) is comparable to that of a K=5, R= 1/2 convolutional coding system at bit error rates of about 10^{-5}. While we do not have an accurate chip count for a soft decision Golay decoder implementation, the complete encoder/decoder with synchronization would certainly require more than the 55 chips of the more powerful K=7, R= 1/2, encoder/decoder with synchronization for data rates up to 100 Kbps. In fact, the 100 Kbps, K=7, R= 1/2 implementation has been refined and the chip count reduced to the point that there seems little sense in settling for a shorter constraint length, poorer performance, system just to save a few chips.

When interleavers are used with Viterbi-decoded convolutional coding systems to break up channel noise bursts, it is usually sufficient for the interleavers to be large enough such that any cwo symbols in the same channel noise burst are separated by about 5 constraint lengths of information bits, i.e., 5K v channel bits.

5.2 Sequential Decoded Convolutional Codes

Sequential decoding is a procedure for systematically searching through a code tree, using received information as a guide, with the objective of eventually tracing out tho path representing the actually transmitted information sequence.

Most sequential decoder implementations to data have used some modification of the Fano algorithm. Briefly, the operation of the Fano algorithm is as follows. Starting at the first node in the code tree, a path is traced through the tree by moving ahead one node at a time. At each node encountered, the decoder evaluates a branch metric for each branch stemming from that node. The branch metric is a function of the transition probabilities between the received symbols and the transmitted symbols along the hypothesized branch.

The decoder will initally choose the branch with the largest metric value (corresponding to the closest fit to the received symbols). The metric is than added to a path metric, which is the running sum of branch metrics along the path presently being followed. Along with the path metric, the decoder keeps track of the running threshold T. As long as the path metric keeps increasing, the decoder assumes it is on the right track and keeps moving forward, [illegible]sing T to lie within a fixed constant, Δ, below the path metric. If, on the other hand, the path metric decreases at a particular node, such that it becomes less than T, the decoder assumes it may have made a mistake and backs up. It will then systematically search nodes at which the path metric is greater than T until it finds a path that starts increasing again, or until it exhausts all nodes lying above T. At this point it is forced to lower T, and search again. Eventually it will find a path that appears to have an increasing path metric.

Even when the data is not segmented into blocks, the decoder will eventually penetrate sufficiently deep into the tree, that with high probability the first few branches followed are correct, and will not be returned to by the decoder in a backward search. At this point, the information bits corresponding to these branches can be considered decoded and the decoder may erase received data pertaining to these branches.

A major problem with sequential decoding is the variability in the number of computations required per information digit decoded. The number of computations is a measure of the time required to decode, for a fixed decoding speed in computations per second. A computation is defined, as either looking forward or backward one branch and evaluating and testing the metric involved.

Upper and lower bounds on the probability that the number of computations performed per digit decoded exceeds a variable L have been derived [4]. For large constraint lengths these bounds show that for the average number of computations per digit decoded to remain finite the code rate must be less than the computational cutoff rate of Section 3.2, i.e., $R < R_o$. Actually for finite constraint lengths the average amount of computation remains finite but large for $R > R_o$.

Because of the variability of the amount of computation required, there is a non-zero probability that incoming received data will fill up the decoder memory faster than old outgoing data can be processed. If the decoder tries to search a node for which received data has passed out of buffer memory, an overflow is said to occur. When an overflow occurs, the decoder must have some mechanism for moving forward to new data, reacquiring code synchronization, and starting to decode again. There are presently two techniques for doing this. One involves segmenting the data into blocks. After each block, a fixed constraint length long sequence is inserted. Should the decoder buffer overflow while decoding a given block, it can simply give up decoding that block and jump to the beginning of the next block to resume decoding. Code sync is immediately attained through knowledge of the fixed data sequence preceding a block..

Another overflow recovery technique does away with data blocking. When an overflow occurs, the decoder jumps ahead to new data, and guesses the coder state at that point based upon received data.

For the blocked data case, the probability of failure to decode an L-branch (Lb-data bits for R=b/v) can be expreseed as [4]

$$P_{FD} = kL\left(mT_c\right)^{-\alpha} \qquad (5.27)$$

where k is a constant usually in the range $1 < k < 10$, m is the computational rate in branches/second, and α is the so-called Pareto exponent determined by the relationship.

$$R = \frac{E_o\left(\alpha\right)}{\alpha} \qquad (5.28)$$

Here $E_o(\alpha)$ is a convex function of α which is determined by the channel transition probabilities [5]. This function has the properties that $E_o(0) = 0$ and $E_o(1) = R_o$. Figures 5.25 and 5.26 show this Pareto exponent versus E_b/N_o for several code rates for 3-bit (T= .58) soft quantized and hard quantized additive white Gaussian noise channels, respectively.

The undetected error probability with sequential decoding (as opposed to the failure to decode discussed above) can be made as small as desired by increasing the code constraint length. Long constraint lengths are practical for sequential decoding because decoder complexity

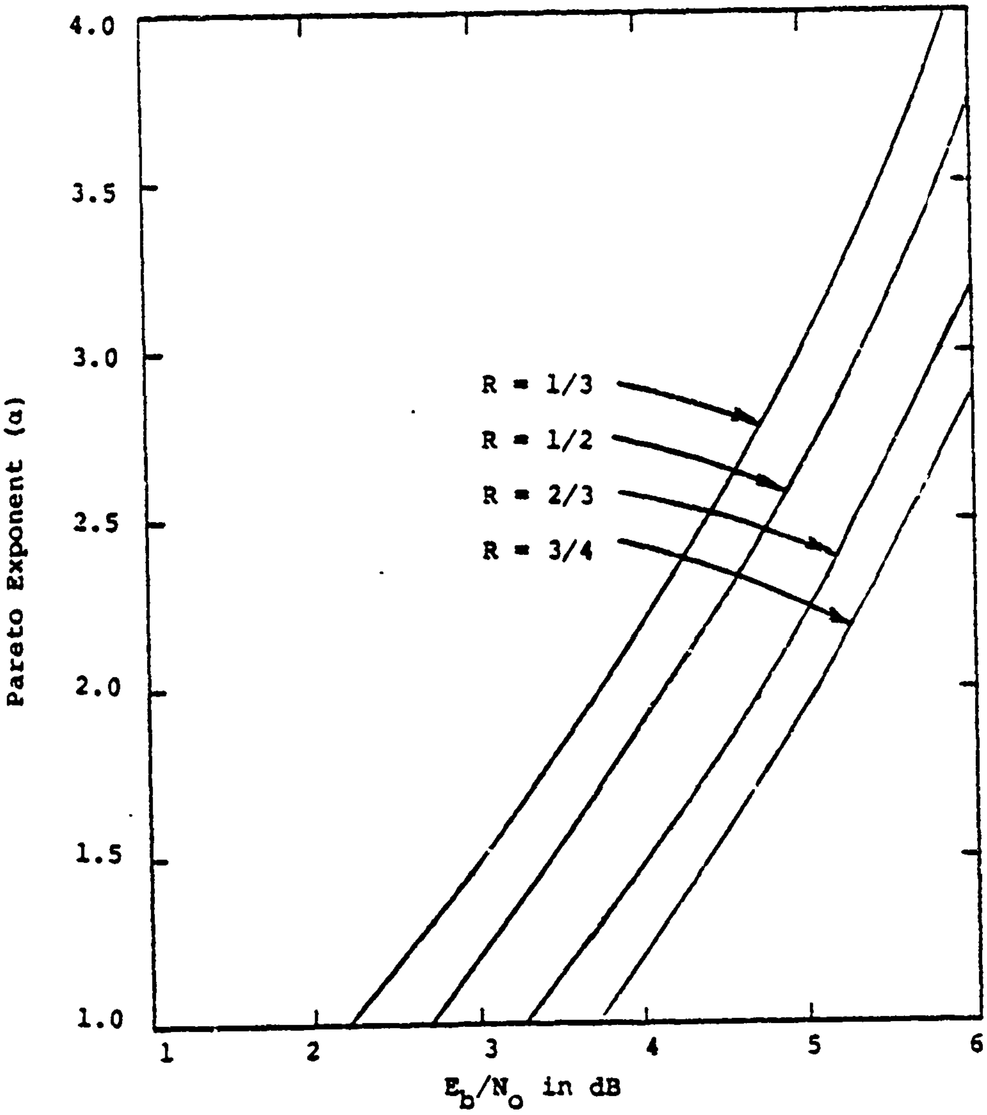


Figure 5.25 Pareto exponent versus E_b/N_o for an AWGN channel with 3-Bit (T=.58) quantization.

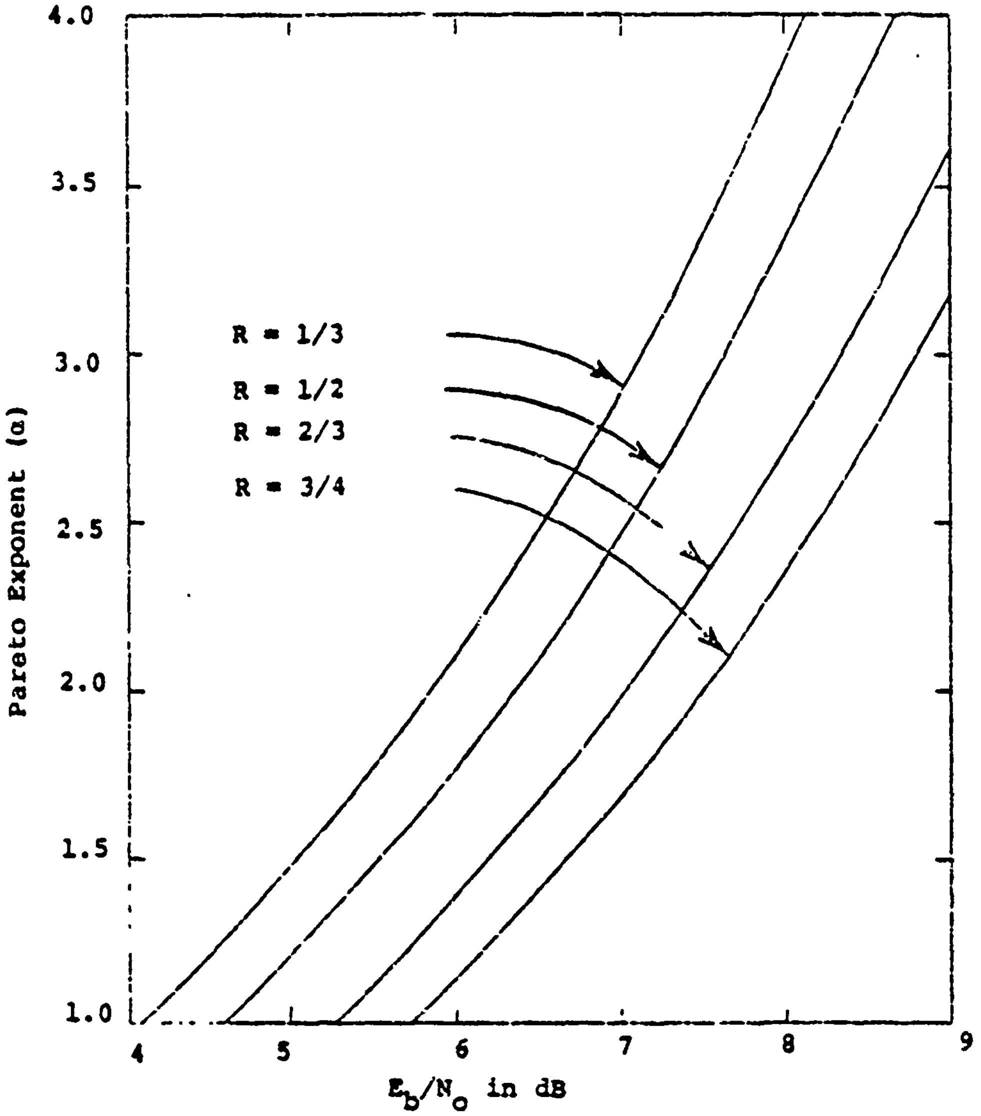


Figure 5.26 Pareto exponent versus E_b/N_o for an AWGN channel with hard quantization.

is only a weak function of the constraint length, unlike Viterbi decoding. This undetected error probability can be determined using the simulation and analysis techniques discussed in the previous section for Viterbi-decoded convolutional codes.

5.2.1 Code Selection

Choosing codes is not as critical for sequential as it is for Viterbi decoding. Decoder complexity is not a strong function of code constraint length; so, the undetected error performance of a code can be improved by increasing K rather than trying to optimize a code for a given value of K. Still there are several reasons for having as good a code as possible.

(1) The constant, k, in (5.27) is somewhat sensitive to the code. Good code distance properties will result in smaller k values.

(2) The encoder replicas in the decoder do grow linearly with K, resulting in some additional cost and complexity.

(3) The guess and restart overflow technique performance degrades with increasing constraint length.

Good long constraint length codes for sequential decoding are given in [30,31].

5.2.2 Performance Results

To illustrate the performance which can be achieved with a sequential-decoded convolutional coding system, this section gives some performance curves for the commercially available LINKABIT LS4816 decoder with BPSK modulation and an additive white Gaussian noise channel. The LS4816 is a high speed, flexible decoder based on the Fano sequential decoding algorithm. The unit operates with rate 1/2 systematic or non-systematic convolutional codes of constraint length selectable between 8 and 48. Hard-or soft-(3 bit) quantized data formatted into frames of from 512 to 4096 code symbols can be processed.

Figures 5.27 and 5.28 show the measured probability of a failure to decode a block (or frame) versus the maximum time allowed for decoding for soft and hard quantization, respectively. The curves in these figures are for the constraint length 24 non-systematic code and a frame length of 1000 information bits. The coded frame format consists of 2000 code symbols plus a terminating sequence of one constraint length of branches (48 code symbols for K=24) The E_b/N_o values given here include the .1 dB loss encountered in adding this terminating sequence. For large maximum allowed decoding times (T_c) the curves in these figures are approximately straight lines with a slope equal in magnitude to the Pareto exponent (α) of (5.27). The information bit rate is the number of information bits per frame (1000 here) divided by T_c.

When the decoder fails to decode a frame and no means

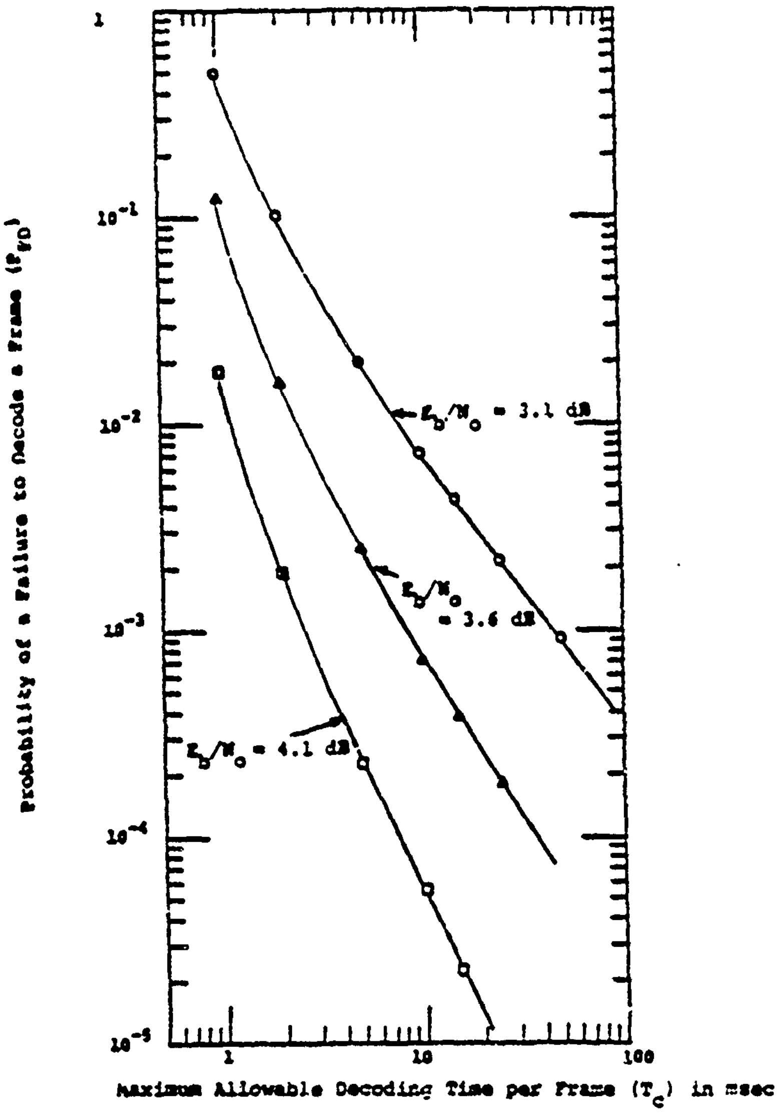


Figure 5.27 Probability of a failure to decode a 1000-information-bit frame for a K=24, R=1/2 sequential decoder with 3-bit quantization (Simulation).

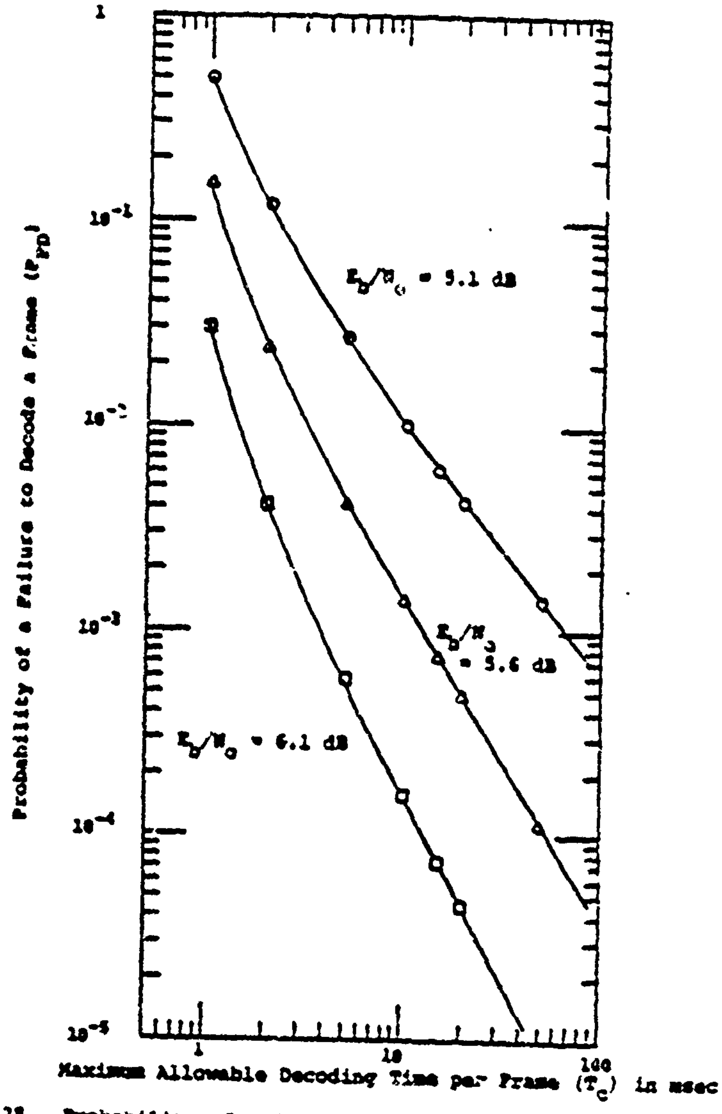


Figure 5.28 Probability of a failure to decode a 1000-information-bit frame for a K=24, R=1/2 sequential decoder with hard quantization (Simulation).

of telling the transmitter to retransmit that frame is available, it may be desirable to have some estimate of the data in that frame. For the systematic code choice the LS 4816 decoder uses the raw undecoded received data bits as the decoder output. For the non-systematic code choice a "quick-look" code [32] is used. The "quick-look" codes have the property that the information sequence can be easily derived from the undecoded received data bits with an error rate which is increased by a minimum amount (about a factor of 2) over the channel error rate.

Figure 5.29 shows the measured bit error probability due to decoding failures versus E_b/N_o for the LS 4816 decoder with the non-systematic, K=24, R= 1/2, 1000-information-bit frame choice on an additive white Gaussian noise channel at a 20 Kbps information bit rate (i.e., T_c= 50 m sec). Bit errors result from the alternate "quick-look" decoding of data when a frame fails to decode in 50 m sec. Since the probability of an undetected error is small, the curves of Figure 5.29 also give the total bit error rate versus E_b/N_o performance.

One possible application of sequential decoded convolutional codes is in a packet satellite communication system. Packet communication involves the transmission of blocks or packets of bits (usually on the order of 1000 bits) over a network with automatic store-and forward and repeat request (ARQ) capabilities. Table 5.6 shows the performance

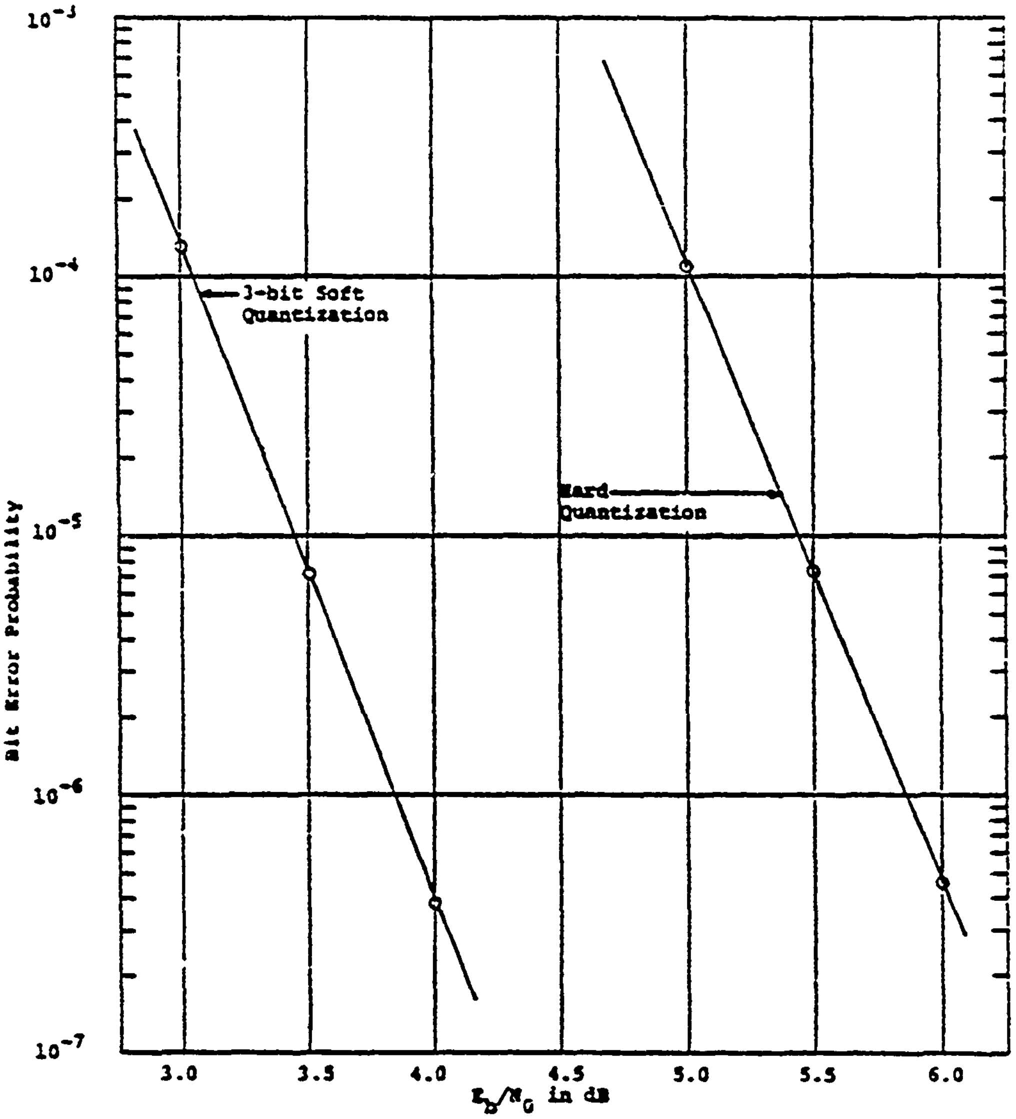


Figure 5.29 Bit error rate due to alternate decoding of 1000-information-bit frames not decoded in 50 ms for a K=24, R=1/2, non-systematic sequential decoded convolutional coding system (Simulation).

INFORMATION BIT ENERGY-TO-NOISE RATIO	PROBABILITY OF NOT DECODING PACKET WITH N-PACKET CODING DELAY AND BUFFER SIZE	
E_b/N_o	N	PROBABILITY OF NOT DECODING
3.2 dB	0.5	7.4×10^{-3}
	1	2.9×10^{-3}
	2	1.1×10^{-3}
	3	4.5×10^{-4}
3.7 dB	0.5	7.4×10^{-4}
	1	2.4×10^{-4}
	2	7.4×10^{-5}
	3	2.5×10^{-5}
4.2 dB	0.5	6.2×10^{-5}
	1	1.8×10^{-5}
	2	5.0×10^{-6}
	3	1.4×10^{-6}

Table 5.6 Measured performance of LS 4816 convolutional encoder-sequential decoder with data rate = 50 Kbps, packet size = 1000 bits, constraint and tail length = 48, code rate = 1/2, and undetected error rate $< 10^{-6}$.

of the LS 4816 decoder for this purpose at a 50 Kbps data rate.

5.2.3 Implementation and Application Considerations

Sequential decoded convolutional coding systems are characterized by the fact that their performance is dependent on the data rate and that the error probability versus Z_b/N_o curves tend to be very steep. For this reason such a system is especially useful in slow to moderate speed applications where very small error rates are required. Another characteristic of sequential decoded systems which influences their application is that the errors tend to occur in long bursts and an indication of the occurence of these bursts can be provided if desired. This characteristic makes this type of system good for applications where the data is blocked and retransmission of unreliable blocks is possible. The large buffers, and thus large decoding delay, required by sequential decoders must also be considered. While Viterbi decoders only have a decoding delay in information bits of about five constraint lengths, sequential decoders will usually have a delay of over 200 bits.

Unlike a Viterbi decoder the implementation complexity of a sequential decoder is only weakly dependent on the constraint length. However, since a sequential decoder must store many branches of received data, the amount of storage required for these branches is a significant factor in the implementation of such a system. Since low data rate codes and soft quantization require more storage per branch these choices increase the implementation complexity of a sequential decoding system.

The LS4816 convolutional encoder-sequential decoder whose performance was given in the previous section is implemented with about 75 standard TTL integrated circuits exclussive of the buffer. For a buffer size of N K bits the buffer requires in addition approximately 5N 2K-bit RAM's.

5.3 Feedback Decoded Convolutional Codes

The Viterbi and sequential decoding methods of the previous two sections are effective ways of achieving small error rates for a variety of channels especially when soft receiver quantization is used. Feedback decoding is a means of achieving more modest coding gains using hard quantized received symbol data. The main advantages of feedback decoding are that the decoder is simple to implement and that interleaving/deinterleaving can easily be included as part of the encoder/decoder.

A feedback decoder traces its way through the code tree by examining a few, say L, branches of data at a time. Initially the decoder examines all possible L-branch-long paths from the initial node and selects as the first branch of decoded data the information symbols corresponding to the first branch on the path most nearly the same as the hard quantized received sequence. Then the decoder shifts one branch, examines all the L-branch-long paths from the terminal node of the first decoded branch, and makes a decision on the second branch of data. This procedure is continued shifting one branch at a time until the received sequence is decoded.

This decoding procedure is called "feedback decoding" because decoding decisions at any given time affect decisions in the future. No feedback channel is used with this type of decoding.

In practice a more algebraic approach to the description given above is usally implemented. An outline of this approach for a systematic rate 1/2 convolutional code is a follows [33]:

(1) Use the hard quantized received symbols to compute a syndrome sequence. This syndrome sequence is similar to the syndrome for block codes except that here it is an arbitrarily long sequence [33].

(2) After each new syndrome bit is computed (i.e., one syndrome bit per branch in this case), a fixed number of syndrome bits (say L) are used to decide if the oldest symbol in the received L-bit information symbol register is correct or not. This decision is performed with a table look-up (i.e., a read only memory integrated circuit chip). If an error is determined, the oldest stored information symbol is corrected (complemented) and provided as the decoder output. Also when an error is determined its effect on the stored syndrome bits is removed by complementing the appropriate syndrome symbols.

Feedback decoding can conceptually be applied to any convolutional code of any rate, systematic or nonsystematic. The main limitation on the implementation complexity is the dacisicn device. For efficient long constraint length codes, it is desirable to make the look-ahead (L) large. However, as L gets large the complexity of the decision device (i.e., the size of the read only memory) soon becomes unreasonable. Threshold decoding [34] is a form of feedback decoding which uses a particularly simple decision mechanism that is practical for large L. Unfortunately, the performance of these codes is quite poor for large constraint lengths and L.

5.3.1 Code Selection

Since the implementation complexity of a feedback decoder is strongly dependent on L, convolutional codes for feedback decoding are choosen to have the best possible distance properties over L branches for a fixed value of L. Bussgang [35] has tabulated rate 1/2 convolutional codes with the largest possible minimum Hamming distance between L-branch paths for L up to 16.

5.3.2 Performance Results

Figure 5.30 shows the binary symmetric channel decoder output bit error rate versus channel error rate performance of four convolutional encoder-feedback decoders marketed by LINKABIT Corporation.

Notice that the performance curves of Figure 5.30 are very near linear on the log-log plot for the range shown. For example, the curve of the R= 1/2 distance 7 code can be closely approximately by

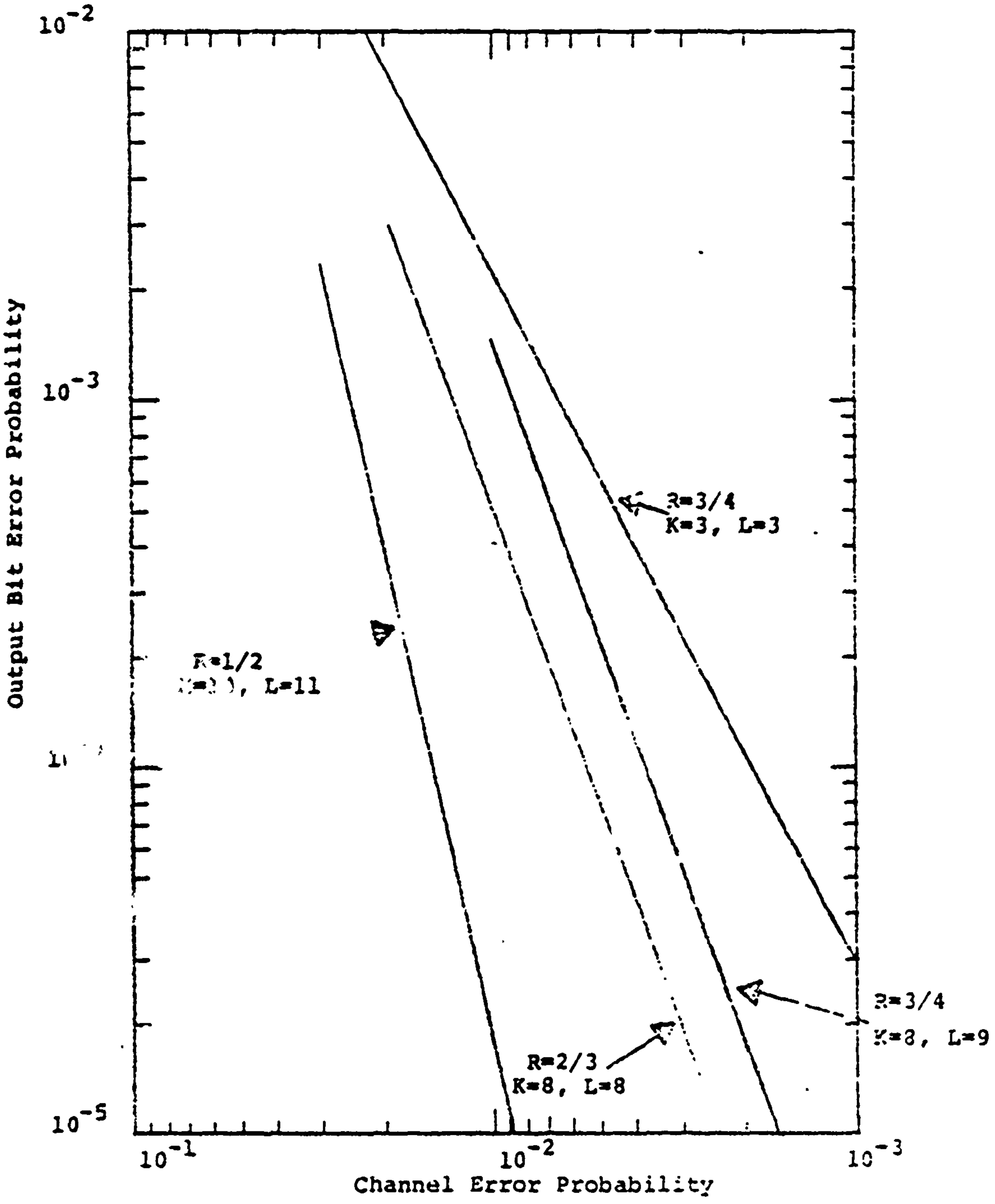


Figure 5.30 Bit error rate versus channel error rate performance of several commerically available feedback-decoded convolutional coding systems.

$$P_b = 2000 \ p^4 \qquad (5.29)$$

Figure 5.31 gives the 375-bit message error probability versus channel error rate performarce of the LINKABIT R= 1/2 distance 7 code. As expected, since the output errors tend to occur in bursts, Figure 5. 31 shows that the M-bit message error probability bound of MP_b is somewhat pessimistic.

5.3.3 Implementation and Application Considerations

One advantage of feedback-decoded convolutional coding is that interleaving/deinterleaving can be easily implemented within the encoder/decoder. N-level interleaving can be implemented by inserting N-bit shift registers between every pair of register stages in the encoder and in the syndrome generator of the decoder. Then the decoder actually decodes N data streams independently. This internal interleaving feature makes feedback decoding attractive for burst error channels such as the HF, troposcatter, and some telephone channels.

As with any rate b/v convolutional code, there exists at the decoder a v-way ambiguity for serial input sequences (i.e., node synchronization is required). This ambiguity can be resolved by observing the rate at which the decoder makes corrections and changing the synchronization position if too many errors seem to be occurring.

The encoder/decoders with node synchronization and 256-way internal interleaving/deinterleaving for the codes

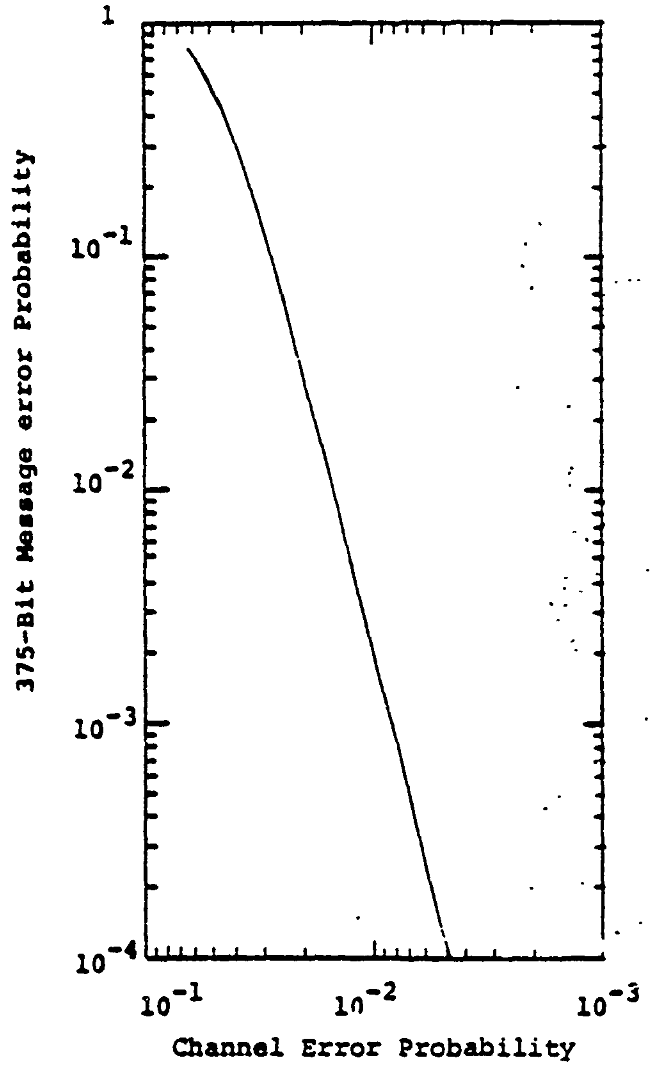


Figure 5.31 Message error probability versus channel error rate for a K=10, L=11, R=1/2 feedback-decoded convolutional coding system.

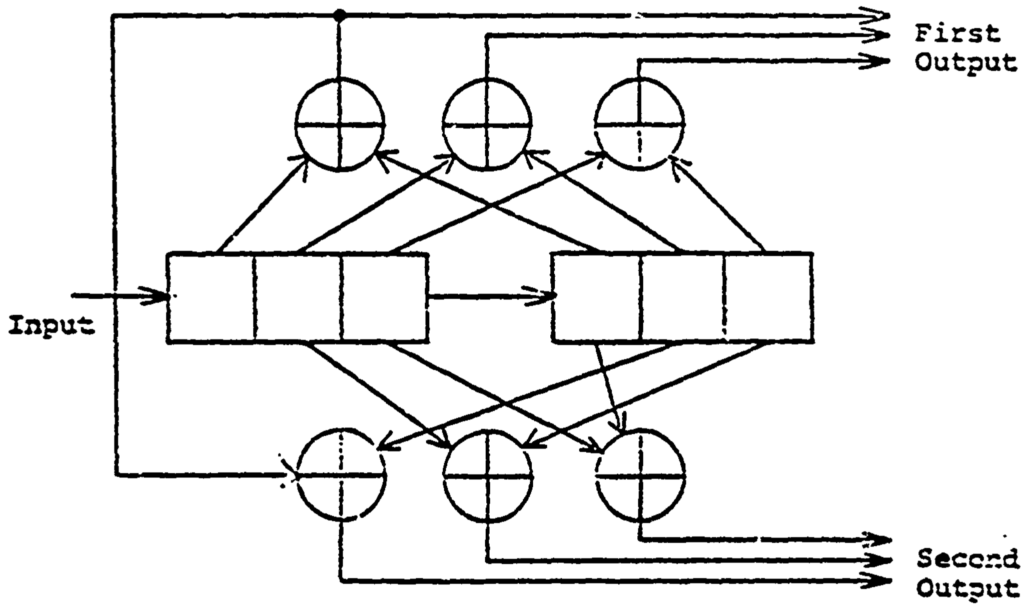
of Figure 5.30 have all been implemented with less than 40 TTL integrated circuit chips.

6.0 Nonbinary Modulation Convolutional Codes

While Section 5 was concerned with binary convolutional codes, nonbinary convolutional codes can easily be defined by replacing the binary symbols and modulo-2 arithmetic by symbols and arithmetic over a nonbinary finite field. Such nonbinary coding systems are especially useful with nonbinary modulation systems in which the modulation symbols are matched to the code symbols. For example, a system with 2^k-ary orthogonal signal modulation is ideally suited to a coding system with 2^k-ary symbols.

One class of nonbinary convolutional codes which has proved effective in obtaining small error rates on channels with fading and non-Gaussian interference is that of Viterbi-decoded dual-k convolutional codes. These codes operate with 2^k-ary symbols and are for channels with 2^k-ary orthogonal signal modulation (e.g., MFSK).

Figure 6.1 shows the encoder for a R = 1/2, dual-3 code which has been implemented. The encoder shifts in one k-bit (k=3 here) information symbol at a time and for each k-bit input, two 2^k-ary output symbols are generated. Each 2^k-ary output is used to select an orthogonal modulator signal. A rate 1/v encoder would, in general, use different linear combinations of the present and past information symbols to produce v, 2^k-ary outputs. Figure 6.2 shows a general 2^k-element finite field representation of a rate 1/v, dual-k convolutional encoder.



Modulo-2 Addition Used

Figure 6.1 Rate 1/2, dual-3 convolutional encoder.

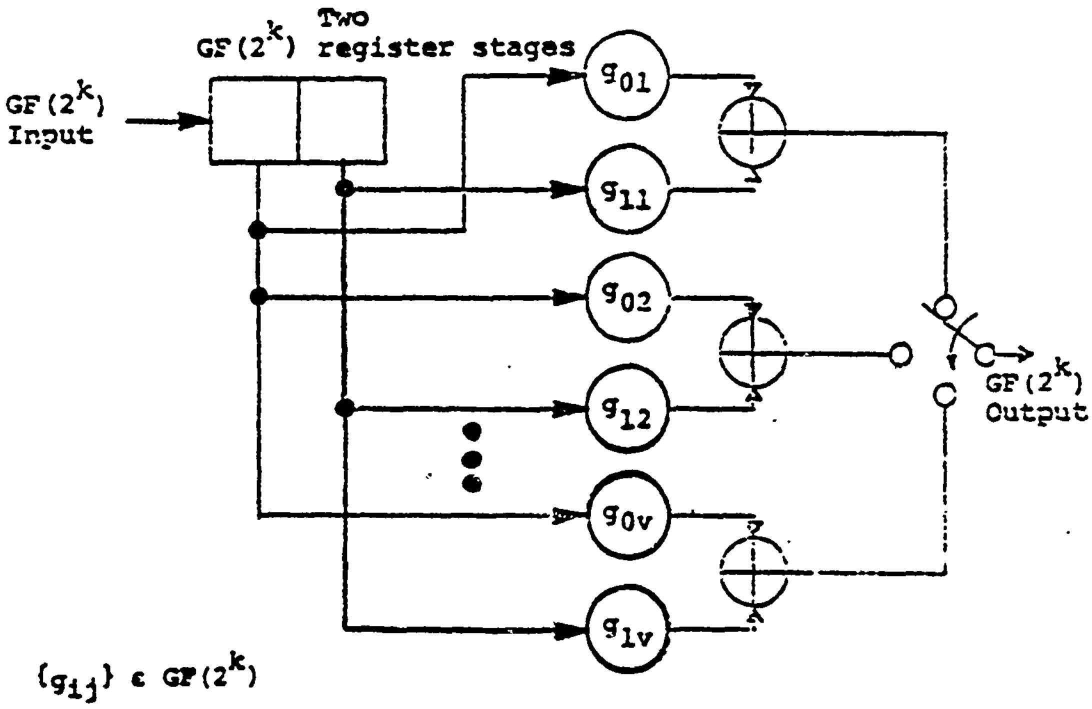


Figure 6.2 Finite field representation of a rate 1/v dual-k convolutional encoder.

The demodulator consists of 2^k filters matched to the 2^k orthogonal signals. So for each channel symbol, 2^k quantized demodulator outputs are generated.

The Viterbi decoder for these codes must make 2^k, 2^k-way comparisons every v channel symbols. That is, after each new branch of data is received the decoder compares the metrics of the 2^k paths entering each node (state) and eliminates all but the best. These 2^k-way comparisons are more difficult to implement than the binary comparisons of rate 1/v codes, but for moderate values of k reasonable implementations are possible. For example, a R = 1/2, dual-3 encoder/decoder with node synchronization and capable of operating at information bit rates of up to 100 Kbps has been implemented by LINKABIT Corporation with about 50 standard TTL integrated circuit chips. These codes have also been implemented in software with a microprocessor.

The error rate performance of these codes can be determined using the simulation or analysis techniques described in Sections 5.1.9 and 5.1.10. The basic simulation procedure is a straightforward modification of the binary coding simulation procedures. The one problem is that these codes are usually used on channels with fading and non-Gaussian interference which are harder to model than the additive white Gaussian noise channel commonly used with binary codes.

The transfer function error rate analysis technique for this class of codes is simplified by the fact that the code transfer function for the codes in this class with the best distance properties has been determined in a closed form [37, 38]. For a R = 1/v dual-k code the result is

$$T\left(D,N,L\right) = \frac{\left(2^k-1\right) D^{2v} NL^2}{1-NL\left[vD^{v-1}+\left(2^k-1-v\right) D^v\right]} \qquad (6.1)$$

In this nonbinary case the distance between two code words is the number of symbols in which the code words are different and the powers of D and N refer to the number of channel and information symbol, rather than bit, errors. As in Section 5.1.10 let

$$\left.\frac{dT\left(D,N,L\right)}{dN}\right|_{\substack{N=1\\L=1}} = \sum_{i=2v}^{\infty} a_i D^i$$

$$= \frac{\left(2^k-1\right) D^{2v}}{\left[1-vD^{v-1} - \left(2^k-1-v\right) D^v\right]^2} \qquad (6.2)$$

Then the bit error probability can be upper bounded by

$$P_b < \frac{2^{k-1}}{2^k-1} \sum_{i=2v}^{\infty} a_i P_i \qquad (6.3)$$

where P_i is the probability of an error in comparing two sequences which differ in i symbols. The $2^{k-1}/(2^k-1)$ factor converts the k-bit symbol error probability to bit error probability [2].

To illustrate the application of this bounding technique for dual-k codes, consider a R = 1/2, dual-3 code with 8-ary orthogonal signal modulation (MFSK) and noncoherent (square-law) demodulation on an independent Rayleigh fading channel with no quantization. Then from Section 3.2.3

$$P_i \le p^i \sum_{j=0}^{i-1} \binom{i-1+j}{j} \left(1-p\right)^j \qquad (6.4)$$

where

$$p = \frac{1}{2+\frac{k}{v}\frac{\bar{E}_b}{N_o}} \qquad (6.5)$$

Using (6.4) for the first two terms of (6.3) and the bound [38]

$$P_i < \frac{1}{\sqrt{5\pi}\; 2\left(1-2p\right)} \left[4p\left(1-p\right)\right]^i \quad , \; i \ge 6 \qquad (6.6)$$

for higher order terms (i ≥ 6) yields the results of Figure 6.3. Figure 6.3 also shows the uncoded bound of (3.11) for L=2 and 4 way diversity.

Comparing the curves in Figure 6.3 it can be seen that the coded system is much better than the uncoded L=2 system with the same number of channel symbols per information symbol and about 3 dB better than the uncoded L=4 system

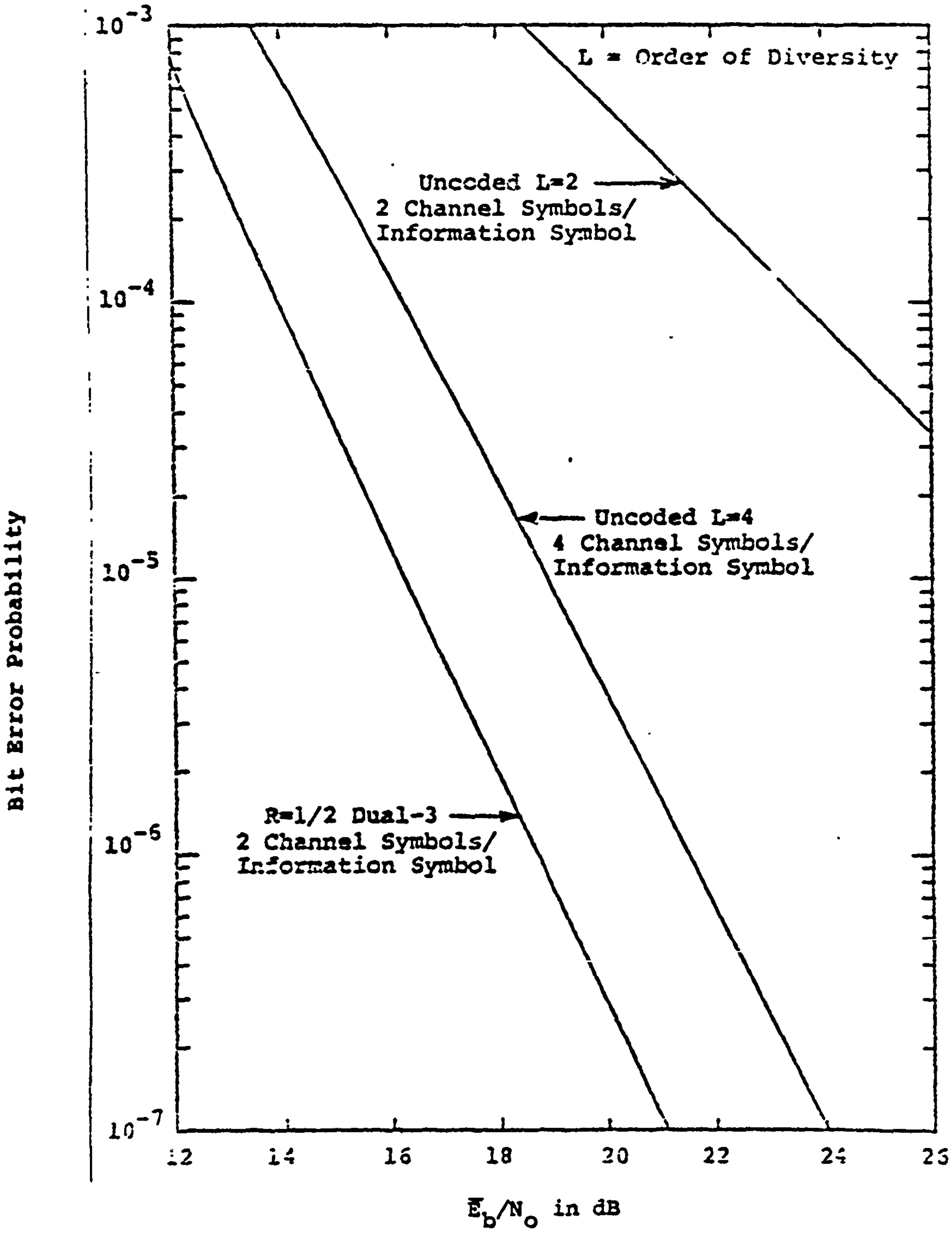


Figure 6.3 Performance of a R=1/2, dual-3 convolutional coding system with noncoherently demodulated 8-ary MFSK on an independent Rayleigh fading channel with no quantization.

which has twice as many channel symbols per information symbol.

7.0 Concatenated Codes

Concatenated coding is a technique which uses two levels of coding as shown in Figure 7.1. Typically, the inner code, i.e., the code that interfaces with the channel, corrects most of the channel errors and then a higher rate (lower redundancy) outer code reduces the error rate to the desired level. The purpose of concatenated coding is either to obtain a small error rate with an overall encoder/decoder implementation complexity which is less than that which would be required by a single coding operation or to improve the error rate performance of an existing coding system. Figure 7.1 also shows interleaving between the coding operations. This is usually required to breakup the error bursts out of the inner coding operation.

7.1 Viterbi-Decoded Convolutional Inner Code and Reed-Solomon Outer Code

One of the most successful concatenated coding systems is a system with a Viterbi-decoded convolutional inner code and a Reed-Solomon outer code with interleaving [24,39,40]. The Viterbi decoder in such a system takes maximum advantage of soft quantized demodulator outputs to improve the channel seen by the outer code by as much as possible. The outer code sees a channel, sometimes called

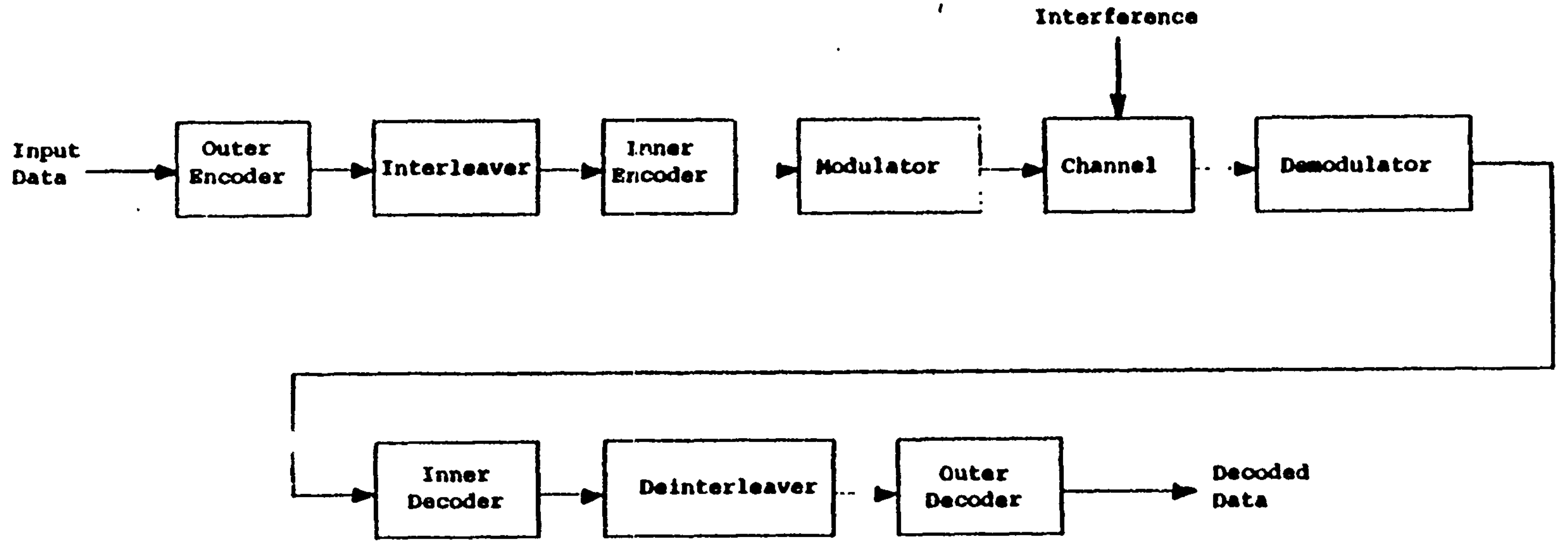


Figure 7.1 Block diagram of a concatenated coding system.

a superchannel, consisting of the inner encoder, the true channel, and the inner decoder which presents hard quantized bursty data with a bit error rate typically around 10^{-2}. While an outer convolutional code could be used, the bursty nature of the errors, the fact that only hard quantized data is available, and the desirability of a high rate code make a Reed-Solomon code, whose symbols are formed from m-bit segments of the binary stream, a good choice for the outer code. Since the performance of a Reed-Solomon code only depends on the number of symbol errors in the block, such a code is undisturbed by burst errors within a m-bit symbol. But the concatenated system performance is severely degraded by highly correlated errors among several successive symbols. Hence the need for symbol (not bit) interleaving between coding operations.

The performance of this concatenated coding system can be determined by measuring the symbol error probability out of the Viterbi decoder by simulation and then, assuming the interleaving makes the symbol errors independent of each other, using binomial type expressions to determine the overall error probability. For example, given the symbol error probability, p_s, versus inner code E_b/N_o for the inner coding system alone, the overall symbol error probability versus total E_b/N_o can be determined from (4.3) with

$$\left(\text{total } \frac{E_b}{N_o}\right) = \left(\text{inner } \frac{E_b}{N_o}\right) \frac{1}{R_{\text{outer}}} \qquad (7.1)$$

where R_{outer} is the outer code rate. The bit error probability can be upper bounded by the symbol error probability (i.e., by assuming that any symbol in error has all the bits in that symbol in error). If in addition, the β_i of (4.3) are upper bounded by (4.4), then the total bit error probability for the concatencated coding system with an m-bit per symbol Reed-Solomon outer code is bounded by

$$P_b < \sum_{k=E+1}^{2^m-1} \frac{k+E}{2^m-1} \binom{2^m-1}{k} p_s^{\,k} \left(1-p_s\right)^{2^m-1-k} \qquad (7.2)$$

where E is the number of symbol errors the Reed-Solomon decoder is capable of correcting and 2^m-1 is the Reed-Solomon symbol block length. The total code rate is the rate of the inner code times the $(2^J-1-2E)/(2^J-1)$ rate of the outer Reed-Solomon code.

Figure 7.2 shows the bit error probability bound of (7.2) versus E_b/N_o performance of this concatenated coding system with a K=7, R=1/2 inner code and an m=8 bit/symbol Reed-Solomon (R-S) outer code with various error correcting capabilities. Additive white Gaussian noise and BPSK or QPSK modulation are assumed. These curves show that for any error rate there is an optimum number of errors that the R-S decoder should be designed to correct. This results

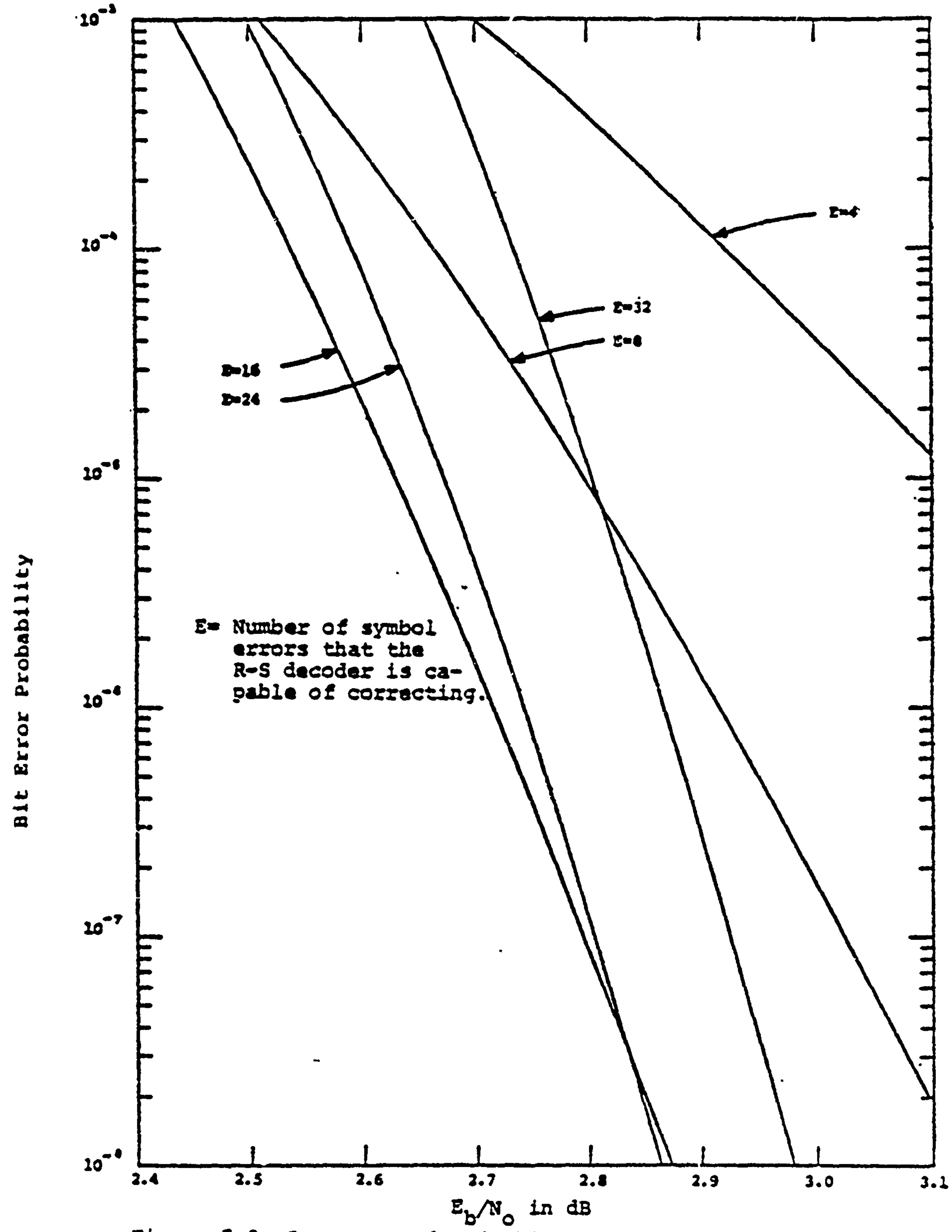


Figure 7.2 Concatenated code bit error probability performance with a K=7, R=1/2 convolutional inner code and an 8-bit/symbol R-S outer code.

from the fact that the more powerful (larger error correcting capability) R-S codes correct more errors but the $1/R_{outer}$, E_b/N_o loss of (7.1) for these more powerful, lower rate, codes is larger. So at some point the $1/R_{outer}$, E_b/N_o loss of (7.1) offsets the E_b/N_o gain obtained by the increased error correcting capability of the code and further reductions in the outer code rate increase the total E_b/N_o required for a given error rate.

Figure 7.3 summarizes the performance of this concatenated system for a K=7, R=1/2 inner code and various R-S outer codes. The optimum number of correctable errors for the m-bit per symbol codes of Figure 7.3 at a bit error rate of 10^{-5} is about 2^{m-4} for m=7,8 and 9 and E=6 for m=6.

The R-S block error probability, where a block consists of $m(2^m-1-2E)$ information bits, can be determined from (4.24). Figure 7.4 shows this block error probability versus E_b/N_o performance for the K=7, R=1/2 inner code and 8 bit/symbol outer codes with various error correcting capabilities.

References 24,39, and 40 give the performance of this concatenated coding system for other constraint length and rate convolutional inner codes. A quick estimate of the performance of this system with a different convolutional code can be obtained by adjusting the E_b/N_o required with the K=7, R=1/2 code by the difference in the E_b/N_o ratios required by the new code and the K=7, R=1/2 codes to achieve

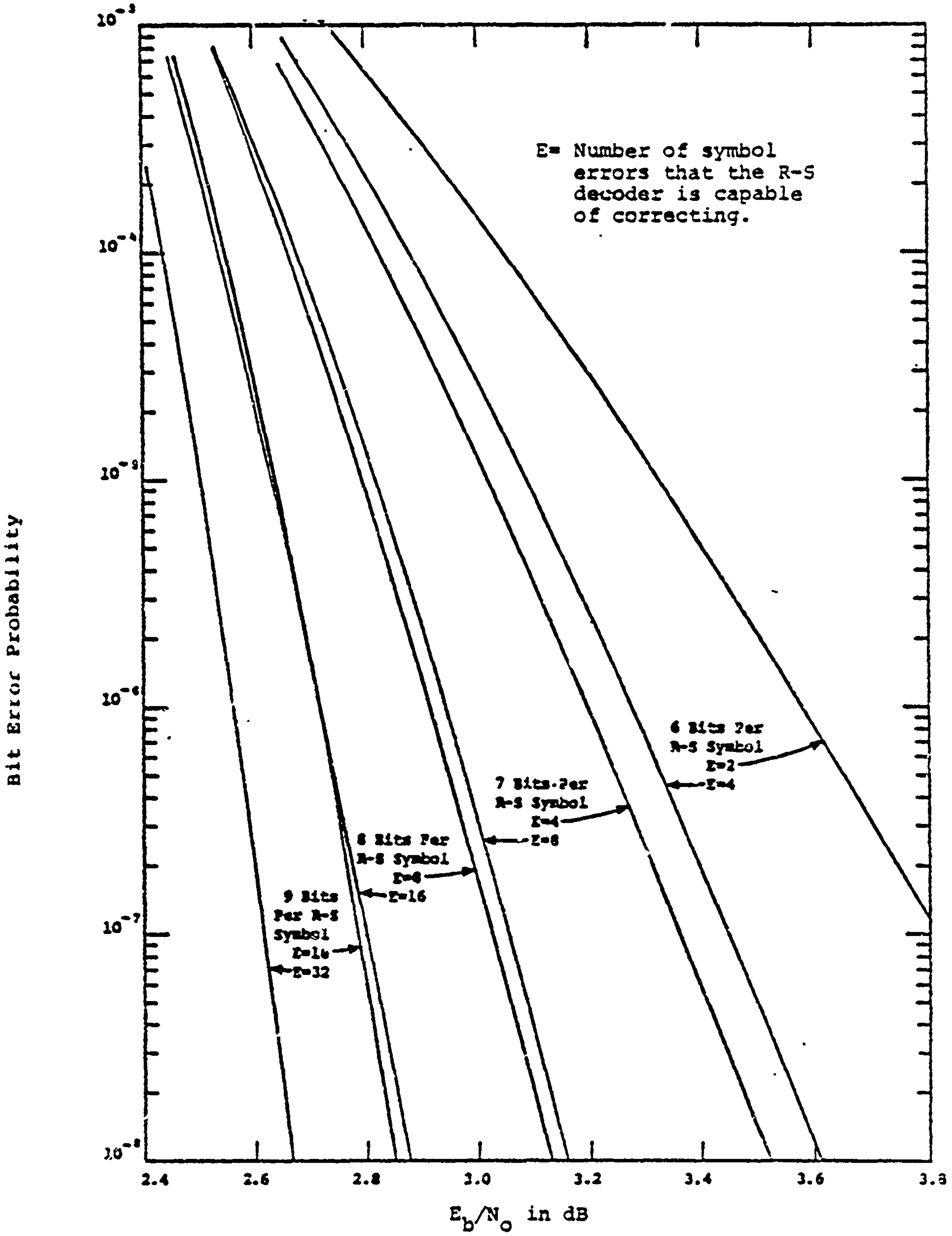


Figure 7.3 Summary of concatenated coding bit error probability performance with a K=7, R=1/2 convolutional inner code and various R-S outer codes.

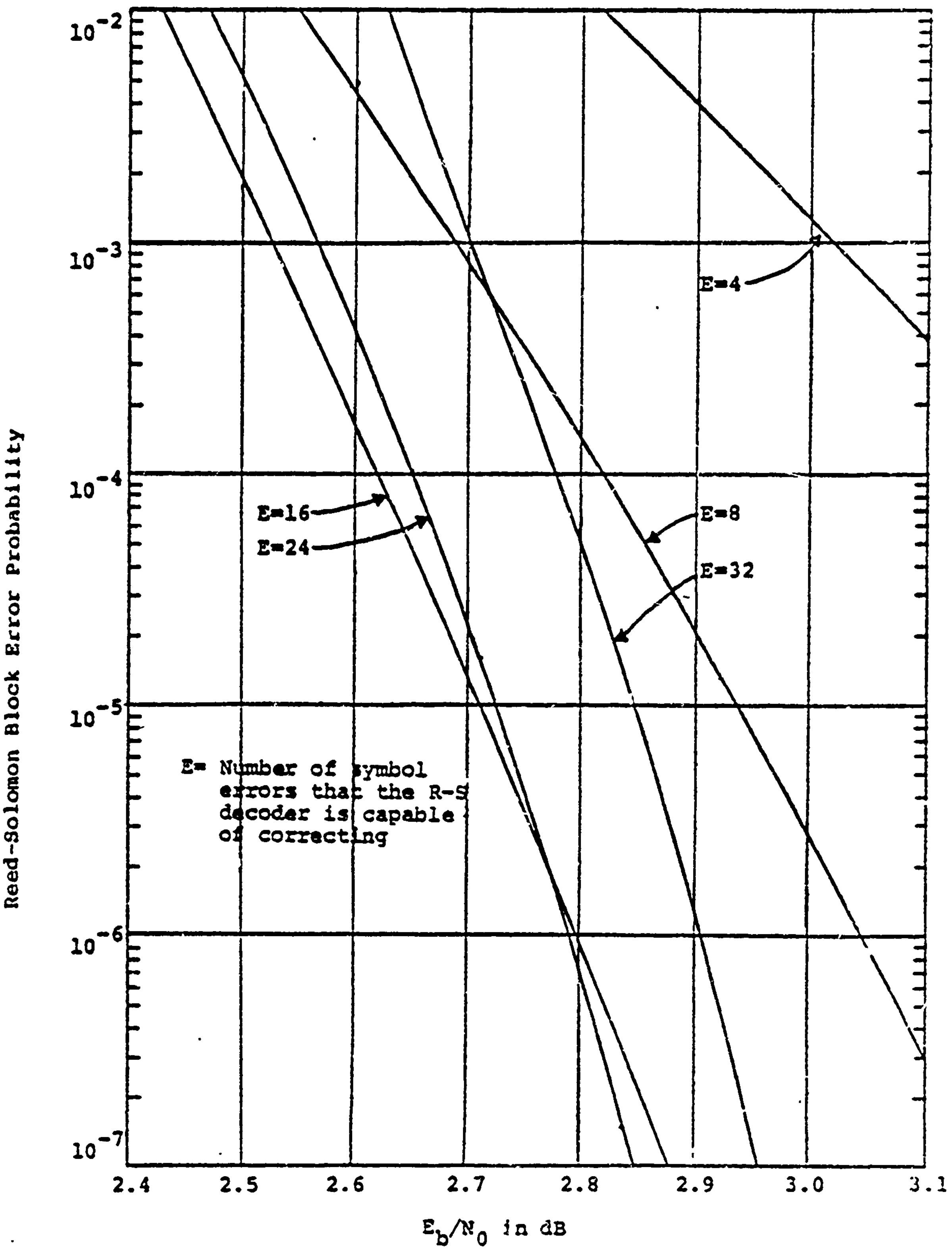


Figure 7.4 Concatenated code block error probability performance with a K=7, R=1/2 convolutional inner code and an 8 bit/symbol R-S outer code

some small inner code, nonconcatenated, error rate, say $P_b = 10^{-2}$.

The implementation and synchronization of this concatenated coding system are discussed in [39,40]. Reference 40 also investigates the sensitivity of this system to nonideal receiver operating conditions such as pha e and AGC errors and shows that it is not overly sensitive to these nonideal operating conditions. In [39] it is estimated that a system with an m=8, E=16 R-S encoder/decoder an interleaver/deinterleaver suitable for operation with a K=7, R=1/2 convolutional inner code, and a block synchronizer all capable of operating at up to 100 Kbps could be implemented with a total of about 220 integrated circuit chips.

7.2 Viterbi-Decoded Convolutional Inner Code and Feedback-Decoded Convolutional Outer Code

Another concatenated coding system which achieves a more modest coding gain is a system with a Viterbi-decoded convolutional inner code and a feedback-decoded convolutional outer code with internal interleaving/deinterleaving. The nice feature of this system is that the interleaving and decoding are simple to implemen

Figure 7.5 shows the bit error probability performance of such a system with a K=7, R=1/3 inner code and the K=8, R=3/4, distance 5 code of Figure 5.30 as the outer code. This figure also shows the performance of the K=7, R=1/3 code

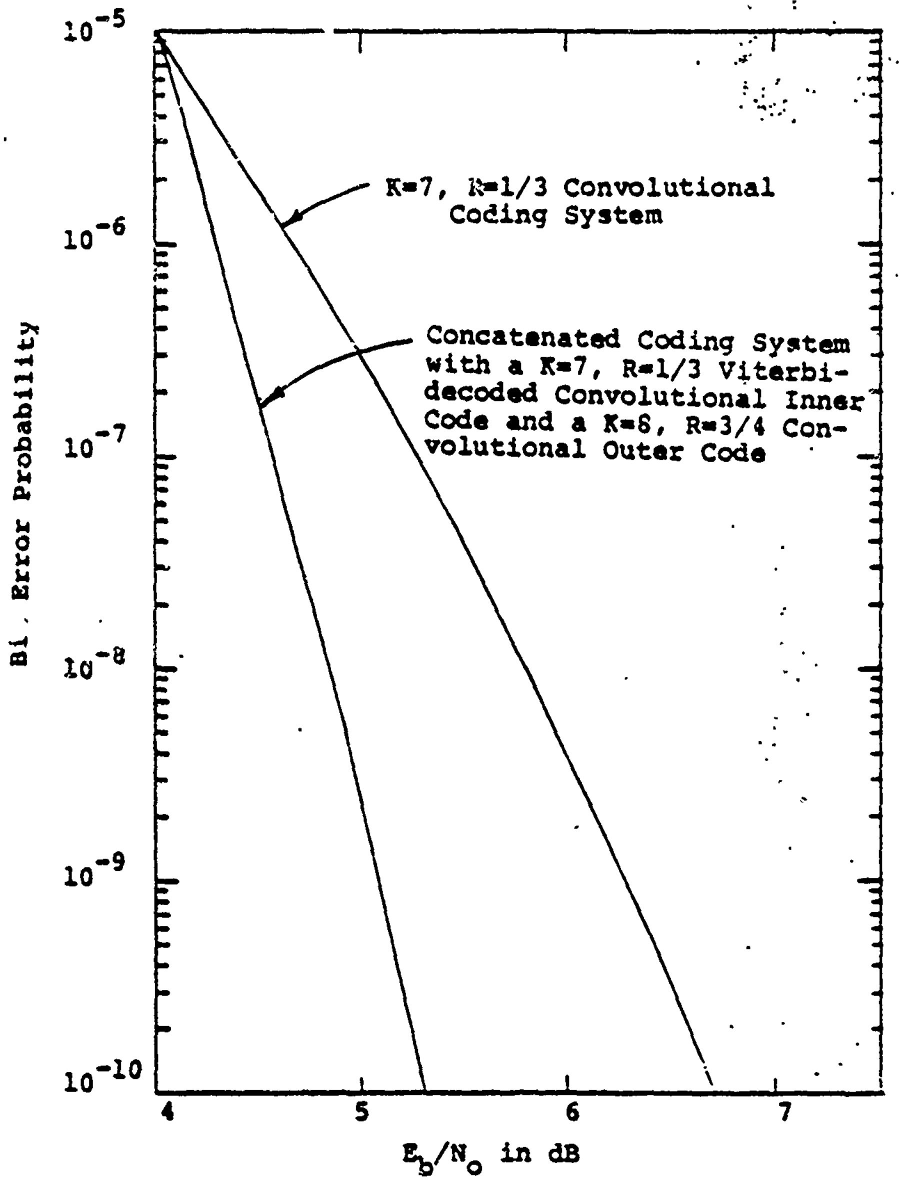


Figure 7.5 Performance of a concatenated coding system with a K=7, R=1/3 Viterbi-decoded convolutional inner code and a K=8, R=3/4 distance 5 feedback-decoded convolutional outer code.

by itself. This concatenated coding system is seen to improve the bit error rate versus E_b/N_o performance of the inner code when the bit error rate is lesɔ than about 10^{-5}.

Appendix A. Glossary of Coding Terminology

1. Addition Modulo-2

   Addition defined for a field with two elements (0 and 1) where 0+0 = 0, 0+1 = 1, 1+0 = 1, and 1+1 = 0. Sometimes denoted by $\oplus$.

2. AWGN

   Additive White Gaussian Noise. Noise with a Gaussian amplitude probability distribution and a constant power spectral density for all frequency ranges which is added to the received signal. In practice as long as the noise power spectral density is constant over the passband of the system, it is considered to be white noise. If the power spectral density of the noise is not constant, the noise is called colored noise.

3. AGC

   Automatic Gain Control.

4. Alphabet

   The set of all possible distinct symbols that a source can generate.

5. ARQ

   Automatic Repeat Request. A feature which allows for requesting the retransmission of blocks, segments, or packets of data in which errors ray have been detected.

6. Bandwidth Expansion

   The ratio of the bandwidth required by a system relative to the bandwidth of some reference system. With coding the bandwidth expansion relative to an uncoded system with the same modulation is the inverse of the code rate.

7. Binary Code

   A code in which the code symbols are from an alphabet with two symbols.

8. Bit

A binary digit and a unit of information in information theory.

9. BSC

Binary Symmetric Channel. A binary input and binary output channel in which the probability of an error is constant independent of the transmitted symbol.

10. BPSK

Binary Phase-Shift Keying. A type of modulation in which two phases (usually 0° and 180°) are used to convey information.

11. QPSK

Quaternary Phase-Shift Keying. A type of modulation in which four phases are used to convey information.

12. PSK

Phase-Shift Keying. A modulation technique that uses phase shifts of a carrier to convey information.

13. FSK

Frequency-Shift Keying. A modulation technique that uses frequency shifts of a carrier to convey informaticn.

14. Block Code

A code in which blocks of information or data symbols uniquely specify blocks of encoded symbols. An (n,k) block code refers to a code with n code symbols for every k information symbols.

15. Branch

In a convolutional code, the set of coded symbols generated for a set of information or data symbols. For a rate R= b/v code a branch has v symbols. A branch also describes the transitions for each set of data symbols in the tree, trellis, and state diagram representations of a convolutional code.

16. Bursts of Errors

In general terms, a sequence of data symbols with a higher than average errcr rate.

17. Channel

The media over which data are transmitted. With coding the channel can be viewed as a device that transforms the encoded symbols at the transmitting station into the decoder inputs at the receiving station.

18. Encoder

A device that maps information or data symbols into coded symbols.

19. Decoder

A device that attempts to recover the transmitting station information or data symbols from noisy receiving station data.

20. Channel Capacity (C)

The maximum rate at which information can be transmitted over a channel (in bits per binary channel symbol) with an arbitrarily small probability of error.

21. Computational Cutoff Rate (R_o)

With arbitrarily large constraint length sequential-decoded convolutional codes, the maximum rate for which the average amount of computation required to decode one bit of data is finite. Also a good limit on the rate of practical coding systems.

22. Code Rate (R)

The ratio of the number of information symbols to the number of encoded symbols.

23. Codeword

An encoded sequence or block of symbols.

24. Codeword Weight

The number of nonzero symbols in a codeword.

25. Coherent Detection

A demodulation technique that requires knowledge of the phase and frequency of the received signal.

26. Constraint Length (K)

With convolutional codes, the total number of binary register stages in the encoder.

27. Convolutional Code

A coding technique in which the encoded sequence is the convolution of the information sequence and the code impulse response.

28. Free Distance (d_f)

For a convolutional code, the minimum number of encoded symbols in which any two arbitrarily long encoded sequences differ.

29. Code Distance (d)

For block codes, the minimum number of symbols in which any two codewords differ.

30. Diversity.

A technique in which two or more independent realizations of a signal are obtained. It is used to combat fading and non-Gaussian interference.

31. E_b

Received energy per information bit.

32. E_b/N_o

Ratio of information bit energy to white noise single-sided (positive frequence only) power spectral density.

33. FEC

Forward Error Correcting. A type of coding where the decoder obtains an estimate of the information sequence without the aid of a feedback channel.

34. Generator Matrix (G)

A matrix mapping of information symbols into encoded symbols in which the codewords are linear combinations of the rows of the matrix.

35. Golay Code

A (23,12) block code with a code distance of 7.

36. Extended Golay Code

A (24,12) block code with a code distance of 8.

37. Hamming Codes

A class of single-error-correcting $(2^m-1, 2^m-1-m)$ block codes.

38. Interleaving

The technique of scrambling or changing the time sequence of codeword symbols.

39. Metric.

A goodness measure used in decoding algorithms.

40. Modem

Modulator and Demodulator

41. Node.

For convolutional codes, the junction of branches in the tree or trellis representations.

42. Noncoherent Demodulation

A type of demodulation in which knowledge of the carrier phase is not required.

43. Orthogonal Signals

Two signals $s_0(t)$ and $s_1(t)$ of duration T are defined as orthogonal if $\int_0^T s_0(t)\, s_1(t)\, dt = 0$.

44. Soft Quantization

A technique of quantizing a demodulator output, defined over a continuum, such that in addition to a decision as to which symbol was transmitted some quality information concerning the confidence we have in that decision is also provided.

45. Systematic Code

A code in which the information symbols appear in the codeword or encoded sequence in unaltered form.

46. Transparent Code

A code for which the complement of any codeword is also a codeword.

47. Tree Diagram

A representation of convolutional codes, useful with sequential or feedback decoding.

48. Trellis Diagram

A representation of convolutional codes useful with Viterbi decoding.

49. Viterbi Decoding

A maximum-likelihood decoding technique used for short constraint length convolutional codes.

50. Reed-Solomon (R-S) Codes

A class of $(2^m-1, 2^m-1-2E)$ E-error-correcting block codes with symbols from a 2^m-ary alphabet.

51. BCH

Bose, Chaudhuri, and Hocquenghem Codes. A class of block codes with well-defined algebraic decoding algorithms.

52. Concatenated Coding

A technique of coding which uses more than one level of coding. For example, with two levels of coding the inner encoder and decoder operate over the true channel while the outer encoder and decoder operate over a channel consisting of the inner encoder, the true channel and the inner decoder.

53. Catastrophic Error Propagation

With convolutional codes, the property of some codes that with an arbitrarily long data sequence a finite number of channel errors can cause an arbitrarily large number of decoded bit errors. This condition can be avoided by proper code selection.

54. Sequential Decoding

A decoding technique for convolutional codes which involves searching through the code tree representation. The search or computation time is a random variable depending on the noise statistics.

55. Feedback Decoding

A decoding technique for convolutional codes in which decoding decisions at any given time affect decisions in the future. No feedback channel is used.

56. Differentially Coherent

A type of demodulation in which the reference signal for demodulation is derived from the receiver input over the previous input symbol.

57. DBPSK or DPSK

Differentially Coherent Binary Phase-Shift Keying.

58. DQPSK

Differentially Coherent Quaternary Phase-Shift Keying.

APPENDIX B. GLOSSARY OF SYMBOLS

C	Channel capacity
d	Code distance
d_f	Free distance
E_b/N_o	Information bit energy-to-noise ratio
E_s/N_o	Channel symbol energy-to-noise ratio
G	Generator matrix of a linear block code
K	Convolutional code constraint length
L	Order of diversity or look-ahead distance of a feedback-decoded convolutional decoder.
R	Code rate
R_o	Computational cutoff rate
●	Modulo-2 addition

REFERENCES

1. A. J. Viterbi, Principles of Coherent Communication New York: McGraw-Hill, 1966.

2. S. W. Golomb (ed.), Digital Communications with Space Applications. Englewood Cliffs, N.J: Prentice-Hall, 1966.

3. C. E. Shannon, "A Mathematical Theory of Communication," Bell System Technical Journal, Vol. 27, pp. 379-423, 1948.

4. J. M Wozencraft and I. M. Jacobs, Principles of Communication Engineering. New York: Wiley, 1965.

5. R. G. Gallager, Information Theory and Reliable Communication. New York: Wiley, 1968.

6. L. E. Miller and J. S. Lee, "Th Probability Density Function for the Output of an Analog Cross-Correlator with Correlated Bandpass Inputs," IEEE Trans. Inform. Theory, Vol. IT-20, pp. 433-440, July 1974.

7. A. Kohlenberg and G. D. Forney, Jr., "Convolutional Coding for Channels with Memory," IEEE Trans. Inform. Theory, Vol. IT-14, pp. 618-626, September 1968.

8. S. Y. Tong, "Burst-Trapping Techniques for a Compound Channel," IEEE Trans. Inform. Theory, Vol. IT-15, pp. 710-715, November 1969.

9. S. Y. Tong, "Performance of Burst-Trapping Codes," Bell System Tech. J., Vol. 49, pp. 477-491, April 1970.

10. H. O. Burton, D.D. Sullivan, and S. Y. Tong, "Generalized Burst-Trapping Codes," IEEE Trans. Inform. Theory, Vol. IT-17, pp. 736-742, November 1971.

11. D. Chase, "A Class of Algorithms for Decoding Block Codes with Channel Measurement Information, "IEEE Trans. Inform. Theory, Vol. IT-18, pp. 170-182, January 1972.

12. "Troposcatter Interleaver Study Report," CNR, Inc., Report Number RADC-TR-75-19 for Rome Air Dev. Center, Air Force Systems Command, Griffiss AFB, N.Y., February 1975.

13. S. K. Leung-Yan-Cheong and M.E. Hellman, "Concerning a Bound on Undetected Error Probability," *IEEE Trans. Inform. Theory*, Vol. IT-22, pp. 235-237, March 1976.

14. W. W. Peterson, *Error Correcting Codes*. Cambridge, Mass.: MIT Press, 1961.

15. E. R. Berlekamp, *Algebraic Coding Theory*. New York: McGraw-Hill, 1968.

16. G. D. Forney, *Concatenated Codes*. Cambridge, Mass.: MIT Press, 1966.

17. C. Cahn, "Performance of a Digital Phase Modulation Communications System," Ramo-Wooldridge Report M170-905, April 1959.

18. R. W. Moss and R. D. Wetherington, "Sensitivity Analysis of Link Transmissions Study," Vol. 1, Georgia Institute Of Tech., Eng. Experiment Station Report prepared for Electronics Systems Division, Air Force Systems Command, Belford, Mass. under Contract F19628-74-C-0180, December 1974.

19. A. J. Viterbi and I. M. Jacobs, "Advances in Coding and Modulation for Noncoherent Channels Affected by Fading, Partial Band, and Multiple-Access Interference," in *Advances in Communication Systems, Vol. 4*, A. J. Viterbi, Ed. New York: Academic Press, pp. 279-308, 1975.

20. A. J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," *IEEE Trans. Inform. Theory*, Vol. IT-13, pp. 260-269, April 1967.

21. L. Lee, "Concatenated Coding Systems Employing Unit-Memory Convolutional Codes and Symbol-Oriented Optimal Decoding Algorithms," Ph. D. dissertation, University of Notre Dame, Notre Dame, Indiana, May 1976.

22. J. L. Massey and M. K. Sain, "Inverses of Linear Sequential Circuits, "*IEEE Trans. Computers*, Vol. C-17, pp. 330-337, April 1968.

23. W. J. Rosenberg, "Structural Properties of Convolutional Codes," Ph. D. dissertation, School Eng. and Appl. Science, University of California, Los Angeles, 1971.

24. J. P. Odenwalder, "Optimal Decoding of Convolutional Codes,"

Ph. D. dissertation, School Eng. and Appl. Science, University of California, Los Angeles, 1970.

25. K. G. Larsen, "Short Convolutional Codes with Maximal Free Distance for Rate 1/2, 1/3 and 1/4," IEEE Trans. Inform. Theory, Vol. IT-19, pp. 371-372, May 1973.

26. E. Paaske, "Short Binary Convolutional Codes with Maximal Free Distance for Rates 2/3 and 3/4," IEEE Trans. Inform. Theory, Vol. IT-20, pp. 683-689, September 1974.

27. J. A. Heller and I. M. Jacobs, "Viterbi-Decoding for Satellite and Space Communication," IEEE Trans. Commun. Technol., Vol. COM-19, pp.835-847, October 1971.

28. A. J. Viterbi, "Convolutional Codes and Their Performance in Communication Systems," IEEE Trans. Commun. Technol., Vol. COM-19, pp. 751-772, October 1971.

29. A. J. Viterbi, "Convolutional Codes and Their Performance in Communication Systems," LINKABIT Corporation, San Diego, Calif., Seminar Notes, 1970.

30. G. D. Forney, Jr., "Use of a Sequential Decoder to Analyze Convolutional Code Structure," IEEE Trans. Inform. Theory, Vol. IT-16, pp. 793-795, November 1970.

31. R. Johannesson, "Robustly Optimal Rate One-Half Binary Convolutional Codes," IEEE Trans. Inform. Theory, Vol. IT-21, pp. 464-468, July 1975.

32. J. L. Massey and D. J. Costello, Jr., "Nonsystematic Convolutional Codes for Sequential Decoding in Space Applications," IEEE Trans. Commun. Technol., Vol. COM-19, pp. 806-813, October 1971.

33. J. A. Heller, "Feedback Decoding of Convolutional Codes," in Advances in Communication Systems, Vol. 4, A. J. Viterbi, Ed., New York: Academic Press, pp. 261-278, 1975.

34. J. L. Massey, Threshold Decoding. Cambridge, Mass.: MIT Press, 1963.

35. J. J. Bussgang,"Some Properties of Binary Convolutional Code Generators," IEEE Trans. Inform. Theory, Vol. IT-11, pp. 90-100, January 1965.

36. "Study of Information Transfer Optimization for Communication Satellites," LINKABIT Corporation, San Diego, California, Final Report under Contract NAS2-6810 for NASA Ames Res. Center, Moffett Field California, NASA Report CR 114561, January 1973.

37. "Verdin Mode Improvement Task," LINKABIT Corporation, San Diego, California, Final Report for Navy Electronic Lab. Center, San Diego, California, (Confidential), April 1975.

38. J. P. Odenwalder, "Dual-k Convolutional Codes for Noncoherently Demodulated Channels," to be published in Proceeding of the 1976 International Telemetering Conf., October 1976.

39. "Hybrid Coding System Study," LINKABIT Corporation, San Diego, CAlifornia, Final Report under Contract NAS2--6722 for NASA Ames Res. Center, Moffett Field, California., NASA Report CR 114486, September 1972.

40. "Concatenated Reed-Solomon/Viterbi Channel Coding for Advanced Planetary Missions: Analysis, Simulations, and Test," LINKABIT Corporation, San Diego, California, Final Report under Contract 953866 for Jet Propulsion Lab., Pasadena, California, December 1974.

41. "Error Protection Manual," Computer Sciences Corporation, Falls Church, Va., Report prepared for Rome Air Dev. Center, Air Force Systems Command, Griffiss AFB, N.Y., 1972.