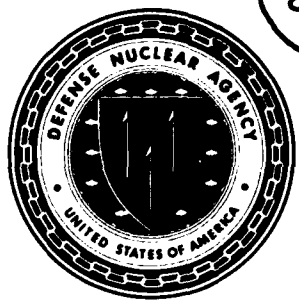


Defense Nuclear Agency
Alexandria, VA 22310-3398

DNA-TR-91-33

Investigation of Coding Techniques for Memory and Delay Efficient Interleaving in Slow Rayleigh Fading

Jay W. Strater
Mission Research Corporation
P.O. Drawer 719
Santa Barbara, CA 93102-0719

November 1991

Technical Report

CONTRACT No. DNA 001-87-C-0169

Destroy this report when it is no longer needed. Do not return to sender.

PLEASE NOTIFY THE DEFENSE NUCLEAR AGENCY, ATTN: CSTI, 6801 TELEGRAPH ROAD, ALEXANDRIA, VA 22310-3398, IF YOUR ADDRESS IS INCORRECT, IF YOU WISH IT DELETED FROM THE DISTRIBUTION LIST, OR IF THE ADDRESSEE IS NO LONGER EMPLOYED BY YOUR ORGANIZATION.

DISTRIBUTION LIST UPDATE

This mailer is provided to enable DNA to maintain current distribution lists for reports. We would appreciate your providing the requested information.

☐ Add the individual listed to your distribution list.

☐ Delete the cited organization/individual.

☐ Change of address.

NOTE:
Please return the mailing label from the document so that any additions, changes, corrections or deletions can be made more easily.

NAME: _____

ORGANIZATION: _____

OLD ADDRESS	CURRENT ADDRESS
_____	_____
_____	_____
_____	_____

TELEPHONE NUMBER: (___) _____

SUBJECT AREA(s) OF INTEREST:

_____	_____
_____	_____
_____	_____

DNA OR OTHER GOVERNMENT CONTRACT NUMBER: _____

CERTIFICATION OF NEED-TO-KNOW BY GOVERNMENT SPONSOR (if other than DNA):

SPONSORING ORGANIZATION: _____

CONTRACTING OFFICER OR REPRESENTATIVE: _____

SIGNATURE: _____

CUT HERE AND RETURN

Director
Defense Nuclear Agency
ATTN: TITL
Washington, DC 20305-1000

Director
Defense Nuclear Agency
ATTN: TITL
Washington, DC 20305-1000

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704.0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	911101	Technical 092087 –123190

4. TITLE AND SUBTITLE
Investigation of Coding Techniques for Memory and Delay Efficient Interleaving in Slow Rayleigh Fading

5. FUNDING NUMBERS
C - DNA 001-87-C-0169
PE - 62715H
PR - RB
TA - RB
WU - DH039580

6. AUTHOR(S)
Jay W. Strater

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Mission Research Corporation
P.O. Drawer 719
Santa Barbara CA 93102-0719

8. PERFORMING ORGANIZATION REPORT NUMBER
MRC-R-1335

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Defense Nuclear Agency
6801 Telegraph Rd.
Alexandria, VA 22310-3398
RAAE/Ullrich

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
DNA-TR-91-33

11. SUPPLEMENTARY NOTES
This work was sponsored by the Defense Nuclear Agency under RDT&E RMC codes B466D RB RB OP140 RAAE 3200E 25904D, B466D RB RB EA104 RAAE 3200A 25904D and B466D RB RB 00140 RAAE 25904D.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

High data rate communication links operating under slow fading channel conditions may have interleaving memory requirements which are too large for practical applications. These requirements can be reduced by employing spacial diversity; however, a less costly alternative is to select coding and interleaving techniques that support memory efficient interleaving. The objective of this investigation has been to find coding and interleaving techniques with relatively small interleaving memory requirements and to accurately quantify these requirements. Toward this objective, convolutional and Reed-Solomon coding with single-stage and concatenated code configurations were evaluated with convolutional interleaving and differential phase shift keying (DPSK) modulation to determine their interleaving memory requirements. Code performance for these link selections was computed by high-fidelity link simulations and approximations over a wide range of E_b/N_o and interleaver span-to-scintillation decorrelation times (T_{il}/τ_0) and the results of these evaluations were converted to interleaving memory requirements. Interleaving delay requirements were also determined and code selections with low interleaving memory and delay requirements were identified.

14. SUBJECT TERMS
Convolutional Coding
Reed-Solomon Coding
Concatenated Coding
Memory and Delay Efficiency
Code Rate Limitation
Practical Code Selection

15. NUMBER OF PAGES
122

16. PRICE CODE

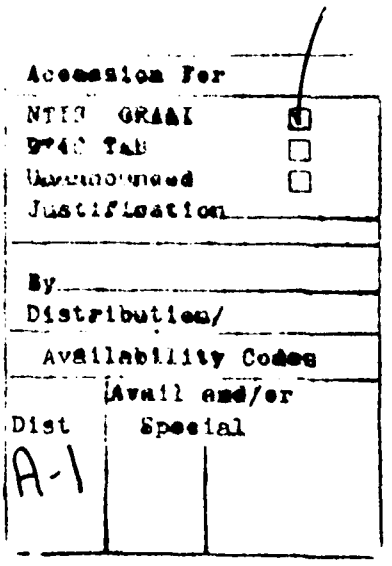
17. Security CLASSIFICATION OF REPORT	18. Security CLASSIFICATION OF THIS PAGE	19. Security CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR

CLASSIFIED BY:
N/A since Unclassified.

DECLASSIFY ON:
N/A since Unclassified.

PREFACE

The author is indebted to Robert L. Bogusch, Allen H. Michelet, and Philip M. Feldman of Mission Research Corporation. Mr. Bogusch provided guidance in the author's coding investigation. Mr. Michelet brought important references to the author's attention. Dr. Feldman developed simulations of a Reed-Solomon encoder and decoder which were used by the author. He also provided information about Reed-Solomon coding theory.



CONVERSION TABLE

Conversion factors for U.S. Customary to metric (SI) units of measurement

MULTIPLY → / TO GET ←	BY → / BY ←	TO GET / DIVIDE
angstrom	1.000000 x E -10	meters (m)
atmosphere (normal)	1.01325 x E +2	kilo pascal (kPa)
bar	1.000000 x E +2	kilo pascal (kPa)
barn	1.000000 x E -28	meter^2 (m^2)
British thermal unit (thermochemical)	1.054350 x E +3	joule (J)
calorie (thermochemical)	4.184000	joule (J)
cal (thermochemical) / cm^2	4.184000 x E -2	mega joule/m^2 (MJ/m^2)
curie	3.700000 x E +1	*giga becquerel (GBq)
degree (angle)	1.745329 x E -2	radian (rad)
degree Farenheit	$t_K = (t_F + 459.67)/1.8$	degree kelvin (K)
electron volt	1.60219 x E -19	joule (J)
erg	1.000000 x E -7	joule (J)
erg/second	1.000000 x E -7	watt (W)
foot	3.048000 x E -1	meter (m)
foot-pound-force	1.355818	joule (J)
gallon (U.S. liquid)	3.785412 x E -3	meter^3 (m^3)
inch	2.540000 x E -2	meter (m)
jerk	1.000000 x E +9	joule (J)
joule/kilogram (J/kg) (radiation dose absorbed)	1.000000	Gray (Gy)
kilotons	4.183	terajoules
kip (1000 lbf)	4.448222 x E +3	newton (N)
kip/inch^2 (ksi)	6.894757 x E +3	kilo pascal (kPa)
ktap	1.000000 x E +2	newton-second/m^2 (N-s/m^2)
micron	1.000000 x E -6	meter (m)
mil	2.540000 x E -5	meter (m)
mile (international)	1.609344 x E +3	meter (m)
ounce	2.834952 x E -2	kilogram (kg)
pound-force (lbs avoirdupois)	4.448222	newton (N)
pound-force inch	1.129848 x E -1	newton-meter (N·m)
pound-force/inch	1.751268 x E +2	newton/meter (N/m)
pound-force/foot^2	4.788026 x E -2	kilo pascal (kPa)
pound-force/inch^2 (psi)	6.894757	kilo pascal (kPa)
pound-mass (lbm avoirdupois)	4.535924 x E -1	kilogram (kg)
pound-mass-foot^2 (moment of inertia)	4.214011 x E -2	kilogram-meter^2 (kg·m^2)
pound-mass/foot^3	1.601846 x E +1	kilogram/meter^3 (kg/m^3)
rad (radiation dose absorbed)	1.000000 x E -2	**Gray (Gy)
roentgen	2.579760 x E -4	coulomb/kilogram (C/kg)
shake	1.000000 x E -8	second (s)
slug	1.459390 x E +1	kilogram (kg)
torr (mm Hg, 0° C)	1.333220 x E -1	kilo pascal (kPa)

*The becquerel (Bq) is the SI unit of radioactivity; 1 Bq = 1 event/s.
**The Gray (Gy) is the SI unit of absorbed radiation.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Continued)

LIST OF ILLUSTRATIONS (Continued)

LIST OF ILLUSTRATIONS (Continued)

SECTION 1

INTRODUCTION

1.1 SIGNIFICANCE AND OBJECTIVE OF INVESTIGATION.

With the advent of higher data rate requirements for military satellite communication links, several problems have arisen in maintaining reliable communication over nuclear disturbed communication channels. One problem is that higher data rates make communication links more vulnerable to signal degradation from frequency selectivity. This occurs when the modulation bandwidth exceeds the channel coherence bandwidth, f_0. Another problem is that higher data rates increase interleaving memory requirements because these are proportional to the interleaving time-span and the data rate. Interleaving time-span must typically be several multiples of the maximum channel scintillation decorrelation time, τ_0, to ensure reliable communication in slow fading conditions.

As an example of these problems, consider a 1 Mbps high data link with standard rate 1/2, constraint length 7 (K=7) convolutional coding with Viterbi decoding, 3-bit quantization, convolutional interleaving, and differential binary phase shift keying (DPSK) modulation. The modulated bandwidth is approximately equal to the modulated symbol rate of 2 MHz, which is the product of the data rate and inverse code rate. If at the beginning of a nuclear event f_0 is as small as 1 MHz, the modulation bandwidth is about twice f_0. This results in significant intersymbol interference and energy loss. If at the end of the Rayleigh fading time period, τ_0 is as large as 1 second and the interleaver span-to-τ_0 ratio (T_{il}/τ_0) is selected as 10 for practical signal-to-noise requirements, the interleaver memory requirement for maintaining a decoded BER of 10^{-5} is over 30 Mbits*.

The focus of this investigation has been on reducing interleaving memory requirements. Frequency selective mitigation was not explicitly investigated because it is well mitigated by employing channel equalization†. Frequency selective mitigation was indirectly included in this investigation, however, by limiting the code rates of investigated codes selections to values no smaller than 1/4. This limited the band-

*This was determined from the evaluations in this investigation.

†Channel equalization mitigates frequency selectivity even for modulated bandwidths as small as one-tenth of f_0 [Bogusch].

width expansion caused by coding and, consequently, it limited link vulnerability to frequency selective degradation.

Interleaving memory reduction was addressed by first considering two different approaches to reducing memory requirements. One approach is to design a communication link with other kinds of signal diversity besides interleaving time diversity. A second approach is to select more efficient coding and interleaving techniques that require less interleaving memory.

In the first approach, adding spatial diversity to supplement interleaving diversity can typically provide enough combined diversity for reliable coding performance even under very slow fading channel conditions, when interleaving alone might otherwise be inadequate. Spatial diversity reduces interleaving span requirements and thereby reduces interleaving memory requirements. However, spatial diversity can impose considerable hardware requirements on a communication system because it requires either multiple demodulators and multiple receiver antennas or multiple transmitters [Bogusch].

The hardware requirements associated with spatial diversity are avoided in the second approach, however. In this approach, interleaving memory requirements can be reduced by selecting coding and interleaving techniques that have relatively small interleaving span requirements for a given E_b/N_0 and BER. Interleaving memory requirements can alternatively be reduced by selecting coding and interleaving techniques with relatively efficient interleaving memory utilization for a given interleaving span and user data rate (data rate prior to encoding). These techniques had not previously been determined, however.

The objective of this investigation has been to find coding and interleaving techniques with relatively small interleaving memory requirements to avoid the need for spacial diversity in high data rate links.

1.2 TWO CODING AND INTERLEAVING APPROACHES.

Two coding and interleaving approaches were identified as having potentially small interleaving memory requirements. One approach is random error correction coding with interleaving for pseudo-random symbol-error dispersal. In this approach, encoder output symbols are interleaved over signal fades so contiguous bursts of received symbol errors are dispersed by deinterleaving prior to decoding. This is the approach currently used in satellite systems, but, for the purpose of achieving greater

interleaver memory efficiency, it can be modified with coding techniques that have better error correction properties and better memory utilization.

A second approach is a coding and interleaving technique that corrects symbol errors from signal fades are not dispersed [Ng]. This approach requires convolutional coding with decomposed convolutional decoding[‡]. It uses the decomposed codes with time-separated sequences of symbol output from separate encoder taps to gain reliable error correction in at least one decomposed decoder.

In this approach, output symbols from convolutional encoder taps are transmitted separately with time separations that span signal fades. At the receiver, the symbol sequences are then reconfigured for decomposed decoding. Symbol errors from signal fades are likely to be isolated in one or a portion of two separate encoder-tap symbol sequences, so one or more decomposed decoders will have a high probability of receiving unperturbed symbol input. Of course, other decomposed codes will encounter concentrated errors so they will fail to correct their input. The final decoded output must, therefore, be selected from the decomposed decoder with the largest Viterbi decoding metrics.

This approach requires specially configured block interleaving to separate encoder tap output. It must have rows (for interleaver output) that equal the number of convolutional encoder taps (inverse code rate) and it must have columns (for interleaver input) that number the symbol separation between encoder-tap output, required for reliable decoding.

This approach has the advantage that the separation between separate encoder-tap output may only be a small multiple of the time-span of average signal fades. The pseudo-random coding/interleaving approach, however, requires interleaving that spans multiple τ_0 and τ_0 is many times larger than average burst-error lengths at practical E_b/N_0. However, the pseudo-random coding/interleaving approach permits more efficient memory utilization because, among other factors, it can be used with convolutional interleaving which has approximately one-forth the memory of block interleaving for comparable interleaver spans.

Regardless of their differences, both coding/interleaving approaches have potential for large interleaving memory savings (and delay savings). Unfortunately, time did not permit evaluation of both approaches. Coding with pseudo-random

[‡]Decomposed convolutional codes are higher rate versions of a lower rate convolutional code. They use a reduced number of taps in a lower rate convolutional code. With rate 1/3 convolutional coding, decomposed codes can be formed as three separate rate 1/2 codes using all 3 possible combinations of the rate 1/3 encoder taps.

interleaving was selected for this investigation and decomposed convolutional coding is recommended for further evaluation.

1.3 SCOPE OF INVESTIGATION.

In the first stage of this investigation, coding and pseudo-random interleaving techniques were assessed for their memory efficiency. This involved determining what coding and interleaving qualities are required for memory efficient interleaving and assessing these qualities using preexisting or derived code performance results. Those techniques with memory efficient interleaving qualities were then selected for further investigation by computer evaluation.

The codes selected at this stage were convolutional and Reed-Solomon codes with single-stage and concatenated code configurations. Code rates, code sizes, and quantization for these codes were selected for good performance and good interleaving memory utilization. Interleavers and other communication link components were also selected at this stage of investigation.

Convolutional interleaving was selected because it has relatively efficient memory utilization and can be implemented relatively simply. It requires approximately 1/4 the memory (and 1/2 the delay) of block interleaving and can be implemented in random access memory (RAM).

DPSK modulation was selected because it is relatively robust in signal scintillation conditions and it is relatively bandwidth efficient. DPSK characteristics make it a good choice for moderately high data rate links in which fast fading is less of a concern than bandwidth efficiency. Coherent binary PSK modulation was another modulation alternative because it is a good choice for very high data rate links without the threat of fast fading. CPSK has the same bandwidth efficiency as DPSK modulation but can operate reliably with several decibels less power than DPSK in slow fading. Unfortunately, time did not permit evaluation of both types of modulation.

Following these selections, the communication links were evaluated by a high-fidelity Monte-Carlo computer program to determine decoded BER performance. Different E_b/N_0 and T_{il}/τ_0 were evaluated with each code selection and the results were converted to interleaving memory and delay requirements. Interleaving delay requirements were included with memory requirements because interleaving delay is also of concern in communication links with slow fading conditions.

1.4 ORGANIZATION OF REPORT.

This report is organized into 6 sections. Following the Introduction, Section 2 discusses the code selections that were made prior to link evaluation.

Section 3 describes the computer program used to evaluate the communication link performance for the coding, interleaving, and modulation selections. Section 3 includes a discussion of how the link components were modeled and how other link parameters, such as data rate and interleaver configurations, were selected.

Section 4 presents equations that were used to convert T_{il}/τ_0 and code parameters to interleaving memory and delay requirements. Section 4 also includes a discussion of interleaver memory reduction from erasure quantization.

Section 5 presents code performance results and interleaving memory and delay requirements. Section 5 also presents the results of interleaver configuration evaluations and erasure quantization thresholds evaluations.

Section 6 presents a summary of the most memory and delay efficient coding selections. Section 6 also discusses recommendations of coding techniques for further investigation.

Section 7 lists the references.

SECTION 2

INITIAL CODE SELECTION

Prior to conducting communication link evaluations, codes were selected for their potential interleaving memory efficiency. This subsection addresses these codes starting with the issues which influenced their selection.

2.1 CODE SELECTION ISSUES.

Several issues influenced coding selections:

- code performance
- memory utilization
- code complexity

Code performance influenced code selections because it determines the interleaving span that is required for E_b/N_0, τ_0, and BER specifications. Interleaving span subsequently influences interleaving memory requirements because memory requirements are proportional to the interleaver span and the data rate (see Section 4).

Codes were selected for the best code performance from preexisting or analytically derived code performance results. Unfortunately, these results were limited to large values of T_{il}/τ_0 and small values of T_{il}/τ_0 were planned for the link evaluations. However, this discrepancy was addressed by favoring lower rate and larger size codes when several code rates or code sizes were seen as good choices. This criterion was used because codes with lower rates or larger sizes have better capability at correcting sequences of multiple errors (bursty errors), which occur with small T_{il}/τ_0 conditions*.

The code rates in these selections were limited to values no smaller than 1/4 to satisfy frequency selective considerations. This limitation did not influence code

*With small T_{il}/τ_0, interleaving is inadequate at randomizing correlated errors from signal fades so it produces bursty errors at deinterleaving.

rate selections, however, because code rates smaller than rate 1/4 were found to be bad choices[†].

Interleaving memory utilization also influenced code selections because it affects interleaving memory requirements and is a function of code rate and quantization (see Section 4). Memory utilization affects memory requirements because it determines how much memory is required to represent an interleaver span for a particular data rate (pre-encoded).

For improved memory utilization, codes were selected with higher code rates than the codes selected for the best performance. Higher code rates were chosen because they reduce encoded data redundancy and thereby improve interleaving memory utilization. Several code quantization levels were also selected to include link designs with good code performance and good memory utilization.

Code complexity influenced code selections because it determines the practicality of the code selections. Code complexity was addressed by imposing a criterion for practical codes selections. Practical codes were required to be codes that are currently used in satellite systems or have comparable requirements to such codes[‡]. Large Reed-Solomon codes with impractical code requirements were also selected. These codes were selected to determine how increases in Reed-Solomon block size reduce interleaving memory requirements.

Besides influencing memory requirements, the code selections also influenced delay requirements. Interleaving delay requirements were smallest for codes selected to maximize code performance. Interleaving delay was larger for codes selected for better memory utilization. Fortunately, link evaluations showed that the code selections with the lowest memory requirements also have the lowest, or close to the lowest, delay requirements.

[†]These selections were based on code performance in slow fading conditions; a condition which could be assumed always present for the high data rate links of primary interest for this investigation. However, if lower data rates are of concern then signal coherence loss from fast fading must be accounted for in the link design. This often requires selecting low rate codes because these codes increase the modulated symbol rate and, thereby, mitigate signal coherence loss.

[‡]Coding complexity is a function of code size and the code rate. Code complexity increases with decreases in code rate and increases in code size.

2.2 CONVOLUTIONAL CODES.

Convolutional codes were chosen because they have good code performance in fading conditions for BER of practical interest. Convolutional codes have relatively small E_b/N_0 requirements at 10^{-5} BER and at 10^{-2} BER, where inner concatenated code performance is needed. Furthermore, in conditions with small T_{il}/τ_0 between 100 and 5, convolutional code performance degrades gradually [Bogusch, page 227].

Convolutional code parameters were selected from preexisting code performance results found in [Bogusch]. These results were simulated with DPSK modulation and slow Rayleigh fading conditions with T_{il}/τ_0 of 100. The large value of T_{il}/τ_0 limited the applicability of these results to small T_{il}/τ_0 conditions. However, as discussed previously, the burst-error correction capability of the codes was emphasized to account for small T_{il}/τ_0 conditions.

Convolutional code rates were assessed from results found in [Bogusch, pages 262 and 265] for rate 1/2, 1/3, and 1/8, constraint length 7 (K=7) coding with Viterbi decoding and soft quantization. These results indicate that, at 10^{-5} BER, rate 1/3 coding has approximately 2 dB better performance than rate 1/2 coding but rate 1/8 coding has only slightly better performance than rate 1/3 coding.

The negligible performance improvement in the code rate reduction from rate 1/3 to 1/8 is because rate 1/8 coding has a larger decoder input BER than rate 1/3 coding. Larger decoder input BER is caused by a reduction in the modulated symbol-to-noise ratio because of larger code redundancy. It limits the advantage of larger decoding gain (decoder input BER-to-output BER gain) in lower rate codes, so a reduction in code rate below 1/8 will eventually have worse decoded performance than that of rate 1/2 coding.

Based on these results, rate 1/3 convolutional coding was selected because it provides a good trade-off between performance, bandwidth, and memory utilization requirements. Rate 1/3 coding has 50 percent larger bandwidth and memory utilization requirements than rate 1/2 coding but it has at least 2 dB better code performance. The lower code rate gives it a better burst error correction capability because it has a larger code distance that rate 1/2 coding (rate 1/2 has a 10-bit free distance and rate 1/3 has a 15-bit free-distance [Proakis]). Codes with a lower code rate than 1/3 have larger burst error correction capabilities; however, they also have significantly larger bandwidth and interleaving memory requirements.

Rate 1/2 convolutional coding was selected in addition to rate 1/3 coding because it has better interleaving memory utilization. Rate 1/2 coding also provided

a reference by which other codes were compared to assess their interleaving memory and delay efficiency. Rate 1/2 coding was used as a reference because it is a common code selection for satellite communication link applications.

Convolutional constraint lengths were assessed next from the results found in [Bogusch, page 267] for rate 1/2, K=7 and K=9 coding with Viterbi decoding and soft quantization. These results indicate that constraint length 9 coding produces only a small performance improvement over constraint length 7 coding.

Based on these results and a consideration of code complexity requirements, constraint length 7 convolutional coding was selected. Constraint length 7 coding has approximately 4 times less decoding complexity than constraint length 9 coding. Constraint 9 coding has a larger burst error correction capability than constraint length 7 coding; however, this capability is small because constraint length 9 coding only increases code free-distance by 2 symbols (from 10 for K=7 to 12 for K=9 [Proakis]).

The quantization that was chosen for the convolutional code selections was 3-bit, erasure, and hard quantization. 3-bit soft quantization was selected for the best code performance. Erasure quantization was selected for better memory utilization but with somewhat degraded code performance[§]. Hard quantization was selected for the most memory efficient utilization with worst code performance. For a description of these techniques see Section 3.1.

Another option in the convolutional code selections was the number of bits per code symbol. The previous code selections have binary code symbols but other popular convolutional code selections are the multi-bit dual-k and triple-k convolutional codes. These codes have k bits per symbol and binary constraint lengths of $2k$ and $3k$ respectively. These codes also have a greater burst error correction capability than binary convolutional codes.

Time did not permit consideration of these codes. Their evaluation would have required separate simulation development which was spent instead on developing Reed-Solomon code routines. Reed-Solomon codes were chosen instead of non-binary

[§]Erasure decisions are neutral demodulation decisions determined for demodulation values that fall below a specified threshold. They are intended to prevent incorrect demodulation decisions that would otherwise incorrectly bias decoding decisions. In fading conditions, erasure-decision decoding performance can be quite good because unreliable demodulation decisions of faded signal values are removed. For the best performance, erasure thresholds are selected to provide the best balance between removing unreliable demodulated decisions and providing good signal representation.

convolutional codes because they were deemed better selections for same applications that are suggested for non-binary convolutional codes[¶].

2.3 REED-SOLOMON CODES.

Reed-Solomon codes were chosen because they have powerful random and burst error correction properties[*]. Reed-Solomon codes have been shown to have good code performance with AWGN channel conditions but their burst error correction capabilities also make them good candidates for fading channel conditions with small values of T_{il}/τ_0 (smaller than 100).

Reed-Solomon code parameters were selected by assessing code performance results which were derived during this stage of the investigation. The results were derived used a method of analysis and simulation which is described in Appendix A. It included DPSK modulation and slow Rayleigh fading conditions with a large T_{il}/τ_0 value. To account for large T_{il}/τ_0, the burst error correction capability of codes was emphasized to account for small T_{il}/τ_0 conditions.

Reed-Solomon code rates were assessed from results with various block-size codes. 31-symbol block sizes were chosen for most of the code comparisons because this block size is a practical one which is used in the Joint Tactical Information Distribution System (JTIDS). Other block sizes up to 256 symbols were also assessed. In all of these assessments and in the evaluations that followed, $\log_2(N+1)$ bits were assigned per symbol, where N denotes the block size in symbols; this is a conventional code assignment.

The code performance of Reed-Solomon 31-block size coding is shown in Figure 1. In these results, the (N, K) code variables signify the encoded symbol block size, N, and the pre-encoded symbol block size K; the codes have code rates of K/N. The results indicate that Reed-Solomon (31,15) coding has the best performance at 10^{-5} BER but the results also indicate that the Reed-Solomon (31,7) coding has close to the same performance. The results indicate that the higher rate 3/4 and 7/8 codes have much worse performance.

[¶]Several code sources suggested that Reed-Solomon coding was a better outer code choice for concatenated coding (see [Odenwalder, Forney])

[*]Reed-Solomon codes have the largest possible minimum distances between code words of any linear codes with the same encoder input and output block lengths. For a description of Reed-Solomon codes see [Odenwalder].

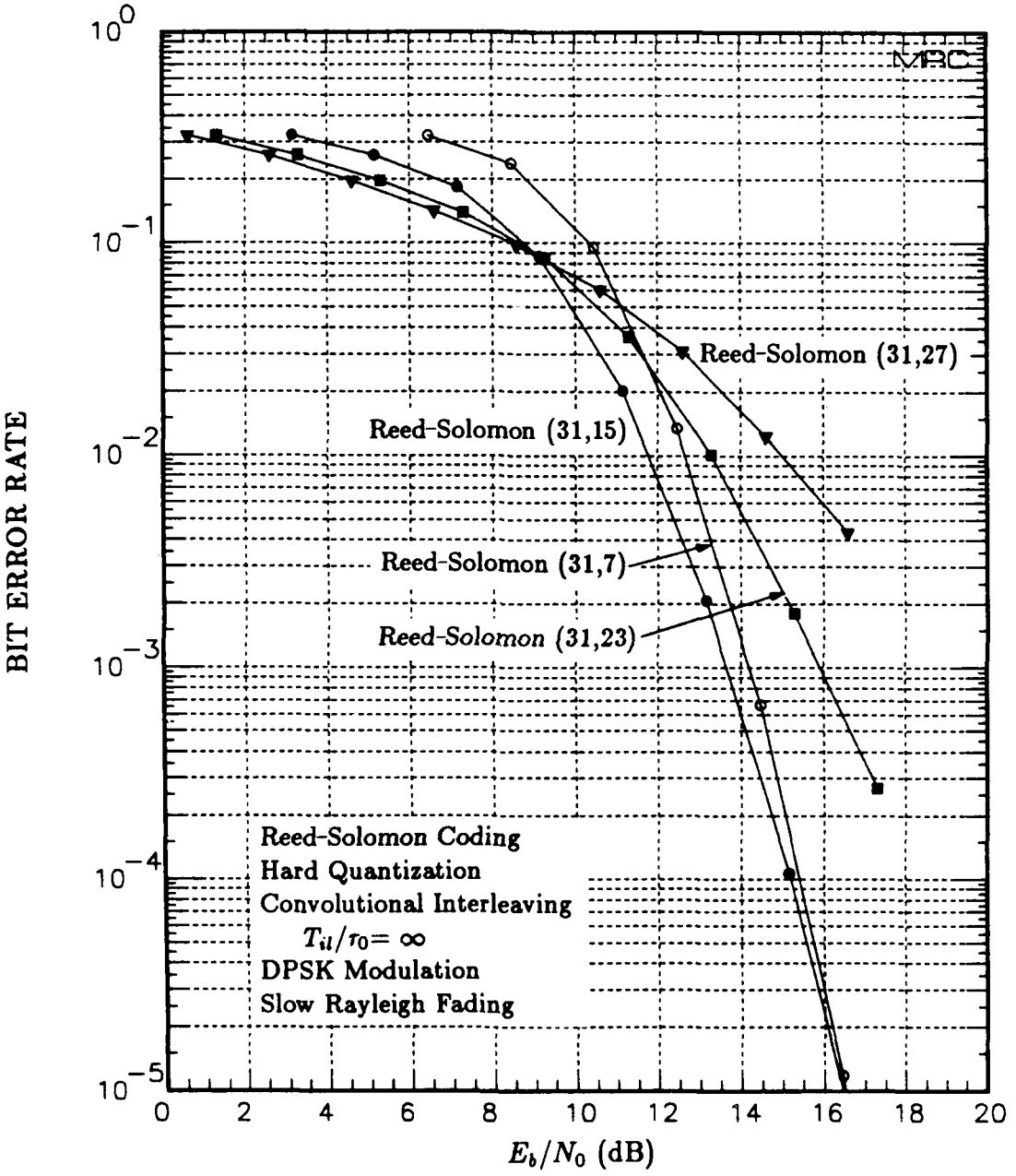


Figure 1. Code performance of Reed-Solomon (31,K) coding with Rayleigh fading conditions.

As in the convolutional coding results, the Reed-Solomon (31,K) code results indicate that code performance improves and then degrades with code rate reduction. The Reed-Solomon rate 1/4 coding performance curve has greater steepness than that of the higher rate codes because of larger coding gain. However, it also has larger E_b/N_0 requirements than the other codes at high BER because of a higher decoder input BER. Reed-Solomon rate 3/4 and 7/8 codes have inverted performance characteristics because of weaker coding gain.

Based on these results and the code rate results for larger codes, rate 1/2 Reed-Solomon coding was selected. Rate 1/4 coding has better burst error correction capability than rate 1/2 coding but rate 1/2 coding has smaller E_b/N_0 requirements, particularly for larger block sizes. Large block size coding performance influenced the code selection because large and small code selections of the same code rate were needed to assess block-size comparisons in the evaluated results.

Rate 3/4 Reed-Solomon coding was selected in addition to rate 1/2 coding, in part, because of it has better interleaving memory utilization. It was also selected because it was found to be a good choice as an outer concatenated code selection; this is addressed in the next subsection.

For a comparison of the best code rate selections in fading and benign conditions, Reed-Solomon (31,K) performance was also derived with AWGN channel conditions. These results, shown in Figure 2, indicate that rate 3/4 coding has the best performance at 10^{-5} BER. Rate 1/2 and rate 7/8 code performance is worse and rate 1/4 code performance is much worse. Compared with the results in Figure 1, these results indicate that rate 3/4 coding is a better choice in benign conditions and rate 1/2 or 1/4 coding are better choices in fading conditions. This indicates that coding gain is more important in the stressed environment of fading conditions than in benign conditions.

Reed-Solomon block sizes were selected next by comparing code performance for rate 1/2 codes with block sizes that ranged from 31 to 256. These results, shown in Figure 3, indicate that increases in block size improves code performance at 10^{-5} BER but this improvement grows more gradually with each increase in block size. The results also indicate that at higher BER, the code performance of the larger codes is worse than that of the smaller codes.

Code performance improves at 10^{-5} BER with increases in code size because larger codes have larger coding gain. This is seen by the steepness of the performance curves in Figure 3. Code performance grows worse at high BER with increases in code size because the larger codes have larger code size and, consequently, a higher symbol

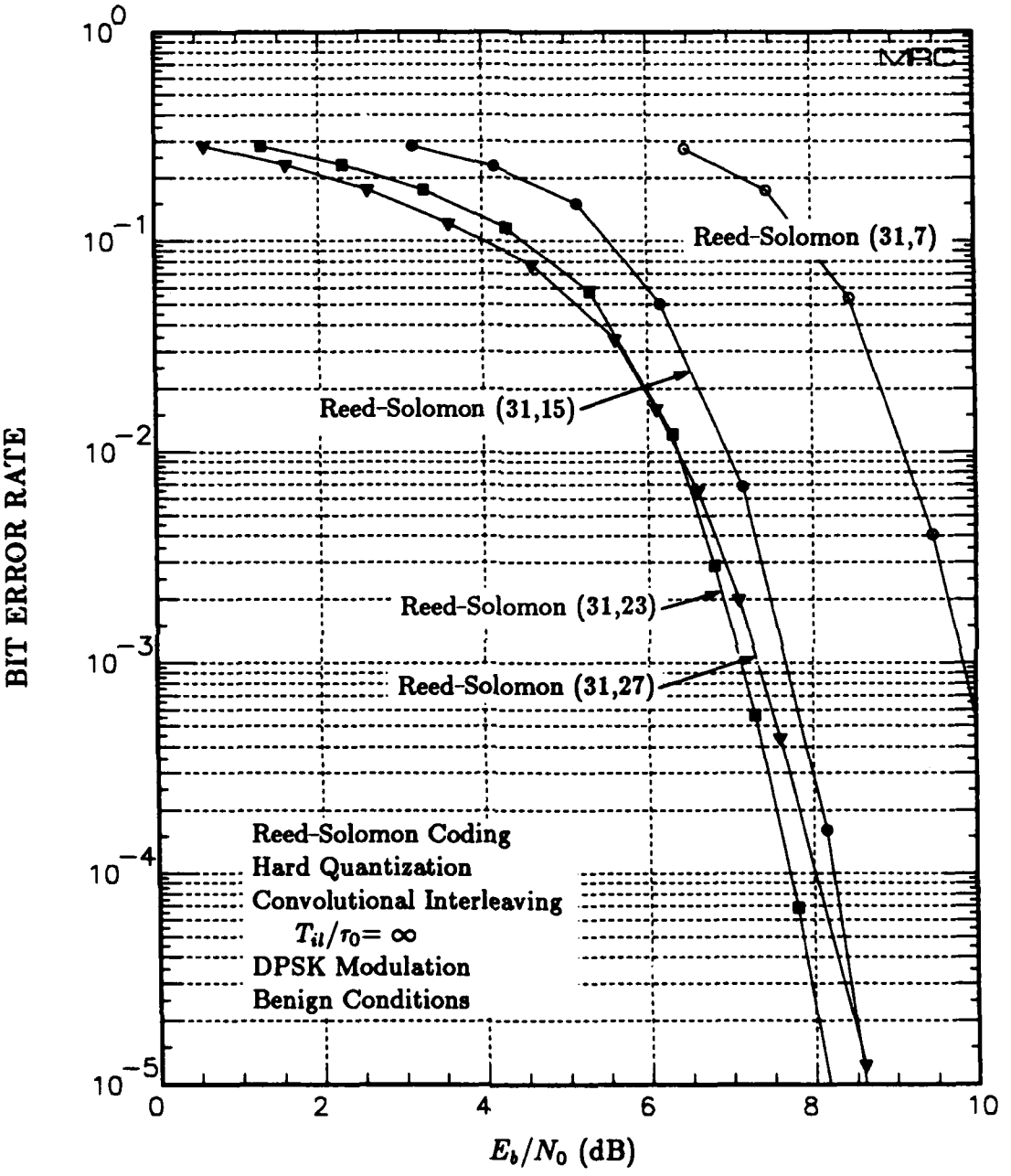


Figure 2. Code performance of Reed-Solomon (31,K) coding with benign conditions.

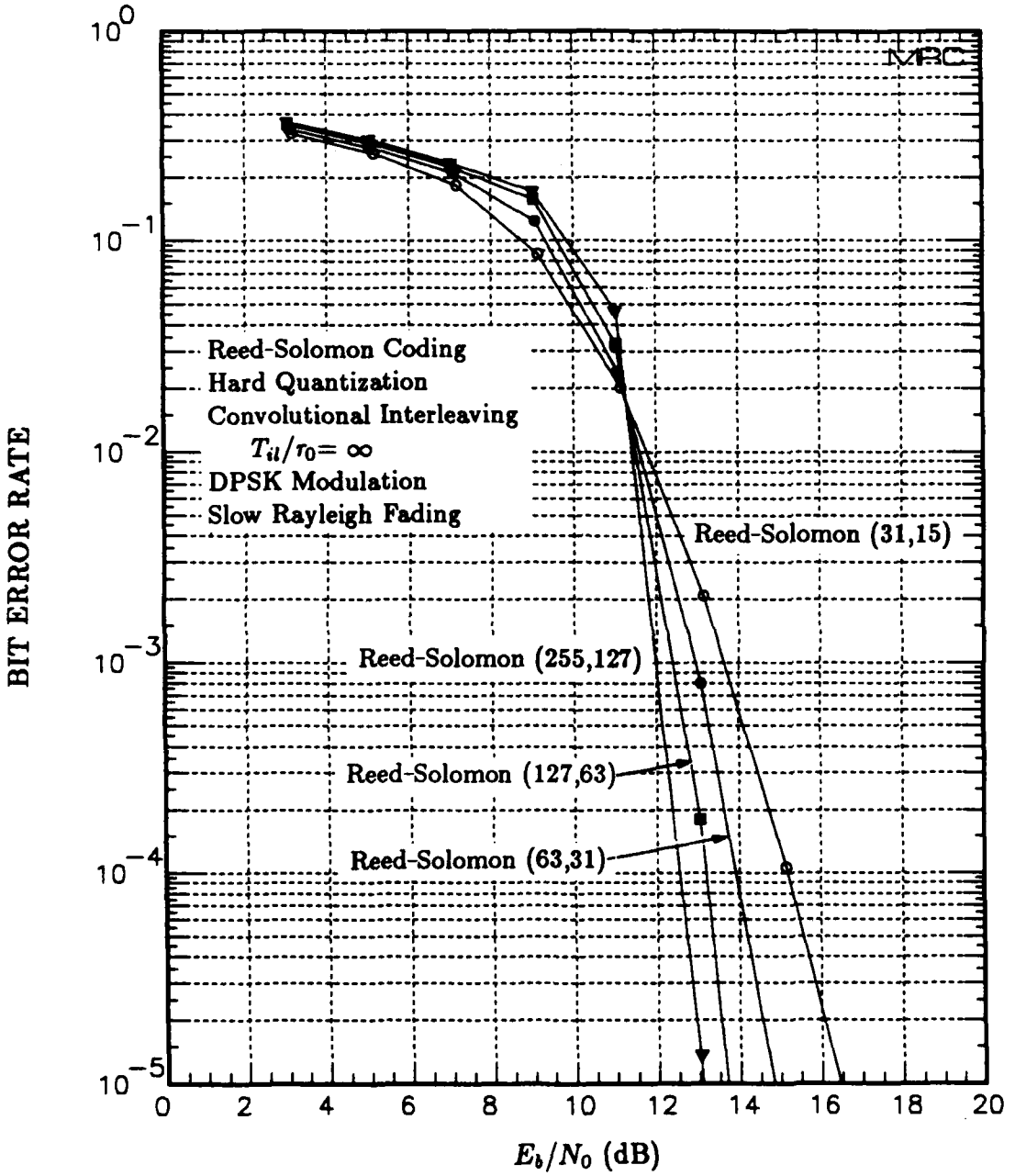


Figure 3. Code performance of Reed-Solomon rate 1/2 coding.

error probability. Larger symbols have a higher symbol error probability because they have a higher probability of encountering a binary demodulated symbol error[†].

Based on these results, Reed-Solomon 31 and 256 block sizes were selected. A block size of 31 was selected because it is found in practical code selections. A block size of 256 was selected to determine the interleaving memory reduction from increases in block size. Larger block size codes have larger burst error correction capabilities so 256-block size codes were expected to have relatively better performance than 31-block size codes for smaller values of T_{il}/τ_0.

Reed-Solomon codes with block sizes smaller than 31 were also considered but such codes were excluded from selection because their code rate requirements for useful code performance are too low for this investigation. Code selections such as Reed-Solomon (16,2) or (14,2) codes have code rates that are much smaller than 1/4 so they have large bandwidth expansions that make them vulnerable to frequency selectivity in high data rate applications. These codes also have relatively inefficient interleaving memory utilization[‡].

The quantization that was chosen for the Reed-Solomon code selections was erasure and hard quantization. Erasure quantization was selected to favor code performance and hard quantization was selected to favor interleaving memory utilization. Multi-bit soft quantization was not chosen because it would have imposed large decoding implementation requirements on the Reed-Solomon code selections.

2.4 CONCATENATED CODES.

Concatenated codes were selected by first identifying good concatenated code designs for fading channel conditions. This involved reviewing code sources to identify concatenated codes with good code performance characteristics in AWGN conditions[§]. These codes and variations of these codes were then assessed in code performance derivations to determine which codes are good choices for fading conditions. The code performance derivations for these assessments were conducted with the same approach adopted in the previous Reed-Solomon code performance assessments (see

[†]With signal fading, the symbol error probability is largest when the demodulated symbols are signal-independent. It is smallest when the demodulated symbols are correlated. For this reason, interleaving of code symbols and not binary demodulated symbols was conducted for all link evaluations. This maintained the correlation between demodulated symbol in slow fading conditions and, thereby, minimised code symbol error probability.

[‡]However, as discussed previously, lower rate codes may be quite useful in lower data rate links.

[§]Concatenated code evaluations with fading channel conditions were not found in code sources.

Appendix A). As such, the derived results had the assumption of large T_{il}/τ_0. This limited the accuracy of these assessments for smaller T_{il}/τ_0 conditions.

In the next step, the inner and outer codes which were identified as good choices were evaluated individually with a high-fidelity link program to determine their performance at small values of T_{il}/τ_0. The results of these evaluations were used to estimate the interleaving memory requirements for the concatenated code selections to determine with a reasonable degree of accuracy which codes have the lowest interleaving memory requirements. The codes with the lowest estimated memory requirements were then evaluated (in concatenation) by the link program to accurately determine their interleaving requirements. Estimation of these codes was required to limit the number of concatenated codes which that were selected for code evaluation.

The codes that were initially identified from the code sources were outer Reed-Solomon codes with high code rates and inner convolutional or Reed-Solomon codes with lower code rates [Odenwalder, Forney]. These codes (and variations thereof) were then assessed to determine good code rate selections. Outer Reed-Solomon coding and inner convolutional coding with DPSK modulation and slow Rayleigh fading were chosen for these assessments.

The results of outer Reed-Solomon (31,K) coding with inner convolutional rate 1/2 coding are shown in Figure 4. These results indicate that rate 3/4 and 7/8 outer Reed-Solomon codes have better performance than lower rate 1/2 and 1/4 outer Reed-Solomon codes.

Based on these and other results, outer rate 3/4 Reed-Solomon coding and inner convolutional rate 1/2 and 1/3 and Reed-Solomon rate 1/2 coding were chosen for the next stage of memory estimation. These code choices were also made because they corresponded to single-stage Reed-Solomon code selections. As such, their evaluations were used to derive concatenated memory estimates and to determine the memory requirements for single-stage coding. The estimates for these codes followed the single-stage code evaluations so they are addressed after the single-stage results in Section 5.4.

Besides the concatenated code selections, interleaving for concatenated coding was assessed to determine the best allocation of inner and outer interleaving resources for minimizing combined memory requirements. This required assessing the effects of interleaving distributions on T_{il}/τ_0 code performance and interleaving memory utilization. From these assessments, large inner interleaving with much smaller outer interleaving was deemed the most memory and delay efficient choice.

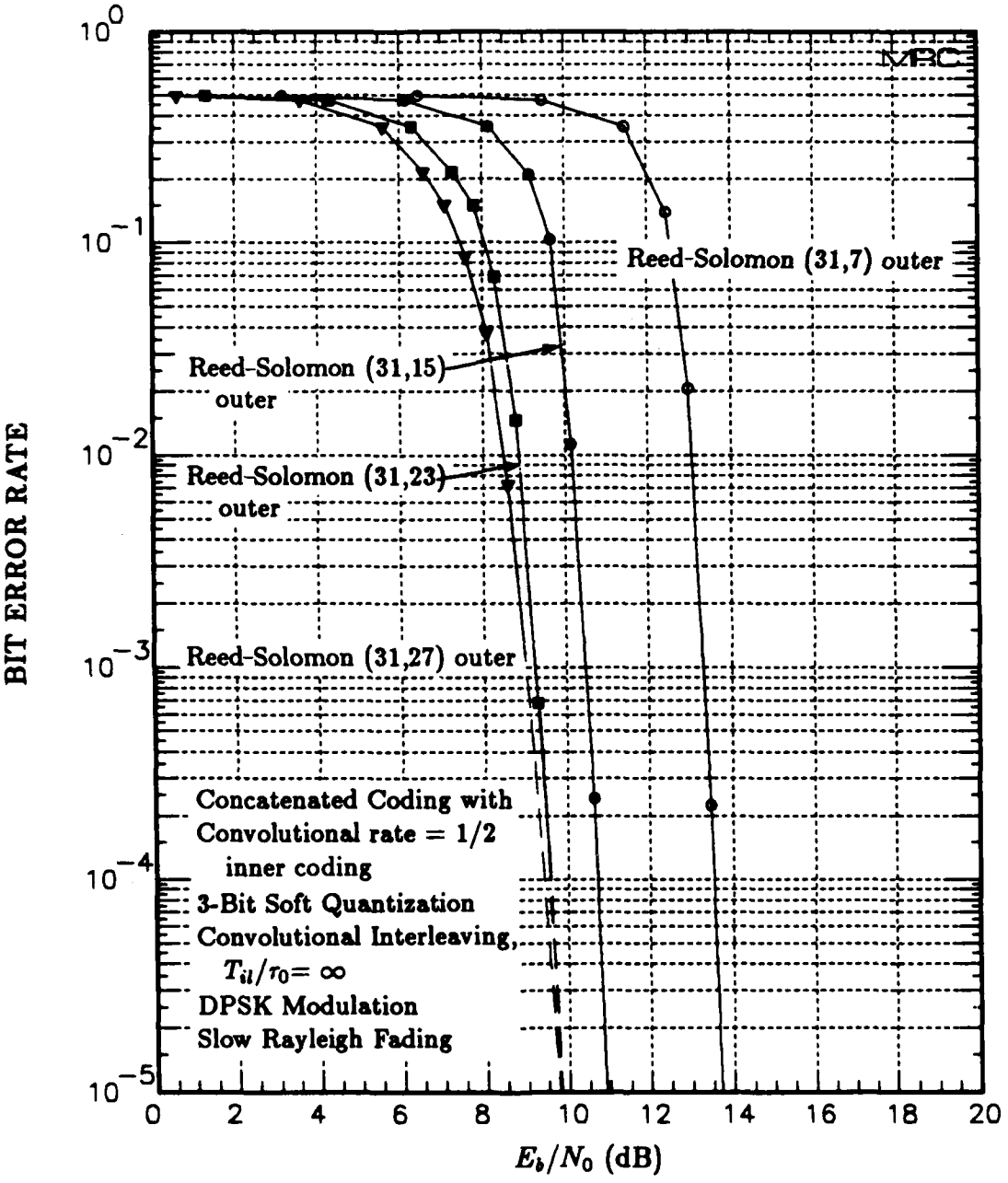


Figure 4. Code performance of concatenated Reed-Solomon (31,K) outer coding and convolutional rate 1/2 inner coding.

Large inner interleaving was selected because it produces the best code performance; it provides diversity to both inner and outer codes[¶]. Small outer interleaving was selected because it improves outer code performance by dispersing naturally bursty errors from inner decoding. Since the inner interleaving span was chosen much larger than the outer interleaving span, only the inner interleaving requirements were included in the results.

Besides these selections, convolutional interleaving was selected for inner interleaving because it is relatively memory and delay efficient (it was also selected for single-stage coding). Block interleaving was selected for outer interleaving because it permitted freedom in selecting row and column configurations. Block interleaving is far less memory efficient than convolutional interleaving but this did not matter since it had negligible size in comparison with the inner interleaver.

The freedom in selecting block interleaving configurations was important because it permitted selecting useful interleaver configurations that would otherwise be suboptimal with convolutional interleaving. In particular, block interleaver configurations were selected with more rows than columns; this is discussed in Section 3.3. Comparable row and maximum register selections in a convolutional interleaver would have resulted in bursty distributions of symbol errors and degraded code performance.

[¶]Inner interleaving has worse memory utilization than outer interleaving because it includes inner code redundancy. However, this disadvantage is small in comparison with the advantage it has in minimizing T_{il}/τ_0 with better code performance.

SECTION 3

COMMUNICATION LINK EVALUATION PROGRAM

This section addresses the communication link program that was used to evaluate coding performance. It is organized in three subsections with the first subsection providing an overview of the evaluated link components. This is followed by a detailed description of the simplified Reed-Solomon code model. The last subsection describes how unspecified link parameters, such as data rate and interleaver configurations, were selected.

3.1 OVERVIEW OF EVALUATED LINK COMPONENTS.

The computer program used to evaluate code performance included convolutional and Reed-Solomon coding, convolutional and block interleaving, DPSK modulation, and slow, flat Rayleigh fading channel conditions. For concatenated code evaluations, it was configured as shown in the link layout of Figure 5. Outer coding and interleaving stages were omitted in single-stage code evaluations.

All of the link components were simulated with explicit models of actual system components except for the Reed-Solomon code components. Reed-Solomon coding was evaluated with simulated and simplified modeling. The Reed-Solomon simulation model was used for evaluating small Reed-Solomon (15,7) codes and the Reed-Solomon simplified model was used to evaluate larger codes, Reed-Solomon (31,23) and (31,15) or larger. The simplified model was used in these evaluations because it provided accurate and relatively fast link evaluations.

The accuracy of the simplified Reed-Solomon code model was determined by code performance comparisons with the simulated Reed-Solomon code model. This comparison and a description of the simplified code model are addressed in the next subsection. Discussions of the simulated Reed-Solomon code model can be found in [Feldman].

Convolutional coding was evaluated with binary rate 1/2 and 1/3, constraint length 7 code designs. It had Viterbi decoding and was configured for storing decoder input paths over 32 bits prior to making decoding decisions. A good description of convolutional coding and Viterbi decoding can be found in [Bogusch] and [Proakis].

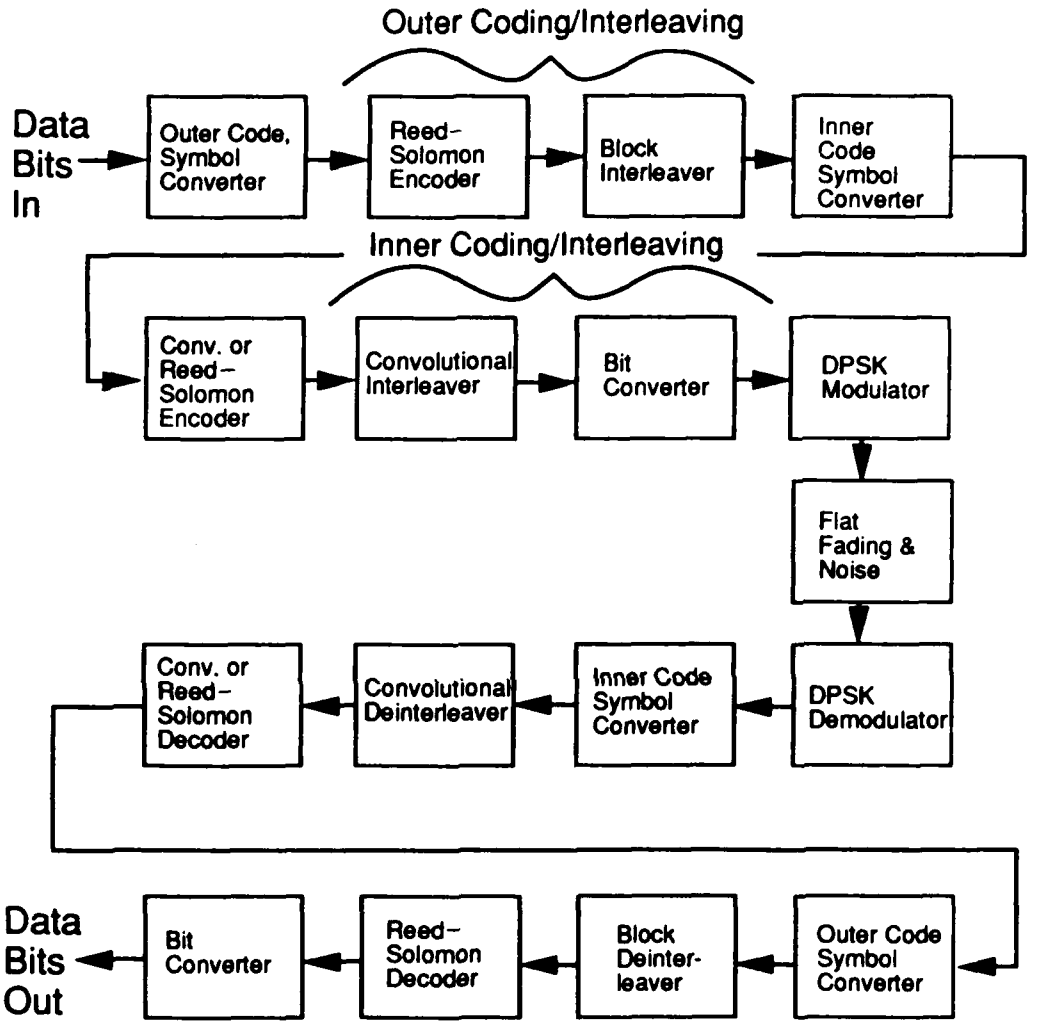


Figure 5. Link layout for the evaluation program with concatenated coding.

Convolutional and block interleaving were evaluated with code symbol processing. The convolutional interleaver processed single-stage and inner concatenated code symbols and the Block interleaving processed outer concatenated code symbols. The configurations selected for these interleavers are addressed in Section 3.3. A good description of convolutional and block interleaving can be found in [Bogusch].

Code symbol conversion accompanied Reed-Solomon coding to permit a bit-synchronized interface between the non-binary Reed-Solomon code symbols, binary DPSK modulation symbols, and binary convolutional symbols. It was evaluated using two conversion arrays with bit sizes equal to the least-common-multiple of the conversion input- and output-symbol bit sizes. It operated by writing symbols into one array while reading symbols out of a second array that contained the previous block of input symbols. Because of the array sizes, symbols filled one array in bit-synchronization with symbols being emptied from the other array. After each complete symbol input and output, the arrays were switched for the next block of input and output.

DPSK modulation and demodulation were evaluated with differential encoding and sampled noise and signal scintillation. The demodulator included inphase and quadrature dot-product signal processing. Frequency and time tracking were not included because tracking noise and potential loss of tracking-lock would only have clouded code performance assessments. A good description of DPSK modulation can be found in [Bogusch] and [Proakis].

The sampled-signal model in the DPSK modem was evaluated with slow, flat fading channel conditions. Inphase and quadrature signal fading samples were computed as the double-pole filtered output of independent white-Gaussian noise samples. Decorrelation time, τ_0, was defined at the time-offset at which correlated power is a 1/e fraction of peak power. A good description of the channel model can be found in [Bogusch].

Quantization was evaluated following DPSK demodulation. It included either hard, erasure, or 3-bit soft quantization.

Hard quantization assigned two quantization values. One value was assigned for demodulation values that was larger than zero and another quantization value to demodulation values that were smaller than zero (for maximum-likelihood decisions). Logic 0 was assigned for a positive demodulation value and a logic 1 was assigned for a negative demodulation value.

Erasure quantization was conducted with similar logic 0 and 1 assignments; however, it also included erasure assignments for demodulation values that fell within

the absolute value of an erasure threshold. Erasure quantization decisions were made according to the following algorithm

$$\begin{array}{lcll} d & > & T_e\,, & l = 1 \\ -T_e & \leq & d \leq T_e\,, & l = E \\ d & < & -T_e\,, & l = 0 \end{array} \qquad (1)$$

where d denotes the demodulation value as normalized by its average absolute value, T_e denotes the erasure threshold, and l denotes the quantization value. The erasure threshold selections are addressed in Section 5.2.

Three-bit soft quantization differed from hard quantization and erasure quantization by assigning more than two non-neutral quantization values. It assigned four quantization values for positive demodulation values and four quantization values for negative demodulation values according to the following algorithm

$$l = [2^Q \left(\frac{d + 1.0}{2.0}\right) F_Q]\,, \quad 0 \leq l \leq 2^Q - 1\ . \qquad (2)$$

Here l denotes the quantization value, Q denotes the number of quantization bits per quantization value (Q equals three for 3-bit quantization), d denotes the normalized demodulation value, and F_Q denotes the quantization scale factor. The quantization scale factor selection is addressed in Section 5.2.

The square brackets of this algorithm denote an integer operation. Consequently, values of d less than -1.0 or greater than 1.0 were clipped at the smallest and largest possible values of l respectively. Quantization values between 0 and $2^Q/2 - 1$ corresponded to logic 0 decisions, with the greatest weighting assigned to 0. Quantization values between $2^Q/2$ and $2^Q - 1$ corresponded to logic 1 decisions, with the greatest weighting assigned to $2^Q - 1$.

In link with Reed-Solomon coding and erasure quantization, Reed-Solomon symbol conversion of erasure quantization symbols was conducted differently than in the standard symbol conversion. In these cases, Reed-Solomon symbols were assigned an erasure for any occurrence of an erasure in the symbols being converted to a code symbol.

3.2 SIMPLIFIED REED-SOLOMON MODELING.

The simplified model for Reed-Solomon coding was developed with similar coding assumptions used in analytical derivations of approximate Reed-Solomon code performance (see [Odenwalder, Hauptschein] or Appendix A). However, unlike the

analytical derivations, the simplified code model was used to accurately evaluate any fading conditions. This is because it was part of the Monte-Carlo communication link program.

The simplified encoder was evaluated with two arrays. The first array receives a block of pre-encoded data and the second array writes out the received data with an appended, fake code parity. The simplified decoder compares the number of errors and erasures it receives in a block of encoded symbols and makes the decision to correct or not to correct the symbols.

The simplified decoder was configured as an inverse of the simplified encoder. It receives encoded symbols in an input array while writing out decoded symbols from a second array. Input and output symbols are bit-synchronized so that encoded symbols fill one block array at the same bit-period that decoded symbols are fully emptied from the other block array. Upon receiving a complete block of encoded symbols and emptying a complete block of decoded symbols, the decoder decides whether to correct or to pass the data portion of its received symbols.

The simplified decoder makes the decision to correct or pass encoded symbols by comparing the number of errors and erasures it receives using the same algorithm found in actual Reed-Solomon decoders. This algorithm is

$$\begin{array}{lcl} If\ 2n+m<D & \Rightarrow & Correct\ Symbols \\ Otherwise & \Rightarrow & Pass\ Symbols\ Uncorrected \end{array} \tag{3}$$

where n and m denote the number of received symbol errors and erasures, respectively, within a block of encoded data. D denotes the minimum code distance which equals the number of symbols in a block of encoded symbols, N, plus one minus the number of symbols in a block of pre-encoded symbols, K. The simplified decoder computes the errors it receives by comparing received symbols with encoded symbols that are regenerated at decoding.

The accuracy of the simplified decoder depended on the similarity between its error count and the estimated error count of an actual Reed-Solomon decoder. The error counts differ when an actual decoder makes an incorrect block symbol correction; this is denoted as a misdecode. When a misdecode occurs, an actual decoder produces more errors on average than would occur in the simplified decoder. This is because an actual decoder tends to produce more symbol errors from a misdecode than the simplified decoder produces with the same block of received symbols*.

*The simplified decoder makes a decision to pass symbols uncorrected when a misdecode would have occurred in an actual decoder. It makes this decision because more than $D/2$ errors occur during a misdecode (see Equation 3).

The similarity between the simplified decoder and an actual decoder depends on the ratio of misdecodes to correct error decisions in an actual decoder. This probability has been determined in various code sources [Feldman, Jennings] and shown to decrease with decreases in code rate and increases in block size. From this relationship, the accuracy of the simplified code model could be assumed for any Reed-Solomon code with a smaller code rate or a larger block size than a code which was already proven to be accurately approximated.

With the objective of finding the smallest codes which could are accurately approximated by simplified code model, the BER performance of the simplified code model was compared with that of the simulated code model using Reed-Solomon (31,15) and (31,23) codes. These codes were selected because they have the smallest block size, 31, that was evaluated in analytical derivations of approximate code performance [Odenwalder, Hauptschein]. Slow fading channel conditions with small T_{il}/τ_0 were included in the accuracy evaluations to account for the same conditions planned for the code performance evaluations.

The results of the Reed-Solomon (31,15) code evaluations are shown in Figure 6. These results indicate that simplified modeling performance is close to that of simulated modeling performance but a performance deviation occurs at high BER with T_{il}/τ_0 of 10. A comparable performance deviation was not found between comparisons of simplified and simulated performance comparisons with benign channel conditions. Consequently, the performance derivation appears to be the result of excessively bursty errors at high BER.

These results indicate that Reed-Solomon (31,15) coding can be accurately evaluated with the simplified coding model over a wide range of BER and T_{il}/τ_0. In particular, Reed-Solomon (31,15) coding can be evaluated with virtually no inaccuracy at BER of 10^{-5}; this was the BER needed for single-stage and outer concatenated coding. For higher BER near 10^{-2}, needed for inner concatenated coding, Reed-Solomon (31,15) can also be evaluated relatively accurately. In this case, the performance deviation between simplified and simulated performance is about 1/4 dB.

Based on this assessment, the simplified code model was chosen to evaluate Reed-Solomon (31,15) coding with single-stage and concatenated code configurations. Because lower rate, larger codes are ensured to be more accurately approximated by the simplified code model, the simplified model was also used to evaluate Reed-Solomon (255,127) and (255,191) codes.

The results of Reed-Solomon (31,23) code comparisons are shown next in Figure 7. These results indicate that, as expected for a higher rate code, Reed-

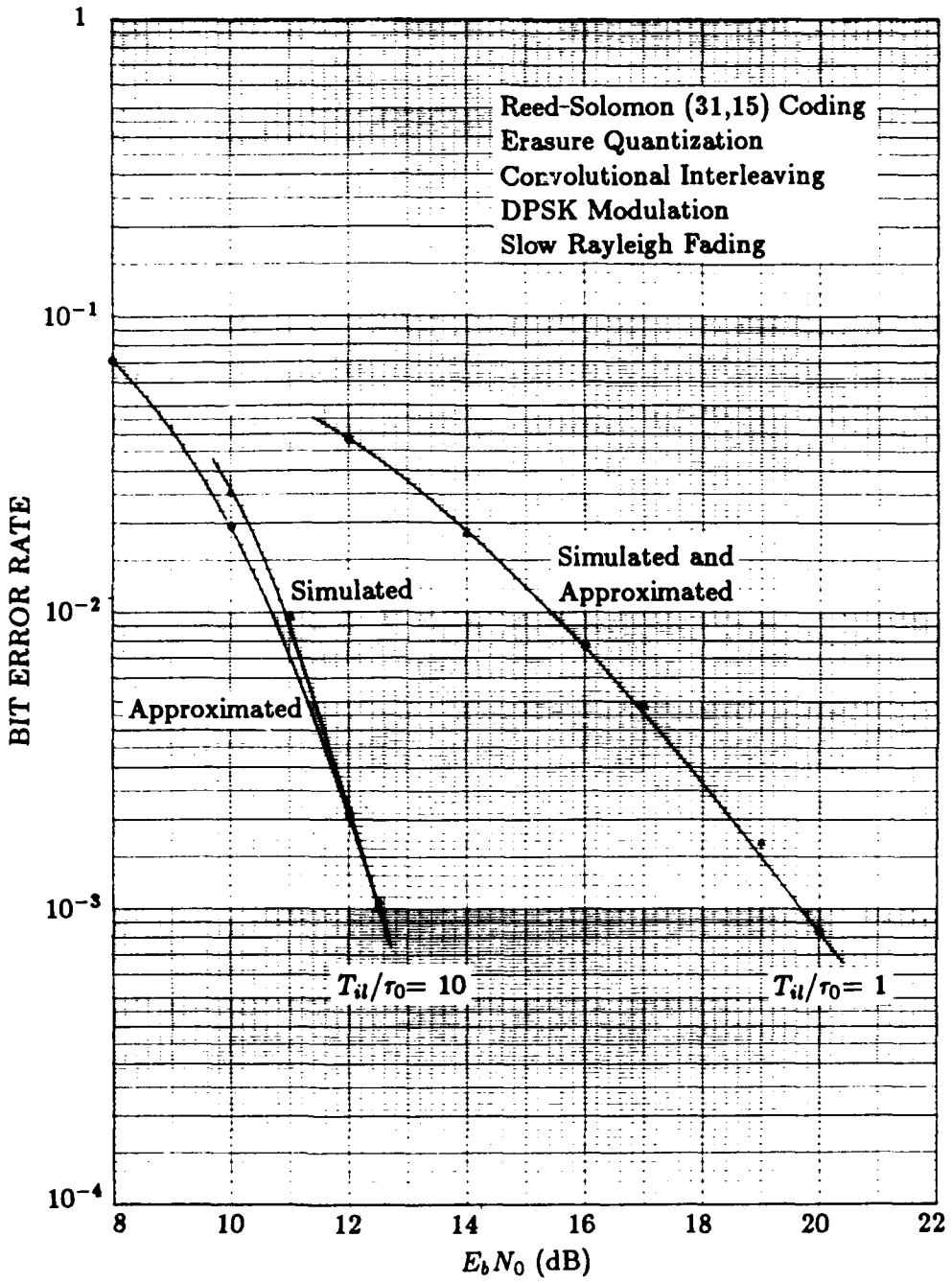


Figure 6. Evaluation of simplified modeling with Reed-Solomon (31,15) coding.

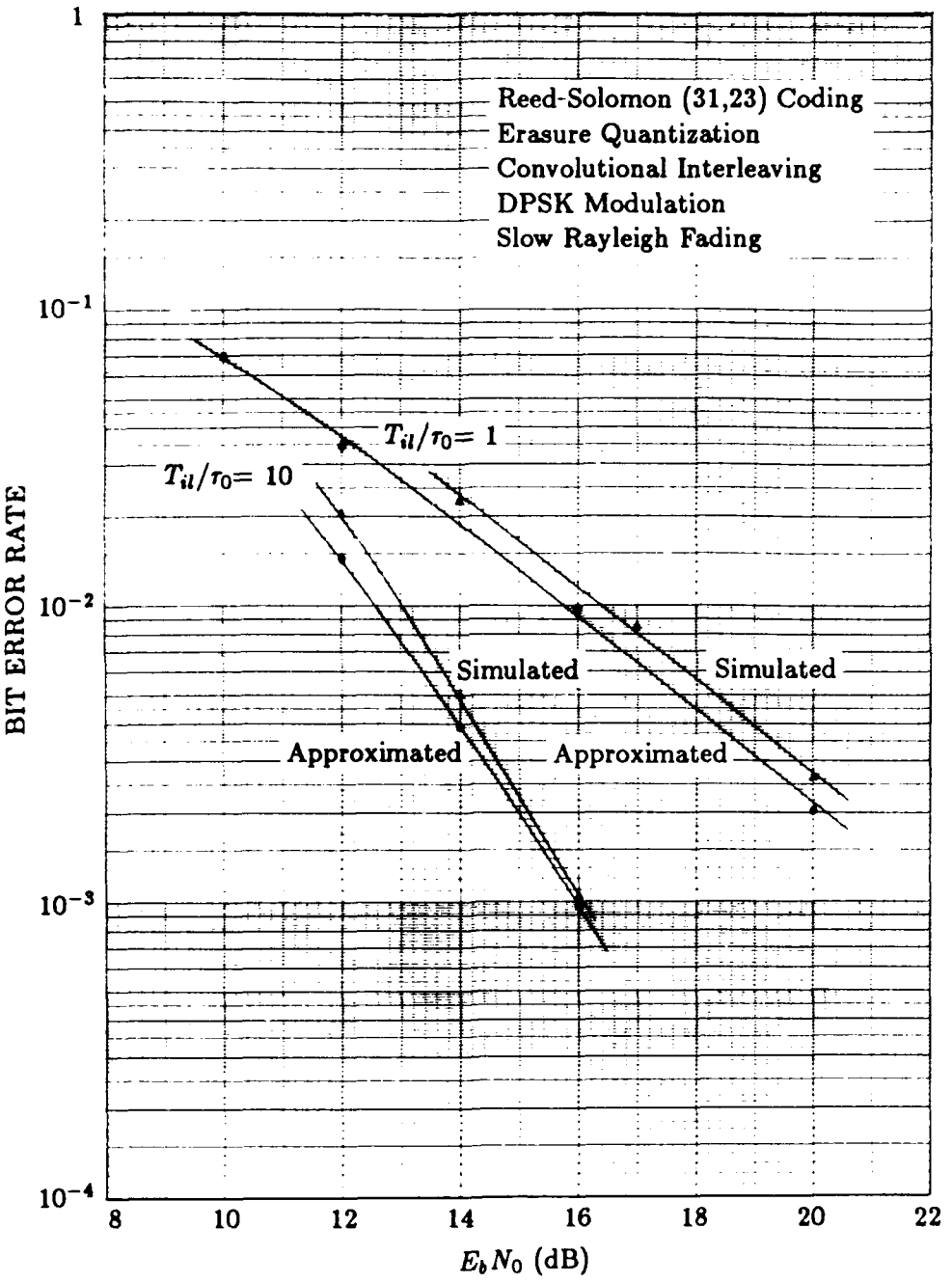


Figure 7. Evaluation of simplified modeling with Reed-Solomon (31,23) coding.

Solomon (31,23) is less accurately approximated than Reed-Solomon (31,15) coding. Reed-Solomon (31,23) coding is less accurate at high BER with T_{il}/τ_0 of 10 and over the entire range of BER for T_{il}/τ_0 of 1. In the latter case, the performance deviation is approximately 1/2 dB.

These results suggest that Reed-Solomon (31,23) coding can be evaluated accurately with the simplified code model but only over a restricted range of BER and T_{il}/τ_0. Specifically, Reed-Solomon (31,23) code performance can be accurately evaluated with the simplified model at low BER (around 10^{-5}) with the exception of when T_{il}/τ_0 is close to one. Based on this assessment, the simplified code model was chosen to evaluate Reed-Solomon (31,23) coding in single-stage or outer concatenated code configurations and with T_{il}/τ_0 larger than one[†].

3.3 SELECTION OF UNSPECIFIED LINK PARAMETERS.

Following the initial coding, interleaving, and modulation selections, several communication link parameters remained unspecified prior to the link evaluations. These parameters were data rate, interleaver configurations, τ_0, E_b/N_0, and the number of bits needed for the link evaluations.

E_b/N_0 was specified in anticipation of the expected BER and τ_0 was selected as one second. However, the remaining parameters were selected to ensure several requirements.

The data rate was selected to accurately reflect a high data rate links with slow fading channel conditions. The interleaver configurations were selected for the best code performance. The number of bits processed in link evaluations were selected to provide accurate BER statistics and to limit the length and, consequently, the time required for link evaluation. The data rate and interleaver symbol sizes were also limited because they influenced the number of bits required for link evaluation.

Interleaver configurations were selected first because they influenced the data rate and the number bits required for link evaluations. For single-stage coding, convolutional interleaving row and maximum register sizes were selected for the best code performance according to interleaver evaluations that are addressed in Section 5.1. These selections depended on the type of code and the T_{il}/τ_0 selection.

[†]Reed-Solomon (31,23) coding was also evaluated by the simplified code model with T_{il}/τ_0 of 1 and with inner Reed-Solomon (31,23) concatenated coding. However, these evaluations proved that Reed-Solomon (31,23) was a bad code choice for small T_{il}/τ_0 conditions and inner coding.

With convolutional coding, row sizes were selected anywhere between 10 and 50. With Reed-Solomon coding, row sizes were selected equal to the number of symbols in the code block. The maximum register sizes were then selected to satisfy the row and T_{il}/τ_0 selections. These selections are defined by Equation 8 of Section 5.1.

For concatenated coding, outer block interleaver configurations were selected for the best outer code performance. These configurations were selected with their number of rows equal to the number of symbols in the outer Reed-Solomon code block. The number of interleaver columns was selected as ten for an interleaver span which was ten times that of the outer code block size. This ensured adequate diversity to correct a block of inner decoded errors.

Inner convolutional interleaving configurations were subsequently selected for the best inner code performance and for a time-span that was ten times larger than that of the block interleaver. The convolutional interleaving span was selected to be much larger than the block interleaving span to ensure that convolutional interleaving provided the only appreciable signal fading diversity. This was required for the lowest inner and outer interleaving requirements (see Section 2.4).

The data rate (user bit rate) was computed next because it has to satisfy interleaver configuration and T_{il}/τ_0 interleaving span selections. It was computed from the interleaving requirements by dividing the interleaver span in user bits by the interleaver span in seconds. The equation for this computation is

$$R_d = \frac{n_{row} n_{reg} nr}{T_{il}} \quad . \tag{4}$$

Here n_{row} denotes the number of interleaver rows, n_{reg} denotes the symbol size of the largest interleaver register, n denotes the number of bits per code symbol (single-stage or inner code), r denotes the code rate (single-stage or concatenated), and T_{il} denotes the interleaver span in seconds. Since τ_0 was chosen as one second, T_{il}/τ_0 was substituted for T_{il}.

In addition to satisfying interleaver selections, the data rate also had to satisfy requirements for accurately modeling high data rate, slow fading link conditions. To satisfy these requirements, a general data rate requirement was established and used to determine whether the data rate computed in Equation 4 was adequate for accurately modeling high data rate conditions.

The general data rate requirements was determined by conducting link evaluations with different data rates to determine the data rate-to-τ_0 ratio at which code performance deviates from a constant high data rate value. Below this value, performance is altered by factors such as fast fading signal variation over symbol periods

and added time-diversity from small code time-span-to-τ_0 ratios. The results of these evaluations indicated that modulated time periods must be smaller than about one thirtieth of τ_0 to accurately model high data rate link conditions. These results agree with a similar study of DPSK modulation in [Bogusch, page 119].

Based on these results, the data rate required for ensuring accurate evaluation of high data link conditions was defined as

$$R_d \geq \frac{30r}{\tau_0} \tag{5}$$

where R_d denotes the data rate and r denotes the code rate (single-stage or concatenated). In the comparisons of this data rate requirement with that of Equation 4, it turns out that Equation 5 was satisfied for all link selections.

The number of user bits required for the link evaluations was chosen last because this value is a function of the data rate. For accurate, minimum length link evaluations, the number of bits were selected to accumulate approximately 500 or more decoded errors and to ensure that the evaluations spanned 500 or more τ_0. This value was computed as

$$N = \min\left(\frac{500}{BER}\ ,\ 500R_d\tau_0\right) \tag{6}$$

where BER denotes estimated BER and R_d denotes the data rate in bits per second. It turns out that, for most of the link evaluations, 500 τ_0 spans was the dominant requirement in this equation because the data rate was typically very large.

SECTION 4

DEFINITIONS OF INTERLEAVER MEMORY AND DELAY

Following the code performance evaluations interleaving memory requirements were converted from T_{il}/τ_0, code rate, and quantization parameters and interleaving delay requirements were converted from T_{il}/τ_0 selections.

This section addresses the formulas that were used to translate T_{il}/τ_0 and code parameters into convolutional interleaving requirements. It is organized in three subsections with the first subsection providing a formulation of the interleaving memory conversion. The second subsection addresses the interleaving memory savings that were included from combining symbols with erasure quantization. The third subsection defines the interleaving delay.

4.1 INTERLEAVING MEMORY.

Convolutional interleaving memory requirements were defined as 1/2 the product of the interleaving span, in seconds, and the sum of the interleaver and deinterleaver bit rates. This definition, given in Equation 7, includes separate bit rates for interleaving and deinterleaving to account for differences in interleaver and deinterleaver symbol sizes.

$$M = \frac{1}{2} T_{il}(R_{il} + R_{dil}) \ \ bits \tag{7}$$

where T_{il} denotes the interleaving span, R_{il} denotes the interleaver bit rate and R_{dil} denotes the deinterleaver bit rate.

Equation 7 is not a convenient formulation of interleaving memory requirements, however. Instead, memory requirements are better defined directly in terms of the user bit rate. For this, memory requirements were reformulated by first defining R_{il} and R_{dil} in terms of the user bit rate.

R_{il} is proportional to the user bit rate by the inverse code rate. It equals

$$R_{il} = R_d \frac{1}{r} \tag{8}$$

where R_d denotes the user bit rate and r denotes the code rate (single-stage or concatenated).

R_{dil} is related to the user bit rate by the inverse code rate and the quantization. It equals

$$R_{dil} = R_d \frac{1}{r} \frac{n_Q}{n} \tag{9}$$

where n_Q denotes the number of bits in a pre-decoded symbol (quantized) and n denotes the number of bits in an encoded symbol.

Incorporating the data rate definitions of Equations 8 and 9 in the memory definition of Equation 7, interleaving memory was redefined as

$$M = \frac{T_{il} R_d}{2r} (1 + \frac{n_Q}{n}) \ \ bits \ . \tag{10}$$

Although Equation 10 conveniently relates the user data rate to memory requirements, it was further modified to define interleaving memory requirements in more general units of measurement than bits. Equation 10 was modified to define memory requirements in terms of τ_0 units for interleaving span compatibility with T_{il}/τ_0 selections in the results; this avoided unnecessary memory dependence on τ_0. Equation 10 was also modified to define memory requirements in terms of R_d units to remove unnecessary memory dependence on the specific data rate.

The resulting memory equation is

$$M = \frac{T_{il}}{\tau_0} \frac{1}{2r} (1 + \frac{n_Q}{n}) \ \ R_d \tau_0 \ units \ . \tag{11}$$

This equation was used to compute the memory requirements in the results.

For an example equal to how Equation 11 was used, consider convolutional rate 1/2 coding with Viterbi decoding and 3-bit soft quantization. This code has r equal to 1/2, n equal to 1 because it is a binary code, and n_Q equal to 3 because it has 3-bit quantization. If T_{il}/τ_0 of 10 is chosen, the convolutional code has a memory requirement equal to $10 \cdot (1 + 3) = 40 \ R_d \tau_0$ units.

Next, consider Reed-Solomon (31,15) coding with erasure quantization. This code has r equal to 15/31 and n equal to 5 because the Reed-Solomon codes were selected with $\log_2(N + 1)$ bits per symbol. n_Q for this code is equal to 6 because it has an extra bit for a symbol erasure. If T_{il}/τ_0 of 10 is also chosen, the Reed-Solomon code has a memory requirement equal to $10 \cdot \frac{31}{2 \cdot 15} \cdot (1 + \frac{6}{5}) = 22.7 \ R_d \tau_0$ units.

Besides illustrating memory computation, these examples indicate the benefit of more efficient memory utilization. Both codes in these examples have nearly the same code performance, with 16 dB to 17 dB E_b/N_0 requirements for 10^{-5} BER.

However, the Reed-Solomon code has much smaller memory requirements than the convolutional code because it uses only one 1 extra bit in quantizing 5-bit symbols. The convolutional code uses 2 extra bits in quantizing 1-bit symbols.

The interleaving memory requirements for all of the code rate and code quantization selections are tabulated in Appendix B for reference. These requirements are normalized as functions of T_{il}/τ_0 to allow convenient conversion to $R_d\tau_0$ units for any T_{il}/τ_0 selection.

4.2 DEINTERLEAVER MEMORY SAVINGS FROM COMBINING ERASURE QUANTIZED SYMBOLS.

In the last example of the previous subsection, erasure quantization was found to be relatively memory efficient since it only required one extra bit to represent an erased symbol value. This is not the most memory efficient way to represent an erased symbol, however. An erasure represents only one symbol value but a bit provides representation for two values.

An erasure-quantized symbol with n pre-quantized bits has 2^n possible logic values (0 and 1 values) and one possible erasure value. However, when represented by $n+1$ bits, the symbol has 2^{n+1} possible represented values. This is more than 1/3 the number of symbol values that are needed to represent the symbol.

A more efficient method of representing erasure symbols is to assign bits after combining the symbols. To illustrate this, consider a pre-quantized symbol with two bits. If after erasure quantization two of these symbols are combined then the total number of symbol values is 25. Five bits are required to represent these symbols but 6 bits would be required if the symbols were not combined. This translates to a 20 percent savings in deinterleaver memory.

To determine what memory savings from erasure-symbol combining to use for adjusting non-combined memory requirements, deinterleaving memory requirements for each code size selection with many symbol combinations were first computed. This required using a general formula for the memory requirements to expedite the memory computation. The equation used for these computations was

$$r_E = \frac{\left[\log_2(2^n+1)^N\right]_{round-up}}{N(n+1)} \tag{12}$$

where r_E denotes the ratio of deinterleaver memory requirements with erasure-symbol combining to the memory requirements without erasure-symbol combining. N denotes

the number of combined symbols and n denotes the number of bits per symbol prior to erasure quantization (or per encoded symbol).

Equation 12 was derived by taking the ratio of the number of bits required for symbol combining to the number of bits required without symbol combining. The bit requirement for combined symbols was determined as the rounded-up integer value of the base-2 log of the total number of possible symbol values. The total number of symbol possibilities was determined as the N'th power of $2^n + 1$ values per symbol.

Evaluations of Equation 12 indicated that deinterleaver memory requirements fluctuate but, on average, decrease with increases in N. The evaluations also indicated that deinterleaver memory requirements asymptotically approaches a minimum value as N becomes large. Unfortunately, large values of N impose a large computational burden on deinterleaving since each deinterleaver input or output requires adding a symbol to or extracting a symbol from a block of combined symbols. Fortunately, however, the evaluations indicated that memory reduction rapidly approaches a minimum value with increases in N. This meant that choosing relatively small values for N would still provide good a reduction in memory requirements.

Based on these evaluations, ten erasure combined symbols were selected for each code size selection. The deinterleaver memory ratio used to adjust memory requirements for convolutional coding (with n of 1) is 0.80. The deinterleaver memory ratio for Reed-Solomon 31-block size coding (with n of 5) is 0.85. The deinterleaver memory ratio for Reed-Solomon 255-block size coding (with n of 8) is 0.90. Each reduction is within about one percent of its minimum value.

The deinterleaver memory ratios listed here were used to adjust interleaving memory requirements defined in Equation 11 by scaling n_Q/n of Equation 11. For these adjustments, n_Q was assigned as value of $n + 1$ to include the extra bit needed for storing erasure without combining.

4.3 INTERLEAVING DELAY.

Convolutional interleaving delay requirements were defined as the interleaving time-span. Rather than defining interleaving delay in seconds, however, it was defined in units of τ_0 to conveniently permit direct delay translation of T_{il}/τ_0 selections in the results. The interleaving delay was defined as

$$D = \frac{T_{il}}{\tau_0} \ \tau_0 \ units \ . \tag{13}$$

SECTION 5

RESULTS AND OBSERVATIONS OF LINK EVALUATIONS

This section presents code performance results of the link evaluations and the corresponding interleaving memory and delay requirements. This section also includes evaluations of convolutional interleaver configurations, evaluations of erasure quantization thresholds, and estimations of interleaving memory requirements for concatenated codes.

This section is organized in five subsections with interleaver evaluations addressed first. This is followed by erasure threshold evaluations, single-stage coding results, memory estimates for concatenated codes, and concatenated coding results.

5.1 CONVOLUTIONAL INTERLEAVER EVALUATIONS.

Convolutional interleaver configurations were evaluated to determine the interleaver configurations that produce the best code performance. These configurations were then used in the code performance evaluations which followed. The interleaver evaluations were necessary because code performance is sensitive to interleaver configurations when T_{il}/τ_0 is smaller than 100.

The interleaver evaluations were conducted by evaluating code performance with different interleaver configurations and then choosing the configurations that corresponded to the lowest BER. Different codes and different T_{il}/τ_0, and E_b/N_0 values were chosen for these evaluations to assess how the best interleaver configurations depend on code and link parameters. Based on these assessments, relationships between the interleaver configurations and code and link parameters were established. These relationships were used to select interleaver configurations for all of the code and link parameter selections in the link evaluations.

Interleaver evaluations were assessed from both convolutional and Reed-Solomon code evaluations. The interleaver evaluations with convolutional coding were assessed from results in [Bogusch] and the interleaver evaluations with Reed-Solomon coding were assessed from results evaluated during this investigation. The Reed-Solomon code evaluations comprise the majority of the interleaver evaluations so they are addressed first.

Interleaver evaluations with Reed-Solomon coding were conducted with Reed-Solomon (31,15) and (31,23) codes and several values of T_{il}/τ_0 and E_b/N_0. These results are presented in plots of BER performance versus the number of interleaver rows. Maximum interleaver register sizes are also included in the results and are presented in normalized units of symbol rate and τ_0 for selection convenience*.

Interleaver evaluation with Reed-Solomon (31,15) coding, T_{il}/τ_0 of 10, and two values of E_b/N_0 are shown in Figure 8. These results indicate that interleaving with 31 rows produces the best code performance at both values of E_b/N_0. Since the best interleaver configuration occurs at both E_b/N_0, the results suggest that the best configurations are insensitive to E_b/N_0. Since the code block size has 31 symbols, the results also suggest that the best row selection equals the Reed-Solomon block size. Further evaluations were needed to support these observations, however.

Interleaver evaluations with Reed-Solomon (31,15) coding, T_{il}/τ_0 of 1, and two values of E_b/N_0 were conducted next to determine how the best interleaver configurations vary with T_{il}/τ_0 and to determine whether these configurations are also insensitive to E_b/N_0. These results, shown in Figure 9, indicate that interleaving with 5 rows produces the best code performance at both values of E_b/N_0. This supports the observation that the best interleaver configurations are insensitive to E_b/N_0 but it also indicates the best row size decreases with T_{il}/τ_0. The results indicate, however, that the performance with 31 rows is only marginally worse than the performance with 5 rows. This suggests that the best interleaver configurations are also relatively insensitive to T_{il}/τ_0.

A final interleaver evaluation was conducted with Reed-Solomon (31,23) coding and T_{il}/τ_0 of 10 to determine whether the best number of interleaver rows remains equal to or close to the Reed-Solomon block size. These results, shown in Figure 10, indicate that interleaving with 31 rows does produce the best code performance. The results show almost identical characteristics to the results of Reed-Solomon (31,15) coding and T_{il}/τ_0 of 10. This similarity coupled with previous observations suggests that, regardless of the Reed-Solomon code, T_{il}/τ_0, or E_b/N_0 selections, the best interleaver selection (or close to the best selection) has the same number of rows as the Reed-Solomon block size.

*Typical selection of interleaver configurations involves first choosing the number of interleaver rows to satisfy code correction requirements. The maximum register size is then chosen to satisfy the T_{il}/τ_0 and row selection. From the interleaver evaluation plots, the maximum register size needed to satisfy these design requirements can be selected by simply multiplying symbol rate and τ_0 by the register size representation.

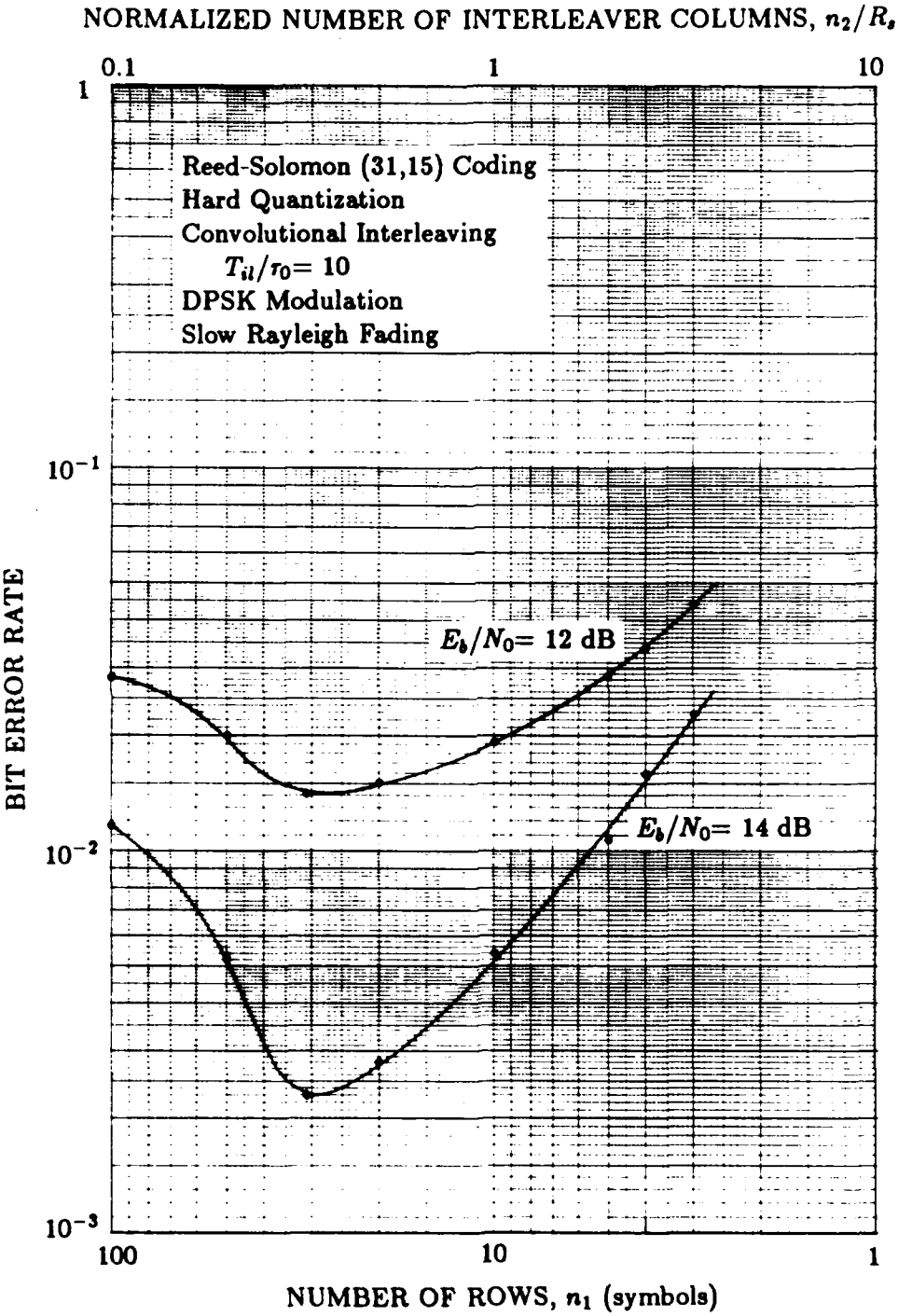


Figure 8. Evaluation of convolutional interleaving configurations with Reed-Solomon (31,15) coding and T_{il}/τ_0 of 10.

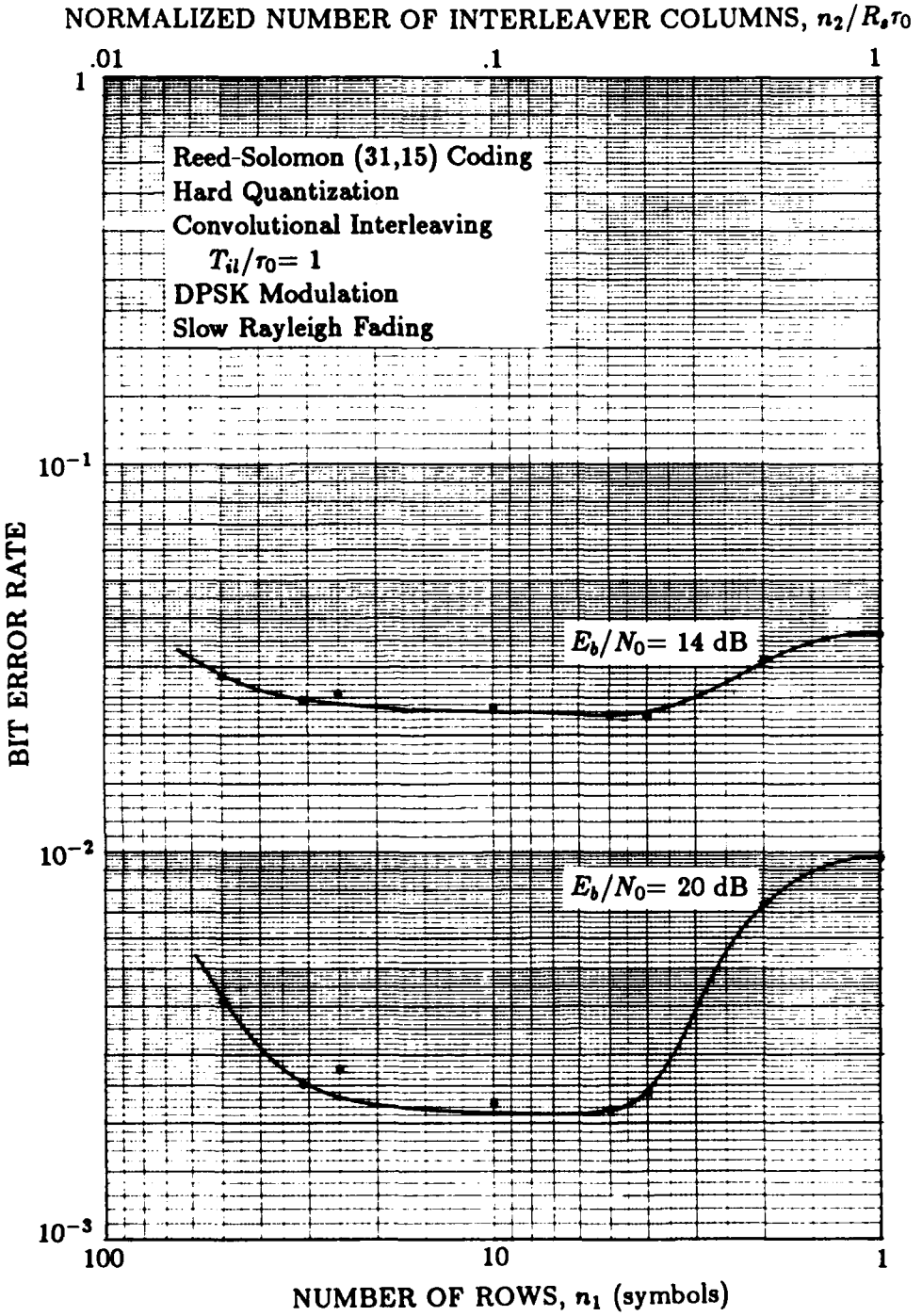


Figure 9. Evaluation of convolutional interleaving configurations with Reed-Solomon (31,15) coding and T_{il}/τ_0 of 1.

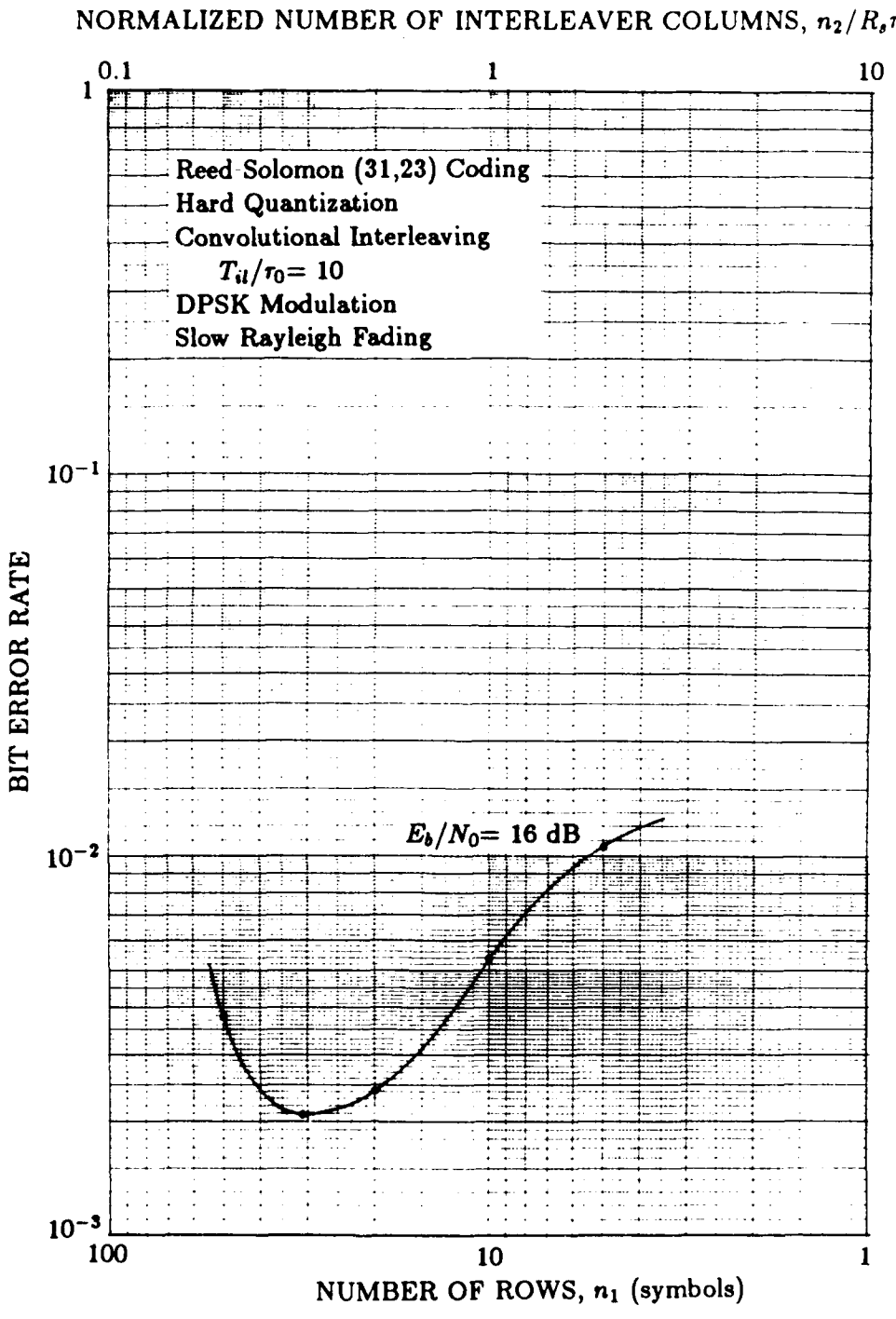


Figure 10. Evaluation of convolutional interleaving configurations with Reed-Solomon (31,23) coding and T_{il}/τ_0 of 10.

Based on this conclusion, interleaver configurations with Reed-Solomon coding were selected with their number of rows equal to the code block size. The validity of this selection was tested in evaluations of 256 block Reed-Solomon codes. In all of the tests, the selections proved to be good ones[†].

After selecting the number of interleaver rows, the maximum interleaver register size (in symbols) was selected to satisfy the desired interleaver span and the row selection. It was selected as

$$n_{reg} = \frac{R_s \tau_0}{n_{row}} \frac{T_{il}}{\tau_0} \tag{14}$$

where R_s is the symbol rate and n_{row} is the number of interleaver rows.

For interleaver selections with convolutional coding, interleaver assessments were made from evaluations in [Bogusch]. These evaluations were conducted with convolutional rate 1/2 coding with T_{il}/τ_0 of 10. The evaluations indicate that the best number of interleaver rows is between 10 to 50. This range of row sizes indicates that convolutional coding is less sensitive to row selections than Reed-Solomon coding.

Based these results, interleaver configurations for convolutional coding were selected with row sizes between 10 and 50. These selections were made for both rate 1/2 and 1/3 coding and for different values of T_{il}/τ_0 and E_b/N_0 because interleaver configurations are relatively insensitive to these parameters. Maximum interleaver register sizes were selected after row selections using the formula in Equation 14.

5.2 ERASURE THRESHOLD EVALUATIONS.

Similar to the interleaver evaluations, erasure threshold evaluations were conducted prior to code performance evaluations to find the erasure thresholds that produce the best code performance. Erasure thresholds were evaluated by evaluating code performance with different erasure thresholds and selecting the thresholds that corresponded to the lowest BER performance. Unlike the interleaver evaluations, however, the erasure threshold evaluations were conducted for all selected codes and for different values of T_{il}/τ_0 in each case. Extensive evaluation was required because relationships between the best thresholds and code and link parameters were not accurately predictable[‡].

[†]With T_{il}/τ_0 smaller than 10, some interleaver configurations were chosen with a smaller number of rows than the Reed-Solomon block size. These selections were made to optimize code performance but, as indicated in Figure 9, they provided only marginal code improvement.

[‡]The exception to this is E_b/N_0 values. The best erasure thresholds were found to be E_b/N_0 insensitive.

The best erasure thresholds determined by the threshold evaluations are listed in Table 1. Since these results required a large number of evaluations, the number of thresholds selected for each code and T_{il}/τ_0 evaluation was limited to 3 or 4[§]. This did not adversely affect the best threshold selections because the difference in code performance between the thresholds was relatively small. For unspecified thresholds with T_{il}/τ_0 values of 100 and 3, the tabulated results with T_{il}/τ_0 of 10 and 1 were used to extrapolate and interpolate the unspecified thresholds.

Table 1. Erasure threshold selections.

Code	T_{IL}/τ_0	Erasure Threshold
Convolutional	10	0.15
rate 1/2, K = 7	1	0.05
Convolutional	10	0.2
rate = 1/3, K = 7	1	0.1
Reed-Solomon	10	0.1
(31,15)	1	0.01
Reed-Solomon	10	0.05
(31,23)	1	0.01
Reed-Solomon	10	0.05
(63,31)	1	0.01
Reed-Solomon	10	0.05
(63,47)		
Reed-Solomon	10	0.05
(255,127)	1	0.01
Reed-Solomon	10	0.01
(255,191)	1	0.01

The thresholds in Table 1 indicate several characteristics about how erasure thresholds vary with code and T_{il}/τ_0 selections. One characteristic is that the best

[§]These thresholds comprised one of the following ratios of the nominal signal value: 0.01, 0.05, 0.1, 0.15, 0.2, and 0.3.

threshold choices decrease with reductions in T_{il}/τ_0. Other characteristics are that the best threshold choices decrease with increases in code rate and increases in Reed-Solomon block size.

The first characteristic is the result of longer bursts of erasures that occur as interleaving becomes less effective at randomizing symbol error. As T_{il}/τ_0 decreases, the erasure threshold must be reduced to reduce the number of erasures received at the decoder to reach a better balance between errors and erasures.

The other threshold characteristics are the result of different code properties. In the case of code rate, an increase in code rate decreases code corrective capability, subsequently decreasing the number of correctable errors and erasures. Consequently, higher rate codes require smaller erasure settings to reduce the number of erasures that are received at decoding. In the case of Reed-Solomon block size, an increase in block size increases the probability of the decoder receiving a symbol erasure since larger codes have more bits per symbol[¶]. Consequently, larger Reed-Solomon codes require a smaller erasure setting to reduce the number of erasures they receive at decoding.

In addition to the erasure threshold evaluations, a scale factor of 1.0 was selected for soft quantization with convolutional coding. This scale factor was selected from scale factor evaluations with convolutional rate 1/2 coding and T_{il}/τ_0 of 10. Only one evaluation was conducted and only one scale factor was selected because scale factors are relatively insensitive to convolutional code, T_{il}/τ_0, and E_b/N_0 selections.

5.3 SINGLE-STAGE CODING RESULTS.

This subsection presents the results of code performance evaluations and interleaving memory and delay requirements for the single-stage codes.

The code performance results are presented in plots of decoded BER performance versus E_b/N_0 requirements. E_b/N_0 requirements for 10^{-5} BER with different T_{il}/τ_0 selections were extracted from these results and used to determine interleaving memory and delay requirements. Because of time limitations, the code

[¶]More bits per symbol causes a higher probability of receiving an erasure because code symbol erasures are converted from binary demodulated erasures. Any demodulated erasure within a code symbol-span is converted to a code symbol erasure (see Section 3.1).

performance results were evaluated with BER larger then 10^{-4}*. E_b/N_0 values at 10^{-5} BER were subsequently extrapolated from the results at higher BER.

The interleaving memory requirements are presented in plots of $R_d\tau_0$ memory units versus E_b/N_0 requirements with 10^{-5} BER. The memory values were converted from T_{il}/τ_0, code rate, and quantization selections according to the memory conversion of Section 3. The interleaving delay requirements are presented in plots of τ_0 delay units versus E_b/N_0 requirements with 10^{-5} BER. The delay values were equated to T_{il}/τ_0.

This section is divided in three subsections with the first subsection containing the convolutional coding results. This is followed by the Reed-Solomon (31,K) coding results and the Reed-Solomon (255,K) coding results.

5.3.1 Convolutional Coding.

The code performance results of convolutional rate 1/2, K=7 coding with hard-decision (hard quantization), erasure-decision (erasure quantization), and 3-bit soft-decision (3-bit soft quantization) decoding are shown in Figures 11, 12, and 13 respectively. These results were evaluated with T_{il}/τ_0 values of 100, 10, 3, and 1.

These results translate to the interleaving memory requirements shown in Figure 14. The results of Figure 14 indicate that memory requirements for erasure-decision coding are in general (over all T_{il}/τ_0) smaller than those of hard-decision or soft-decision coding. The reason the disparity in erasure-decision and hard-decision memory requirements is because erasure-decision coding has much better code performance than hard-decision coding. The reason for the disparity between erasure-decision and soft-decision memory requirements is because erasure-decision coding has much better memory utilization than soft-decision coding.

The interleaving delay requirements for these codes are shown next in Figure 15. These results indicate that hard-decision coding has much larger delay requirements than those of soft-decision and erasure-decision coding. The results also indicate that erasure-decision coding has similar delay requirements to those of soft-decision coding for delays larger than about 10 τ_0. erasure-decision coding has smaller delay requirements for delays smaller than 10 τ_0.

*Even with BER larger than 10^{-4}, the processing time required for code performance evaluations at separate E_b/N_0 ran several hours on the ELXSI 6400 computer. This was significant considering that several E_b/N_0 evaluations were required for each T_{il}/τ_0 performance curve and four T_{il}/τ_0 performance curves were evaluated with each code selection.

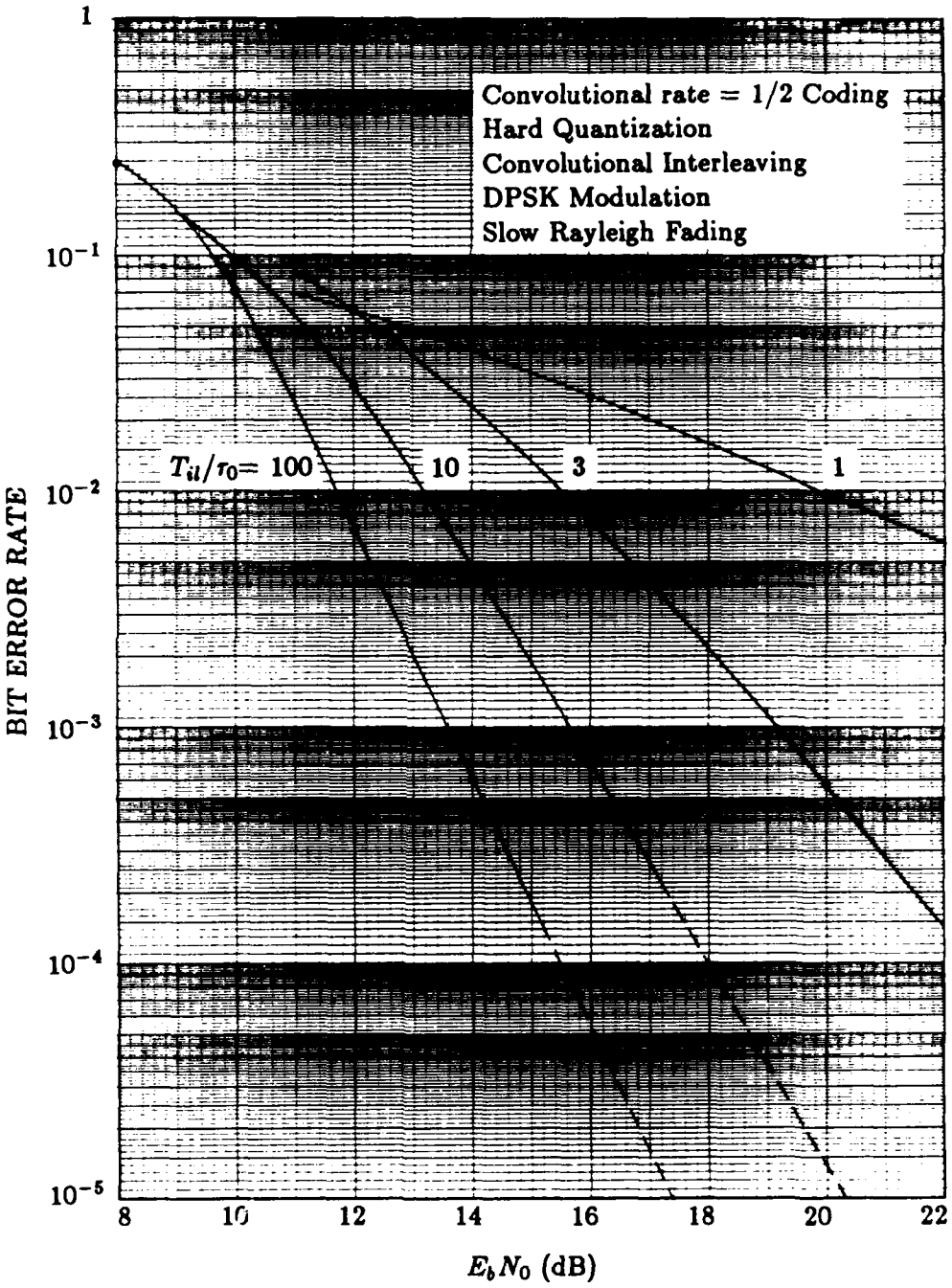


Figure 11. Code performance of convolutional rate 1/2 coding with hard quantization.

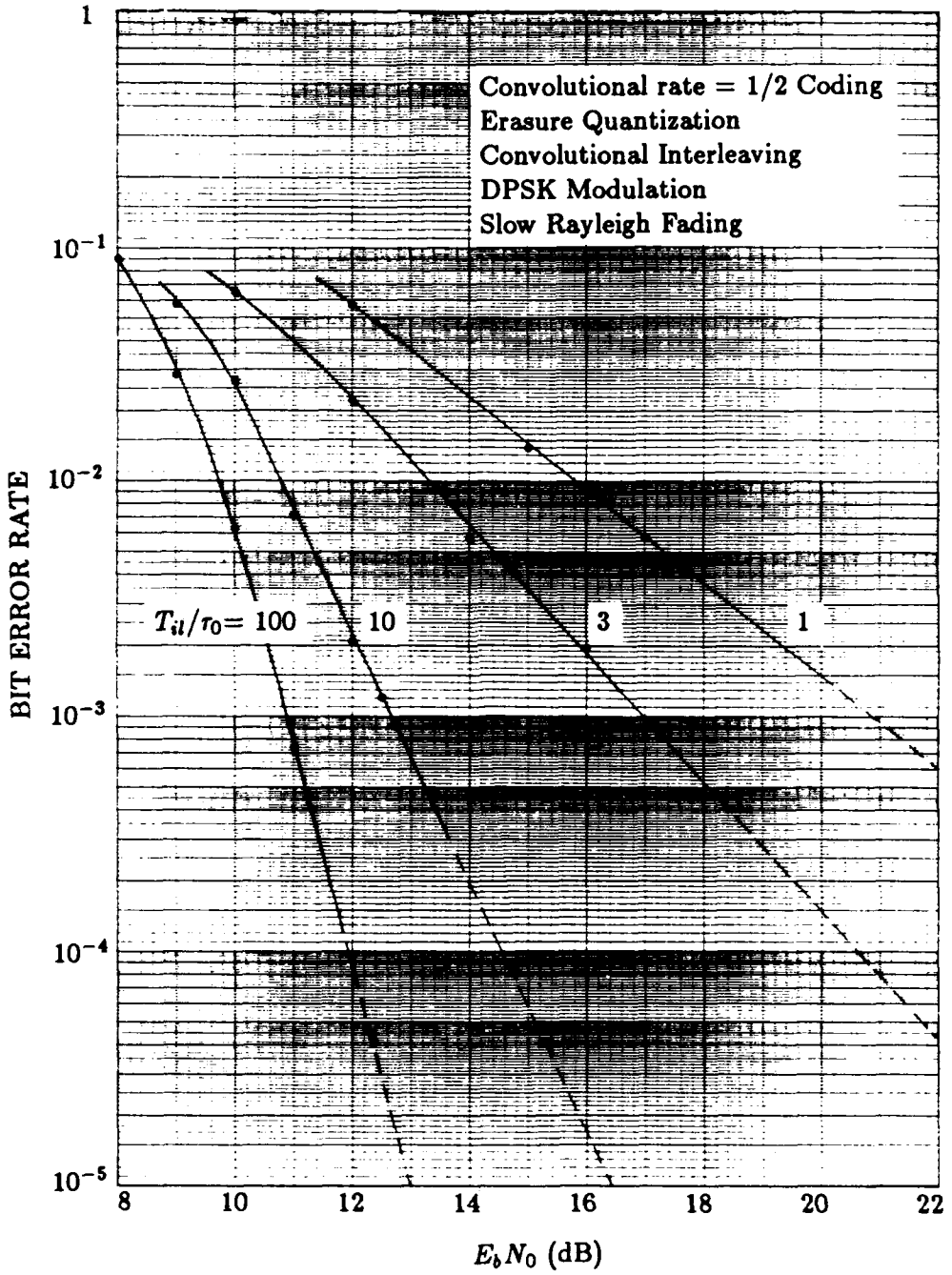


Figure 12. Code performance of convolutional rate 1/2 coding with erasure quantization.

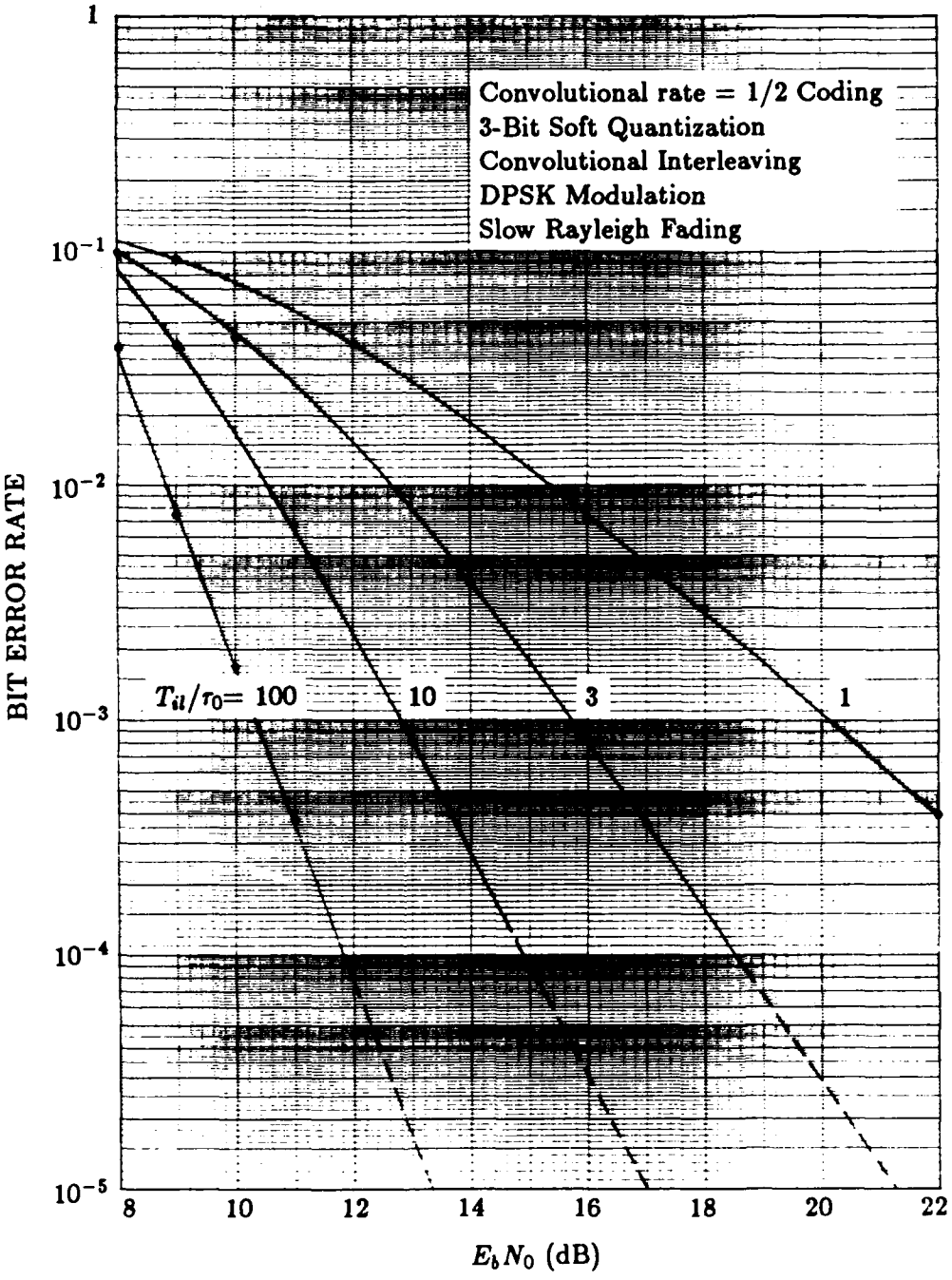


Figure 13. Code performance of convolutional rate 1/2 coding with 3-bit soft quantization.

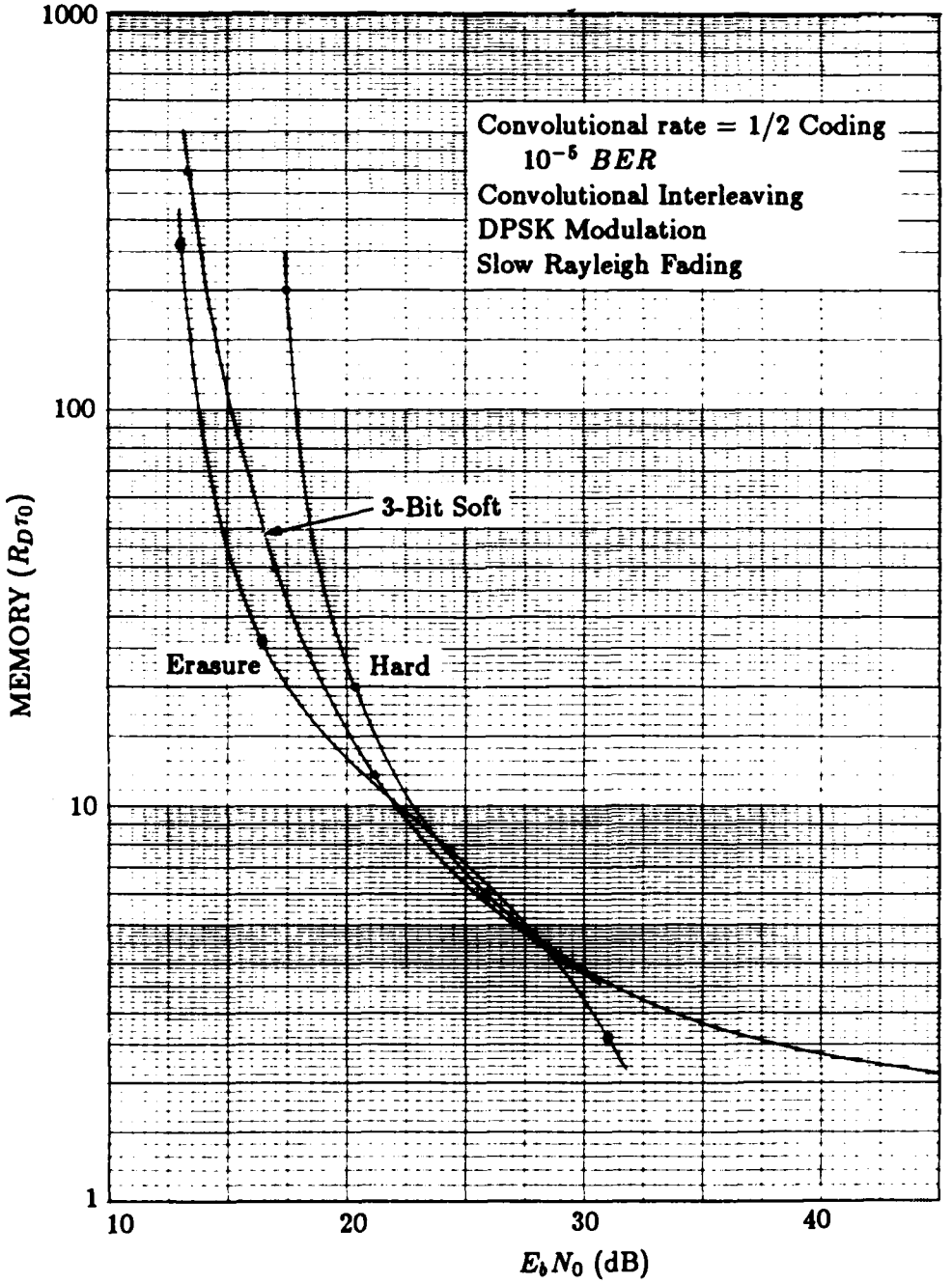


Figure 14. Interleaver memory requirements for convolutional rate 1/2 coding with 10^{-5} BER.

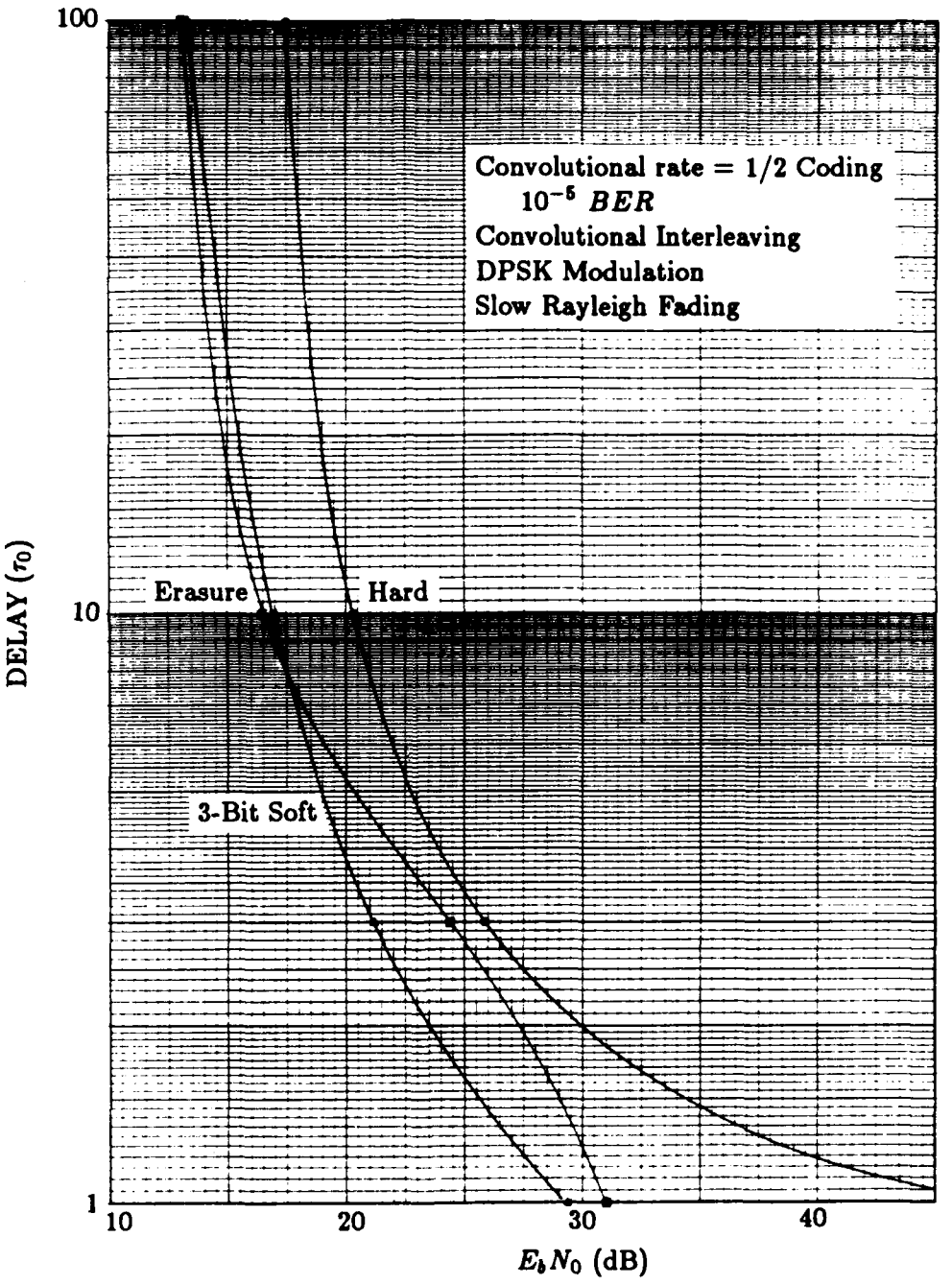


Figure 15. Interleaver delay requirements for convolutional rate 1/2 coding with 10^{-5} BER.

Since the delay requirements were determined simply by equating delay with T_{il}/τ_0, the delay results also reflect code performance. Consequently, the results of Figure 15, indicate that erasure-decision code performance is much better than hard-decision code performance and erasure-decision code performance is similar to erasure-decision code performance. The difference between hard- and erasure-decision code performance is not surprising since hard quantization has only two levels of quantization and erasure quantization has three levels of quantization. However, the similarity between erasure-decision and soft-decision code performance is surprising because the soft quantization was conducted with eight levels of quantization. To explain this similarity consider the following comparison.

The soft quantization operation was conducted as shown in the upper portion of Figure 16 with a quantization scale factor of 1.0. The erasure quantization operation for these evaluations was conducted as shown in the lower portion of Figure 16 with an erasure threshold of 0.15 (for T_{il}/τ_0 of 10). The quantization value is denoted as l and the normalized demodulation value (quantization input) is denoted as d.

Figure 16 indicates that both quantization operations had different quantization boundaries and that erasure quantization included a neutral (erasure) metric assignment. As a result, erasure quantization apparently compensated for performance loss from smaller overall signal resolution by not weighting unreliable signal values during periods of signal fading. These observations suggest that an alternative quantization operation that contains a combination of erasure and soft quantization may provide even better performance.

The code performance results of convolutional rate 1/3, K=7 coding with hard-decision, erasure-decision, and soft-decision decoding performance are shown next in Figures 17, 18, and 19. These results were also evaluated with T_{il}/τ_0 values of 100, 10, 3, and 1.

These results translate to the interleaving memory and delay requirements shown in Figures 20 and 21. As in the memory requirements for rate 1/2 coding, the memory requirements in Figure 20 indicate that erasure-decision coding generally has lower memory requirements than those of soft-decision and hard-decision coding. The delay requirements in Figure 21 indicate that hard-decision coding has much larger memory requirements than those of soft-decision and erasure-decision coding. These results also indicate that erasure-decision coding has 1 to 2 dB larger E_b/N_0 requirements than those of soft-decision coding at comparable delays. In the memory results, this loss is compensated by more efficient memory utilization.

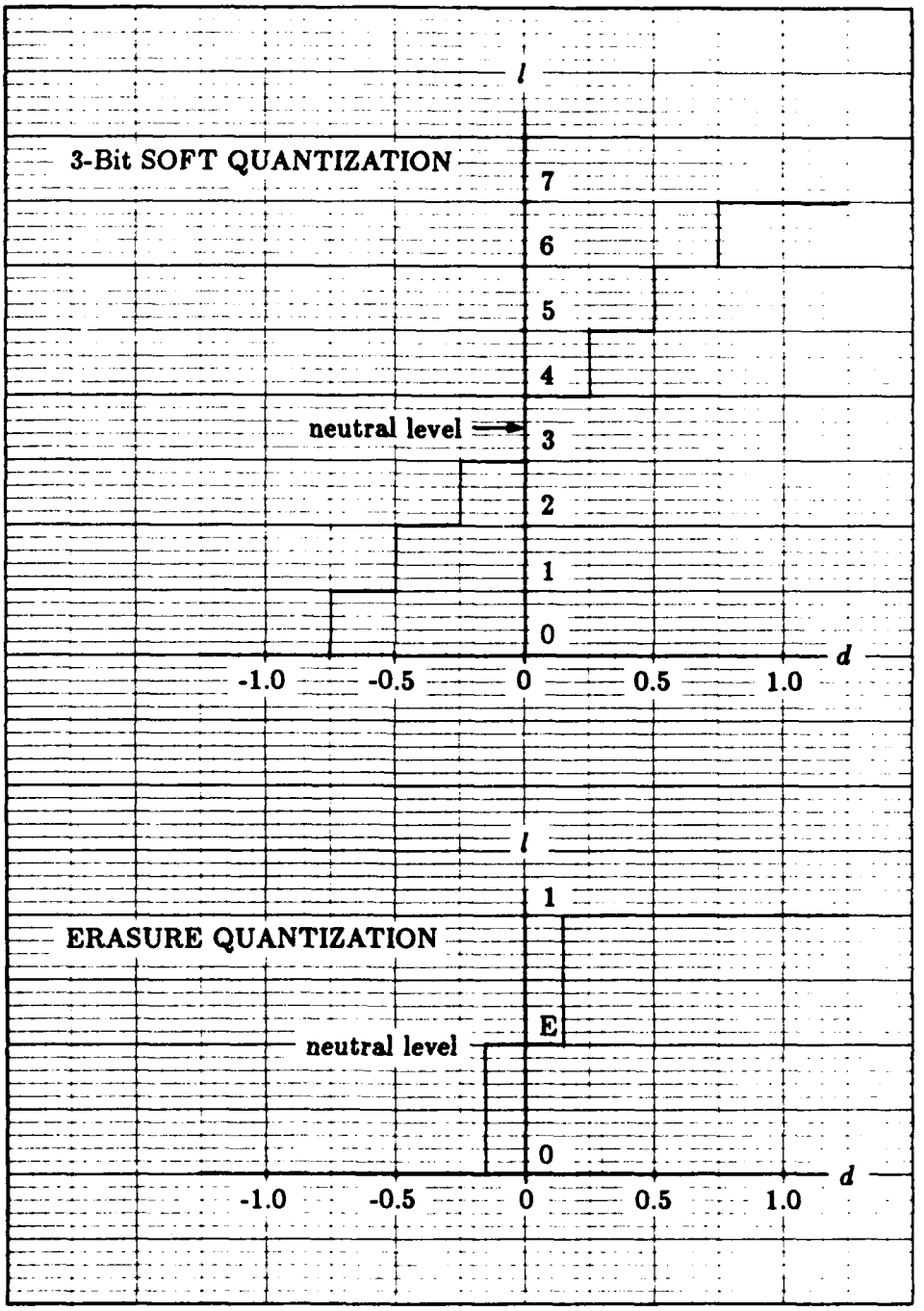


Figure 16. Comparison of erasure and 3-bit soft quantization for convolutional rate 1/2 coding.

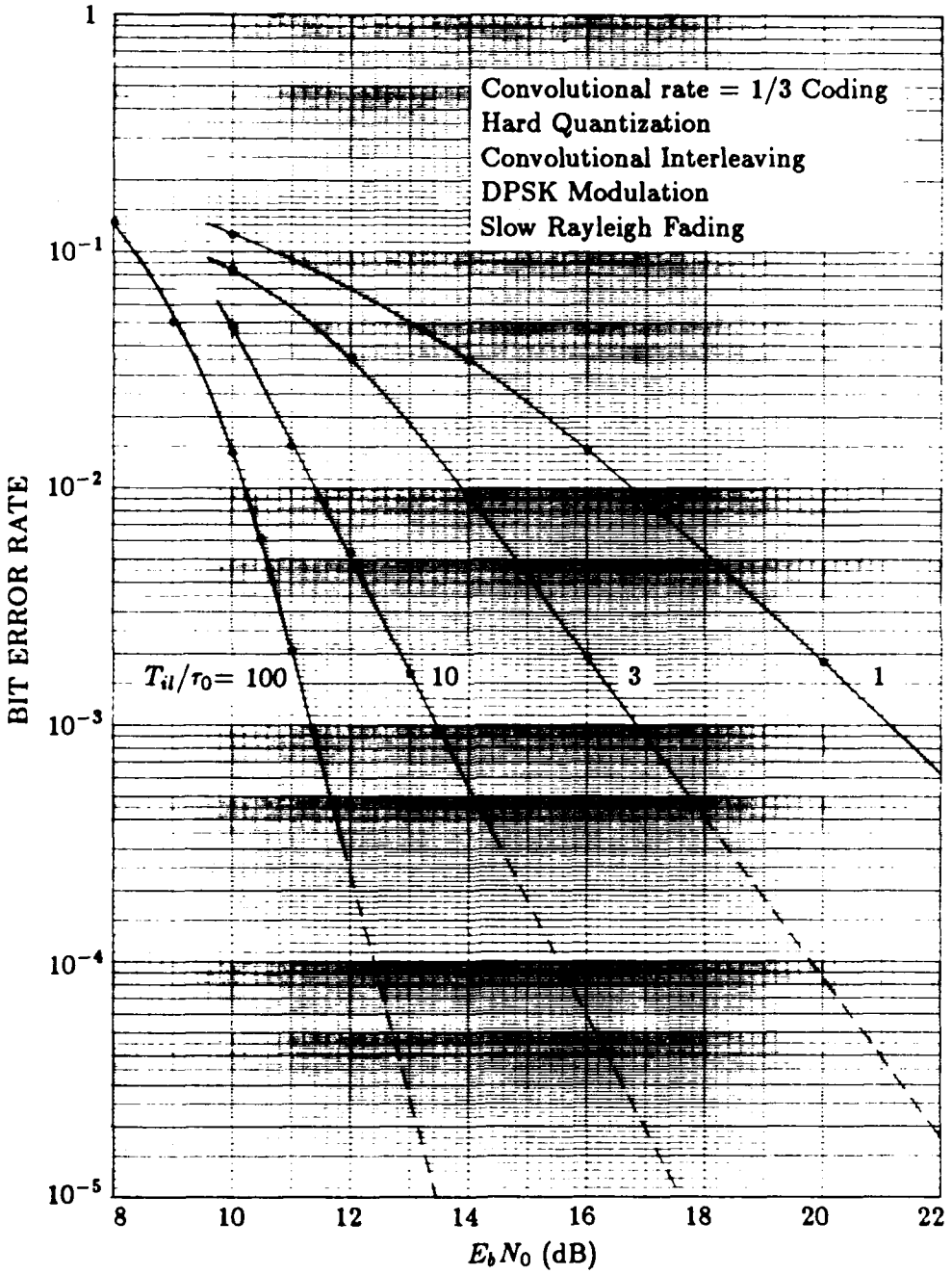


Figure 17. Code performance of convolutional rate 1/3 coding with hard quantization.

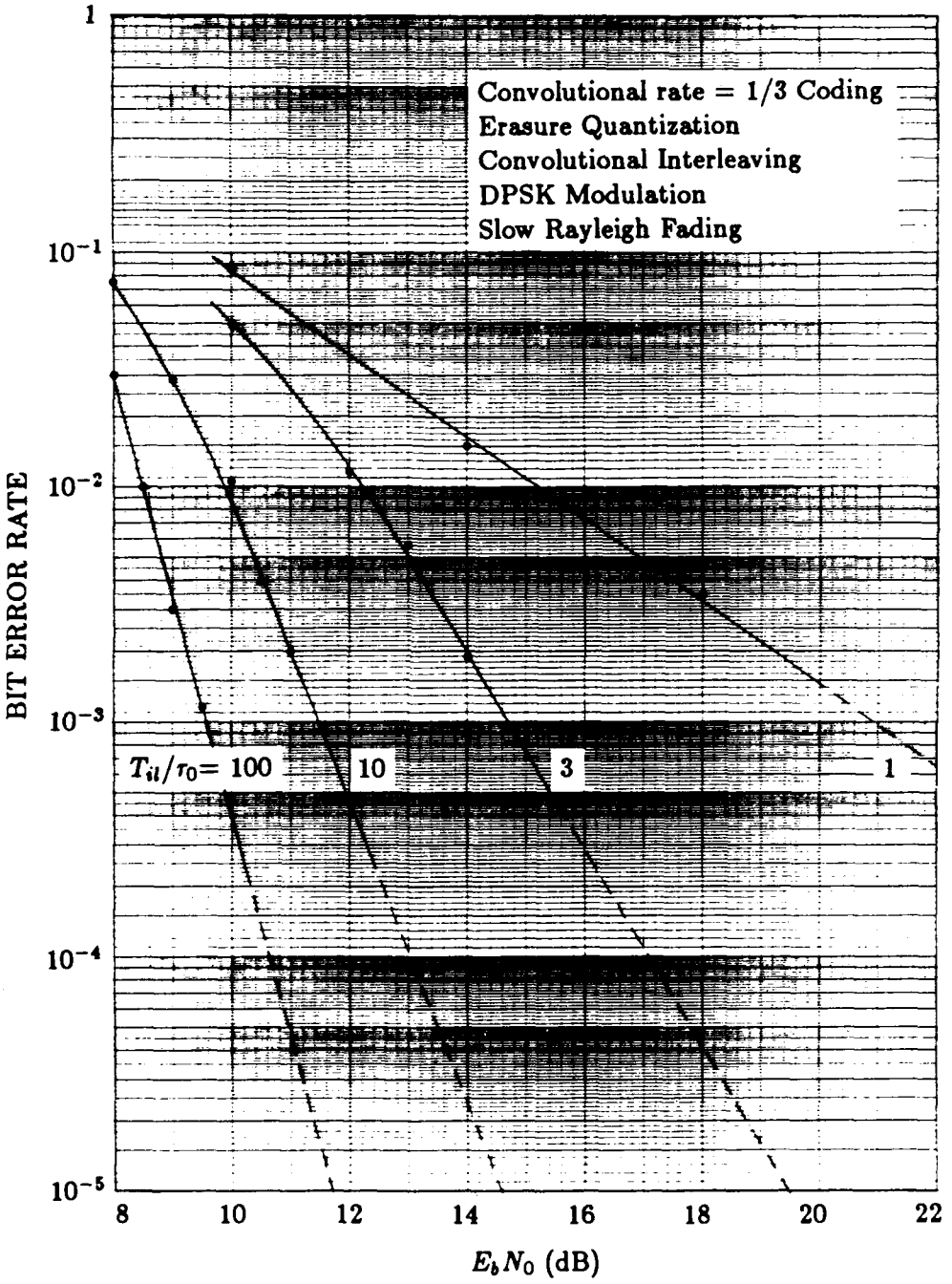


Figure 18. Code performance of convolutional rate 1/3 coding with erasure quantization.

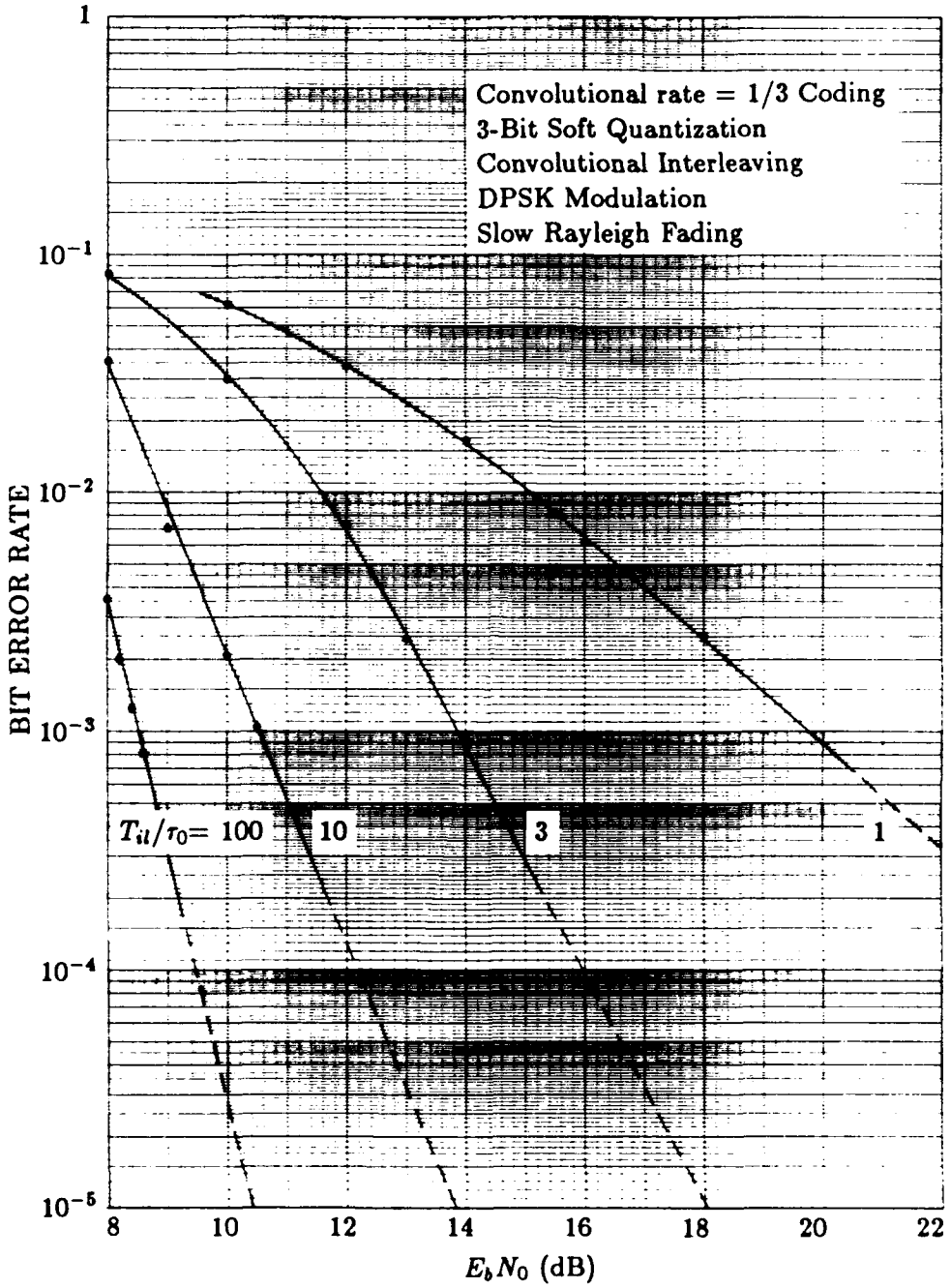


Figure 19. Code performance of convolutional rate 1/3 coding with 3-bit soft quantization.

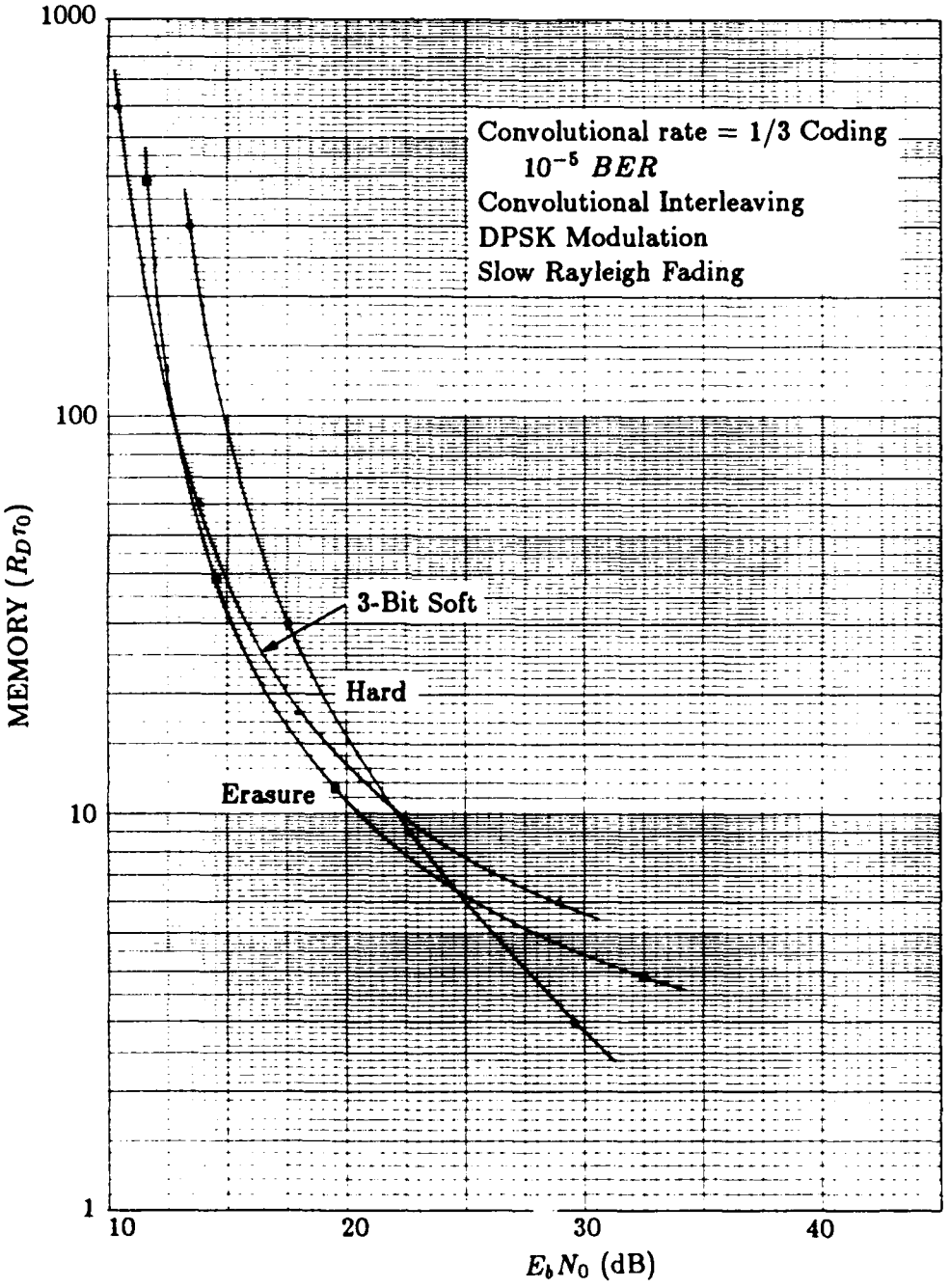


Figure 20. Interleaver memory requirements for convolutional rate 1/3 coding with 10^{-5} BER.

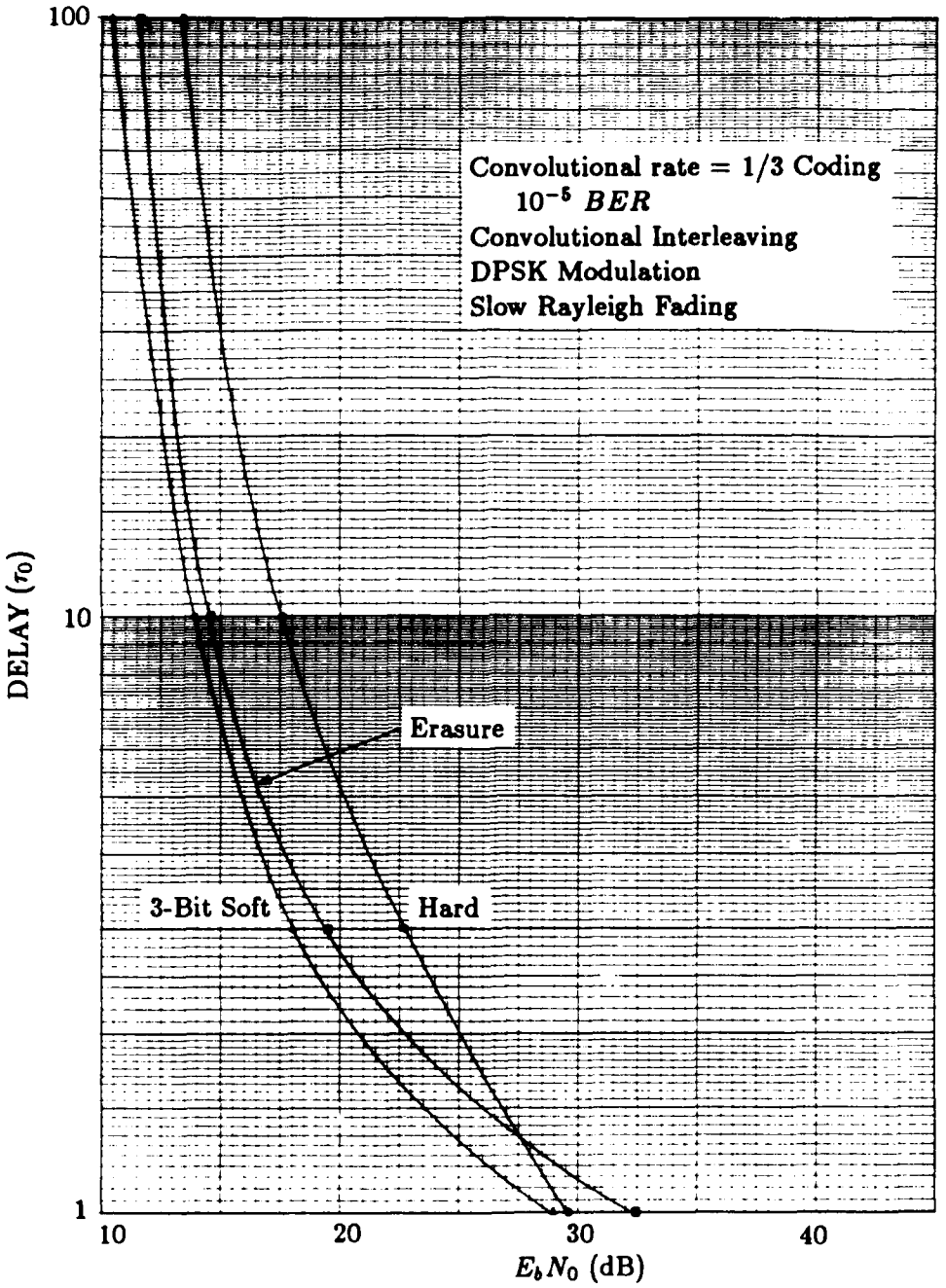


Figure 21. Interleaver delay requirements for convolutional rate 1/3 coding with 10^{-5} *BER*.

Comparing the interleaving memory and delay requirements of rate 1/3 and rate 1/2 coding, the rate 1/3 code results show substantially smaller interleaving requirements than those of the rate 1/2 code results. The comparisons also show that, with erasure quantization, both codes generally have the smallest memory requirements. The comparisons further show that erasure-decision codes have comparable or somewhat larger delay requirements than those of the soft-decision codes.

To assess what interleaving memory and delay savings occur from choosing an efficient rate 1/3 code selection instead of a standard rate 1/2 code selection, consider the requirements for convolutional rate 1/3 erasure-decision coding and convolutional rate 1/2 soft-decision coding. If T_{il}/τ_0 is 10, the rate 1/2 code has an interleaving delay requirement of 10 τ_0. It requires about 17 dB E_b/N_0 for this requirement (with 10^{-5} BER) and, for 17 dB, has a memory requirement of about 40 $R_d\tau_0$. For the same memory and delay requirements, the rate 1/3 code selection has an E_b/N_0 requirement of about 14.5 dB.

This comparison illustrates that substantial E_b/N_0 and memory and delay reductions may be possible simply by choosing a rate 1/3 convolutional code with erasure quantization. Rate 1/3 convolutional coding does not have the lowest requirements over the complete range of E_b/N_0 and interleaving requirements, however. Therefore, the best code selection depends on the E_b/N_0 and interleaving memory and delay specifications. This will become more apparent in the Reed-Solomon code evaluations.

5.3.2 Reed-Solomon (31,K) Coding.

The code performance results of Reed-Solomon (31,15) and (31,23) coding with hard-decision and erasure-decision decoding are shown in Figures 22, 23, 24, and 25. These results were evaluated with T_{il}/τ_0 values of 100, 10, 3, and 1.

These results translate to the interleaving memory requirements shown in Figure 26. The results of Figure 26 indicate that Reed-Solomon (31,15) coding has smaller memory requirements than those of Reed-Solomon (31,23) coding with both hard and erasure quantization. The results also indicate that the memory requirements with erasure quantization are smaller than those with hard quantization in both Reed-Solomon (31,15) and (31,23) codes. The memory savings from erasure quantization is larger with Reed-Solomon (31,15) coding than with Reed-Solomon (31,23) coding, however.

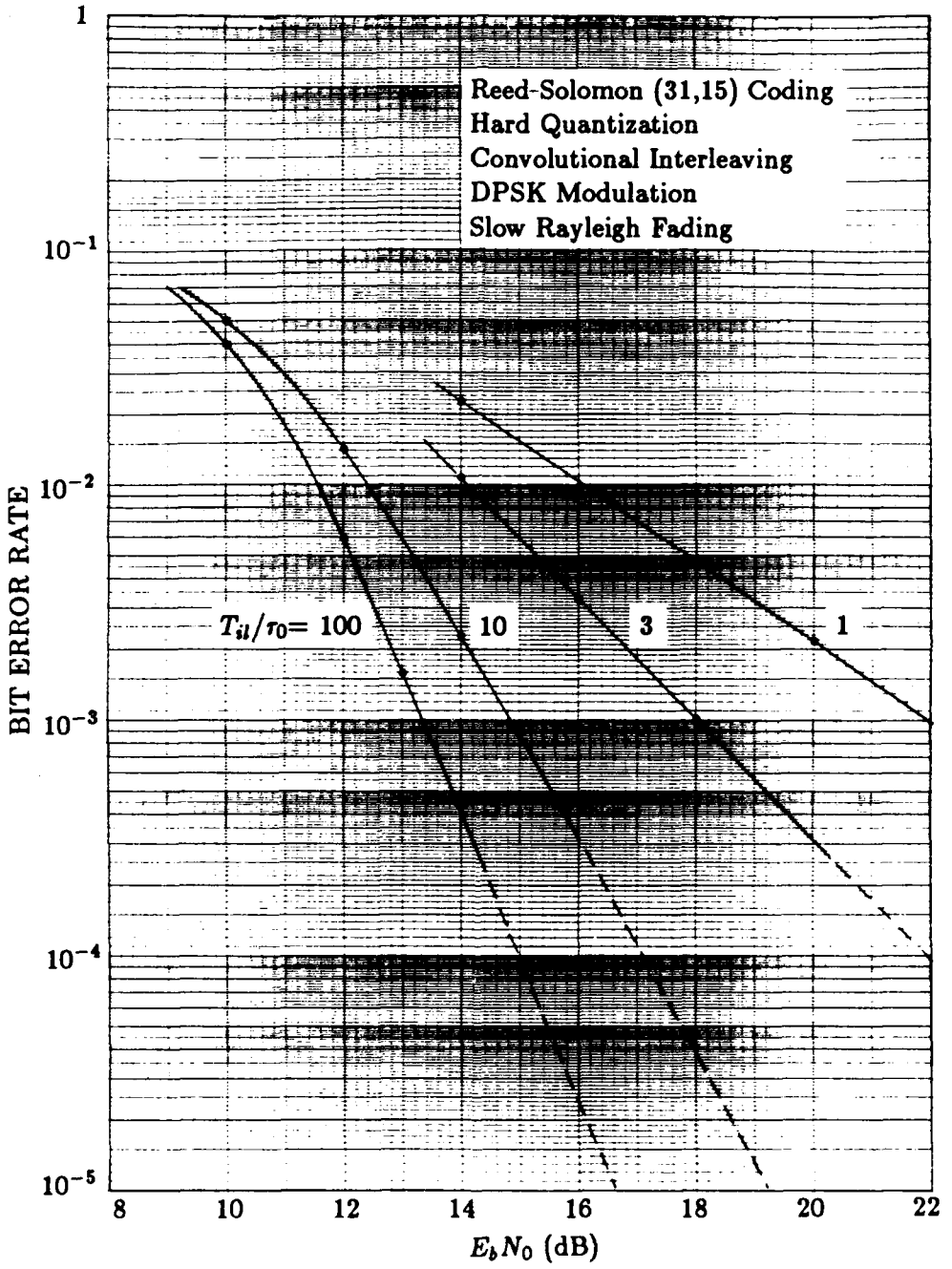


Figure 22. Code performance of Reed Solomon (31,15) coding with hard quantization.

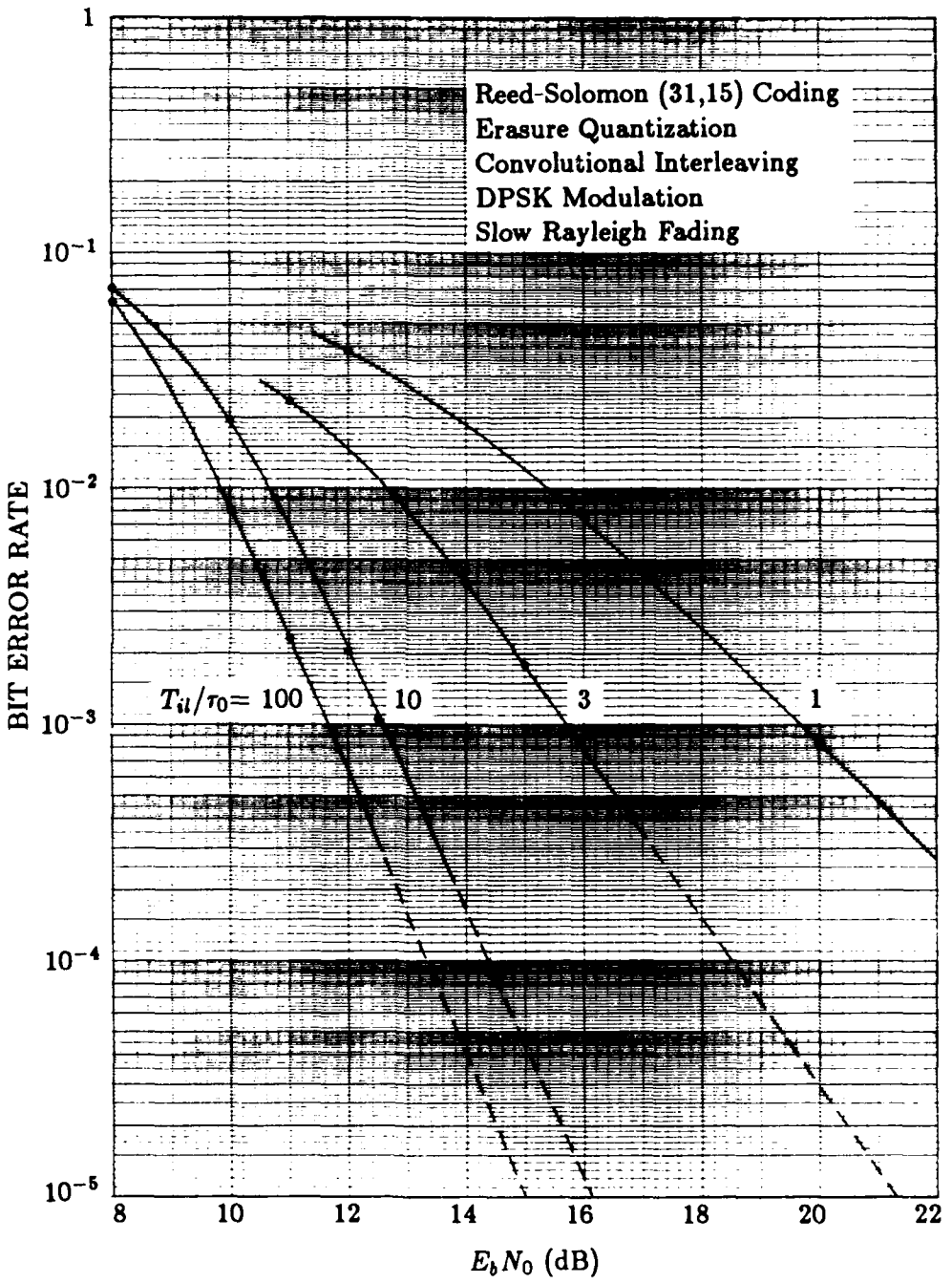


Figure 23. Code performance of Reed-Solomon (31,15) coding with erasure quantization.

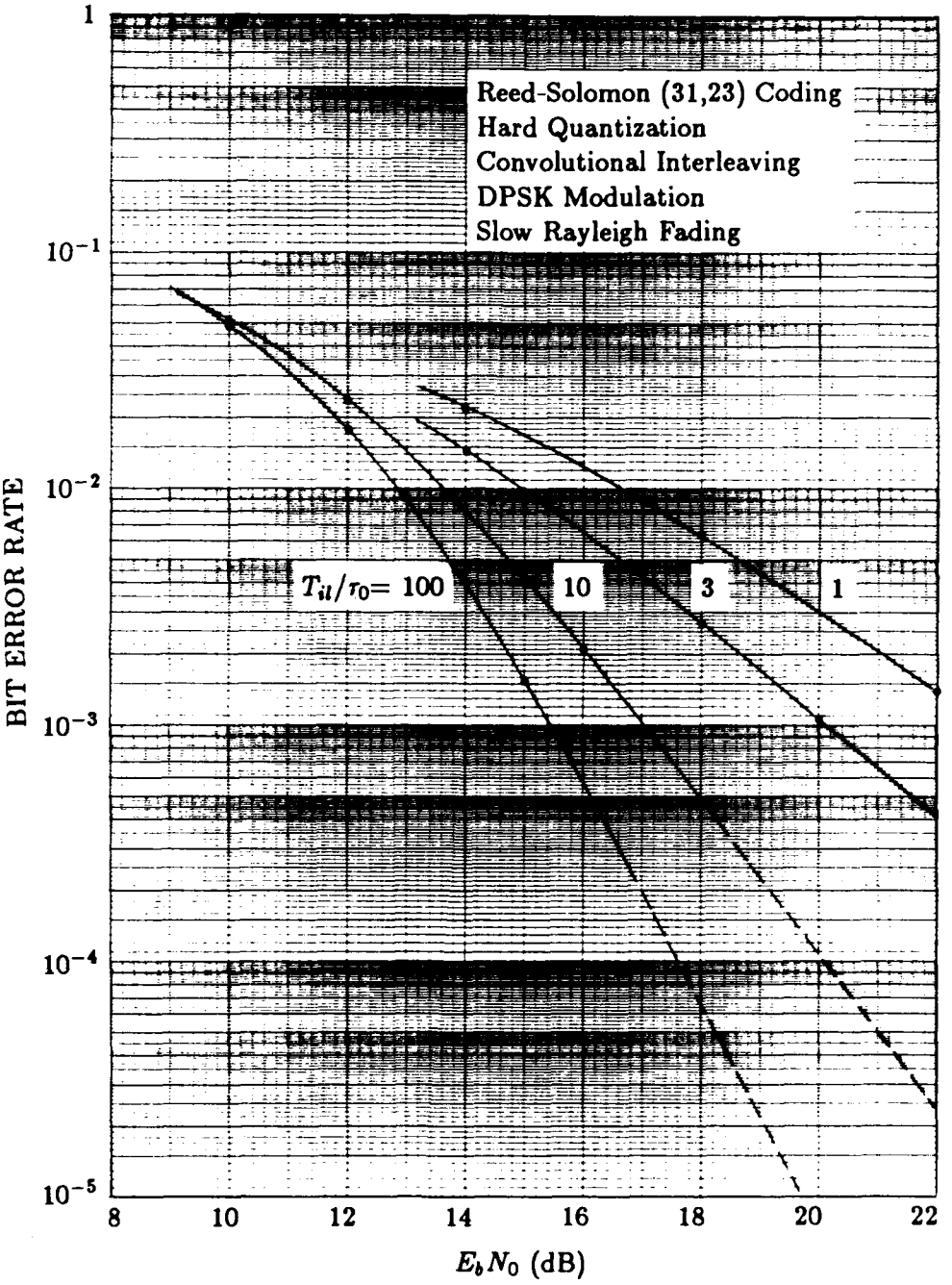


Figure 24. Code performance of Reed-Solomon (31,23) coding with hard quantization.

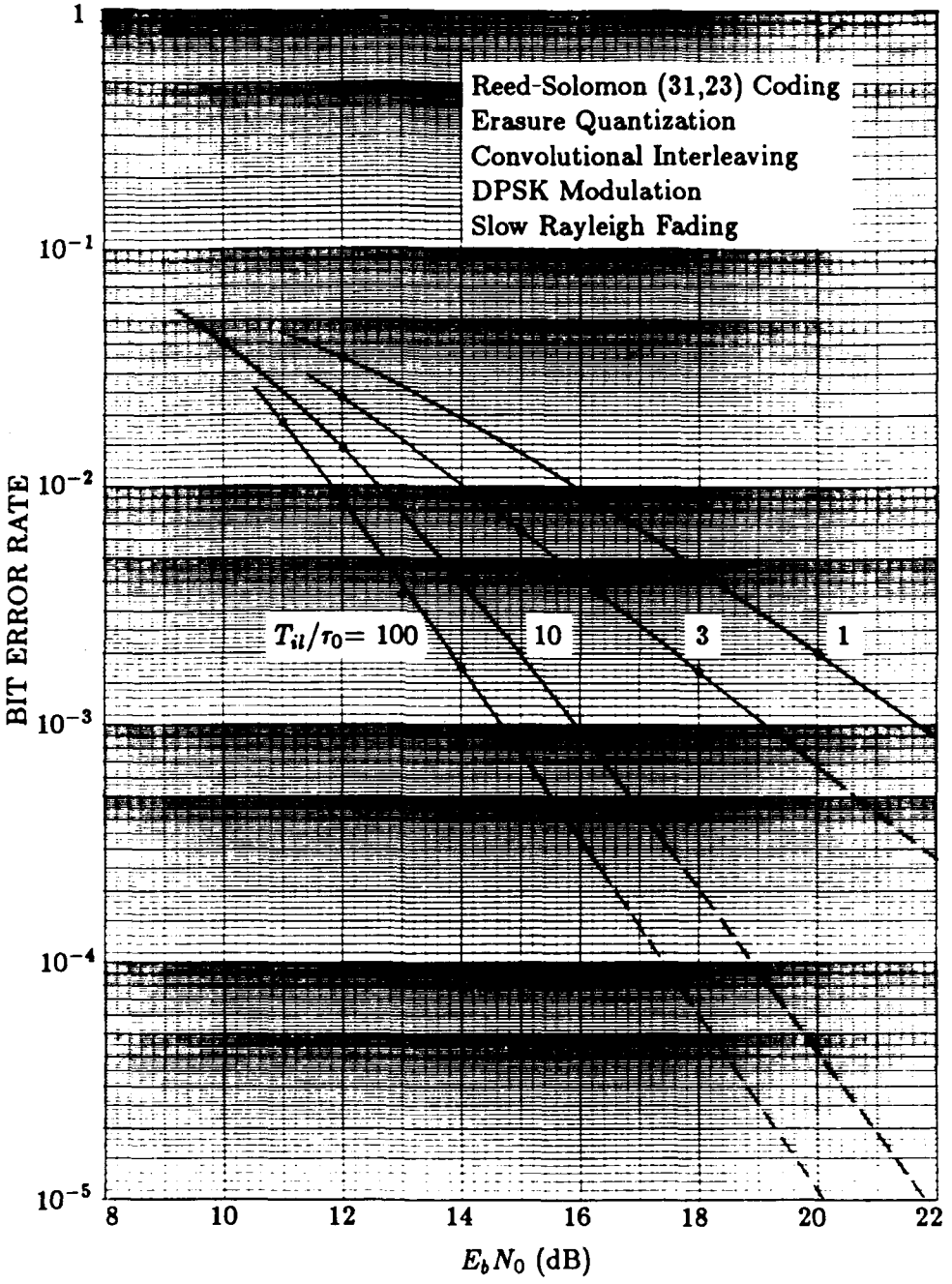


Figure 25. Code performance of Reed-Solomon (31,23) coding with erasure quantization.

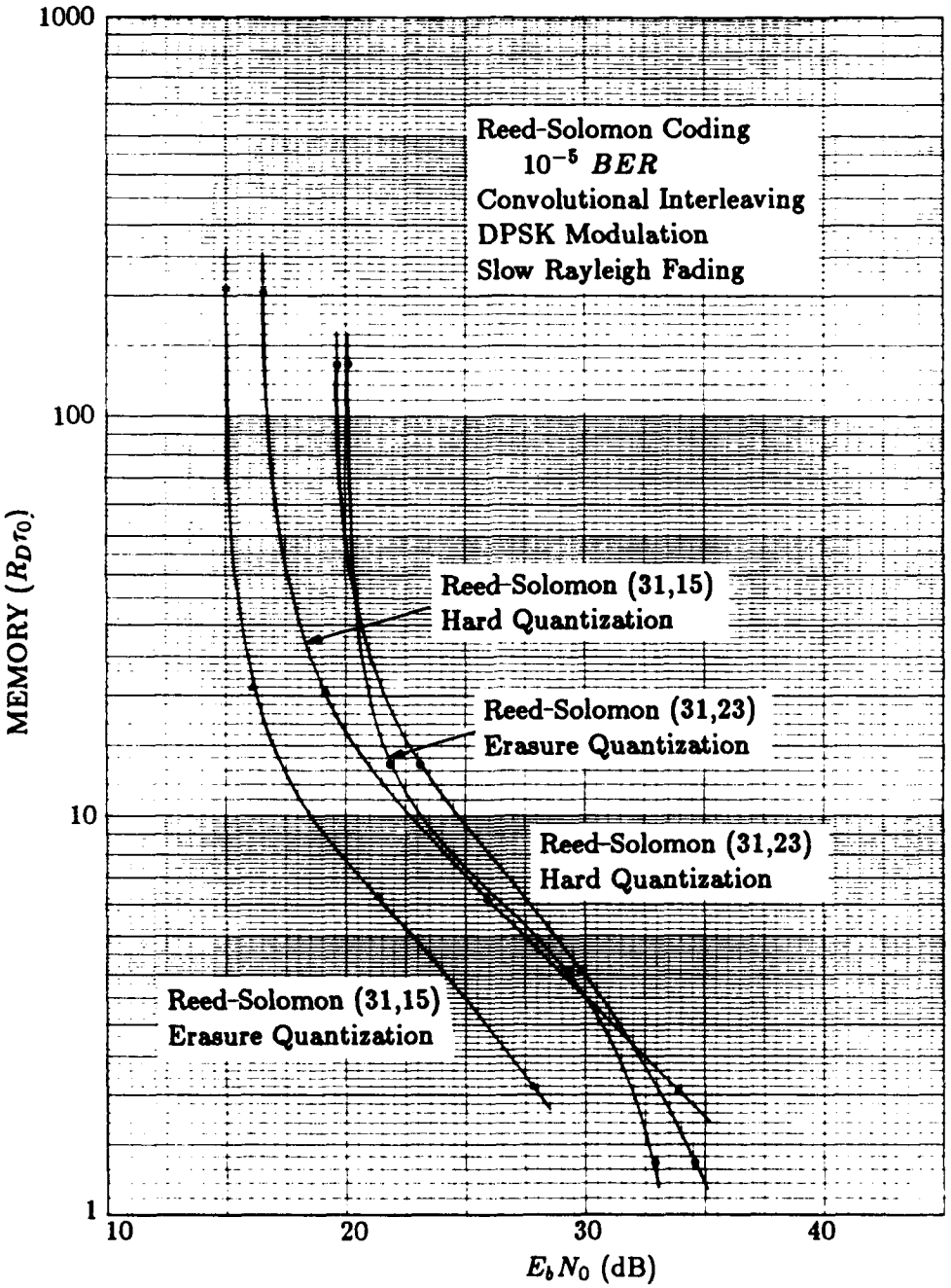


Figure 26. Interleaver memory requirements for Reed-Solomon (31,K) coding with 10^{-5} BER.

The interleaving delay requirements for these code selections are shown next in Figure 27. These results indicate that, as in the memory results, Reed-Solomon (31,15) coding, with both erasure and hard quantization, has smaller delay requirements than those of Reed-Solomon (31,23) coding. Comparison of the memory and delay results indicates that Reed-Solomon (31,15) erasure-decision coding clearly has the lowest memory and delay requirements.

Compared to convolutional rate 1/3 erasure-decision coding, Reed-Solomon (31,15) erasure-decision coding has larger memory requirements for E_b/N_0 below about 16 dB and smaller memory requirements for E_b/N_0 above 16 dB. Reed-Solomon (31,15) erasure-decision coding also has larger delay requirements that those of convolutional rate 1/3 coding over most of the T_{il}/τ_0 range. It has comparable or lower delay requirements than those of the convolutional rate 1/2 code selections, however.

For a quantitative comparison of the memory and delay requirements for Reed-Solomon (31,15) erasure-decision coding and convolutional rate 1/2 soft-decision coding, consider the same memory and delay requirements that were used for previously comparing convolutional rate 1/2 and 1/3 codes. With 40 $R_d\tau_0$ memory, Reed-Solomon (31,15) erasure-decision coding has an E_b/N_0 requirement which is about 1.5 dB smaller than that of convolutional rate 1/2, soft-decision coding. With 10 τ_0 delay, Reed-Solomon (31,15) erasure-decision coding has an E_b/N_0 requirement which is about 1 dB smaller than that of convolutional rate 1/2, soft-decision coding.

At these memory and delay requirements, the Reed-Solomon (31,15) code savings are not as substantial as those of convolutional rate 1/3 coding. However, for smaller interleaving requirements, Reed-Solomon (31,15) has smaller E_b/N_0 requirements than both rate 1/2 and rate 1/3 convolutional codes.

5.3.3 Reed-Solomon (255,K) Coding.

The code performance results of Reed-Solomon (255,127) and (255,191) coding with hard-decision and erasure-decision decoding are shown in Figures 28, 29, 30, and 31. These results were evaluated with T_{il}/τ_0 values of 100, 10, 3, and 1.

These results translate to the interleaving memory requirements shown in Figure 32. The results of Figure 32 indicate that the memory requirements with erasure quantization are smaller than with hard quantization for both (255,127) and (255,191) codes. The memory savings from erasure quantization is much larger with Reed-Solomon (255,127) erasure-decision coding than with Reed-Solomon (255,191) erasure-decision coding, however.

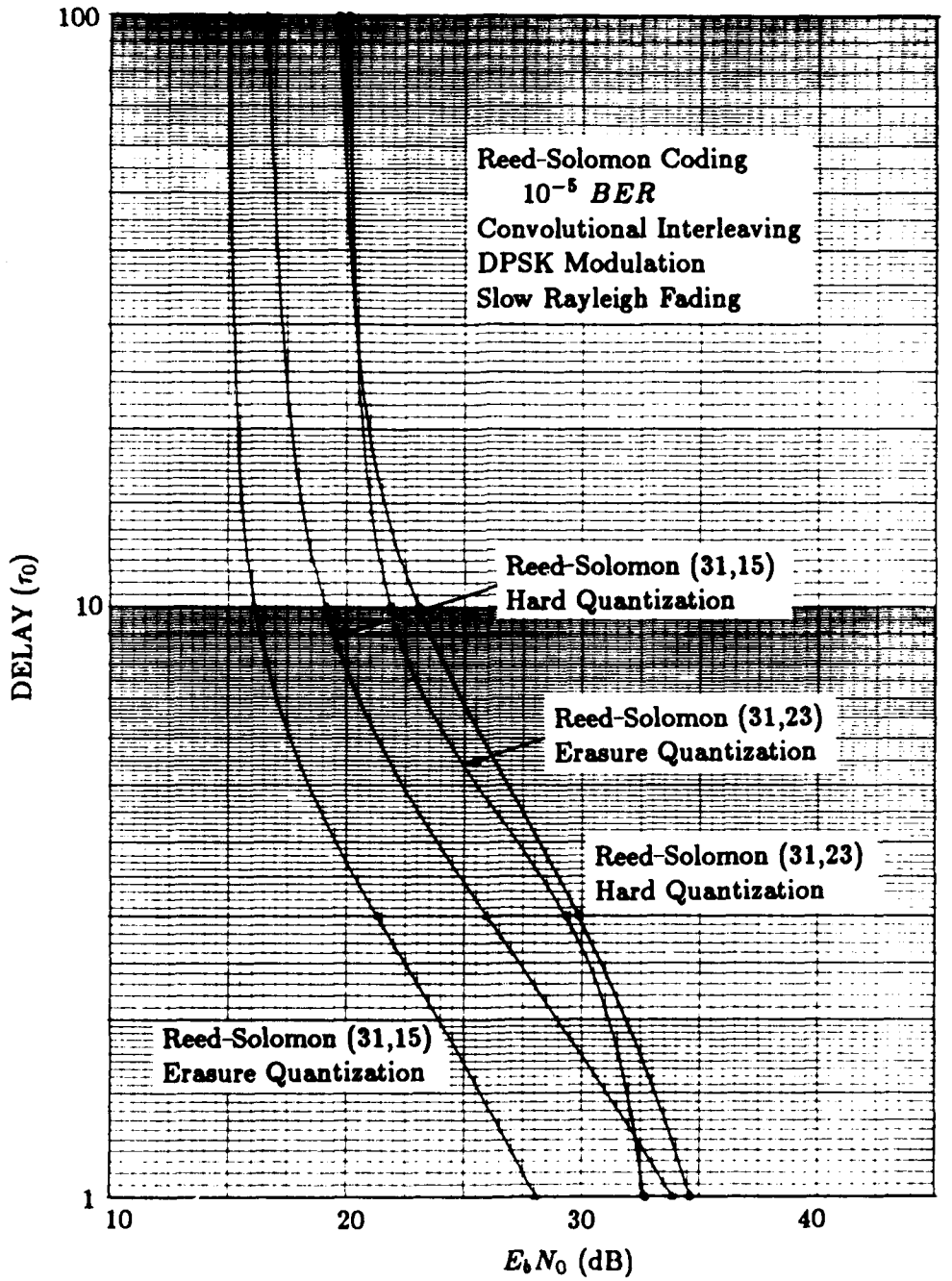


Figure 27. Interleaver delay requirements for Reed-Solomon (31,K) coding with 10^{-5} *BER*.

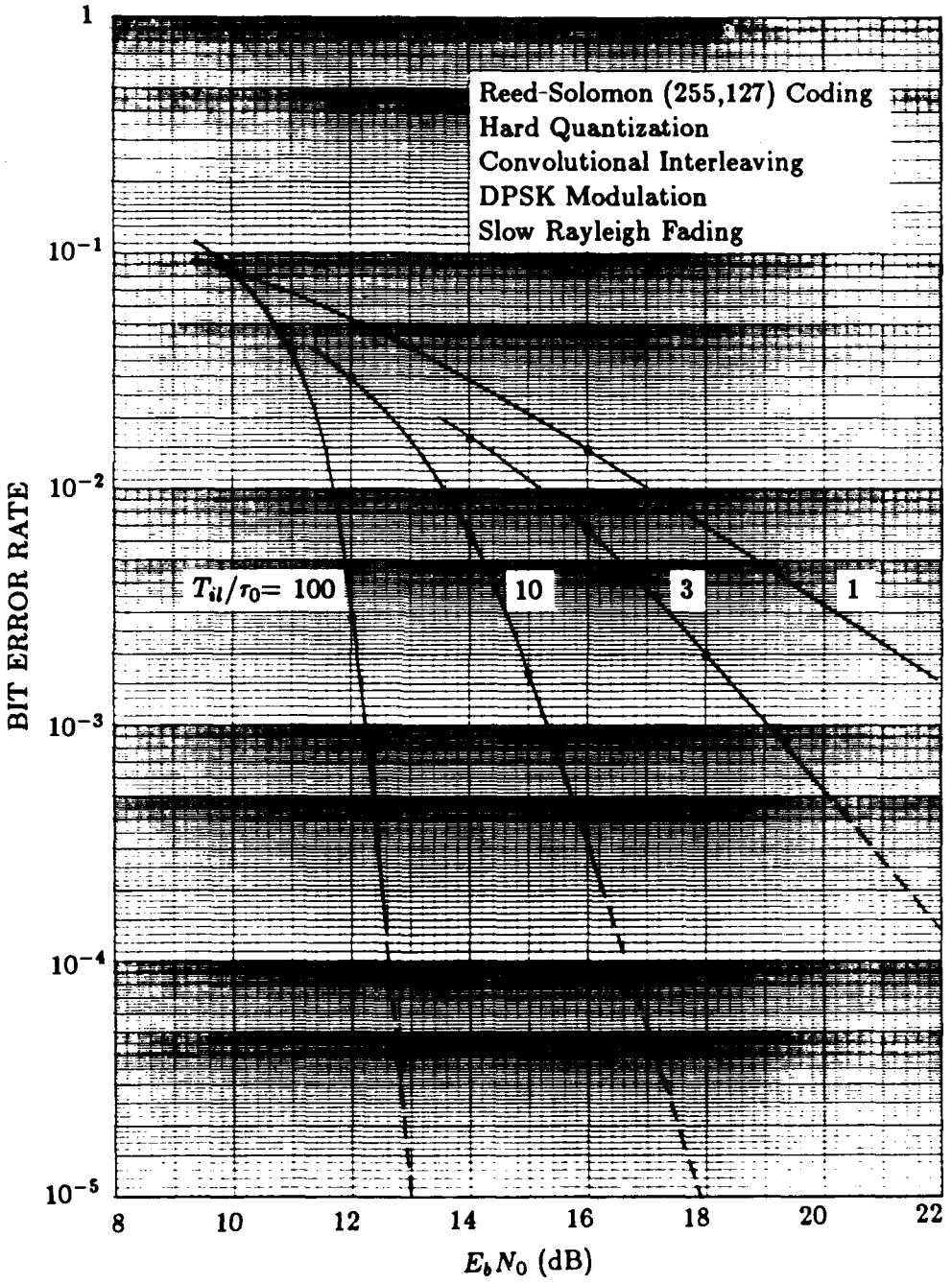


Figure 28. Code performance of Reed-Solomon (255,127) coding with hard quantization.

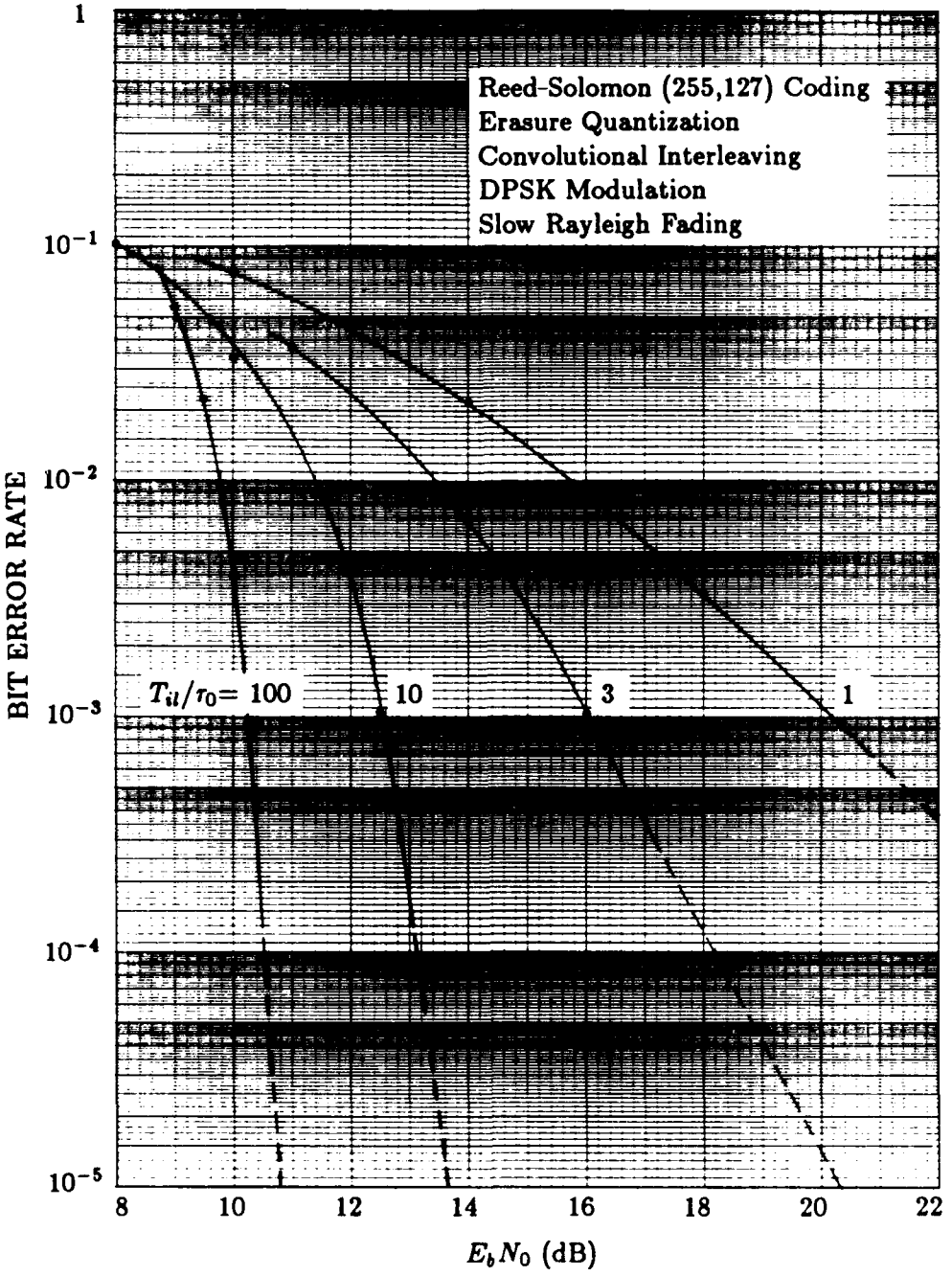


Figure 29. Code performance of Reed-Solomon (255,127) coding with erasure quantization.

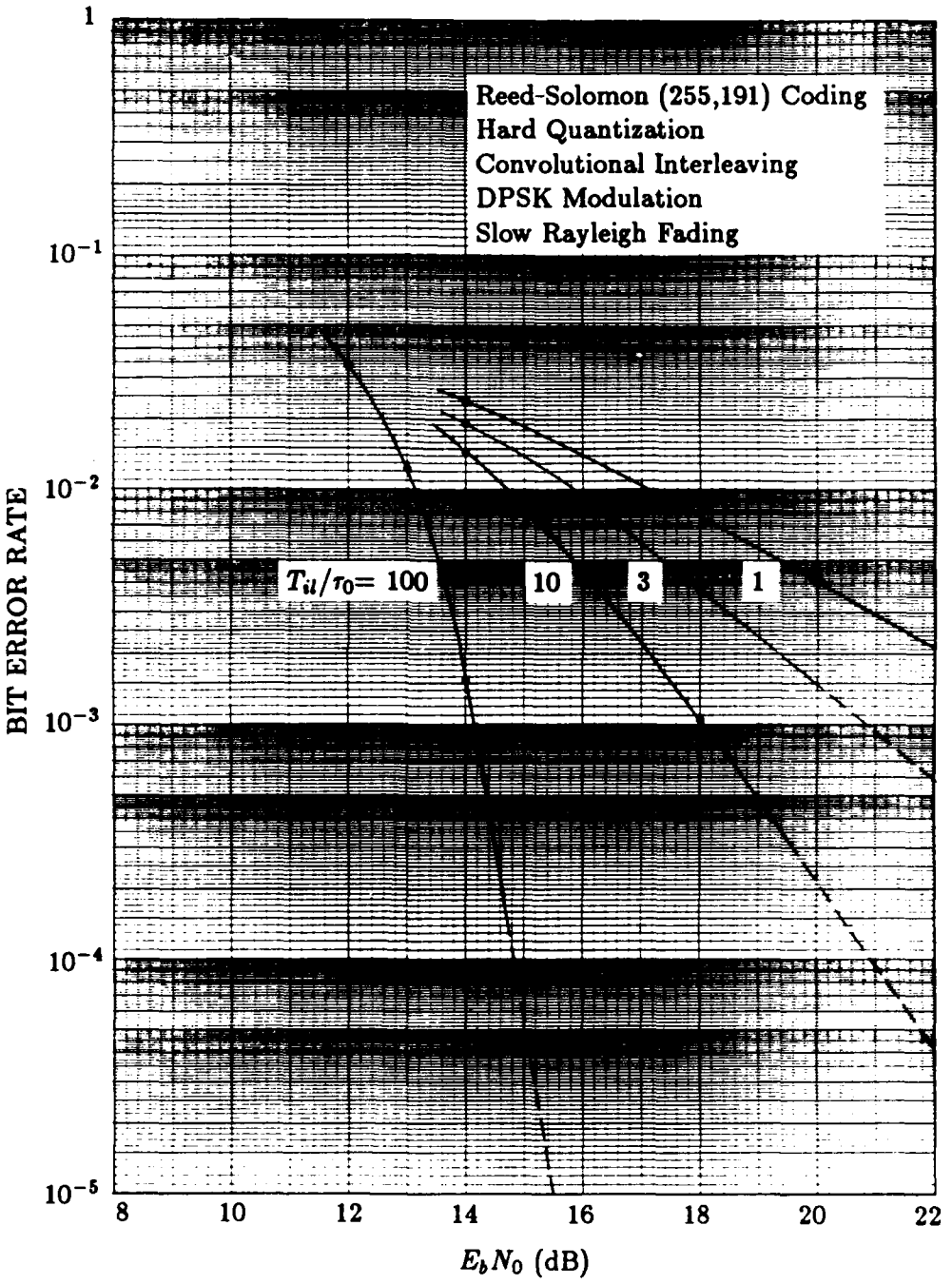


Figure 30. Code performance of Reed-Solomon (255,191) coding with hard quantization.

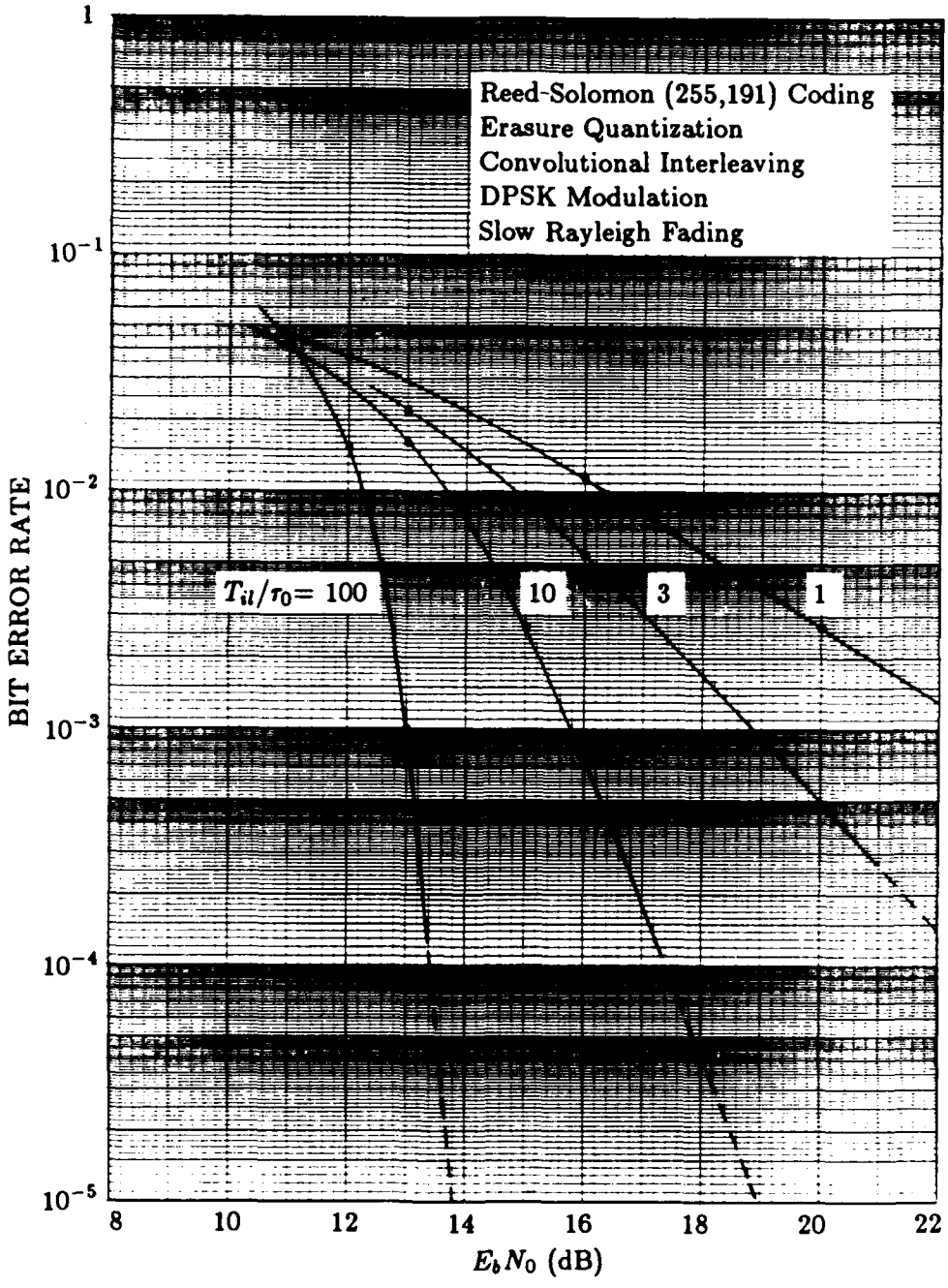


Figure 31. Code performance of Reed-Solomon (255,191) coding with erasure quantization.

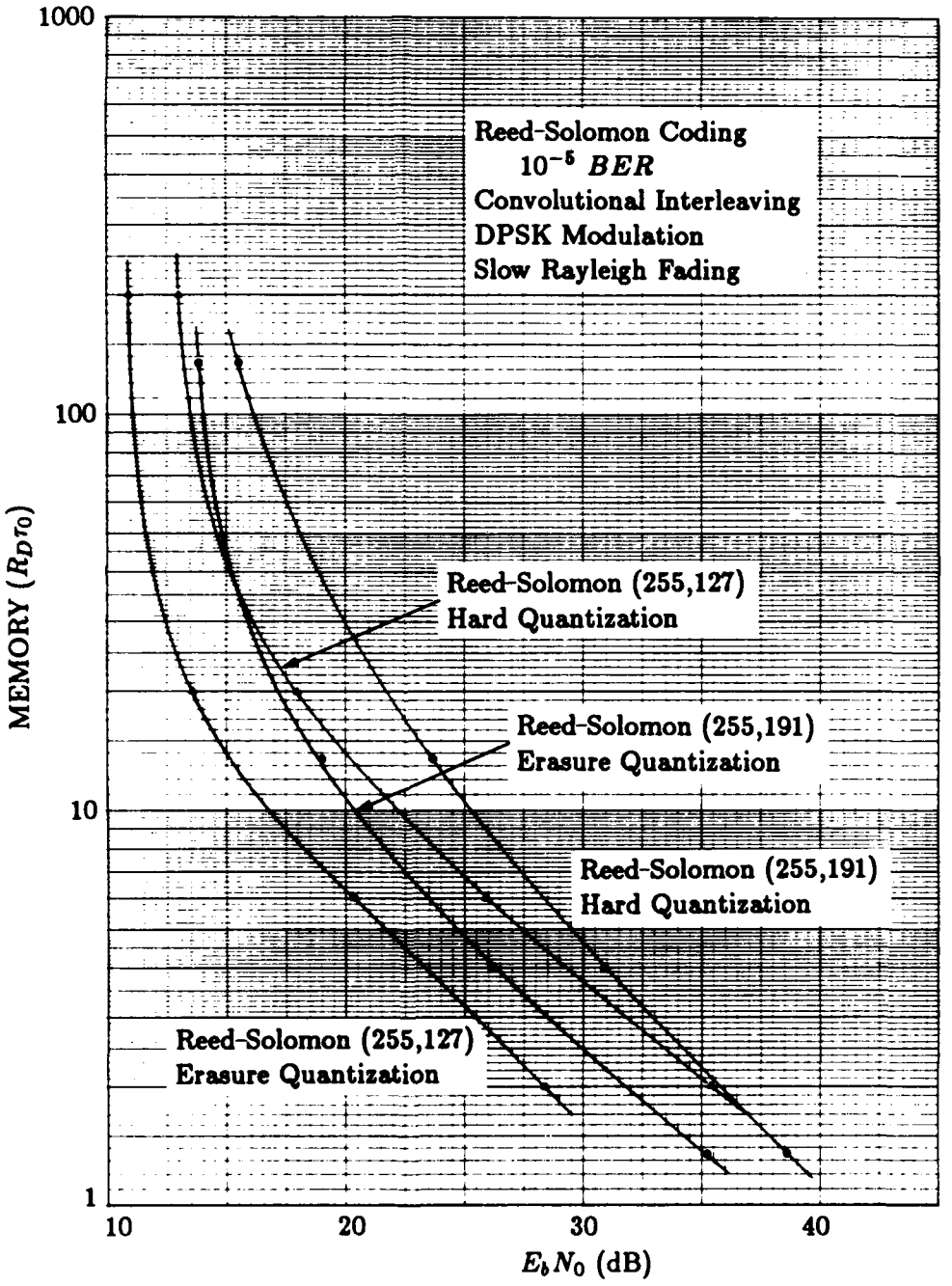


Figure 32. Interleaver memory requirements for Reed Solomon (255,K) coding with 10^{-5} *BER*.

The interleaving delay requirements for these code selections are shown next in Figure 33. These results indicate that Reed-Solomon (255,127) coding, with both erasure and hard quantization, has smaller delay requirements than those of Reed-Solomon (255,191) coding. The results also indicate that the delay requirements with erasure quantization are smaller than those with hard quantization for both (255,127) and (255,191) codes. Comparison of the memory and delay results indicate that Reed-Solomon (255,127) erasure-decision coding clearly has both the lowest memory and delay requirements.

Compared with the previous code selections, Reed-Solomon (255,127) erasure-decision coding has the lowest overall memory requirements. Reed-Solomon (255,127) erasure-decision coding also has the lowest delay requirements for E_b/N_0 below about 15 dB and the lowest delay requirements next to those of convolutional rate 1/3 coding for E_b/N_0 above 15 dB. Despite these requirements, however, Reed-Solomon (255,127) coding is not a candidate for code selection because of its code complexity.

Reed Solomon(255,127) coding was evaluated only to determine how an increase in block size reduces memory and delay requirements. This is seen by comparing Reed Solomon (255,127) and (31,15) code results. From these comparisons, Reed-Solomon (255,127) erasure-decision coding has much smaller interleaving memory and delay requirements than Reed-Solomon (31,15) erasure-decision coding for E_b/N_0 below about 18 dB. It has only marginally smaller requirements for E_b/N_0 larger than 18 dB.

5.4 ESTIMATION OF MEMORY REQUIREMENTS FOR CONCATENATED CODING.

After single-stage code performance evaluations, interleaving memory requirements were estimated for concatenated code combinations of the single-stage code selections. These estimates were made using the single-stage code results and were used to identify concatenated codes with potentially small memory requirements for code performance evaluations.

This subsection addresses the estimated memory requirements for concatenated codes and the procedure which was used to determine the estimates.

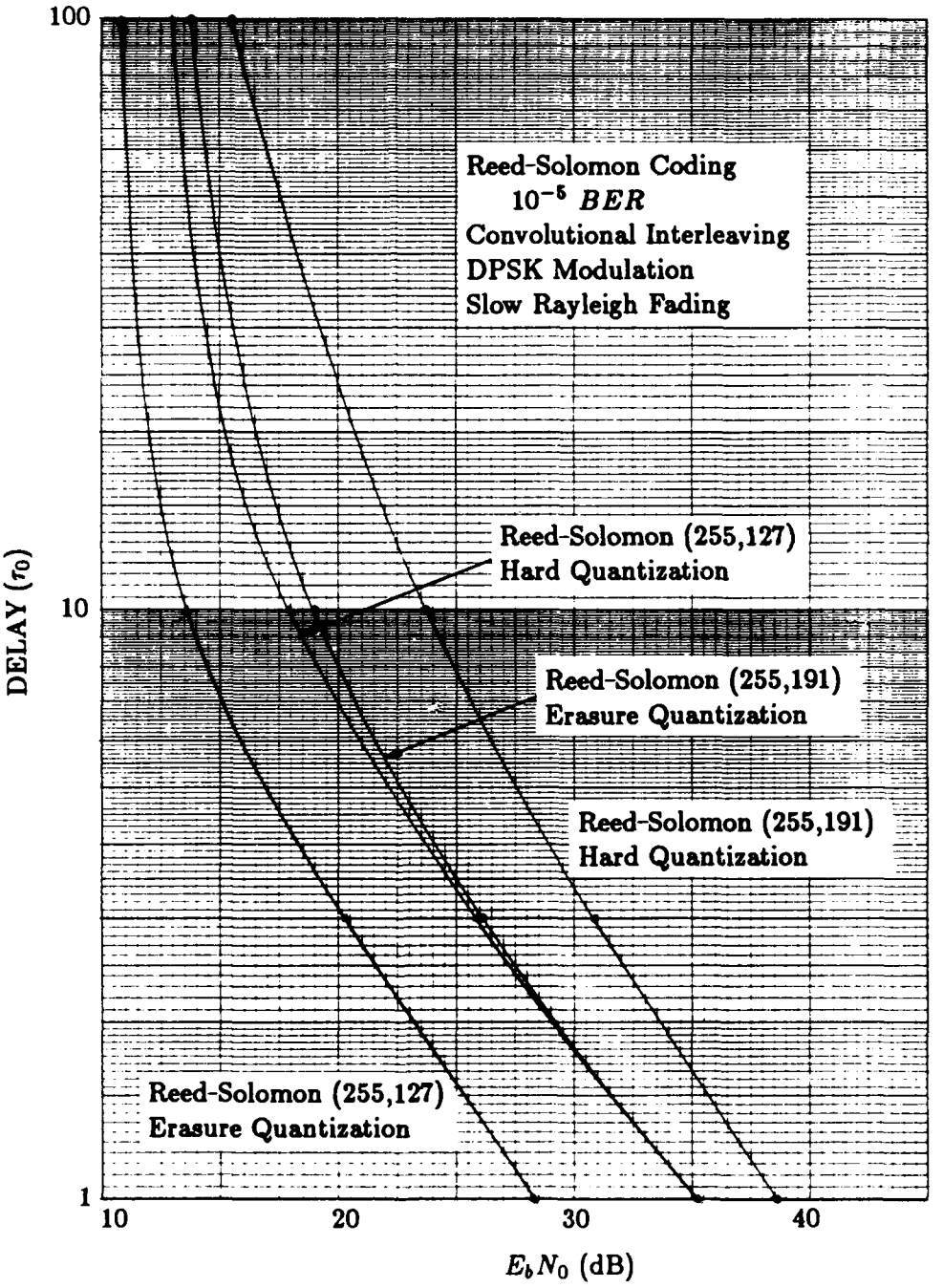


Figure 33. Interleaver delay requirements for Reed-Solomon (255,K) coding with 10^{-5} *BER*.

5.4.1 Estimation Procedure.

The interleaving memory requirements for concatenated coding were estimated using transfer functions (decoder input BER-to-output BER functions) for outer code selections and code performance curves for inner code selections. These estimates were conducted in a four step approach for each concatenated code candidate and T_{il}/τ_0 selection.

In the first step of this approach, the outer decoder input BER requirement for an outer decoded BER of 10^{-5} was determined. This involved finding the decoder input BER which corresponded to a $10-5$ decoded BER from the outer code's transfer function curves. The outer code transfer function curves were obtained from single-stage code evaluations and are presented in Appendix C.

In the next step, the inner code E_b/N_0 requirement for an inner decoded BER which equals the outer decoder input BER requirement was estimated. This involved finding the E_b/N_0 requirement that corresponded to the outer decoder input BER requirement in the inner code's performance curves. This value was chosen for the same T_{il}/τ_0 value used in the first step.

In the third step, the estimated E_b/N_0 requirement for the inner code was converted to an E_b/N_0 requirement at the user data rate to permit comparison of memory estimates for different codes at comparable user data rates. This involved scaling the inner code E_b/N_0 requirement by the inverse of the outer code rate to remove outer code redundancy.

In last step, the T_{il}/τ_0 selection used in steps 1 and 2 was converted to an interleaving memory requirement (using the formulas in Section 4) and plotted relative to the estimated E_b/N_0 requirement from the last step. The memory estimates were made for T_{il}/τ_0 of 100, 10, 3, and 1 for each concatenated code.

The accuracy of these estimates depended on the validity of assuming that the inner decoded BER can be equated with the outer decoder input BER in single-stage code results. The validity of this assumption in turn depended on the similarity between the second-order (or higher-order) statistics of the inner decoded bit errors and the outer decoder input bit errors.

Inner decoded BER and outer decoder input BER were equated for the same T_{il}/τ_0 selections to account for the same channel statistics. However, the second-order error statistics corresponding to these values differed because the decoder input errors were demodulated and not decoded in the single-stage code evaluations. De-

coded errors are burstier than demodulated errors, so the outer code BER estimates did not include the added error burstyness from inner decoded errors[†].

As a result of this disparity, the estimated interleaving memory requirements have lower E_b/N_0 requirements than those of actual memory requirements. The magnitude of this difference depends on the burstyness of the errors received at the outer decoder as well as the correction characteristics of the outer code.

Code performance comparisons of estimated and evaluated memory requirements (which followed the evaluations) indicated that the E_b/N_0 difference between these requirements increases with decreases in T_{il}/τ_0 and outer code rate. The reason for these characteristics in the first case is likely because of the longer error bursts that accompany reductions in T_{il}/τ_0. Longer error bursts cause a greater disparity between decoded and demodulated error burstyness for the same BER.

This disparity is exemplified in the difference between the E_b/N_0 requirements for estimations and evaluations of outer Reed-Solomon rate 3/4 coding with inner convolutional rate 1/2 coding. With T_{il}/τ_0 of 10, 3, and 1, the E_b/N_0 difference at 10^{-5} BER is approximately 3, 8, and 13 dB respectively. However, with T_{il}/τ_0 of 100, the difference in requirements is zero because the outer decoder receives random errors regardless of whether demodulation or inner decoding precede the outer decoder[‡].

With decreases in outer code rate, the disparity between estimated and evaluated memory requirements is less obvious and less significant than the disparity caused by decreases in T_{il}/τ_0. This disparity in the requirements is likely the result of steeper transfer function curves. With steeper transfer functions, lower rate codes have a larger increase in decoded BER from bursty decoder input errors than higher rate codes.

5.4.2 Memory Estimates.

Prior to selecting concatenated codes for evaluation, the memory efficiency of concatenated codes was determined by comparing their estimated memory requirements with the evaluated requirements of single-stage codes. The single-stage codes

[†]Decoded errors are burstier for the same T_{il}/τ_0 and BER values because coding corrects random and short error bursts and, with convolutional coding, causes longer bursts of errors when it incorrectly decodes. This has the net effect of increasing the error correlation at decoder output and causing longer error bursts than demodulated errors at the same BER.

[‡]In this case inner interleaving randomizes signal fading errors and outer interleaving randomizes naturally-bursty errors from inner decoding decisions.

selected for these comparisons had good memory efficiency and comparable complexity to the concatenated codes. Concatenated codes with memory estimates which were smaller than those of the single-stage codes were chosen for code performance evaluation and concatenated codes with memory estimates which were larger than those of the single-stage codes were not selected for evaluation[§].

The concatenated codes that were selected for memory comparison comprised many combinations of single-stage codes. Outer concatenated code choices included all rate 3/4 Reed-Solomon code selections and some rate 1/2 Reed-Solomon code selections. The inner concatenated code choices included all lower rate convolutional and Reed-Solomon code selections. Each type of quantization selected for the single-stage code evaluations were also selected for the inner and outer concatenated codes. However, estimates of memory requirements for with outer Reed-Solomon concatenated codes with erasure quantization were ignored after discovering that their estimated performance was highly inaccurate[¶].

The single-stage codes that were selected for comparison with the concatenated codes are Reed-Solomon (255,127) erasure-decision coding and Reed-Solomon (31,15) erasure-decision coding. Both codes have relatively small memory requirements but, because Reed-Solomon (255,127) coding has impractical implementation requirements, it served as reference only for the concatenated codes with Reed-Solomon (255,K) coding. Reed-Solomon (31,15) coding served as reference for the practical concatenated codes, without Reed-Solomon (255,K) coding.

From the memory assessments, the following codes were found to have the lowest memory requirements: outer rate 3/4 Reed-Solomon concatenated coding with inner convolutional rate 1/2 or rate 1/3, erasure- and soft-decision coding and inner Reed-Solomon (31,15) erasure-decision coding. Only the concatenated codes with outer Reed-Solomon (31,23) coding were found to have significantly lower memory requirements than their the single-stage code reference, however.

The estimated memory requirements for outer Reed-Solomon (31,K) concatenated coding with inner convolutional rate 1/2, erasure-decision coding are shown

[§]This selection was made because the actual memory requirements of concatenated codes with estimated requirements which exceed the evaluated memory requirements of reference codes were guaranteed to be larger the reference code requirements. This is because estimated memory are smaller than actual memory requirements (see the previous subsection).

[¶]This inaccuracy was the result of overly optimistic estimates of outer code performance with erasure quantization. The inaccuracy resulted because the memory savings from erasure versus hard quantization in single-stage codes is much larger than in outer concatenated codes. Outer erasure decision coding was found only to have about 1/4 dB improvement over outer hard-decision coding.

in Figure 34. These estimates indicate that the outer Reed-Solomon (31,23) concatenated code has substantially smaller memory requirements than those of the outer Reed-Solomon (31,15) concatenated code. However, the estimates also indicate that both code selections have potentially smaller memory requirements those of Reed-Solomon (31,15) erasure-decision coding for E_b/N_0 below about 15 dB.

The estimated memory requirements for outer Reed-Solomon (31,23) concatenated coding with inner convolutional rate 1/3, erasure-decision coding and inner Reed-Solomon (31,15) erasure-decision coding are shown next in Figure 35. These estimates indicate that the memory requirements for the inner convolutional rate 1/3 concatenated code are nearly identical to those of inner convolutional rate 1/2 concatenated code with Reed-Solomon (31,23) outer coding. The estimates also indicate that the memory requirements for the Reed-Solomon (31,15) concatenated code are smaller than those of the inner convolutional concatenated codes for E_b/N_0 above about 12 dB. However, the estimates indicate that both code selections have potentially smaller memory requirements than those of Reed-Solomon (31,15) erasure-decision coding for E_b/N_0 below about 16 dB.

Based on these estimates, outer Reed-Solomon (31,23) concatenated coding with inner Reed-Solomon (31,15) erasure-decision coding and inner convolutional rate 1/2, erasure-decision coding were selected for computer-evaluation. Unfortunately, time did not permit evaluation of outer Reed-Solomon (31,23) concatenated coding with inner convolutional rate 1/3, erasure-decision coding even though this was also estimated as having potentially good memory savings. Concatenated codes with outer Reed-Solomon (255,K) coding were mistakenly evaluated instead because their memory requirements were initially underestimated. Nonetheless, all of the code evaluations provided insight toward better understanding concatenated code interleaving requirements.

5.5 CONCATENATED CODING RESULTS.

Concatenated code evaluations included outer Reed-Solomon (31,K) and (255,K) concatenated codes with numerous inner convolutional and Reed-Solomon codes. From these evaluations, the most memory and delay efficient codes were found to be outer Reed-Solomon (31,23) coding with inner convolutional rate 1/2, erasure-decision coding and inner Reed-Solomon (31,15) erasure-decision coding. The other code selections generally had higher memory and delay requirements and were inefficient in comparison with single-stage code references of comparable complexity.

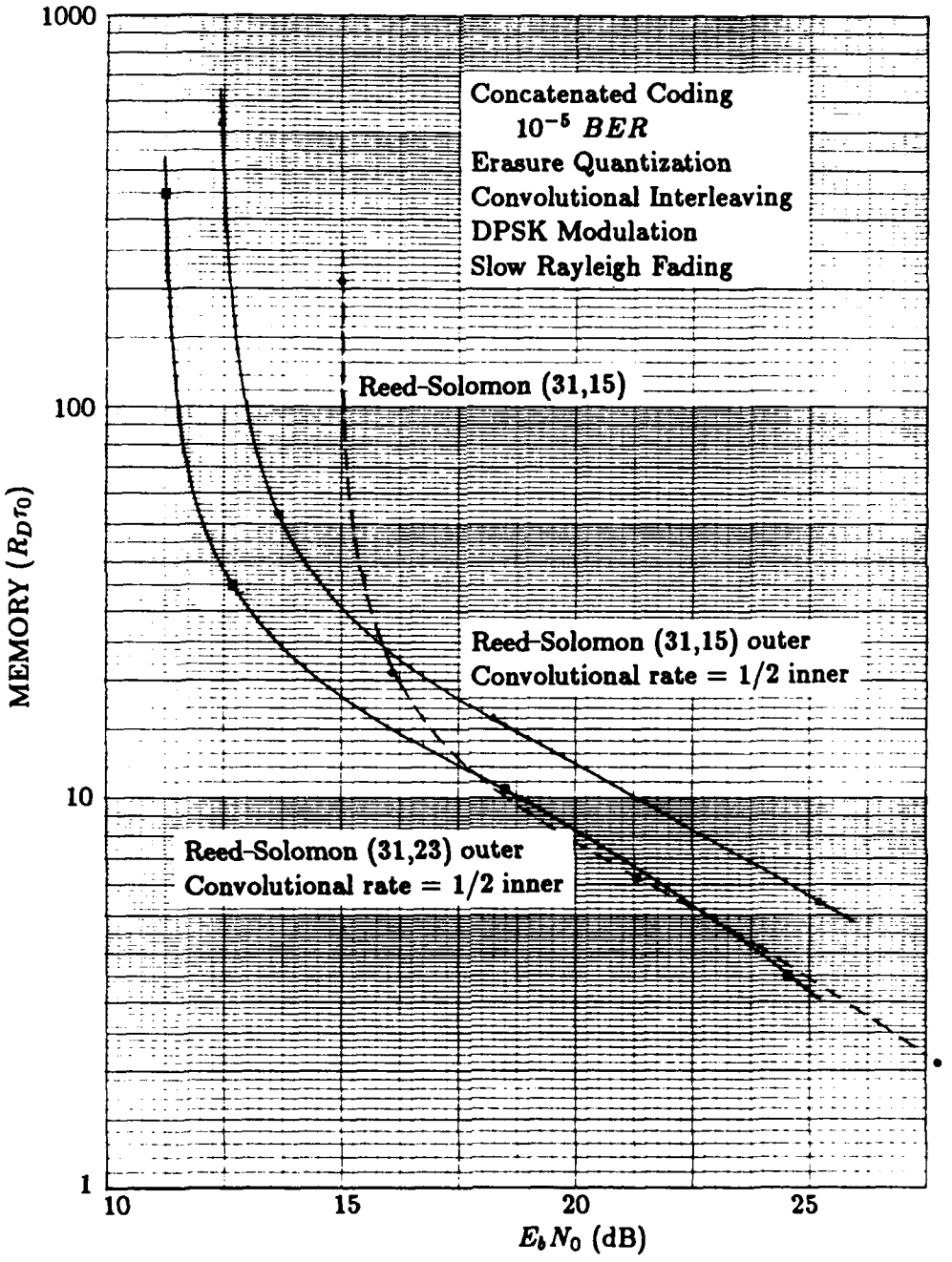


Figure 34. Estimated interleaving memory requirements for concatenated Reed-Solomon (31,K) outer coding and convolutional rate 1/2 inner coding with 10^{-5} *BER*.

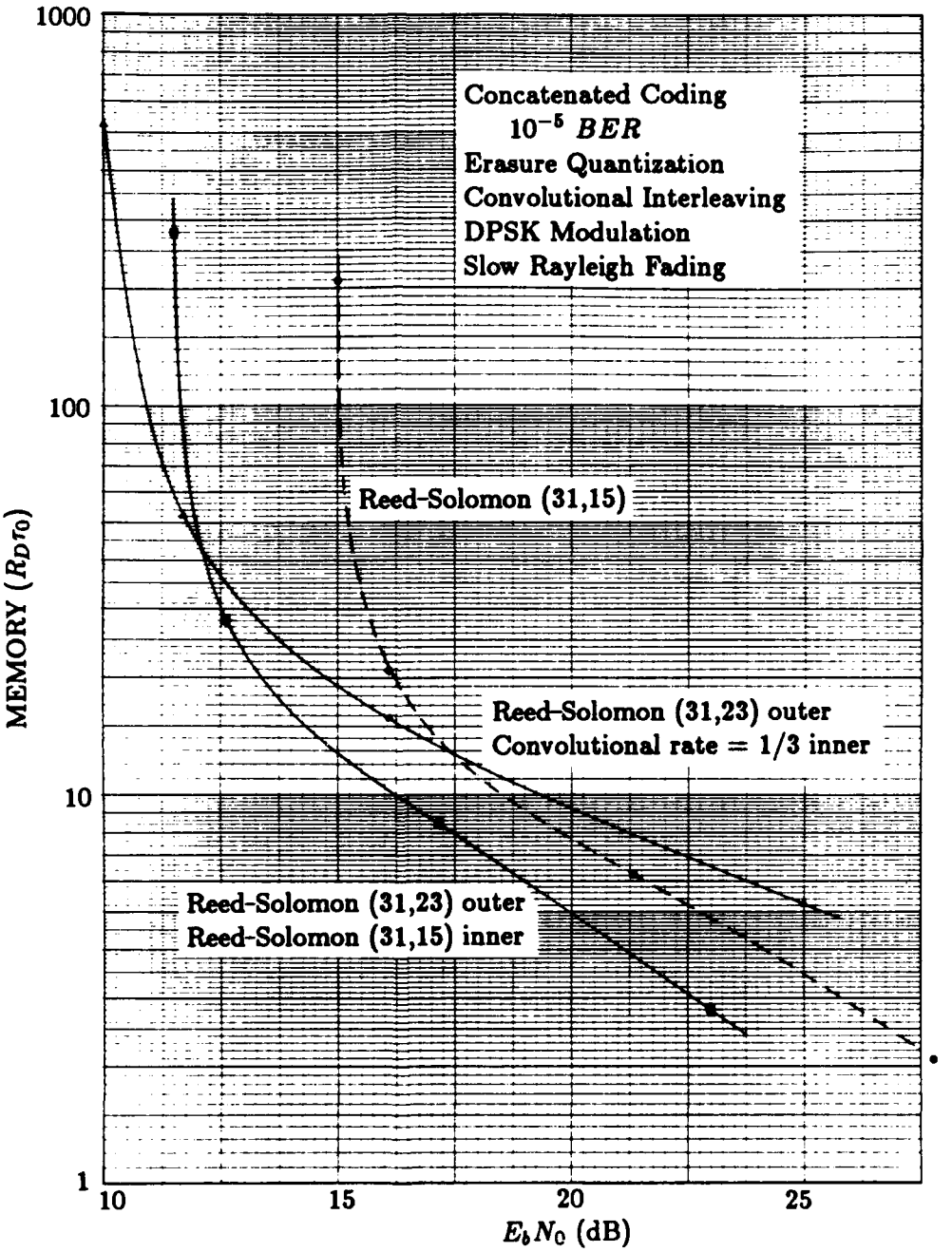


Figure 35. Estimated interleaving memory requirements for concatenated Reed-Solomon (31,23) outer coding and convolutional rate 1/3 and Reed-Solomon (31,15) inner coding with 10^{-5} *BER*.

The concatenated code evaluations confirmed that outer rate 3/4 coding with lower rate inner coding are the best code choices. The evaluations also confirmed that erasure quantization is the best quantization choice. The evaluations indicated that inner coding is sensitive to quantization selections but outer coding is relatively insensitive to quantization selections. In the latter case, the code evaluations conducted with outer Reed-Solomon erasure- and hard-decision coding and inner Reed-Solomon erasure-decision coding indicated that only about 1/4 dB is gained from outer erasure quantization.

To address only the most useful concatenated code selections, this subsection presents the code performance and interleaving requirements of outer Reed-Solomon (31,23) concatenated coding with inner convolutional rate 1/2, erasure-decision coding and inner Reed-Solomon (31,15) erasure-decision coding. The results are presented in the same type of plots as the single-stage code results. However, these plots include Reed-Solomon (31,15) erasure-decision coding for reference.

5.5.1 Concatenated Coding with Reed-Solomon (31,23) Outer Coding.

The code performance results are shown in Figure 36 for T_{il}/τ_0 of 10 and in Figure 37 for T_{il}/τ_0 of 3. The results in Figure 36 indicate that both concatenated codes have E_b/N_0 requirements that are approximately 1 dB smaller than those of Reed-Solomon (31,15) erasure-decision coding at a BER of 10^{-5}. The results in Figure 37 indicate that the inner Reed-Solomon (31,15) concatenated code has similar E_b/N_0 requirements to Reed-Solomon (31,15) erasure-decision coding but the inner convolutional rate 1/2 concatenated code has substantially larger requirements. Fortunately, with T_{il}/τ_0 of 100, the code performance for the concatenated codes is substantially better than that of Reed-Solomon (31,15) erasure-decision coding, as established in the previous subsection*.

These results translate to the interleaving memory requirements shown in Figure 38. The results of Figure 38 indicate that both concatenated codes have smaller memory requirements than those of Reed-Solomon (31,15), erasure-decision coding for E_b/N_0 below about 16 dB and both concatenated codes have larger requirements for E_b/N_0 above 16 dB.

The interleaving delay requirements for the concatenated codes are shown next in Figure 39. These results indicate similar characteristics to the memory require-

*This result was estimated but it is considered accurate because, with T_{il}/τ_0 of 100, the difference between estimated and evaluated code performance is approximately zero.

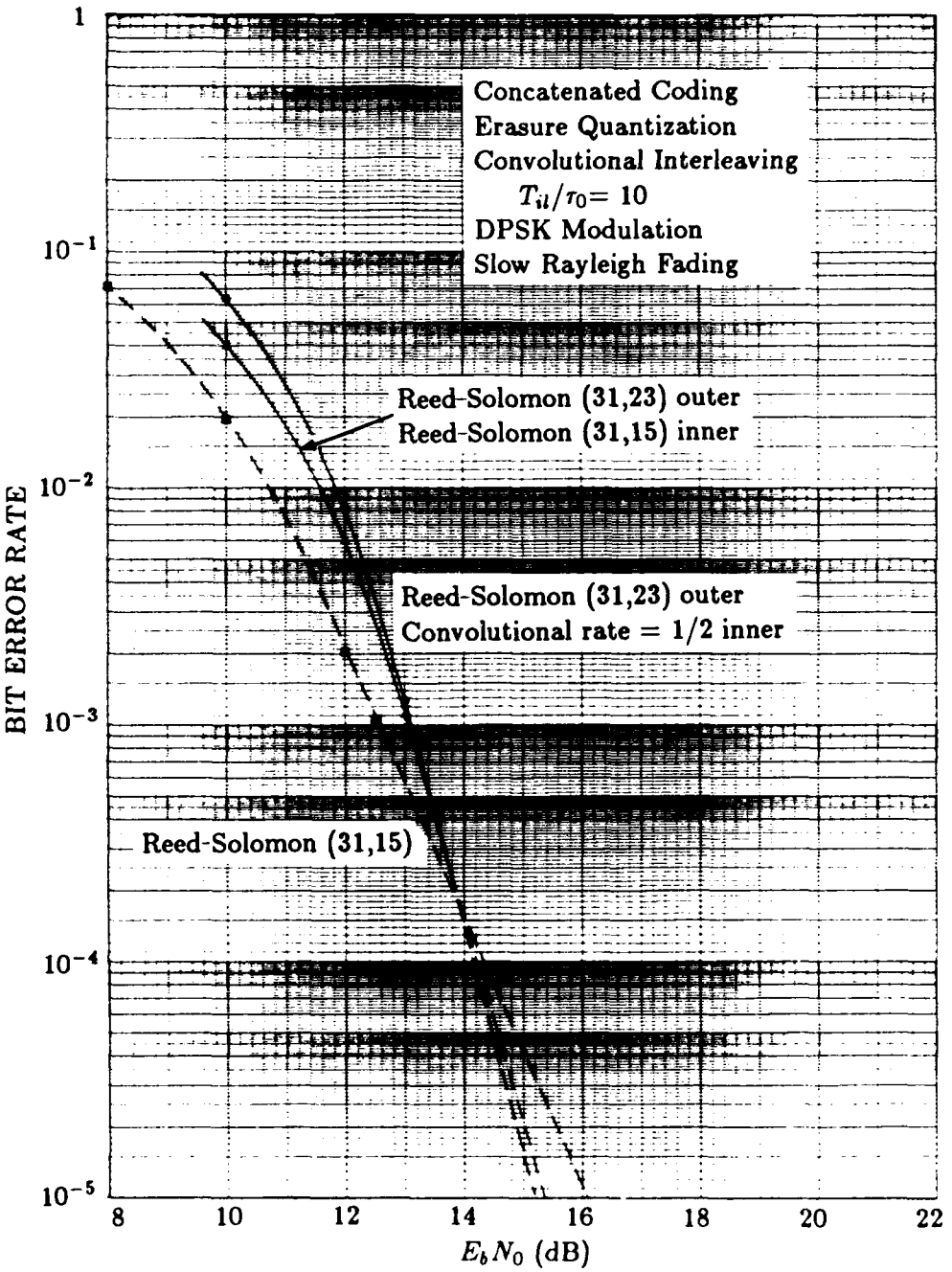


Figure 36. Code performance of concatenated Reed-Solomon (31,23) outer coding and convolutional rate 1/2 and Reed-Solomon (31,15) inner coding with T_{il}/τ_0 of 10.

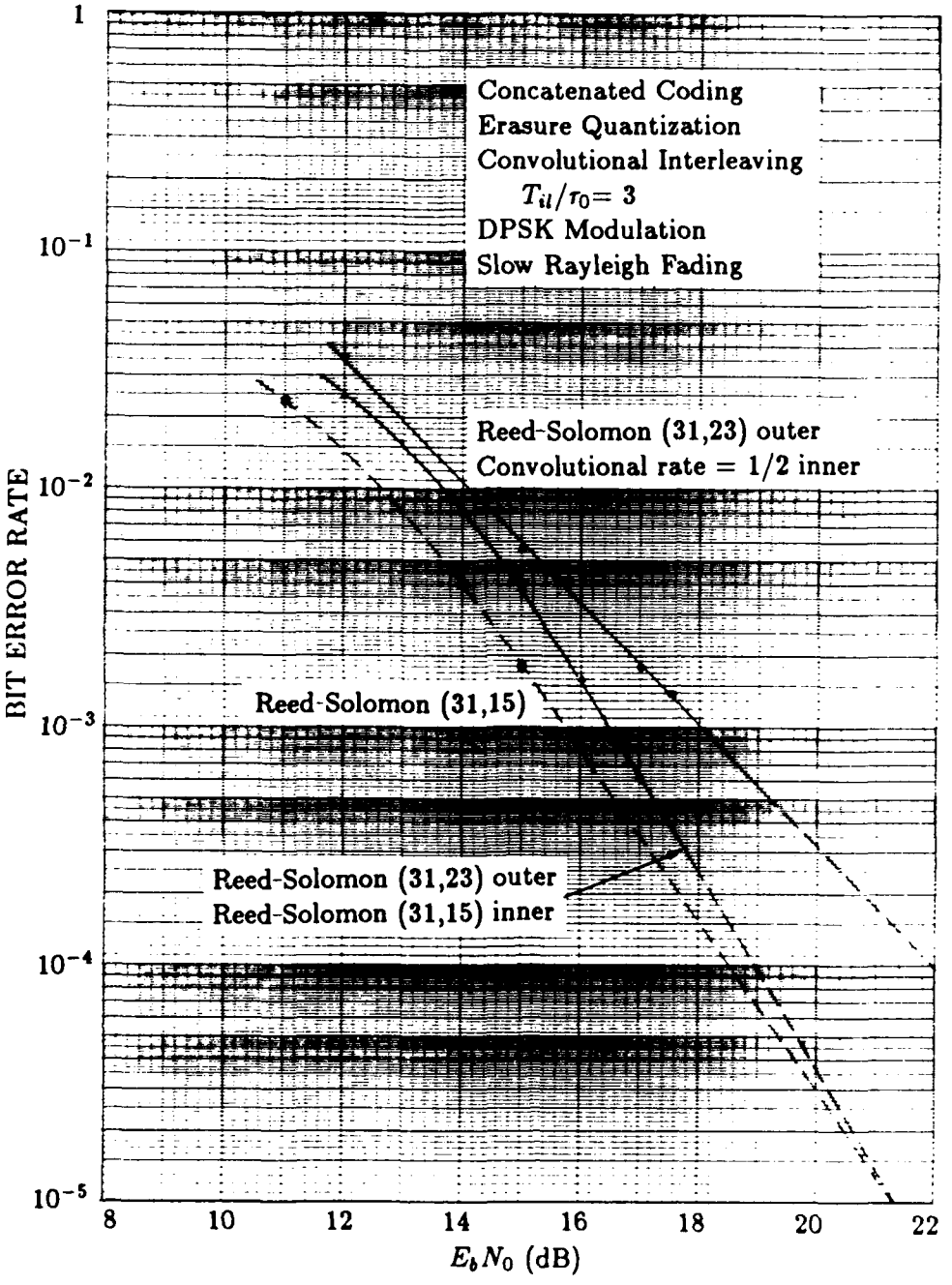


Figure 37. Code performance of concatenated Reed-Solomon (31,23) outer coding and convolutional rate 1/2 and Reed-Solomon (31,15) inner coding with T_{il}/τ_0 of 3.

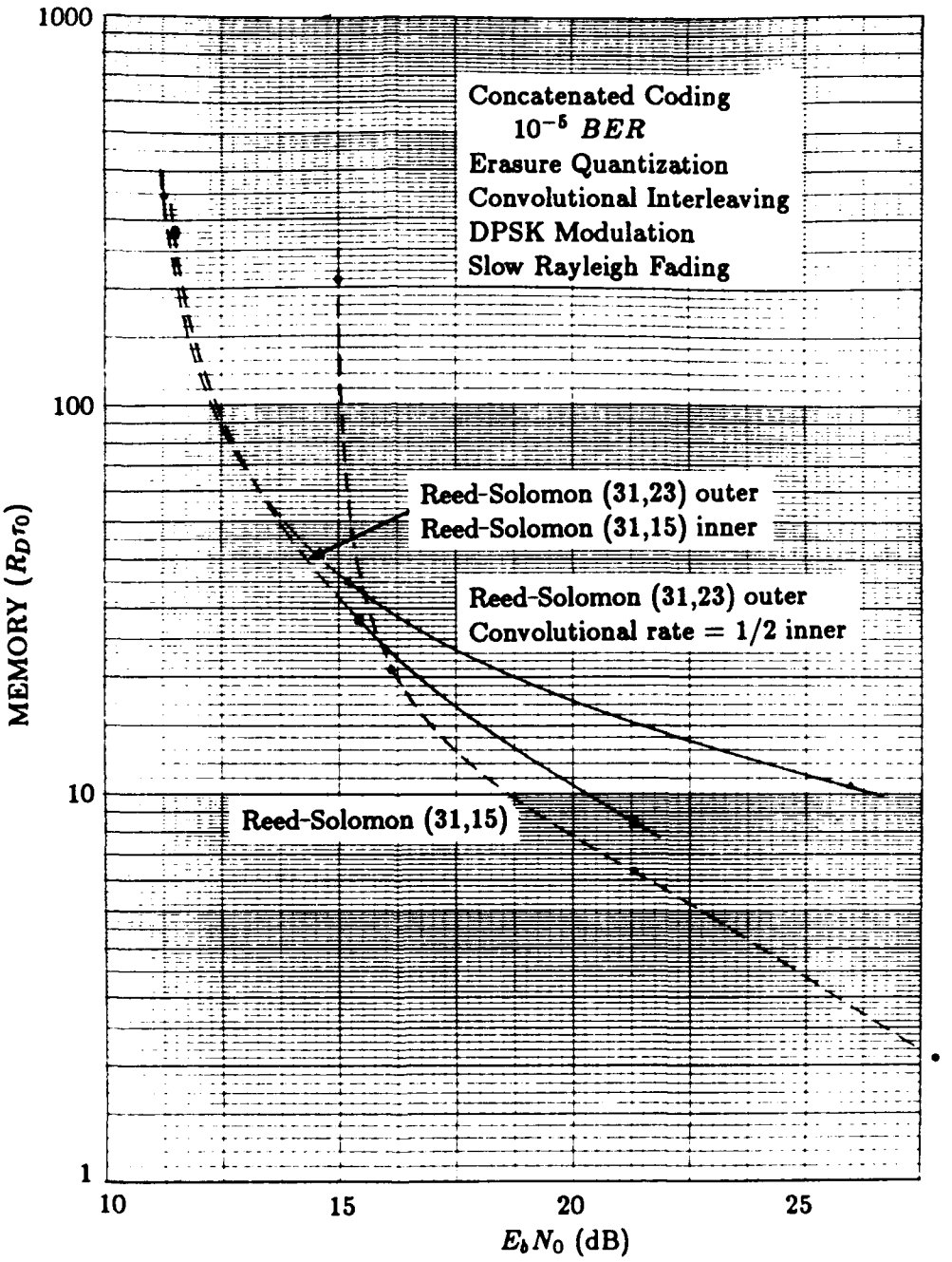


Figure 38. Interleaver memory requirements for concatenated Reed-Solomon (31,23) outer coding and convolutional rate 1/2 and Reed-Solomon (31,15) inner coding with 10^{-5} BER.

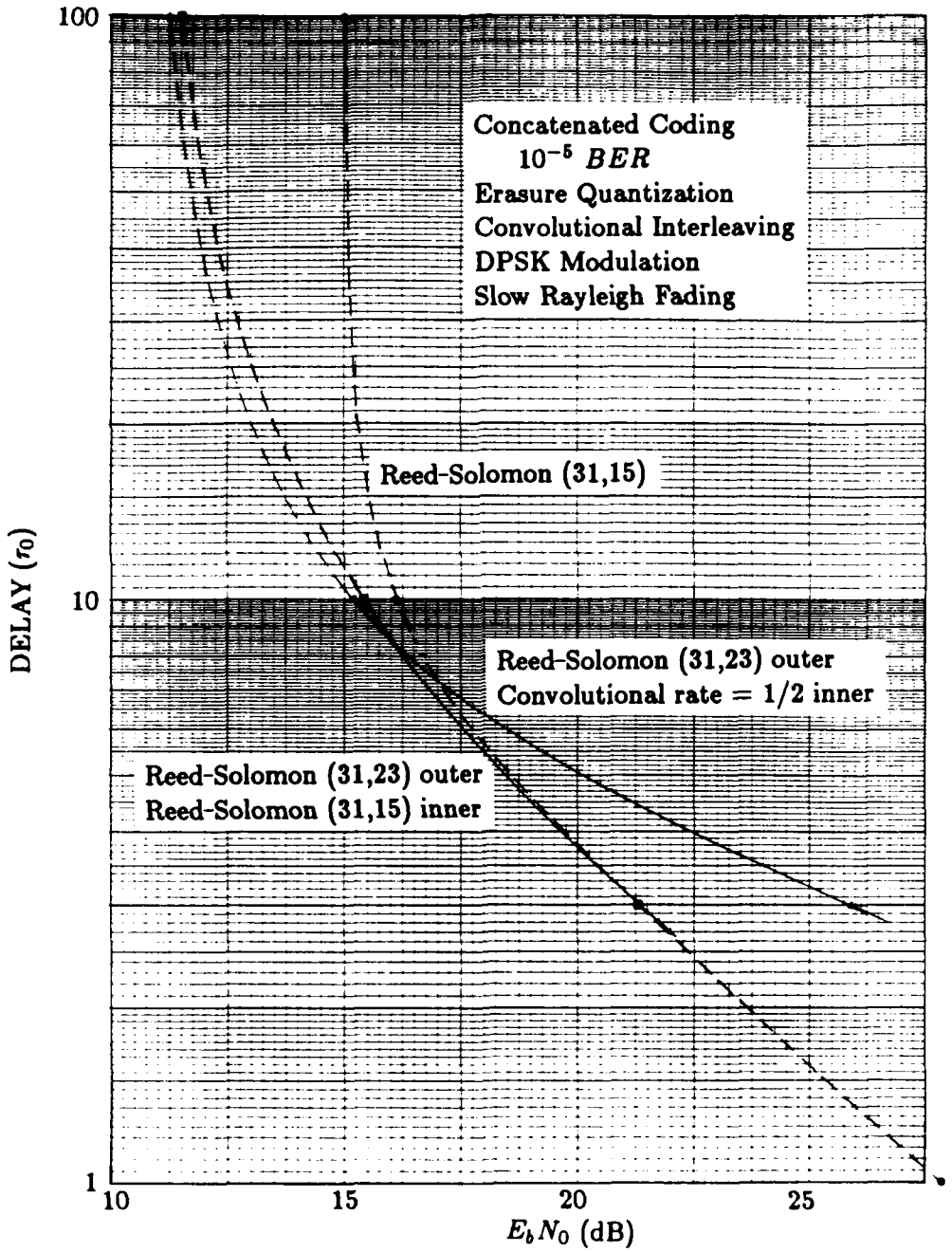


Figure 39. Interleaver delay requirements for concatenated Reed-Solomon (31,23) outer coding and convolutional rate 1/2 and Reed-Solomon (31,15) inner coding with 10^{-5} *BER*.

ments. The concatenated codes have smaller delay requirements than those of Reed-Solomon erasure-decision coding for E_b/N_0 below about 16 dB and they have larger requirements for E_b/N_0 above 16 dB. The results also indicate that the inner convolutional rate 1/2 concatenated code has approximately 1 dB smaller E_b/N_0 requirements than those of the inner Reed-Solomon (31,15) concatenated code at comparable delay and at E_b/N_0 below about 15 dB.

These results indicate that the concatenated codes have better interleaving memory and delay efficiency than Reed-Solomon (31,15) erasure-decision coding for E_b/N_0 below about 16 dB. However, the concatenated code requirements have to be compared with convolutional rate 1/3, erasure-decision coding to comprehensively assess the concatenated codes' interleaving efficiency.

Compared with convolutional rate 1/3 erasure-decision coding, the concatenated codes have smaller memory requirements for E_b/N_0 below about 14 dB. The inner convolutional rate 1/2 concatenated code has delay requirements which are smaller than those of convolutional rate 1/3 erasure-decision coding for E_b/N_0 below about 14 dB. The inner Reed-Solomon (31,15) concatenated code has comparable delay requirements to those of convolutional rate 1/3 erasure-decision coding for E_b/N_0 below 14 dB.

These comparisons indicate that both concatenated codes are good code choices for memory and delay efficient interleaving with E_b/N_0 requirements smaller than about 14 dB. However, the best code selection depends on E_b/N_0 and interleaving requirements specifications. This is addressed next in Section 6.

SECTION 6

SUMMARY AND RECOMMENDATIONS

This section presents a summary of the evaluated codes with the smallest interleaving requirements and a discussion of how to select the best codes for different E_b/N_0, memory, and delay specifications. This section also presents recommendations for further investigations of coding techniques with interleaving memory efficiency.

6.1 SUMMARY OF RESULTS.

Before addressing the results, it is important to note that the results for Reed-Solomon coding were evaluated with a simplified code model. As a consequence, these results are less accurate than simulated results. However, based on comparisons with simulated Reed-Solomon results (see Section 3.2), the Reed-Solomon results presented in this summary are believed to be very accurate, with less than 1/4 dB inaccuracy.

Of the practical codes evaluated in this investigation, three code selections were found to have the smallest interleaving requirements: outer Reed-Solomon (31,23) concatenated coding with inner convolutional rate 1/2, erasure-decision coding has the lowest memory and delay requirements for E_b/N_0 below about 14 dB, convolutional rate 1/3 coding has the lowest delay requirements and low memory requirements for E_b/N_0 above about 14 dB, and Reed-Solomon (31,15) erasure-decision coding has the lowest memory requirements and low delay requirements for E_b/N_0 above about 16 dB.

These codes do not have the overall lowest interleaving requirements, however. Reed-Solomon (255,127) erasure-decision coding has the lowest overall interleaving memory requirements and low overall delay requirements. Nonetheless, Reed-Solomon (255,127) coding is not a practical code candidate. It was evaluated only to determine how increases in block size reduce interleaving requirements.

Plots of the interleaving memory and delay requirements for these codes are shown in Figures 40, 41, 42, 43, 44, and 45. These plots include convolutional rate 1/2, soft-decision coding as reference. The concatenated and convolutional coding plots include additional codes for comparison.

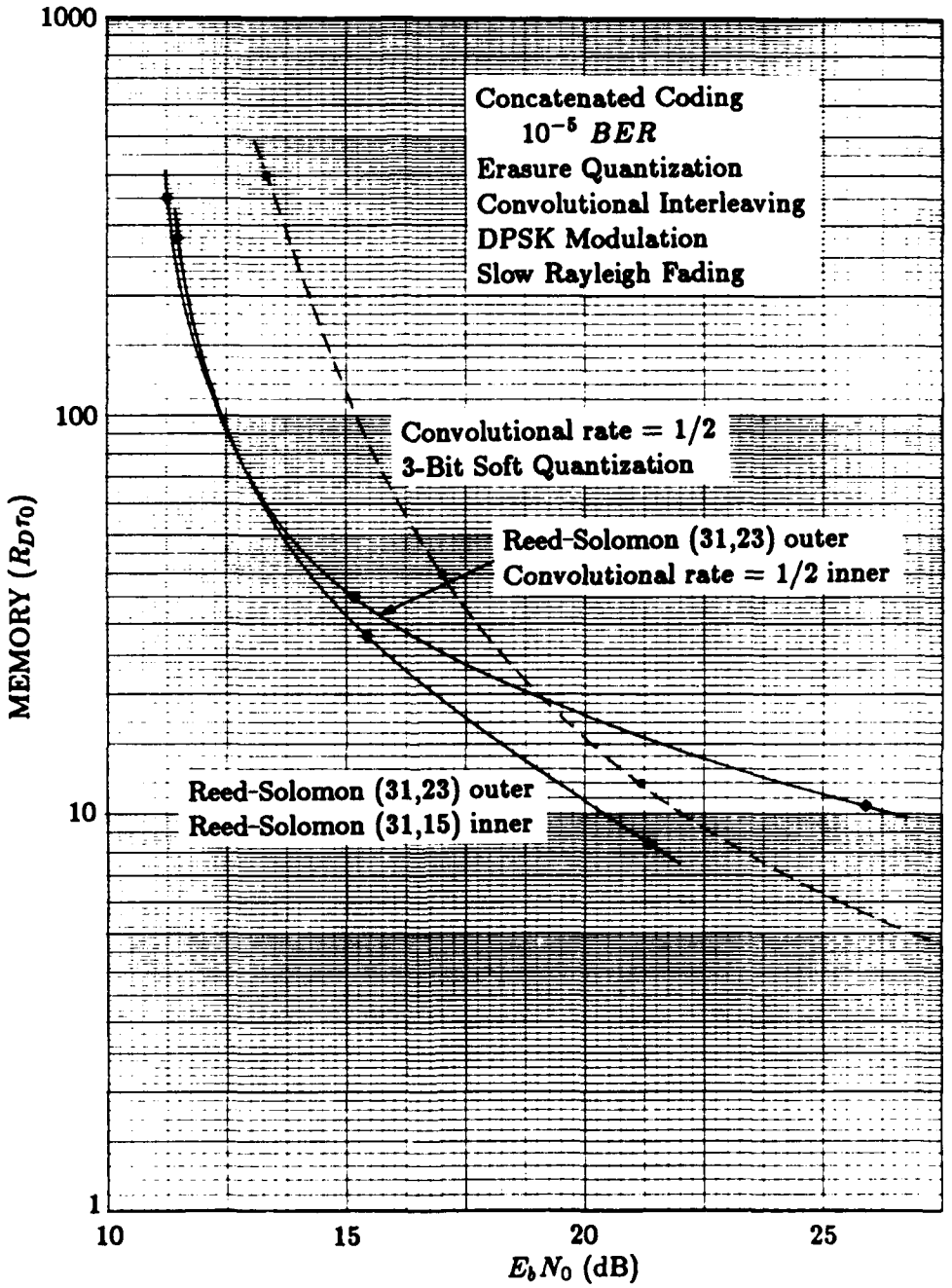


Figure 40. Interleaving memory requirements for efficient concatenated codes with 10^{-5} BER.

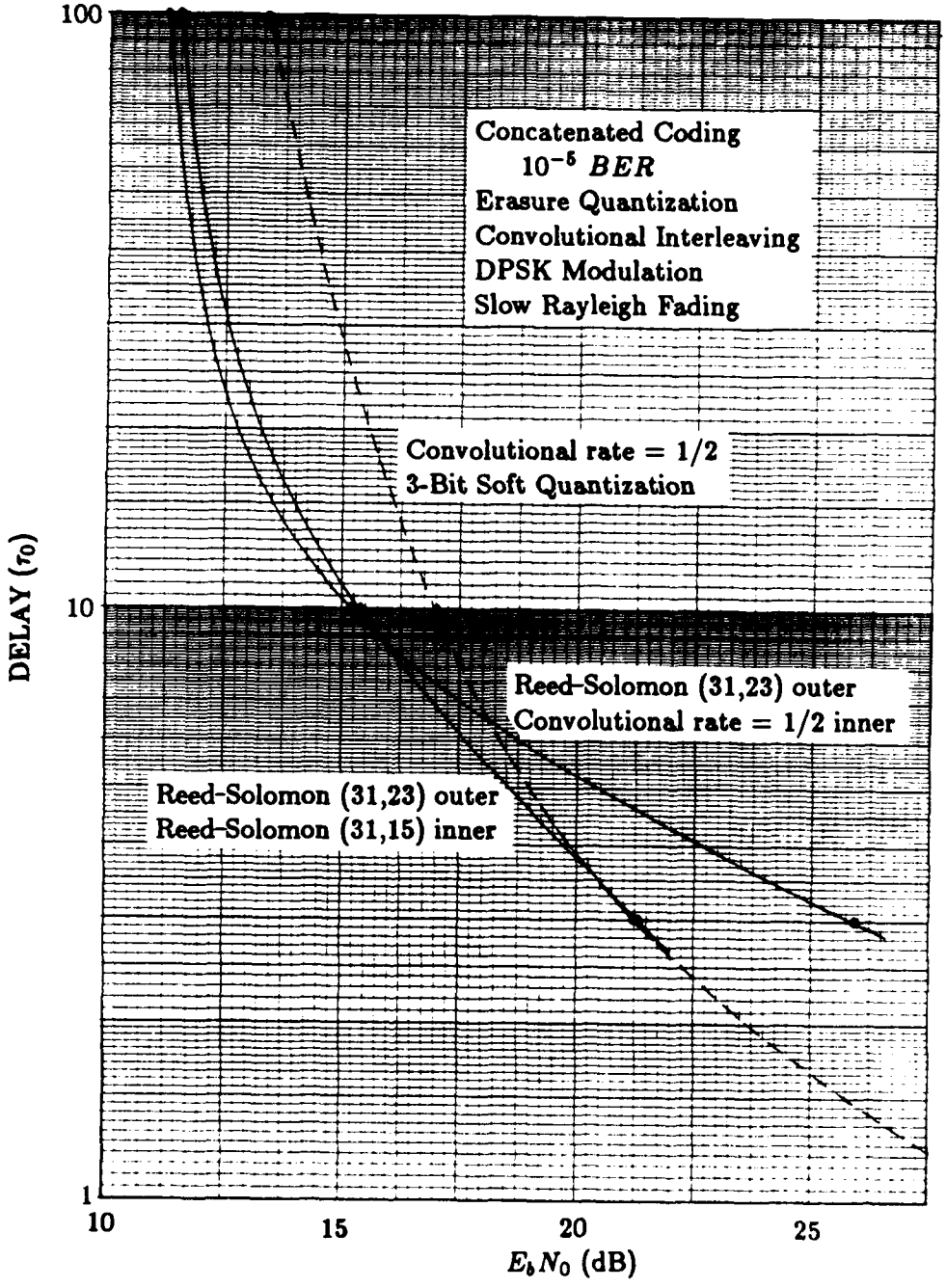


Figure 41. Interleaving delay requirements for efficient concatenated codes with 10^{-5} BER.

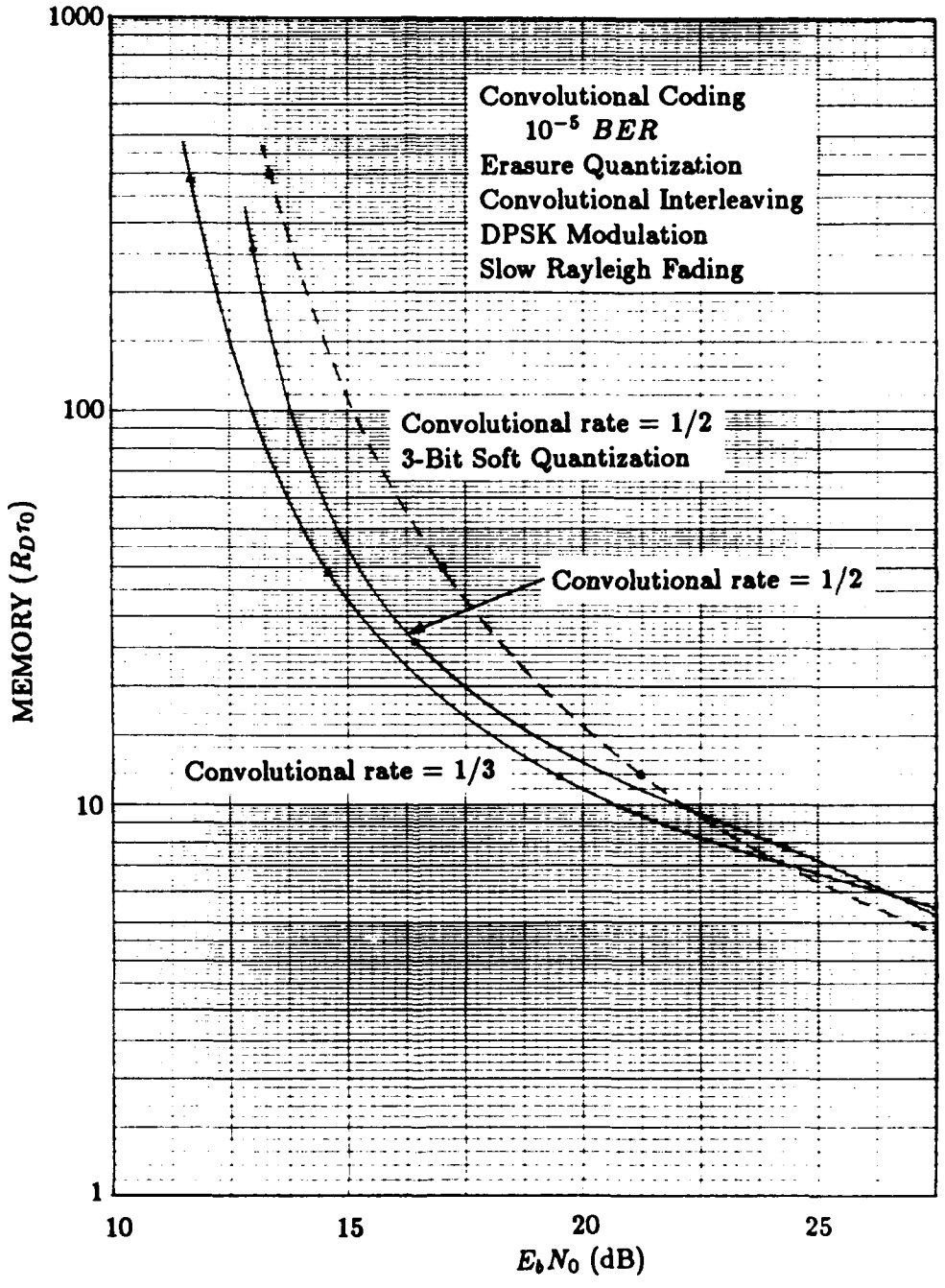


Figure 42. Interleaving memory requirements for efficient convolutional codes with 10^{-5} BER.

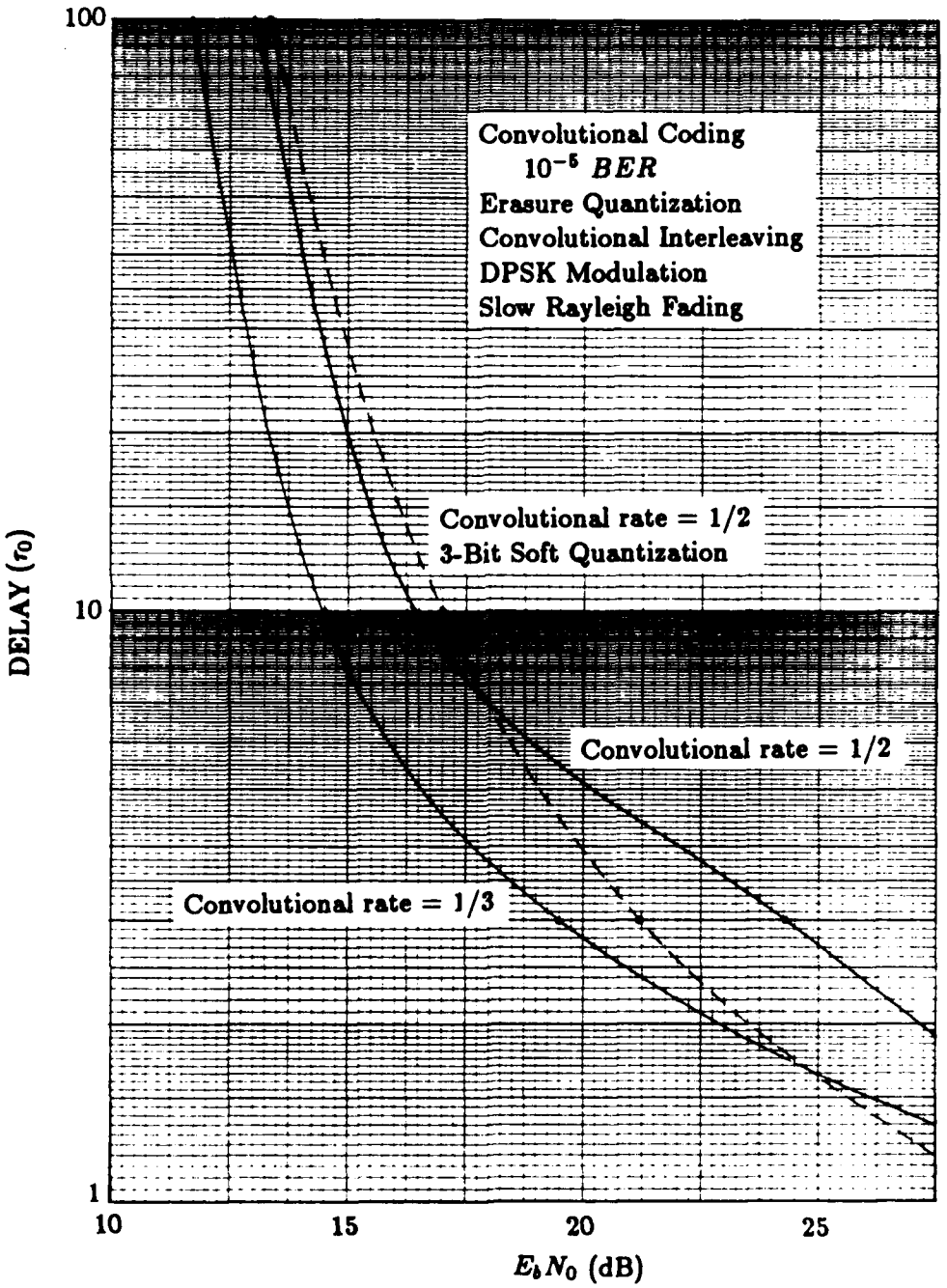


Figure 43. Interleaving delay requirements for efficient convolutional codes with 10^{-5} BER.

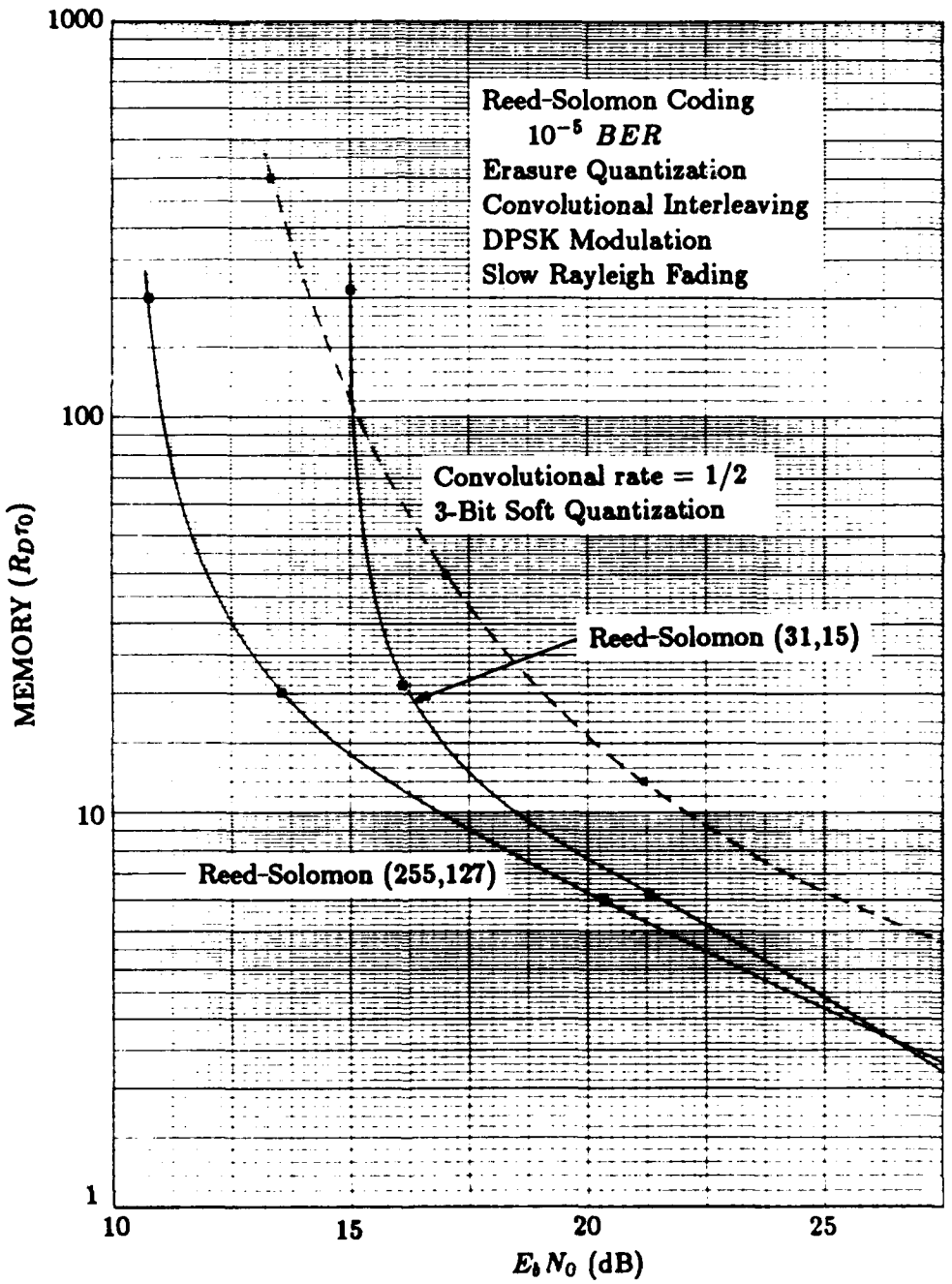


Figure 44. Interleaving memory requirements for efficient Reed-Solomon codes with 10^{-5} BER.

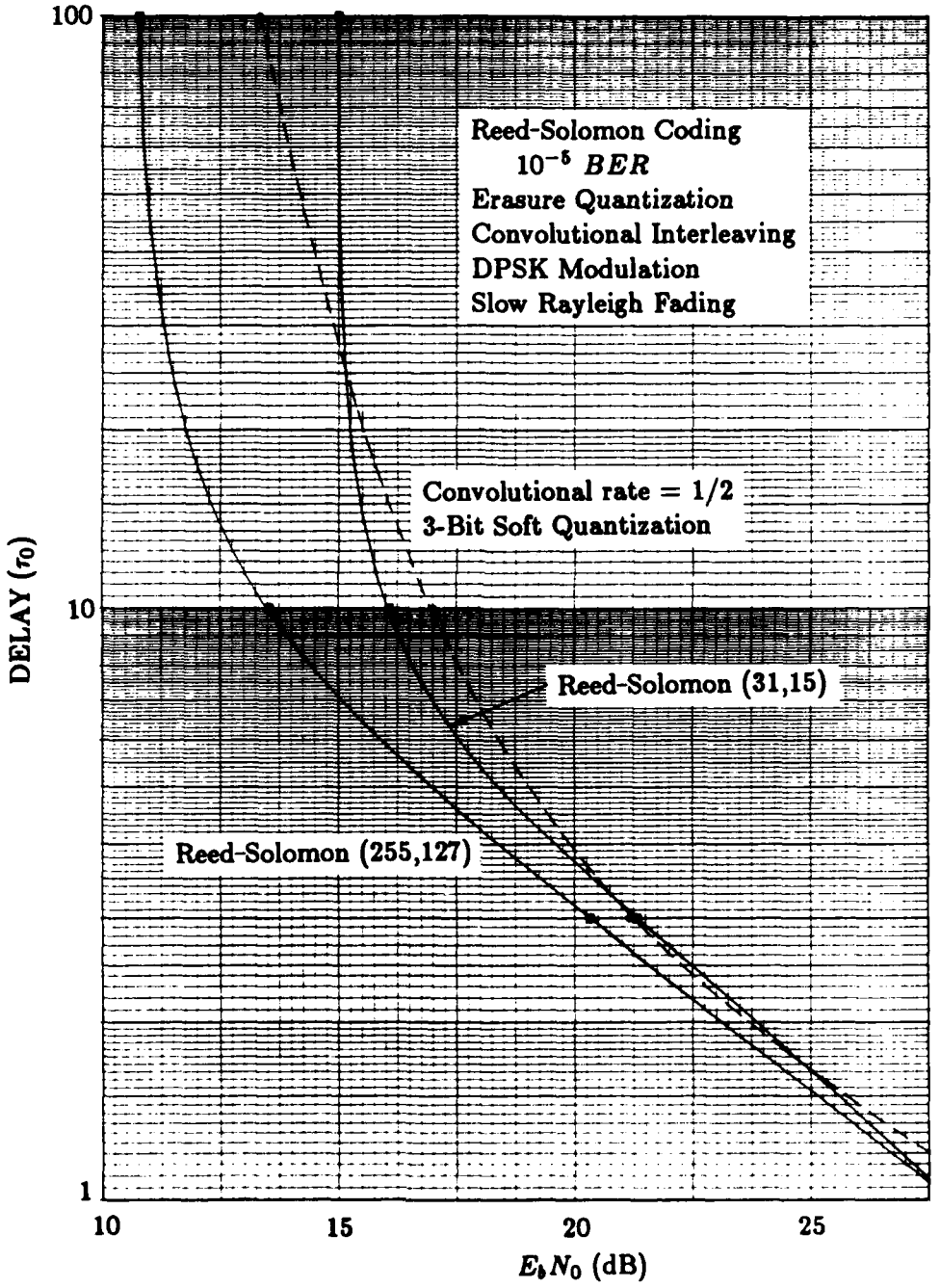


Figure 45. Interleaving delay requirements for efficient Reed-Solomon codes with 10^{-5} BER.

The first two figures (40 and 41) show the memory and delay requirements for outer Reed-Solomon (31,23) concatenated coding with inner convolutional rate 1/2, erasure-decision coding and inner Reed-Solomon (31,15) erasure-decision coding. For E_b/N_0 smaller than 14 dB, interleaving requirements for both codes are very similar but the inner convolutional rate 1/2 code has about 1 dB smaller E_b/N_0 requirements at comparable delay. Compared to the convolutional code reference, the concatenated codes have up to 3 dB smaller E_b/N_0 requirements for E_b/N_0 values below 14 dB.

The next two figures (42 and 43) show the memory and delay requirements for convolutional rate 1/3 and rate 1/2, erasure-decision coding. Convolutional rate 1/2 erasure-decision coding was included in these results to assess the effect of code rate reduction. Convolutional rate 1/3, erasure-decision coding has substantially smaller interleaving delay requirements but similar interleaving memory requirements to those of convolutional rate 1/2, erasure-decision coding. These results suggest that convolutional rate 1/4 coding probably has smaller delay requirements than rate 1/2 or 1/3 coding but convolutional rate 1/4 coding may have larger memory requirements as well.

Comparison of the interleaving requirements for convolutional rate 1/3, erasure-decision coding and convolutional rate 1/2, soft-decision coding indicates that convolutional rate 1/3, erasure-decision coding has up to 2.5 dB smaller E_b/N_0 requirements at comparable memory and delay. These savings occur for E_b/N_0 between 14 and 18 dB.

The last two figures (44 and 45) show the memory and delay requirements for Reed-Solomon (31,15) and (255,127) erasure-decision coding. Comparison of interleaving requirements for these two codes indicates that the Reed-Solomon (255,127) erasure-decision coding has substantially smaller interleaving requirements than those of Reed-Solomon (31,15) erasure-decision coding for E_b/N_0 below about 18 dB. However, at larger values of E_b/N_0 the difference in interleaving requirements is relatively small. These results suggest that rate 1/2 Reed-Solomon coding with block sizes of 63 or 127, may have substantially smaller memory and delay requirements than Reed-Solomon (31,15) erasure-decision coding for E_b/N_0 below 18 dB, however, interleaving savings at larger E_b/N_0 will be negligible.

Comparison of the interleaving memory requirements for Reed-Solomon (31,15) erasure-decision coding and the convolutional rate 1/2 code reference indicates that Reed-Solomon (31,15) erasure-decision coding has about 4.5 dB lower E_b/N_0 requirements for comparable memory at values at E_b/N_0 above 17 dB. Reed-Solomon (31,15) erasure-decision coding has comparable delay requirements to convolutional rate 1/2, soft-decision coding, however.

In addition to these results, all of the results in this investigation showed that erasure quantization produced the lowest interleaving memory and the lowest or close to the lowest delay requirements. Convolutional coding selections generally had lower delay requirements for 3-bit soft quantization. With concatenated coding, the results indicated that erasure quantization with inner coding substantially reduced interleaving memory requirements, but erasure quantization with outer coding only lead to a small (1/4 dB) E_b/N_0 savings.

6.2 CODE SELECTION.

Having established which of the evaluated codes have the lowest memory and delay requirements, a question still remains as to how choose the best of code from different E_b/N_0, memory, and delay specifications.

If E_b/N_0 is specified, then the code with the lowest memory and delay requirements at the specified value of E_b/N_0 can be selected directly from plots of interleaving requirements. If, alternatively, E_b/N_0 is not specified but the maximum permitted value of interleaving memory and/or delay are specified then the best selection is to find the code with the smallest E_b/N_0 requirement needed to satisfy the memory and delay specifications. Since interleaving requirements are presented in plots of $R_d\tau_0$ memory and τ_0 delay units, interleaving memory and delay specifications in bits and seconds respectively must be divided by plotted units to determine their specifications in the plotted results. The code with smallest E_b/N_0 requirement that satisfies both interleaver requirements can then be determined.

To illustrate these two code selection approaches, first consider only having an E_b/N_0 requirement. If this requirement is less than 14 dB then outer Reed-Solomon (31,23) coding with inner convolutional rate 1/2, erasure-decision coding is a good selection. If the requirement is larger than 14 dB and minimizing delay is important then convolutional rate 1/3 erasure-decision coding is a good selection. If the requirement is larger than 16 dB and minimizing memory is important then Reed-Solomon (31,15) erasure-decision is a good choice.

Next consider a specification for the maximum acceptable interleaving memory and delay values. In this case the data rate and τ_0 must also be known before the code with the minimum E_b/N_0 requirement can be determined. Assume that the communication link supports a high data rate of 1 Mbps and the communication link must operate reliably with a maximum τ_0 of 250 ms. If the interleaver memory specifi-

cation is 4 Mbits then the memory requirement in $R_d\tau_0$ units is 16. If the interleaving delay specification is 1 second then the memory requirement in τ_0 units is 4.

Referring back to Figures 40, 41, 42, 43, 44, and 45, Reed-Solomon (31,15) erasure-decision coding has the smallest E_b/N_0 requirement for the 16 $R_d\tau_0$ memory specification. This requirement is about 17 dB. Convolutional rate 1/3, erasure-decision coding has the smallest E_b/N_0 requirement for the 4 τ_0 delay specification. This requirement is about 18 dB. The E_b/N_0 requirement for convolutional rate 1/3, erasure-decision coding to satisfy the memory specification is about 17.5 dB and the E_b/N_0 requirement for Reed-Solomon (31,15) erasure-decision coding to satisfy the delay specification is about 19.5 dB. Convolutional rate 1/3, erasure-decision coding is the best code choice since it has the lowest E_b/N_0 requirement needed to satisfy both interleaving memory requirements. This requirement is about 18 dB.

Since this approach requires a backward evaluation of the interleaving requirement plots, it is not directly evident which codes are the best code selections for different memory and delay specifications. However, by relating E_b/N_0 to memory and delay requirements for the codes with the smallest interleaving requirements, outer Reed-Solomon (31,23) coding with inner convolutional rate 1/2, erasure-decision coding is seen as the best selection for relatively large memory and delay specifications in plot-units. Convolutional rate 1/3 erasure-decision coding is the best code selection for relatively small τ_0 delay specifications and Reed-Solomon erasure-decision coding is the best code selection for relatively small $R_d\tau_0$ memory specifications.

6.3 RECOMMENDATION FOR FURTHER INVESTIGATION.

During this investigation, several codes were identified as having potentially smaller interleaving requirements than the interleaving requirements for the evaluated codes. However, either evaluation time-constraints or complexity and bandwidth considerations limited their selection.

Of the codes identified with potentially small interleaving requirements that satisfy the complexity and code rate restrictions of this investigation, convolutional rate 1/4 erasure-decision coding and outer Reed-Solomon (31,23) concatenated coding with inner convolutional rate 1/3 coding are the most promising choices. The convolutional rate 1/4 code will likely provide very low delay requirements and moderately low memory requirements at relatively large E_b/N_0 and small interleaving specifications. The inner convolutional rate 1/3 concatenated code will likely provide relatively

low memory and delay requirements at smaller E_b/N_0 and larger interleaving specifications.

If the code complexity and code rate restrictions of this investigation are relaxed then lower rate convolutional codes and larger size Reed-Solomon codes may also be good choices. Lower rate convolutional codes will likely reduce delay requirements at relatively large E_b/N_0 and small interleaving specifications. Larger size Reed-Solomon codes will likely reduce memory and delay requirements at smaller E_b/N_0 and larger interleaving specifications.

In addition to these recommendations, an alternative approach to pseudo-random error correction coding with decomposed convolutional coding is recommended for further investigation. This approach, discussed in the Introduction, has potential to substantially reduce interleaving requirements and still satisfy the code complexity and code rate requirements of this investigation.

SECTION 7

LIST OF REFERENCES

1. Bogusch, R. L., *Digital Communications in Fading Channels: Modulation and Coding*, MRC-R-1043, Mission Research Corporation, 11 March 1987.

2. Feldman, P. M., *Calculating Reed-Solomon Misdecode Probabilities*, MRC-N-907, Mission Research Corporation, May 1990.

3. Forney, Jr., G. D., *Concatenated Codes*, M.I.T. Press, Cambridge, MA 1966.

4. Hauptschein, A., "Practical High Performance Concatenated Coded Spread Spectrum Channel for JTIDS," *National Telecommunications Conference, Vol. 3*, pp. 35:4-1–4-8, 1977.

5. Jennings, S., "On Computing the Performance Probabilities of Reed-Solomon Codes," *Algebraic Algorithms and Error-Correcting Codes, 3rd Int. Conf.*, pp.61-68, July 15-19, 1985.

6. Ng, W-H. and N-N. Hsieh, "Mitigating Effects of Interference Through Coding and Interleaving," *Proc. IEEE MILCOM 1986*, pp. 20.5.1-5.5, October 5-9, 1986.

7. Odenwalder, J. P., *Error Control Coding Handbook*, Linkabit Corporation, 15 July 1976.

8. Proakis, J. G., *Digital Communications*, McGraw-Hill Book Company, 1983.

APPENDIX A

DERIVATION OF REED-SOLOMON CODE PERFORMANCE

A.1 APPROACH.

The approach used to derive Reed-Solomon *BER* performance in Section 2 involved a combination of analysis and simulation. Decoder input symbol error probability was first simulated with modulation and/or inner code selections. The symbol error probability was then converted to decoder output bit error probability (decoded *BER*) by analytical expression for the Reed-Solomon decoder input symbol error probability to output bit error probability transfer function. The symbol probability determined for these derivations was for the Reed-Solomon symbols with $log_2(N+1)$ bits per symbol, where N is the Reed-Solomon block size in symbols.

Simulation of Reed-Solomon symbol error probability was required because analytical expressions of it with the modulation and code selections in the derivations of Section 2 are very complex. In particular, DPSK modulation, chosen for Reed-Solomon single-stage code performance derivations, has a complex error probability expression for Reed-Solomon code symbols because derivations were conducted with multiple-bit code symbols, slow Rayleigh fading, and the assumption of code symbol interleaving*. Inner convolutional coding, chosen for concatenated code derivations, has far a more complex symbol error probability.

Despite the requirement for symbol error probability simulations, Reed-Solomon *BER* performance was derived relatively quickly by this approach. The transfer function expression used in this approach required that several conditions be satisfied, however, to ensure that Reed-Solomon *BER* is accurately derived.

One of these conditions is that the input symbols to the Reed-Solomon decoder be independent regardless of channel conditions. With fading channel conditions this is satisfied when T_{il}/τ_0 is approximately 100 or larger. Another condition is that binary data within Reed-Solomon encoded symbols represent a random mix of possible values. This condition occurs for random user data and is typical of most communication applications. A third condition is that Reed-Solomon misdecode error probability be negligible. This condition is generally satisfied for Reed-Solomon codes

*In these cases, binary demodulated symbols within code symbols remain correlated.

with block sizes no smaller than 31 and with code rates no smaller than about 3/4 (in the 31 block size codes); this is addressed in Section 3.2.

The first two conditions were satisfied in the derivations of Section 2 by simulating symbol error probability with large T_{il}/τ_0 selections (of 100) using random user data. The last condition was satisfied, for the most part, by deriving code performance for 31 block size or larger codes. Rate 7/8 Reed-Solomon (31,27) coding was included in these derivations, however.

A.2 TRANSFER FUNCTION.

The transfer function used in this approach was the same function used in several code sources [Odenwalder, Hauptschein] for deriving Reed-Solomon BER performance under benign (AWGN) conditions. It was accurately applied to fading channel conditions but satisfying the previously described conditions. To provide the reader with a thorough understanding of the transfer function, its is addressed next.

The Reed-Solomon transfer function is derived by first considering the Reed-Solomon decoding-decision algorithm. The decoder makes the decision to pass uncorrected symbols when the following expression is satisfied

$$2e > N - K \quad . \tag{15}$$

Here N denotes the number of symbols in an encoded block of symbols, K denotes the number of pre-encoded symbols in a block of symbols, and e denotes the number of errors in block of symbols received at the decoder.

Assuming the decoder estimates the correct number of symbol errors (and does not make a misdecode), the Reed-Solomon decoder passes $[(N-K)/2]$ or more symbol errors when Equation 15 is satisfied. Assuming that the decoder receives independent symbols (from large T_{il}/τ_0), it has an uncorrected block error probability equal to the sum of the probability of $[(N-K)/2]$ to N symbol errors within N symbols. This probability equals

$$P_{blk} = \sum_{k=[(N-K)/2]}^{N} \frac{N!}{(N-k)!k!} P_{si}^{k}(1-P_{si})^{N-k} \tag{16}$$

where P_{si} denotes the decoded input symbol-error probability.

The decoded symbol-error probability is formulated next by associating Equation 16. This is done by first noting that the decoded symbol-error probability

for a block of uncorrected data symbols is approximately equal to the symbol-error probability for the same block of data and parity symbols. Next, note that the symbol error probability for a block of data and parity symbols is equal to the average number of symbol errors within an block of received symbols, divided by the number of symbols within a block (N). The decoded-symbol error probability can then be equated to the average number of symbols in block of received symbols by averaging k in Equation 16 and dividing this value by N. The result is the equation

$$P_{so} = \frac{1}{N} \sum_{k=[(N-K)/2]}^{N} k \frac{N!}{(N-k)!k!} P_{si}^{k} (1 - P_{si})^{N-k} \quad . \tag{17}$$

Assuming random bits within decoded symbols, the Reed-Solomon decoded bit-error probability is related to the decoded symbol-error probability by the following equation, derived in [Proakis]

$$P_b = \frac{2^{n-1}}{2^n - 1} P_{so} \tag{18}$$

where P_b denotes the decoded bit error probability and n equals the number of bits per code symbol.

The Reed-Solomon decoded bit error probability is defined by combining the expressions in Equation 17 and Equation 18. The resulting equation equals

$$P_b = \frac{2^{n-1}}{2^n - 1} \sum_{k=[(N-K/2]}^{N} \frac{(N-1)!}{(N-k)!(k-1)!} P_{si}^{k} (1 - P_{si})^{N-k} \quad . \tag{19}$$

APPENDIX B

INTERLEAVING MEMORY REQUIREMENTS

This appendix contains convolutional interleaving memory requirements for all of the codes that were evaluated. These requirements are presented in Tables 2 and 3 and are defined in $R_d T_{il}$ units to permit quick conversion to $R_d \tau_0$ units from a T_{il}/τ_0 specification. For this conversion, the results simply need to be multiplied by the T_{il}/τ_0 value.

For code selections with erasure quantization, two interleaving memory requirements are included in the tables. These requirements follow pre-decoded symbol size specifications, n_Q, with the subscript E. The first value specifies the memory requirement without erasure-symbol combining and the second value specifies the memory requirement with erasure-symbol combining for 10 combined symbols. The ratio of deinterleaver memory requirements for these values is by the variable r_E.

The memory requirements were derived from a modification of Equation 11 of Section 4.1. This requirement is

$$M = \frac{\tau_0}{2r}\left(1 + r_E \frac{n_Q}{n}\right) \; R_d T_{il} \; units \tag{20}$$

where r denotes the code rate, n denotes the number of bits in an encoded symbol, n_Q denotes the number of bits in a quantized code symbol. r_E was derived from Equation 12 of Section 4.2.

Table 2. Interleaving memory requirements for single-stage codes.

Code	Code Rate r	Bits Per Encoded Symbol n	Bits Per Pre-Decoded Symbol n_Q	Interleaver Memory $M(R_dT_{il})$	Erasure Compression	
					Compression Ratio r_E	Interleaver Memory $M(R_dT_{il})$
Convolutional rate 1/2, K = 7	1/2	1	1	2.00		
			2_E	3.00	0.80	2.60
			3	4.00		
Convolutional rate 1/3, K = 7	1/3	1	1	3.00		
			2_E	4.50	0.80	3.90
			3	6.00		
Reed-Solomon (31,15)	15/31	5	5	2.07		
			6_E	2.27	0.85	2.09
Reed-Solomon (31,23)	15/23	5	5	1.35		
			6_E	1.48	0.85	1.36
Reed-Solomon (255,127)	127/255	8	8	2.01		
			9_E	2.13	0.90	2.02
Reed-Solomon (255,191)	199/255	8	8	1.34		
			9_E	1.42	0.90	1.34

Table 3. Interleaving memory requirements for concatenated codes.

Concatenated			Inner Code			Erasure Compaction	
Outer Code	Inner Code	Code Rate r	Bits Per Encoded Symbol n	Bits Per Pre-Decoded Symbol n_Q	Interleaver Memory $M(R_dT_{il})$	Compaction Ratio r_E	Interleaver Memory $M(R_dT_{il})$
Reed-Solomon (255,191)	Convolutional rate 1/2, K = 7	0.37	1	1	2.67		
				2_E	4.01	0.80	3.47
				3	5.34		
	Convolutional Rate 1/3, K = 7	0.25	1	1	4.01		
				2_E	6.01	0.80	5.21
				3	8.01		
	Reed-Solomon (31,15)	0.36	5	5	2.76		
				6_E	3.04	0.85	2.79
	Reed-Solomon (31,23)	0.56	5	5	1.80		
				6_E	1.98	0.85	1.82
Reed-Solomon (255,127)	Convolutional rate 1/2, K = 7	0.25	1	1	4.02		
				2_E	6.02	0.80	5.22
				3	8.03		
	Reed-Solomon (31,15)	0.24	5	5	4.15		
				6_E	4.56	0.85	4.19
Reed-Solomon (31,23)	Convolutional rate 1/2, K = 7	0.37	1	1	2.70		
				2_E	4.04	0.80	3.50
				3	5.39		
	Convolutional rate 1/3, K = 7	0.34	1	1	2.91		
				2_E	4.36	0.80	3.78
				3	5.81		
	Reed-Solomon (31,15)	0.36	5	5	2.79		
				6_E	3.06	0.85	2.81
	Reed-Solomon (31,23)	0.55	5	5	1.82		
				6_E	2.00	0.85	1.83
Reed-Solomon (31,15)	Convolutional rate 1/2, K = 7	0.24	1	1	4.13		
				2_E	6.20	0.80	5.37
				3	8.27		
	Reed-Solomon (31,15)	0.23	5	5	4.27		
				6_E	4.70	0.85	4.31

APPENDIX C

SINGLE-STAGE CODE TRANSFER FUNCTIONS

This appendix contains plots of the Reed-Solomon code transfer functions that were used to derive the memory estimates for the concatenated codes in Section 5.4. These plots, shown in Figures 46, 47, 48, and 49, were derived from decoder input and outer BER values in code performance evaluations. They include T_{il}/τ_0 values of 100, 10, 3, and 1. The results also include, for reference, the transfer functions of Reed-Solomon erasure-decision coding with T_{il}/τ_0 of 10.

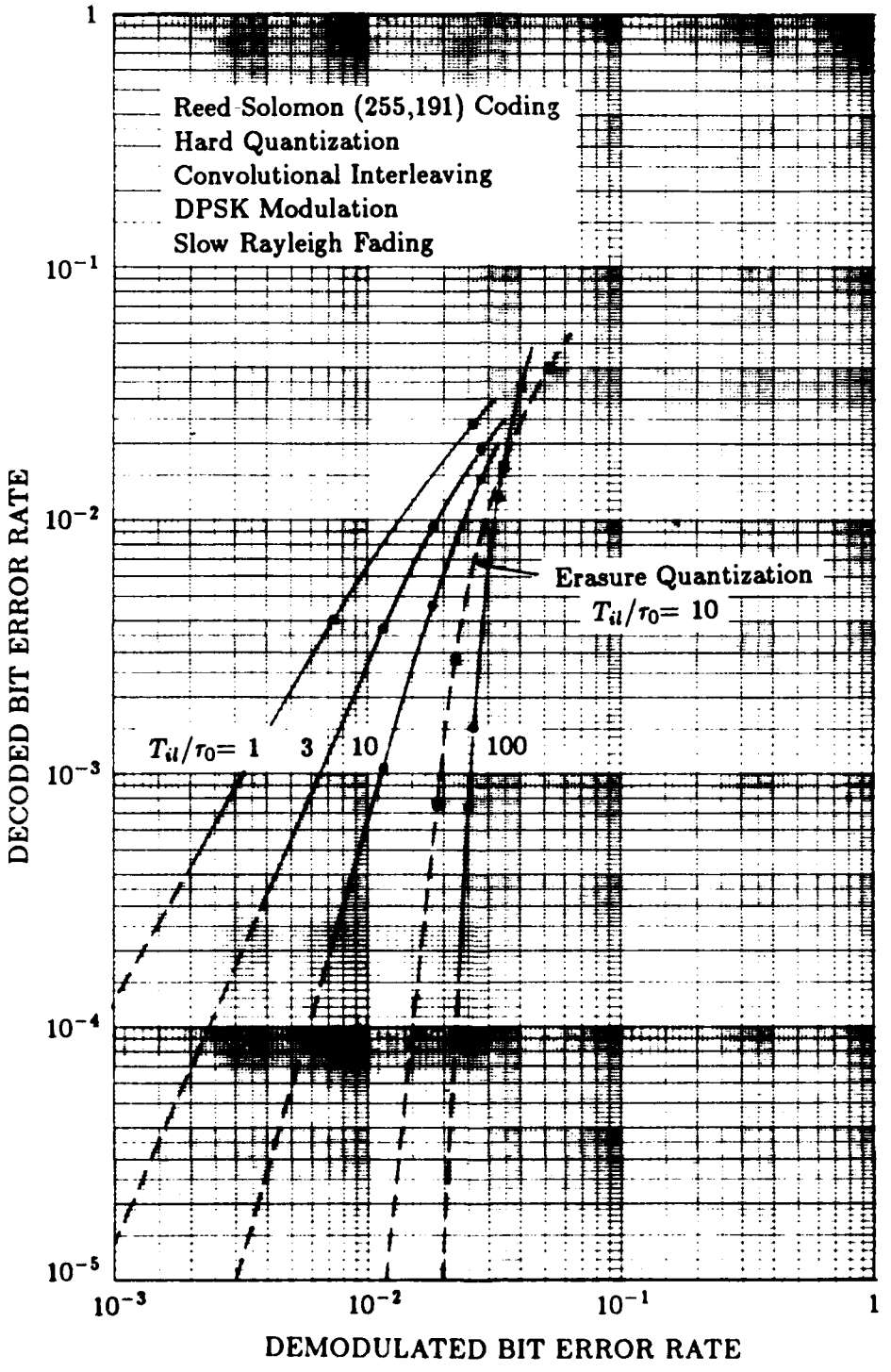


Figure 46. Transfer function for Reed-Solomon (255,191) coding with hard quantization.

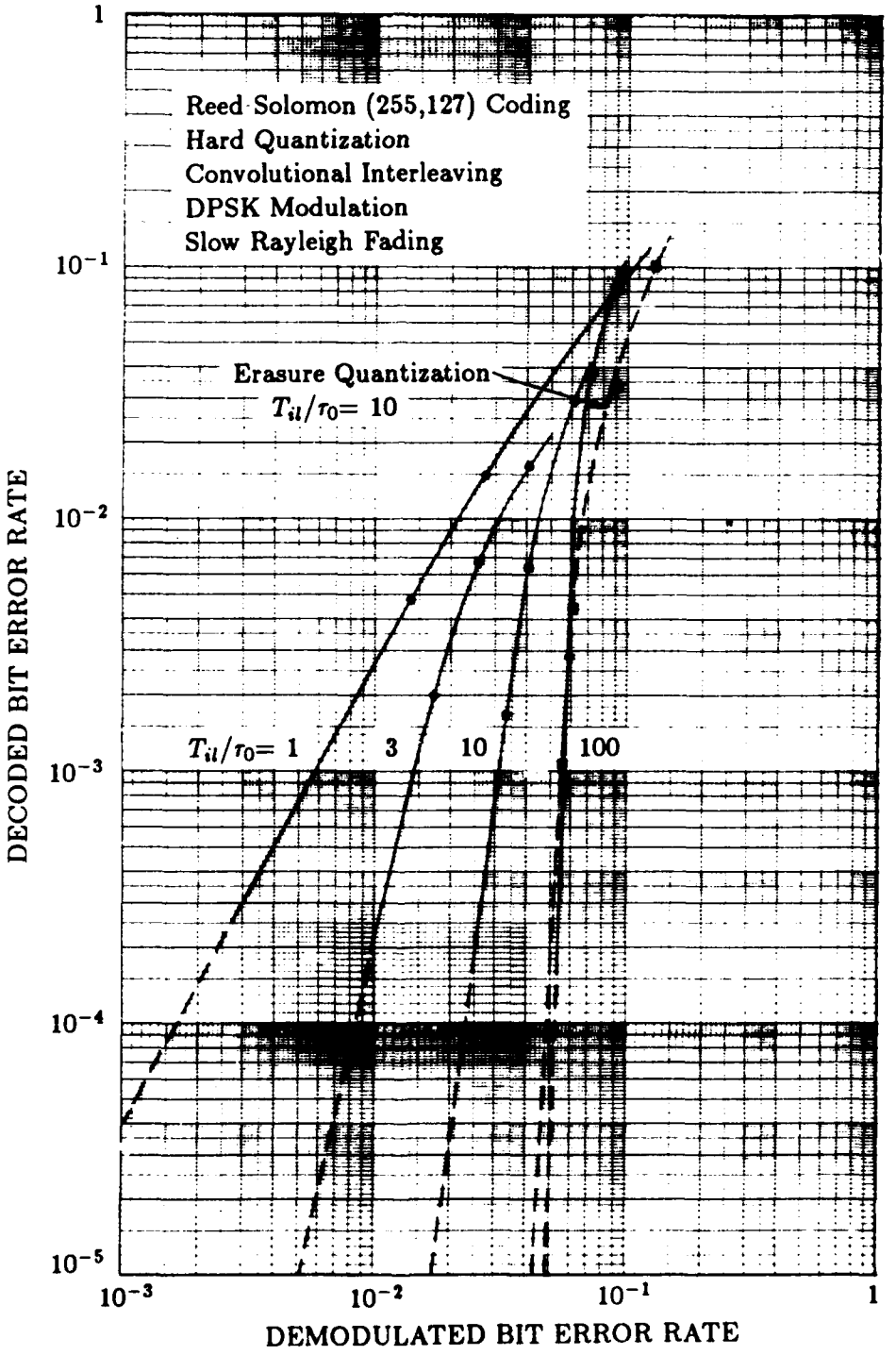


Figure 47. Transfer function for Reed-Solomon (255,127) coding with hard quantization.

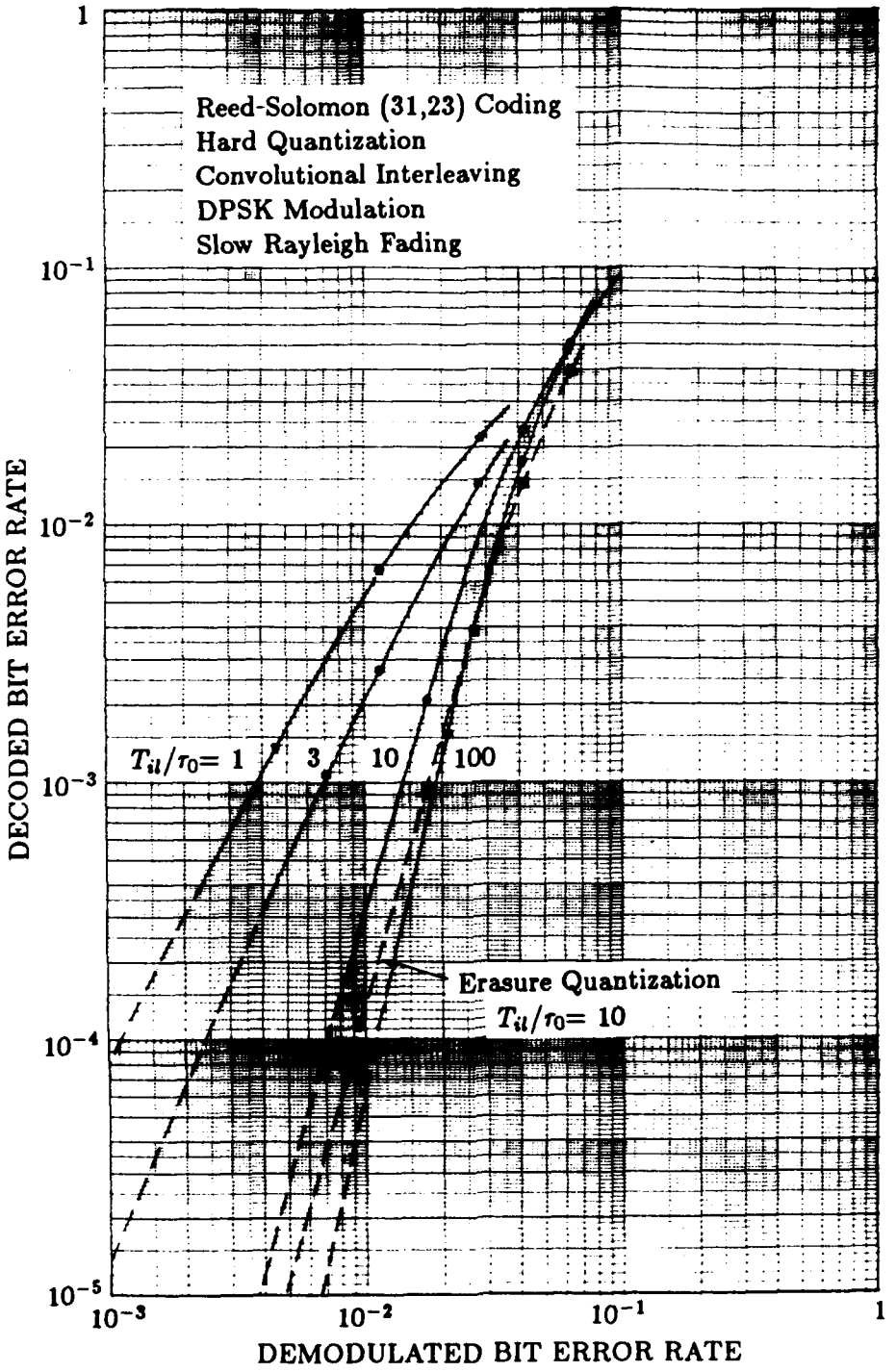


Figure 48. Transfer function for Reed-Solomon (31,23) coding with hard quantization.

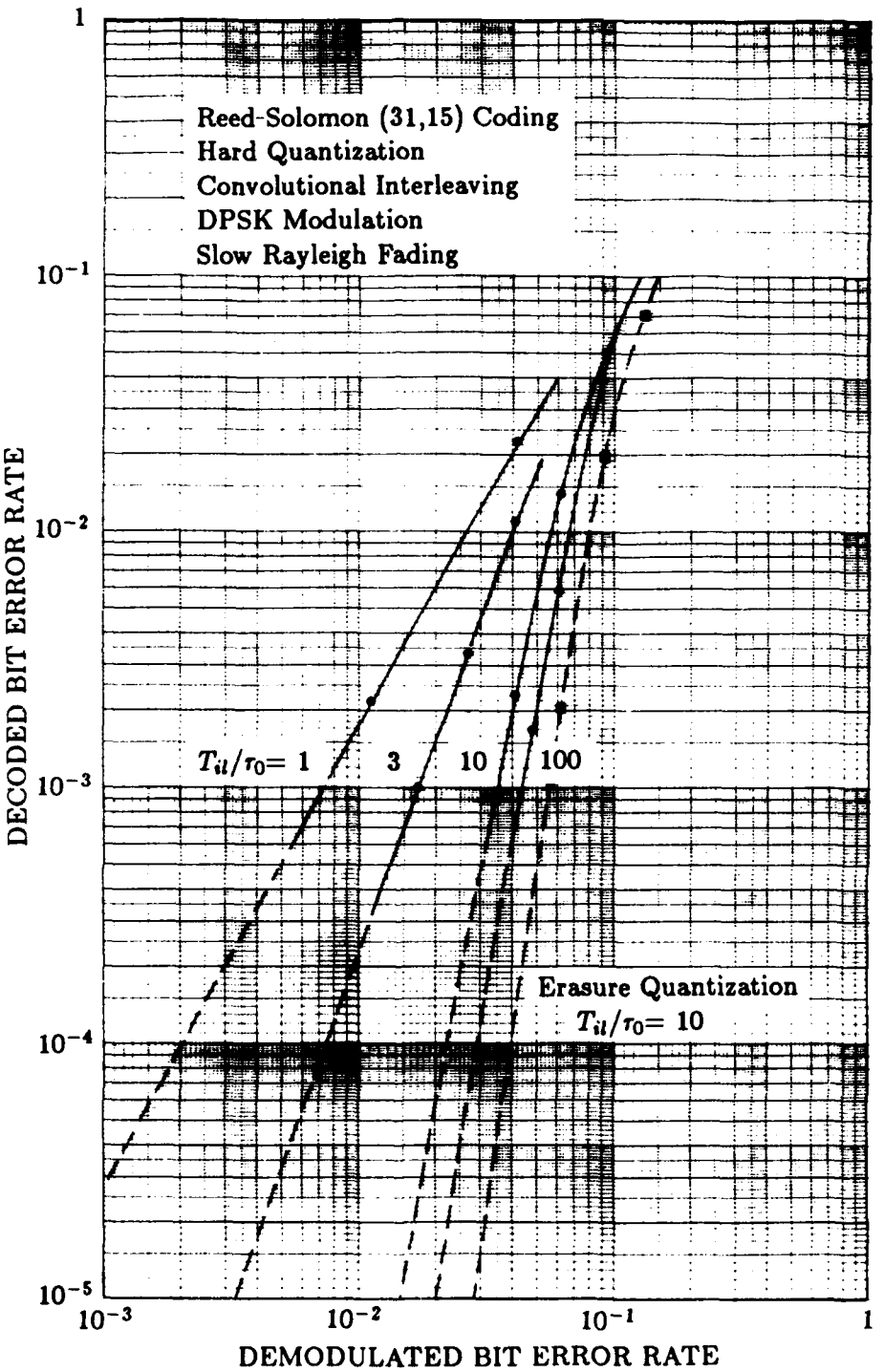


Figure 49. Transfer function for Reed-Solomon (31,15) coding with hard quantization.

DISTRIBUTION LIST

DNA-TR-91-33

DEPARTMENT OF DEFENSE

ASSISTANT TO THE SECRETARY OF DEFENSE
ATTN: EXECUTIVE ASSISTANT

DEFENSE ADVANCED RSCH PROJ AGENCY
ATTN: CHIEF SCIENTIST
ATTN: GSD R ALEWINE

DEFENSE COMMUNICATIONS AGENCY
ATTN: C4S/SSM
ATTN: SSS

DEFENSE COMMUNICATIONS ENGINEER CENTER
ATTN: CODE R430 BOEHM

DEFENSE INTELLIGENCE AGENCY
ATTN: DB-TPO
ATTN: DB-6
ATTN: DC-6
ATTN: DIR
ATTN: DT-1B

DEFENSE NUCLEAR AGENCY
ATTN: NANF
ATTN: NASF
ATTN: OPNA
ATTN: PRPD R YOHO
ATTN: RAAE
3 CYS ATTN: RAAE K SCHWARTZ
ATTN: RAAE A CHESLEY
ATTN: RAAE S BERGGREN
ATTN: RAEE
2 CYS ATTN: TITL

DEFENSE TECHNICAL INFORMATION CENTER
2 CYS ATTN: DTIC/FDAB

FIELD COMMAND DEFENSE NUCLEAR AGENCY
ATTN: FCNM
ATTN: FCPRA LTC R HEDTKE
2 CYS ATTN: FCTT W SUMMA

JOINT DATA SYSTEM SUPPORT CTR
ATTN: JNSV

STRATEGIC AND THEATER NUCLEAR FORCES
ATTN: COL R DAWSON
ATTN: DR E SEVIN
ATTN: DR SCHNEITER

STRATEGIC DEFENSE INITIATIVE ORGANIZATION
ATTN: EN
ATTN: EN LTC C JOHNSON
ATTN: TDW

THE JOINT STAFF
ATTN: JKC (ATTN: DNA REP)
ATTN: JKCS
ATTN: JLWT
ATTN: JPEM

THE JOINT STAFF
ATTN: J6

U S NUCLEAR CMD & CENTRAL SYS SUP STAFF
ATTN: SAB H SEQUINE

DEPARTMENT OF THE ARMY

ARMY LOGISTICS MANAGEMENT CTR
ATTN: DLSIE

HARRY DIAMOND LABORATORIES
ATTN: SLCIS-IM-TL

U S ARMY ATMOSPHERIC SCIENCES LAB
ATTN: SLCAS-AE DR NILES
ATTN: SLCAS-AR DR H HOLT

U S ARMY COMMUNICATIONS R&D COMMAND
ATTN: AMSEL-RD-ESA

U S ARMY ENGINEER DIV HUNTSVILLE
ATTN: PRESTON J KISS

U S ARMY FOREIGN SCIENCE & TECH CTR
ATTN: AIFRTA

U S ARMY NUCLEAR & CHEMICAL AGENCY
ATTN: MONA-NU

U S ARMY NUCLEAR EFFECTS LABORATORY
ATTN: ATAA-PL
ATTN: ATAA-TDC
ATTN: ATRC-WCC LUIS DOMINGUEZ

U S ARMY STRATEGIC DEFENSE CMD
ATTN: CSSD-H-LS B CARRUTH
ATTN: CSSD-H-SA
ATTN: CSSD-H-SAV
ATTN: CSSD-SA-EV RON SMITH

U S ARMY STRATEGIC DEFENSE COMMAND
ATTN: CSSD-GR-S W DICKINSON

USA SURVIVABILITY MANAGMENT OFFICE
ATTN: SLCSM-SE J BRAND

DEPARTMENT OF THE NAVY

COMMAND & CONTROL PROGRAMS
ATTN: OP 941

DEPARTMENT OF THE NAVY
ATTN: JCMG-707

NAVAL AIR SYSTEMS COMMAND
ATTN: PMA 271

NAVAL ELECTRONICS ENGRG ACTVY, PACIFIC
ATTN: CODE 250

NAVAL RESEARCH LABORATORY
ATTN: CODE 2000 J BROWN
ATTN: CODE 2627
ATTN: CODE 4104 H HECKATHORN
ATTN: CODE 4183
ATTN: CODE 4701
ATTN: CODE 4720 J DAVIS
ATTN: CODE 4780 B RIPIN

ATTN: CODE 4780 DR P BERNHARDT
ATTN: CODE 4780 J HUBA
ATTN: CODE 4785 P RODRIGUEZ
ATTN: CODE 5300
ATTN: CODE 5326 G A ANDREWS
ATTN: CODE 5340 E MOKOLE
ATTN: CODE 8344 M KAPLAN
ATTN: JACOB GRUN

NAVAL SURFACE WARFARE CENTER
ATTN: CODE H-21

NAVAL TECHNICAL INTELLIGENCE CTR
ATTN: DA44

NAVAL UNDERWATER SYSTEMS CENTER
ATTN: CODE 3411 J KATAN

OFFICE OF CHIEF OF NAVAL OPERATIONS
ATTN: OP 654
ATTN: OP 941D

SPACE & NAVAL WARFARE SYSTEMS CMD
ATTN: CODE 3101 T HUGHES
ATTN: PD50TD1 G BRUNHART
ATTN: PME 106-4 S KEARNEY
ATTN: PME-106 F W DIEDERICH

THEATER NUCLEAR WARFARE PROGRAM OFC
ATTN: PMS-42331F D SMITH

DEPARTMENT OF THE AIR FORCE

AFIA/INKS
ATTN: AFIA/INKS MAJ SCHROCK
ATTN: AFIA/INKS

AIR FORCE CTR FOR STUDIES & ANALYSIS
ATTN: AFCSA/SASC

AIR FORCE ELECTRONIC WARFARE CENTER
ATTN: LT M MCNEELY
ATTN: SAZ

AIR FORCE GEOPHYSICS LABORATORY
ATTN: J KLOUBACHAR
ATTN: OP W BLUMBERG
ATTN: SANTI BASU

AIR FORCE OFFICE OF SCIENTIFIC RSCH
ATTN: AFOSR/NP

AIR FORCE SYSTEMS COMMAND
ATTN: XTTW

AIR FORCE SYSTEMS COMMAND
ATTN: DCS/REQUIREMENTS J COLYER

AIR UNIVERSITY LIBRARY
ATTN: AUL-LSE

HQ AWS, DET 3 (CSTC/WE)
ATTN: WE

NATIONAL TEST FACILITY
ATTN: NTB/JPO DR C GIESE

PHILLIPS LABORATORY
ATTN: NTCA
ATTN: NTCTS LTC C ROYER
ATTN: NTN

STRATEGIC AIR COMMAND/XRFS
ATTN: IN
ATTN: XRFS

DEPARTMENT OF ENERGY

EG&G, INC
ATTN: D WRIGHT

LAWRENCE LIVERMORE NATIONAL LAB
ATTN: L-97 T DONICH

SANDIA NATIONAL LABORATORIES
ATTN: P L MATTERN 8300

SANDIA NATIONAL LABORATORIES
ATTN: D DAHLGREN 6410
ATTN: DIV 2344 ROBERT M AXLINE
ATTN: DIV 9414 R BACKSTROM
ATTN: ORG 9110 G CABLE
ATTN: ORG 9110 W D BROWN
ATTN: TECH LIB 3141

OTHER GOVERNMENT

CENTRAL INTELLIGENCE AGENCY
ATTN: OSWR/NED
ATTN: OSWR/SSD FOR L BERG

DEPARTMENT OF COMMERCE
ATTN: G REEVE
ATTN: J HOFFMEYER
ATTN: W UTLAUT

U S DEPARTMENT OF STATE
ATTN: PM/TMP

DEPARTMENT OF DEFENSE CONTRACTORS

AEROSPACE CORP
ATTN: A MORSE
ATTN: BRIAN PURCELL
ATTN: C CREWS 92957
ATTN: C RICE
ATTN: DR J M STRAUS
ATTN: G LIGHT
ATTN: I GARFUNKEL
ATTN: J KLUCK
ATTN: J THACKER
ATTN: M ROLENZ

AT&T BELL LABORATORIES
ATTN: DENIS S LONGO
ATTN: JOSEPH A SCHOLL
ATTN: N BEAUCHAMP

ATLANTIC RESEARCH SERVICES CORP
ATTN: R MCMILLAN

ATMOSPHERIC AND ENVIRONMENTAL RESEARCH INC
ATTN: M KO

AUSTIN RESEARCH ASSOCIATES
ATTN: J THOMPSON

AUTOMETRIC, INC
ATTN: C LUCAS

BDM INTERNATIONAL INC
ATTN: W LARRY JOHNSON

BERKELEY RSCH ASSOCIATES, INC
ATTN: J WORKMAN
ATTN: N T GLADD
ATTN: S BRECHT

BOEING CO
ATTN: G HALL

CHARLES STARK DRAPER LAB, INC
ATTN: A TETEWSKI

COMMUNICATIONS SATELLITE CORP
ATTN: RICHARD A ARNDT

CORNELL UNIVERSITY
ATTN: D FARLEY JR
ATTN: M KELLY

DYNETICS, INC
ATTN: WILLIAM D TEPPER

ELECTROSPACE SYSTEMS, INC
ATTN: LINDA CALDWELL
ATTN: P PHILLIPS

EOS TECHNOLOGIES, INC
ATTN: B GABBARD
ATTN: R LELEVIER

FORD AEROSPACE CORPORATION
ATTN: PATRICIA BIRDWELL

GENERAL ELECTRIC COMPANY
ATTN: JOSEPH E STROSSER

GENERAL RESEARCH CORP INC
ATTN: J EOLL

GRUMMAN AEROSPACE CORP
ATTN: J DIGLIO

HARRIS CORPORATION
ATTN: LYMUEL MCRAE

HSS, INC
ATTN: D HANSEN

INFORMATION SCIENCE, INC
ATTN: W DUDZIAK

INSTITUTE FOR DEFENSE ANALYSES
ATTN: E BAUER
ATTN: H WOLFHARD

JAYCOR
ATTN: A GLASSMAN
ATTN: J SPERLING

JOHNS HOPKINS UNIVERSITY
ATTN: C MENG
ATTN: H G TORNATORE
ATTN: J D PHILLIPS
ATTN: R STOKES

KAMAN SCIENCES CORP
ATTN: DASIAC
ATTN: E CONRAD
ATTN: G DITTBERNER

KAMAN SCIENCES CORPORATION
ATTN: B GAMBILL
ATTN: DASIAC
ATTN: R RUTHERFORD

LOCKHEED MISSILES & SPACE CO, INC
ATTN: J HENLEY
ATTN: J KUMER
ATTN: R SEARS

LOCKHEED MISSILES & SPACE CO, INC
ATTN: CARL CRABILL
ATTN: D T RAMPTON
ATTN: E M DIMZCELI

LOGICON R & D ASSOCIATES
ATTN: D CARLSON
ATTN: S WOODFORD

LTV AEROSPACE & DEFENSE COMPANY
2 CYS ATTN: LIBRARY EM-08

M I T LINCOLN LAB
ATTN: I KUPIEC L-100

MARTIN MARIETTA DENVER AEROSPACE
ATTN: J BENNETT

MAXIM TECHNOLOGIES, INC
ATTN: B PHILLIPS
ATTN: J SCHLOBOHM
ATTN: N CIANOS

MAXWELL LABS, INC
ATTN: K WARE

MCDONNELL DOUGLAS CORPORATION
ATTN: J GROSSMAN
ATTN: R HALPRIN

METATECH CORPORATION
ATTN: W RADASKY

METEOR COMMUNICATIONS CORP
ATTN: R LEADER

MISSION RESEARCH CORP
ATTN: R ARMSTRONG
ATTN: W WHITE

MISSION RESEARCH CORP
ATTN: R L BOGUSCH

MISSION RESEARCH CORP
ATTN: DAVE GUICE

MISSION RESEARCH CORP
ATTN: B R MILNER
ATTN: C LONGMIRE
ATTN: D KNEPP
ATTN: D LANDMAN
ATTN: F FAJEN
ATTN: F GUIGLIANO
ATTN: G MCCARTOR

2 CYS ATTN: J W STRATER
ATTN: K COSNER
ATTN: M FIRESTONE
ATTN: R BIGONI
ATTN: R DANA
ATTN: R HENDRICK
ATTN: R KILB
ATTN: S GUTSCHE
ATTN: TECH LIBRARY

MITRE CORPORATION
ATTN: M HORROCKS
ATTN: R C PESCI
ATTN: W FOSTER

MITRE CORPORATION
ATTN: G CAMPARETTO

NORTHWEST RESEARCH ASSOC, INC
ATTN: E FREMOUW

PACIFIC-SIERRA RESEARCH CORP
ATTN: E FIELD JR
ATTN: H BRODE

PHOTOMETRICS, INC
ATTN: I L KOFSKY

PHOTON RESEARCH ASSOCIATES
ATTN: D BURWELL

PHYSICAL RESEARCH INC
ATTN: W. SHIH

PHYSICAL RESEARCH INC
ATTN: A CECERE

PHYSICAL RESEARCH, INC
ATTN: R DELIBERIS
ATTN: T STEPHENS

PHYSICAL RESEARCH, INC
ATTN: J DEVORE
ATTN: J THOMPSON
ATTN: W SCHLUETER

PHYSICS INTERNATIONAL CO
ATTN: C GILMAN

R & D ASSOCIATES
ATTN: G HOYT
ATTN: L DERAAB

RAND CORP
ATTN: C CRAIN
ATTN: E BEDROSIAN

RAND CORP
ATTN: B BENNETT

RJO ENTERPRISES/POET FAC
ATTN: STEVEN KRAMER

SCIENCE APPLICATIONS INTL CORP
ATTN: C SMITH
ATTN: D SACHS
2 CYS ATTN: L LINSON

SCIENCE APPLICATIONS INTL CORP
ATTN: H SUNKENBERG
ATTN: LIBRARY

SCIENCE APPLICATIONS INTL CORP
ATTN: S ROSENCWEIG

SPARTA INC
ATTN: D DEAN

SRI INTERNATIONAL
ATTN: R LIVINGSTON
ATTN: R T TSUNODA
ATTN: W CHESNUT

STEWART RADIANCE LABORATORY
ATTN: R HUPPI

TELECOMMUNICATION SCIENCE ASSOCIATES
ATTN: R BUCKNER

TELEDYNE BROWN ENGINEERING
ATTN: J FORD
ATTN: J WOLFSBERGER JR
ATTN: N PASSINO

THE TITAN CORPORATION
ATTN: M ROSENBLATT

TOYON RESEARCH CORP
ATTN: J ISE

TRW INC
ATTN: ED SIMMONS

TRW SPACE & DEFENSE SECTOR
ATTN: D M LAYTON

USER SYSTEMS, INC
ATTN: S W MCCANDLESS JR

UTAH STATE UNIVERSITY
ATTN: K BAKER
ATTN: L JENSEN

VISIDYNE, INC
ATTN: J CARPENTER

FOREIGN

FOA 2
ATTN: B SJOHOLM

DIRECTORY OF OTHER

BOSTON UNIVERSITY
ATTN: MICHAEL MENDILLO