



*M. Nesenberg*  
*(WORK COPY)*

MAY 1975

USPS 1702-113

**STUDY OF ERROR CONTROL CODING  
FOR THE U.S. POSTAL SERVICE  
ELECTRONIC MESSAGE SYSTEM**

**Prepared by  
Martin Nesenbergs  
of the**

**INSTITUTE FOR  
TELECOMMUNICATION SCIENCES  
OFFICE OF TELECOMMUNICATIONS  
U.S. DEPARTMENT OF COMMERCE  
BOULDER, COLORADO 80302**

**RESEARCH & DEVELOPMENT DEPARTMENT**

STUDY OF ERROR CONTROL CODING FOR THE U.S. POSTAL SERVICE  
ELECTRONIC MESSAGE SYSTEM

Prepared by

Martin Nesenbergs

of the

Institute for Telecommunication Sciences  
Office of Telecommunications  
U.S. Department of Commerce  
Boulder, Colorado 80302

Prepared for

Advanced Mail Systems Directorate  
U.S. Postal Service  
Rockville, Maryland 20852

May 1975

USPS/1702-113

PB-252689



## PREFACE

The study presented in this report was supported by the Advanced Mail Systems Development Directorate of the U.S. Postal Service, Rockville, Maryland 20852, under Agreement Number 74-01237. The U.S.P.S. technical monitor for the program was Mr. Ralph P. Marcotte of the Systems Division. The program director at the Institute for Telecommunication Sciences, Office of Telecommunications was Dr. Peter M. McManamon.

Other staff members of the Institute for Telecommunication Sciences who assisted in technical review were Dr. Arthur D. Spaulding and Mr. John J. Juroshek. Typing and drafting assistance were provided by Ms. Patricia A. Moreno and Ms. Mary McClanahan.

The opinions expressed herein are those of the author and are not to be construed as representing policies or doctrines of the U.S.P.S.

## TABLE OF CONTENTS

	Page
PREFACE	iii
LIST OF FIGURES	vii
LIST OF TABLES	x
GLOSSARY	xi
ABSTRACT	1
1. INTRODUCTION	1
2. OVERALL CODEC ALTERNATIVES	3
2.1. Candidate Codes	3
2.2. The Second and Third Finalists	5
3. INNER CODE FOR THE CONCATENATED HYBRID	8
3.1. Convolutional Codes	8
3.2. Systematic and Nonsystematic Aspects	13
3.3. Decoder Types	14
3.4. Viterbi Decoding	21
3.5. Performance on the Ideal Channel	29
3.6. Performance on Realistic Channels	32
4. OUTER CODE FOR THE CONCATENATED HYBRID	39
4.1. Codec	39
4.2. ARQ Performance	45
5. OVERALL PERFORMANCE	66
6. CONCLUSIONS	71
7. REFERENCES	74



## LIST OF FIGURES

		Page
Figure 1.	Hybrid FEC and ARQ scheme with concatenated coding.	9
Figure 2.	Convolutional encoder of rate $2/3$ , constraint length 48, with 10 delay taps for each parity check.	11
Figure 3.	Systematic and nonsystematic versions of rate $1/2$ convolutional encoder.	12
Figure 4.	State diagram of the systematic code.	15
Figure 5.	State diagram of the nonsystematic code.	16
Figure 6.	Basic feedback decoder for the rate $1/2$ code.	19
Figure 7.	Example of a feedback decoder - the MLD decoder for rate $2/3$ convolutional code.	20
Figure 8.	Trellis diagram and encoder outputs.	24
Figure 9.	The role of branch metric/best path metric pairs in successful decoding of three errors.	26
Figure 10.	Unsuccessful decoding of five errors.	28
Figure 11.	Various convolutional code performances on the ideal theoretical channel.	30
Figure 12.	Definition of coding gains and modem losses.	34
Figure 13.	Viterbi decoding gains on the ideal and extrapolated-measured channels.	36
Figure 14.	Actual coding gains for rate $1/2$ and $3/4$ , soft decision, Viterbi decoders.	37

Figure 15.	The effects of irreducible error probability.	38
Figure 16.	Outer encoder.	41
Figure 17.	Outer decoder.	42
Figure 18.	The ARQ duplex loop.	46
Figure 19.	Synopsis of ARQ.	48
Figure 20.	The least and most repeats in the ARQ loop.	50
Figure 21.	The least and most errors in the ARQ loop.	51
Figure 22.	The bounds on throughput rate for $n=10,000$ and $k=9,900$ .	54
Figure 23.	The bounds on throughput rate for $n=10,000$ and $k=9,950$ .	55
Figure 24.	The bounds on throughput rate for $n=10,000$ and $k=9,990$ .	56
Figure 25.	The bounds on throughput rate for $n=31,600$ and $k/n=0.99$ .	58
Figure 26.	The bounds on throughput rate for $n=100,000$ and $k/n=0.99$ .	59
Figure 27.	The bounds on throughput rate for $n=316,000$ and $k/n=0.99$ .	60
Figure 28.	The bounds on throughput rate for $n=1,000,000$ and $k/n=0.99$ .	61
Figure 29.	Output word error probability for $n=10,000$ and $k/n=0.999$ .	63
Figure 30.	Output word error probability for $n=31,600$ and $k/n=0.999$ .	64
Figure 31.	Output word error probability for $n=10,000$ and $k/n=0.995$ .	65



- Figure 32. Estimated overall throughput rate for specific burst channels ( $v$ ), code length ( $n$ ), and signal-to-noise ratio ( $E_b/N_0$ ). 69
- Figure 33. Guideline for rough estimation of the overall throughput rate. 70

## LIST OF TABLES

	Page
Table 1. Comparison of Hybrid, FEC and ARQ, overall coding alternatives. Code rates are assumed in the 0.50-0.75 range.	6
Table 2. Comparison of forward acting decoders for convolutional inner code candidates. Code rates are assumed in the 0.50-0.75 range.	17
Table 3. Possible error detecting codes for ARQ.	44



## GLOSSARY

$A_k$	k-th acknowledgement
ARQ	Automatic repeat request
codec	(en)coder and decoder
d	minimum distance, block code
D	path delay
$d_f$	free distance, convolutional code
$E_b$	energy per information bit
FEC	forward error correction
$G_a$	actual coding gain
$G_b$	basic coding gain
k	number of information bits, block code
K	constraint length, convolutional code
L	path merge length, Viterbi decoding
$L_c$	modem loss, coded
$L_u$	modem loss, uncoded
M	number of phases, M-ary phase shift keying
MLD	majority logic decision
modem	modulator and demodulator
MPSK	M-ary phase shift keying
n	length of codeword, block code
N	noise power
$N_0$	noise power spectral density
$N(k)$	number of k's
v	burst length parameter
p	binary error probability, overall output
P	word error probability, overall output
$p_e$	binary error probability, data channel or in general context

$p_I$	binary error probability, inner decoder
$P_0$	word error probability, outer decoder
$p_n(j)$	probability that $j$ out of $n$ bits are in error
$\text{Pr}(\dots)$	probability of event (...)
PSK	phase shift keying
$Q$	state qualifier for ARQ
$R$	overall throughput rate
$R_I$	rate of the inner code
$R_0$	rate of the outer code
$S$	signal power
$W$	bandwidth
WGN	white Gaussian noise
$W_k$	$k$ -th word



STUDY OF ERROR CONTROL CODING FOR THE U.S. POSTAL SERVICE  
ELECTRONIC MESSAGE SYSTEM

Martin Nesenbergs\*

ABSTRACT

A U.S. Postal Service (USPS) electronic message system could incorporate many types of error control coding, or no coding at all. This report reviews a variety of possible codes, lists their advantages and disadvantages, and selects a preferred alternative. It turns out to be a concatenation of an inner convolutional (rate 1/2 to rate 3/4) code with Viterbi decoding, and an outer long block, high efficiency code. The two codes have separate functions, in the sense that the inner code performs forward error correction and the outer code does error detection only. The report describes the structures, properties, and implementations of the coding hybrid. After that, the performance of the preferred coding scheme is estimated. The resultant error probability gains, which are shown to be considerable, are balanced against system slowdown and bandwidth expansion.

Key words: ARQ, coding gains, concatenated codes, error probability, FEC, hybrid operation, modem losses, throughput, Viterbi decoding

1. INTRODUCTION

Digital communication via satellite has been considered for the USPS electronic message system. Extensive background studies (McManamon et al., 1974) have been directed at the large scale system features, such as

---

\*The author is with the Institute for Telecommunication Sciences, Office of Telecommunications, U.S. Department of Commerce, Boulder, Colorado 80302



useful frequency ranges, present and future traffic loads, network configurations, transmission impediments, and state of the art engineering tradeoffs. System design, operation, and performance verification have to wait until basic feasibility questions are answered in the affirmative. One such question concerns the quality of the received output data. Under the heading "System Performance", this multi-faceted issue was raised in section 7, volume II of McManamon et al. (1974).

The expected data rates, bandwidths, signal powers, and the noise densities described above revealed that perhaps the hardest performance criterion to be satisfied is the bit error rate, also called error probability. A target value of  $10^{-12}$  was given for this error rate, but at the time of the above report, it was not entirely clear if or how this could be met. Section 7.8, volume II of McManamon et al. (1974) suggested error control codes as a possible alternative to ensure sufficiently low error rates. Unfortunately, being quite brief and introductory in nature, the above treatment raised more questions about coding than it managed to answer. It also overlooked some important and useful coding techniques.

Hence, the purpose of this study - a second look at the error control codes as they could be applied today to a high rate satellite data network. The approach taken is (a) to consider all promising alternatives, (b) to identify the most promising alternative as the "first choice", (c) to verify that the first choice can be practically implemented and operated, and (d) to estimate the performance, i.e., gains and losses, of the proposed scheme.



Before proceeding with the coding tasks, we must emphasize a few key points. The data network is assumed to consist of two-way links over which forward and feedback messages can be sent, if needed. The data terminals are of sufficient size to justify the initial cost and maintenance of the codecs (short for encoder-decoder). The implementation of the coding part of the terminal will be shown as relatively modest, of complexity somewhere between a minicomputer and a microprocessor. Of course, the device can always be incorporated in a large general purpose computer (if such is available), or it can be constructed as a special unit. Finally, the codecs are stipulated to function perfectly, so as not to introduce additional errors in the received messages.

## 2. OVERALL CODEC ALTERNATIVES

### 2.1. Candidate Codes

The question of whether to use any error control coding at all cannot be answered without examination of the more promising codec alternatives. This section compares the relative advantages and disadvantages of the best presently known coding schemes, and then ventures to select a "first choice" overall error control system for a U.S. Postal Service network. Perhaps even the best choice is not good enough. Accordingly, later sections will scrutinize the details of the candidate codecs, with emphasis on implementation, operation, performance, and anticipation of sundry difficulties.

The presence of one-way, two-way, and more elaborate data channel arrangements within the satellite network



permits forward-acting error correction (FEC), automatic error detection and repeat requests (ARQ), and any hybrid combination of the two. Furthermore, most channel arrangements can function well with a variety of known codec types (Gallager, 1968; Massey, 1973; Wolf, 1973; Sastry, 1974). The assortment of codes can be divided into three broad classes: the block codes (Berlekamp, 1968; Peterson and Weldon, 1972), the convolutional or tree codes (Wozencraft, 1957; Viterbi, 1971), and all the other codes. Among the others, it is befitting to single out concatenated codes (Forney, 1966; Zeoli, 1971; Hoffman and Odenwalder, 1972), which are constructed interactively from two or more codes.

A previous investigation (McManamon et al., 1974) indicated that a hybrid, FEC and ARQ, with joint correction and detection roles, may be preferable over other overall coding schemes for reasons of flexibility and error correcting performance. Basically, a hybrid arrangement can do nearly everything that the pure FEC and ARQ options can do. When transmission conditions are good and channel errors occur seldom, the hybrid resembles FEC in throughput rate and error probability performance. When the channel deteriorates and more errors confront the decoder, the hybrid can be relied on to reject error infested data portions. The hybrid then turns to ARQ action and asks for repetition. No matter how low the signal-to-noise ratio, or whatever the channel malfunction, the output error probability can be kept at or below a level determined by the code. The principles, implementation, and estimated performance of ARQs appears the same for high and low speed data links. The low speed cases have been extensively reviewed by van Duuren (1961), Nesenbergs (1963), Townsend and Watts (1964), Benice and Frey (1964),



and Burton (1970). From now on we will assume that some hybrid scheme is the most suitable for the high speed digital message network.

Our next task is to compare the alternative implementations of the hybrid scheme. Here we have a number of options, including the previously mentioned block code with correction threshold (McManamon et al., 1974). Besides that, there are quite a few others. A selected list is tabulated in table 1. The table applies to code rates in the 0.50 to 0.75 range, assuming negligible ARQ slowdowns. The left hand column of table 1 lists six alternatives for the hybrid codec. The other five columns summarize the key codec characteristics in an informative way. The final, i.e., the rightmost, column attempts to rate the hybrid schemes in order of their suitability for the system. Actually, only the first three most promising schemes are seriously contemplated; the others are dismissed as being not suited.

The three leading candidates are:

- (1) Concatenated inner convolutional code and an outer block code.
- (2) Concatenated inner convolutional code with an outer convolutional code.
- (3) Convolutional code with computation threshold.

## 2.2. The Second and Third Finalists

Since considerable subsequent discussion will deal with alternative (1), the intent here is to comment briefly on choices (2) and (3). The second finalist (2) differs from (1) only in the non-block nature of the outer code. This is not a serious drawback, as the error detecting power of the outer code is seemingly not lessened. The

Table 1. Comparison of Hybrid, FEC and ARQ, overall coding alternatives. Code rates are assumed in the 0.50-0.75 range.

Codec Type	CHARACTERISTIC					Relative Rating
	Error Rate Performance	Experimental Evidence	Logic Speed Data Speed	Cost and Complexity	Others	
Block code with correction/detection threshold	Poor	None known	10	Very high	Needs interleaving for deep fades	Not suited
Convolutional code with computation threshold	Reasonable to good	None known	3	Medium	ARQ sync problem	3
Concatenated code of inner block and outer block codes	Poor to mediocre	Some	10	High	Interleaving may help	Not suited
Concatenated code of inner convolutional and outer block codes	Nearly optimal	Well tested	1-3	Low		1
Concatenated code of inner block and outer convolutional codes	Unknown	None known	10	High	ARQ sync problem	Not suited
Concatenated code of inner convolutional and outer convolutional codes	Perhaps good	None known	1-3	Reasonable	ARQ sync problem	2



only problem concerns frame timing of repeat segments in the ARQ operation. The encoded sequence keeps continuously evolving without any punctuation marks. Some reference points, called commas, would have to be implanted for identification of repeat segments. Both transmitter and receiver must have exact knowledge where the repeats start and end, and which are multiple repeats. Otherwise, the inner code of (2) can be the same as in (1).

The third finalist (3) is a non-concatenated convolutional code with a modified decoding algorithm. Consider an early form of a sequential decoder (Wozencraft, 1957; Wozencraft and Reiffen, 1961; Fano, 1963; Savage, 1966; Jacobs, 1967; Gallager, 1968; Forney and Langelier, 1969). This device seems to be quite capable of forward acting error correction, as long as the error rate is sufficiently low. As the error rate increases, however, the computational job of the sequential decoder grows. By necessity, every decoder can handle a certain computation rate and not more. When the noisy input data warrants computation in excess of the computational ceiling, input data would be lost if not for buffer storage. Unfortunately, a buffer of finite size carries a certain probability of overflow and associated implications of output errors or data loss. Both theory and practice have shown that the buffer overflow is of primary concern in the design and operation of sequential decoders.

In scheme (3) one proposes to do the following. Permit a small buffer storage only. When the computation load grows and indicates impending overflow, let the ARQ operation take over. Clearly, this strategy can be used with different types of decoders, including simplified versions of the sequential machine itself. Since sequential decoding of convolutional codes is a powerful (if not cheap) error



control tool, one expects that scheme (3) can be engineered to perform well. The only nagging doubts concern the previously mentioned ARQ frame timing and identification for convolutional coding, and the lack of experimental evidence for this particular hybrid technique.

### 3. INNER CODE FOR THE CONCATENATED HYBRID

#### 3.1. Convolutional Codes

In this section we start a detailed scrutiny of the preferred alternative for the concatenated hybrid FEC and ARQ scheme, that is, the first choice of table 1. Specifically, we are concerned with the inner code in the concatenated arrangement shown in figure 1.

A forward error correcting convolutional code seems to be the best choice here. In fact, the FEC convolutional coding systems that have been built, simulated, and/or tested show an abundance of advantages over comparable rate block codes (Kohlenberg and Forney, 1968; Heller, 1968, 1969; Forney, 1970; Clark, 1971; Massey and Costello, 1971; Viterbi, 1971; Cain, 1972; Cahn et al., 1973; Jacobs, 1974). The crucial points can be summarized as follows:

- (a) Convolutional coders are state-of-the-art now. Systems have been built and tested in sufficient numbers to feel quite confident about these devices. One manufacturer has a number of ready made versions on the market. While other type FEC coders appear to require further research and development, good convolutional coders are already here.



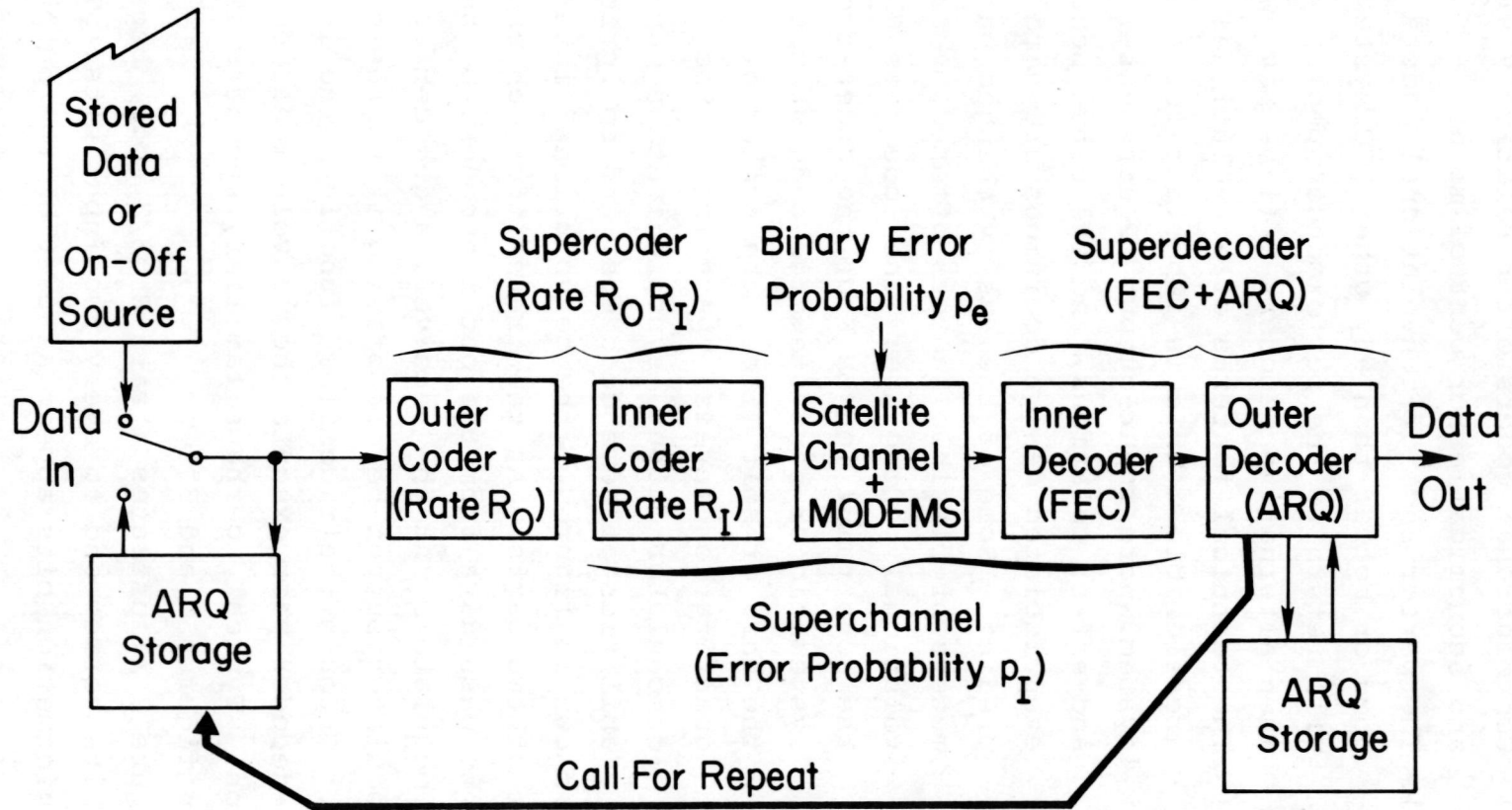


Figure 1. Hybrid FEC and ARQ scheme with concatenated coding.

- (b) Convolutional coders offer relatively simple and economical error control means.
- (c) The performance of convolutional coders is as good or better than any other FEC system. We will return to the performance specifics later, when alternative decoders will be reviewed.
- (d) The technical features of convolutional codecs are flexible. This is important for a large system where input-output requirements, cause-and-effect relations, and the consequences of even isolated design decisions are hard to predict. Some of the easily changed parameters or flexible factors offered by convolutional coding are: the code rate  $R_T$ , the constraint length  $K$  of the code, the systematic or nonsystematic code structure, the hard or soft (i.e., the number of quantization levels) operation at the demodulator output, synchronization capabilities of some decoders, and so forth.

As shown in figure 1, the convolutional inner codec consists of two parts. At the transmitter, an encoder is used to pass data from the outer encoder to the satellite channel modulator. At the receiver, a decoder processes the demodulator output before passing it to the outer decoder. Through their combined function, the inner encoder-decoder pair execute the convolutional coder operation. Typical, perhaps familiar, encoders are shown in figures 2 and 3.

Figure 2 illustrates a rate  $R_T=2/3$  code, because three output bits correspond to every two input bits. The two input information bits appear as part of the output, while a third bit checks the parity over selected past



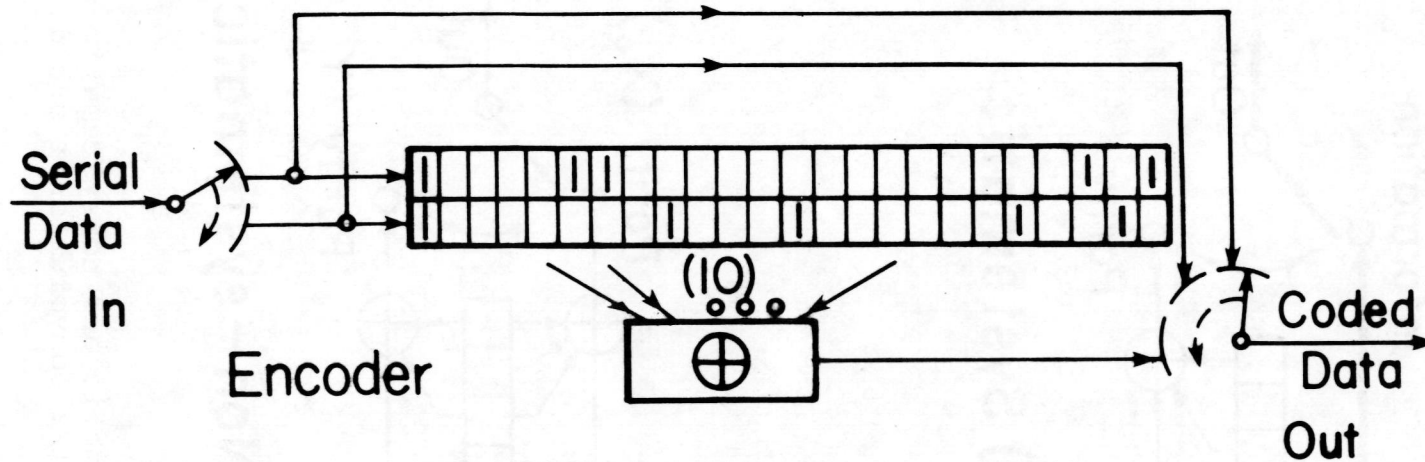
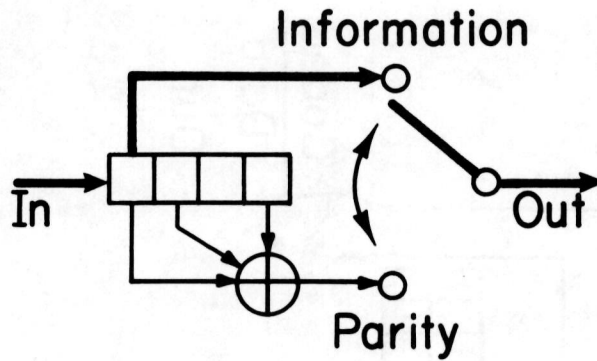
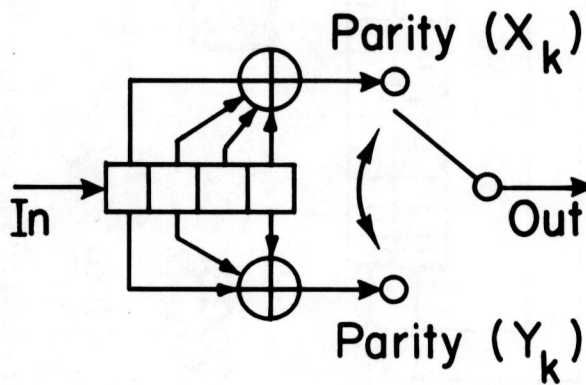


Figure 2. Convolutional encoder of rate 2/3, constraint length 48, with 10 delay taps for each parity check.



**(A) Systematic**



**(B) Non-systematic**

Figure 3. Systematic and nonsystematic versions of rate 1/2 convolutional encoder.



information bits. As shown, the parity check includes exactly ten previous bits extending as far back as 48 information bits, or 72 total bits. The former is the number of memory cells in the delay shift register shown in figure 2, and is known as the (encoding) constraint length  $K$  of the code. The code is said to be systematic, because unperturbed input bits can be found at prescribed places in the output sequence.

### 3.2. Systematic and Nonsystematic Aspects

The definition of a systematic convolutional code is further elaborated in figure 3. Both parts (A) and (B) show rate  $R_T=1/2$  and constraint length  $K=4$  convolutional encoders. Encoder (A) contains the original information bits among its outputs; it is systematic. Encoder (B) is nonsystematic, even though a mod-2 gate of outputs,  $X_k \oplus Y_k$ , produces a delayed replica of the input sequence. The reason for concern with these matters is to find the most powerful convolutional codes, and not waste time on the bad ones. It has been established (Bussgang, 1965; Lin and Lyne, 1967; Heller, 1969; Massey and Costello, 1971; Gilhausen et al., 1971, 1972; Forney, 1972, 1973; Massey, 1973; Paaske, 1974) that the performance of the best nonsystematic convolutional codes is superior.

Since for block codes the two code families are often indistinguishable, some explanation seems in order. The performance of a convolutional code over any channel depends largely upon the relative distances between codewords and, in particular, upon the so called minimum free distance,  $d_f$ , of the code. This  $d_f$  is defined as the least number of 1's that can occur over all non-trivial closed paths passing through the all-zero state in the state diagram of that code

(Viterbi, 1967; Massey and Costello, 1971; Peterson and Weldon, 1972; Forney, 1973). The state diagram of example (A), figure 3, is shown in figure 4, and that of (B) in figure 5. In the case of the systematic code (A), the passage through states 000, 100, 010, 001, and back to 000, produces outputs 11, 01, 00, and 01. There are four 1's among the outputs, and no other closed non-trivial path yields less. Hence, the free distance  $d_f=4$  for the systematic code (A). Furthermore, it can be shown that no other systematic code with the same rate and constraint length can have a larger free distance. In figure 5, one traverses a different path through the states; namely, 000, 100, 110, 011, 001, and 000. The output sequence 11, 00, 01, 01, 11 implies  $d_f=6$  for this nonsystematic code. It turns out that  $d_f=6$  is the best that one can do with  $R_I=1/2$  and  $K=4$ .

### 3.3. Decoder Types

The choice of constraint length  $K$  depends on the decoder type used in practice. There are three prominent decoding techniques for convolutional codes:

Sequential decoding

Feedback decoding

Viterbi decoding

All three offer their peculiar variations, refinements, as well as advantages and disadvantages. A summary of the leading options is given in table 2. As before, the table is a review of pertinent inner code decoders and their characteristics. The justification of the claims is found in appropriate references (which includes most of the list).



Convolutional,  
 Best ( $d_f = 4$ )  
 Systematic,  
 Rate 1/2,  
 Constraint  
 Length  $K = 4$

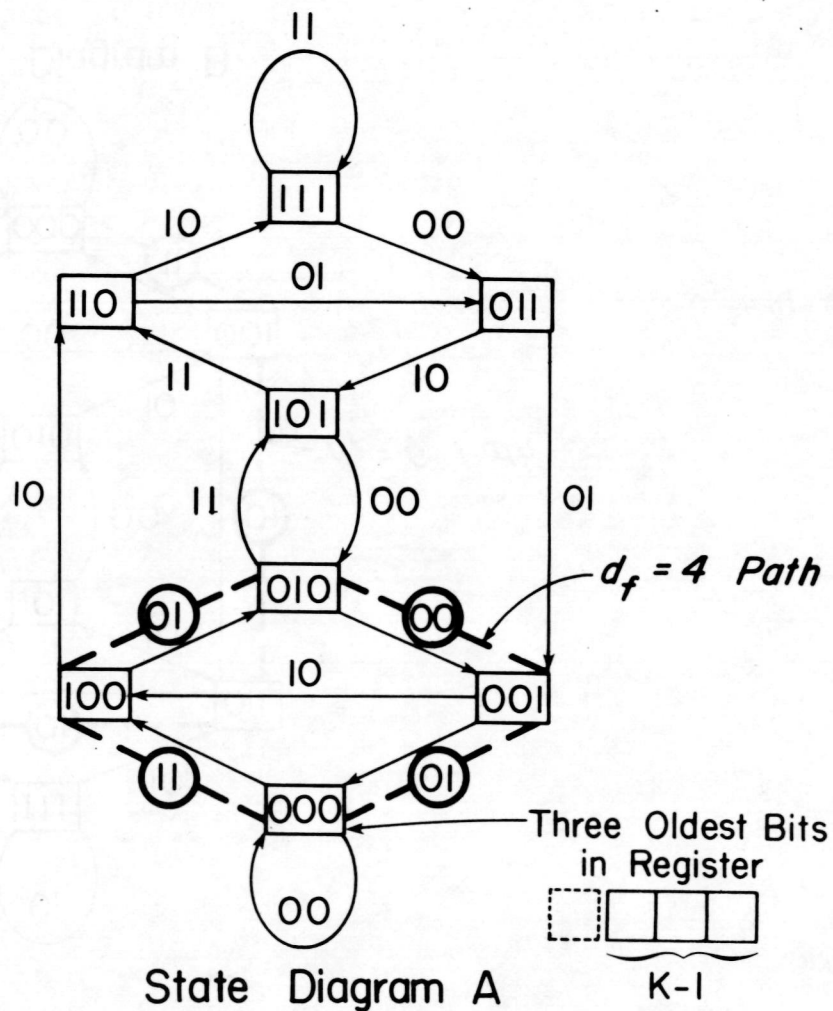


Figure 4. State diagram of the systematic code.

Convolutional,  
 Best ( $d_f = 6$ )  
 Non-systematic,  
 Rate 1/2,  
 Constraint  
 Length  $K = 4$ .

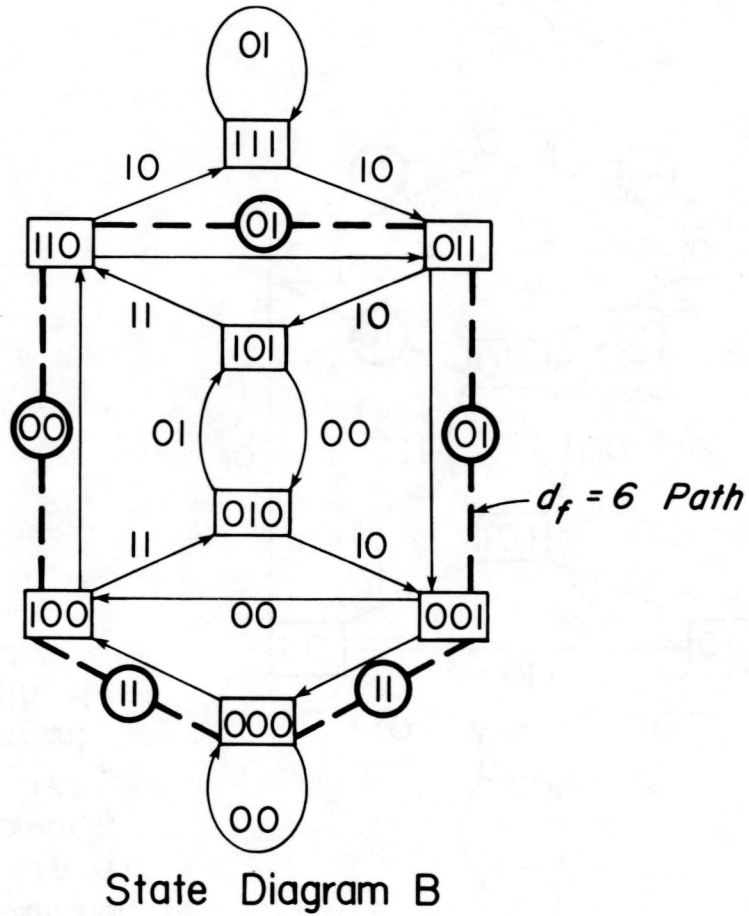


Figure 5. State diagram of the nonsystematic code.



Table 2. Comparison of Forward Acting Decoders For Convolutional Inner Code Candidates. Code Rates Are Assumed in the 0.50-0.75 Range

Decoder Type	CHARACTERISTIC					Relative Rating
	Error Rate Performance	Experimental Evidence	$\frac{\text{Logic Speed}}{\text{Data Speed}}$	Cost and Complexity	Others	
Sequential with Fano algorithm	Great	Some tests	2-4	Extremely high		4
Sequential with stack, bucket, and other variations	Great	Some tests	1.5-2	Quite high	Further progress likely	3
Feedback (general)	Poor		1	Low		5
Feedback (threshold or MLD)	Very poor	Some tests	1	Very low		6
Viterbi with hard decisions	Good	Well tested	2-3	Reasonable	Many good options	2
Viterbi with soft decisions	Nearly optimum and robust	Well tested	2-3	Reasonable	Many good options	1



Sequential decoding evolved from the original work of Wozencraft (Wozencraft, 1957; Wozencraft and Reiffen, 1961), to the Fano algorithm (Fano, 1963; Massey and Sain, 1968; Gallager, 1968), to later stack, bucket, bootstrap, and other variations (Zigangirov, 1966; Jelinek, 1969; Massey, 1973). All of these techniques can work with long constraint lengths and so produce powerful error correction performance. The performance has been verified by tests and/or simulation (Forney, 1967; Forney and Langelier, 1969; Lumb, 1969; Cain, 1971; Layland and Lushbaugh, 1971; Forney and Bower, 1971; Gilhausen et al., 1971; Gilhausen and Lumb, 1972; Odenwalder et al., 1972; Cahn et al., 1973; Dodds, 1973). More recent sequential decoding versions have sought to reduce the computation load, the associated logic speed and the memory requirements, while maintaining high level of performance. To a large extent, such objectives have been met by the Zigangirov (1966) and Jelinek (1969) stack method, and other recent extensions. Unfortunately, the complexity of sequential decoders is still large and burdensome.

A far simpler family of convolutional decoders are the feedback decoders, and the related threshold or majority logic decision (MLD) decoders (Massey, 1963; Rudolph, 1967; Gallager, 1968; Goldman, 1969; Peterson and Weldon, 1972; Massey, 1973; Wolf, 1973). A general feedback decoder is illustrated in figure 6. Note the crucial role of the syndrome register. Through a usually prewired or fixed read-only storage logic, the syndrome dictates output correction as well as its own update, called "feedback." A special case of the feedback decoder is the fixed threshold count-of-1's MLD decoder. Such an MLD device is shown in figure 7. This decoder happens to decode the



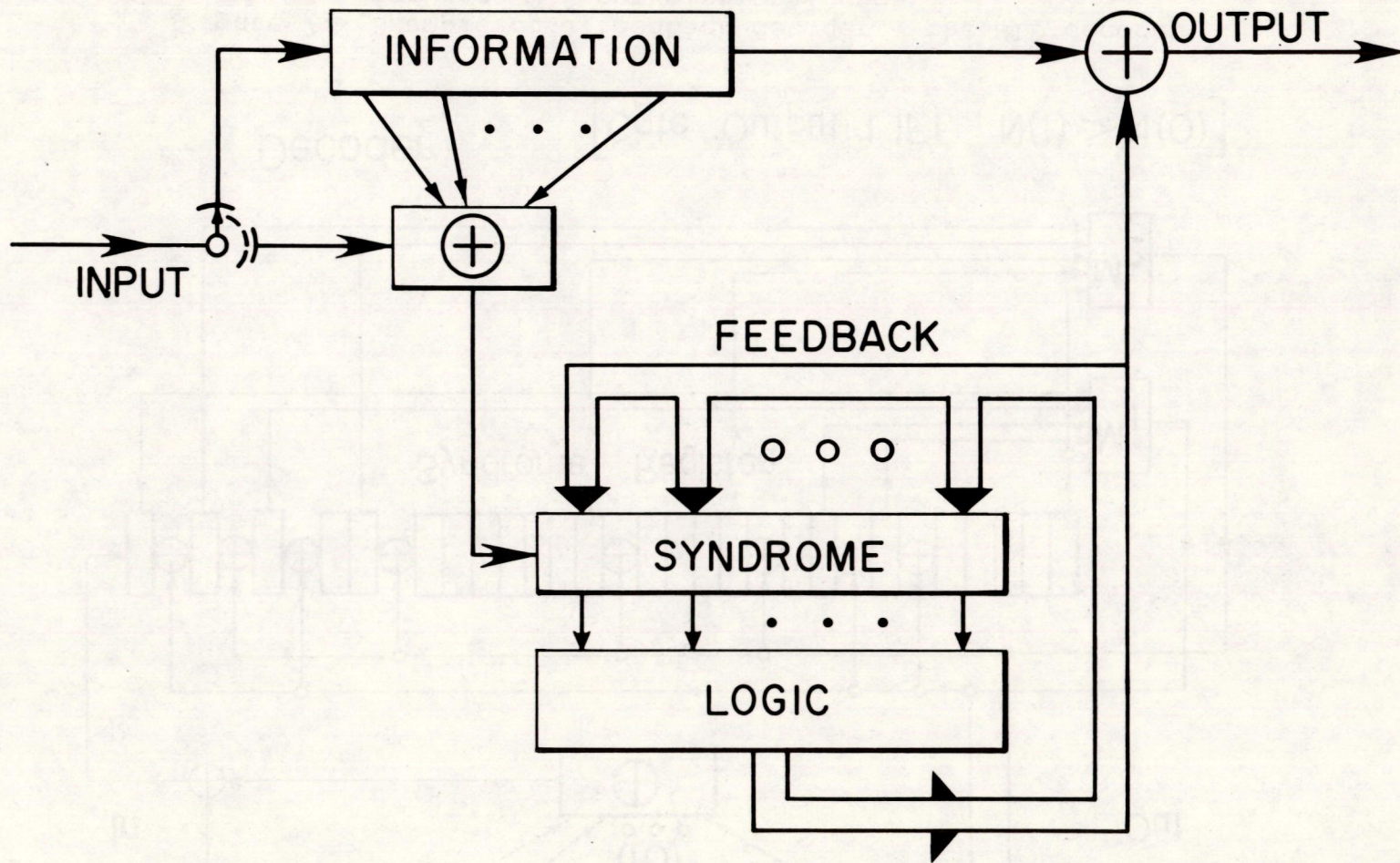
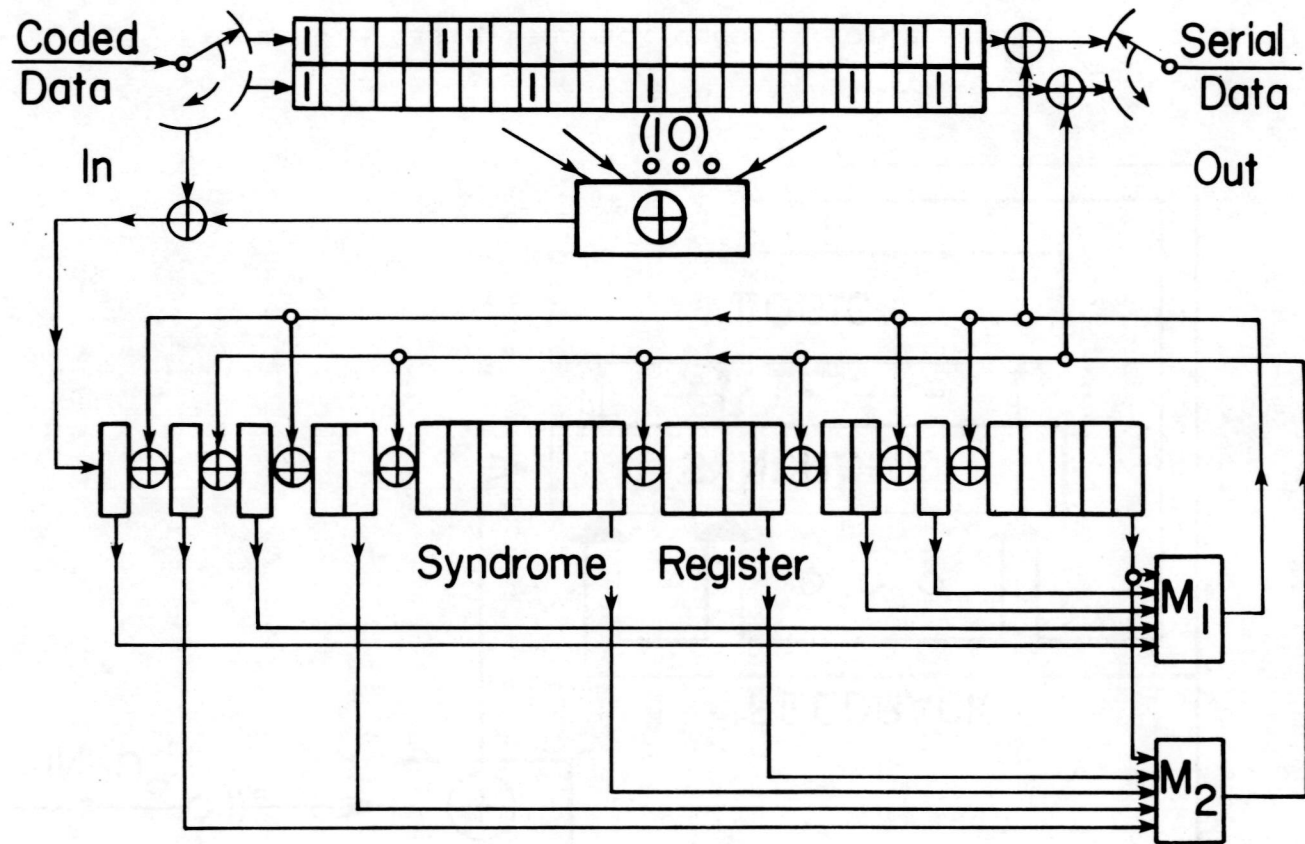


Figure 6. Basic feedback decoder for the rate 1/2 code.



Decoder

[Gate Output 1 IFF  $N(1) > N(0)$ ]

Figure 7. Example of a feedback decoder - the MLD decoder for rate 2/3 convolutional code.



rate 2/3 code introduced earlier in figure 2. The two threshold gates  $M_1$  and  $M_2$  serve the two parallel information registers. The function of the two gates is an identical majority poll of the five inputs shown. Thus, if the number of input 0's is  $N(0)$  and the number of input 1's is  $N(1)$ , then

$$\begin{aligned} \text{Gate Output} &= 0 && \text{if } N(0) > N(1), \\ &= 1 && \text{if } N(0) < N(1). \end{aligned}$$

A 1 at the gate output inverts the appropriate information bit and, by feedback action, deletes either

$$N(1) - N(0) \geq 1$$

or

$$N(1) - N(0) - 1 \geq 0$$

net 1's from the syndrome register.

The codec consisting of encoder (fig. 2) and decoder (fig. 7), implements a rate  $R_I=2/3$  self-orthogonal code. Its operation is based on, so called, "disjoint and full" difference triangles (Massey, 1963; Peterson and Weldon, 1972):

		21				19				
	16		6			11		12		
	15	1		5		7	4		8	
2	17		18		23	3	10		14	22

The free distance of this code  $d_f=6$  is nearly optimal for such self-orthogonal codes, and yet only two errors are always correctable among 72 channel bits (Robinson and Bernstein, 1967). This error correcting scheme is not very powerful (it further worsens as constraint length increases), but the codec cost and complexity is extremely low.

### 3.4. Viterbi Decoding

The preferred alternative in table 2 uses the Viterbi decoding algorithm (Viterbi, 1967, 1971; Heller, 1968,

1969; Omura, 1969; 1971; Clark and Davis, 1971; Clark, 1971; Heller and Jacobs, 1971; Kobayashi, 1971; Viterbi and Odenwalder, 1972; Batson et al., 1972; Forney, 1972; 1973; Massey, 1973; Wolf, 1973; Jacobs, 1974). One must distinguish two cases. The "hard decision" case refers to demodulator output and decoder input being an ordinary binary decision, zero or one. The "soft decision" admits more than two quantized demodulator outputs. A typical case of eight quantization levels is:

```

000 ----- a sure zero,
001
010
011 ----- a doubtful zero,
100 ----- a doubtful one,
101
110
111 ----- a sure one.

```

There are Viterbi decoders with an incorporated switch to either use or omit the less significant quantization digits. Thus, an operator can choose between the hard and soft options. Both options have been explored and tested. While the speed requirements and costs are comparable, one finds the soft version superior in performance.

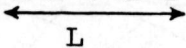
Before turning to the performance numbers of the various alternatives, it may be beneficial to get an elementary grasp of the operation and implementation of our first choice - the Viterbi decoding system. A simple illustration can start with the encoder of figure 3, part (B). Here, the (encoder) constraint length is  $K=4$ , roughly a half of the  $K=6$  to  $K=10$  values used in actual systems. The rate  $R_I=1/2$  code is nonsystematic, with the state diagram given in figure 5. Each input bit directs the system from a



previous state to the next state. If the present state is 110, then input 0 causes a transition to state 011 and input 1 to state 111. The output bits produced by register content 0110 are 01, and so forth. These output bits are shown in the state diagram (fig. 5) next to the transition arrows between states.

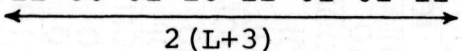
The trellis diagram is an equivalent, somewhat longer, and, in a way, a more vivid description of the encoding process. The corresponding trellis diagram is shown in figure 8. Note again how a single input bit 0 takes the encoder from state 110 to state 011 and produces output 01. A length  $L$  non-zero input sequence gives forth a  $2(L+3)$  long output. An example of  $L=5$ ,

$$\dots 0 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0 \dots$$

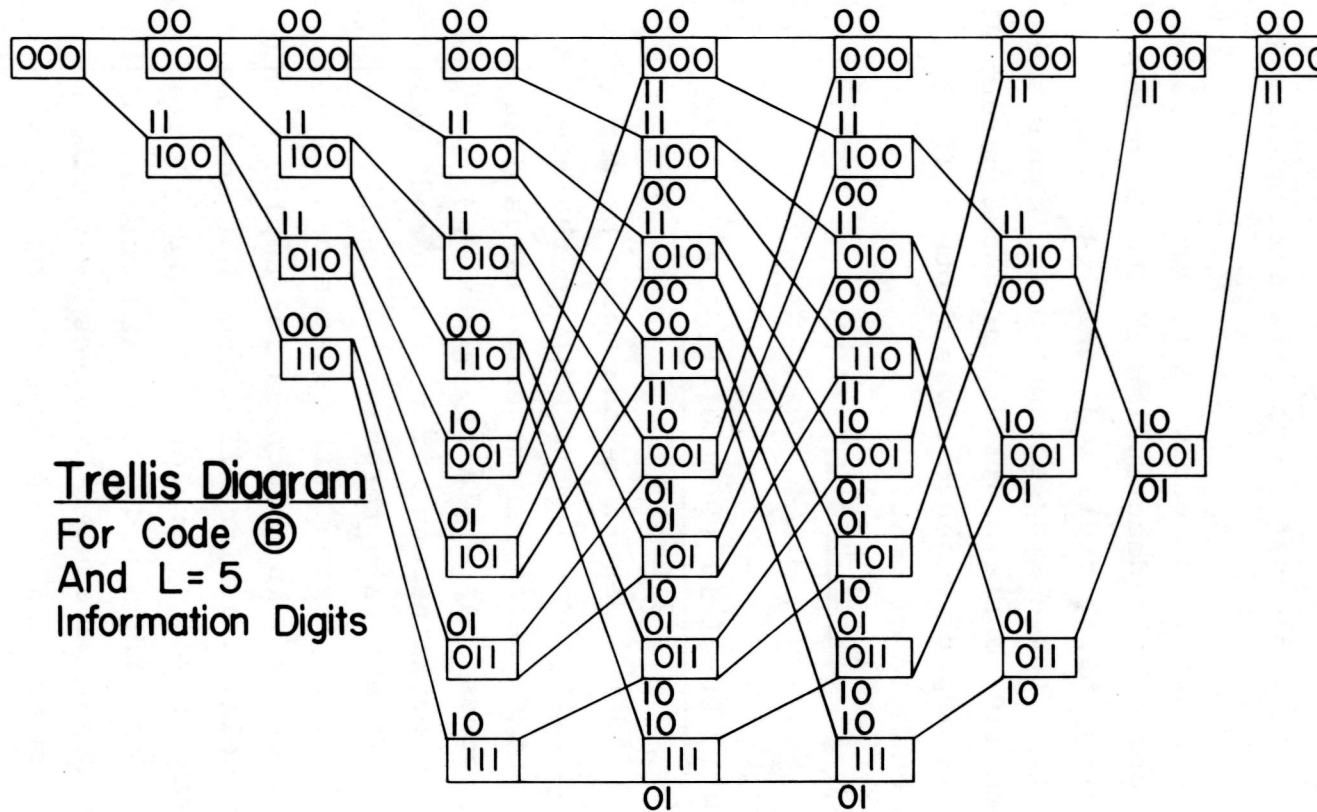


yields an output

$$\dots 00 \ 11 \ 00 \ 01 \ 10 \ 11 \ 01 \ 01 \ 11 \ 00 \dots$$



The extension of  $L$  to  $L+3$  describes the merge properties of a non-zero path from the all-zero path in the trellis. It is also an important delay parameter to be used later in the decoding process. As explained elsewhere, the Viterbi algorithm ascertains and, if need be, modifies the estimated most likely path through the trellis. With a diminishing probability, some path updates can take place after considerable delay. In communication application, outputs cannot be delayed arbitrarily long, but must be delivered after some fixed -- preferably small -- delay. It turns out that, through a process called merging of paths, the finite delay parameter  $L$  serves to terminate the decoder path memory, and to deliver the output bits on schedule. The effect of various  $L$  choices has been



**Trellis Diagram**  
 For Code (B)  
 And L=5  
 Information Digits

Encoder Outputs.

Example: When data bit 0 changes state  
 110 to 011, the output is "01"

Figure 8. Trellis diagram and encoder outputs.



determined by Gilhausen et al. (1971). It is shown that  $L=4K$ , which is a delay four times the constraint length, for all practical purposes gives as much performance gain as can be expected of the arbitrarily long memory. In the previous example of figure 3 (B) and figure 8, the  $L=5$  value is thus too short for best results, and entails at least a 1 dB signal-to-noise ratio loss. A higher parameter value, such as  $L \geq 16$ , should be used to avoid the loss.

The decoder selects the most likely path by continuously keeping track of appropriate path likelihood measures, commonly called metrics. The multitude of paths are all constituted of branches, that is, directed links between successive states. When a set of branches combine to form a path, the branch metrics add to form the (total) path metric of that path. As an illustration, consider the hard decision example of figure 8. The bookkeeping of branch and path metrics is shown in figure 9. If the original input data sequence is

0 0 1 1 1 0 0 0,

the encoded and transmitted sequence is

00 00 11 00 10 10 01 11.

For a moment, assume no errors in transmission, so that the received sequence agrees with the transmitted one. A metric that is defined to count disagreements will assign to the all 0's path a sequence of branch metric/path metric pairs

0/0 0/0 2/2 0/2 1/3 1/4 1/5 2/7,

with a final path metric 7. Likewise, the all ( $L=5$ ) 1's path would be encoded as

11 00 10 01 01 10 01 11,

and the metric pairs would evolve as

2/2 0/2 1/3 1/4 2/6 0/6 0/6 0/6,

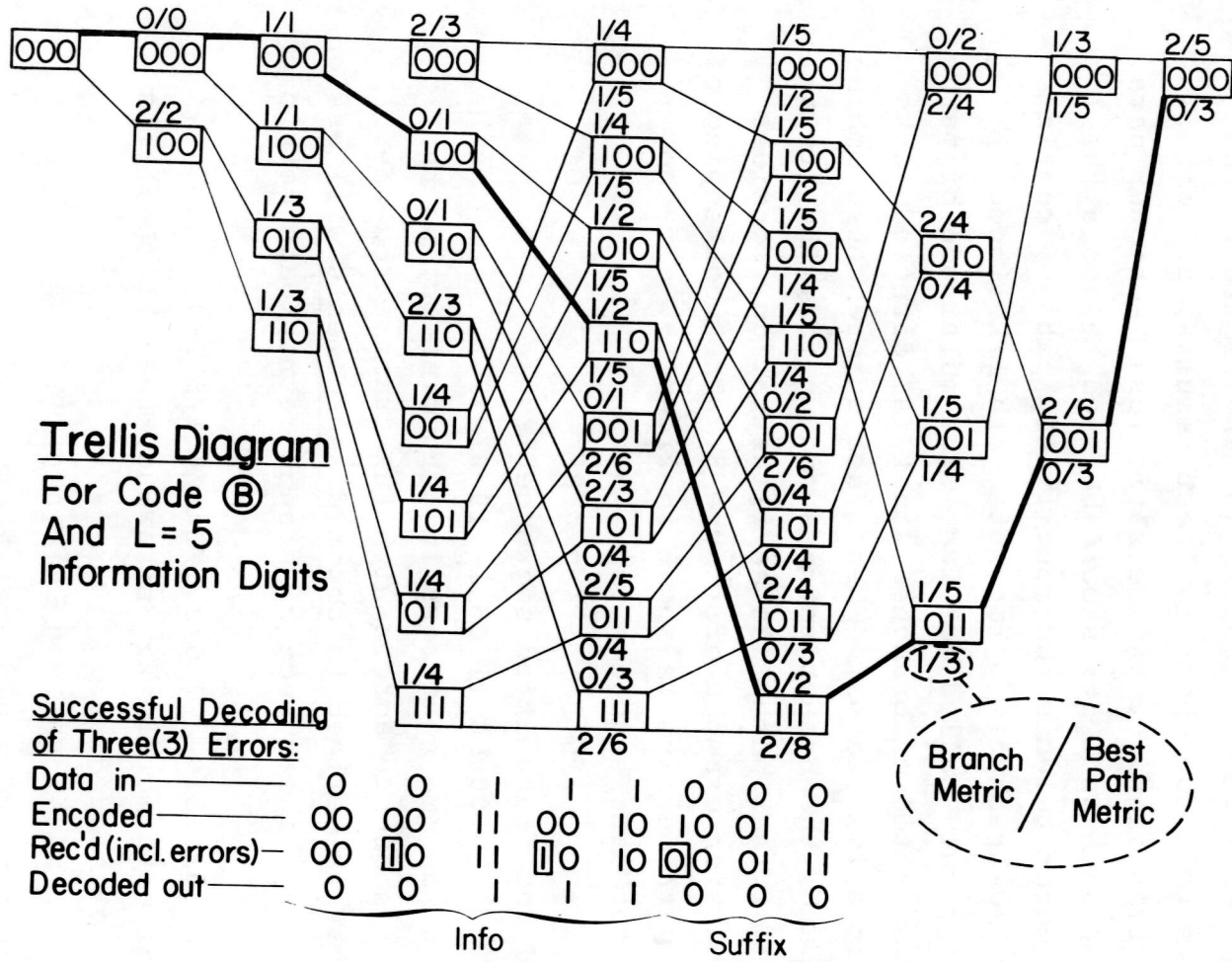


Figure 9. The role of branch metric/best path metric pairs in successful decoding of three errors.



causing the final metric 6. In fact, all conceivable paths will have nonzero metrics, except for the correct path, where only 0/0 metric pairs arise. Note, that the 3-bit suffix does not carry information. Its sole purpose is to merge the paths of the trellis.

Next, permit three errors to infest the same transmitted sequence, as shown by the received sequence

00  $\overline{1}$ 0 11  $\overline{1}$ 0 10  $\overline{0}$ 0 01 11.

From figure 9, one sees how the branch and path metric picture changes. While the all 0's path shows a total metric 8, the best final-zero path now has a metric of 5. Likewise, the all 1's path, with a total metric 9, is worse than the best final-one path. The latter has a path metric of 3, and through back tracing it leads to the actual correct path. Thus, the correct path is unique, it is identifiable, and the correct decoding output is delivered.

Of course, there are other error patterns that cannot be corrected. In figure 10 we show a five error event that causes erroneous output. An incorrect path with a total path metric 2 is selected ahead of the correct path, which turns out to have a path metric 5. As a consequence, the original data sequence

0 1 1 0    1 0 0 0

is incorrectly decoded as

0 1 0 0    1 0 0 0.

This is not too unfortunate, as the five original channel errors are reduced to a single output error. For longer delay parameters L and other error events, one anticipates output errors to occur in bursts with durations typically of the same order as the constraint length K of the code.

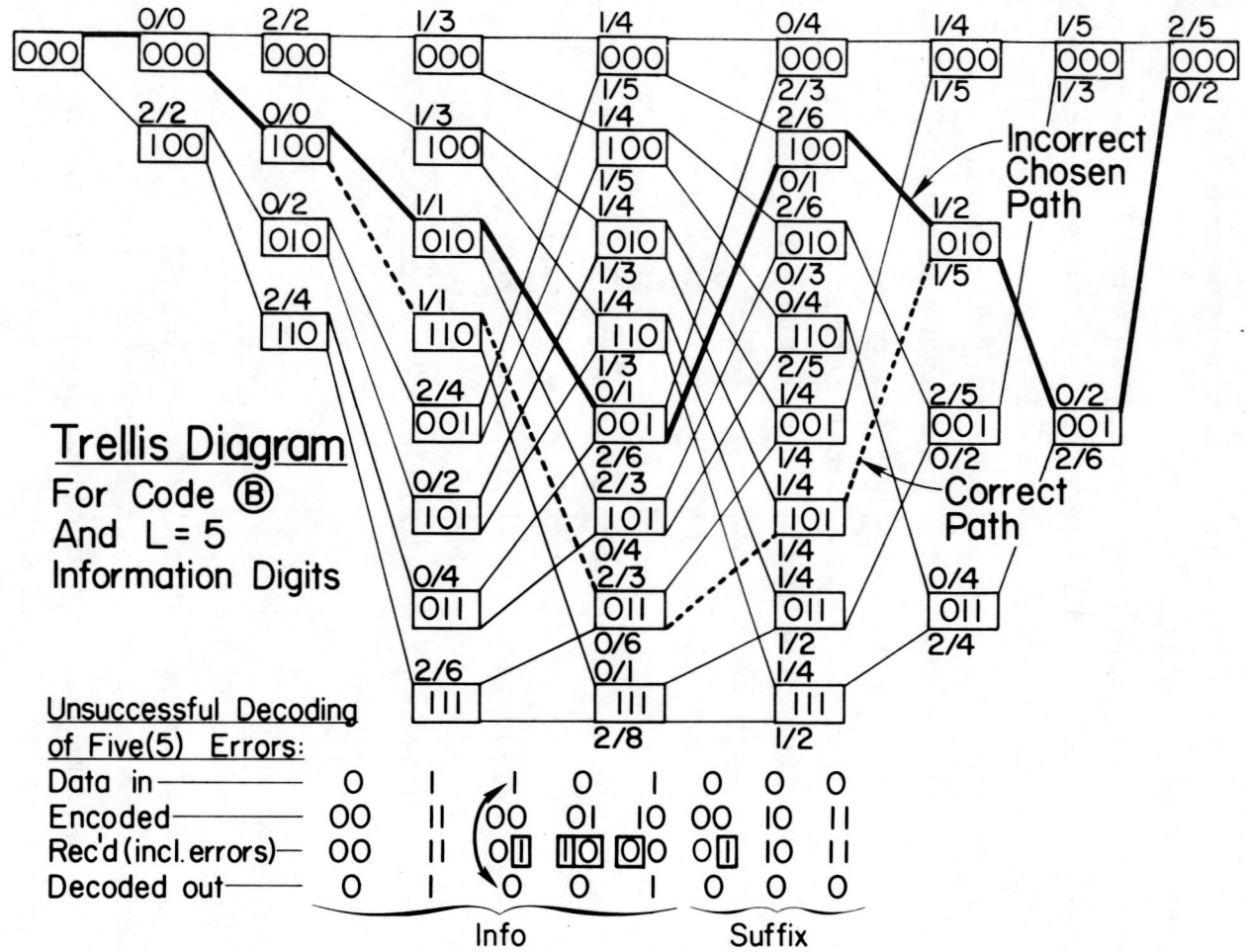


Figure 10. Unsuccessful decoding of five errors.



### 3.5. Performance on the Ideal Channel

Simulated ideal channel performance of convolutional FEC codes is summarized in figure 11. The ordinate is the output bit-error probability,  $p_I$ , and the abscissa is the normalized signal-to-noise ratio,  $E_b/N_0$  in decibels. Quantity  $E_b$  is the signal energy per information bit, and  $N_0$  is the one-sided noise spectral density. The curves of figure 11 depict the binary (i.e.,  $M=2$ ) coherent PSK channel in the presence of white Gaussian noise. As discussed before (McManamon et al., 1974), the main features and results readily generalize to the other MPSK modems, as well as other channel models.

The four vertical lines in figure 11 are designated as channel capacity limits. For instance, with no restriction on signal shape or duration, Shannon's ultimate theorem for error-free reception

$$R_I = W \log_2 \left( 1 + \frac{S}{N} \right)$$

may be contemplated. If one divides by rate  $R_I$  and lets bandwidth  $W$  assume larger and larger values, then

$$\ln 2 = \ln \left[ \left( 1 + \frac{R_I E_b}{W N_0} \right)^{\frac{W}{R_I}} \right] \xrightarrow{W \rightarrow \infty} \frac{E_b}{N_0} .$$

If so, the ultimate zero-error operation cannot be achieved unless  $E_b/N_0$  is larger than  $\ln 2$ , or -1.6 dB. If one insists on hard decision operation, the capacity threshold is multiplied by a factor  $\pi/2$ , or 2.0 dB. To compensate,  $E_b/N_0$  must then be above 0.4 dB. Likewise, if one permits arbitrarily fine soft decisions to accompany optimal sequential decoding, then  $E_b/N_0=1.4$  dB is needed. And, if one restricts the bandwidth to  $R_I W=1/2$  and specifies



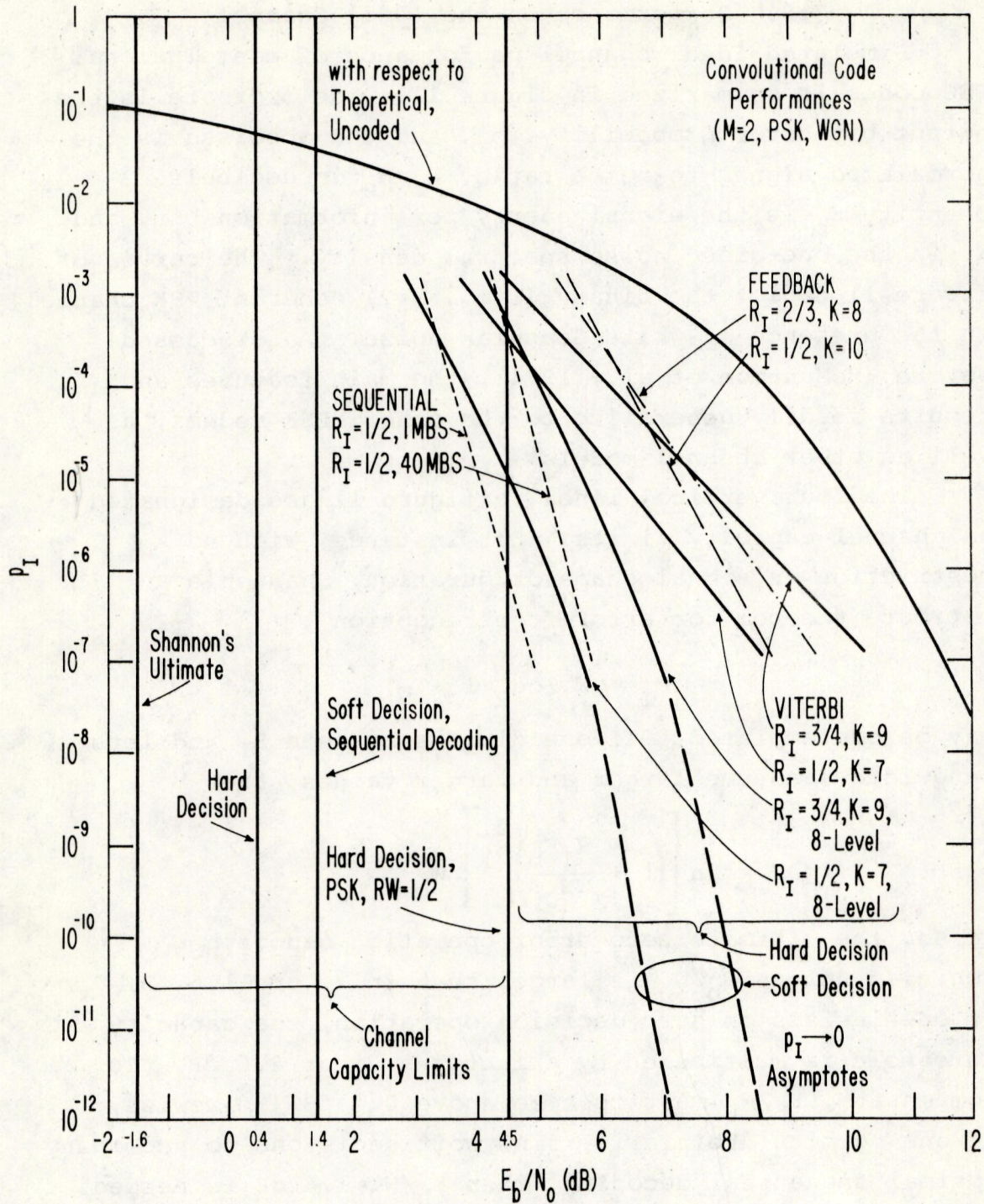


Figure 11. Various convolutional code performances on the ideal theoretical channel.



PSK hard decision operation, then more than 3 dB is sacrificed and the rightmost threshold  $E_b/N_0=4.5$  dB is required.

Without coding, the theoretical modem performance is a smooth function of  $E_b/N_0$ . The indicated curve crosses all the capacity limits and upper-bounds all the observed data in figure 11. The present observed coding data extends over the  $p_I$  range from  $10^{-3}$  to  $10^{-7}$  (Forney, 1970; Forney and Bower, 1971; Gilhousen et al., 1971; Batson et al., 1972; Odenwalder et al., 1972; Cahn et al., 1973; Jacobs, 1974). Three types of convolutional codecs are indicated. They are:

Sequential decoding, rate 1/2, and two different data throughput rates, 1 Mbps and 40 Mbps.

Hard decision.

Feedback decoding with rates 1/2 and 2/3, and constraint lengths 10 and 8, respectively.

Hard decision.

Viterbi decoding with rates 1/2 and 3/4, and constraint lengths 7 and 9, respectively.

Hard and soft decision.

In support of earlier table 2, figure 11 shows the mentioned observed performances. Error correction per se improves with lower coding rate  $R_I$  and higher constraint length  $K$ . At  $p_I=10^{-7}$ , for example, the rate  $R_I=1/2$  feedback decoding scheme requires roughly  $E_b/N_0=9$  dB. Viterbi decoding without quantization appears only slightly better, while hard decision sequential decoding offers a 3 dB improvement at the  $p_I=10^{-7}$  level. There is some uncertainty about the actual gain, as the complexity and speed of logic affects implementation and performance for sequential machines. Such is the nature of the 1 dB



advantage seen at the 1 Mbps data rate tests over the 40 Mbps tests.

Fortunately, the performance of Viterbi decoding can be substantially improved by the previously cited soft decision operation. In figure 11 two such 8-level quantization curves are shown. They depict code rates  $R_I=1/2$  and  $R_I=3/4$ . At  $p_I=10^{-7}$ , the rate 1/2 code is comparable to the sequential hard decision case, and requires a mere  $E_b/N_0 \geq 6$  dB to perform that well. While all this seems remarkably good, one is reminded of the interest in the objective of an overall  $10^{-12}$  error probability. The performance extrapolation beyond observed data, toward lower  $p_I$  values, is indicated in figure 11. The asymptotic  $p_I \rightarrow 0$  behavior is derived from the code distance structure, more specifically from the free distance  $d_f$  introduced above (Jacobs, 1974). Note, that  $E_b/N_0=7.3$  dB seems to provide  $p_I=10^{-12}$  for soft decision, rate  $R_I=1/2$ , Viterbi decoder. This signal-to-noise ratio should be contrasted with  $E_b/N_0=14.4$  that is required by uncoded ideal theoretical operation at the same  $p_I$  level, and the considerably larger extrapolated-measured requirement of 20.5 dB, or even worse (McManamon et al., 1974).

### 3.6. Performance on Realistic Channels

Actual channels, even on satellite links, are known to depart from the ideal white Gaussian noise model. There are many reasons why modems fall short of optimal performance and suffer a variety of operational losses. As a consequence, the previous ideal performance estimates must be modified. We will do so next, with particular attention to the two  $R_I=1/2$  and  $R_I=3/4$  Viterbi soft decision decoders described earlier and highlighted in figure 11. The

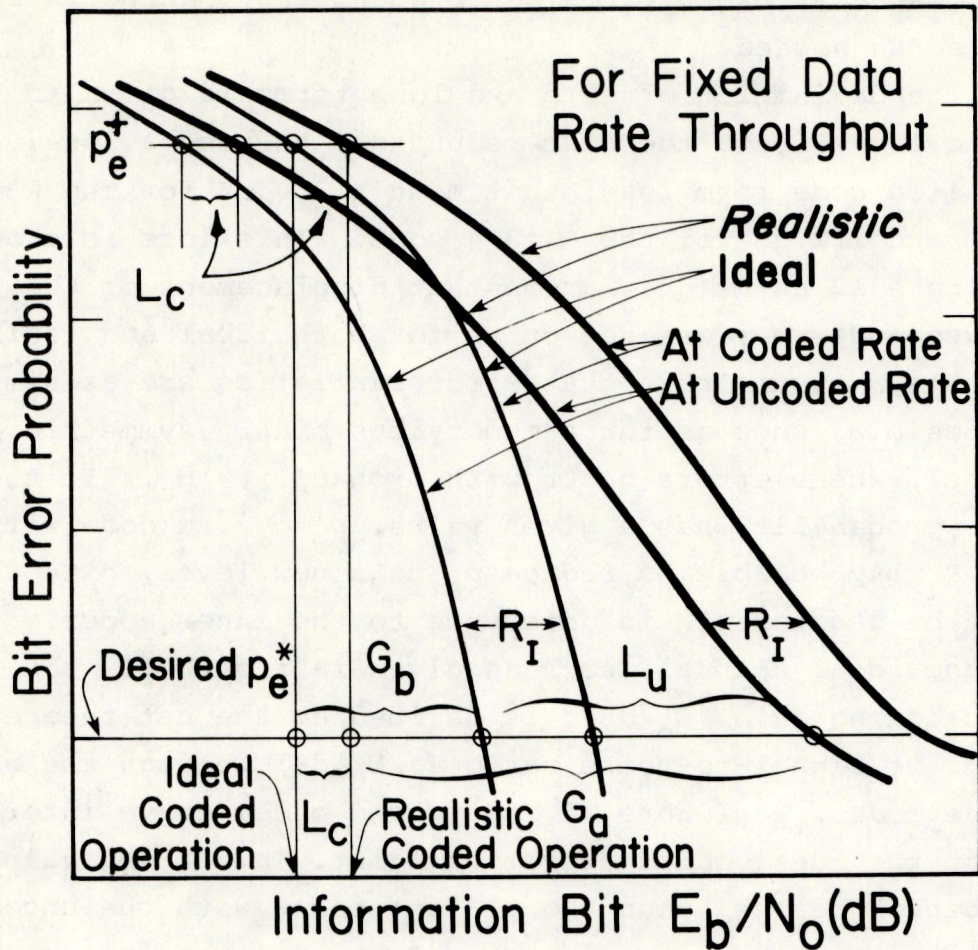


performance at other rates, such as  $R_I=2/3$ , can be interpolated as needed.

The definition of gain and loss terms is given in figure 12. First, the code redundancy carries a loss, the so called code rate loss, that equals 3.0 dB for the  $R_I=1/2$  code, and 1.2 dB for the  $R_I=3/4$  code. This loss is shown in figure 12 as  $R_I$  (dB), a constant displacement of the uncoded modem performance curve for both ideal and realistic channels. Assume next that error statistics are as simple as possible, such as for a memoryless binary symmetric channel, where errors occur with probability  $p_e$ . If the error probability has a given value,  $p_e=p_e^+$ , a code with rate  $R_I$  may be able to reduce  $p_e$  to a new level,  $p_e^*$ , which could be denoted  $p_I^*$  in deference to the inner coder. For the ideal channel, such as given in figure 11, the basic coding gain,  $G_b$  (dB), is defined as the net theoretical saving of signal-to-noise ratio  $E_b/N_0$  (dB) through the use of the code. In figure 12, the coded modem curve intersects  $p_e^+$  at the ideal coded operation point. This  $E_b/N_0$  value had better be less than the  $p_e^*$  intersect with the uncoded curve, for the basic coding gain  $G_b$  to be positive.

The actual coding gain,  $G_a$  (dB), is defined in a similar way. Note that the actual modem-channel pair must expend more signal energy to achieve the same performance,  $p_e^+$  or  $p_e^*$ , as the ideal. Furthermore, at specified  $p_e$  levels these degradations can be different. One has different modem losses at different performance levels. Thus in figure 12, the modem loss at  $p_e^+$  is shown to be  $L_c$  (dB), and at  $p_e^*$  it is  $L_u$  (dB). Typically,  $L_u > L_c$  implies that realistic coded system requires more  $E_b/N_0$  than does the corresponding ideal coded system, but not much more. One defines, therefore, the actual coding gain as the net





Terms:

$G_a$  (dB) = Actual Coding Gain

$G_b$  (dB) = Basic Coding Gain

$R_I$  (dB) =  $-10 \log_{10} R_I$  (Code Rate Loss)

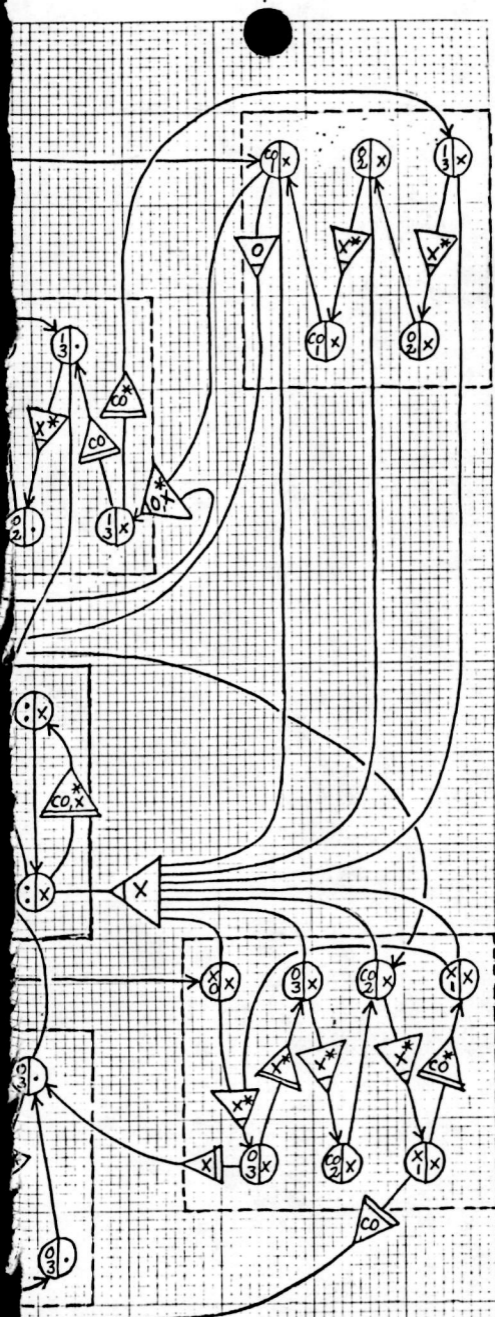
$L_u$  (dB) = Modem Loss, at  $p_e^*$  without Coding

$L_c$  (dB) = Modem Loss, at  $p_e^*$  with Coding

Result:  $G_a = G_b + L_u - L_c$

Figure 12. Definition of coding gains and modem losses.





END DIAGRAM

$E_b/N_0$  (dB) saving under the actual channel conditions. One sees from figure 12 that

$$G_a = G_b + L_u - L_c \quad (\text{dB})$$

must hold, and that  $G_a > G_b$  is apt to be true. The last assertion has validity for those real life  $p_e$  vs.  $E_b/N_0$  plots that flatten out and display an irreducible error rate. Then there may exist  $p_e^*$  levels at which the actual coding gain is infinite.

Recent data on applicable coding gains is summarized by Jacobs (1974), and the references listed. The gains can be extended to our ideal and extrapolated-measured curves, under all the specified conditions. For both  $M=2$  and  $M=4$  coherent PSK, rate  $R=1/2$ , constraint length  $K=7$ , eight level soft decision, a Viterbi decoder should perform as indicated in figure 13. In the center of the figure, a desired  $p_e^*=10^{-7}$  level is identified. At that level,

$$L_u \cong 3.5 \text{ dB},$$

$$L_c \cong 1.8 \text{ dB},$$

$$G_a \cong 7.5 \text{ dB},$$

$$G_b \cong 5.8 \text{ dB}.$$

The actual coding gains have been computed for various desired  $p_e^*$  levels. The results are given in figures 14 and 15 for the same two Viterbi arrangements cited earlier. Note the more or less invariant 1 dB  $E_b/N_0$  difference between the rate 1/2 and rate 3/4 codes. If the uncoded extrapolated-measured data calls for 20.5 dB signal-to-noise ratio at the  $10^{-12}$  level, then the present FEC scheme seems capable of the same reliability at 10.3 dB and 11.3 dB, for the two rates considered. This actual coding gain of 9 to 10 dB appears considerable, yet it is not the end of the story. More decibels can be conserved by the outer code, which is the topic of the next section.



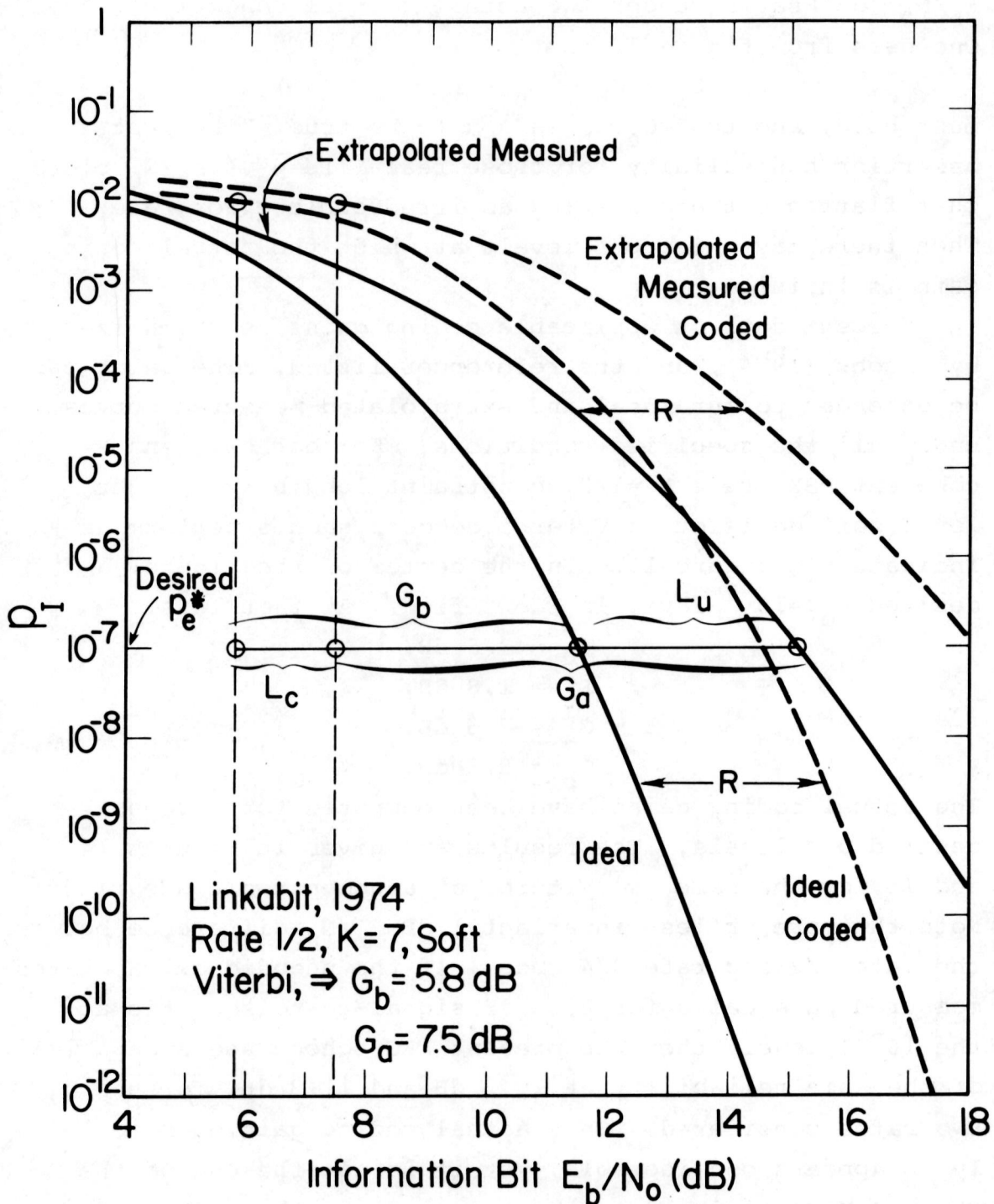


Figure 13. Viterbi decoding gains on the ideal and extrapolated-measured channels.

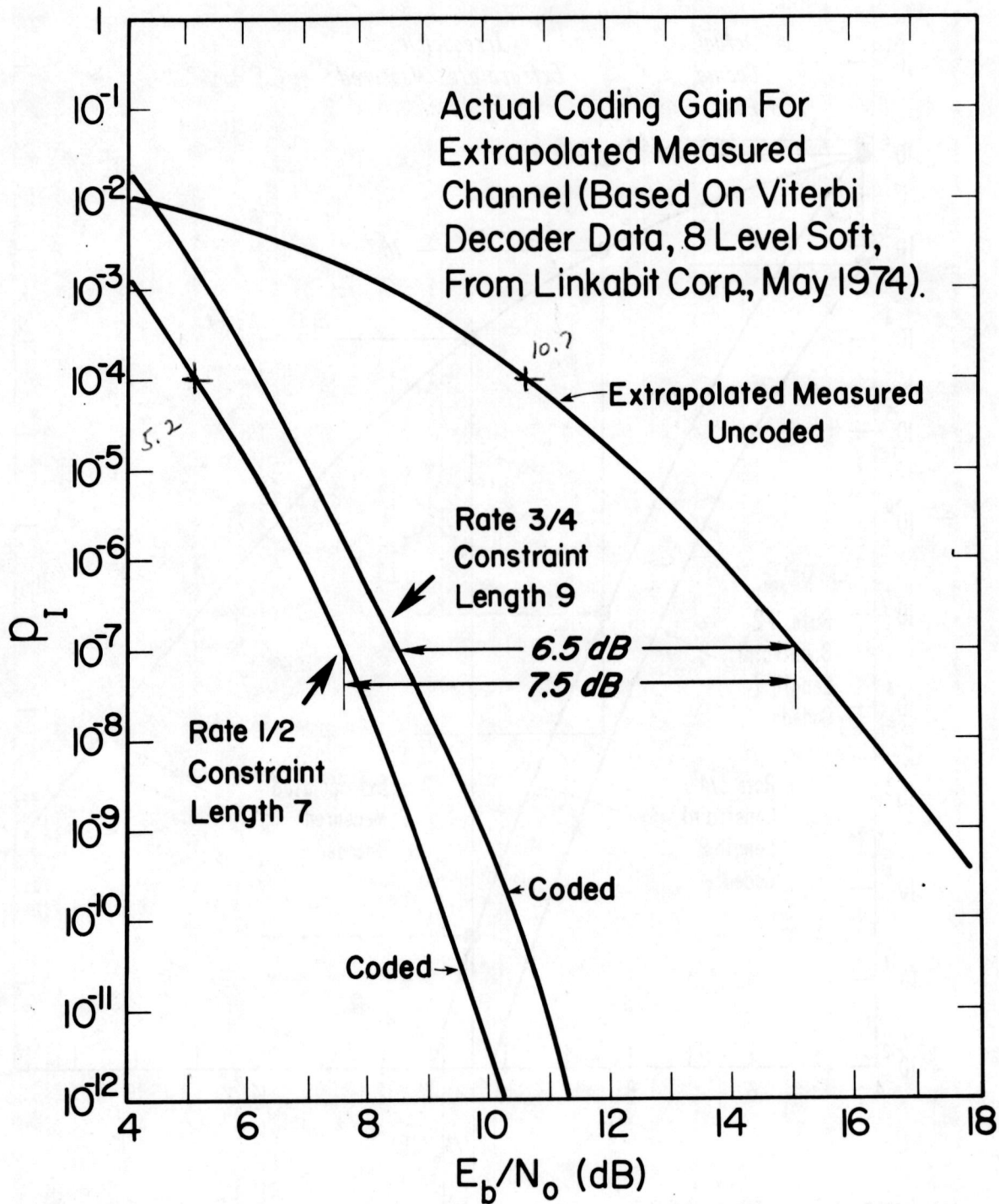


Figure 14. Actual coding gains for rate 1/2 and 3/4, soft decision, Viterbi decoders.



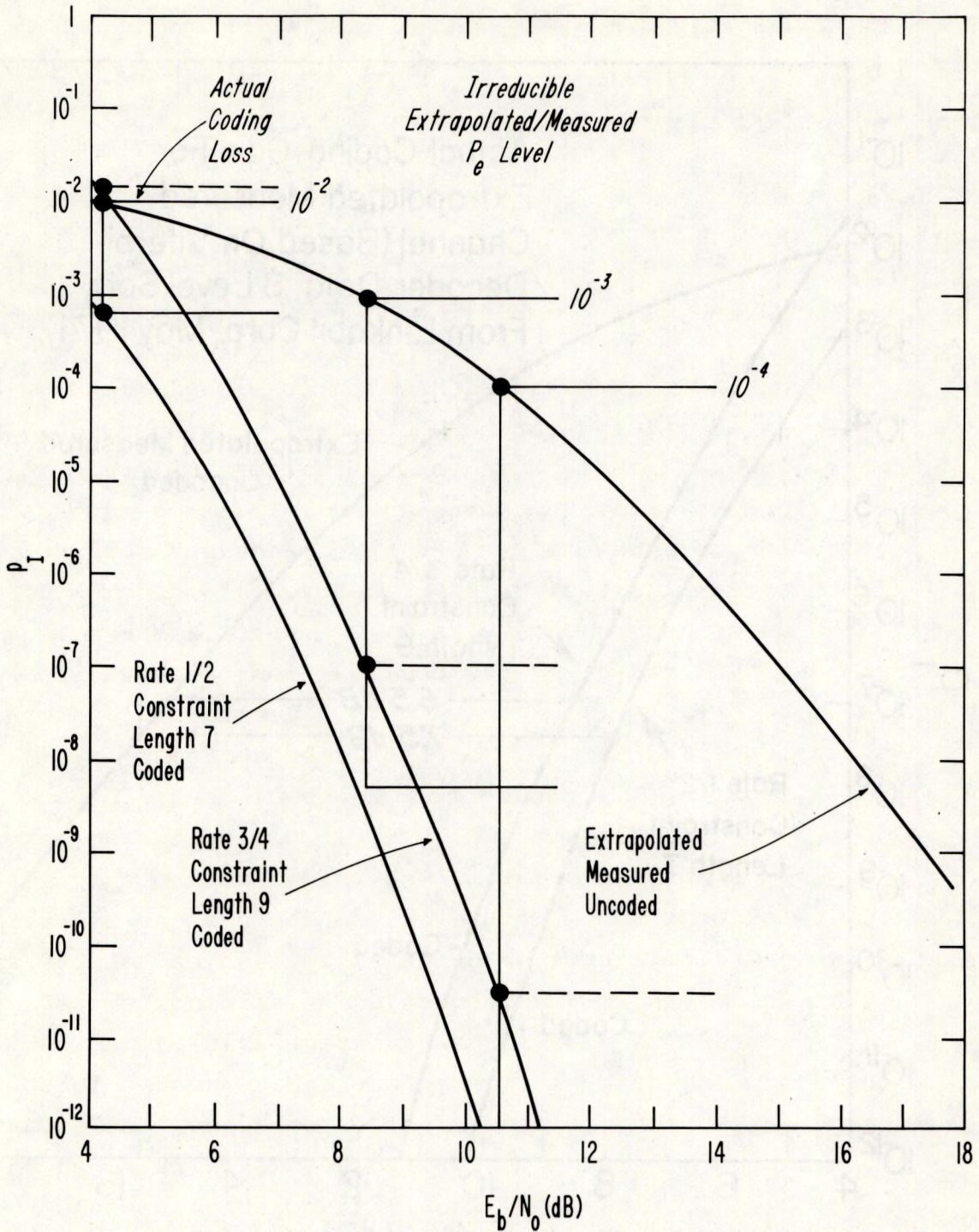


Figure 15. The effects of irreducible error probability.

Figure 15 also emphasizes that coding can be effective to reduce so called irreducible error rate levels. The effect is demonstrated by the horizontal lines to the right in figure 15. If, for example, the extrapolated-measured curve bottoms out at  $p_I=10^{-3}$  due to whatever malfunction, the output error probability after decoding will be roughly  $10^{-8}$  and  $10^{-7}$  for the codes with rates 1/2 and 3/4, respectively. An irreducible level at  $10^{-4}$  can be brought down to  $10^{-13}$  and  $10^{-11}$ . All irreducible levels lower than  $5(10^{-5})$  can satisfy the  $10^{-12}$  objective by the FEC action alone. The fact that the outer ARQ coding can help even more will be described next.

#### 4. OUTER CODE FOR THE CONCATENATED HYBRID

##### 4.1. Codec

The selection of a hybrid FEC and ARQ scheme was discussed in section 2. Moreover, table 1 and pertaining arguments led to the announcement that a concatenation of an inner convolutional code with an outer block code has the most promise of all the alternatives considered. The overall system was briefly sketched in figure 1, identifying the outer code with the ARQ function. That outer code will be the topic of this section. Matters such as the encoder, the decoder, and the role and the performance of the error detecting outer code will be elaborated.

To begin, the outer code selection is relatively straight forward. Whatever the length,  $n$ , of the block code, one uses an almost negligible number of parity check bits for error detection. The abbreviation  $(n,k)$  is used for such a block code, where  $k$  is the number of information bits and  $n-k$  the

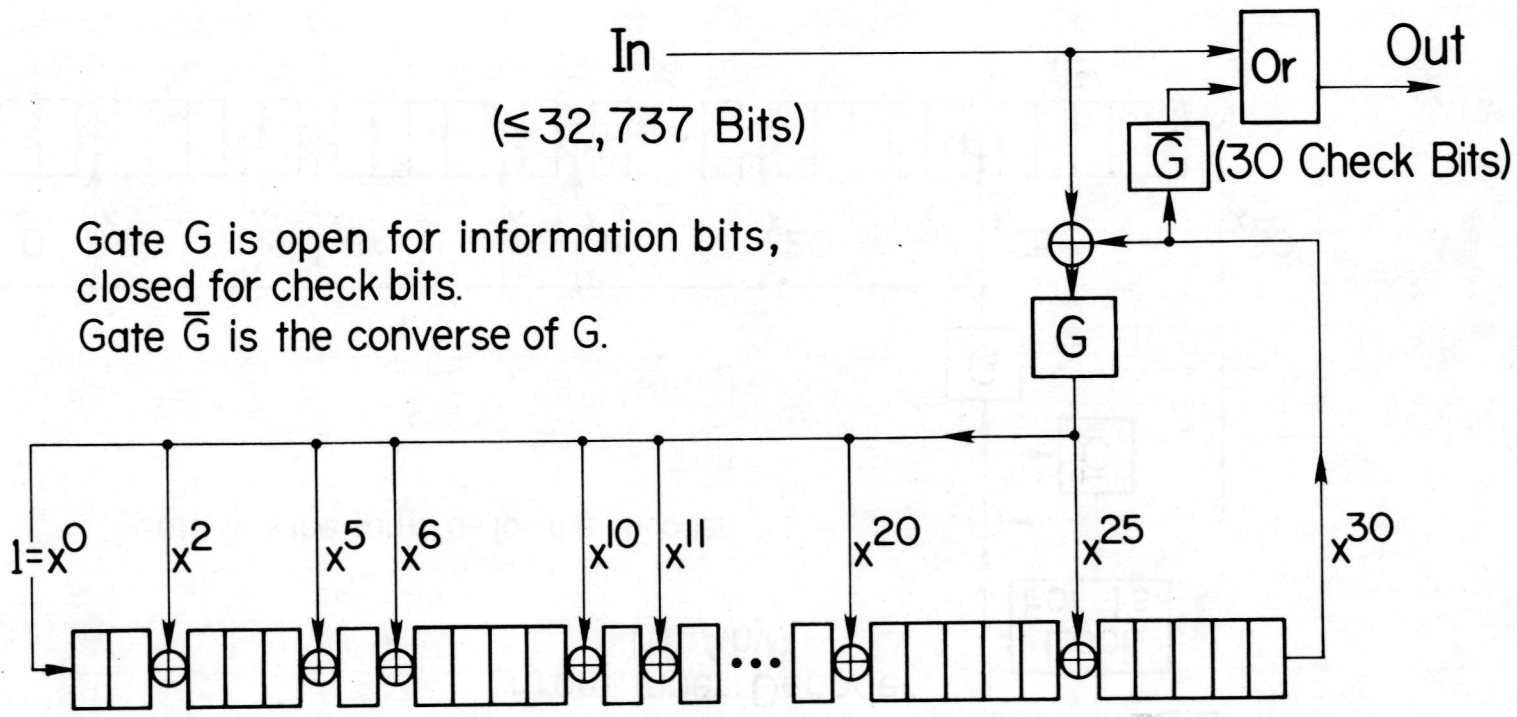


number of check bits. It seems reasonable to start with block length  $n$  somewhere between 10,000 and 1,000,000. Redundancy of 1 percent [i.e., the ratio  $(n-k)/n$  expressed in percent] then provides 100 to 10,000 check bits. As will be shown later, this number of parity checks may be superfluous and a much smaller percent redundancy may suffice. Conclusion: the redundancy loss for the outer code of rate  $R_0$  is negligible,  $R_0 \approx 1$ .

Past experience with ARQ systems (van Duuren, 1961; Nesenbergs, 1963; Townsend and Watts, 1964; Bernice and Frey, 1964; Burton, 1970) has shown that the block length  $n$  affects primarily the throughput rate  $R$ . Roughly, throughput vanishes in an ARQ system whenever the input error probability exceeds  $1/n$ . Hence,  $n$  larger than  $10^6$  appears undesirable. Similarly, the number of check bits  $n-k$  determines the likelihood of the  $n$ -bit word containing undetected errors.

The outer encoder need not be more involved than a standard linear feedback shift-register (Gallager, 1968; Peterson and Weldon, 1972). The specific array of feedback taps can be chosen to guarantee as high a minimum distance as is possible for BCH or other known block codes. An encoder so derived is shown in figure 16. Used with  $k$  information bits,  $k$  not exceeding 32,737, the  $n-k=30$  added parity bits ensure that all four or less inner decoder output errors are detected. When the number of errors is five or more, the probability of detecting the errors is still exceedingly high.

The outer decoder, figure 17, is almost a duplicate of the outer encoder. The device simply verifies whether all  $n-k$  parity checks are satisfied by the received code word. If not, the repeat of the word must be requested from the



Gate G is open for information bits,  
closed for check bits.  
Gate  $\bar{G}$  is the converse of G.

41

Figure 16. Outer encoder.



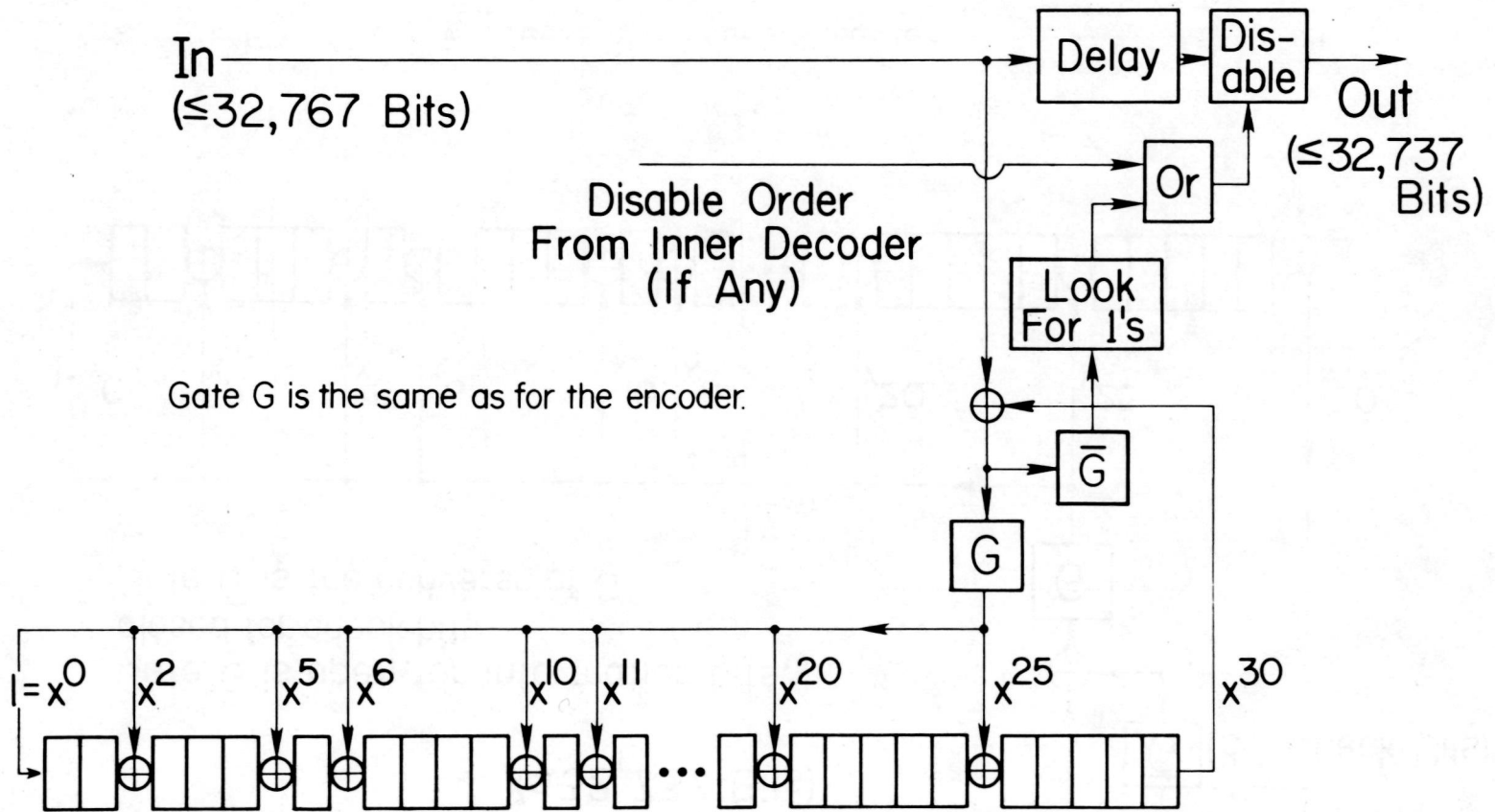


Figure 17. Outer decoder.

transmitter, and the present word output must be disabled. This process is known as the ARQ action. If all parity checks are satisfied, there is no need for ARQ action. Figure 17 also permits a secondary disable order to be used by the receiver terminal for unspecified reasons.

To select the block  $(n,k)$  code and the appropriate codecs, a convenient shorthand for the feedback taps should be used. Thus, the nine taps in figures 16 and 17 can be represented in binary as

101 001 100 011 000 000 001 000 010 000 1,  
or in octal as 5,1,4,... or by a conventional numerical listing of the locations of the taps:

0,2,5,6,10,11,20,25,30.

Whereas the coding literature seems to prefer the binary and octal representations, we shall employ the numerical location list. As seen from figures 16 and 17, such a location list consists of all registers with exterior connections. It is also evident that the first (entry 0) and the last (entry  $n-k$ ) taps must be always present. Hence, one needs only the list of the interior taps, that is, those taps that involve exclusive -OR gates. In figures 16 and 17 there are seven such interior taps, namely

2,5,6,10,11,20, and 25.

There are still nearly  $2^{n-k-1}$  different codecs possible via specifications of the shift-register taps. In table 3 we present a short selection of codes that happen to belong to the BCH category (Berlekamp, 1968; Peterson and Weldon, 1972). Note that the block length  $n$  can be increased relatively rapidly, compared to the modest number of registers,  $n-k$ . The number of inner taps seems to grow more or less proportionally to the length of the shift-register.



Table 3. Possible Error Detecting Codes for ARQ

Upper bound on block length n	Number of registers, same as n-k	Number of inner taps to exclusive -OR gates	Numerical listing of all the inner taps, same as ex- clusive -OR gates		
4 095	12	3	1,4,6		
4 095	24	13	2,3,4,7	8,9,10,11	12,14,15,16,22
8 191	13	3	1,3,4		
8 191	26	11	1,3,6,8	10,12,16,18	20,22,23
16 383	14	3	1,6,10		
16 383	28	11	3,5,6,9	10,14,15,18	19,22,24
*32 767	30	7	2,5,6,10	11,20,25	
32 767	60	33	4,6,7,8	12,13,15,16	17,18,19,21
			22,24,25,28	29,31,35,36	37,39,40,42
			43,45,46,47	50,54,56,57	58
65 535	32	15	1,2,3,6	7,8,10,13	16,17,18,21
			23,25,27		
65 535	64	28	2,10,15,18	19,20,23,26	28,31,33,35
			36,37,40,41	42,44,45,46	47,49,52,53
			55,59,60,62		
131 071	51	21	1,2,3,4	5,7,8,9	12,13,14,17
			22,26,27,34	35,36,37,38	42

\*The encoder of this code is shown in figure 16, the decoder in figure 17.

## 4.2. ARQ Performance

The performance of the outer ARQ codec involves at least three criteria:

- (i) the throughput rate  $R$ , a quantity that is related but not identical to the code rate  $R_0$ ;
- (ii) the output word error probability  $P_0$ ;
- (iii) the statistics of delays and random queues caused by the ARQ action, and the associated storage overflow implications.

In this section we will concentrate on the first two entities,  $R$  and  $P_0$ , for a complete ARQ duplex arrangement, as outlined in figure 18.

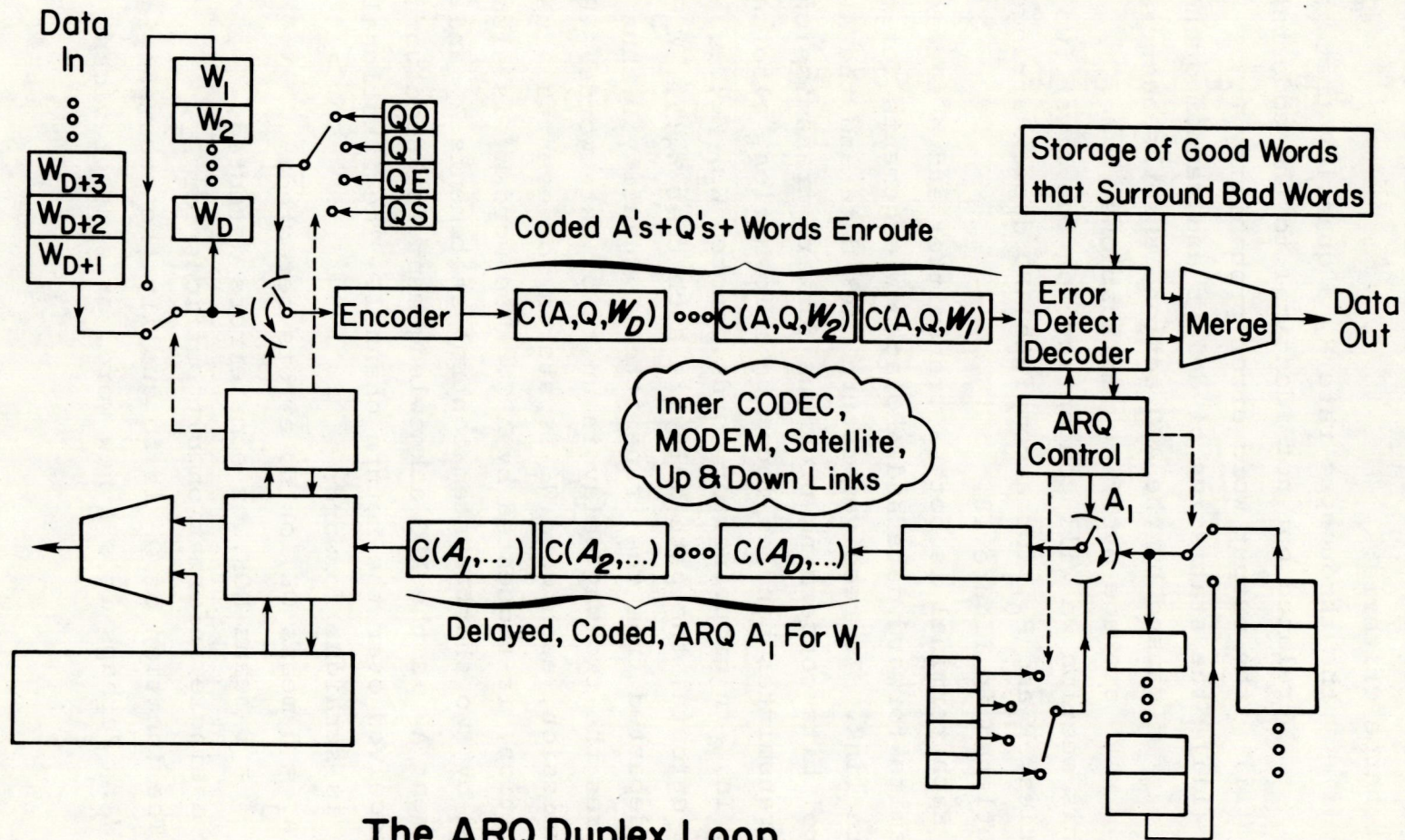
Each terminal is both a transmitter and a receiver. It transmits forward data and ARQ acknowledgements for the return link. It receives return link data and the ARQ control bits concerning previous forward transmissions. The transmitter handles incoming information, also called "Data In," in groupings or words denoted  $W_1, W_2, \dots$ . While  $W_1$  is enroute, the path delay is such that  $W_2, \dots, W_D$  have also departed from the transmitter. Quantity  $D$  thus measures the two-way delay in units of data words. Before transmission, each word  $W_i$  is stored for eventual future repetition, if requested by ARQ. The word  $W_i$  is also joined by two short system control statements  $A_j$  and  $Q_k$ . Statement  $A_j$  is the ARQ acknowledgement for some word,  $W_j$ , received over the return channel. The functional role of  $A_j$  is straight forward:

- $A_j = 0$  means OK, or no errors detected;
- $A_j = 1$  means NOK, or some errors detected.

Thus, a single information bit suffices for  $A_j$ .

The function of  $Q_k$  is to qualify the accompanying word  $W_k$ . Perhaps it's a new word, or some previous





### The ARQ Duplex Loop

Figure 18. The ARQ duplex loop.

word  $W_i$  that is being repeated. The nature of the repeat, such as due to  $A_k$  being NOK, or  $A_k$  being erased (lost), or some system outage, may be important. Loosely speaking,  $Q_k$  is an indicator for the ARQ loop's present state. The following short table outlines a possible four state scheme for the ARQ duplex of figure 18:

A = Q0 (OK) → Clear  $W_1$ , Store  $W_{D+1}$ , Send  $C(A, Q0, W_{D+1})$   
 = Q1 (NOK) → Keep  $W_1$ , Send  $C(A, Q1, W_1)$   
 = QE (Erasure) → Keep  $W_1$ , Send  $C(A, QE, W_1)$   
 = QS (System) → Keep  $W_1$ , Send  $C(A, QS, W_1)$

Here, two bits suffice to characterize Q. After assembling, the three parts,  $A_i$ ,  $Q_j$ , and  $W_k$ , are encoded together and become  $C(A_i, Q_j, W_k)$ , or  $C(A, Q, W)$  for short. Because their bit count is so small compared to  $W$ , the segments A and Q can be safely ignored in data rate calculation.

The key to performance (i) and (ii) calculation is given by the rules of the ARQ. A synoptic start is offered in figure 19. There are three diagnoses possible on both the forward and feedback paths. A word can emerge from the inner decoder either correct, with errors that are detected, or with undetected errors. The same things happen to the ARQ acknowledgement, except that an incorrect A is said to be transposed, and a doubtful A is said to be erased.

The ARQ control logic contained in the duplex loop (fig. 18) determines the consequences for all nine alternatives shown in figure 19. Without going into too much detail, such a diagram can still be helpful. It assists to compute, to upper bound, or to lower bound performance indicators, such as R and  $P_0$ . In what follows, we shall



		Forward Path		
		Correct	Detected Error	Undetected Error
Feedback Path	$W_1$ is $A_1$ is			
	Correct		<i>Sure Repeat</i>	<i>Sure Error</i>
	Erased	<i>Possible Repeats</i>		
Transposed		<i>Error Possible</i>		

## Synopsis of ARQ

Figure 19. Synopsis of ARQ.

be content to estimate some very primitive bounds on the two quantities.

Let  $p_n(j)$  be the probability of  $j$  errors in the  $n$  bit word, where  $n$  is a large integer and  $0 \leq j \leq n$ . This probability is observed in the data stream, where it emerges from the inner decoder and enters the outer decoder. We assume that the same  $p_n(j)$  holds for the forward and feedback paths, but that the two links are otherwise statistically independent. It follows that the throughput rate  $R$  must be bounded by

$$\frac{k}{n} p_n^2(0) \leq \frac{R}{R_I} \leq \frac{k}{n} p_n(0).$$

This allegation is verified by turning to figure 20 and considering the least possible repeats, as well as the most possible, in the ARQ synopsis (fig. 19).

For example, the left side of this inequality follows from

$R \geq R_I R_0 \Pr\{W_1 \text{ accepted in the most repeat case}\}$ ,  
where  $R_0 = k/n$  and

$$\begin{aligned} \Pr\{\dots\} &= \Pr(A_1 \text{ correct}) \Pr(W_1 \text{ correct}) \\ &\quad + \Pr(A_1 \text{ correct}) \Pr(W_1 \text{ undet. error}). \end{aligned}$$

When acknowledgement  $A_1$  is shorter than word  $W_1$ ,

$$\Pr(A_1 \text{ correct}) \geq \Pr(W_1 \text{ correct}) = p_n(0),$$

and

$$\frac{R}{R_I} \geq \frac{k}{n} p_n^2(0).$$

Likewise, the right side follows from

$R \leq R_I R_0 \Pr\{W_1 \text{ accepted in the least repeat case}\}$ ,  
where now

$$\Pr\{\dots\} = \Pr(W_1 \text{ correct}) + \text{negligible terms.}$$

The ARQ word-error probability behavior is somewhat different, and calls for a different characterization (see fig. 21):



		Forward Path		
		$W_1$ is $A_1$ is	Correct	Detected Error
Feedback Path	Correct		<i>Least Repeats</i>	
	Erased		<i>Most Repeats</i>	
	Transposed			

### Synopsis of ARQ

Figure 20. The least and most repeats in the ARQ loop.

		Forward Path		
		Correct	Detected Error	Undetected Error
Feedback Path	$W_1$ is $A_1$ is			
	Correct			<i>Least Errors</i>
	Erased			
Transposed			<i>Most Errors</i>	

## Synopsis of ARQ

Figure 21. The least and most errors in the ARQ loop.



$$P_0 \geq \Pr(\text{accepted } W_1 \text{ has some error in the least error case})$$

$$\geq \frac{\Pr(W_1 \text{ undet. error})}{\Pr(W_1 \text{ correct}) + \Pr(W_1 \text{ undet. error})}$$

$$\geq \Pr(W_1 \text{ undet. error})$$

$$P_0 \leq \Pr(\text{accepted } W_1 \text{ has some error in the most error case})$$

$$\leq \Pr(W_1 \text{ undet. error}) + \Pr(A_1 \text{ transposed}).$$

Since acknowledgement  $A_1$  is an extremely short segment (such as one or two bits) of an  $n$ -bit word, one expects

$$\Pr(A_1 \text{ transposed}) \ll \Pr(W_1 \text{ undet. error}).$$

One can show, for example, that

$$\frac{\Pr(A_1 \text{ transposed})}{\Pr(W_1 \text{ undet. error})} \cong (\text{constant}) \frac{d}{n}.$$

Here,  $d$  stands for the minimum distance of the code. A BCH code with  $n=2^m-1$  and  $n-k=mt$  has  $d \geq 2t+1$ , as is well known (Berlekamp, 1968; Gallager, 1968; Peterson and Weldon, 1972). Thus,  $d \ll n$  even for the best codes, and especially for  $n \gg 1$ . One concludes that the upper and lower bounds are indistinguishable (Nesenbergs, 1963; Lucky et al., 1968):

$$P_0 \cong \Pr(W_1 \text{ undet. error}) \\ \cong 2^{-(n-k)} \sum_{j=d}^n p_n(j).$$

The bounds on the throughput rate  $R$  and the outer coder word-error probability  $P_0$  have been computed for two channel models. The first model, denoted as MOD I, is the memoryless binary symmetric channel with probability  $p_I$ . For MOD I:

$$p_n(j) = \binom{n}{j} p_I^j (1 - p_I)^{n-j}, \quad 0 \leq j \leq n.$$

The computation of sums of such binomial terms is an old problem, known to be difficult for large  $n$  and small  $p_I$ , which is exactly our case. An approximation far better than the Chernoff bound or the direct Gaussian approximation is given by the asymptotic expression (Gallager, 1968),

$$\sum_{j=d}^n \binom{n}{j} p_I^j (1 - p_I)^{n-j} \cong \frac{\left(\frac{np_I}{d}\right)^d (1 - p_I)^{n-d+1}}{\sqrt{2\pi d} \left(1 - \frac{d}{n}\right)^{n-d+\frac{1}{2}} \left(1 - \frac{np_I}{d}\right)},$$

whenever  $np_I < d$ . If  $np_I \geq d$ , one can conveniently upper bound the sum by unity.

The second channel model, denoted as MOD II, postulates error bits to be effectively bunched in solid bursts of length  $\nu$ . It follows from a Poisson approximation that for MOD II:

$$p_n(0) \cong \exp\left(-\frac{np_I}{\nu}\right),$$

$$\sum_{j=d}^n p_n(j) \cong np_I / (np_I + \nu),$$

where again  $np_I < d$ . To carry out the computation, we have used three values  $\nu=5, 15, \text{ and } 45$ .

The computed results are given in a host of figures, starting with figure 22. The first three figures, 22 through 24, depict the ARQ throughput rate  $R/R_I$  as a function of the inner codec error probability  $p_I$ , all for block length



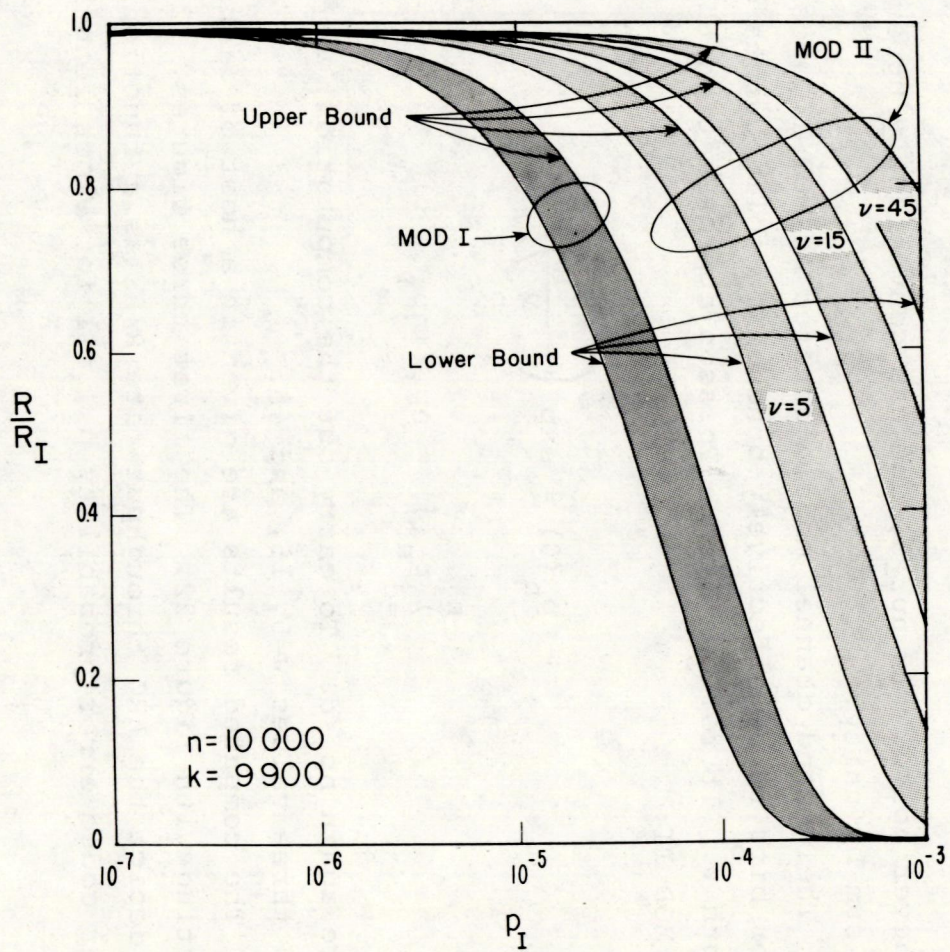


Figure 22. The bounds of throughput rate for  $n=10,000$  and  $k=9,900$ .



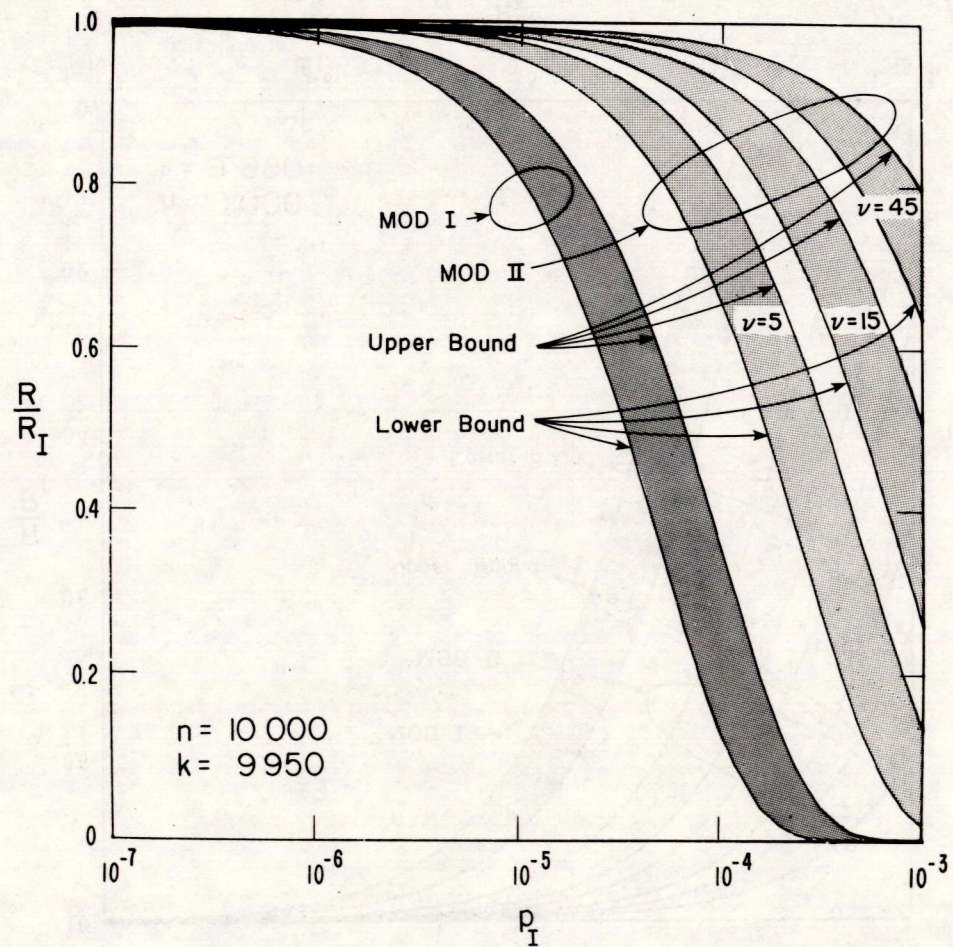


Figure 23. The bounds on throughput rate for  $n=10,000$  and  $k=9,950$ .



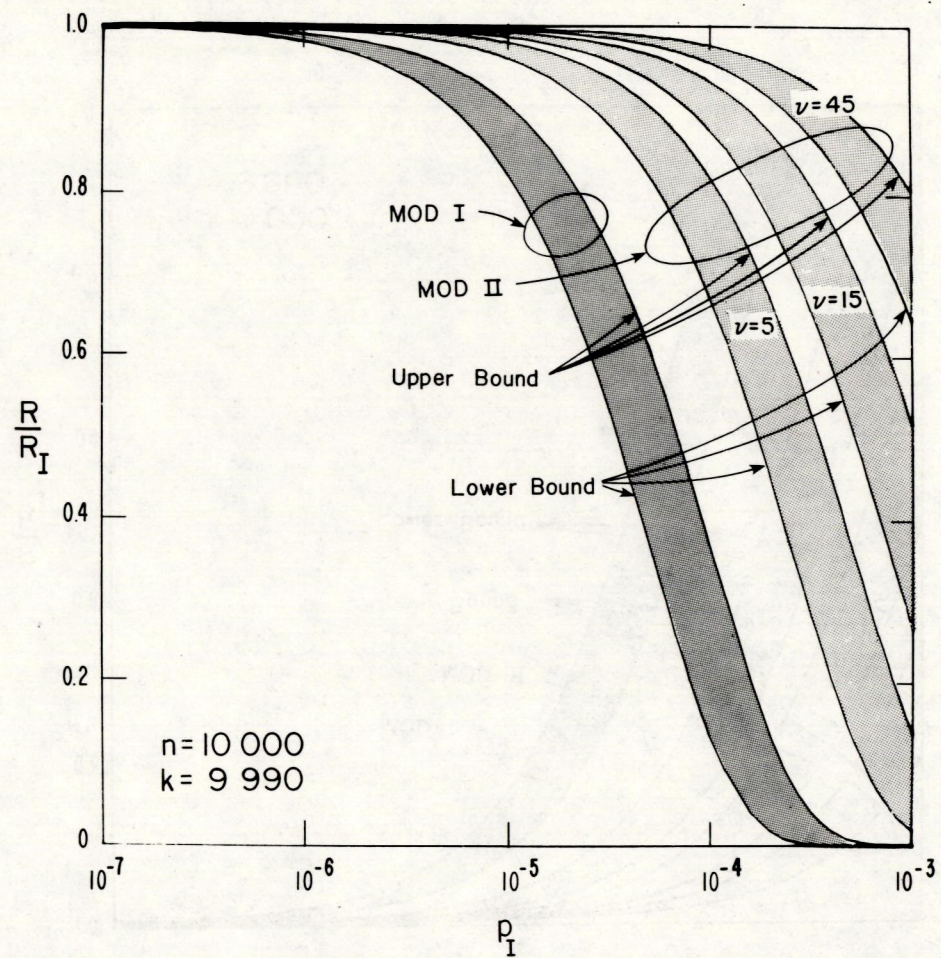


Figure 24. The bounds on throughput rate for  $n=10,000$  and  $k=9,990$ .

$n=10,000$ . The figures differ in  $k$  value: Figure 22 has  $k=9,900$ , figure 23 has  $k=9,950$ , and figure 24 has  $k=9,990$ . Each figure has eight curves. The leftmost two curves belong to MOD I, while the other six curves belong to MOD II. Furthermore, each adjacent pair of curves, as for MOD I, represent the lower and upper bounds on the outer code throughput rate  $R/R_I$ . The three lower and upper bound pairs for MOD II are descriptive of solid burst lengths  $v=5, 15, \text{ and } 45$ , respectively.

One notes the following features:

- (1) The effect of  $k$  is a very minor change in the vertical scale of the curves. Since this effect is entirely predictable, one can omit  $k$  dependence in other similar plots. Accordingly, figures 25 to 28 are all for  $k/n=0.99$ , but with different  $n$  values.
- (2) The upper and lower bounds for the same channel model are separated very nearly by a factor of 2 in the horizontal,  $p_I$ , dependence.
- (3) One can view MOD I as a special  $v=1$  case of MOD II. The effect of different  $v_1$  and  $v_2$  values then seems to correspond to a horizontal  $p_I$  shift by a factor  $v_1/v_2$ . Figure 25, for example, has a throughput upper bound  $R/R_I=0.5$  for MOD I at  $p_I \approx 2(10^{-5})$ , for MOD II,  $v=5$ , at  $p_I \approx 10^{-4}$ ,  $v=15$  at  $p_I \approx 3(10^{-4})$ , and so forth. An increase in error bursting,  $v$ , benefits the throughput rate. Unfortunately, realistic channels, including inner Viterbi decoders, are not sufficiently well understood to state what  $v$  value is applicable. One can speculate



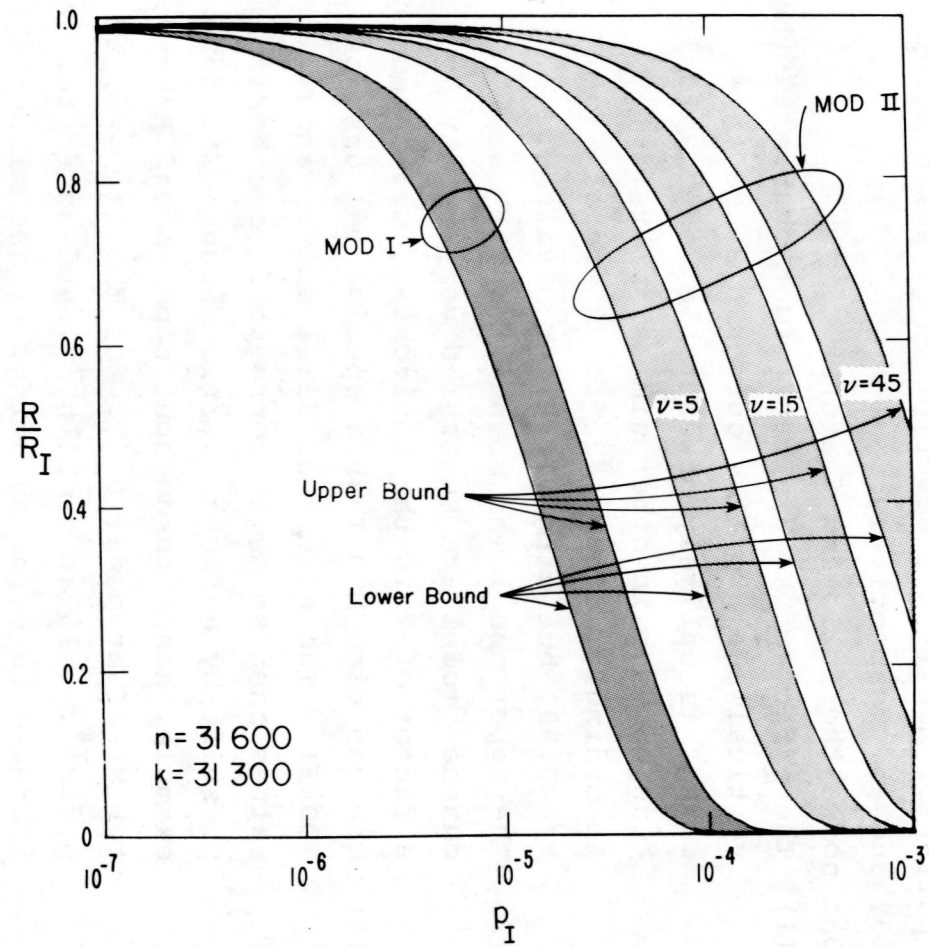


Figure 25. The bounds on throughput rate for  $n=31,600$  and  $k/n=0.99$ .

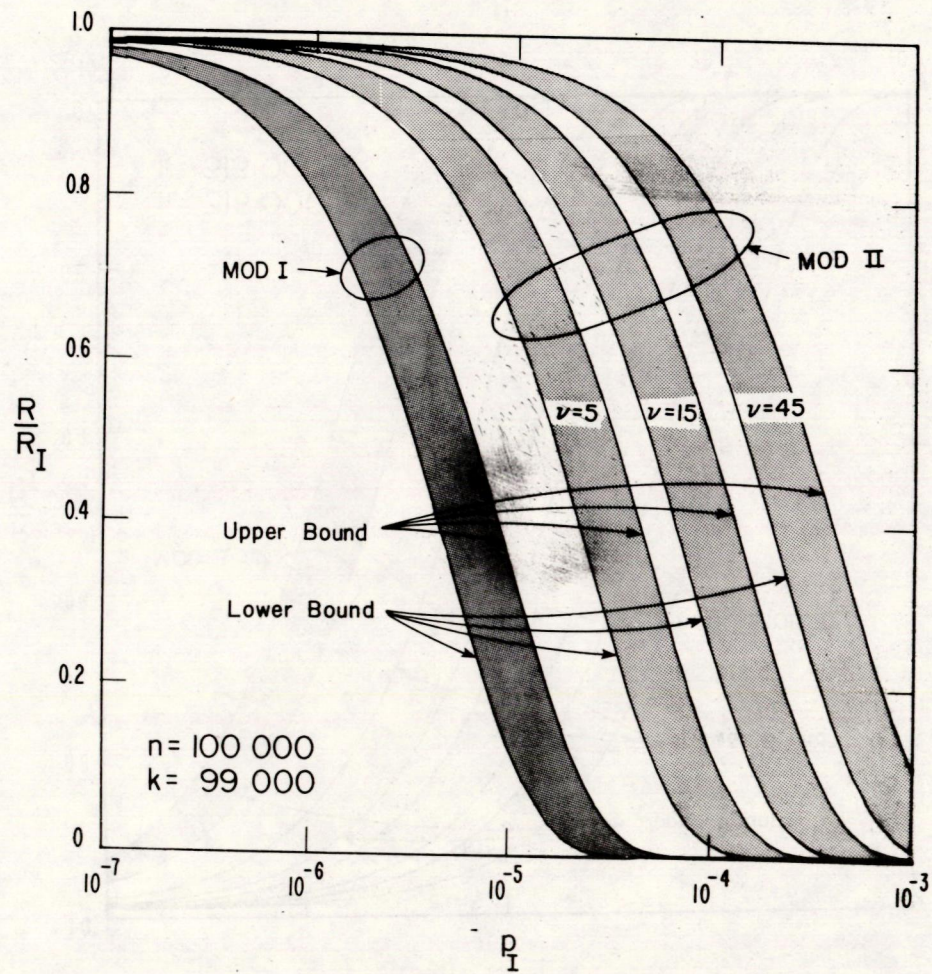


Figure 26. The bounds on throughput rate for  $n=100,000$  and  $k/n=0.99$ .



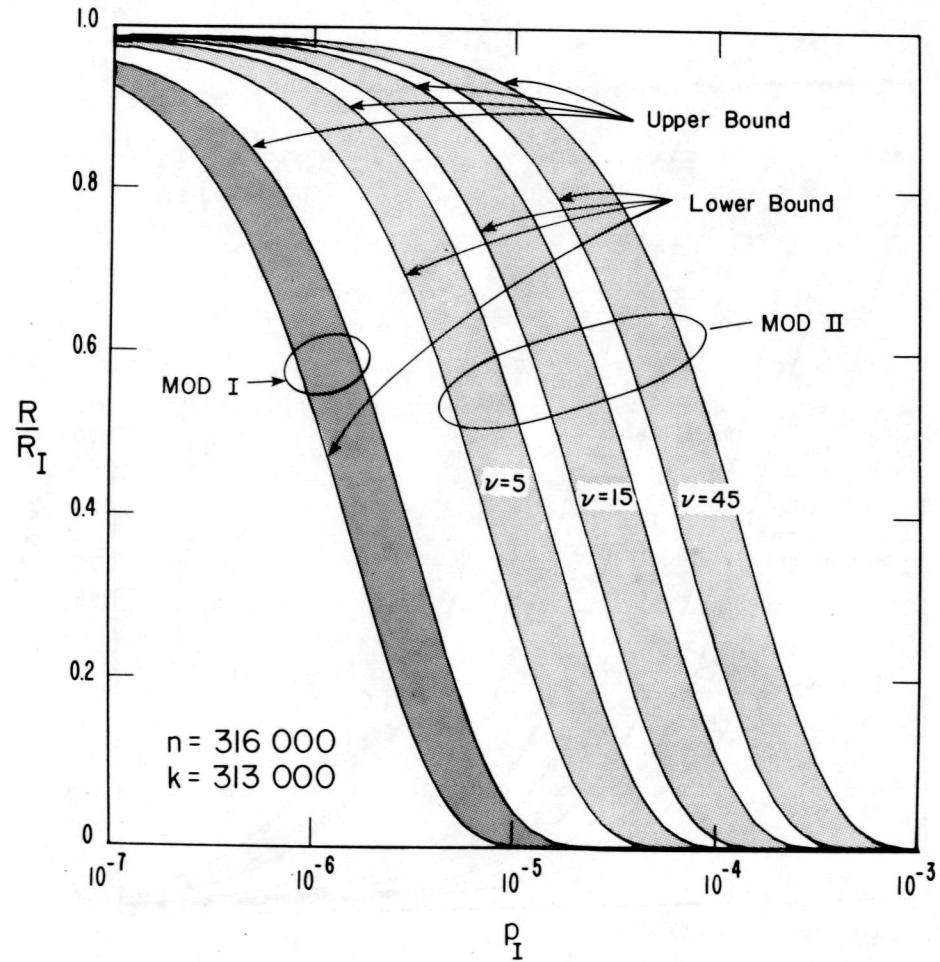


Figure 27. The bounds on throughput rate for  $n=316,000$  and  $k/n=0.99$ .

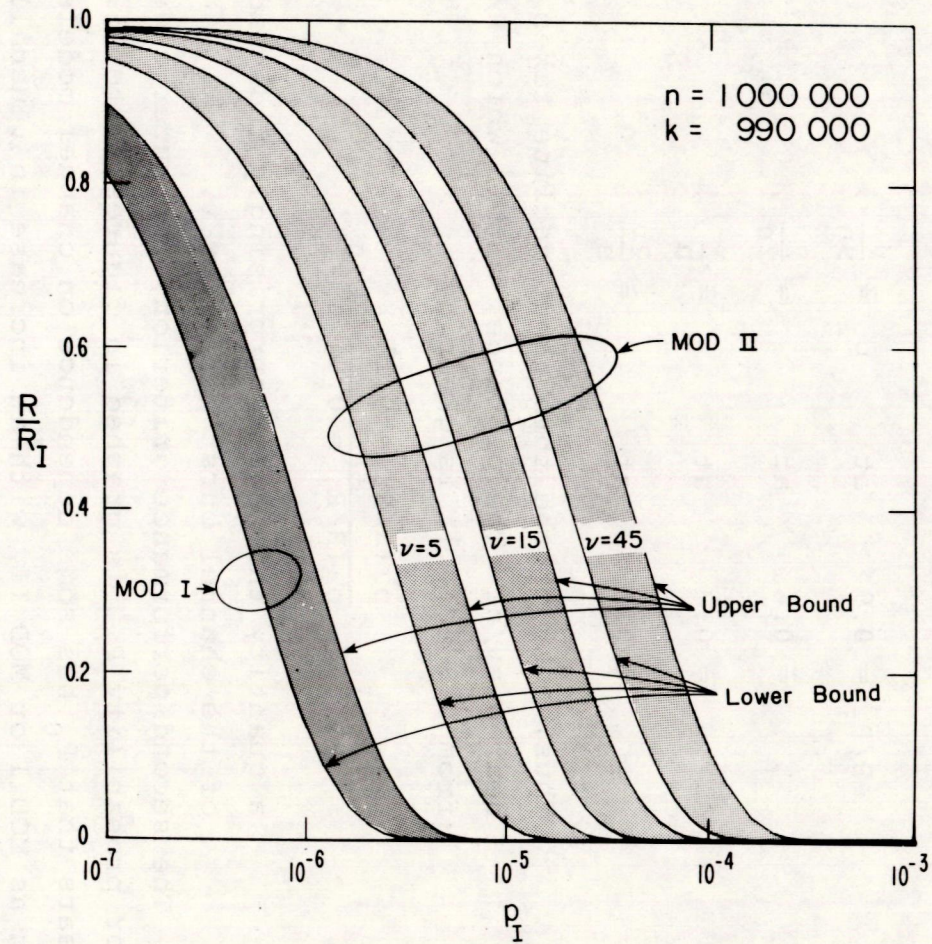


Figure 28. The bounds on throughput rate for  $n=1,000,000$  and  $k/n=0.99$ .



that for nonsystematic convolutional codes, with constraint length  $K=8$  and delay parameter  $L=40$ , a good guess could be  $v \approx 5$ . But, without further evidence the entire matter is too vague to tell.

(4) As another rough rule of thumb one can note:

$$\begin{array}{rcl}
 R/R_I \approx 0.9 & \text{at} & P_I \approx \frac{v}{9n} , \\
 \approx 0.6 & \text{at} & \approx \frac{v}{2n} , \\
 \approx 0.3 & \text{at} & \approx \frac{v}{n} , \\
 \approx 0.1 & \text{at} & \approx \frac{2v}{n} .
 \end{array}$$

Thus, most of the throughput decline takes place near  $p_I = v/n$ . The  $p_I$  range, within which  $R/R_I$  drops from 0.9 to 0.1, is given by

$$\frac{P_I(R/R_I = 0.1)}{P_I(R/R_I = 0.9)} \approx 20 ,$$

a quantity that seems surprisingly independent of the channel burst parameter  $v$ .

The second performance criterion, the outer code word error probability  $P_0$ , is graphed in figures 29 to 31. It appears that  $P_0$  has some dependence on channel models, such as MOD I or MOD II, with an increase in  $v$  usually implying a relatively small  $P_0$  improvement. Moreover, since the outer decoder monitors the concatenated coding output, the overall output word error probability is simply  $P=P_0$ , for the  $n$ -bit word. The output bit error probability,  $p$ , is related to  $P_0$  also, but not in a known simple way. The obvious bounds,

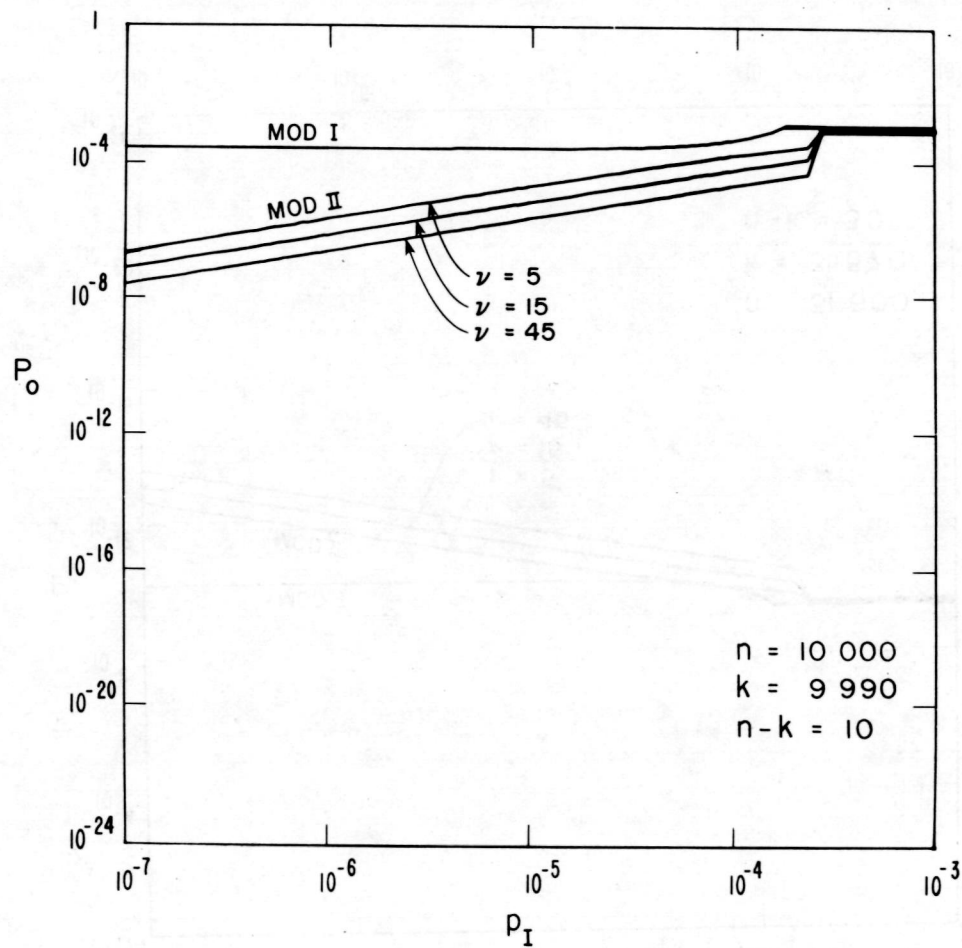


Figure 29. Output word error probability for  $n=10,000$  and  $k/n=0.999$ .



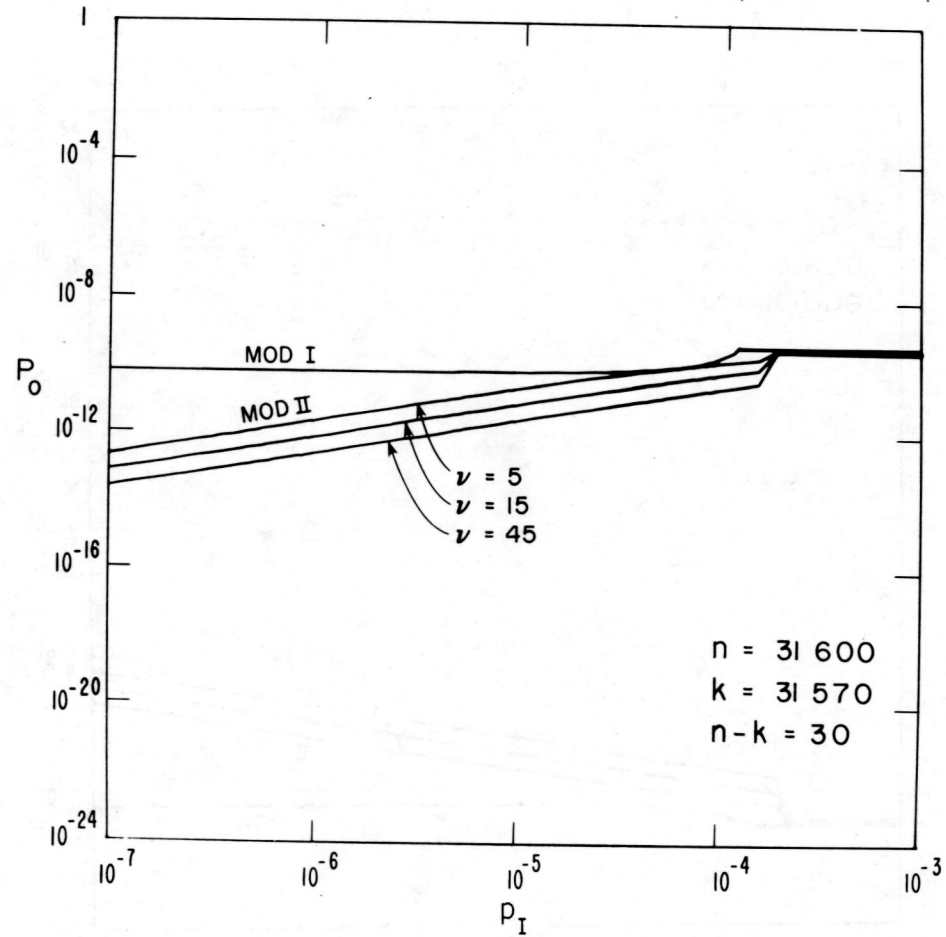


Figure 30. Output word error probability for  $n=31,600$  and  $k/n=0.999$ .

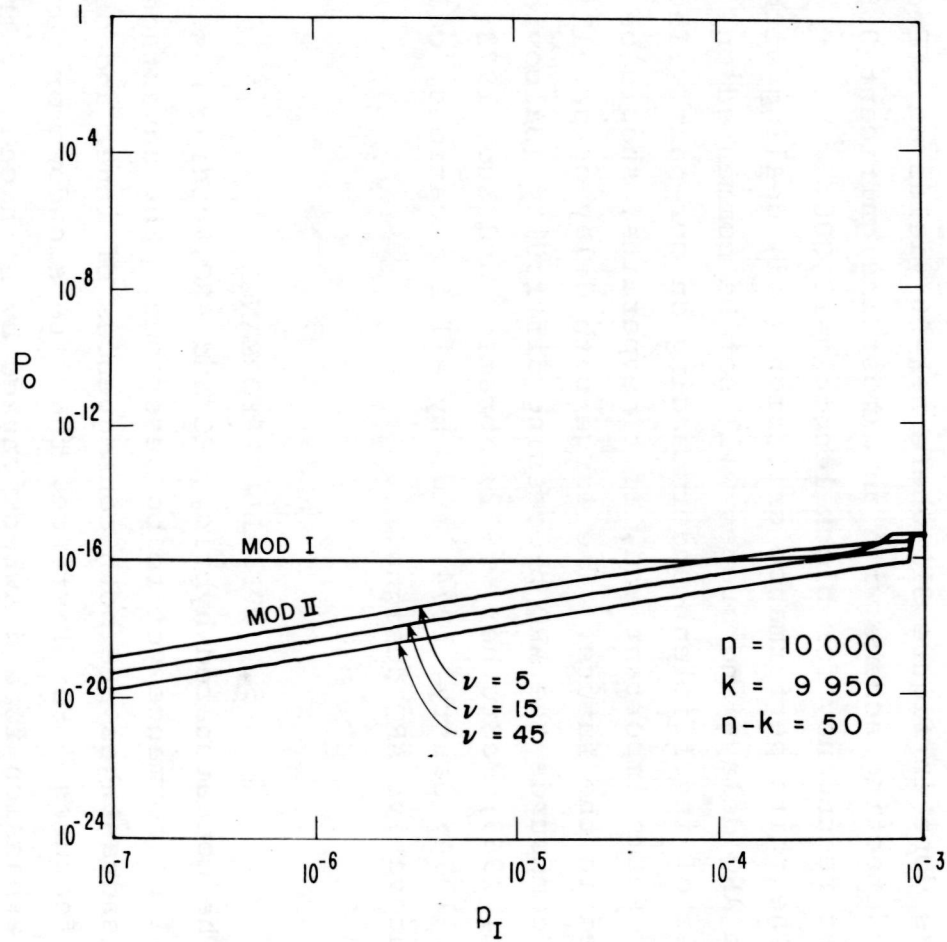


Figure 31. Output word error probability for  $n=10,000$  and  $k/n=0.995$ .



$$\frac{1}{n}P_0 \leq p \leq P_0 ,$$

can be sharpened (Viterbi, 1966) under some specific and elaborate circumstances. Unfortunately, all such refinements remain relatively ineffective compared to the gross  $2^{-(n-k)}$  bound on  $P_0$ . To assure that  $P$  and  $p$  never exceed  $10^{-12}$ , it suffices to set

$$n-k \geq \frac{12}{\log 2} \cong 40 .$$

This is done in figure 31, where  $n-k=50$  yields an effective error detecting scheme with an almost insignificant 0.5 percent redundancy for block length  $n=10,000$ .

The third performance criterion (iii) dealing with random ARQ delays and queues will not be commented on, because of insufficient understanding on our part. The matter seems important and, if time permits, should be studied in the future. The large path delay on satellite links compounds the ARQ processing difficulty (Balcovic and Muench, 1969; McGruther, 1972; Abramson and Kuo, 1973; Lucky, 1974; Sastry, 1974) and, by all indications, calls for innovative ARQ designs.

## 5. OVERALL PERFORMANCE

The concatenated hybrid, FEC and ARQ, coding has an overall performance yet to be described. The advantages and disadvantages are both consequences of inner and outer coder features. For instance, the rate slowdown or bandwidth expansion is a drawback caused by both coders, though in different ways. Another disadvantage is the implementation, maintenance, and other complexity related costs. The advantages boil down to one: more reliable data output.



This error rate reduction, of course, is also a resultant of the workings of both codes. Previously, we have described the two codecs separately; now we shall combine them and summarize the overall performance.

First, consider the overall output error probability. As pointed out in the ARQ performance discussion (sec. 4.2), this performance criterion is for all practical purposes the same  $P_0$  as observed at the ARQ output. It was shown that a modest investment in  $n-k \geq 40$  parity check bits for the outer code implies overall word and bit probabilities,  $P$  and  $p$ , at or below  $10^{-12}$ . Other parameters and system arrangements may reduce the error rate further, but not substantially. One concludes that a block code with  $n \geq 10,000$  and not less than 40 parity checks (which amounts to 0.4 percent redundancy) will provide error control to keep the output binary error probability at the desired level.

Our next concern is the overall throughput rate of the hybrid scheme. The work on inner and outer coders implies that the overall rate  $R$  cannot exceed the product of the two constituent rates  $R_I$  and  $R_0$ . If there are no ARQ repeats, then  $R = R_I R_0$ . In the presence of repeats,  $R < R_I R_0$ . The inner rate  $R_I$  was assumed to be  $1/2$ , or  $3/4$ , or perhaps  $2/3$ . The outer rate  $R_0$  is further reduced by ARQ. Since the effects are complicated, exact calculation was abandoned in favor of simpler bounds. Such upper and lower bounds on the relative rate  $R/R_I$  were presented in figures 22 to 28. The abscissa for these plots was the binary error probability,  $p_I$ , at the Viterbi decoder output. The  $p_I$  itself was shown to be a function of channel signal-to-noise ratio,  $E_b/N_0$  (dB), in figures 13 to 15, and subject to many assumptions. A merger of these two families of curves



produces the sought overall throughput rate,  $R$ , versus the signal-to-noise ratio, all capable of meeting the  $10^{-12}$  binary error rate objective. Four such relationships are plotted in figure 32. While many more such curves are possible, the given four appear rather typical. Note the pronounced  $E_b/N_0$  threshold effect. Above the threshold, data passes with negligible or no slowdown. Near the threshold, repeats and non-repeats occur with comparable frequency. Further below the threshold, the throughput  $R$  ceases and the link gets turned off by the ARQ.

Since different conditions and parameter choices give cause for distinct curves in figure 32, it seems useful to graphically lump together and characterize the various threshold curves. Such an attempt has led to figure 33 and, alas, the resultant mastercurve does not seem to be an accurate model for specific curves (as plotted in fig. 32). Hence a warning: Use figure 33 for rough estimates only. A key parameter in figure 33 is the  $n/v$  ratio. Given the same  $n/v$ , the rate  $1/2$  and rate  $3/4$  inner codes lead to almost indistinguishable characteristics, as far as the  $0.9 R_I R_0$ ,  $0.5 R_I R_0$ , and  $0.1 R_I R_0$  points are concerned. One concludes that the  $3/4$  rate inner code is preferable, since it offers more throughput for the above threshold operation.

It is seen from figures 32 and 33 that the  $E_b/N_0$  threshold occurs somewhere between 5 and 7 dB. More precise numbers can be pinned down with the help of methods and parameters presented earlier, but one wonders about the relative errors inherent in such estimates and the confidence levels they merit. Nevertheless, it is significant to stress that an operational signal-to-noise ratio,  $E_b/N_0=6.5$  dB, though it might be a ballpark figure,



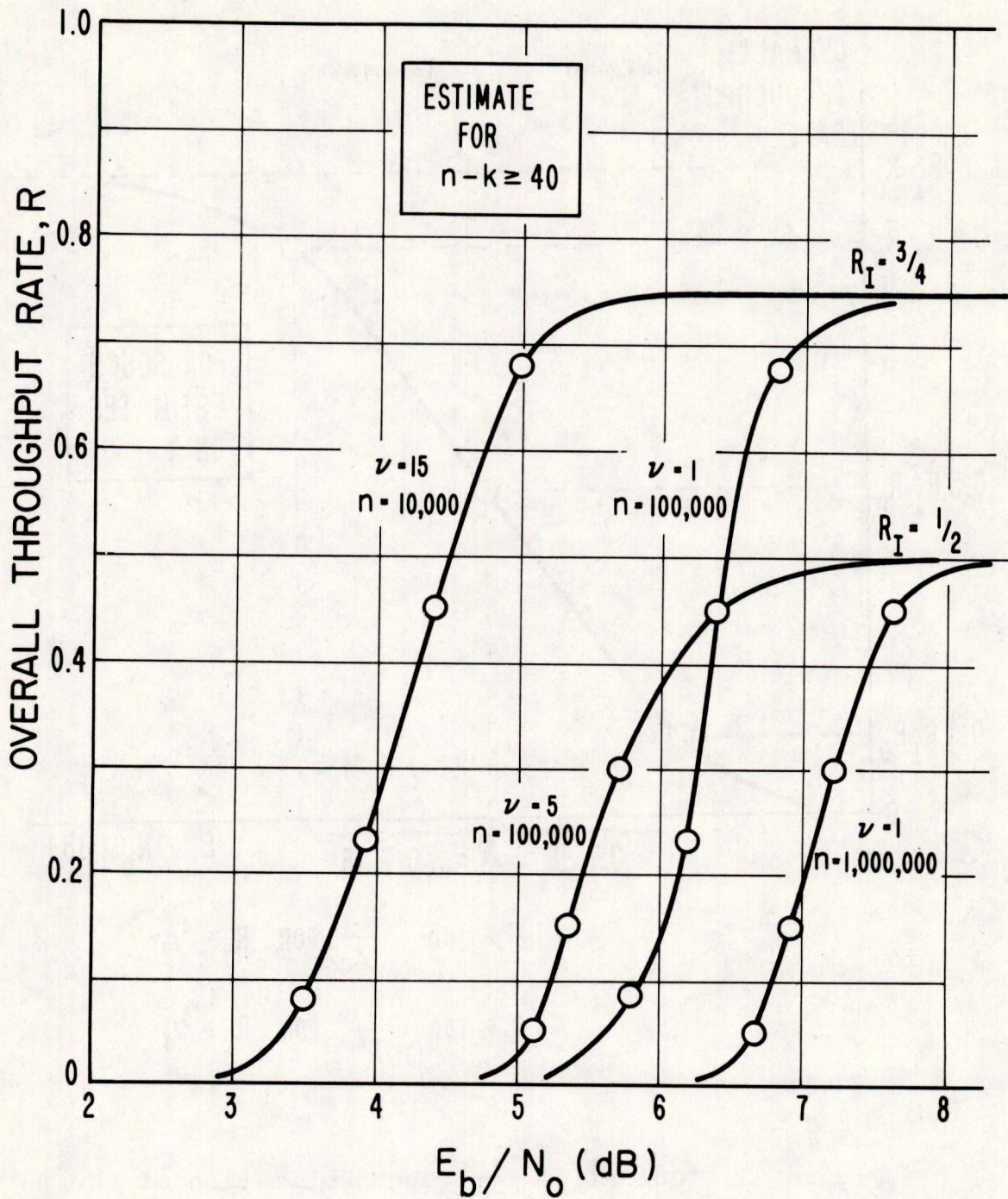


Figure 32. Estimated overall throughput rate for specific burst channels ( $\nu$ ), code length ( $n$ ), and signal-to-noise ratio ( $E_b/N_0$ ).



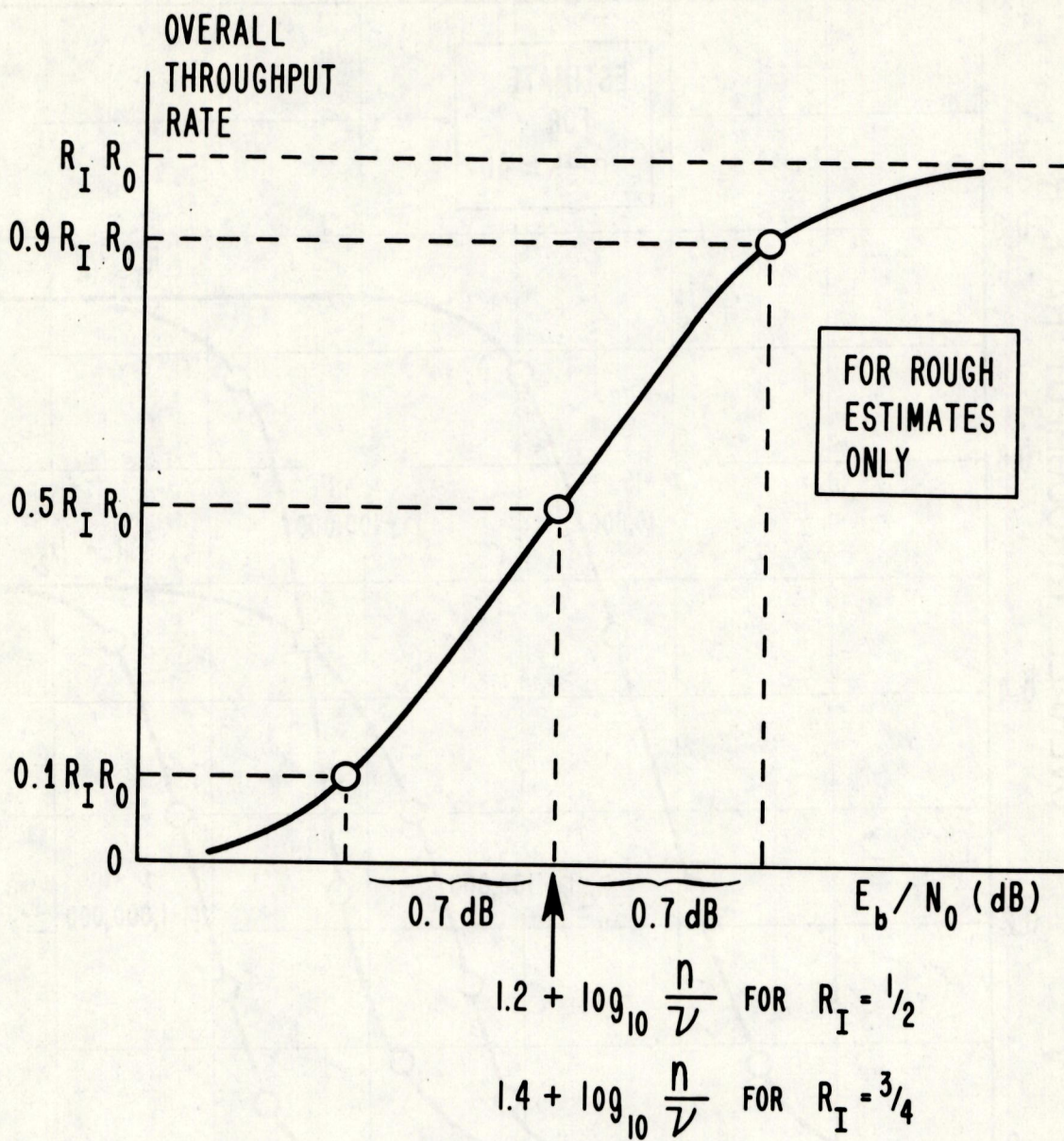


Figure 33. Guideline for rough estimation of the overall throughput rate.



is still some 14 dB below the uncoded extrapolated-measured requirement for the  $10^{-12}$  error rate level (McManamon et al., 1974). And, if uncoded operation were to be unsatisfactory due to irreducible error rate effects (see also earlier sec. 3.6), then error control coding would indeed become indispensable.

A brief final comment concerning figure 33 and the slight advantage of the rate 3/4 inner code. Perhaps, further increase of  $R_I$  could yield even better results. Taken to the ultimate limit,  $R_I \rightarrow 1$ , can one then discard the inner coder and the entire concatenated hybrid scheme? A simple check with figure 14, however, tells a different story. To operate at the required  $P_e$  level, the ARQ outer code by itself (i.e., without the inner code) would require  $E_b/N_0$  somewhere between 10 and 12 dB to stay above the threshold. Thus, a good inner FEC code is needed to provide reliable transmissions down to  $E_b/N_0 \approx 6$  dB. As mentioned earlier, this figure includes the 1-3 dB code rate loss, but fails to reflect the 33 percent to 100 percent bandwidth expansion (or transmission rate reduction).

## 6. CONCLUSIONS

Many coding alternatives have been considered here. The less suited ones were dismissed with little comment, the better ones were scrutinized in more detail. An alleged "first choice" error control code was selected. It turned out to be a hybrid of forward error correction (FEC) and automatic repeat request (ARQ), implemented by concatenation of two codes: an inner convolutional code with Viterbi decoding and an outer error detecting block code. To be certain that our chosen code is as good as it



appears, we have delved quite intensely in all its technical aspects. Features such as implementation, operational principles, present day state-of-the-art, costs, complexities, robustness, versatility, and the expected performance on realistic channels have been compiled, reviewed, and in some cases computed anew.

The main advantage of the selected codec is to ensure, with high confidence, practically error free operation. It can do so at its nominal throughput rate, as long as the normalized signal-to-noise ratio,  $E_b/N_0$  (dB), exceeds roughly 6 dB. If, due to a sudden fade, the signal-to-noise ratio falls below the threshold, the codec will effectively interrupt the data flow and stop delivery of erroneous digits.

The main disadvantage of the scheme is the need to transmit and process redundant bits. If the modems and the link bandwidths are fixed, the only way to accommodate additional bits is to reduce the information throughput. To maintain constant information throughput, on the other hand, requires bandwidth expansion. The additional occupied bandwidth is given by  $(1-R)/R$ , where  $R$  is the overall throughput rate. Thus, a 33 percent expansion appears necessary for  $R=3/4$  and the system given. The second disadvantage of introducing codecs is the burden of the devices themselves. Fortunately, the coding scheme selected can be simply implemented. Whether viewed in hardware or software sense, the added system is of the same complexity as a mini-computer.

A final unresolved item concerns the random queues and delays caused by the ARQ on the long-delay satellite links. The strategy and proper handling of these possibly large queues is a complex, interrelated process involving



some balance between data processing and communications. System studies of these queues, particularly for network applications, will require computer simulation because of the analytical complexities discussed in this report.



## 7. REFERENCES

- Abramson, N., and F.F. Kuo (1973), Computer-Communication Networks (Prentice-Hall, Englewood Cliffs, N.J.).
- Balcovic, M.D., and P.E. Muench (1969), Effect of propagation delay, caused by satellite circuits, on data communication systems that use block retransmission for error correction, Conf. Record of 1969 ICC, Boulder, Colo., pp. 29/31-29/36.
- Batson, B.H., R.W. Moorehead, and S.Z.H. Taqvi (1972), Simulation results for the Viterbi decoding algorithm, NASA Tech. Rept. R-396, Manned Spacecraft Center, Houston, Tex.
- Benice, R.J., and A.H. Frey (1964), An analysis of retransmission systems, IEEE Trans. Commun. Technol., COM-12, pp. 135-145.
- Berlekamp, E.R. (1968), Algebraic Coding Theory (McGraw-Hill Book Co., New York, N.Y.).
- Burton, H.O. (1970), A survey of error correcting techniques for data on telephone facilities, Conf. Record of 1970 ICC, San Francisco, Calif., pp. 16/25-16/32.
- Bussgang, J.J. (1965), Some properties of binary convolutional code generators, IEEE Trans. Inform. Theory, IT-11, pp. 90-100.
- Cahn, C.R., G.K. Huth, and C.R. Moore (1973), Simulation of sequential decoding with phase-locked demodulation, IEEE Trans. Commun., COM-21, pp. 89-97.
- Cain, J.B. (1971), Hard decision sequential decoding, Memo. Rept. No. 43, Radiation, Melbourne, Fla.



- Cain, J.B. (1972), An analysis of two branch synchronization schemes for convolutional codes, Memo. Rept. No. 46, Radiation, Melbourne, Fla.
- Clark, G.C., and R.C. Davis (1971), Two recent applications of error-correction coding to communications system design, IEEE Trans. Commun. Technol., COM-19, pp. 856-863.
- Clark, G.C. (1971), Implementation of maximum likelihood decoders for convolutional codes, Proc. ITC 1971, Washington, D.C., VII, pp. 449-459.
- Dodds, J.G. (1973), Analysis and simulation of parallel sequential decoding, Tech. Doc. 290, Naval Elec. Lab. Center, San Diego, Calif.
- Fano, R.M. (1963), A heuristic discussion of probabilistic decoding, IEEE Trans. Inform. Theory, IT-9, pp. 64-74.
- Forney, G.D. (1966), Concatenated Codes (MIT Press, Cambridge, Mass.).
- Forney, G.D. (1967), Review of random tree codes, NASA CR 73176, Final Report, NASA Ames Res. Cent., Moffett Field, Calif.
- Forney, G.D., and R.M. Langelier (1969), A high-speed sequential decoder for satellite communications, Conf. Record of 1969 ICC, Boulder, Colo., pp. 39/9-39/17.
- Forney, G.D. (1970), Coding and its application in space communication, IEEE Spectrum, Vol. 7, No. 6, pp. 47-58.
- Forney, G.D., and E.K. Bower (1971), A high-speed sequential decoder: Prototype design and test, IEEE Trans. Commun. Technol., COM-19, pp. 821-835.



- Forney, G.D. (1972), Convolutional codes II: Maximum likelihood decoding, Tech. Dept. 7004-1, Stanford Electronic Labs., Stanford, Calif.
- Forney, G.D. (1973), The Viterbi algorithm, Proc. IEEE (Invited Paper), Vol. 61, pp. 268-278.
- Gallager, R.G. (1968), Information Theory and Reliable Communication (John Wiley and Sons, New York, N.Y.).
- Gilhousen, K.S., J.A. Heller, I.M. Jacobs, and A.J. Viterbi (1971), Coding systems study for high data rate telemetry links, NASA CR 114278, Rept. prepared by Linkabit of San Diego, Calif., for NASA Ames Res. Cen., Moffett Field, Calif.
- Gilhousen, K.S., and D.R. Lumb (1972), A very high speed hard decision sequential decoder, Conf. Record of 1972 NTC, Houston, Tex., pp. 13C/1-13C/5.
- Goldman, H.D., W. Rowan, and M. Tolhurst (1969), Threshold decoding selected rate one-half and one-third convolutional codes, IEEE Trans. Inform. Theory, IT-15, pp. 179-184.
- Heller, J.A. (1968), Short constraint length convolutional codes, JPL Space Programs Summary 37-54, Vol. III, Pasadena, Calif., pp. 171-177.
- Heller, J.A. (1969), Improved performance of short constraint length convolutional codes, JPL Space Programs Summary 37-56, Vol. III, Pasadena, Calif., pp. 83-84.
- Heller, J.A., and I.M. Jacobs (1971), Viterbi decoding for satellite and space communications, IEEE Trans. Commun. Technol., COM-19, pp. 835-848.
- Hoffman, L.B., and J.P. Odenwalder (1972), Hybrid and concatenated coding application, Proc. ITC 72, Los Angeles, Calif., VIII, pp. 522-531.

- Jacobs, I.M. (1967), Sequential decoding for efficient communication from deep space, IEEE Trans. Commun. Technol., COM-15, pp. 492-501.
- Jacobs, I.M. (1974), Practical applications of coding, IEEE Trans. Inform. Theory, IT-20, pp. 305-310.
- Jelinek, F. (1969), A fast sequential decoding algorithm using a stack, IBM J. Res. and Dev., Vol. 13, pp. 675-685.
- Kobayashi, H. (1971), Correlative level coding and maximum-likelihood decoding, IEEE Trans. Inform. Theory, IT-17, pp. 586-594.
- Kohlenberg, A., and G.D. Forney (1968), Convolutional coding for channels with memory, IEEE Trans. Inform. Theory, IT-14, pp. 618-626.
- Layland, J.W., and W.A. Lushbaugh (1971), A flexible high-speed sequential decoder for deep space channels, IEEE Trans. Commun. Technol., COM-19, pp. 813-820.
- Lin, S., and H. Lyne (1967), Some results on binary convolutional code generators, IEEE Trans. Inform. Theory, IT-13, pp. 134-139.
- Lucky, R.W., J. Salz, and E.J. Weldon, Jr. (1968), Principles of Data Communications (McGraw-Hill Book Co., New York, N.Y.).
- Lucky, R.W. (1973), A survey of the communication theory literature: 1968-1973, IEEE Trans. Inform. Theory, IT-19, pp. 725-739.
- Lumb, D.R. (1969), Test and preliminary flight results on the sequential decoding of convolutionally encoded data from Pioneer IX, Conf. Record of 1969 ICC, Boulder, Colo., pp. 39/1-39/8.



- Massey, J.L. (1963), Threshold Decoding (MIT Press, Cambridge, Mass.).
- Massey, J.L., and M.K. Sain (1968), Trunk and tree searching properties of the Fano sequential decoding algorithm, Proc. of 6th Allerton Conf., Univ. of Illinois, pp. 153-160.
- Massey, J.L., and D.J. Costello (1971), Nonsystematic convolutional codes for sequential decoding in space applications, IEEE Trans. Commun. Technol., COM-19, pp. 806-813.
- Massey, J.L. (1973), Coding techniques for digital communications, Tutorial Notes at 1973 ICC, Seattle, Wash.
- McGruther, W.G. (1972), Throughput of high speed data transmission systems using block retransmission error control schemes over voicebandwidth channels, Conf. Record of 1972 ICC, Philadelphia, Penn., pp. 15/19-15/24.
- McManamon, P.M., J.R. Juroshek, and M. Nesenbergs (1974), Study of satellite frequency requirements for the U.S. Postal Service electronic mail system; Volume II: Frequency allocations and network designs, U.S. Dept. of Commerce, Office of Telecommunications Report, OTR 74-27, Boulder, Colo.
- Nesenbergs, M. (1963), Comparison of the 3-out-of-7 ARQ with Bose-Chaudhuri-Hocquenghem coding systems, IEEE Trans. Commun. Systems, CS-11, pp. 202-212.
- Odenwalder, J.P., K.S. Gilhousen, J.A. Heller, I.M. Jacobs, F. Jelinek, and A.J. Viterbi (1972), Hybrid coding systems study, NASA CR 114486, Rept. prepared by Linkabit of San Diego, Calif., for NASA Ames Res. Cen., Moffett Field, Calif.



- Omura, J.K. (1969), On the Viterbi decoding algorithm, IEEE Trans. Inform. Theory, IT-15, pp. 177-179.
- Omura, J.K. (1971), Optimal receiver design for convolutional codes and channels with memory via control theoretical concepts, Inform. Sci., Vol. 3, pp. 243-266.
- Paaske, E. (1974), Short binary convolutional codes with maximum free distance for rates  $2/3$  and  $3/4$ , IEEE Trans. Inform. Theory, IT-20, pp. 683-689.
- Peterson, W.W., and E.J. Weldon (1972), Error-correcting Codes, second edition (MIT Press, Cambridge, Mass.).
- Robinson, J.P., and A.J. Bernstein (1967), A class of binary recurrent codes with limited error propagation, IEEE Trans. Inform. Theory, IT-13, pp. 106-113.
- Rudolph, L.D. (1967), A class of majority logic decodable codes, IEEE Trans. Inform. Theory, IT-13, pp. 305-307.
- Sastry, A.R.K. (1974), Error control techniques for satellite communications: An overview, Conf. Record of 1974 ICC, Minneapolis, Minn., pp. 6A/1-6A/5.
- Savage, J.E. (1966), Sequential decoding - the computation problem, Bell System Tech. J., Vol. 45, pp. 149-176.
- Townsend, R.W., and R.N. Watts (1964), Effectiveness of error control in data communications over a switched telephone network, Bell System Tech. J., Vol. 43, pp. 2611-2638.
- van Duuren, H.C.A. (1961), Error probability and transmission speed on circuits using error detection and automatic repetition of signals, IRE Trans. Commun. Systems, CS-9, pp. 38-50.
- Viterbi, A.J. (1966), Principles of Coherent Communications (McGraw-Hill Book Co., New York, N.Y.).
- Viterbi, A.J. (1967), Error bounds for convolutional codes and an asymptotically optimum decoding algorithm, IEEE Trans. Inform. Theory, IT-13, pp. 260-269.



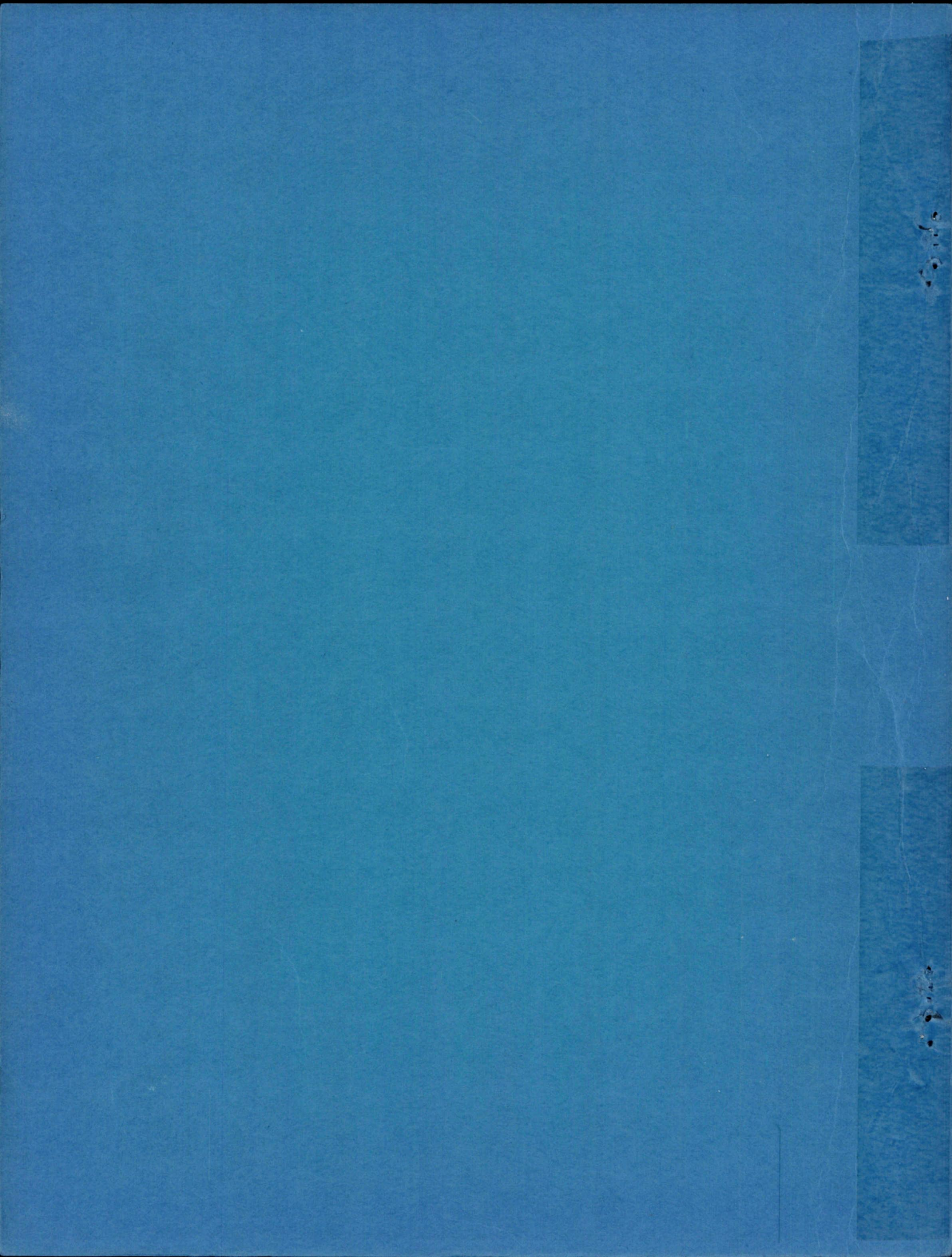
- Viterbi, A.J., and J.P. Odenwalder (1969), Further results on optimal decoding of convolutional codes, IEEE Trans. Inform. Theory (Corresp.), IT-15, pp. 732-734.
- Viterbi, A.J. (1971), Convolutional codes and their performance in communication systems, IEEE Trans. Commun. Technol., COM-19, pp. 751-772.
- Wolf, J.K. (1973), A survey of coding theory: 1967-1972, IEEE Trans. Inform. Theory, IT-19, pp. 381-389.
- Wozencraft, J.M. (1957), Sequential decoding for reliable communication, Conf. Record of 1957 Nat. IRE Conv., Vol. 5, Pt. 2, pp. 11-25.
- Wozencraft, J.M., and B. Reiffen (1961), Sequential Decoding (MIT Press, Cambridge, Mass.).
- Zeoli, G.W. (1971), Coupled decoding of block-convolutional concatenated codes, Ph.D. Dissertation, Dept. Elec. Eng., Univ. of California, Los Angeles.
- Zigangirov, K.S. (1966), Some sequential decoding procedures, Problemy Peredachi Informatsii (USSR), Vol. 2, pp. 13-25.



## BIBLIOGRAPHIC DATA SHEET

	1. PUBLICATION OR REPORT NO.	2. Gov't Accession No.	3. Recipient's Accession No.
4. TITLE AND SUBTITLE Study of Error Control Coding for the U.S. Postal Service Electronic Message System		5. Publication Date December	6. Performing Organization Code OT/ITS
7. AUTHOR(S) Martin Nesenbergs		9. Project/Task/Work Unit No.	
8. PERFORMING ORGANIZATION NAME AND ADDRESS U.S. Department of Commerce Office of Telecommunications Institute for Telecommunication Sciences 325 Broadway, Boulder, Colorado 80302		10. Contract/Grant No.	
11. Sponsoring Organization Name and Address U.S. Postal Service 11711 Park Lawn Drive Rockville, Maryland 20852		12. Type of Report and Period Covered	
		13.	
14. SUPPLEMENTARY NOTES			
15. ABSTRACT (A 200-word or less factual summary of most significant information. If document includes a significant bibliography of literature survey, mention it here.) A U.S. Postal Service (USPS) electronic message system could incorporate many types of error control coding, or no coding at all. This report reviews a variety of possible codes, lists their advantages and disadvantages, and selects a preferred alternative. It turns out to be a concatenation of an inner convolutional (rate 1/2 to rate 3/4) code with Viterbi decoding, and an outer long block, high efficiency code. The two codes have separate functions, in the sense that the inner code performs forward error correction and the outer code does error detection only. The report describes the structures, properties, and implementations of the coding hybrid. After that, the performance of the preferred coding scheme is estimated. The resultant error probability gains, which are shown to be considerable, are balanced against system slowdown and bandwidth expansion.			
16. Key words (Alphabetical order, separated by semicolons) ARQ; coding gains; concatenated codes; error probability; FEC; hybrid operation; modem losses; throughput; Viterbi decoding			
17. AVAILABILITY STATEMENT <input checked="" type="checkbox"/> UNLIMITED. <input type="checkbox"/> FOR OFFICIAL DISTRIBUTION.		18. Security Class (This report) Unclassified	20. Number of pages 92
		19. Security Class (This page) Unclassified	21. Price:





STUDY OF ERROR CONTROL CODING  
FOR THE U.S. POSTAL SERVICE  
ELECTRONIC MESSAGE SYSTEM

Martin Nesenbergs



STUDY OF ERROR CONTROL CODING FOR THE U.S. POSTAL SERVICE  
ELECTRONIC MESSAGE SYSTEM

Prepared by

Martin Nesenbergs

of the

Institute for Telecommunication Sciences  
Office of Telecommunications  
U.S. Department of Commerce  
Boulder, Colorado 80302

Prepared for

Advanced Mail Systems Directorate  
U.S. Postal Service  
Rockville, Maryland 20852

May 1975

USPS/1702-113

PB-252689

## PREFACE

The study presented in this report was supported by the Advanced Mail Systems Development Directorate of the U.S. Postal Service, Rockville, Maryland 20852, under Agreement Number 74-01237. The U.S.P.S. technical monitor for the program was Mr. Ralph P. Marcotte of the Systems Division. The program director at the Institute for Telecommunication Sciences, Office of Telecommunications was Dr. Peter M. McManamon.

Other staff members of the Institute for Telecommunication Sciences who assisted in technical review were Dr. Arthur D. Spaulding and Mr. John J. Juroshek. Typing and drafting assistance were provided by Ms. Patricia A. Moreno and Ms. Mary McClanahan.

The opinions expressed herein are those of the author and are not to be construed as representing policies or doctrines of the U.S.P.S.



## TABLE OF CONTENTS

	Page
PREFACE	iii
LIST OF FIGURES	vii
LIST OF TABLES	x
GLOSSARY	xi
ABSTRACT	1
1. INTRODUCTION	1
2. OVERALL CODEC ALTERNATIVES	3
2.1. Candidate Codes	3
2.2. The Second and Third Finalists	5
3. INNER CODE FOR THE CONCATENATED HYBRID	8
3.1. Convolutional Codes	8
3.2. Systematic and Nonsystematic Aspects	13
3.3. Decoder Types	14
3.4. Viterbi Decoding	21
3.5. Performance on the Ideal Channel	29
3.6. Performance on Realistic Channels	32
4. OUTER CODE FOR THE CONCATENATED HYBRID	39
4.1. Codec	39
4.2. ARQ Performance	45
5. OVERALL PERFORMANCE	66
6. CONCLUSIONS	71
7. REFERENCES	74

## LIST OF FIGURES

		Page
Figure 1.	Hybrid FEC and ARQ scheme with concatenated coding.	9
Figure 2.	Convolutional encoder of rate 2/3, constraint length 48, with 10 delay taps for each parity check.	11
Figure 3.	Systematic and nonsystematic versions of rate 1/2 convolutional encoder.	12
Figure 4.	State diagram of the systematic code.	15
Figure 5.	State diagram of the nonsystematic code.	16
Figure 6.	Basic feedback decoder for the rate 1/2 code.	19
Figure 7.	Example of a feedback decoder - the MLD decoder for rate 2/3 convolutional code.	20
Figure 8.	Trellis diagram and encoder outputs.	24
Figure 9.	The role of branch metric/best path metric pairs in successful decoding of three errors.	26
Figure 10.	Unsuccessful decoding of five errors.	28
Figure 11.	Various convolutional code performances on the ideal theoretical channel.	30
Figure 12.	Definition of coding gains and modem losses.	34
Figure 13.	Viterbi decoding gains on the ideal and extrapolated-measured channels.	36
Figure 14.	Actual coding gains for rate 1/2 and 3/4, soft decision, Viterbi decoders.	37



Figure 15.	The effects of irreducible error probability.	38
Figure 16.	Outer encoder.	41
Figure 17.	Outer decoder.	42
Figure 18.	The ARQ duplex loop.	46
Figure 19.	Synopsis of ARQ.	48
Figure 20.	The least and most repeats in the ARQ loop.	50
Figure 21.	The least and most errors in the ARQ loop.	51
Figure 22.	The bounds on throughput rate for $n=10,000$ and $k=9,900$ .	54
Figure 23.	The bounds on throughput rate for $n=10,000$ and $k=9,950$ .	55
Figure 24.	The bounds on throughput rate for $n=10,000$ and $k=9,990$ .	56
Figure 25.	The bounds on throughput rate for $n=31,600$ and $k/n=0.99$ .	58
Figure 26.	The bounds on throughput rate for $n=100,000$ and $k/n=0.99$ .	59
Figure 27.	The bounds on throughput rate for $n=316,000$ and $k/n=0.99$ .	60
Figure 28.	The bounds on throughput rate for $n=1,000,000$ and $k/n=0.99$ .	61
Figure 29.	Output word error probability for $n=10,000$ and $k/n=0.999$ .	63
Figure 30.	Output word error probability for $n=31,600$ and $k/n=0.999$ .	64
Figure 31.	Output word error probability for $n=10,000$ and $k/n=0.995$ .	65

- Figure 32. Estimated overall throughput rate for specific burst channels ( $\nu$ ), code length ( $n$ ), and signal-to-noise ratio ( $E_b/N_0$ ). 69
- Figure 33. Guideline for rough estimation of the overall throughput rate. 70



LIST OF TABLES

	Page
Table 1. Comparison of Hybrid, FEC and ARQ, overall coding alternatives. Code rates are assumed in the 0.50-0.75 range.	6
Table 2. Comparison of forward acting decoders for convolutional inner code candidates. Code rates are assumed in the 0.50-0.75 range.	17
Table 3. Possible error detecting codes for ARQ.	44

## GLOSSARY

$A_k$	k-th acknowledgement
ARQ	Automatic repeat request
codec	(en)coder and decoder
d	minimum distance, block code
D	path delay
$d_f$	free distance, convolutional code
$E_b$	energy per information bit
FEC	forward error correction
$G_a$	actual coding gain
$G_b$	basic coding gain
k	number of information bits, block code
K	constraint length, convolutional code
L	path merge length, Viterbi decoding
$L_c$	modem loss, coded
$L_u$	modem loss, uncoded
M	number of phases, M-ary phase shift keying
MLD	majority logic decision
modem	modulator and demodulator
MPSK	M-ary phase shift keying
n	length of codeword, block code
N	noise power
$N_0$	noise power spectral density
$N(k)$	number of k's
v	burst length parameter
p	binary error probability, overall output
P	word error probability, overall output
$p_e$	binary error probability, data channel or in general context



$p_I$	binary error probability, inner decoder
$P_0$	word error probability, outer decoder
$p_n(j)$	probability that $j$ out of $n$ bits are in error
$\text{Pr}(\dots)$	probability of event (...)
PSK	phase shift keying
$Q$	state qualifier for ARQ
$R$	overall throughput rate
$R_I$	rate of the inner code
$R_0$	rate of the outer code
$S$	signal power
$W$	bandwidth
WGN	white Gaussian noise
$W_k$	$k$ -th word

STUDY OF ERROR CONTROL CODING FOR THE U.S. POSTAL SERVICE  
ELECTRONIC MESSAGE SYSTEM

Martin Nesenbergs\*

ABSTRACT

A U.S. Postal Service (USPS) electronic message system could incorporate many types of error control coding, or no coding at all. This report reviews a variety of possible codes, lists their advantages and disadvantages, and selects a preferred alternative. It turns out to be a concatenation of an inner convolutional (rate 1/2 to rate 3/4) code with Viterbi decoding, and an outer long block, high efficiency code. The two codes have separate functions, in the sense that the inner code performs forward error correction and the outer code does error detection only. The report describes the structures, properties, and implementations of the coding hybrid. After that, the performance of the preferred coding scheme is estimated. The resultant error probability gains, which are shown to be considerable, are balanced against system slowdown and bandwidth expansion.

Key words: ARQ, coding gains, concatenated codes, error probability, FEC, hybrid operation, modem losses, throughput, Viterbi decoding

1. INTRODUCTION

Digital communication via satellite has been considered for the USPS electronic message system. Extensive background studies (McManamon et al., 1974) have been directed at the large scale system features, such as

---

\*The author is with the Institute for Telecommunication Sciences, Office of Telecommunications, U.S. Department of Commerce, Boulder, Colorado 80302



useful frequency ranges, present and future traffic loads, network configurations, transmission impediments, and state of the art engineering tradeoffs. System design, operation, and performance verification have to wait until basic feasibility questions are answered in the affirmative. One such question concerns the quality of the received output data. Under the heading "System Performance", this multi-faceted issue was raised in section 7, volume II of McManamon et al. (1974).

The expected data rates, bandwidths, signal powers, and the noise densities described above revealed that perhaps the hardest performance criterion to be satisfied is the bit error rate, also called error probability. A target value of  $10^{-12}$  was given for this error rate, but at the time of the above report, it was not entirely clear if or how this could be met. Section 7.8, volume II of McManamon et al. (1974) suggested error control codes as a possible alternative to ensure sufficiently low error rates. Unfortunately, being quite brief and introductory in nature, the above treatment raised more questions about coding than it managed to answer. It also overlooked some important and useful coding techniques.

Hence, the purpose of this study - a second look at the error control codes as they could be applied today to a high rate satellite data network. The approach taken is (a) to consider all promising alternatives, (b) to identify the most promising alternative as the "first choice", (c) to verify that the first choice can be practically implemented and operated, and (d) to estimate the performance, i.e., gains and losses, of the proposed scheme.

Before proceeding with the coding tasks, we must emphasize a few key points. The data network is assumed to consist of two-way links over which forward and feedback messages can be sent, if needed. The data terminals are of sufficient size to justify the initial cost and maintenance of the codecs (short for encoder-decoder). The implementation of the coding part of the terminal will be shown as relatively modest, of complexity somewhere between a minicomputer and a microprocessor. Of course, the device can always be incorporated in a large general purpose computer (if such is available), or it can be constructed as a special unit. Finally, the codecs are stipulated to function perfectly, so as not to introduce additional errors in the received messages.

## 2. OVERALL CODEC ALTERNATIVES

### 2.1. Candidate Codes

The question of whether to use any error control coding at all cannot be answered without examination of the more promising codec alternatives. This section compares the relative advantages and disadvantages of the best presently known coding schemes, and then ventures to select a "first choice" overall error control system for a U.S. Postal Service network. Perhaps even the best choice is not good enough. Accordingly, later sections will scrutinize the details of the candidate codecs, with emphasis on implementation, operation, performance, and anticipation of sundry difficulties.

The presence of one-way, two-way, and more elaborate data channel arrangements within the satellite network



permits forward-acting error correction (FEC), automatic error detection and repeat requests (ARQ), and any hybrid combination of the two. Furthermore, most channel arrangements can function well with a variety of known codec types (Gallager, 1968; Massey, 1973; Wolf, 1973; Sastry, 1974). The assortment of codes can be divided into three broad classes: the block codes (Berlekamp, 1968; Peterson and Weldon, 1972), the convolutional or tree codes (Wozencraft, 1957; Viterbi, 1971), and all the other codes. Among the others, it is befitting to single out concatenated codes (Forney, 1966; Zeoli, 1971; Hoffman and Odenwalder, 1972), which are constructed interactively from two or more codes.

A previous investigation (McManamon et al., 1974) indicated that a hybrid, FEC and ARQ, with joint correction and detection roles, may be preferable over other overall coding schemes for reasons of flexibility and error correcting performance. Basically, a hybrid arrangement can do nearly everything that the pure FEC and ARQ options can do. When transmission conditions are good and channel errors occur seldom, the hybrid resembles FEC in throughput rate and error probability performance. When the channel deteriorates and more errors confront the decoder, the hybrid can be relied on to reject error infested data portions. The hybrid then turns to ARQ action and asks for repetition. No matter how low the signal-to-noise ratio, or whatever the channel malfunction, the output error probability can be kept at or below a level determined by the code. The principles, implementation, and estimated performance of ARQs appears the same for high and low speed data links. The low speed cases have been extensively reviewed by van Duuren (1961), Nesenbergs (1963), Townsend and Watts (1964), Benice and Frey (1964),

and Burton (1970). From now on we will assume that some hybrid scheme is the most suitable for the high speed digital message network.

Our next task is to compare the alternative implementations of the hybrid scheme. Here we have a number of options, including the previously mentioned block code with correction threshold (McManamon et al., 1974). Besides that, there are quite a few others. A selected list is tabulated in table 1. The table applies to code rates in the 0.50 to 0.75 range, assuming negligible ARQ slowdowns. The left hand column of table 1 lists six alternatives for the hybrid codec. The other five columns summarize the key codec characteristics in an informative way. The final, i.e., the rightmost, column attempts to rate the hybrid schemes in order of their suitability for the system. Actually, only the first three most promising schemes are seriously contemplated; the others are dismissed as being not suited.

The three leading candidates are:

- (1) Concatenated inner convolutional code and an outer block code.
- (2) Concatenated inner convolutional code with an outer convolutional code.
- (3) Convolutional code with computation threshold.

## 2.2. The Second and Third Finalists

Since considerable subsequent discussion will deal with alternative (1), the intent here is to comment briefly on choices (2) and (3). The second finalist (2) differs from (1) only in the non-block nature of the outer code. This is not a serious drawback, as the error detecting power of the outer code is seemingly not lessened. The



Table 1. Comparison of Hybrid, FEC and ARQ, overall coding alternatives. Code rates are assumed in the 0.50-0.75 range.

Codec Type	CHARACTERISTIC					Relative Rating
	Error Rate Performance	Experimental Evidence	Logic Speed Data Speed	Cost and Complexity	Others	
Block code with correction/detection threshold	Poor	None known	10	Very high	Needs inter-leaving for deep fades	Not suited
Convolutional code with computation threshold	Reasonable to good	None known	3	Medium	ARQ sync problem	3
Concatenated code of inner block and outer block codes	Poor to mediocre	Some	10	High	Inter-leaving may help	Not suited
Concatenated code of inner convolutional and outer block codes	Nearly optimal	Well tested	1-3	Low		1
Concatenated code of inner block and outer convolutional codes	Unknown	None known	10	High	ARQ sync problem	Not suited
Concatenated code of inner convolutional and outer convolutional codes	Perhaps good	None known	1-3	Reasonable	ARQ sync problem	2

only problem concerns frame timing of repeat segments in the ARQ operation. The encoded sequence keeps continuously evolving without any punctuation marks. Some reference points, called commas, would have to be implanted for identification of repeat segments. Both transmitter and receiver must have exact knowledge where the repeats start and end, and which are multiple repeats. Otherwise, the inner code of (2) can be the same as in (1).

The third finalist (3) is a non-concatenated convolutional code with a modified decoding algorithm. Consider an early form of a sequential decoder (Wozencraft, 1957; Wozencraft and Reiffen, 1961; Fano, 1963; Savage, 1966; Jacobs, 1967; Gallager, 1968; Forney and Langelier, 1969). This device seems to be quite capable of forward acting error correction, as long as the error rate is sufficiently low. As the error rate increases, however, the computational job of the sequential decoder grows. By necessity, every decoder can handle a certain computation rate and not more. When the noisy input data warrants computation in excess of the computational ceiling, input data would be lost if not for buffer storage. Unfortunately, a buffer of finite size carries a certain probability of overflow and associated implications of output errors or data loss. Both theory and practice have shown that the buffer overflow is of primary concern in the design and operation of sequential decoders.

In scheme (3) one proposes to do the following. Permit a small buffer storage only. When the computation load grows and indicates impending overflow, let the ARQ operation take over. Clearly, this strategy can be used with different types of decoders, including simplified versions of the sequential machine itself. Since sequential decoding of convolutional codes is a powerful (if not cheap) error



control tool, one expects that scheme (3) can be engineered to perform well. The only nagging doubts concern the previously mentioned ARQ frame timing and identification for convolutional coding, and the lack of experimental evidence for this particular hybrid technique.

### 3. INNER CODE FOR THE CONCATENATED HYBRID

#### 3.1. Convolutional Codes

In this section we start a detailed scrutiny of the preferred alternative for the concatenated hybrid FEC and ARQ scheme, that is, the first choice of table 1. Specifically, we are concerned with the inner code in the concatenated arrangement shown in figure 1.

A forward error correcting convolutional code seems to be the best choice here. In fact, the FEC convolutional coding systems that have been built, simulated, and/or tested show an abundance of advantages over comparable rate block codes (Kohlenberg and Forney, 1968; Heller, 1968, 1969; Forney, 1970; Clark, 1971; Massey and Costello, 1971; Viterbi, 1971; Cain, 1972; Cahn et al., 1973; Jacobs, 1974). The crucial points can be summarized as follows:

- (a) Convolutional coders are state-of-the-art now. Systems have been built and tested in sufficient numbers to feel quite confident about these devices. One manufacturer has a number of ready made versions on the market. While other type FEC coders appear to require further research and development, good convolutional coders are already here.

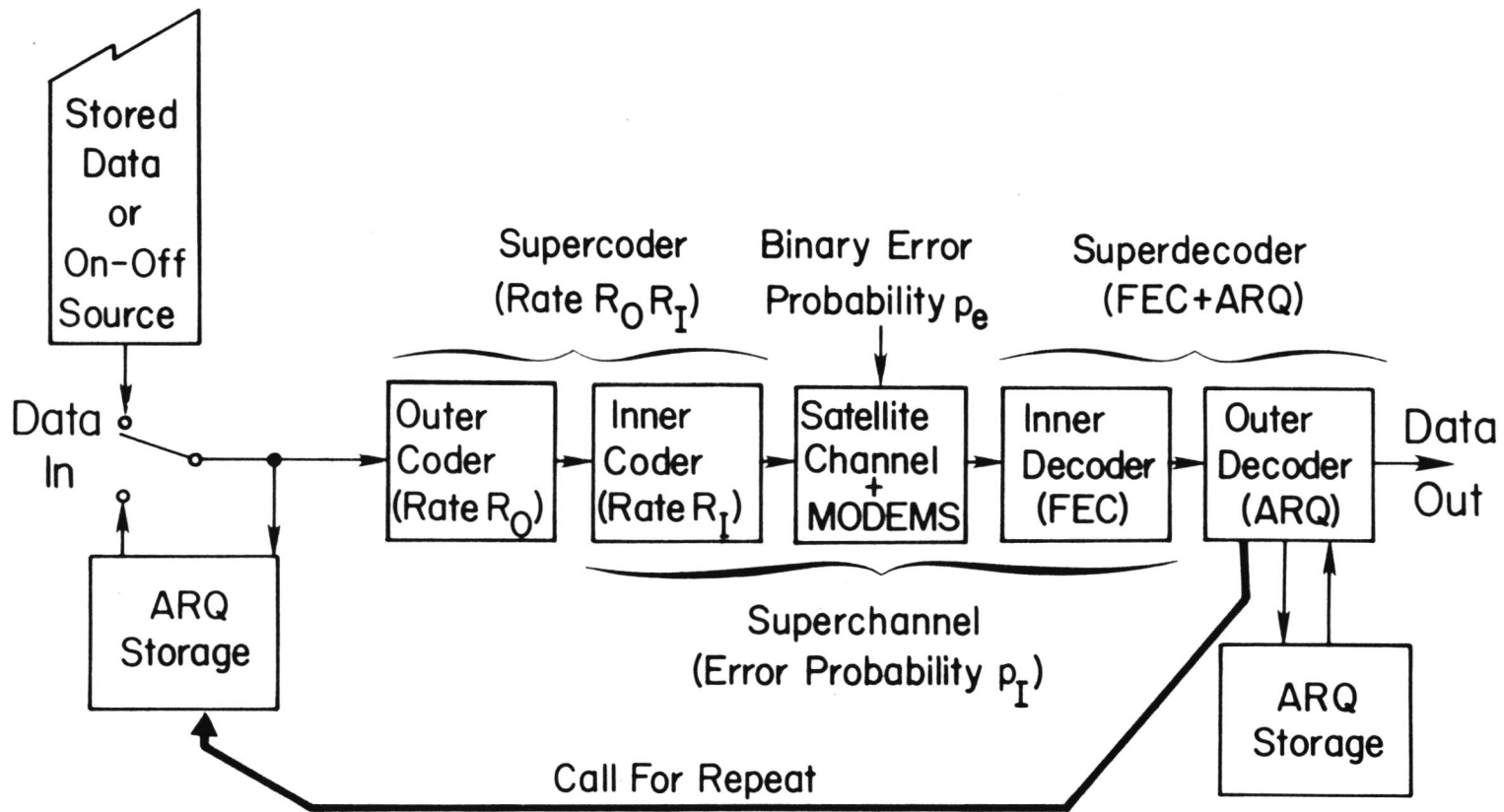


Figure 1. Hybrid FEC and ARQ scheme with concatenated coding.



- (b) Convolutional coders offer relatively simple and economical error control means.
- (c) The performance of convolutional coders is as good or better than any other FEC system. We will return to the performance specifics later, when alternative decoders will be reviewed.
- (d) The technical features of convolutional codecs are flexible. This is important for a large system where input-output requirements, cause-and-effect relations, and the consequences of even isolated design decisions are hard to predict. Some of the easily changed parameters or flexible factors offered by convolutional coding are: the code rate  $R_I$ , the constraint length  $K$  of the code, the systematic or nonsystematic code structure, the hard or soft (i.e., the number of quantization levels) operation at the demodulator output, synchronization capabilities of some decoders, and so forth.

As shown in figure 1, the convolutional inner codec consists of two parts. At the transmitter, an encoder is used to pass data from the outer encoder to the satellite channel modulator. At the receiver, a decoder processes the demodulator output before passing it to the outer decoder. Through their combined function, the inner encoder-decoder pair execute the convolutional coder operation. Typical, perhaps familiar, encoders are shown in figures 2 and 3.

Figure 2 illustrates a rate  $R_I=2/3$  code, because three output bits correspond to every two input bits. The two input information bits appear as part of the output, while a third bit checks the parity over selected past

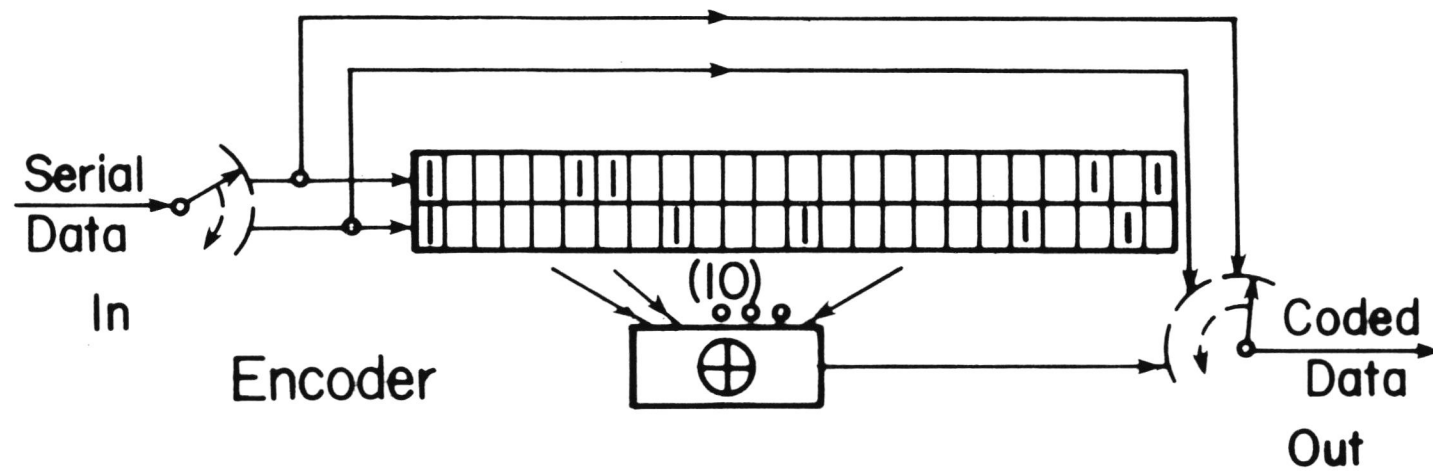
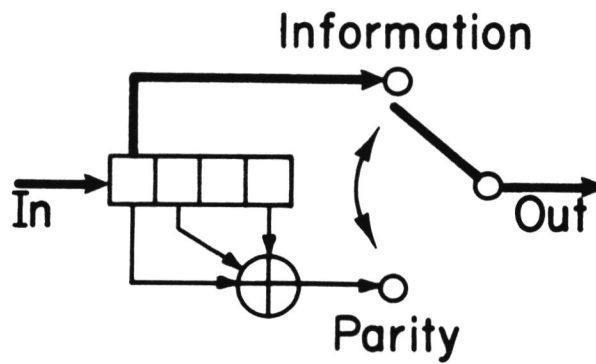
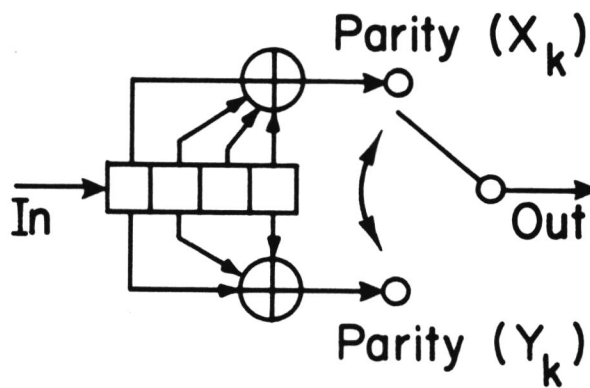


Figure 2. Convolutional encoder of rate 2/3, constraint length 48, with 10 delay taps for each parity check.





**(A) Systematic**



**(B) Non-systematic**

Figure 3. Systematic and nonsystematic versions of rate 1/2 convolutional encoder.

information bits. As shown, the parity check includes exactly ten previous bits extending as far back as 48 information bits, or 72 total bits. The former is the number of memory cells in the delay shift register shown in figure 2, and is known as the (encoding) constraint length  $K$  of the code. The code is said to be systematic, because unperturbed input bits can be found at prescribed places in the output sequence.

### 3.2. Systematic and Nonsystematic Aspects

The definition of a systematic convolutional code is further elaborated in figure 3. Both parts (A) and (B) show rate  $R_T=1/2$  and constraint length  $K=4$  convolutional encoders. Encoder (A) contains the original information bits among its outputs; it is systematic. Encoder (B) is nonsystematic, even though a mod-2 gate of outputs,  $X_k \oplus Y_k$ , produces a delayed replica of the input sequence. The reason for concern with these matters is to find the most powerful convolutional codes, and not waste time on the bad ones. It has been established (Busgang, 1965; Lin and Lyne, 1967; Heller, 1969; Massey and Costello, 1971; Gilhausen et al., 1971, 1972; Forney, 1972, 1973; Massey, 1973; Paaske, 1974) that the performance of the best nonsystematic convolutional codes is superior.

Since for block codes the two code families are often indistinguishable, some explanation seems in order. The performance of a convolutional code over any channel depends largely upon the relative distances between codewords and, in particular, upon the so called minimum free distance,  $d_f$ , of the code. This  $d_f$  is defined as the least number of 1's that can occur over all non-trivial closed paths passing through the all-zero state in the state diagram of that code



(Viterbi, 1967; Massey and Costello, 1971; Peterson and Weldon, 1972; Forney, 1973). The state diagram of example (A), figure 3, is shown in figure 4, and that of (B) in figure 5. In the case of the systematic code (A), the passage through states 000, 100, 010, 001, and back to 000, produces outputs 11, 01, 00, and 01. There are four 1's among the outputs, and no other closed non-trivial path yields less. Hence, the free distance  $d_f=4$  for the systematic code (A). Furthermore, it can be shown that no other systematic code with the same rate and constraint length can have a larger free distance. In figure 5, one traverses a different path through the states; namely, 000, 100, 110, 011, 001, and 000. The output sequence 11, 00, 01, 01, 11 implies  $d_f=6$  for this nonsystematic code. It turns out that  $d_f=6$  is the best that one can do with  $R_I=1/2$  and  $K=4$ .

### 3.3. Decoder Types

The choice of constraint length  $K$  depends on the decoder type used in practice. There are three prominent decoding techniques for convolutional codes:

Sequential decoding

Feedback decoding

Viterbi decoding

All three offer their peculiar variations, refinements, as well as advantages and disadvantages. A summary of the leading options is given in table 2. As before, the table is a review of pertinent inner code decoders and their characteristics. The justification of the claims is found in appropriate references (which includes most of the list).

Convolutional,  
 Best ( $d_f = 4$ )  
 Systematic,  
 Rate 1/2,  
 Constraint  
 Length  $K = 4$

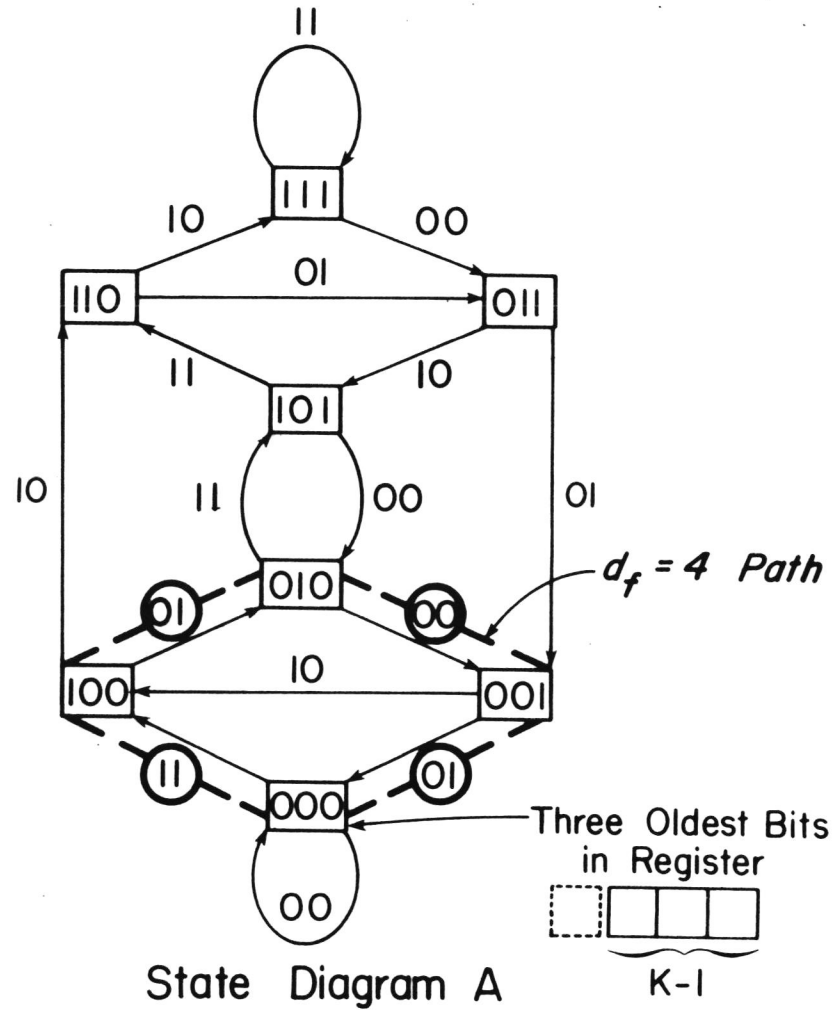
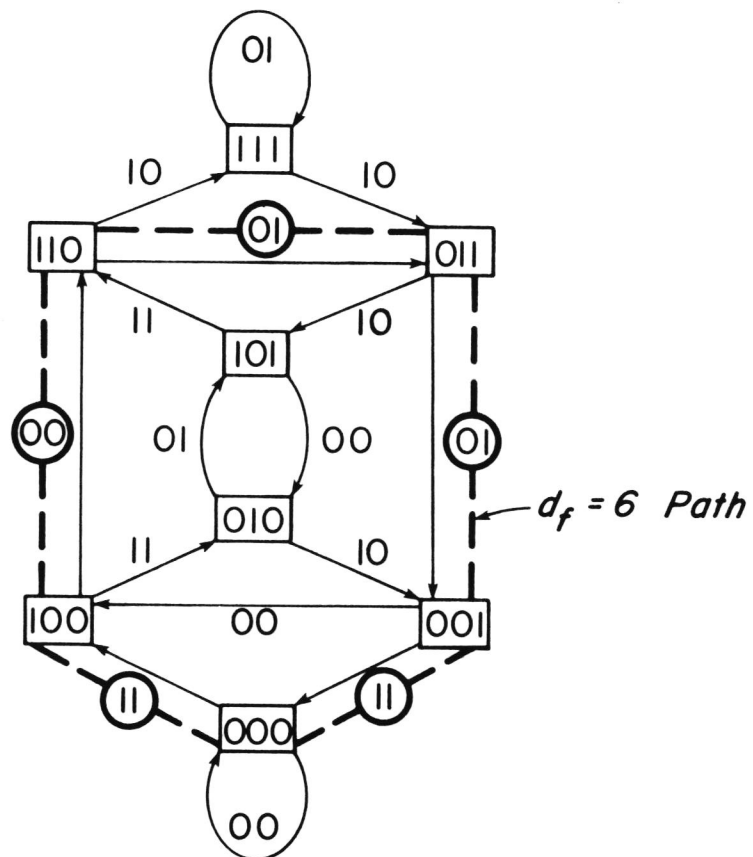


Figure 4. State diagram of the systematic code.



Convolutional,  
 Best ( $d_f = 6$ )  
 Non-systematic,  
 Rate 1/2,  
 Constraint  
 Length  $K = 4$ .



State Diagram B

Figure 5. State diagram of the nonsystematic code.

Table 2. Comparison of Forward Acting Decoders For Convolutional Inner Code Candidates. Code Rates Are Assumed in the 0.50-0.75 Range

Decoder Type	CHARACTERISTIC					Relative Rating
	Error Rate Performance	Experimental Evidence	Logic Speed Data Speed	Cost and Complexity	Others	
Sequential with Fano algorithm	Great	Some tests	2-4	Extremely high		4
Sequential with stack, bucket, and other variations	Great	Some tests	1.5-2	Quite high	Further progress likely	3
Feedback (general)	Poor		1	Low		5
Feedback (threshold or MLD)	Very poor	Some tests	1	Very low		6
Viterbi with hard decisions	Good	Well tested	2-3	Reasonable	Many good options	2
Viterbi with soft decisions	Nearly optimum and robust	Well tested	2-3	Reasonable	Many good options	1



Sequential decoding evolved from the original work of Wozencraft (Wozencraft, 1957; Wozencraft and Reiffen, 1961), to the Fano algorithm (Fano, 1963; Massey and Sain, 1968; Gallager, 1968), to later stack, bucket, bootstrap, and other variations (Zigangirov, 1966; Jelinek, 1969; Massey, 1973). All of these techniques can work with long constraint lengths and so produce powerful error correction performance. The performance has been verified by tests and/or simulation (Forney, 1967; Forney and Langelier, 1969; Lumb, 1969; Cain, 1971; Layland and Lushbaugh, 1971; Forney and Bower, 1971; Gilhausen et al., 1971; Gilhausen and Lumb, 1972; Odenwalder et al., 1972; Cahn et al., 1973; Dodds, 1973). More recent sequential decoding versions have sought to reduce the computation load, the associated logic speed and the memory requirements, while maintaining high level of performance. To a large extent, such objectives have been met by the Zigangirov (1966) and Jelinek (1969) stack method, and other recent extensions. Unfortunately, the complexity of sequential decoders is still large and burdensome.

A far simpler family of convolutional decoders are the feedback decoders, and the related threshold or majority logic decision (MLD) decoders (Massey, 1963; Rudolph, 1967; Gallager, 1968; Goldman, 1969; Peterson and Weldon, 1972; Massey, 1973; Wolf, 1973). A general feedback decoder is illustrated in figure 6. Note the crucial role of the syndrome register. Through a usually prewired or fixed read-only storage logic, the syndrome dictates output correction as well as its own update, called "feedback." A special case of the feedback decoder is the fixed threshold count-of-1's MLD decoder. Such an MLD device is shown in figure 7. This decoder happens to decode the

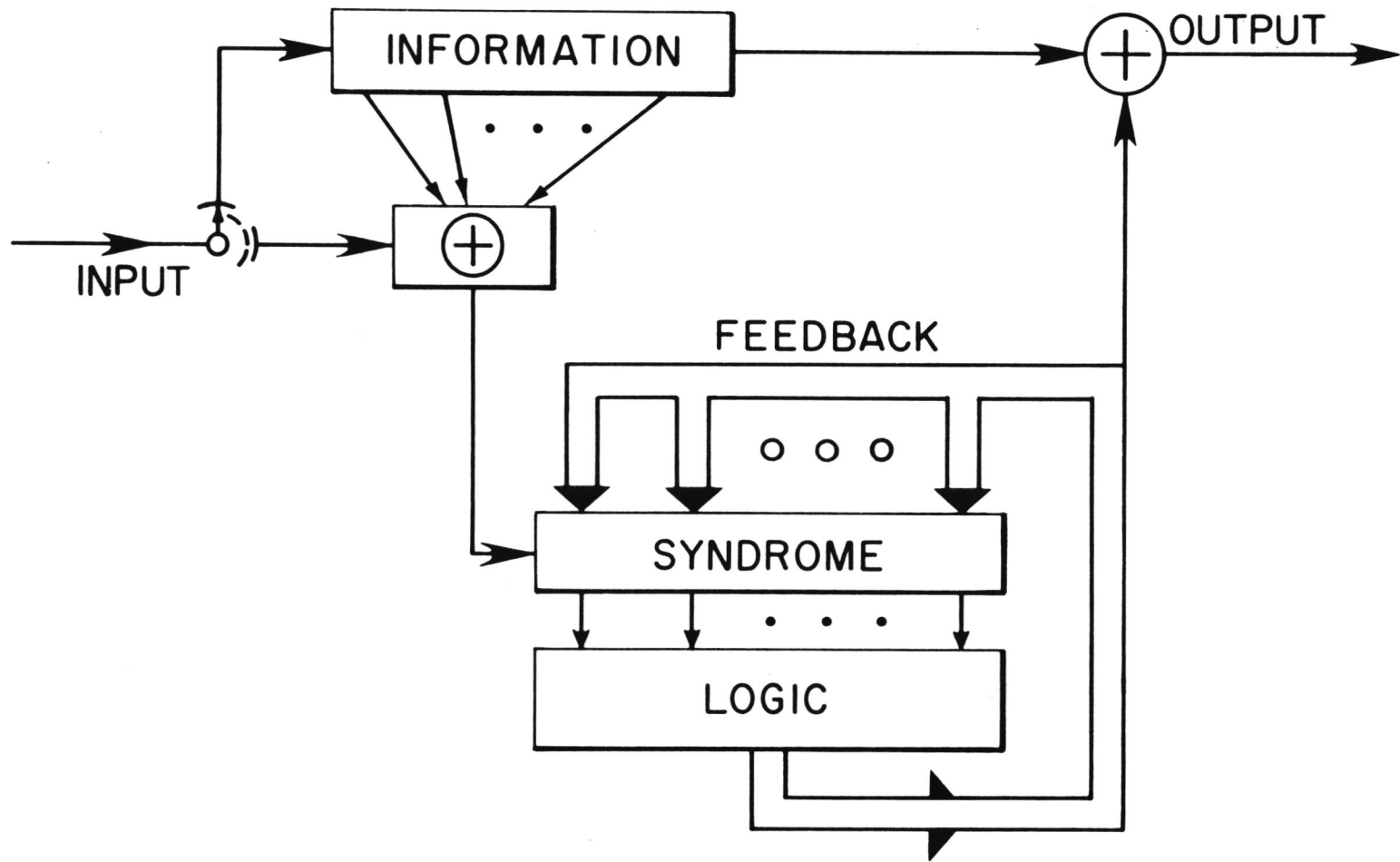


Figure 6. Basic feedback decoder for the rate 1/2 code.





rate 2/3 code introduced earlier in figure 2. The two threshold gates  $M_1$  and  $M_2$  serve the two parallel information registers. The function of the two gates is an identical majority poll of the five inputs shown. Thus, if the number of input 0's is  $N(0)$  and the number of input 1's is  $N(1)$ , then

$$\begin{aligned} \text{Gate Output} &= 0 && \text{if } N(0) > N(1), \\ &= 1 && \text{if } N(0) < N(1). \end{aligned}$$

A 1 at the gate output inverts the appropriate information bit and, by feedback action, deletes either

$$N(1) - N(0) \geq 1$$

or

$$N(1) - N(0) - 1 \geq 0$$

net 1's from the syndrome register.

The codec consisting of encoder (fig. 2) and decoder (fig. 7), implements a rate  $R_I=2/3$  self-orthogonal code. Its operation is based on, so called, "disjoint and full" difference triangles (Massey, 1963; Peterson and Weldon, 1972):

		21				19				
	16		6			11		12		
	15	1		5		7		4	8	
2	17		18		23	3	10		14	22

The free distance of this code  $d_f=6$  is nearly optimal for such self-orthogonal codes, and yet only two errors are always correctable among 72 channel bits (Robinson and Bernstein, 1967). This error correcting scheme is not very powerful (it further worsens as constraint length increases), but the codec cost and complexity is extremely low.

### 3.4. Viterbi Decoding

The preferred alternative in table 2 uses the Viterbi decoding algorithm (Viterbi, 1967, 1971; Heller, 1968,



1969; Omura, 1969; 1971; Clark and Davis, 1971; Clark, 1971; Heller and Jacobs, 1971; Kobayashi, 1971; Viterbi and Odenwalder, 1972; Batson et al., 1972; Forney, 1972; 1973; Massey, 1973; Wolf, 1973; Jacobs, 1974). One must distinguish two cases. The "hard decision" case refers to demodulator output and decoder input being an ordinary binary decision, zero or one. The "soft decision" admits more than two quantized demodulator outputs. A typical case of eight quantization levels is:

```

000 ----- a sure zero,
001
010
011 ----- a doubtful zero,
100 ----- a doubtful one,
101
110
111 ----- a sure one.

```

There are Viterbi decoders with an incorporated switch to either use or omit the less significant quantization digits. Thus, an operator can choose between the hard and soft options. Both options have been explored and tested. While the speed requirements and costs are comparable, one finds the soft version superior in performance.

Before turning to the performance numbers of the various alternatives, it may be beneficial to get an elementary grasp of the operation and implementation of our first choice - the Viterbi decoding system. A simple illustration can start with the encoder of figure 3, part (B). Here, the (encoder) constraint length is  $K=4$ , roughly a half of the  $K=6$  to  $K=10$  values used in actual systems. The rate  $R_I=1/2$  code is nonsystematic, with the state diagram given in figure 5. Each input bit directs the system from a

previous state to the next state. If the present state is 110, then input 0 causes a transition to state 011 and input 1 to state 111. The output bits produced by register content 0110 are 01, and so forth. These output bits are shown in the state diagram (fig. 5) next to the transition arrows between states.

The trellis diagram is an equivalent, somewhat longer, and, in a way, a more vivid description of the encoding process. The corresponding trellis diagram is shown in figure 8. Note again how a single input bit 0 takes the encoder from state 110 to state 011 and produces output 01. A length L non-zero input sequence gives forth a  $2(L+3)$  long output. An example of  $L=5$ ,

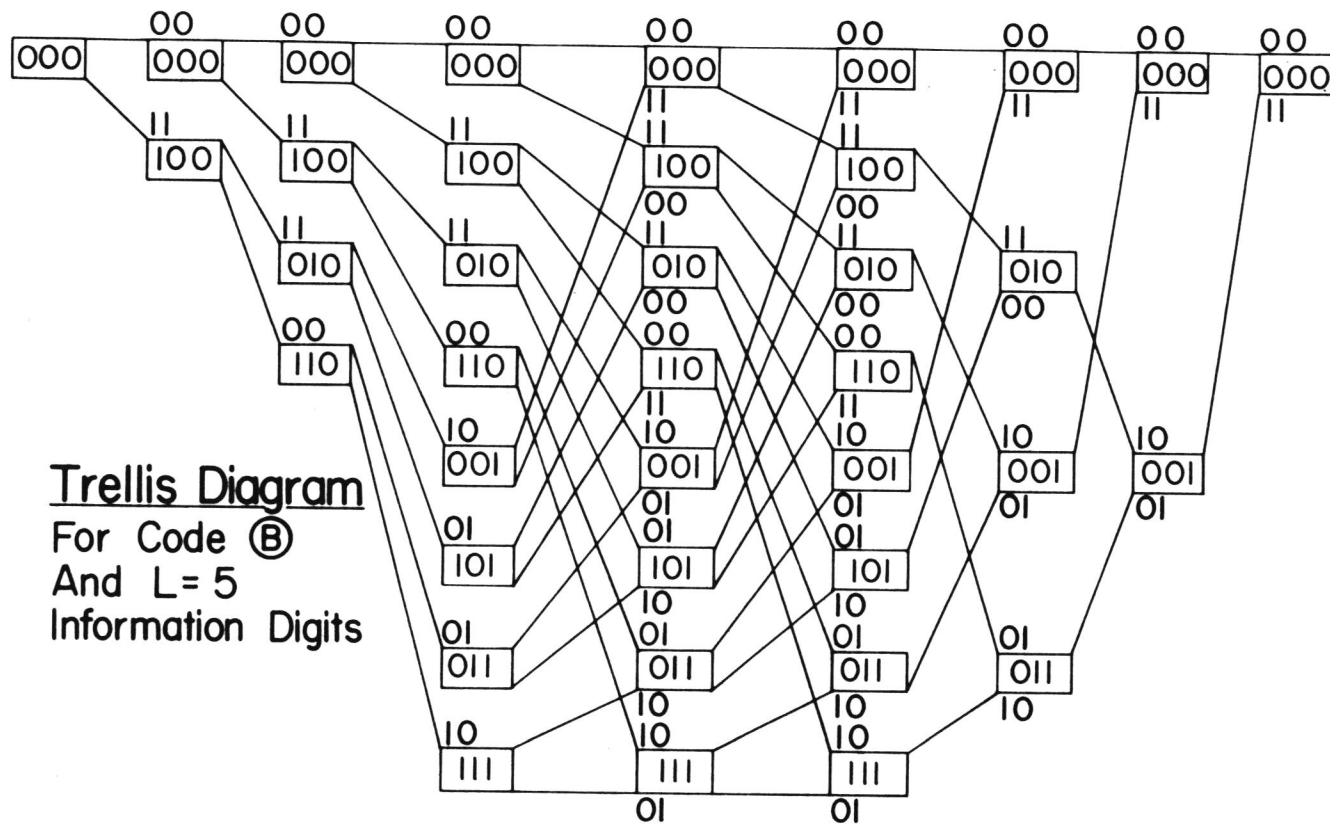
$$\dots 0 \underset{\substack{\longleftarrow \\ L \\ \longrightarrow}}{1 \ 1 \ 0 \ 1 \ 1} 0 \dots$$

yields an output

$$\dots 00 \underset{\substack{\longleftarrow \\ 2(L+3) \\ \longrightarrow}}{11 \ 00 \ 01 \ 10 \ 11 \ 01 \ 01 \ 11} 00 \dots$$

The extension of L to L+3 describes the merge properties of a non-zero path from the all-zero path in the trellis. It is also an important delay parameter to be used later in the decoding process. As explained elsewhere, the Viterbi algorithm ascertains and, if need be, modifies the estimated most likely path through the trellis. With a diminishing probability, some path updates can take place after considerable delay. In communication application, outputs cannot be delayed arbitrarily long, but must be delivered after some fixed -- preferably small -- delay. It turns out that, through a process called merging of paths, the finite delay parameter L serves to terminate the decoder path memory, and to deliver the output bits on schedule. The effect of various L choices has been





Trellis Diagram  
 For Code (B)  
 And L=5  
 Information Digits

Encoder Outputs.

Example: When data bit 0 changes state  
 110 to 011, the output is "01".

Figure 8. Trellis diagram and encoder outputs.

determined by Gilhausen et al. (1971). It is shown that  $L \approx 4K$ , which is a delay four times the constraint length, for all practical purposes gives as much performance gain as can be expected of the arbitrarily long memory. In the previous example of figure 3 (B) and figure 8, the  $L=5$  value is thus too short for best results, and entails at least a 1 dB signal-to-noise ratio loss. A higher parameter value, such as  $L \geq 16$ , should be used to avoid the loss.

The decoder selects the most likely path by continuously keeping track of appropriate path likelihood measures, commonly called metrics. The multitude of paths are all constituted of branches, that is, directed links between successive states. When a set of branches combine to form a path, the branch metrics add to form the (total) path metric of that path. As an illustration, consider the hard decision example of figure 8. The bookkeeping of branch and path metrics is shown in figure 9. If the original input data sequence is

0 0 1 1 1 0 0 0,

the encoded and transmitted sequence is

00 00 11 00 10 10 01 11.

For a moment, assume no errors in transmission, so that the received sequence agrees with the transmitted one.

A metric that is defined to count disagreements will assign to the all 0's path a sequence of branch metric/path metric pairs

0/0 0/0 2/2 0/2 1/3 1/4 1/5 2/7,

with a final path metric 7. Likewise, the all ( $L=5$ ) 1's path would be encoded as

11 00 10 01 01 10 01 11,

and the metric pairs would evolve as

2/2 0/2 1/3 1/4 2/6 0/6 0/6 0/6,



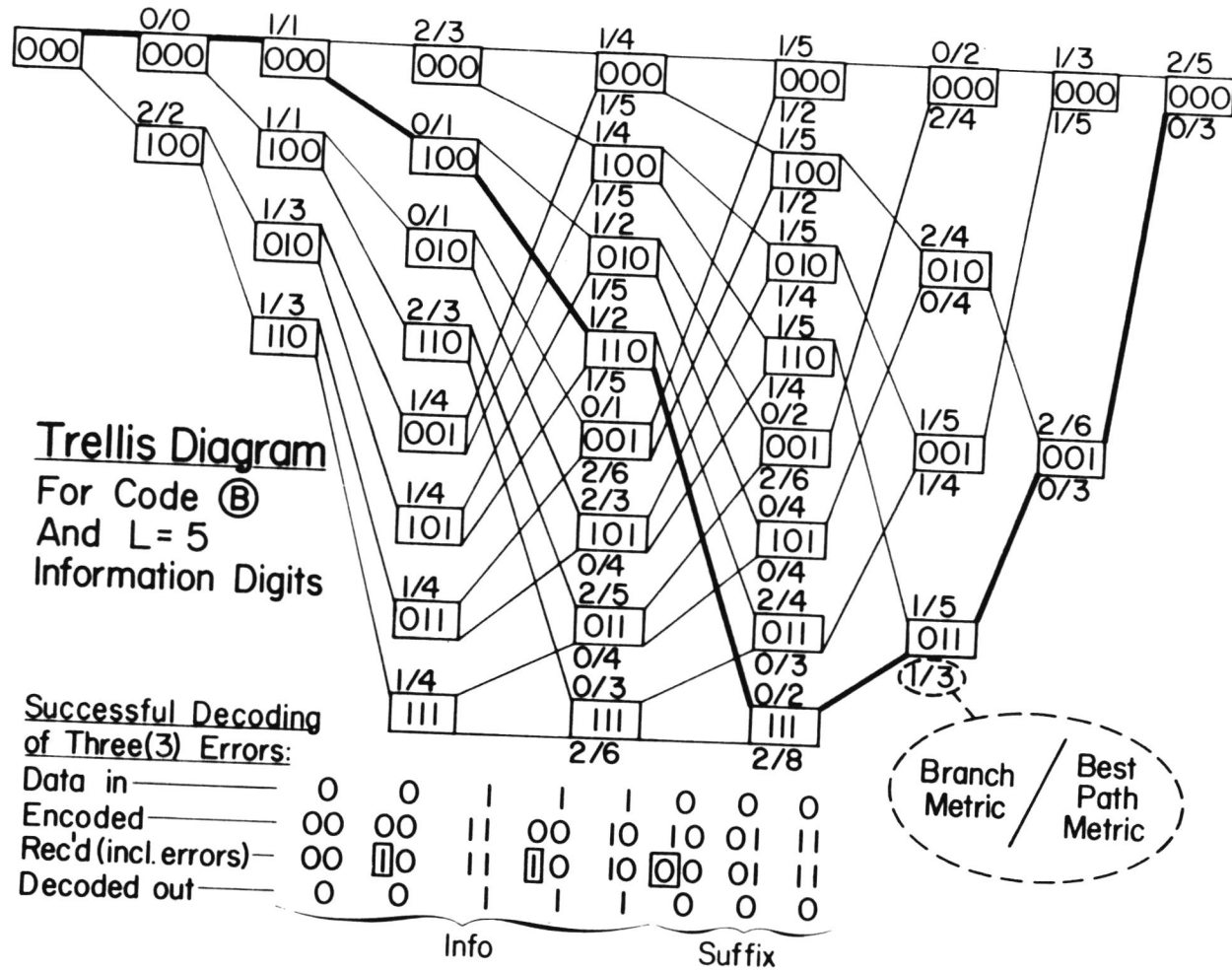


Figure 9. The role of branch metric/best path metric pairs in successful decoding of three errors.

causing the final metric 6. In fact, all conceivable paths will have nonzero metrics, except for the correct path, where only 0/0 metric pairs arise. Note, that the 3-bit suffix does not carry information. Its sole purpose is to merge the paths of the trellis.

Next, permit three errors to infest the same transmitted sequence, as shown by the received sequence

00 10 11 10 10 00 01 11.

From figure 9, one sees how the branch and path metric picture changes. While the all 0's path shows a total metric 8, the best final-zero path now has a metric of 5. Likewise, the all 1's path, with a total metric 9, is worse than the best final-one path. The latter has a path metric of 3, and through back tracing it leads to the actual correct path. Thus, the correct path is unique, it is identifiable, and the correct decoding output is delivered.

Of course, there are other error patterns that cannot be corrected. In figure 10 we show a five error event that causes erroneous output. An incorrect path with a total path metric 2 is selected ahead of the correct path, which turns out to have a path metric 5. As a consequence, the original data sequence

0 1 1 0      1 0 0 0

is incorrectly decoded as

0 1 0 0      1 0 0 0.

This is not too unfortunate, as the five original channel errors are reduced to a single output error. For longer delay parameters L and other error events, one anticipates output errors to occur in bursts with durations typically of the same order as the constraint length K of the code.



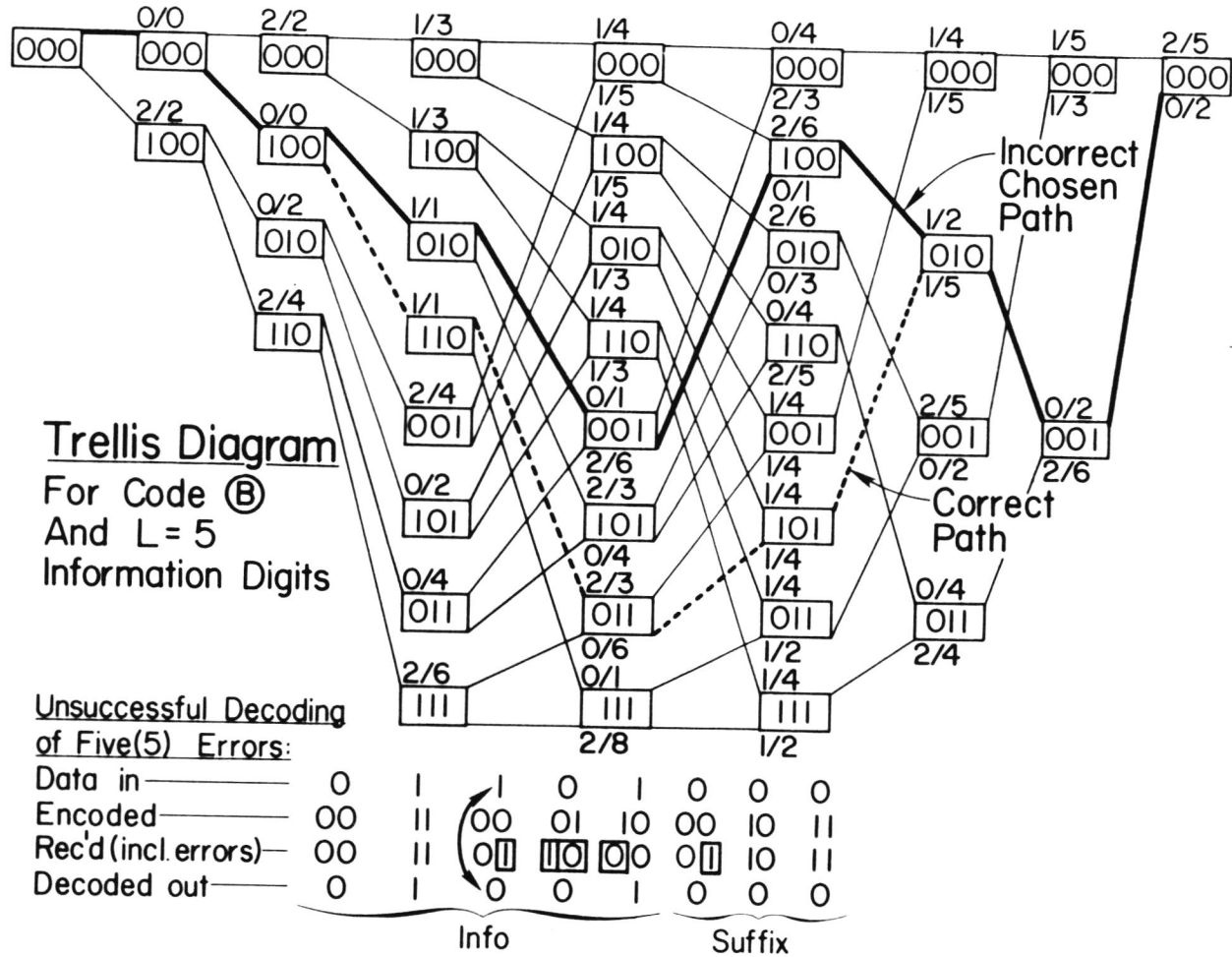


Figure 10. Unsuccessful decoding of five errors.

### 3.5. Performance on the Ideal Channel

Simulated ideal channel performance of convolutional FEC codes is summarized in figure 11. The ordinate is the output bit-error probability,  $p_I$ , and the abscissa is the normalized signal-to-noise ratio,  $E_b/N_0$  in decibels. Quantity  $E_b$  is the signal energy per information bit, and  $N_0$  is the one-sided noise spectral density. The curves of figure 11 depict the binary (i.e.,  $M=2$ ) coherent PSK channel in the presence of white Gaussian noise. As discussed before (McManamon et al., 1974), the main features and results readily generalize to the other MPSK modems, as well as other channel models.

The four vertical lines in figure 11 are designated as channel capacity limits. For instance, with no restriction on signal shape or duration, Shannon's ultimate theorem for error-free reception

$$R_I = W \log_2 \left( 1 + \frac{S}{N} \right)$$

may be contemplated. If one divides by rate  $R_I$  and lets bandwidth  $W$  assume larger and larger values, then

$$\ln 2 = \ln \left[ \left( 1 + \frac{R_I E_b}{W N_0} \right)^{\frac{W}{R_I}} \right] \xrightarrow{W \rightarrow \infty} \frac{E_b}{N_0} .$$

If so, the ultimate zero-error operation cannot be achieved unless  $E_b/N_0$  is larger than  $\ln 2$ , or -1.6 dB. If one insists on hard decision operation, the capacity threshold is multiplied by a factor  $\pi/2$ , or 2.0 dB. To compensate,  $E_b/N_0$  must then be above 0.4 dB. Likewise, if one permits arbitrarily fine soft decisions to accompany optimal sequential decoding, then  $E_b/N_0=1.4$  dB is needed. And, if one restricts the bandwidth to  $R_I W=1/2$  and specifies



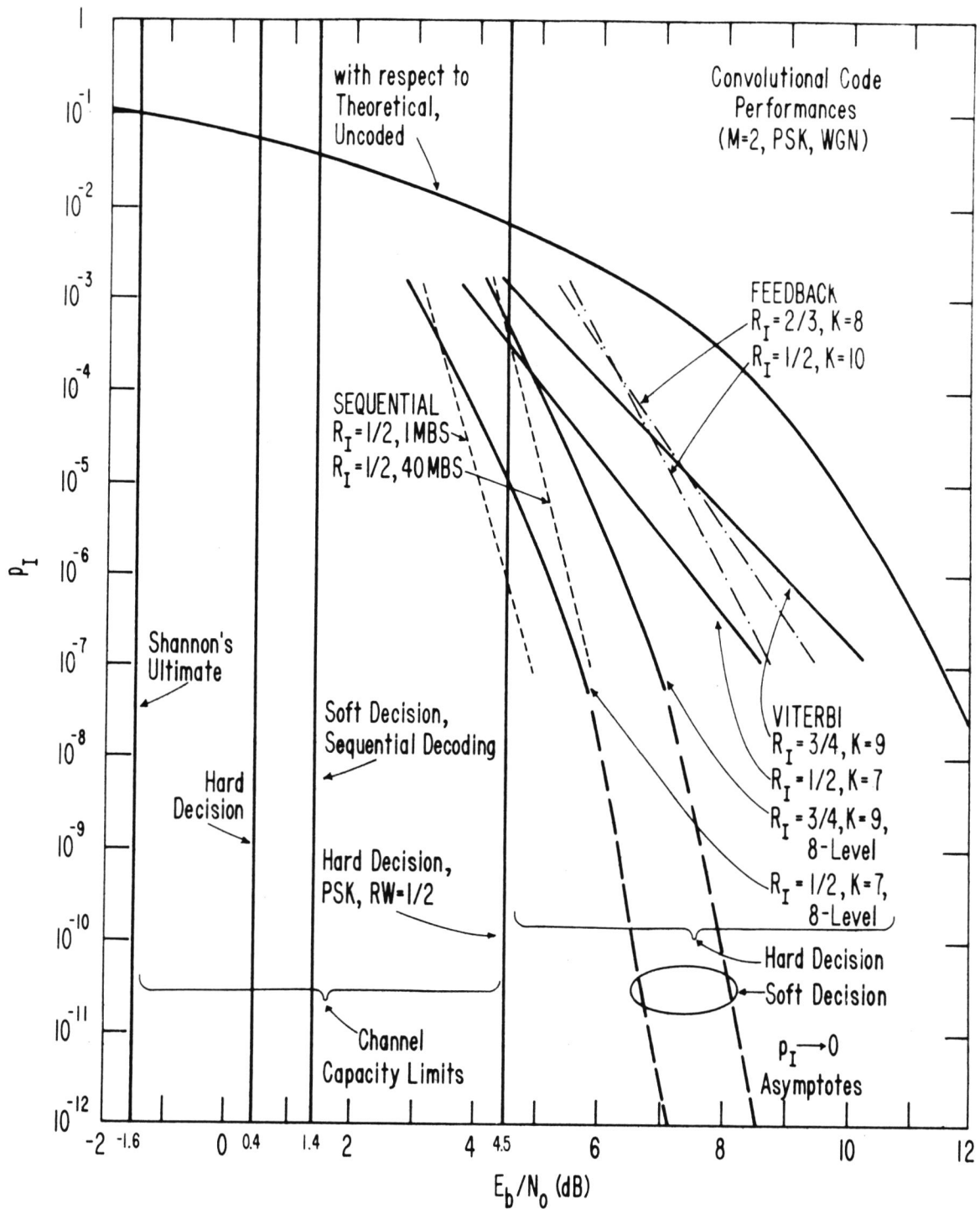


Figure 11. Various convolutional code performances on the ideal theoretical channel.

PSK hard decision operation, then more than 3 dB is sacrificed and the rightmost threshold  $E_b/N_0=4.5$  dB is required.

Without coding, the theoretical modem performance is a smooth function of  $E_b/N_0$ . The indicated curve crosses all the capacity limits and upper-bounds all the observed data in figure 11. The present observed coding data extends over the  $p_I$  range from  $10^{-3}$  to  $10^{-7}$  (Forney, 1970; Forney and Bower, 1971; Gilhousen et al., 1971; Batson et al., 1972; Odenwalder et al., 1972; Cahn et al., 1973; Jacobs, 1974). Three types of convolutional codecs are indicated. They are:

Sequential decoding, rate 1/2, and two different data throughput rates, 1 Mbps and 40 Mbps.

Hard decision.

Feedback decoding with rates 1/2 and 2/3, and constraint lengths 10 and 8, respectively.

Hard decision.

Viterbi decoding with rates 1/2 and 3/4, and constraint lengths 7 and 9, respectively.

Hard and soft decision.

In support of earlier table 2, figure 11 shows the mentioned observed performances. Error correction per se improves with lower coding rate  $R_I$  and higher constraint length  $K$ . At  $p_I=10^{-7}$ , for example, the rate  $R_I=1/2$  feedback decoding scheme requires roughly  $E_b/N_0=9$  dB. Viterbi decoding without quantization appears only slightly better, while hard decision sequential decoding offers a 3 dB improvement at the  $p_I=10^{-7}$  level. There is some uncertainty about the actual gain, as the complexity and speed of logic affects implementation and performance for sequential machines. Such is the nature of the 1 dB



advantage seen at the 1 Mbps data rate tests over the 40 Mbps tests.

Fortunately, the performance of Viterbi decoding can be substantially improved by the previously cited soft decision operation. In figure 11 two such 8-level quantization curves are shown. They depict code rates  $R_I=1/2$  and  $R_I=3/4$ . At  $p_I=10^{-7}$ , the rate 1/2 code is comparable to the sequential hard decision case, and requires a mere  $E_b/N_0 \geq 6$  dB to perform that well. While all this seems remarkably good, one is reminded of the interest in the objective of an overall  $10^{-12}$  error probability. The performance extrapolation beyond observed data, toward lower  $p_I$  values, is indicated in figure 11. The asymptotic  $p_I \rightarrow 0$  behavior is derived from the code distance structure, more specifically from the free distance  $d_f$  introduced above (Jacobs, 1974). Note, that  $E_b/N_0=7.3$  dB seems to provide  $p_I=10^{-12}$  for soft decision, rate  $R_I=1/2$ , Viterbi decoder. This signal-to-noise ratio should be contrasted with  $E_b/N_0=14.4$  that is required by uncoded ideal theoretical operation at the same  $p_I$  level, and the considerably larger extrapolated-measured requirement of 20.5 dB, or even worse (McManamon et al., 1974).

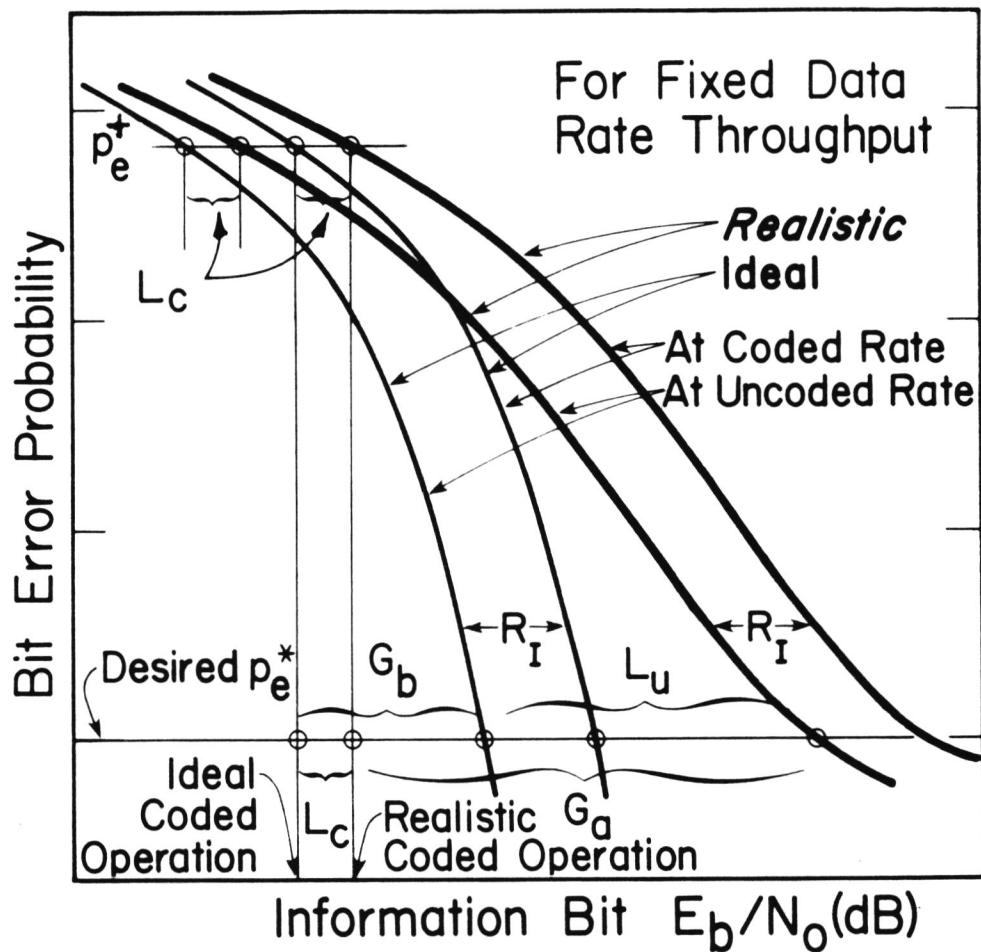
### 3.6. Performance on Realistic Channels

Actual channels, even on satellite links, are known to depart from the ideal white Gaussian noise model. There are many reasons why modems fall short of optimal performance and suffer a variety of operational losses. As a consequence, the previous ideal performance estimates must be modified. We will do so next, with particular attention to the two  $R_I=1/2$  and  $R_I=3/4$  Viterbi soft decision decoders described earlier and highlighted in figure 11. The

performance at other rates, such as  $R_I=2/3$ , can be interpolated as needed.

The definition of gain and loss terms is given in figure 12. First, the code redundancy carries a loss, the so called code rate loss, that equals 3.0 dB for the  $R_I=1/2$  code, and 1.2 dB for the  $R_I=3/4$  code. This loss is shown in figure 12 as  $R_I$  (dB), a constant displacement of the uncoded modem performance curve for both ideal and realistic channels. Assume next that error statistics are as simple as possible, such as for a memoryless binary symmetric channel, where errors occur with probability  $p_e$ . If the error probability has a given value,  $p_e=p_e^+$ , a code with rate  $R_I$  may be able to reduce  $p_e$  to a new level,  $p_e^*$ , which could be denoted  $p_I^*$  in deference to the inner coder. For the ideal channel, such as given in figure 11, the basic coding gain,  $G_b$  (dB), is defined as the net theoretical saving of signal-to-noise ratio  $E_b/N_0$  (dB) through the use of the code. In figure 12, the coded modem curve intersects  $p_e^+$  at the ideal coded operation point. This  $E_b/N_0$  value had better be less than the  $p_e^*$  intersect with the uncoded curve, for the basic coding gain  $G_b$  to be positive.

The actual coding gain,  $G_a$  (dB), is defined in a similar way. Note that the actual modem-channel pair must expend more signal energy to achieve the same performance,  $p_e^+$  or  $p_e^*$ , as the ideal. Furthermore, at specified  $p_e$  levels these degradations can be different. One has different modem losses at different performance levels. Thus in figure 12, the modem loss at  $p_e^+$  is shown to be  $L_c$  (dB), and at  $p_e^*$  it is  $L_u$  (dB). Typically,  $L_u > L_c$  implies that realistic coded system requires more  $E_b/N_0$  than does the corresponding ideal coded system, but not much more. One defines, therefore, the actual coding gain as the net



Terms:

$G_a$  (dB) = Actual Coding Gain

$G_b$  (dB) = Basic Coding Gain

$R_I$  (dB) =  $-10 \log_{10} R_I$  (Code Rate Loss)

$L_u$  (dB) = Modem Loss, at  $p_e^*$  without Coding

$L_c$  (dB) = Modem Loss, at  $p_e^*$  with Coding

Result:  $G_a = G_b + L_u - L_c$

Figure 12. Definition of coding gains and modem losses.



$E_b/N_0$  (dB) saving under the actual channel conditions. One sees from figure 12 that

$$G_a = G_b + L_u - L_c \quad (\text{dB})$$

must hold, and that  $G_a > G_b$  is apt to be true. The last assertion has validity for those real life  $p_e$  vs.  $E_b/N_0$  plots that flatten out and display an irreducible error rate. Then there may exist  $p_e^*$  levels at which the actual coding gain is infinite.

Recent data on applicable coding gains is summarized by Jacobs (1974), and the references listed. The gains can be extended to our ideal and extrapolated-measured curves, under all the specified conditions. For both  $M=2$  and  $M=4$  coherent PSK, rate  $R=1/2$ , constraint length  $K=7$ , eight level soft decision, a Viterbi decoder should perform as indicated in figure 13. In the center of the figure, a desired  $p_e^*=10^{-7}$  level is identified. At that level,

$$L_u \cong 3.5 \text{ dB},$$

$$L_c \cong 1.8 \text{ dB},$$

$$G_a \cong 7.5 \text{ dB},$$

$$G_b \cong 5.8 \text{ dB}.$$

The actual coding gains have been computed for various desired  $p_e^*$  levels. The results are given in figures 14 and 15 for the same two Viterbi arrangements cited earlier. Note the more or less invariant 1 dB  $E_b/N_0$  difference between the rate 1/2 and rate 3/4 codes. If the uncoded extrapolated-measured data calls for 20.5 dB signal-to-noise ratio at the  $10^{-12}$  level, then the present FEC scheme seems capable of the same reliability at 10.3 dB and 11.3 dB, for the two rates considered. This actual coding gain of 9 to 10 dB appears considerable, yet it is not the end of the story. More decibels can be conserved by the outer code, which is the topic of the next section.

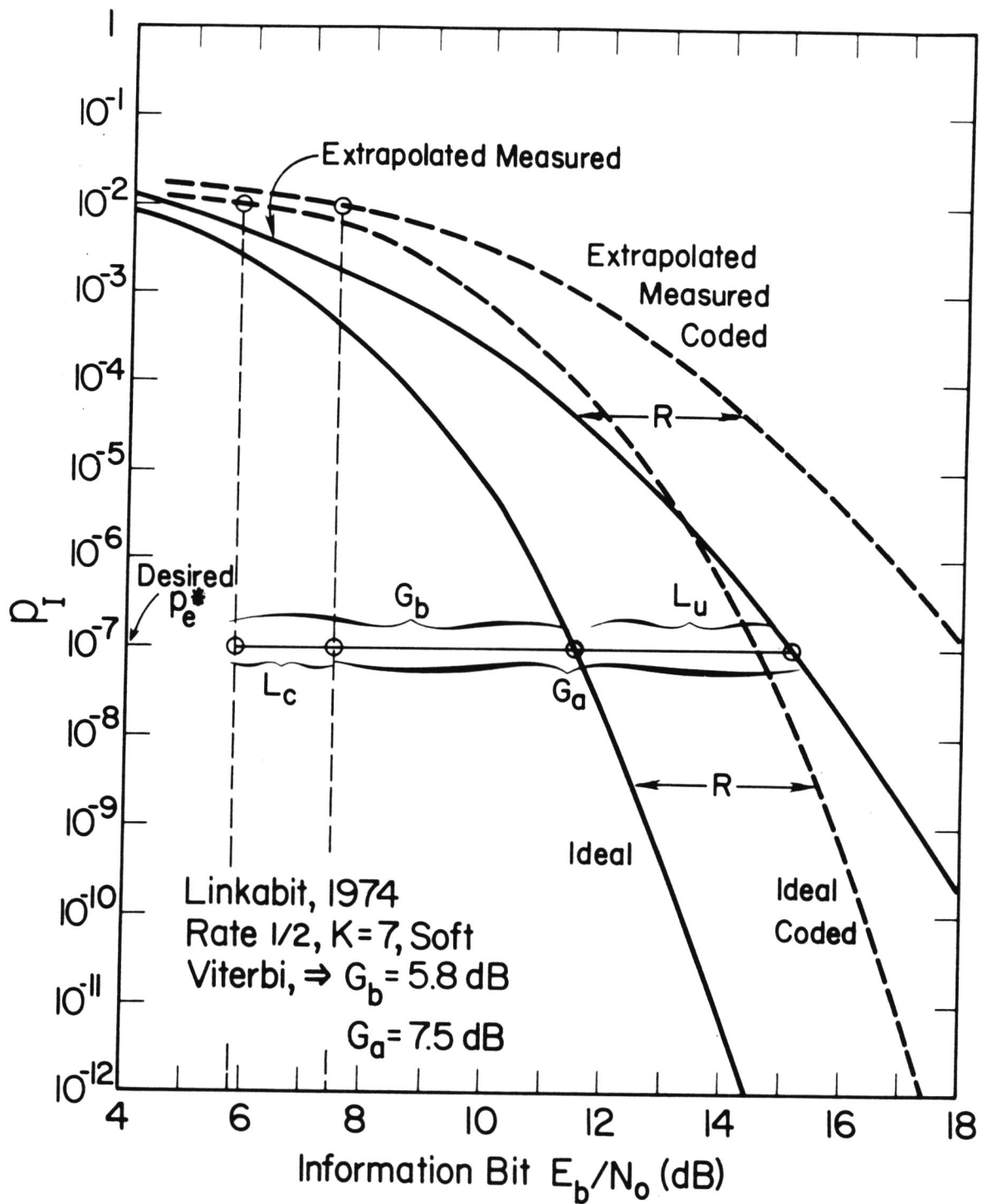


Figure 13. Viterbi decoding gains on the ideal and extrapolated-measured channels.

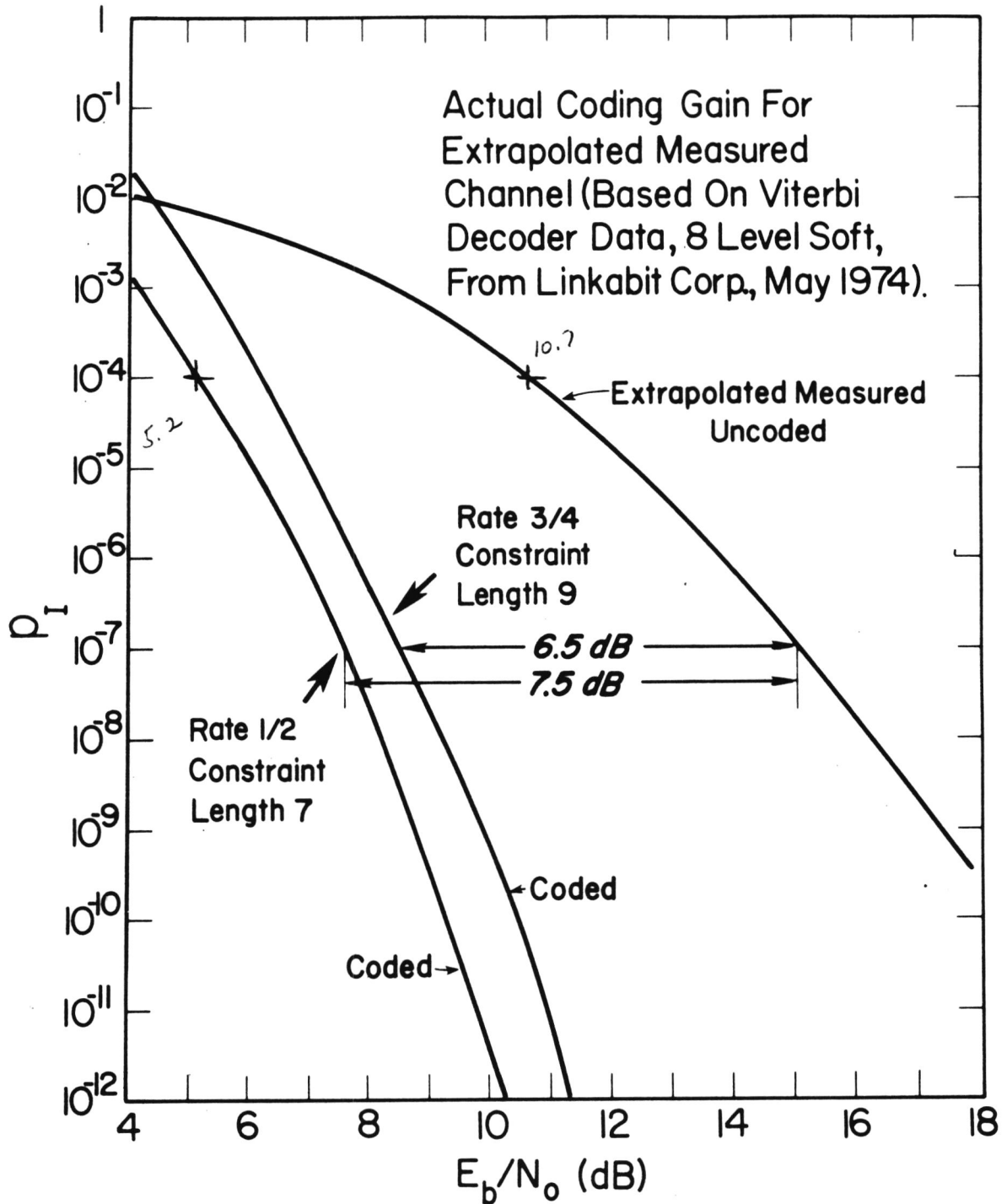


Figure 14. Actual coding gains for rate 1/2 and 3/4, soft decision, Viterbi decoders.



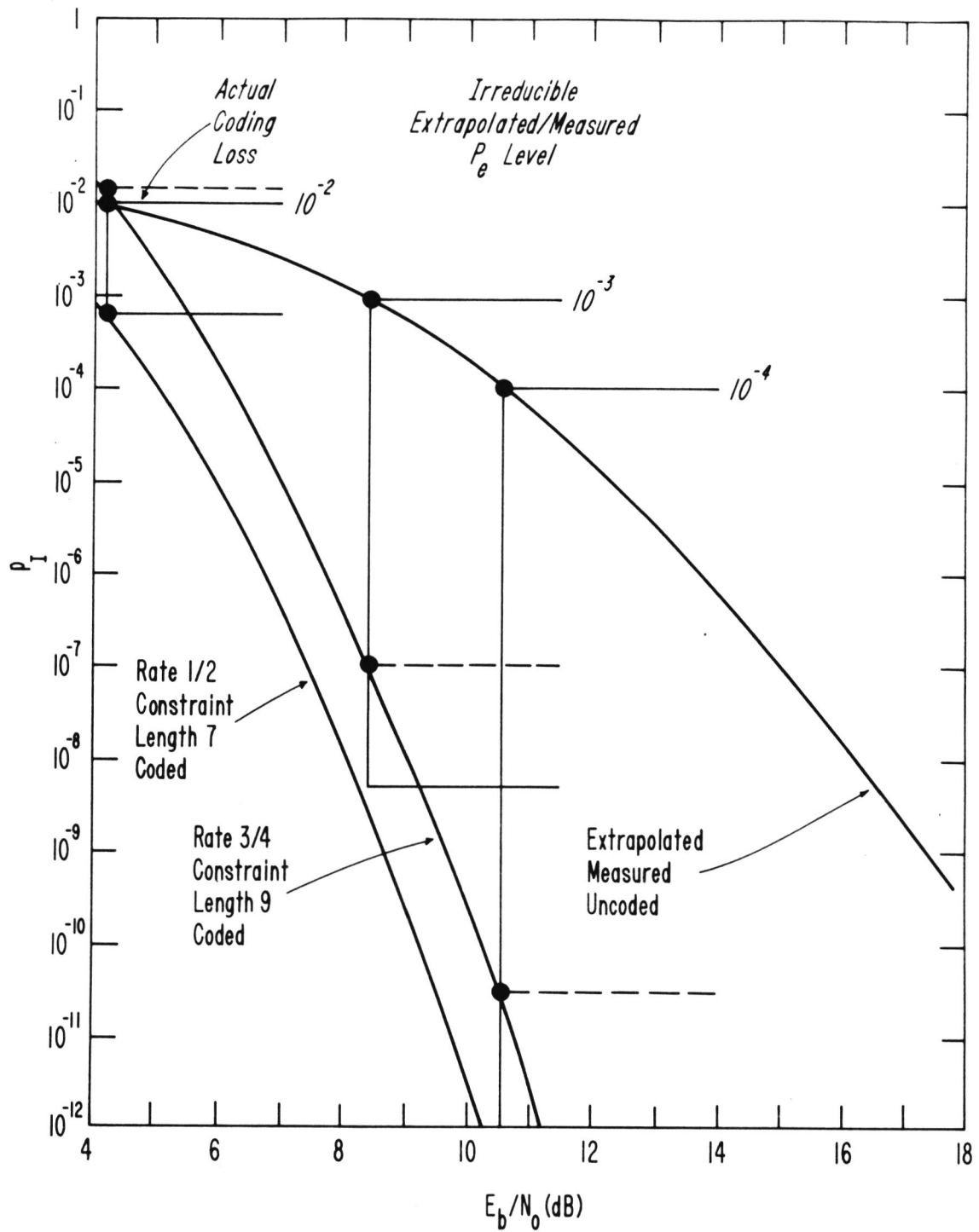


Figure 15. The effects of irreducible error probability.

Figure 15 also emphasizes that coding can be effective to reduce so called irreducible error rate levels. The effect is demonstrated by the horizontal lines to the right in figure 15. If, for example, the extrapolated-measured curve bottoms out at  $p_I=10^{-3}$  due to whatever malfunction, the output error probability after decoding will be roughly  $10^{-8}$  and  $10^{-7}$  for the codes with rates 1/2 and 3/4, respectively. An irreducible level at  $10^{-4}$  can be brought down to  $10^{-13}$  and  $10^{-11}$ . All irreducible levels lower than  $5(10^{-5})$  can satisfy the  $10^{-12}$  objective by the FEC action alone. The fact that the outer ARQ coding can help even more will be described next.

#### 4. OUTER CODE FOR THE CONCATENATED HYBRID

##### 4.1. Codec

The selection of a hybrid FEC and ARQ scheme was discussed in section 2. Moreover, table 1 and pertaining arguments led to the announcement that a concatenation of an inner convolutional code with an outer block code has the most promise of all the alternatives considered. The overall system was briefly sketched in figure 1, identifying the outer code with the ARQ function. That outer code will be the topic of this section. Matters such as the encoder, the decoder, and the role and the performance of the error detecting outer code will be elaborated.

To begin, the outer code selection is relatively straight forward. Whatever the length,  $n$ , of the block code, one uses an almost negligible number of parity check bits for error detection. The abbreviation  $(n,k)$  is used for such a block code, where  $k$  is the number of information bits and  $n-k$  the

number of check bits. It seems reasonable to start with block length  $n$  somewhere between 10,000 and 1,000,000. Redundancy of 1 percent [i.e., the ratio  $(n-k)/n$  expressed in percent] then provides 100 to 10,000 check bits. As will be shown later, this number of parity checks may be superfluous and a much smaller percent redundancy may suffice. Conclusion: the redundancy loss for the outer code of rate  $R_0$  is negligible,  $R_0 \approx 1$ .

Past experience with ARQ systems (van Duuren, 1961; Nesenbergs, 1963; Townsend and Watts, 1964; Bernice and Frey, 1964; Burton, 1970) has shown that the block length  $n$  affects primarily the throughput rate  $R$ . Roughly, throughput vanishes in an ARQ system whenever the input error probability exceeds  $1/n$ . Hence,  $n$  larger than  $10^6$  appears undesirable. Similarly, the number of check bits  $n-k$  determines the likelihood of the  $n$ -bit word containing undetected errors.

The outer encoder need not be more involved than a standard linear feedback shift-register (Gallager, 1968; Peterson and Weldon, 1972). The specific array of feedback taps can be chosen to guarantee as high a minimum distance as is possible for BCH or other known block codes. An encoder so derived is shown in figure 16. Used with  $k$  information bits,  $k$  not exceeding 32,737, the  $n-k=30$  added parity bits ensure that all four or less inner decoder output errors are detected. When the number of errors is five or more, the probability of detecting the errors is still exceedingly high.

The outer decoder, figure 17, is almost a duplicate of the outer encoder. The device simply verifies whether all  $n-k$  parity checks are satisfied by the received code word. If not, the repeat of the word must be requested from the



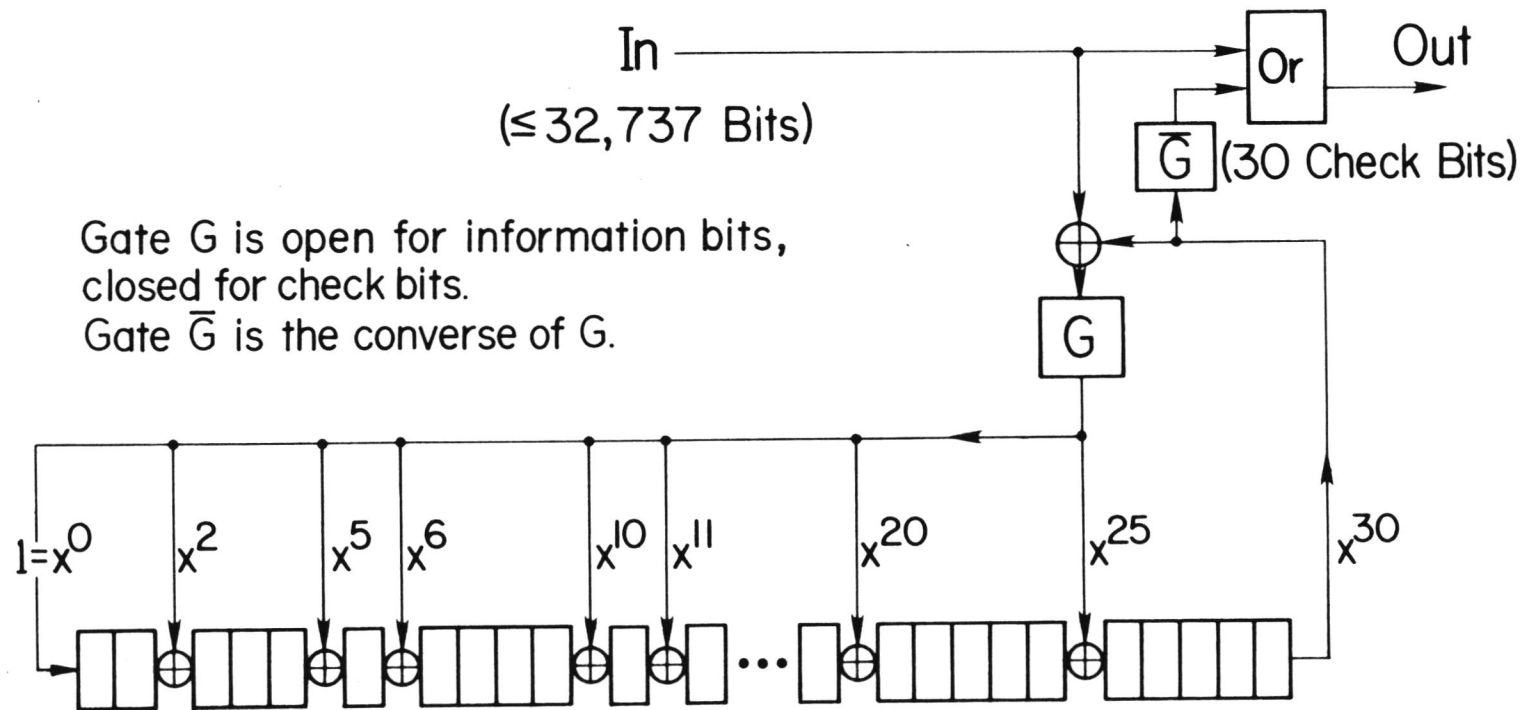


Figure 16. Outer encoder.

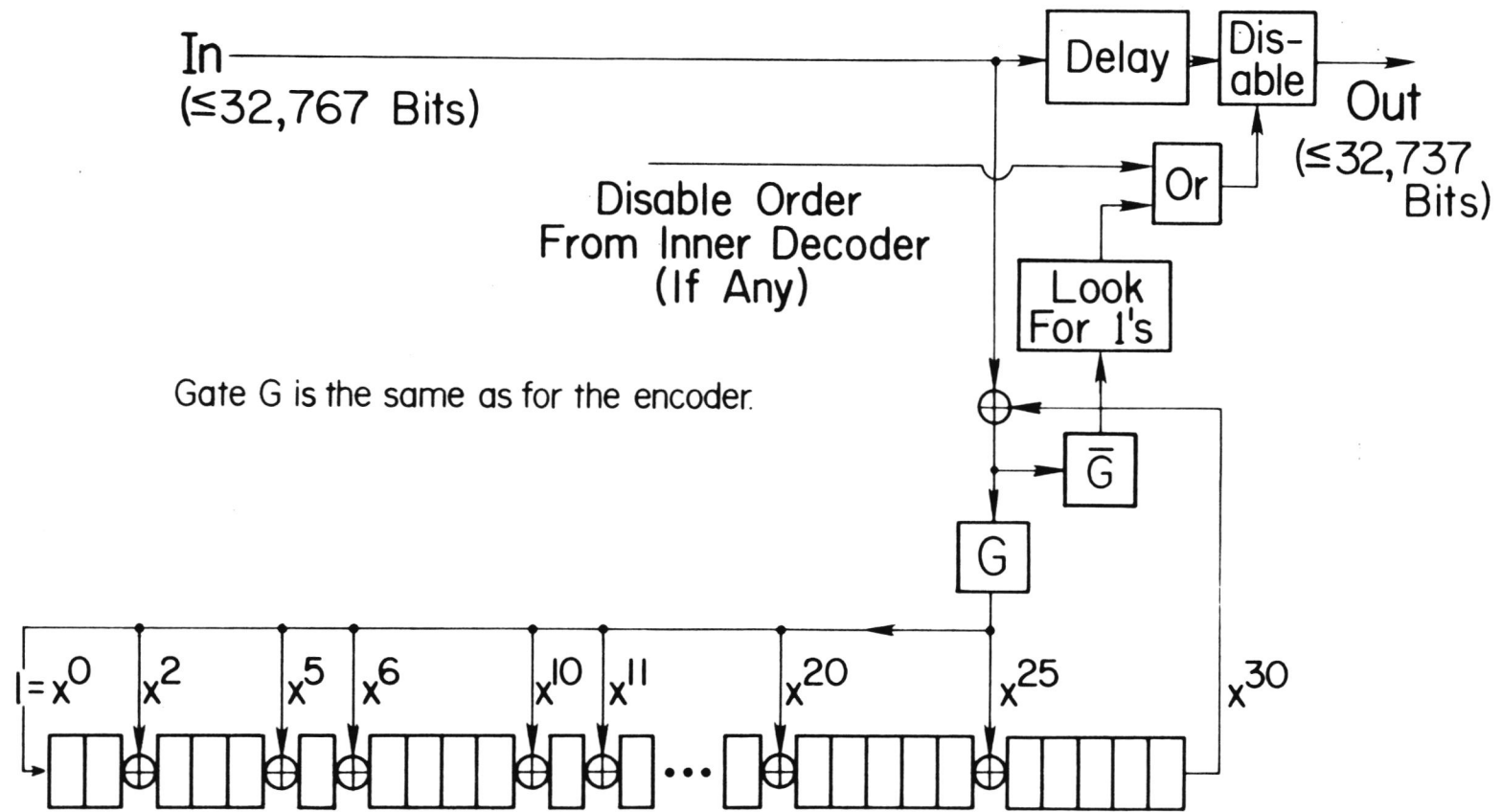


Figure 17. Outer decoder.

transmitter, and the present word output must be disabled. This process is known as the ARQ action. If all parity checks are satisfied, there is no need for ARQ action. Figure 17 also permits a secondary disable order to be used by the receiver terminal for unspecified reasons.

To select the block  $(n,k)$  code and the appropriate codecs, a convenient shorthand for the feedback taps should be used. Thus, the nine taps in figures 16 and 17 can be represented in binary as

101 001 100 011 000 000 001 000 010 000 1,  
or in octal as 5,1,4,... or by a conventional numerical listing of the locations of the taps:

0,2,5,6,10,11,20,25,30.

Whereas the coding literature seems to prefer the binary and octal representations, we shall employ the numerical location list. As seen from figures 16 and 17, such a location list consists of all registers with exterior connections. It is also evident that the first (entry 0) and the last (entry  $n-k$ ) taps must be always present. Hence, one needs only the list of the interior taps, that is, those taps that involve exclusive -OR gates. In figures 16 and 17 there are seven such interior taps, namely

2,5,6,10,11,20, and 25.

There are still nearly  $2^{n-k-1}$  different codecs possible via specifications of the shift-register taps. In table 3 we present a short selection of codes that happen to belong to the BCH category (Berlekamp, 1968; Peterson and Weldon, 1972). Note that the block length  $n$  can be increased relatively rapidly, compared to the modest number of registers,  $n-k$ . The number of inner taps seems to grow more or less proportionally to the length of the shift-register.



Table 3. Possible Error Detecting Codes for ARQ

Upper bound on block length n	Number of registers, same as n-k	Number of inner taps to exclusive -OR gates	Numerical listing of all the inner taps, same as ex- clusive -OR gates		
4 095	12	3	1,4,6		
4 095	24	13	2,3,4,7	8,9,10,11	12,14,15,16,22
8 191	13	3	1,3,4		
8 191	26	11	1,3,6,8	10,12,16,18	20,22,23
16 383	14	3	1,6,10		
16 383	28	11	3,5,6,9	10,14,15,18	19,22,24
*32 767	30	7	2,5,6,10	11,20,25	
32 767	60	33	4,6,7,8	12,13,15,16	17,18,19,21
			22,24,25,28	29,31,35,36	37,39,40,42
			43,45,46,47	50,54,56,57	58
65 535	32	15	1,2,3,6	7,8,10,13	16,17,18,21
			23,25,27		
65 535	64	28	2,10,15,18	19,20,23,26	28,31,33,35
			36,37,40,41	42,44,45,46	47,49,52,53
			55,59,60,62		
131 071	51	21	1,2,3,4	5,7,8,9	12,13,14,17
			22,26,27,34	35,36,37,38	42

\*The encoder of this code is shown in figure 16, the decoder in figure 17.

## 4.2. ARQ Performance

The performance of the outer ARQ codec involves at least three criteria:

- (i) the throughput rate  $R$ , a quantity that is related but not identical to the code rate  $R_0$ ;
- (ii) the output word error probability  $P_0$ ;
- (iii) the statistics of delays and random queues caused by the ARQ action, and the associated storage overflow implications.

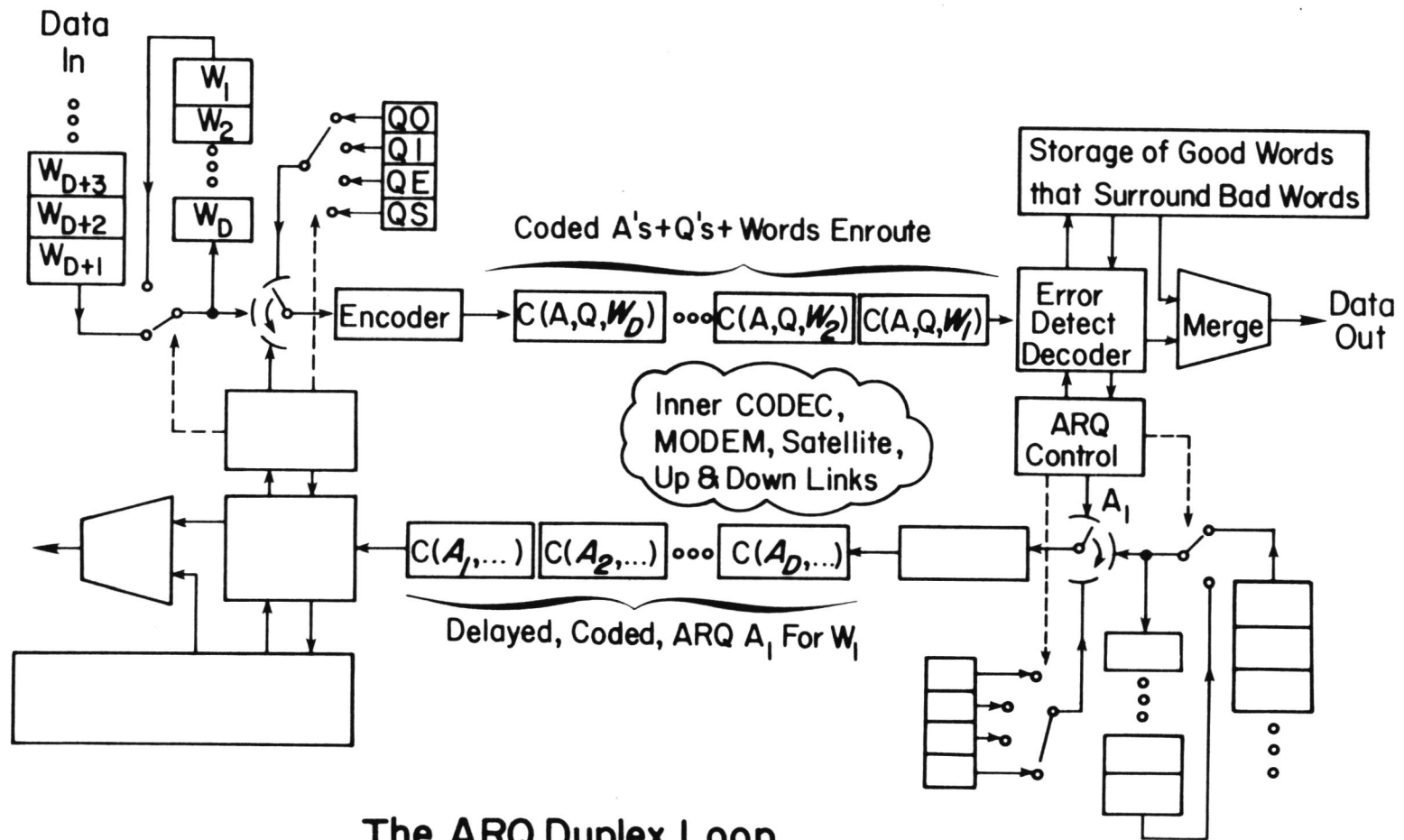
In this section we will concentrate on the first two entities,  $R$  and  $P_0$ , for a complete ARQ duplex arrangement, as outlined in figure 18.

Each terminal is both a transmitter and a receiver. It transmits forward data and ARQ acknowledgements for the return link. It receives return link data and the ARQ control bits concerning previous forward transmissions. The transmitter handles incoming information, also called "Data In," in groupings or words denoted  $W_1, W_2, \dots$ . While  $W_1$  is enroute, the path delay is such that  $W_2, \dots, W_D$  have also departed from the transmitter. Quantity  $D$  thus measures the two-way delay in units of data words. Before transmission, each word  $W_i$  is stored for eventual future repetition, if requested by ARQ. The word  $W_i$  is also joined by two short system control statements  $A_j$  and  $Q_k$ . Statement  $A_j$  is the ARQ acknowledgement for some word,  $W_j$ , received over the return channel. The functional role of  $A_j$  is straight forward:

$A_j = 0$  means OK, or no errors detected;  
 $A_j = 1$  means NOK, or some errors detected.

Thus, a single information bit suffices for  $A_j$ .

The function of  $Q_k$  is to qualify the accompanying word  $W_k$ . Perhaps it's a new word, or some previous



### The ARQ Duplex Loop

Figure 18. The ARQ duplex loop.



word  $W_i$  that is being repeated. The nature of the repeat, such as due to  $A_k$  being NOK, or  $A_k$  being erased (lost), or some system outage, may be important. Loosely speaking,  $Q_k$  is an indicator for the ARQ loop's present state. The following short table outlines a possible four state scheme for the ARQ duplex of figure 18:

A = Q0 (OK) → Clear  $W_1$ , Store  $W_{D+1}$ , Send  $C(A, Q0, W_{D+1})$   
 = Q1 (NOK) → Keep  $W_1$ , Send  $C(A, Q1, W_1)$   
 = QE (Erasure) → Keep  $W_1$ , Send  $C(A, QE, W_1)$   
 = QS (System) → Keep  $W_1$ , Send  $C(A, QS, W_1)$

Here, two bits suffice to characterize Q. After assembling, the three parts,  $A_i$ ,  $Q_j$ , and  $W_k$ , are encoded together and become  $C(A_i, Q_j, W_k)$ , or  $C(A, Q, W)$  for short. Because their bit count is so small compared to  $W$ , the segments A and Q can be safely ignored in data rate calculation.

The key to performance (i) and (ii) calculation is given by the rules of the ARQ. A synoptic start is offered in figure 19. There are three diagnoses possible on both the forward and feedback paths. A word can emerge from the inner decoder either correct, with errors that are detected, or with undetected errors. The same things happen to the ARQ acknowledgement, except that an incorrect A is said to be transposed, and a doubtful A is said to be erased.

The ARQ control logic contained in the duplex loop (fig. 18) determines the consequences for all nine alternatives shown in figure 19. Without going into too much detail, such a diagram can still be helpful. It assists to compute, to upper bound, or to lower bound performance indicators, such as R and  $P_0$ . In what follows, we shall

		Forward Path		
		$W_1$ is $A_1$ is	Correct	Detected Error
Feedback Path	Correct		<i>Sure Repeat</i>	<i>Sure Error</i>
	Erased	<i>Possible Repeats</i>		
	Transposed		<i>Error Possible</i>	

## Synopsis of ARQ

Figure 19. Synopsis of ARQ.

be content to estimate some very primitive bounds on the two quantities.

Let  $p_n(j)$  be the probability of  $j$  errors in the  $n$  bit word, where  $n$  is a large integer and  $0 \leq j \leq n$ . This probability is observed in the data stream, where it emerges from the inner decoder and enters the outer decoder. We assume that the same  $p_n(j)$  holds for the forward and feedback paths, but that the two links are otherwise statistically independent. It follows that the throughput rate  $R$  must be bounded by

$$\frac{k}{n} p_n^2(0) \leq \frac{R}{R_I} \leq \frac{k}{n} p_n(0).$$

This allegation is verified by turning to figure 20 and considering the least possible repeats, as well as the most possible, in the ARQ synopsis (fig. 19).

For example, the left side of this inequality follows from

$R \geq R_I R_0 \Pr\{W_1 \text{ accepted in the most repeat case}\}$ ,  
where  $R_0 = k/n$  and

$$\begin{aligned} \Pr\{\dots\} &= \Pr(A_1 \text{ correct}) \Pr(W_1 \text{ correct}) \\ &\quad + \Pr(A_1 \text{ correct}) \Pr(W_1 \text{ undet. error}). \end{aligned}$$

When acknowledgement  $A_1$  is shorter than word  $W_1$ ,

$$\Pr(A_1 \text{ correct}) \geq \Pr(W_1 \text{ correct}) = p_n(0),$$

and

$$\frac{R}{R_I} \geq \frac{k}{n} p_n^2(0).$$

Likewise, the right side follows from

$R \leq R_I R_0 \Pr\{W_1 \text{ accepted in the least repeat case}\}$ ,  
where now

$$\Pr\{\dots\} = \Pr(W_1 \text{ correct}) + \text{negligible terms.}$$

The ARQ word-error probability behavior is somewhat different, and calls for a different characterization (see fig. 21):



		Forward Path		
		$W_1$ is	Correct	Detected Error
Feedback Path	$A_1$ is			
	Correct		<i>Least Repeats</i>	
	Erased		<i>Most Repeats</i>	
Transposed				

## Synopsis of ARQ

Figure 20. The least and most repeats in the ARQ loop.

		Forward Path		
		$W_1$ is $A_1$ is	Correct	Detected Error
Feedback Path	Correct			<i>Least Errors</i>
	Erased			
	Transposed		<i>Most Errors</i>	

## Synopsis of ARQ

Figure 21. The least and most errors in the ARQ loop.

$$\begin{aligned}
P_0 &\geq \Pr(\text{accepted } W_1 \text{ has some error in the least error case}) \\
&\geq \frac{\Pr(W_1 \text{ undet. error})}{\Pr(W_1 \text{ correct}) + \Pr(W_1 \text{ undet. error})} \\
&\geq \Pr(W_1 \text{ undet. error})
\end{aligned}$$

$$\begin{aligned}
P_0 &\leq \Pr(\text{accepted } W_1 \text{ has some error in the most error case}) \\
&\leq \Pr(W_1 \text{ undet. error}) + \Pr(A_1 \text{ transposed}).
\end{aligned}$$

Since acknowledgement  $A_1$  is an extremely short segment (such as one or two bits) of an  $n$ -bit word, one expects

$$\Pr(A_1 \text{ transposed}) \ll \Pr(W_1 \text{ undet. error}).$$

One can show, for example, that

$$\frac{\Pr(A_1 \text{ transposed})}{\Pr(W_1 \text{ undet. error})} \cong (\text{constant}) \frac{d}{n}.$$

Here,  $d$  stands for the minimum distance of the code. A BCH code with  $n=2^m-1$  and  $n-k=mt$  has  $d \geq 2t+1$ , as is well known (Berlekamp, 1968; Gallager, 1968; Peterson and Weldon, 1972). Thus,  $d \ll n$  even for the best codes, and especially for  $n \gg 1$ . One concludes that the upper and lower bounds are indistinguishable (Nesenberg, 1963; Lucky et al., 1968):

$$\begin{aligned}
P_0 &\cong \Pr(W_1 \text{ undet. error}) \\
&\cong 2^{-(n-k)} \sum_{j=d}^n p_n(j).
\end{aligned}$$

The bounds on the throughput rate  $R$  and the outer coder word-error probability  $P_0$  have been computed for two channel models. The first model, denoted as MOD I, is the memoryless binary symmetric channel with probability  $p_I$ . For MOD I:

$$p_n(j) = \binom{n}{j} p_I^j (1 - p_I)^{n-j}, \quad 0 \leq j \leq n.$$



The computation of sums of such binomial terms is an old problem, known to be difficult for large  $n$  and small  $p_I$ , which is exactly our case. An approximation far better than the Chernoff bound or the direct Gaussian approximation is given by the asymptotic expression (Gallager, 1968),

$$\sum_{j=d}^n \binom{n}{j} p_I^j (1 - p_I)^{n-j} \cong \frac{\left(\frac{np_I}{d}\right)^d (1 - p_I)^{n-d+1}}{\sqrt{2\pi d} \left(1 - \frac{d}{n}\right)^{n-d+\frac{1}{2}} \left(1 - \frac{np_I}{d}\right)},$$

whenever  $np_I < d$ . If  $np_I \geq d$ , one can conveniently upper bound the sum by unity.

The second channel model, denoted as MOD II, postulates error bits to be effectively bunched in solid bursts of length  $v$ . It follows from a Poisson approximation that for MOD II:

$$p_n(0) \cong \exp\left(-\frac{np_I}{v}\right),$$

$$\sum_{j=d}^n p_n(j) \cong np_I / (np_I + v),$$

where again  $np_I < d$ . To carry out the computation, we have used three values  $v=5, 15, \text{ and } 45$ .

The computed results are given in a host of figures, starting with figure 22. The first three figures, 22 through 24, depict the ARQ throughput rate  $R/R_I$  as a function of the inner codec error probability  $p_I$ , all for block length

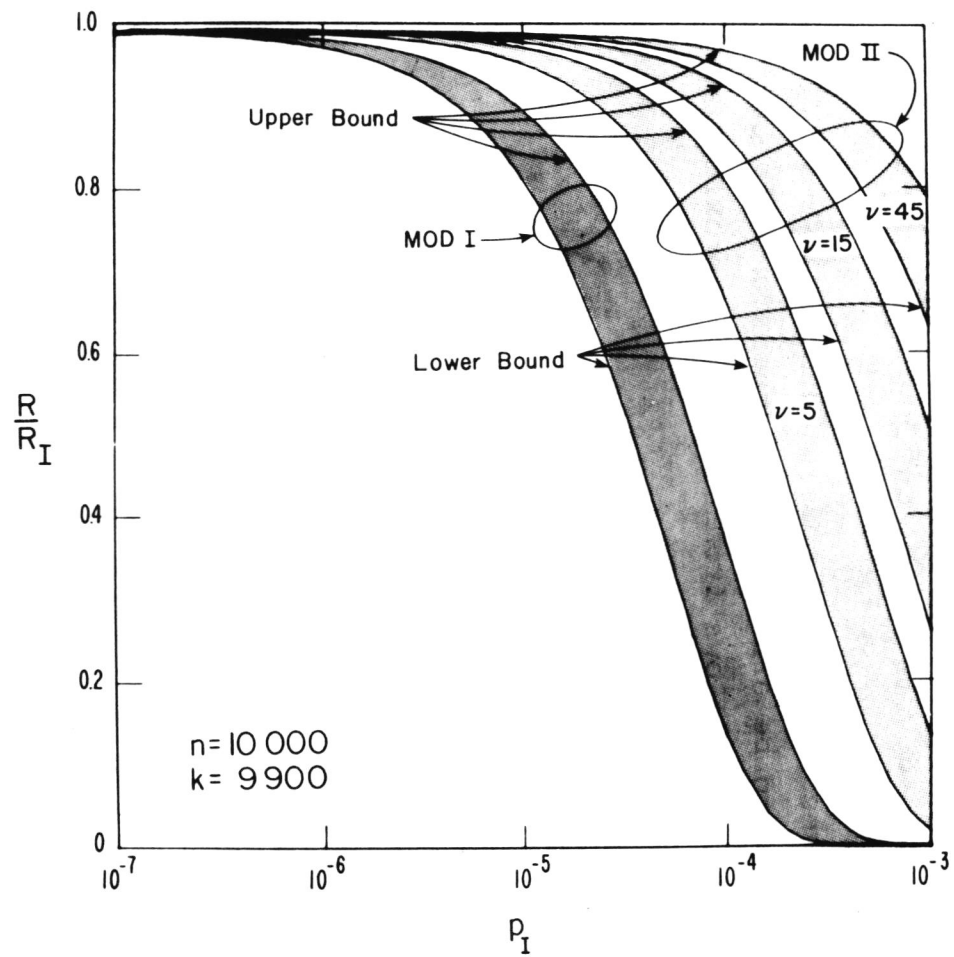


Figure 22. The bounds of throughput rate for  $n=10,000$  and  $k=9,900$ .

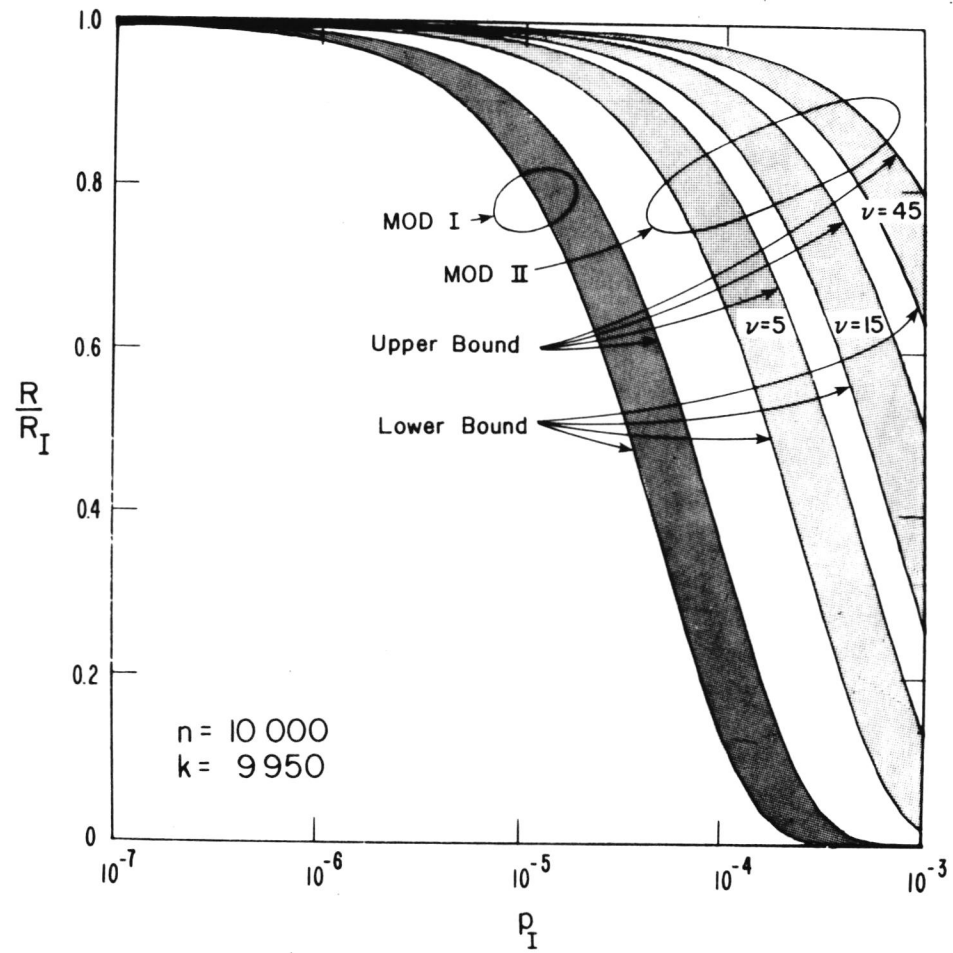


Figure 23. The bounds on throughput rate for  $n=10,000$  and  $k=9,950$ .



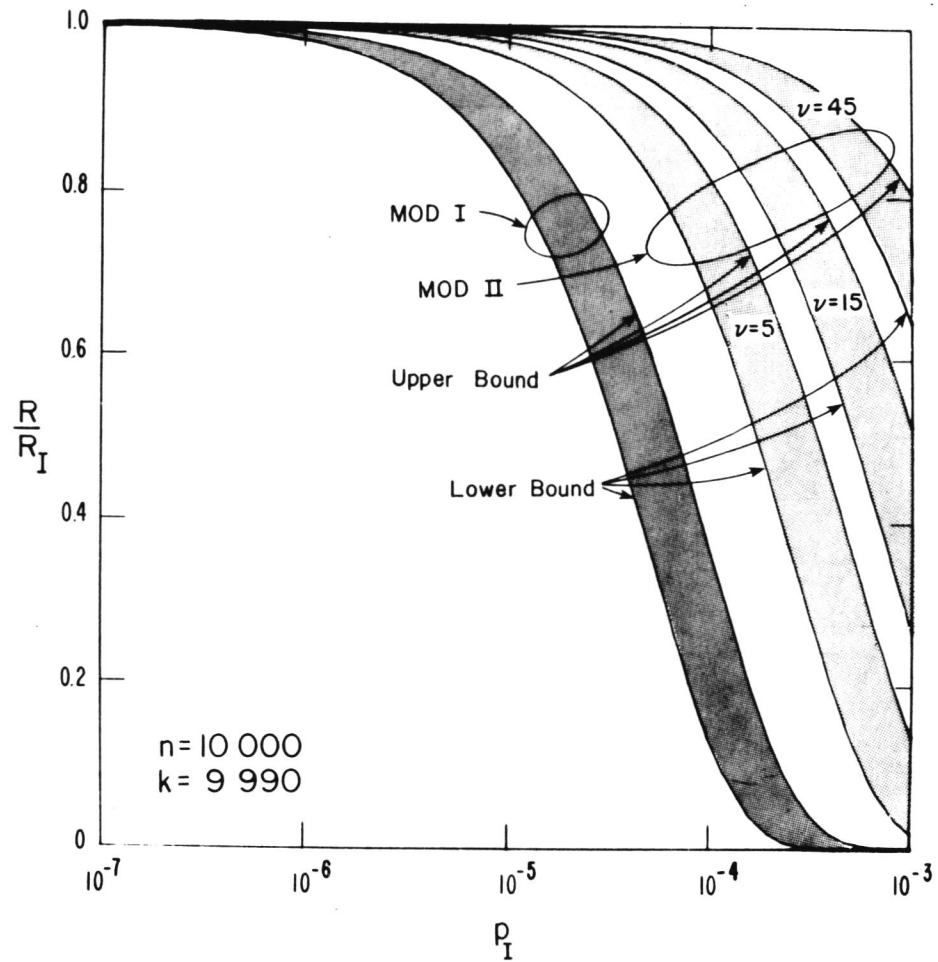


Figure 24. The bounds on throughput rate for  $n=10,000$  and  $k=9,990$ .

$n=10,000$ . The figures differ in  $k$  value: Figure 22 has  $k=9,900$ , figure 23 has  $k=9,950$ , and figure 24 has  $k=9,990$ . Each figure has eight curves. The leftmost two curves belong to MOD I, while the other six curves belong to MOD II. Furthermore, each adjacent pair of curves, as for MOD I, represent the lower and upper bounds on the outer code throughput rate  $R/R_I$ . The three lower and upper bound pairs for MOD II are descriptive of solid burst lengths  $v=5, 15, \text{ and } 45$ , respectively.

One notes the following features:

- (1) The effect of  $k$  is a very minor change in the vertical scale of the curves. Since this effect is entirely predictable, one can omit  $k$  dependence in other similar plots. Accordingly, figures 25 to 28 are all for  $k/n=0.99$ , but with different  $n$  values.
- (2) The upper and lower bounds for the same channel model are separated very nearly by a factor of 2 in the horizontal,  $p_I$ , dependence.
- (3) One can view MOD I as a special  $v=1$  case of MOD II. The effect of different  $v_1$  and  $v_2$  values then seems to correspond to a horizontal  $p_I$  shift by a factor  $v_1/v_2$ . Figure 25, for example, has a throughput upper bound  $R/R_I=0.5$  for MOD I at  $p_I \approx 2(10^{-5})$ , for MOD II,  $v=5$ , at  $p_I \approx 10^{-4}$ ,  $v=15$  at  $p_I \approx 3(10^{-4})$ , and so forth. An increase in error bursting,  $v$ , benefits the throughput rate. Unfortunately, realistic channels, including inner Viterbi decoders, are not sufficiently well understood to state what  $v$  value is applicable. One can speculate

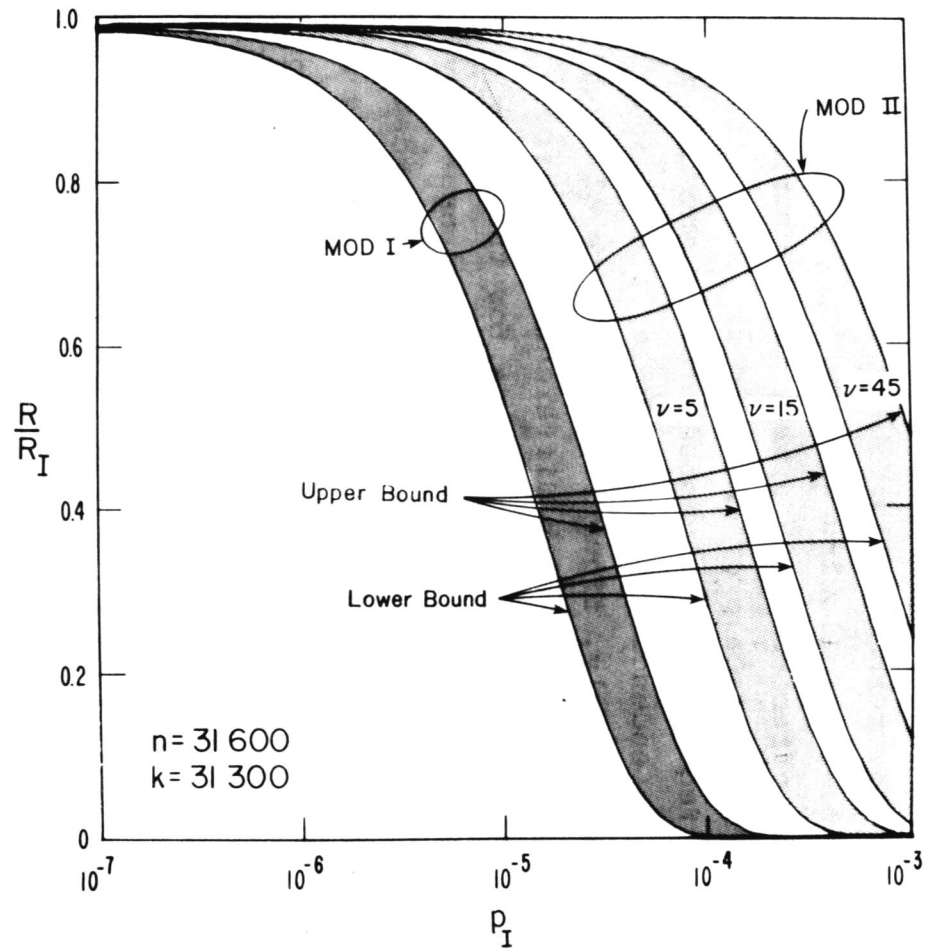


Figure 25. The bounds on throughput rate for  $n=31,600$  and  $k/n=0.99$ .



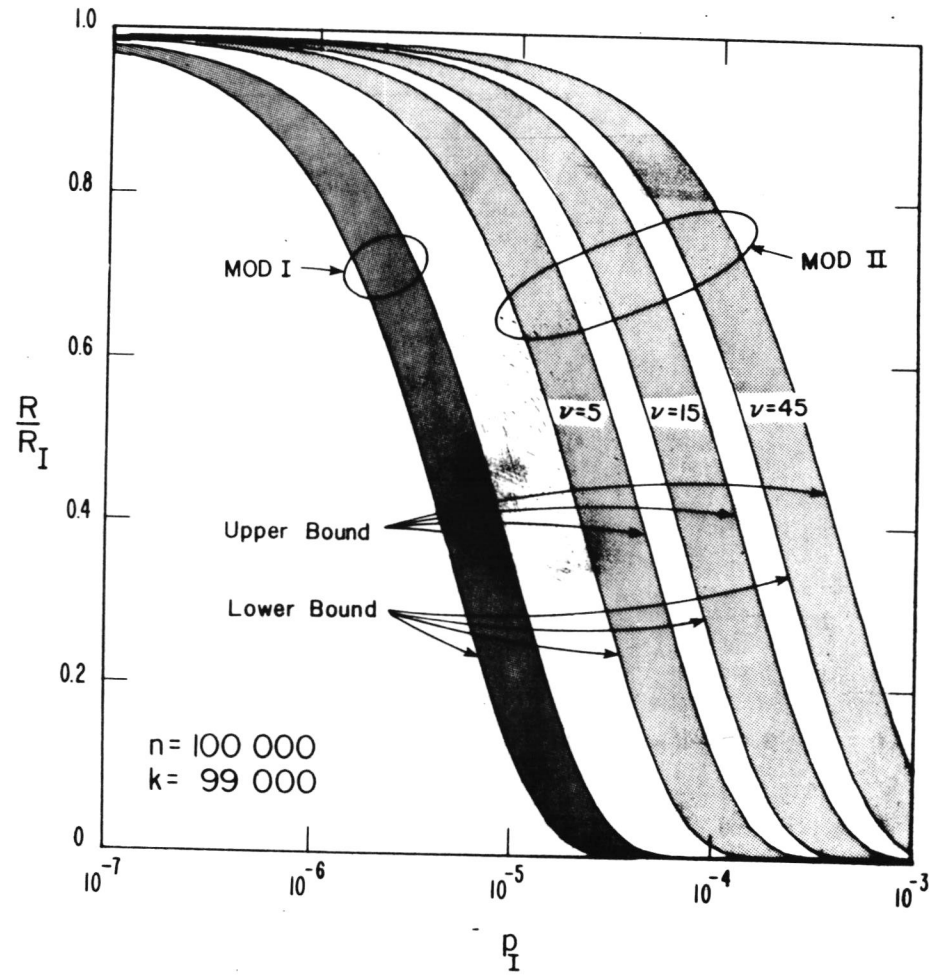


Figure 26. The bounds on throughput rate for  $n=100,000$  and  $k/n=0.99$ .

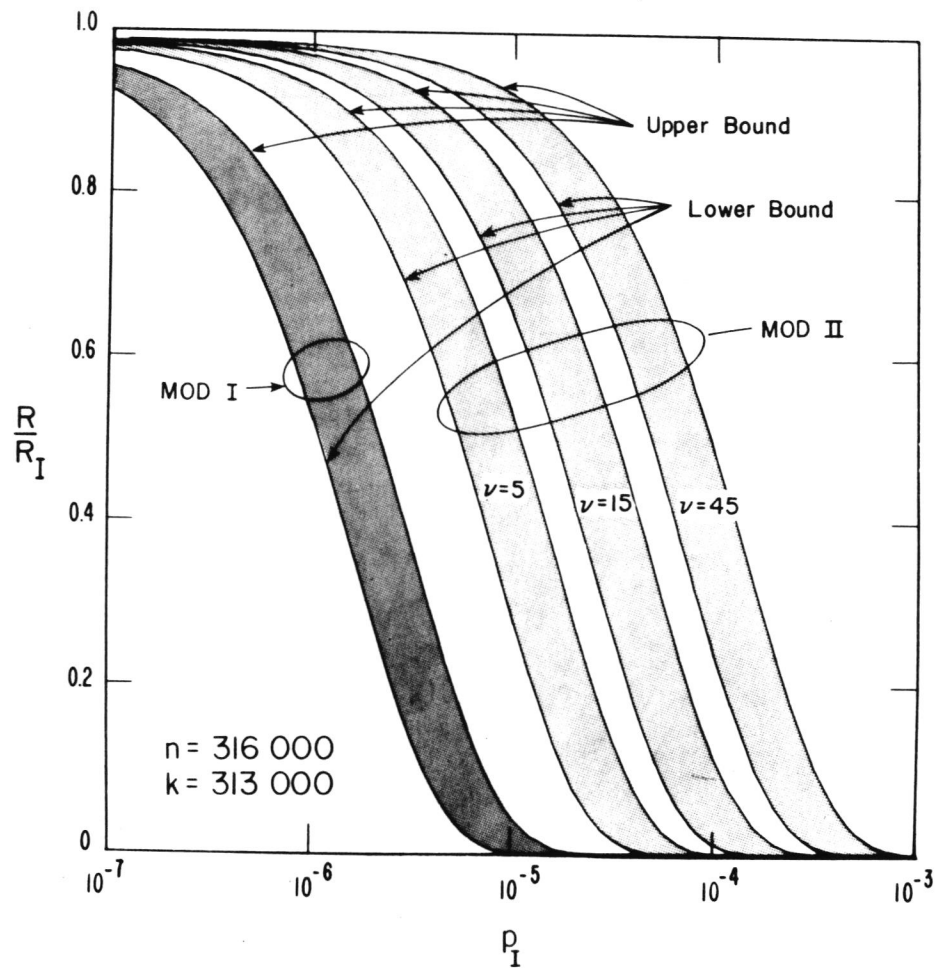


Figure 27. The bounds on throughput rate for  $n=316,000$  and  $k/n=0.99$ .

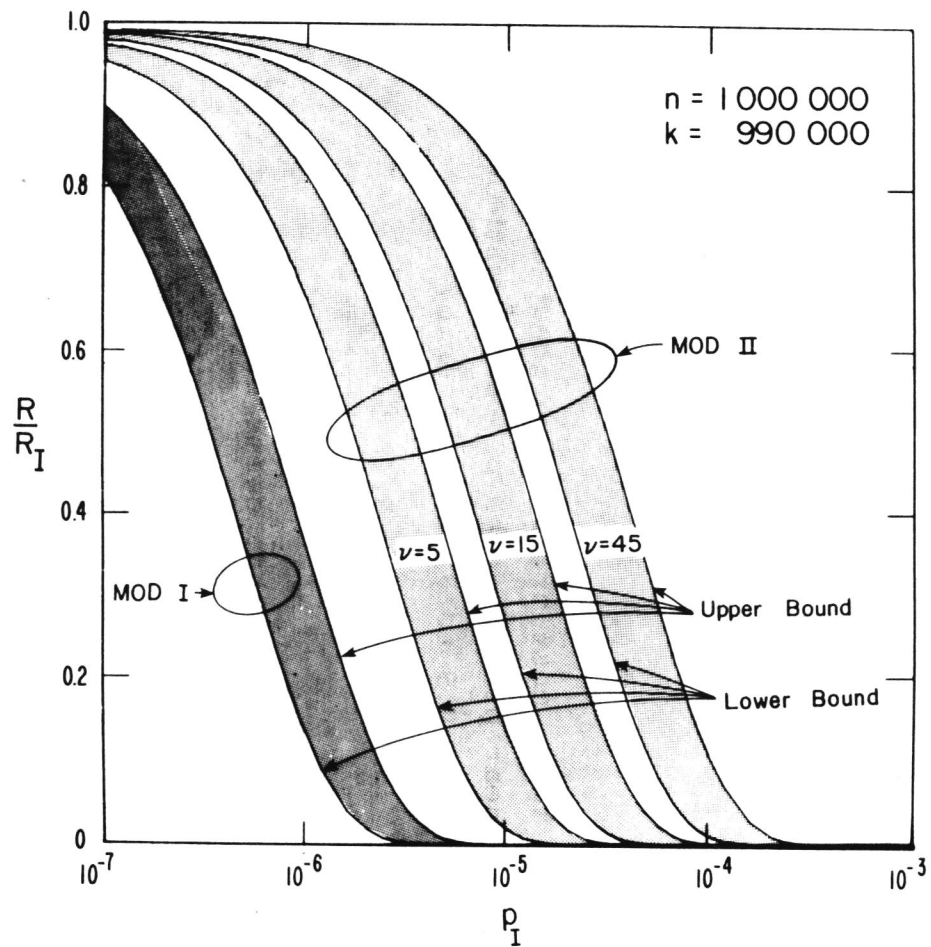


Figure 28. The bounds on throughput rate for  $n=1,000,000$  and  $k/n=0.99$ .



that for nonsystematic convolutional codes, with constraint length  $K=8$  and delay parameter  $L=40$ , a good guess could be  $v \approx 5$ . But, without further evidence the entire matter is too vague to tell.

(4) As another rough rule of thumb one can note:

$$\begin{array}{lll}
 R/R_I \approx 0.9 & \text{at} & P_I \approx \frac{v}{9n} , \\
 \approx 0.6 & \text{at} & \approx \frac{v}{2n} , \\
 \approx 0.3 & \text{at} & \approx \frac{v}{n} , \\
 \approx 0.1 & \text{at} & \approx \frac{2v}{n} .
 \end{array}$$

Thus, most of the throughput decline takes place near  $p_I = v/n$ . The  $p_I$  range, within which  $R/R_I$  drops from 0.9 to 0.1, is given by

$$\frac{p_I (R/R_I = 0.1)}{p_I (R/R_I = 0.9)} \approx 20 ,$$

a quantity that seems surprisingly independent of the channel burst parameter  $v$ .

The second performance criterion, the outer code word error probability  $P_0$ , is graphed in figures 29 to 31. It appears that  $P_0$  has some dependence on channel models, such as MOD I or MOD II, with an increase in  $v$  usually implying a relatively small  $P_0$  improvement. Moreover, since the outer decoder monitors the concatenated coding output, the overall output word error probability is simply  $P=P_0$ , for the  $n$ -bit word. The output bit error probability,  $p$ , is related to  $P_0$  also, but not in a known simple way. The obvious bounds,

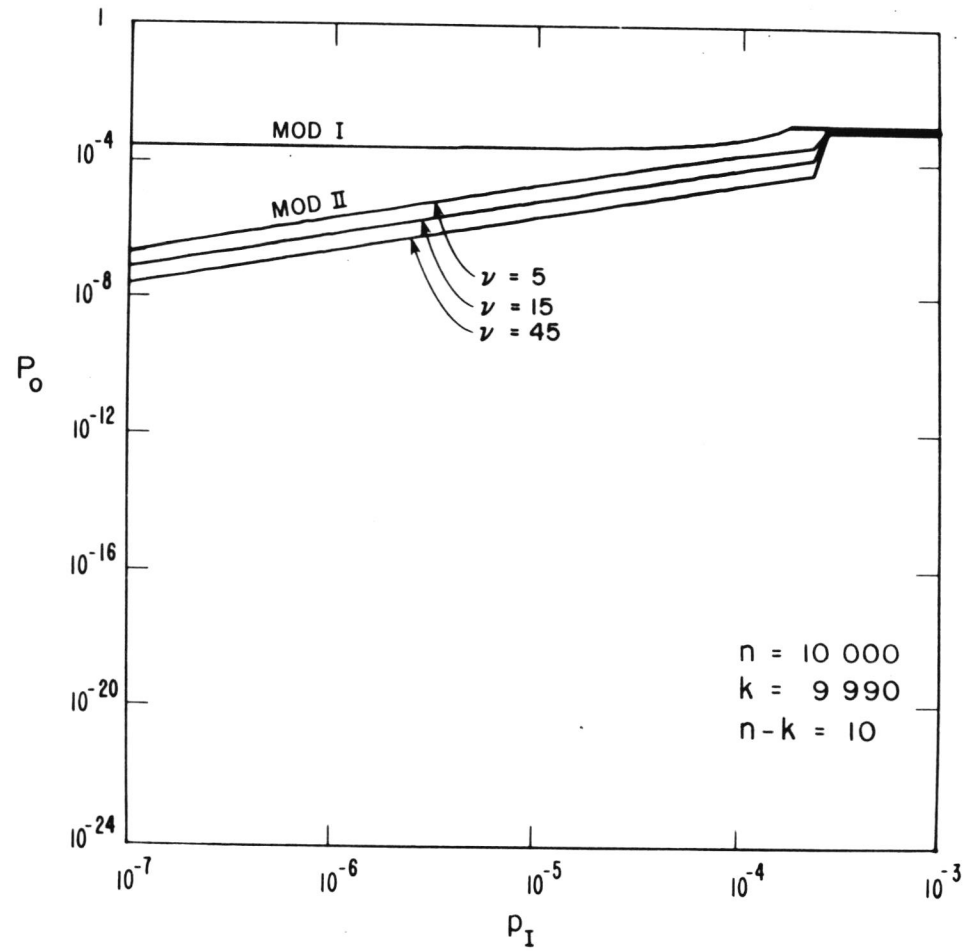


Figure 29. Output word error probability for  $n=10,000$  and  $k/n=0.999$ .

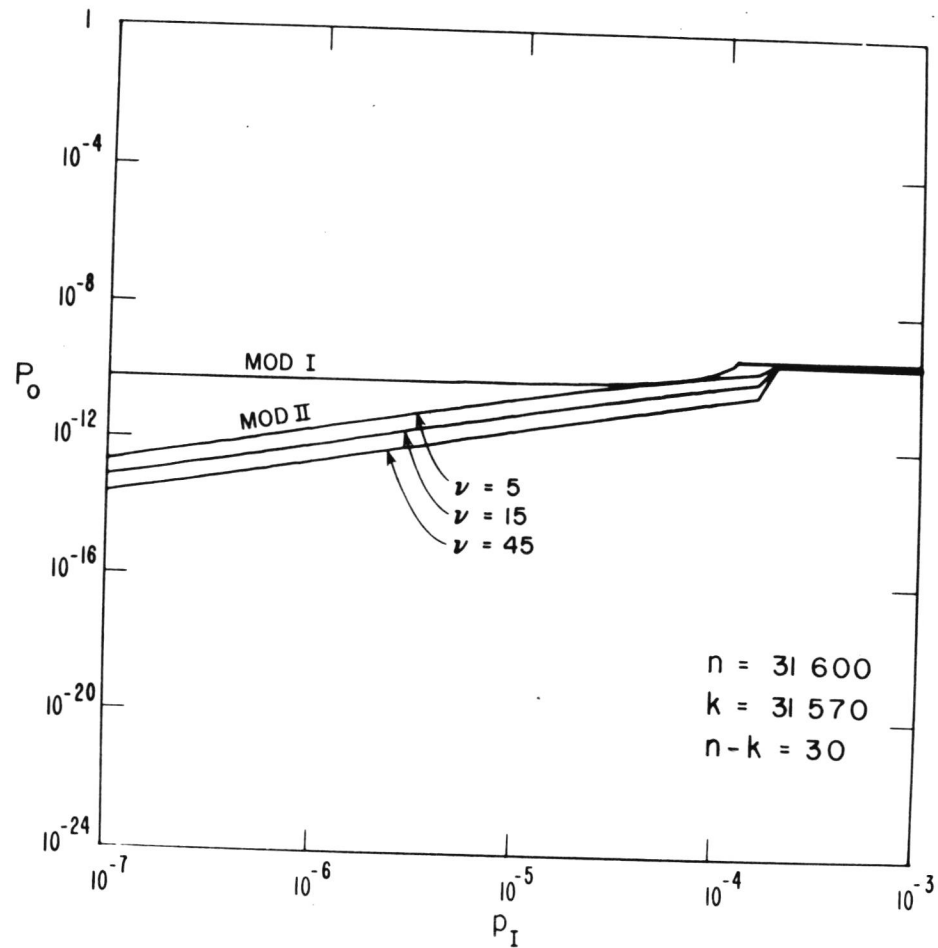


Figure 30. Output word error probability for  $n=31,600$  and  $k/n=0.999$ .



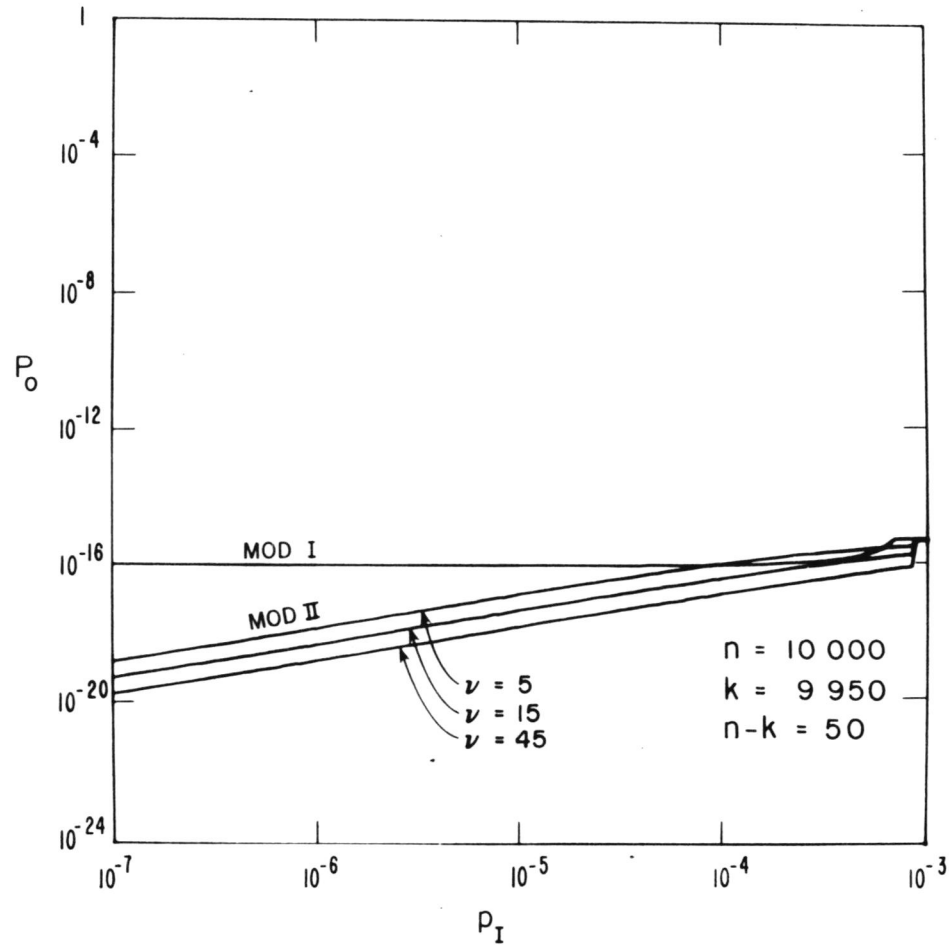


Figure 31. Output word error probability for  $n=10,000$  and  $k/n=0.995$ .

$$\frac{1}{n}P_0 \leq p \leq P_0 ,$$

can be sharpened (Viterbi, 1966) under some specific and elaborate circumstances. Unfortunately, all such refinements remain relatively ineffective compared to the gross  $2^{-(n-k)}$  bound on  $P_0$ . To assure that  $P$  and  $p$  never exceed  $10^{-12}$ , it suffices to set

$$n-k \geq \frac{12}{\log 2} \cong 40 .$$

This is done in figure 31, where  $n-k=50$  yields an effective error detecting scheme with an almost insignificant 0.5 percent redundancy for block length  $n=10,000$ .

The third performance criterion (iii) dealing with random ARQ delays and queues will not be commented on, because of insufficient understanding on our part. The matter seems important and, if time permits, should be studied in the future. The large path delay on satellite links compounds the ARQ processing difficulty (Balcovic and Muench, 1969; McGruther, 1972; Abramson and Kuo, 1973; Lucky, 1974; Sastry, 1974) and, by all indications, calls for innovative ARQ designs.

## 5. OVERALL PERFORMANCE

The concatenated hybrid, FEC and ARQ, coding has an overall performance yet to be described. The advantages and disadvantages are both consequences of inner and outer coder features. For instance, the rate slowdown or bandwidth expansion is a drawback caused by both coders, though in different ways. Another disadvantage is the implementation, maintenance, and other complexity related costs. The advantages boil down to one: more reliable data output.

This error rate reduction, of course, is also a resultant of the workings of both codes. Previously, we have described the two codecs separately; now we shall combine them and summarize the overall performance.

First, consider the overall output error probability. As pointed out in the ARQ performance discussion (sec. 4.2), this performance criterion is for all practical purposes the same  $P_0$  as observed at the ARQ output. It was shown that a modest investment in  $n-k \geq 40$  parity check bits for the outer code implies overall word and bit probabilities,  $P$  and  $p$ , at or below  $10^{-12}$ . Other parameters and system arrangements may reduce the error rate further, but not substantially. One concludes that a block code with  $n \geq 10,000$  and not less than 40 parity checks (which amounts to 0.4 percent redundancy) will provide error control to keep the output binary error probability at the desired level.

Our next concern is the overall throughput rate of the hybrid scheme. The work on inner and outer coders implies that the overall rate  $R$  cannot exceed the product of the two constituent rates  $R_I$  and  $R_0$ . If there are no ARQ repeats, then  $R = R_I R_0$ . In the presence of repeats,  $R < R_I R_0$ . The inner rate  $R_I$  was assumed to be  $1/2$ , or  $3/4$ , or perhaps  $2/3$ . The outer rate  $R_0$  is further reduced by ARQ. Since the effects are complicated, exact calculation was abandoned in favor of simpler bounds. Such upper and lower bounds on the relative rate  $R/R_I$  were presented in figures 22 to 28. The abscissa for these plots was the binary error probability,  $p_I$ , at the Viterbi decoder output. The  $p_I$  itself was shown to be a function of channel signal-to-noise ratio,  $E_b/N_0$  (dB), in figures 13 to 15, and subject to many assumptions. A merger of these two families of curves



produces the sought overall throughput rate,  $R$ , versus the signal-to-noise ratio, all capable of meeting the  $10^{-12}$  binary error rate objective. Four such relationships are plotted in figure 32. While many more such curves are possible, the given four appear rather typical. Note the pronounced  $E_b/N_0$  threshold effect. Above the threshold, data passes with negligible or no slowdown. Near the threshold, repeats and non-repeats occur with comparable frequency. Further below the threshold, the throughput  $R$  ceases and the link gets turned off by the ARQ.

Since different conditions and parameter choices give cause for distinct curves in figure 32, it seems useful to graphically lump together and characterize the various threshold curves. Such an attempt has led to figure 33 and, alas, the resultant mastercurve does not seem to be an accurate model for specific curves (as plotted in fig. 32). Hence a warning: Use figure 33 for rough estimates only. A key parameter in figure 33 is the  $n/v$  ratio. Given the same  $n/v$ , the rate 1/2 and rate 3/4 inner codes lead to almost indistinguishable characteristics, as far as the  $0.9 R_I R_0$ ,  $0.5 R_I R_0$ , and  $0.1 R_I R_0$  points are concerned. One concludes that the 3/4 rate inner code is preferable, since it offers more throughput for the above threshold operation.

It is seen from figures 32 and 33 that the  $E_b/N_0$  threshold occurs somewhere between 5 and 7 dB. More precise numbers can be pinned down with the help of methods and parameters presented earlier, but one wonders about the relative errors inherent in such estimates and the confidence levels they merit. Nevertheless, it is significant to stress that an operational signal-to-noise ratio,  $E_b/N_0=6.5$  dB, though it might be a ballpark figure,

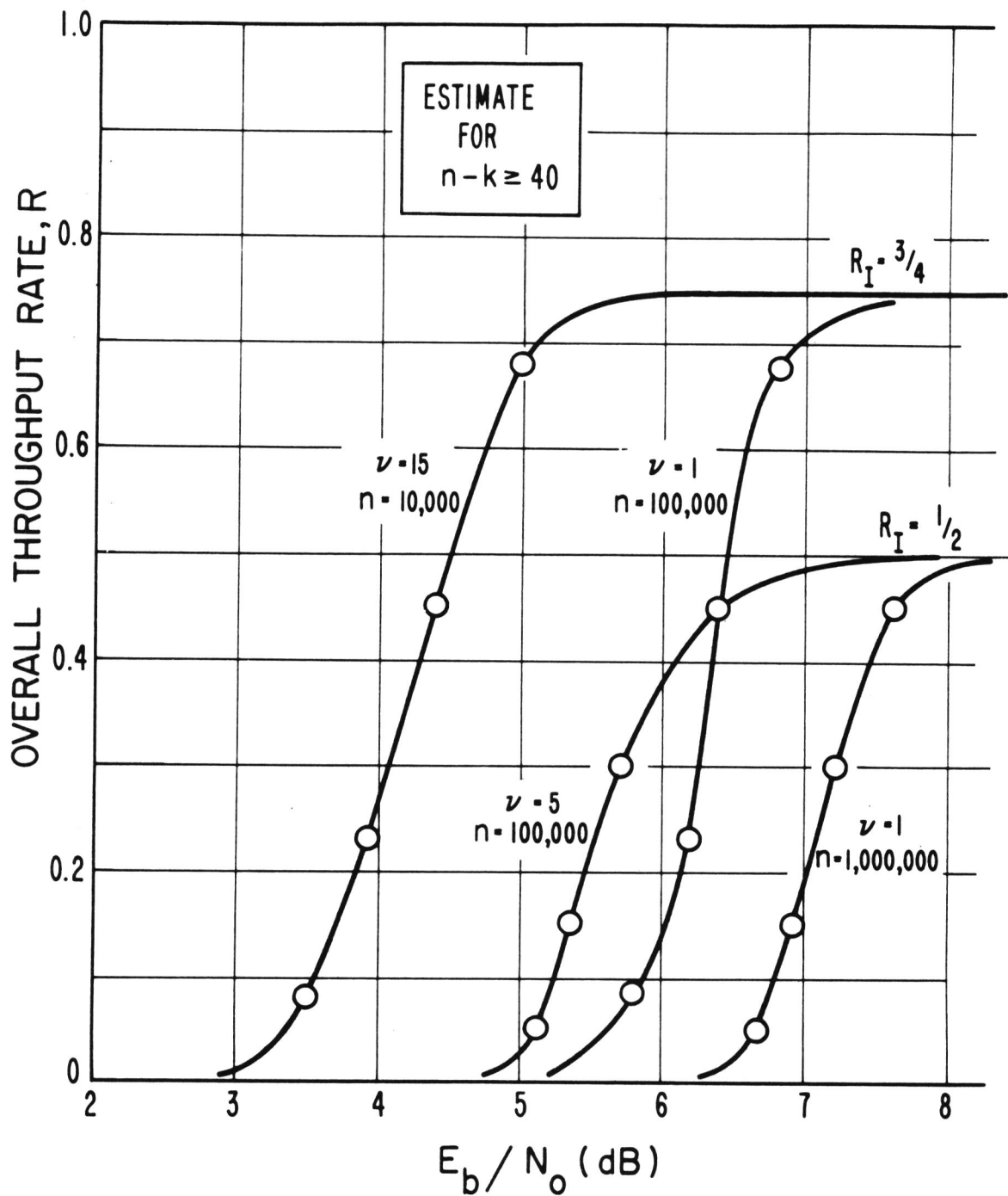


Figure 32. Estimated overall throughput rate for specific burst channels ( $\nu$ ), code length ( $n$ ), and signal-to-noise ratio ( $E_b/N_0$ ).

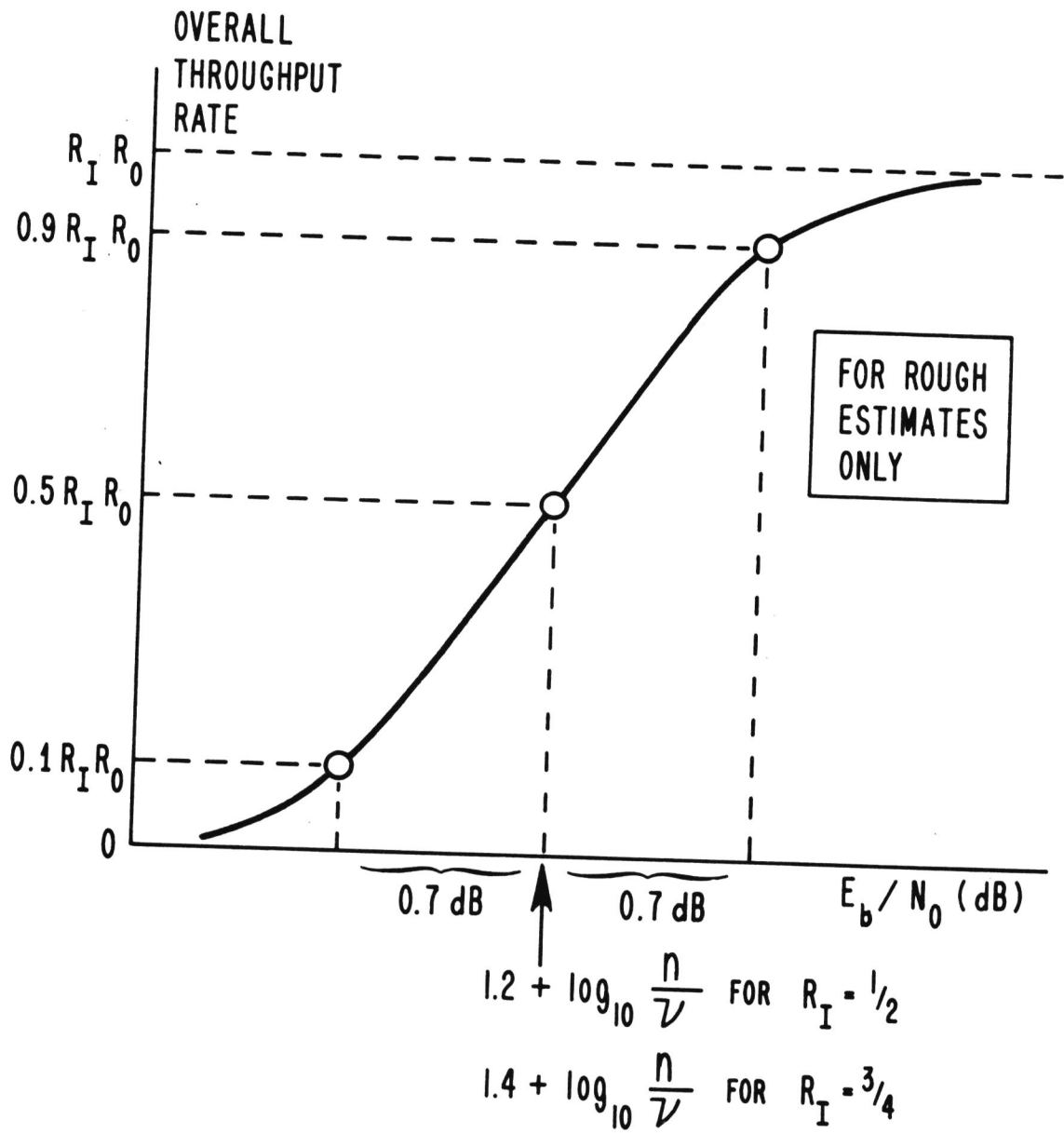


Figure 33. Guideline for rough estimation of the overall throughput rate.



is still some 14 dB below the uncoded extrapolated-measured requirement for the  $10^{-12}$  error rate level (McManamon et al., 1974). And, if uncoded operation were to be unsatisfactory due to irreducible error rate effects (see also earlier sec. 3.6), then error control coding would indeed become indispensable.

A brief final comment concerning figure 33 and the slight advantage of the rate 3/4 inner code. Perhaps, further increase of  $R_I$  could yield even better results. Taken to the ultimate limit,  $R_I \rightarrow 1$ , can one then discard the inner coder and the entire concatenated hybrid scheme? A simple check with figure 14, however, tells a different story. To operate at the required  $P_e$  level, the ARQ outer code by itself (i.e., without the inner code) would require  $E_b/N_0$  somewhere between 10 and 12 dB to stay above the threshold. Thus, a good inner FEC code is needed to provide reliable transmissions down to  $E_b/N_0 \approx 6$  dB. As mentioned earlier, this figure includes the 1-3 dB code rate loss, but fails to reflect the 33 percent to 100 percent bandwidth expansion (or transmission rate reduction).

## 6. CONCLUSIONS

Many coding alternatives have been considered here. The less suited ones were dismissed with little comment, the better ones were scrutinized in more detail. An alleged "first choice" error control code was selected. It turned out to be a hybrid of forward error correction (FEC) and automatic repeat request (ARQ), implemented by concatenation of two codes: an inner convolutional code with Viterbi decoding and an outer error detecting block code. To be certain that our chosen code is as good as it

appears, we have delved quite intensely in all its technical aspects. Features such as implementation, operational principles, present day state-of-the-art, costs, complexities, robustness, versatility, and the expected performance on realistic channels have been compiled, reviewed, and in some cases computed anew.

The main advantage of the selected codec is to ensure, with high confidence, practically error free operation. It can do so at its nominal throughput rate, as long as the normalized signal-to-noise ratio,  $E_b/N_0$  (dB), exceeds roughly 6 dB. If, due to a sudden fade, the signal-to-noise ratio falls below the threshold, the codec will effectively interrupt the data flow and stop delivery of erroneous digits.

The main disadvantage of the scheme is the need to transmit and process redundant bits. If the modems and the link bandwidths are fixed, the only way to accommodate additional bits is to reduce the information throughput. To maintain constant information throughput, on the other hand, requires bandwidth expansion. The additional occupied bandwidth is given by  $(1-R)/R$ , where  $R$  is the overall throughput rate. Thus, a 33 percent expansion appears necessary for  $R=3/4$  and the system given. The second disadvantage of introducing codecs is the burden of the devices themselves. Fortunately, the coding scheme selected can be simply implemented. Whether viewed in hardware or software sense, the added system is of the same complexity as a mini-computer.

A final unresolved item concerns the random queues and delays caused by the ARQ on the long-delay satellite links. The strategy and proper handling of these possibly large queues is a complex, interrelated process involving

some balance between data processing and communications. System studies of these queues, particularly for network applications, will require computer simulation because of the analytical complexities discussed in this report.



## 7. REFERENCES

- Abramson, N., and F.F. Kuo (1973), Computer-Communication Networks (Prentice-Hall, Englewood Cliffs, N.J.).
- Balcovic, M.D., and P.E. Muench (1969), Effect of propagation delay, caused by satellite circuits, on data communication systems that use block retransmission for error correction, Conf. Record of 1969 ICC, Boulder, Colo., pp. 29/31-29/36.
- Batson, B.H., R.W. Moorehead, and S.Z.H. Taqvi (1972), Simulation results for the Viterbi decoding algorithm, NASA Tech. Rept. R-396, Manned Spacecraft Center, Houston, Tex.
- Benice, R.J., and A.H. Frey (1964), An analysis of retransmission systems, IEEE Trans. Commun. Technol., COM-12, pp. 135-145.
- Berlekamp, E.R. (1968), Algebraic Coding Theory (McGraw-Hill Book Co., New York, N.Y.).
- Burton, H.O. (1970), A survey of error correcting techniques for data on telephone facilities, Conf. Record of 1970 ICC, San Francisco, Calif., pp. 16/25-16/32.
- Bussgang, J.J. (1965), Some properties of binary convolutional code generators, IEEE Trans. Inform. Theory, IT-11, pp. 90-100.
- Cahn, C.R., G.K. Huth, and C.R. Moore (1973), Simulation of sequential decoding with phase-locked demodulation, IEEE Trans. Commun., COM-21, pp. 89-97.
- Cain, J.B. (1971), Hard decision sequential decoding, Memo. Rept. No. 43, Radiation, Melbourne, Fla.

- Cain, J.B. (1972), An analysis of two branch synchronization schemes for convolutional codes, Memo. Rept. No. 46, Radiation, Melbourne, Fla.
- Clark, G.C., and R.C. Davis (1971), Two recent applications of error-correction coding to communications system design, IEEE Trans. Commun. Technol., COM-19, pp. 856-863.
- Clark, G.C. (1971), Implementation of maximum likelihood decoders for convolutional codes, Proc. ITC 1971, Washington, D.C., VII, pp. 449-459.
- Dodds, J.G. (1973), Analysis and simulation of parallel sequential decoding, Tech. Doc. 290, Naval Elec. Lab. Center, San Diego, Calif.
- Fano, R.M. (1963), A heuristic discussion of probabilistic decoding, IEEE Trans. Inform. Theory, IT-9, pp. 64-74.
- Forney, G.D. (1966), Concatenated Codes (MIT Press, Cambridge, Mass.).
- Forney, G.D. (1967), Review of random tree codes, NASA CR 73176, Final Report, NASA Ames Res. Cent., Moffett Field, Calif.
- Forney, G.D., and R.M. Langelier (1969), A high-speed sequential decoder for satellite communications, Conf. Record of 1969 ICC, Boulder, Colo., pp. 39/9-39/17.
- Forney, G.D. (1970), Coding and its application in space communication, IEEE Spectrum, Vol. 7, No. 6, pp. 47-58.
- Forney, G.D., and E.K. Bower (1971), A high-speed sequential decoder: Prototype design and test, IEEE Trans. Commun. Technol., COM-19, pp. 821-835.

- Forney, G.D. (1972), Convolutional codes II: Maximum likelihood decoding, Tech. Dept. 7004-1, Stanford Electronic Labs., Stanford, Calif.
- Forney, G.D. (1973), The Viterbi algorithm, Proc. IEEE (Invited Paper), Vol. 61, pp. 268-278.
- Gallager, R.G. (1968), Information Theory and Reliable Communication (John Wiley and Sons, New York, N.Y.).
- Gilhousen, K.S., J.A. Heller, I.M. Jacobs, and A.J. Viterbi (1971), Coding systems study for high data rate telemetry links, NASA CR 114278, Rept. prepared by Linkabit of San Diego, Calif., for NASA Ames Res. Cen., Moffett Field, Calif.
- Gilhousen, K.S., and D.R. Lumb (1972), A very high speed hard decision sequential decoder, Conf. Record of 1972 NTC, Houston, Tex., pp. 13C/1-13C/5.
- Goldman, H.D., W. Rowan, and M. Tolhurst (1969), Threshold decoding selected rate one-half and one-third convolutional codes, IEEE Trans. Inform. Theory, IT-15, pp. 179-184.
- Heller, J.A. (1968), Short constraint length convolutional codes, JPL Space Programs Summary 37-54, Vol. III, Pasadena, Calif., pp. 171-177.
- Heller, J.A. (1969), Improved performance of short constraint length convolutional codes, JPL Space Programs Summary 37-56, Vol. III, Pasadena, Calif., pp. 83-84.
- Heller, J.A., and I.M. Jacobs (1971), Viterbi decoding for satellite and space communications, IEEE Trans. Commun. Technol., COM-19, pp. 835-848.
- Hoffman, L.B., and J.P. Odenwalder (1972), Hybrid and concatenated coding application, Proc. ITC 72, Los Angeles, Calif., VIII, pp. 522-531.



- Jacobs, I.M. (1967), Sequential decoding for efficient communication from deep space, IEEE Trans. Commun. Technol., COM-15, pp. 492-501.
- Jacobs, I.M. (1974), Practical applications of coding, IEEE Trans. Inform. Theory, IT-20, pp. 305-310.
- Jelinek, F. (1969), A fast sequential decoding algorithm using a stack, IBM J. Res. and Dev., Vol. 13, pp. 675-685.
- Kobayashi, H. (1971), Correlative level coding and maximum-likelihood decoding, IEEE Trans. Inform. Theory, IT-17, pp. 586-594.
- Kohlenberg, A., and G.D. Forney (1968), Convolutional coding for channels with memory, IEEE Trans. Inform. Theory, IT-14, pp. 618-626.
- Layland, J.W., and W.A. Lushbaugh (1971), A flexible high-speed sequential decoder for deep space channels, IEEE Trans. Commun. Technol., COM-19, pp. 813-820.
- Lin, S., and H. Lyne (1967), Some results on binary convolutional code generators, IEEE Trans. Inform. Theory, IT-13, pp. 134-139.
- Lucky, R.W., J. Salz, and E.J. Weldon, Jr. (1968), Principles of Data Communications (McGraw-Hill Book Co., New York, N.Y.).
- Lucky, R.W. (1973), A survey of the communication theory literature: 1968-1973, IEEE Trans. Inform. Theory, IT-19, pp. 725-739.
- Lumb, D.R. (1969), Test and preliminary flight results on the sequential decoding of convolutionally encoded data from Pioneer IX, Conf. Record of 1969 ICC, Boulder, Colo., pp. 39/1-39/8.

- Massey, J.L. (1963), Threshold Decoding (MIT Press, Cambridge, Mass.).
- Massey, J.L., and M.K. Sain (1968), Trunk and tree searching properties of the Fano sequential decoding algorithm, Proc. of 6th Allerton Conf., Univ. of Illinois, pp. 153-160.
- Massey, J.L., and D.J. Costello (1971), Nonsystematic convolutional codes for sequential decoding in space applications, IEEE Trans. Commun. Technol., COM-19, pp. 806-813.
- Massey, J.L. (1973), Coding techniques for digital communications, Tutorial Notes at 1973 ICC, Seattle, Wash.
- McGruther, W.G. (1972), Throughput of high speed data transmission systems using block retransmission error control schemes over voicebandwidth channels, Conf. Record of 1972 ICC, Philadelphia, Penn., pp. 15/19-15/24.
- McManamon, P.M., J.R. Juroshek, and M. Nesenbergs (1974), Study of satellite frequency requirements for the U.S. Postal Service electronic mail system; Volume II: Frequency allocations and network designs, U.S. Dept. of Commerce, Office of Telecommunications Report, OTR 74-27, Boulder, Colo.
- Nesenbergs, M. (1963), Comparison of the 3-out-of-7 ARQ with Bose-Chaudhuri-Hocquenghem coding systems, IEEE Trans. Commun. Systems, CS-11, pp. 202-212.
- Odenwalder, J.P., K.S. Gilhousen, J.A. Heller, I.M. Jacobs, F. Jelinek, and A.J. Viterbi (1972), Hybrid coding systems study, NASA CR 114486, Rept. prepared by Linkabit of San Diego, Calif., for NASA Ames Res. Cen., Moffett Field, Calif.

- Omura, J.K. (1969), On the Viterbi decoding algorithm, IEEE Trans. Inform. Theory, IT-15, pp. 177-179.
- Omura, J.K. (1971), Optimal receiver design for convolutional codes and channels with memory via control theoretical concepts, Inform. Sci., Vol. 3, pp. 243-266.
- Paaske, E. (1974), Short binary convolutional codes with maximum free distance for rates  $2/3$  and  $3/4$ , IEEE Trans. Inform. Theory, IT-20, pp. 683-689.
- Peterson, W.W., and E.J. Weldon (1972), Error-correcting Codes, second edition (MIT Press, Cambridge, Mass.).
- Robinson, J.P., and A.J. Bernstein (1967), A class of binary recurrent codes with limited error propagation, IEEE Trans. Inform. Theory, IT-13, pp. 106-113.
- Rudolph, L.D. (1967), A class of majority logic decodable codes, IEEE Trans. Inform. Theory, IT-13, pp. 305-307.
- Sastry, A.R.K. (1974), Error control techniques for satellite communications: An overview, Conf. Record of 1974 ICC, Minneapolis, Minn., pp. 6A/1-6A/5.
- Savage, J.E. (1966), Sequential decoding - the computation problem, Bell System Tech. J., Vol. 45, pp. 149-176.
- Townsend, R.W., and R.N. Watts (1964), Effectiveness of error control in data communications over a switched telephone network, Bell System Tech. J., Vol. 43, pp. 2611-2638.
- van Duuren, H.C.A. (1961), Error probability and transmission speed on circuits using error detection and automatic repetition of signals, IRE Trans. Commun. Systems, CS-9, pp. 38-50.
- Viterbi, A.J. (1966), Principles of Coherent Communications (McGraw-Hill Book Co., New York, N.Y.).
- Viterbi, A.J. (1967), Error bounds for convolutional codes and an asymptotically optimum decoding algorithm, IEEE Trans. Inform. Theory, IT-13, pp. 260-269.



- Viterbi, A.J., and J.P. Odenwalder (1969), Further results on optimal decoding of convolutional codes, IEEE Trans. Inform. Theory (Corresp.), IT-15, pp. 732-734.
- Viterbi, A.J. (1971), Convolutional codes and their performance in communication systems, IEEE Trans. Commun. Technol., COM-19, pp. 751-772.
- Wolf, J.K. (1973), A survey of coding theory: 1967-1972, IEEE Trans. Inform. Theory, IT-19, pp. 381-389.
- Wozencraft, J.M. (1957), Sequential decoding for reliable communication, Conf. Record of 1957 Nat. IRE Conv., Vol. 5, Pt. 2, pp. 11-25.
- Wozencraft, J.M., and B. Reiffen (1961), Sequential Decoding (MIT Press, Cambridge, Mass.).
- Zeoli, G.W. (1971), Coupled decoding of block-convolutional concatenated codes, Ph.D. Dissertation, Dept. Elec. Eng., Univ. of California, Los Angeles.
- Zigangirov, K.S. (1966), Some sequential decoding procedures, Problemy Peredachi Informatsii (USSR), Vol. 2, pp. 13-25.

## BIBLIOGRAPHIC DATA SHEET

1. PUBLICATION OR REPORT NO.		2. Gov't Accession No.	3. Recipient's Accession No.
4. TITLE AND SUBTITLE Study of Error Control Coding for the U.S. Postal Service Electronic Message System		5. Publication Date December	6. Performing Organization Code OT/ITS
7. AUTHOR(S) Martin Nesenbergs		9. Project/Task/Work Unit No.	
8. PERFORMING ORGANIZATION NAME AND ADDRESS U.S. Department of Commerce Office of Telecommunications Institute for Telecommunication Sciences 325 Broadway, Boulder, Colorado 80302		10. Contract/Grant No.	
11. Sponsoring Organization Name and Address U.S. Postal Service 11711 Park Lawn Drive Rockville, Maryland 20852		12. Type of Report and Period Covered	
		13.	
14. SUPPLEMENTARY NOTES			
15. ABSTRACT (A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here.) A U.S. Postal Service (USPS) electronic message system could incorporate many types of error control coding, or no coding at all. This report reviews a variety of possible codes, lists their advantages and disadvantages, and selects a preferred alternative. It turns out to be a concatenation of an inner convolutional (rate 1/2 to rate 3/4) code with Viterbi decoding, and an outer long block, high efficiency code. The two codes have separate functions, in the sense that the inner code performs forward error correction and the outer code does error detection only. The report describes the structures, properties, and implementations of the coding hybrid. After that, the performance of the preferred coding scheme is estimated. The resultant error probability gains, which are shown to be considerable, are balanced against system slowdown and bandwidth expansion.			
16. Key words (Alphabetical order, separated by semicolons) ARQ; coding gains; concatenated codes; error probability; FEC; hybrid operation; modem losses; throughput; Viterbi decoding			
17. AVAILABILITY STATEMENT <input checked="" type="checkbox"/> UNLIMITED.  <input type="checkbox"/> FOR OFFICIAL DISTRIBUTION.		18. Security Class (This report) Unclassified	20. Number of pages 92
		19. Security Class (This page) Unclassified	21. Price:

