UT TENSIO SIC VIS

DÉPARTEMENT DE GÉNIE ÉLECTRIQUE

SECTION AUTOMATIQUE

Rapport Technique EP-75-R-32

ERROR CONTROL TECHNIQUES FOR DATA TRANSMISSION OVER SATELLITE CHANNELS

Jean Conan, David Haccoun, Hoang Hai Hoc

Département de Génie Electrique

Novembre 1975

Ecole Polytechnique de Montréal

Campus de l'Université
de Montréal
Case postale 6079
Succursale 'A'
Montréal, Québec
H3C 3A7

Dav

Rapport Technique EP-75-R-32

ERROR CONTROL TECHNIQUES FOR DATA TRANSMISSION OVER SATELLITE CHANNELS

Jean Conan, David Haccoun, Hoang Hai Hoc

Département de Génie Electrique

Novembre 1975

ERROR CONTROL TECHNIQUES FOR DATA TRANSMISSION

OVER SATELLITE CHANNELS

Jean Conan, David Haccoun*, Hoang Hai Hoc**

Département de Génie Electrique

RAPPORT TECHNIQUE

EP-75-R-32

Ecole Polytechnique
Université de Montréal

* Subventionné par CNRC, Octroi N$^{\underline{o}}$ A-9336

** Subventionné par CNRC, Octroi N$^{\underline{o}}$ A-8816

TABLE OF CONTENTS

PAGE

PAGE

ERROR CONTROL TECHNIQUES FOR DATA TRANSMISSION OVER SATELLITE CHANNELS

I. INTRODUCTION

In a digital communication system, information or data originated at the source must be transmitted to a distant user through a noisy channel. Because of the channel noise, the transmitted signals do not arrive at the receiver exactly as transmitted and hence errors are made in conveying the source message to the user. The performance for a digital communication system is the probability of error stated in terms of the probability of message, word, or bit error. The choice of the most appropriate unit will depend mainly on the application of the system. Whenever the data is organized in blocks, then a good measure for the system's performance will be the probability of block error; on the other hand, if the information to be sent is a continuous stream of data, then the probability of bit error might be the appropriate measure of performance. In certain applications such as computer-to-computer communication, a repeat for the erroneous block of information is requested through a feedback channel whenever an error is detected. For these systems, the performance criterion is the probability of undetected block error which may be required to be less than 10^{-10}.

Additional requirements on the data itself may concern the tolerable delay in the processing and delivering of the data. Although some systems such as deep space telemetry can tolerate delays of several days in processing the data, many communication systems demand essentially real-time processing. The requirements of data accuracy, data rates and delays together with systems constraints on available transmitting power and bandwidth are key factors to be considered in the implementation of the system. A good design consists often in making sensible trade offs between complexity and performance in the most economical manner.

In this report we consider the problem of error control for data transmission through a geostationary satellite link in a TDMA transmission mode. A satellite link possesses properties different from terrestrial links with regard to propagation delay and noise distribution. A satellite data channel may be modelled as a discrete memoryless gaussian noise channel with a propagation delay exceeding 250 msec. With a binary symmetric channel model, the transition probability (or bit error probability) p is typically smaller than 10^{-2}.

Error control techniques are usually divided into three types, namely, forward error control (FEC) schemes, error detection with retransmission (referred to as ARQ or automatic repeat request), and hybrid systems that employ both FEC and ARQ techniques. After introducing the FEC and ARQ techniques in general, the applicability of these systems to a satellite channel will be considered in some details in section II and III. Finally a new hybrid scheme employing an outer error detection system together with an inner system using convolutional coding and Viterbi decoding is presented in Section IV, and the performance of this system is numerically compared to that of FEC and ARQ schemes alone.

I.1 Forward-Acting Error Control Systems

In a digital communication system employing FEC technique - See Figure (1.1), the data source generates binary information symbols at the rate R_s bits/sec. These information symbols are encoded for the purpose of error protection, and the encoder output is a binary sequence of rate R_n symbols/sec. The code rate R in bits/symbol is then given by

$$R = R_s/R_n \tag{1.1}$$

since $R < 1$, $R_n > R_s$ and hence the transmission symbol speed is larger than the data speed delivered by the source. Equivalently, the introduction of error control coding requires a bandwidth expansion since a wider bandwidth is required to transmit R_n symbols/sec than to transmit R_s symbols/sec.

The encoded sequence is modulated and transmitted over the noisy channel. At the receiving end, demodulation and decoding are performed and the data sink receives the binary symbols at the rate R_s bits/sec. The purpose of the decoding operation is the removal of any channel-introduced errors. The advantages of an FEC system are:

i) No reverse channel is required

ii) A constant information throughput efficiency* is attained

iii) A constant over-all delay is obtained (whenever the decoder operates with a constant decoding lag)

* The throughput efficiency is defined as the ratio of the number of information symbols delivered to the total number of symbols transmitted. Because of the principle of operation of FEC systems, the throughput efficiency remains constant and equal to the code rate R.

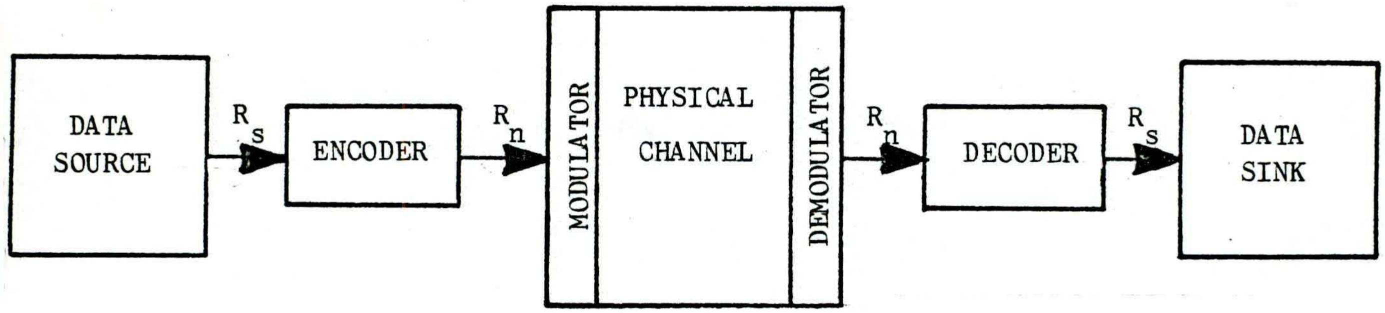


Figure (1.1): Block diagram of a digital communication system employing forward error correction.

Whenever a reverse channel is not available or undesirable, FEC must be used. Although readily available in terrestrial systems, a feedback channel may be a costly proposition for satellite links.

Constant delay may be an important factor for proper operation of terminal equipments. Some terminal equipments even derive timing and synchronization from the input data sequence. Then an occasional large gap in the received data sequence may result in a synchronization loss. In this case, frequent timing and resynchronization information must be provided, resulting in a lower throughput efficiency. Variations in decoding delays, which can occur for some classes of decoding algorithms, may be eliminated by employing a buffer at the decoder output and storing each decoded symbol for some fixed length of time. However, depending on the variations in decoding delay this buffer may overflow, resulting in a complete communication breakdown.

The disadvantages of FEC systems include

i) A relatively moderate throughput efficiency is obtained; this parameter decreases substantially when powerful codes must be used.

ii) The selection of an appropriate error-correcting code and its decoding algorithm might be a difficult task. (Especially when a high reliability level for the transmitted data is required)

iii) The reliability of the received data is very sensitive to any degradation in the channel transmission conditions.

In order to make an intelligent selection of the coding and decoding schemes, a detailed knowledge of the error statistics on the channel is required. Most channels exhibit within a block length (block coding) or constraint length (convolutional coding) a mixture of independent error patterns and bursty behaviour. For these channels the selection of a code is a difficult task. A code suitable for independent errors will fail for long bursts of errors, and a code designed for a bursty channel does not work properly when random errors occur between bursts. Therefore FEC for compound channels (such as the terrestrial links) will result in occasional decoding failures (undetected and/or detected errors). The data user must, of course, tolerate these failures. In order to reduce the number of uncorrectible error patterns, the amount of redundancy in the code must be increased. Naturally this lowers the throughput, increases the required bandwidth and also complicates the decoding procedure since the code structure must be more complex.

Channel degradation increases the undetected or detected errors in the decoder output, because it increases the occurence of uncorrectible error patterns. During periods of high channel noise, an FEC system will deliver unreliable or garbled data to the user.

Another point of interest is the cost of an FEC error control system. This cost is related to the code rate R. Error correction requires more redundant symbols than error detection, and the greater the redundancy, the higher are the encoder and decoder costs. Furthermore the longer the constraint length or block length of the code, the larger will be the amount of required decoder storage.

I.2 ERROR DETECTION AND RETRANSMISSION (ARQ) SYSTEMS

A digital communication system employing ARQ is shown Fig. (1.2). In these systems, block coding is employed, and a sufficient number of redun-

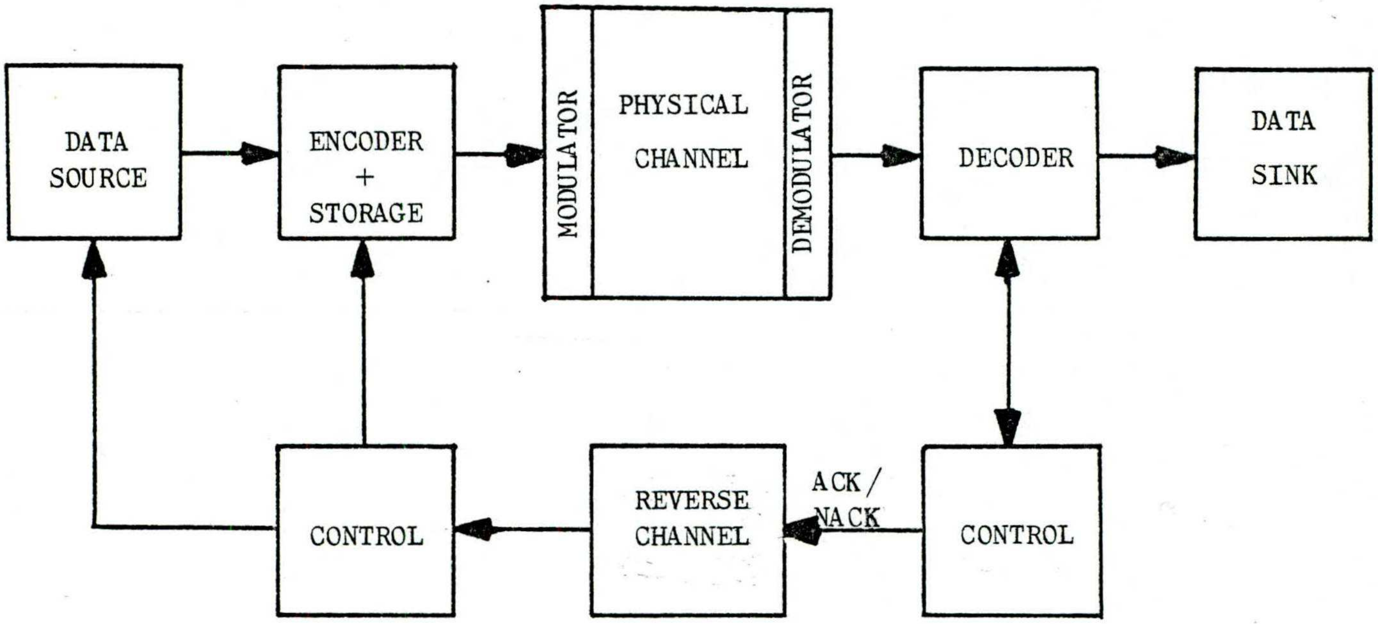


Figure (1.2): DATA COMMUNICATION SYSTEM USING ARQ dant symbols is used in order to achieve the required error detection capability. Basically, no error correction is performed by the decoder, but whenever an error is detected in a block, a retransmission of that block is requested through a reverse channel. A block is accepted by the user only after it appears to be error free. Clearly then, the important measures of performance are: the undetected error probability P_u which is typically very small (($\lll 10^{-10}$), and the throughput efficiency of the system.

The principal advantages of an ARQ error control are:

i) Low undetected error rate

ii) Effectiveness on most real channels

iii) Moderate decoder cost and complexity

The low undetected error probability is the consequence of the power of error detecting codes. Moreover, all information blocks delivered to the user can be accepted with equal confidence, even during periods of poor channel quality.

The information throughput of an ARQ system depends greatly on the number of requested retransmissions, thus on channel quality, and also on the error detecting code.

The selection of a code is less difficult for an ARQ system than it is for an FEC system. This is because a code used for error detection is less sensitive to diverse channel error patterns than a code used for error correction. Since the code can detect the vast majority of all error patterns, it does not matter very much how errors occur on the channel. Consequently the use of ARQ is effective on most channels. It may be noted that this property of ARQ systems is not shared by FEC systems.

The cost of an ARQ systems is substantially smaller than that of an FEC system. This is due to the fact that the error pattern need not be determined as in FEC systems: only the presence of an error pattern has to be detected, i.e. a far simpler decoding operation. Storage in an ARQ system is required at the transmitter in order to repeat previously transmitted blocks upon request from the receiver.

The disadvantages of an ARQ system include the following:

i) A return channel is required

ii) The system operates with a variable decoding delay

iii) The data source must be interruptible

One of the potentially most serious drawbacks of ARQ systems is the requirement for a reverse channel. For some communication systems such as a telephone data transmission system, a reverse channel is readily available. However for some other systems the cost of providing a reverse channel may be prohibitive.

The occurence of retransmissions induces a decoding delay, which is measured by the time between the first arrival of a block at the decoder and its delivery to the user. Naturally a multiple transmission of the same block increases the decoding delay and hence reduces the throughput efficiency. In addition to the decoding lag, the round-trip propagation delay way be important and lowers further the information throughput. For a satellite channel, the round-trip delay is of the order of 500 msec, and as shown in section III, the use of ARQ systems on this channel will drastically reduce the throughput.

Finally during intervals of retransmission the data source must be interrupted in order to avoid an accumulation of information bits at the encoder. Therefore the source must be controllable, and depending on the nature of the data source, this requirement may represent a problem.

II. CODING FOR THE SATELLITE CHANNEL

II.1 The coding problem and capacity

Communication from one ground terminal to another ground terminal via satellite has assumed in the last few years a great importance for both commercial and military uses.

The two principal characteristics of the satellite channel are: 1) the primary additive disturbance can be accurately modelled by Gaussian noise which is white enough to be essentially independent from one bit interval to the next, and 2) For a synchronous geostationary satellite, the delay is large, about 250 msec from one ground terminal to another.

Furthermore, up to the present and foreseeable future, satellite repeaters are more power-limited than bandwidth-limited and hence sufficient bandwidth is available to allow a reasonable bandwidth expansion. In this environment the improvement of the communication efficiency by coding becomes particularly attractive.

A data communication system using a satellite is diagrammed in Fig. (2.1). Assuming the system to transmit R_s information bits per second over the white Gaussian noise channel of noise density N_o, and the received power to be P, then the dimensionless parameter $E_b/N_o = P/N_oR_s$ serves as a figure of merit for different coding and modulation schemes. E_b/N_o is called the signal-to-noise ratio per information bit. The problem then is determining the system that will operate at the lowest E_b/N_o with a given quality.

Following the block diagram of Fig.(2.1), information bits enter the encoder at a rate of R_s bits/sec. The encoder inserts (N-K) redundant bits

[redacted]

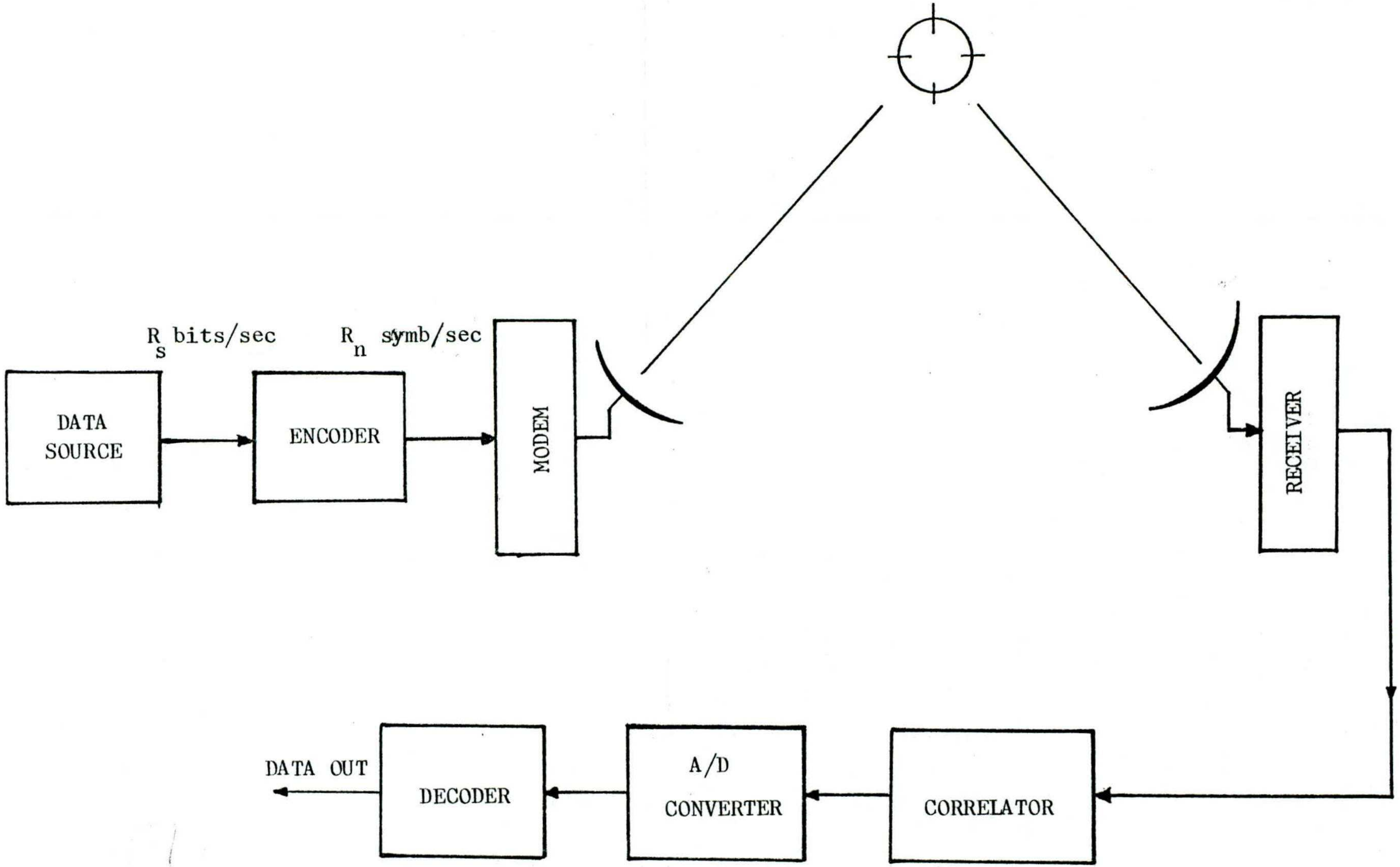


Figure (2.1): Data communication system using a satellite.

for every K information bits, giving a transmission rate of $R_n = R_s N/K$ symbols/sec. At the receiving end, the correlator output for each symbol enters an A/D converter, resulting in either a hard decision (i.e. binary) or a soft decision (i.e. A-ary). The decoder observes these decisions and from the knowledge of the coding scheme used, estimates which information bits were actually sent.

A natural question arises immediately: How much information can be transmitted through this channel, and what are the components of the coder and decoder boxes? The answer to the first part of this question was given by Shannon in 1948 [1]. Shannon showed that within a large class of coding schemes, there existed some schemes that could give arbitrarily low error rates provided the information rate does not exceed a maximum rate called the channel capacity C. Unfortunately Shannon's coding theorems were only existence theorems and gave no clue as to the actual construction of these coding schemes.

For a memoryless Gaussian channel of bandwidth W, it can be shown [2] that the capacity is given by

$$C_n = \frac{1}{2} \log_2 [1 + P/N_o W] \text{ bits/transmitted symbol} \qquad (2.1)$$

or

$$C = W \log_2 [1 + P/N_o W] \text{ bits/sec.} \qquad (2.2)$$

Whenever the information rate R_s is less than C, then there exists some coding-modulation-decoding scheme with a decoded error probability as small as desired. We see that for a fixed power-to-noise ratio P/N_o, more and more efficient communication is possible as the available bandwidth W is increased.

For an infinite-bandwidth gaussian channel the capacity is

$$C_\infty \triangleq \lim_{W \to \infty} C = \frac{P/N_o}{\ln 2} \qquad (2.3)$$

The limiting signal-to-noise ratio becomes

$$\frac{E_b}{N_o} = \frac{P}{R_s N_o} \geq \frac{P}{CN_o} > \frac{P}{C_\infty N_o} = \ln 2 \qquad (2.4)$$

Under no bandwidth limitation, the minimum value of E_b/N_o referred to as E_{bmin}/N_o is called the Shannon limit, and approaches a limit of ln 2 or -1.6dB. Naturally it is desirable to operate with E_b/N_o as close as possible to this limiting value of -1.6dB.

Another useful parameter is the maximum number of bits that be transmitted per hertz of bandwidth. We assume transmitting at capacity and hence

$$\frac{E_b}{N_o} = \frac{P}{N_o C} \qquad (2.5)$$

From Equation (2.2) we then have

$$\frac{C}{W} = \log_2 \left(1 + \frac{P}{N_o W}\right) = \log_2 \left[1 + \frac{P}{N_o C} \frac{C}{W}\right]$$

$$\frac{C}{W} = \log_2 \left(1 + \frac{E_b}{N_o} \cdot \frac{C}{W}\right) \qquad (2.6)$$

A plot of relation (2.6) is given in Figure (2.2). The curve is highly unsymetrical and may be divided into two regions: The bandwidth-limited region for $C/W > 2$ and the power-limited region for $C/W < 2$. It is interesting to note that at a bit per hertz of bandwidth of only $\frac{1}{2}$, we are within 0.8dB of E_{bmin}/N_o (which is reached only for infinite bandwidth).

We now investigate the advantages to be gained by using coding in the power-limited region. Figure(2.3) gives the error probability versus E_b/N_o for no-coding binary schemes. For example for PSK systems, an E_b/N_o of 9.6dB is required to obtain a bit error probability of 10^{-5}. Since E_{bmin}/N_o is -1.6dB,

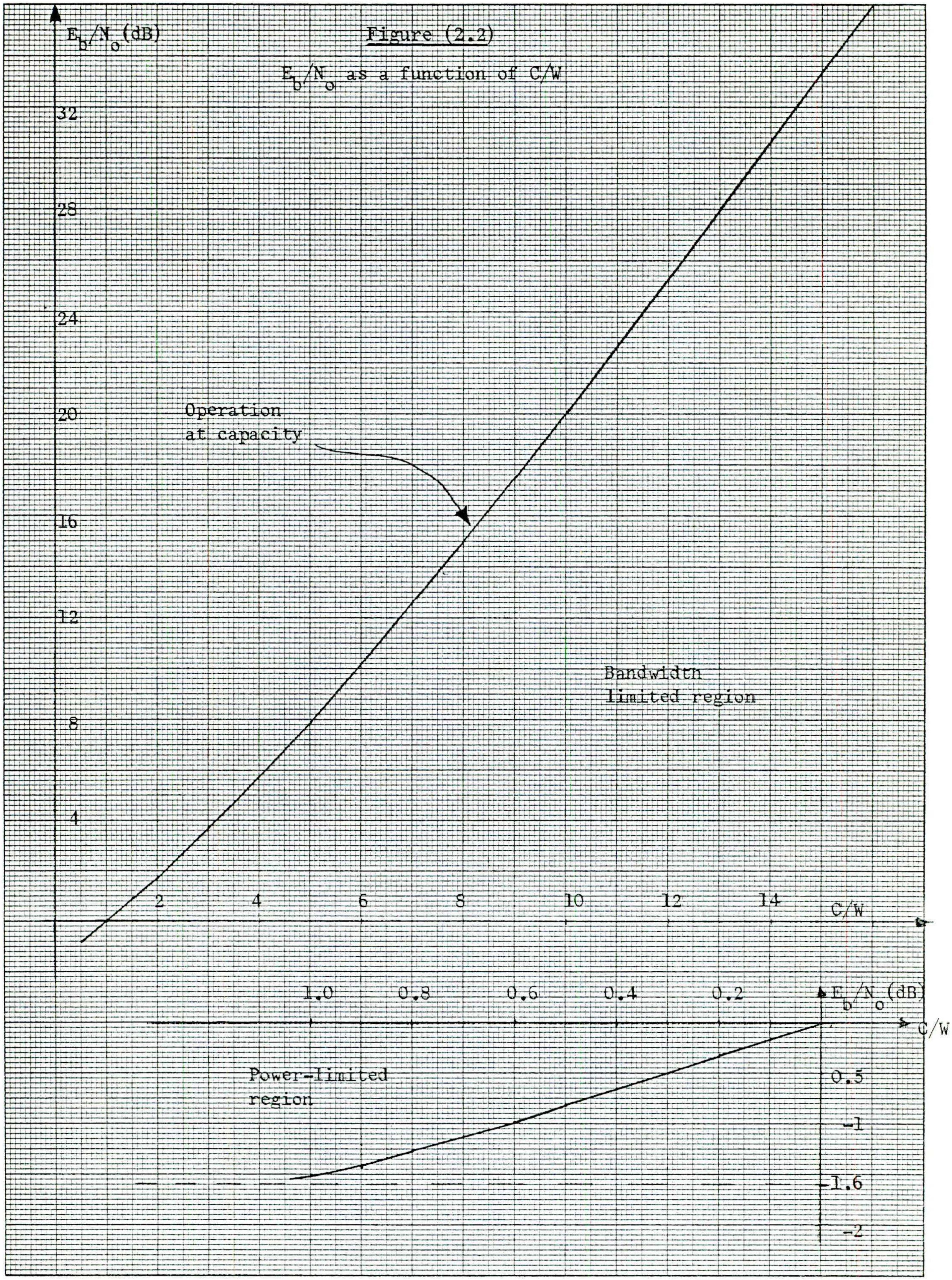


Figure (2.2)

E_b/N_o as a function of C/W

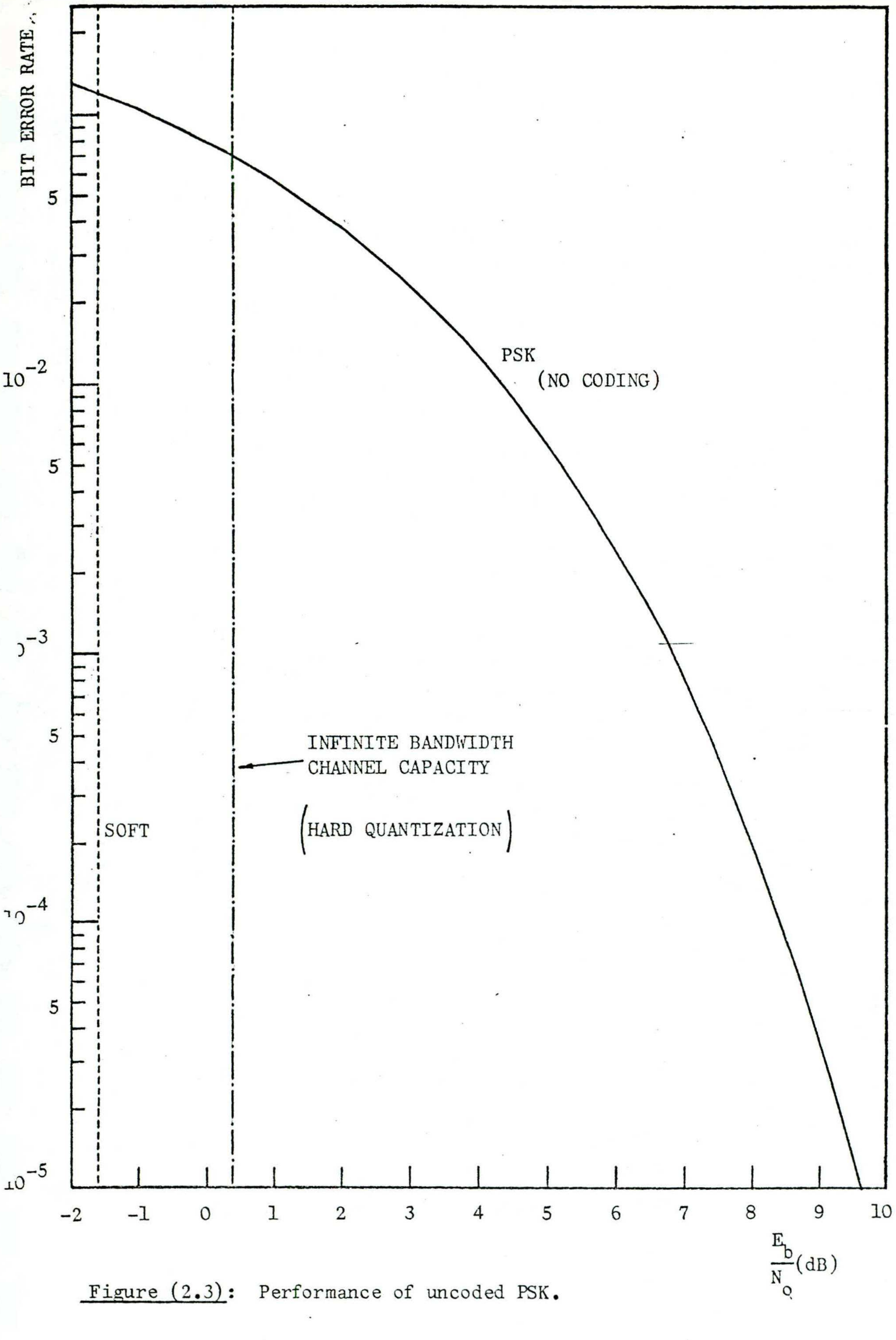


Figure (2.3): Performance of uncoded PSK.

then a potential coding gain of 11.2dB is theoretically possible. This gain is rather attractive in a satellite environment since it may be translated either as a reduced transmitter power or as an increased data rate. Table 2.1 lists the coding gains for different non coded schemes at error rates of 10^{-3} and 10^{-5} [5]

TABLE 2.1

Coding gains E_b/E_{bmin} for several uncoded systems

		E_b/E_{bmin} (dB)	
	M	$P(E) = 10^{-3}$	$P(E) = 10^{-5}$
M Biorthogonal signals	2 32 1024	8.4 6.5 5.1	11.2 8.6 6.6
M Orthogonal signals	2 32 1024	11.4 7.3 5.1	14.2 9.3 6.6
Differential PSK	2	9.5	11.9

Unfortunately coding does not offer much gain in the bandwidth-limited region of Figure (2.2). When power is readily available but bandwidth strictly limited, multilevel signalling is usually utilized. In this scheme, the transmitted symbols may assume A possible levels, leading to a rate of

$$R_A = \log_2 A \quad \text{bits/transmitted symbol} \qquad (2.7)$$

Doubling the number of amplitudes while maintaining the same probability of error (that is the same level separation) requires quadrupling the average power P. As the available power increases and we get deeper in the bandwidth-limited region, capacity without coding can therefore nearly be achieved. In conclusion, coding for a high signal-to-noise ratio gaussian channel is not greatly rewarding. However, it is doubtful that the performance of uncoded systems on real channels approaches the predicted performance because the probability of an atypically large noise level may be substantially larger than that predicted by the gaussian model.

II.2 Decoding techniques for the satellite channel

We have already established that for data transmission, the satellite channel can be accurately modelled as a discrete memoryless gaussian channel. We shall also assume that the channel is power-limited rather than bandwidth-limited. Hence as shown in the previous section, coding offers substantial advantages. In this section we present some coding shemes particularly suitable to the satellite channel. Since the trend for satellite repeaters is toward larger P/No without a corresponding increase in available bandwidth, we should consider only codes which require a modest bandwidth expansion (i.e., less than 3)

For discrete memoryless channels, systems employing convolutional encoding [2] at the input and probabilistic decoding at the output are among the most attractive means of approaching the reliability of communication promised by Shannon's coding theorem. Probabilistic decoding refers to techniques where the decoded message is obtained by probabilistic considerations rather than by a fixed set of algebraic operations. Moreover, no particular algebraic structure is imposed on the code which may be chosen at random. The two principal probabilistic decoding techniques are sequential decoding [2,3] and Viterbi decoding [4]. The applicability of these decoding techniques to the satellite channel has been widely demonstrated [5-6].

A binary convolutional code of rate $R = 1/V$ bit/symbol may be generated by a linear finite-state machine consisting of a K-stage shift register, V modulo-2 adders connected to some of the shift register stages, and a commutator that scans the output of the mod-2 adders. The pattern of connections specifies the code. The machine is called a convolutional code and is sketched in Fig(2.4) for $K = 3$, $V = 3$. Information bits are shifted in the shift register one bit at a time. After each shift, the V mod-2 adders are sampled sequentially, yielding the code symbols. As an input may take 2 values, then for each shift there are 2 V-symbol alternatives (or branches) for the encoder output. This suggests representing the output of a convolutional code by a tree, with 2 branches per node, and V coded symbols per branch. A path in the tree is traced from the root node out, according to the input sequence that entered the encoder. The constraint length of the code is K, since K represents the number of shifts over which a single information bit can influence the encoder output.

The state of the encoder is the contents of the first (K-1) shift register stages, and the encoder state together with the next input uniquely determines the V coded symbols. Hence there are 2^{K-1} possible states. Associating each node of the tree to an encoder state, it follows that beyond the (K-1) th information bit, several nodes correspond to the same encoder state, and the tree diagram collapses in a trellis with exactly 2^{K-1} states. The tree and trellis structure corresponding to the example of Fig.(2.4) are given in Fig.(2.5).

Decoding can be seen as the operation for determining the most likely information sequence, given the received sequence. For convolutional codes, the information sequence is represented by a particular path through the tree or trellis diagram. Sequential decoding and Viterbi decoding are two powerful graph searching techniques. The Viterbi algorithm uses the trellis level, whereas a sequential decoder uses the tree structure of the code and searches only a part of the tree, following only one among the paths that appear to be most likely. As a consequence, the computational effort is constant but large for Viterbi decoding (it increases exponentially with K), whereas it is on the average typically very small but variable for sequential decoding.

The error probability has been bounded for both sequential decoding [2,3] and Viterbi decoding [4]. It decreases exponentially with the constraint length of the code. Although the error performance of sequential decoding is lower bounded by that of Viterbi decoding, asymptotically, as $K \to \infty$ both techniques yield the same error performance.

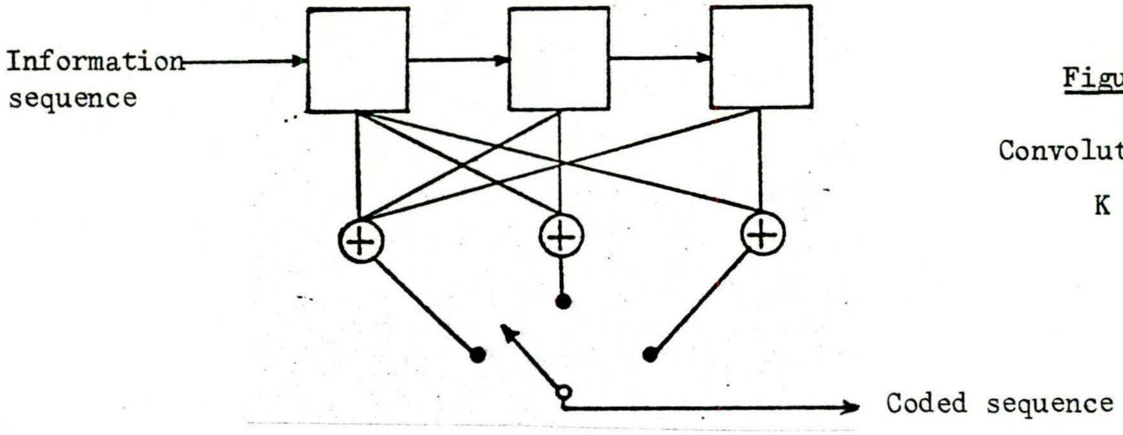


Figure (2.4)

Convolutional coder

K = V = 3

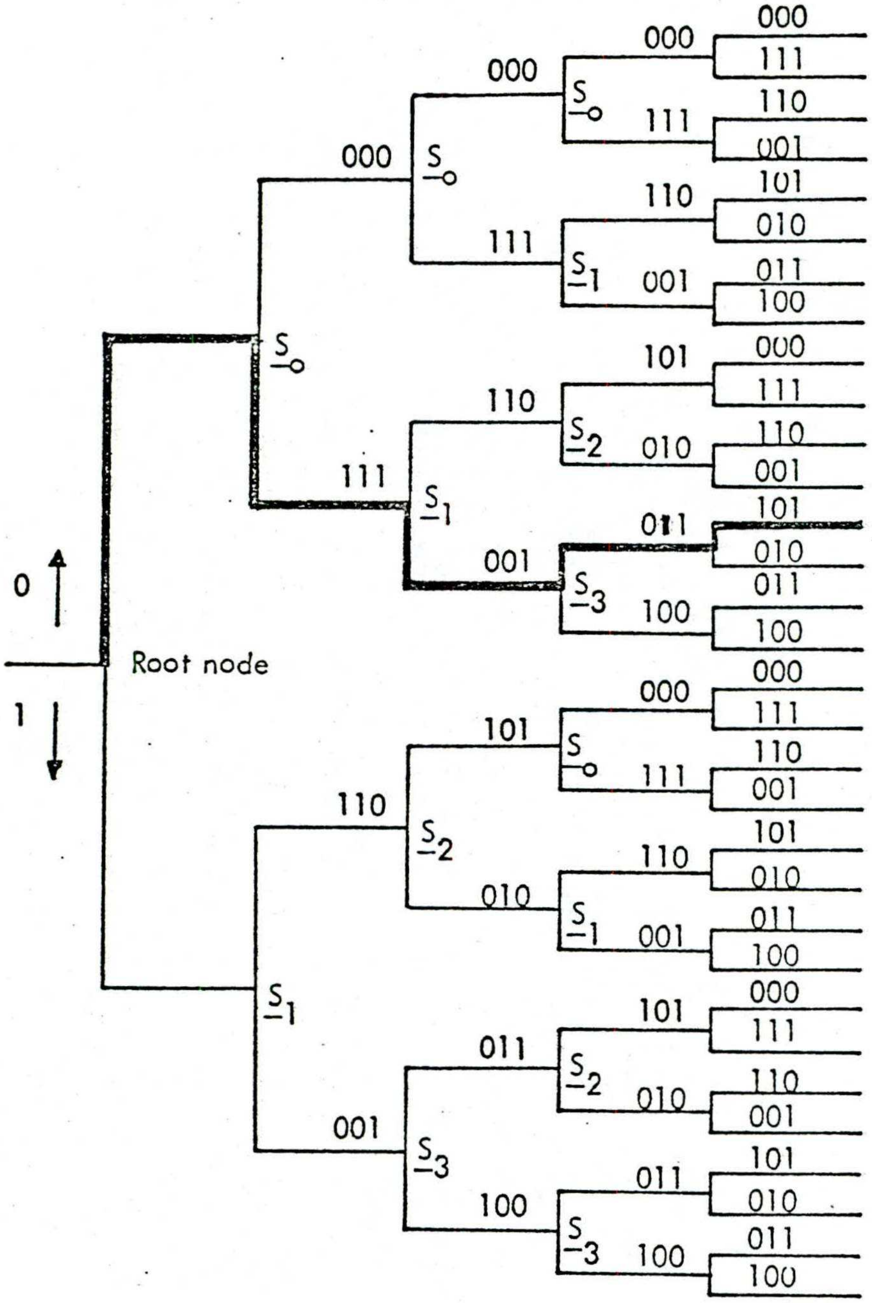


Figure (2.5a): Tree representation of coder of Fig. (2.4) (The heavy line corresponds to the information sequence 01100...)

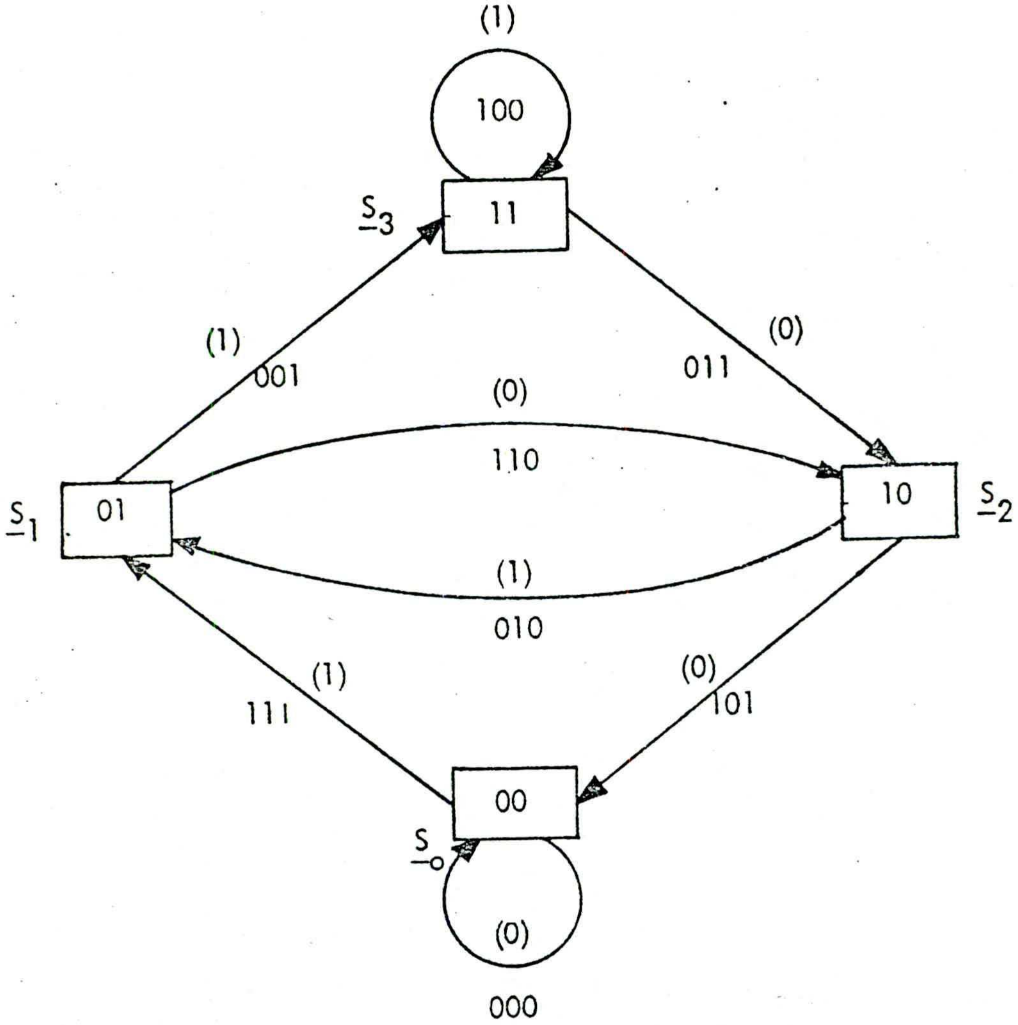


Figure (2.5b) State diagram for encoder of Fig. (2.4)

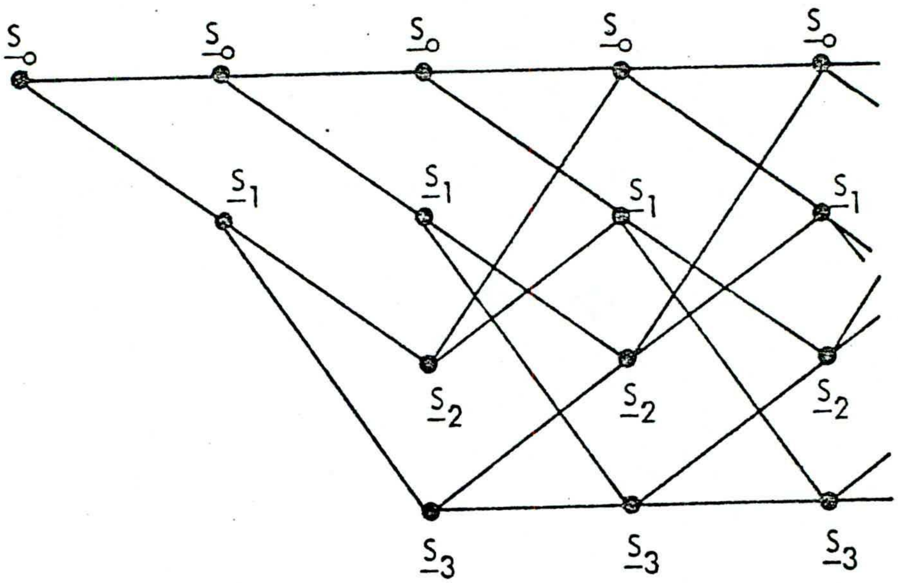


Figure (2.5c) Trellis representation of encoder of Fig. (2.4)

Over memoryless channels there exists also a number of other good decoding techniques. For convolutional codes we have the Massey's threshold decoding [7] scheme, and for block codes efficient algorithms are known for the Golay and the BCH [3] codes. These algorithms are less powerful than Viterbi or Sequential decoding, and in general convolutional codes outperform block codes of the same order of decoder complexity.

In order to gain some insight into the performance of error-correcting codes on random error channels the performance of several codes are shown in Fig.(2.6) and Fig.(2.7). From these figures we observe that the Viterbi decoder provides superior performance to that of both block codes and the threshold decoder. Furthermore by using a 3-bit quantization on the channel output rather than the hard decision of the binary symmetric channel, the performance of the Viterbi decoder may be further improved by 2 dB.

Another point of interest is the steep slope of the sequential decoder curve. A relatively small increase in E_b/N_o results in a very large decrease of the bit error probability (roughly an order of magnitude for each 0.2 dB increase in E_b/N_o).

Table 2.2 illustrates the effectiveness of communication systems using the above coding techniques, relative to no-coding systems with PSK signalling, for a bit error probability fixed at 10^{-5}. The improvement is again measured by difference between the required E_b/N_o values.

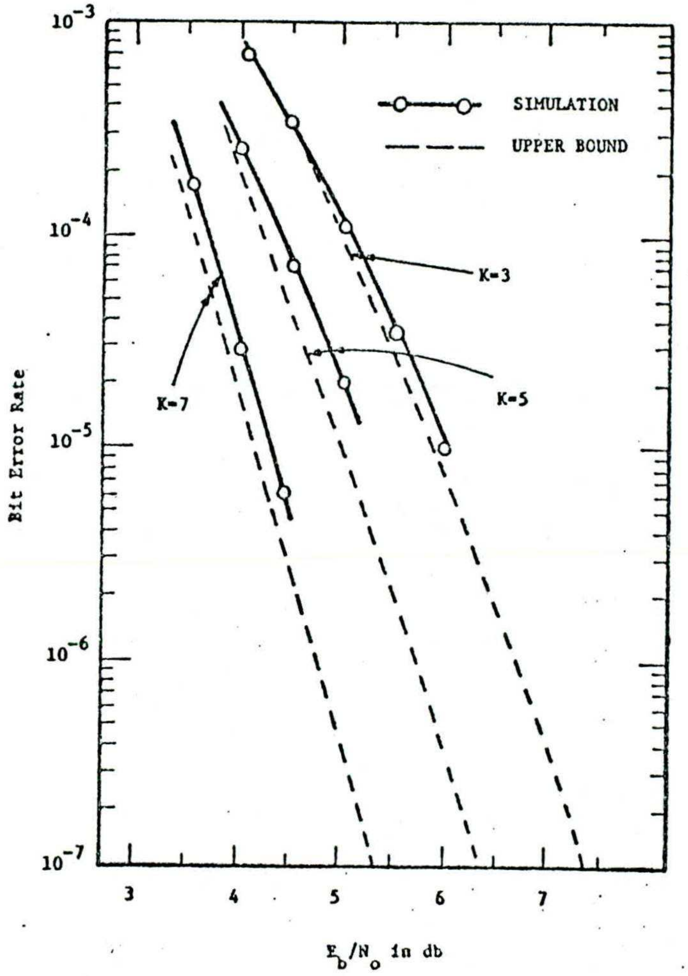


Fig. 2.6 - Bit error rate vs. E_b/N_o for rate 1/2 Viterbi decoding. Eight level quantized simulations with 32 bit paths, and the infinitely finely quantized transfer function bound, K=3, 5, 7.

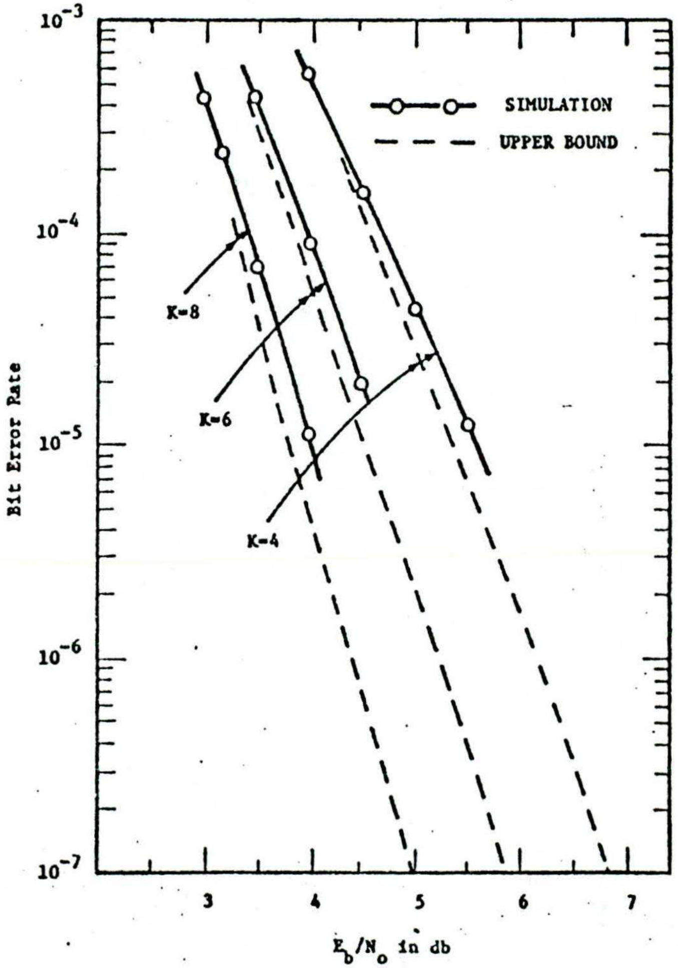


Fig. 2.6 - Bit error rate vs. E_b/N_o for rate 1/2 Viterbi decoding. Eight level quantized simulations with 32 bit paths, and the infinitely finely quantized transfer function bound, K=4, 6, 8.

Figure (2.7): Performance of several coding schemes

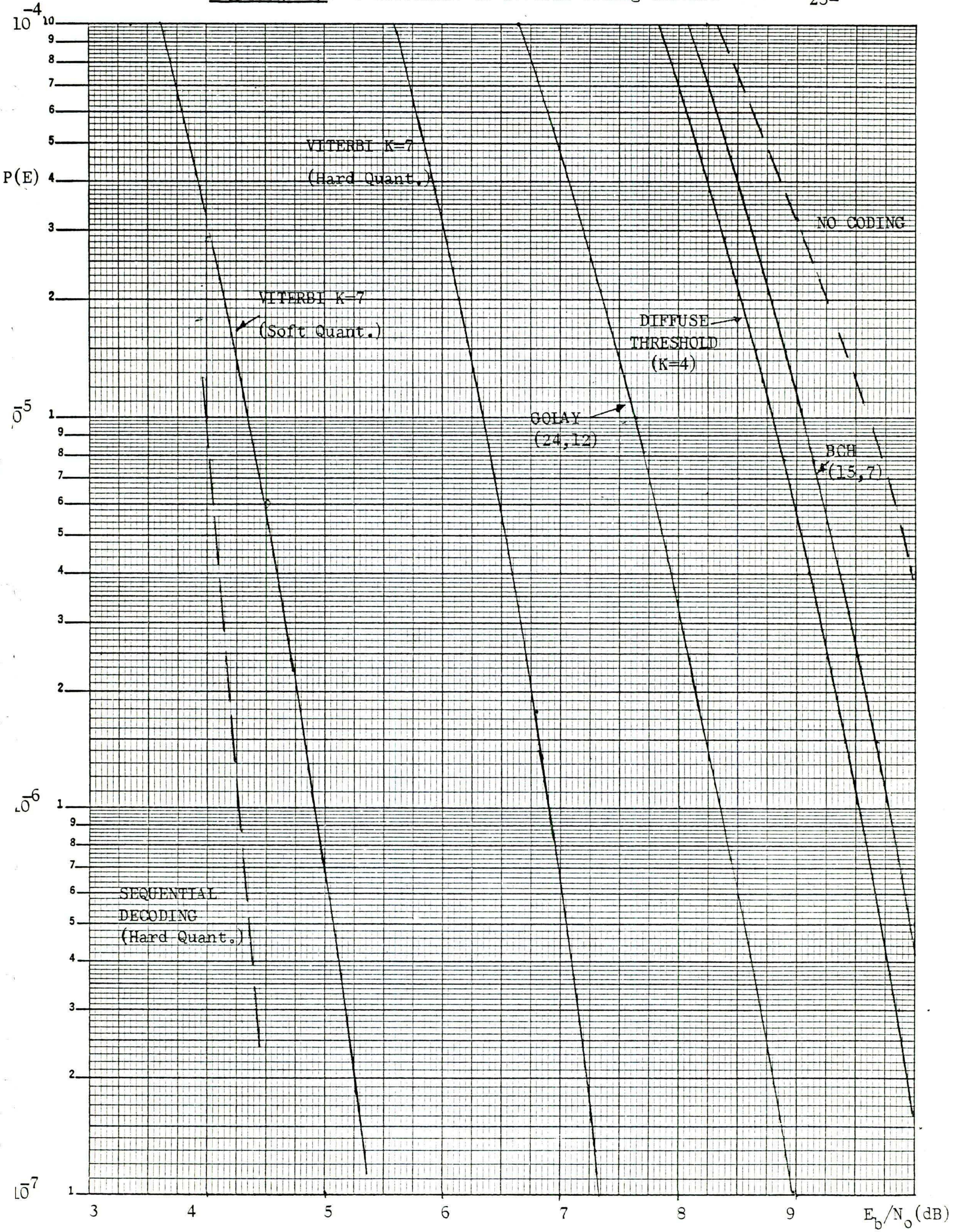


Table 2.2 Coding gain for several coded systems

SYSTEM	Required E_b/N_o (dB) for $P(\epsilon) = 10^{-5}$	Gain in dB over non-coded system (PSK)
No coding (PSK)	9.6	—
(15,7) BCH	8.65	1
Threshold	8.5	1.1
Golay (24,12)	7.5	2.1
Viterbi (K = 7,hard)	6.4	3.2
Viterbi (K = 7,soft)	4.4	5.2
Sequential (hard)	4.6	5

For example, the sequential decoder requires on E_b/N_o equal to 4.6 db whereas PSK system requires 9.6 dB. This 5 dB gain afforded by the sequential decoder can be translated as either a 5 dB reduction in the transmitter power at the same data rate as the PSK system, or as an increased data rate namely $\sqrt{10} = 3.2$ times the uncoded data rate.

For lower error probability, the gain of the coded systems relative to uncoded system is even larger. It must be stressed that this decoding gain is obtained at the expense of an increase in channel bandwidth. The bandwidth expansion is the reciprocal of the code rate, and is 2 for our example. Therefore the transmitted symbol rate is twice that of the data rate. This is a moderate price to pay and is well within the means of the available bandwidth of the satellite.

Finally it should be pointed out that, although the performance curves of Fig.(2.6) and Fig.(2.7) were obtained by computer simulation, the hardware versions of Viterbi and sequential decoders operate fairly close to the theoretical curves.

II.3 Comparison between sequential and Viterbi decoding

Both sequential and Viterbi decoding offer pratical alternatives to the satellite channel. Both are capable of very-high speed operation and have relative merits that are noted below: (Performance curves for these two decoding techniques are given in Fig.(2.8), and the code parameters are representative of pratical up to date hardware limitations).

First we observe that with soft 3-bit quantization, the Viterbi decoder affords a gain of 2 dB. This gain generally applies to any decoding scheme which can accept multiple level inputs. Soft quantization is easily implemented in a Viterbi decoder. However, for sequential decoders, a 3-bit quantization of the channel output requires a buffer three times as large as for the BSC. Therefore the 2 dB gain is usually denied to a sequential decoder.

Next, the sequential decoder curve is much steeper than the Viterbi decoder curves because of the difference in the constraint length used. This implies that sequential decoding is more sensitive to small signal variations.

Another point of comparison is the data rate capability of each technique. Because of the constancy of its computation level, a Viterbi decoder is insensitive to data rate, up to the limiting speed of its digital logic. In sequential decoding, the computational effort is a random variable and depends greatly on the channel. Since the computations in the decoder are usually

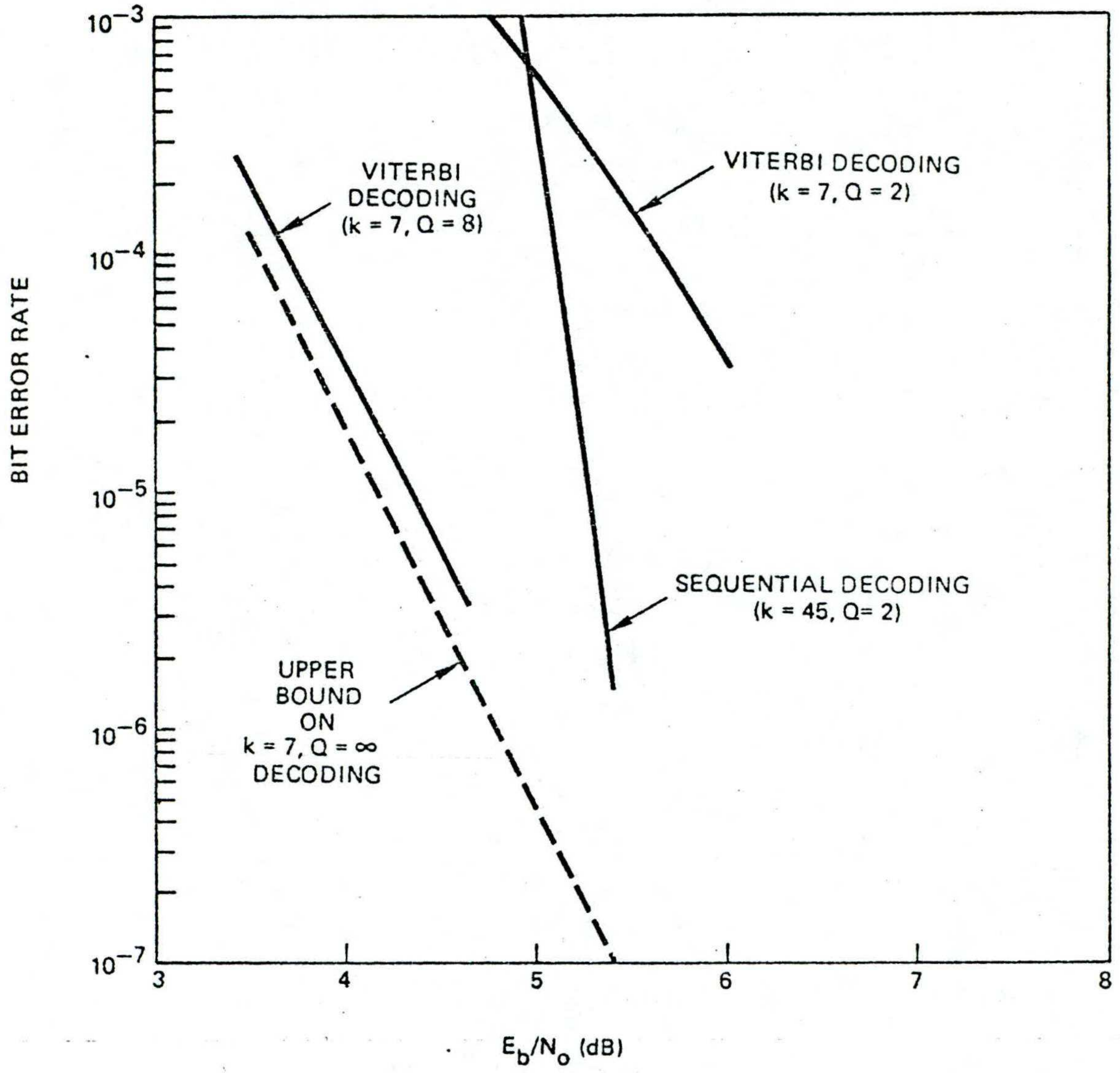


Figure 2.8: Examples of Viterbi and Sequential Decoding.

made at a fixed rate, for a lower data rate the decoder can make more computations per data bit, and hence the number of undetected errors may be reduced. Therefore the sequential decoder error performance is somewhat sensible to the data rate.

In addition to data and error rates, decoder delay is an important constraint. The Viterbi decoding lag (refered to as memory) is constant and typically from 5 to 10 constraint lengths long. On the other hand sequential decoders tend to require long buffers to smooth out the variations in computational load. Consequently the sequential decoder delay varies from a few hundreds to several thousands bits.

For time-division multiplexed systems (TDMA), the ground stations' transmitted information is separated in discrete time slots, with each station assigned a particular time slot or block of time slots. Since the different transmission channels are orthogonal with respect to time, the intermodulation products are eliminated with TDMA. This is a distinct advantage since intermodulation products essentially lower the useful signal power. Now since each station transmits a burst of data during its time slot, termination of encoding during this time slot is desirable. This can be achieved by transmitting a known sequence of length K-1 bits (the tail of the message). Since K is typically larger for sequential decoding than for Viterbi decoding, the system efficiency degradation is less significant with Viterbi decoding.

Finally for a given error rate, the patterns of errors caused by Viterbi and sequential decoders differ. The errors for Viterbi decoders usually occur in short bursts of 3 or 4 constraint lengths, but for sequential decoders,the bursts are usually the result of buffer overflows and span several hundred bits. Thus systems which are sensible to long bursts should consider Viterbi decoding.

In summary, the use of convolutional encoding in conjunction with Viterbi and sequential decoding on Satellite channels has been presented. The choice of the particular decoding technique will depend primarily on the system requirements. For an error probability of 10^{-5} and burst rates of 10 M bits/sec, the available E_b/N_o for the actual ground stations is 4.6 dB. Hence a Viterbi decoder with K = 7 is needed.

III - AUTOMATIC REPEAT REQUEST (ARQ) SYSTEMS

III.1 Introduction

Since the object of this report is restricted to the study of error control techniques for earth station to earth station satellite communication systems, the consideration of Automatic Repeat Request (ARQ) systems will be limited to point to point binary data communication systems. The general scheme associated with error control techniques known as ARQ is represented on Figure (3.1)

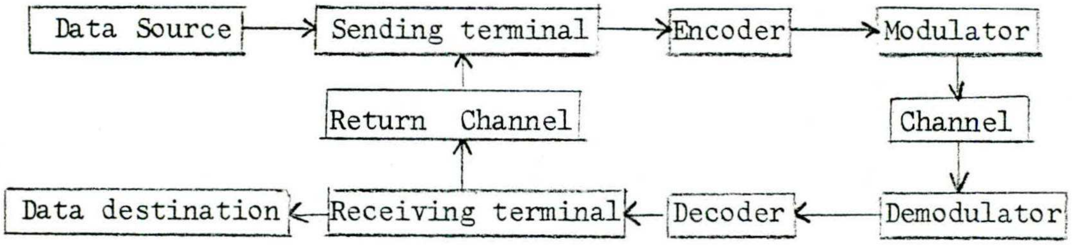


Figure 3.1 Block diagram of an ARQ system

In an ARQ system, data are delivered from a controllable source to a sending terminal. The sending terminal arranges data into blocks, buffers the blocks, attaches control and synchronization bits, and more generally controls the line. The data blocks are then processed by an encoder which adds redundancy by inserting parity check symbols, each of which is determined from a subset of data/control bits. Each encoded block contains K information bits and (N - K) parity binary symbols, and the code is said to be an (N, K) block code with dimensionless coding rate $R = \frac{K}{N}$. The encoded block is then transferred to the modulator and transmitted over the channel. At the receiving end the block is demodulated and passed on to the receiving terminal through the decoder, which computes the parity symbols from the received data symbols and compares them with the received parity symbols.

If there are no discrepancies, the block is delivered to the data destination, and the receiving terminal notifies the sending terminal, through a suitable return channel, that the block has been correctly received. If discrepancies exist, the sending terminal is so notified and the block is retransmitted. With this system, erroneous data are delivered to the destination only if the decoder fails to detect the presence of errors.

From a purely technical point of view, the two major criteria which are used to evaluate

1) probability of having erroneous data delivered to the destination, i.e. the probability P_u of undetected block error.
2) system throughput, i.e. the ratio between the effective rate of information in bits/sec transferred from source to destination and the modem signalling speed in symbols/sec.

In the performance evaluation of ARQ systems, the following key factors are to be accounted for:

a) channel error characteristics.
b) system parameters such as propagation delay, modem bit rate, turnaround time (for half duplex mode).
c) characteristics of the error detecting code such as error detection capability and coding rate.
d) specific techniques used to carry out the retransmission such as stop-and-wait ARQ, continuous ARQ ...

Since the most striking characteristics of ARQ systems are their inherent reliability (very low probability of undetected block error) and their relative insensitivity of this reliability to conditions on the channel; the key figure of merit in the evaluation of ARQ systems is the transmission efficiency or system throughput.

In the following subsections, the dependency of the channel throughput efficiency upon system parameters such as channel error statistics, coding rate, signalling speed, block size and the actual retransmission policy will be considered in details. An attempt will be made to include the code as an integral part of the design by taking into account the constraint which can be imposed on the value of the undetected error probability which is required.

III.2 Stop-and-Wait ARQ

The simplest and most widely used detection-retransmission scheme is the stop-and-wait ARQ system. In this system, after sending a block, the transmitting terminal waits for a positive or negative acknowledgment from the receiving terminal before sending another block or retransmitting the same block.

Assuming that the effect of acknowledgment errors is negligible, the system throughput is then given by

$$\eta = \frac{K(1 - P_B)}{(N + TR_{un})} \quad \text{bits/symbol} \qquad (3.1)$$

P_B = detectable block error probability

N = block size

K = number of information symbols per block

R_{un} = signalling speed on the channel in symbols/sec

T = round trip delay + turnaround times (half duplex mode) + transmission time of the ACK/NACK message

In order to derive expression (3.1), we remark that the probability of having a block transmitted exactly n times is P_B^{n-1} $(1 - P_B)$. Since the time period required to transmit n times a block of size N is equal to $n(T + N/R_{un})$, the average time period required to transfer a block of size N from the source to the destination is then

$$\bar{T} = (1 - P_B)(T + N/R_{un}) \sum_{n=1}^{+\infty} nP_B^{n-1} = \frac{(1 - P_B)(N + TR_{un})}{R_{un}} \frac{d}{dP_B} \mathop{}_{n=0}^{\infty} P_B^{\,n}$$

$$= \frac{(1 - P_B)(N + TR_{un})}{R_{un}} \frac{d}{dP_B} \left[\frac{1}{(1 - P_B)}\right]$$

$$\rightarrow \quad \bar{T} = \frac{N + TR_{un}}{R_{un}(1 - P_B)}$$

Since K bits are transmitted on the average during the time $\bar{T}$. The average data rate is then $\bar{R} = \frac{K}{\bar{T}}$ bits/sec and the definition of the throughput efficiency as the ratio of the average data rate by the signalling speed yields expression (3.1).

In order to explore the dependency between the system throughput η and the channel errors and system parameters; we assume that the channel is binary symmetrical with crossover probability p, and that the rate of the error detection code is constant and independent of the block size N. Thus assuming all errors are detected we have

$$P_B \simeq 1 - (1 - p)^N \simeq 1 - e^{-Np} \qquad (3.2)$$

and hence (3.1) can be written as $\eta = \frac{RNe\ -Np}{N + TR_{un}}$ where $R = K/N$ is constant $\simeq 1$

The block size N should then be chosen to maximize the throughput η. One then obtains, defining $C \triangleq R_{un}T$ as the delay counted in symbols

$$N_{opt} = \frac{C}{2} \left(\sqrt{1 + \frac{4}{Cp}} - 1\right) \qquad (3.3)$$

$$\eta_{opt} = R \frac{N_{opt} e^{-N_{opt}p}}{N_{opt} + C} \qquad (3.4)$$

In the case of satellite links with round-trip propagation delay estimated at 512 msec, the variations of N_{opt} and η_{opt} as functions of modem signalling rate R_{un} are illustrated in Figures (3.2a) and (3.2b) using as parameter the bit error rate (BER)p.

It is worth noting that the optimal block size increases approximately as the square root of the modem signalling rate. For high speed transmission (50 K Baud for example) optimal block size may be quite large (a value of 76,800 bits is associated to a bit error rate of 10^{-4}). Even in applications where such large blocks are pratical, they may give rise to the apparent difficulty of determining error detection codes with acceptable detection capability. However, the difficulty does not actually exist since the block size N refers only to the number of symbols transfered as a transmission unit. Such a transmission unit could consist of a sequence of several error detection encoded sub-blocks. N does not necessarily have any relation to the block size of the code; it is merely the length of a message which is eigher accepted as a whole or generates an ARQ request. Besides applications in which it is impractical to use blocks sufficiently long because of restrictions imposed by the data, it is always possible to design acceptable error detection codes that assure a high value ($\simeq 1$) for the coding rate R. Hence, it might not be too unrealistic to assume that the code rate is constant and independent of the block size N. More insight into this important and yet unsolved problem will be given in section III.3.

Altghough stop-and-wait ARQ systems are widely used because of the simplicity of operation, and the minimum buffer requirements. They are inherently inefficient due to the idle time spent waiting for acknowledgment for each transmitted block. This inefficiency becomes unacceptable on systems which employ high speed modem and systems involving a high propagation delay such as satellite links. In the next section, we consider a relatively simple method of circumventing this difficulty.

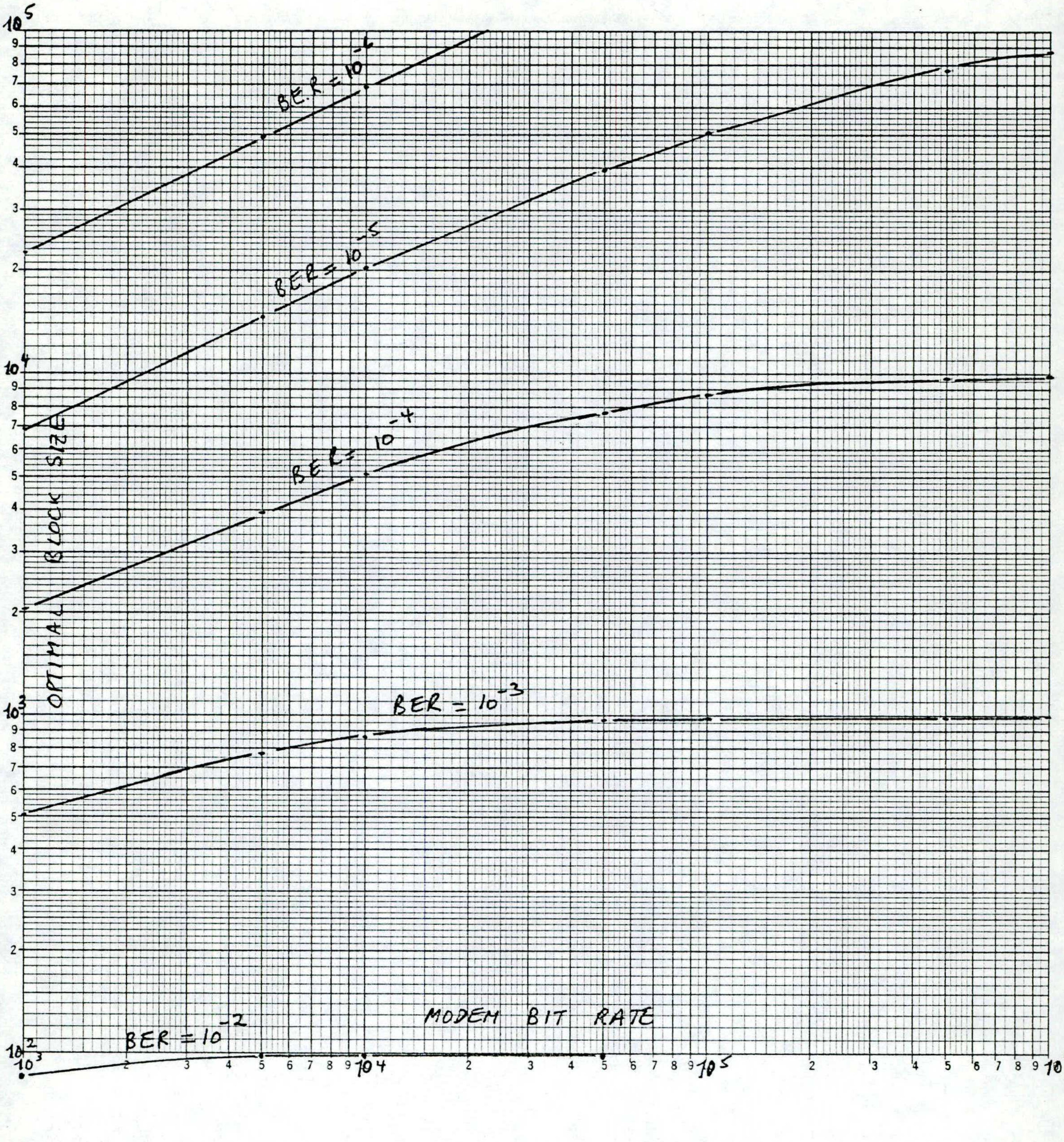


Figure 3.2.a Optimal Block Size versus Modem Bit Rate for Stop-and-Wait ARQ Systems

III.3 Continuous ARQ

The continuous system transmits blocks of N bits consecutively, without any delay between blocks, as long as positive acknowledgments are received at the sending terminal. Whenever a negative acknowledgment is received, the sending terminal essentially "backs-up" to the erroneous block and retransmits that block and all subsequent blocks (in order to preserve the natural ordering of the blocks).

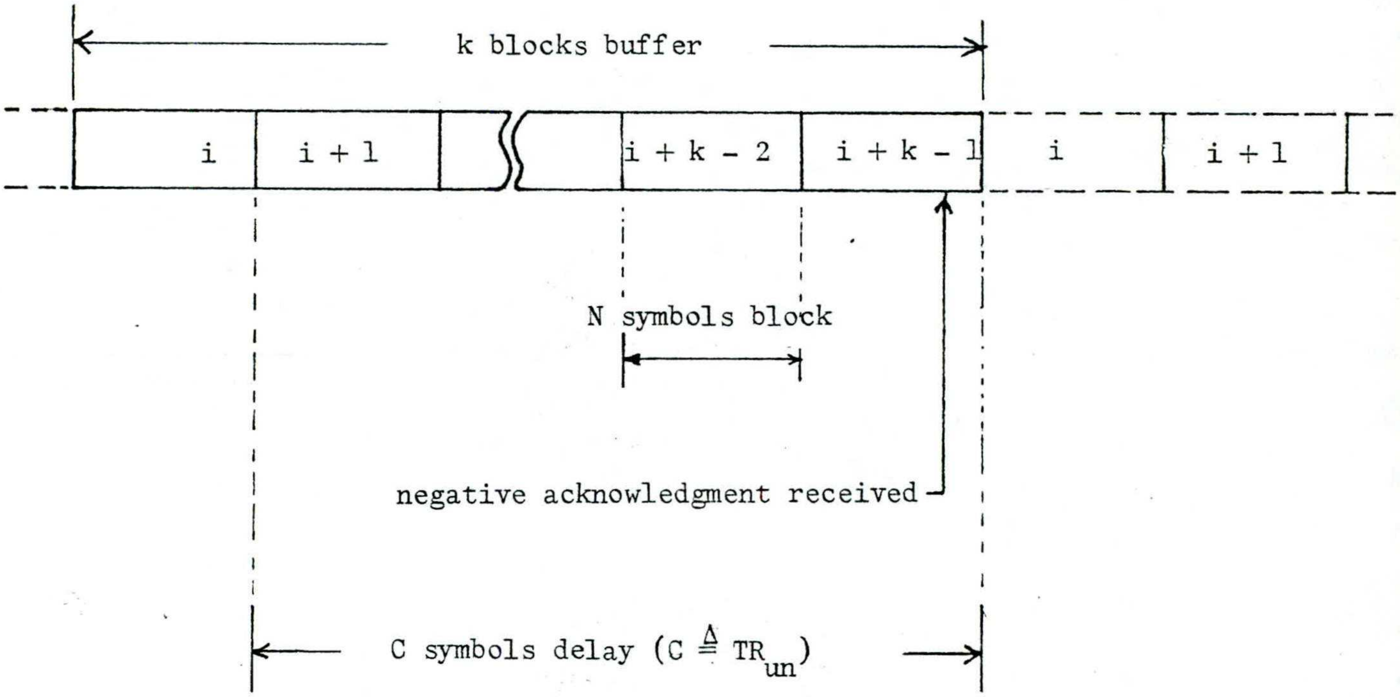

k blocks buffer
i
i + 1
i + k - 2
i + k - 1
i
i + 1
N symbols block
negative acknowledgment received
C symbols delay ($C \triangleq TR_{un}$)

Figure 3.3 Block sequencing between repetitions of block # i for the continuous ARQ scheme.

To compute the system throughput let us assume that the system operates in steady state and that the effect of acknowledgment errors is negligible.

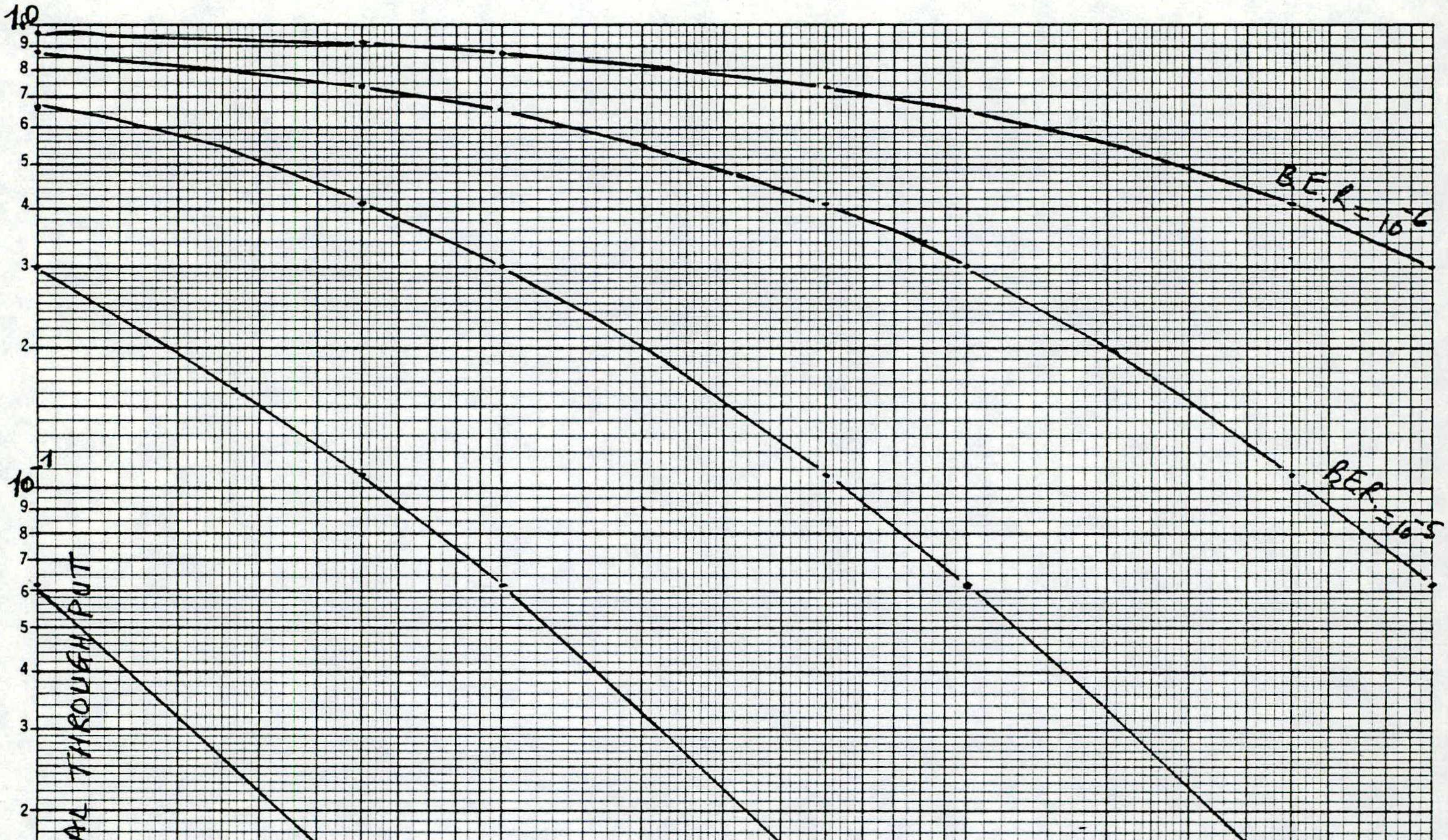


Figure 3.2.b Optimal Throughput versus Modem Bit Rate for Stop-and-Wait ARQ Systems

Consider relevant system states in which a block # i is the transmitted head block (awaiting for its own significant acknowledgment) for the first time. The average a counted in symbols, separating two consecutive relevant system states is given by the following mathematical expectation

$$\bar{C} = \sum_{j=1}^{\infty} (j(N+C) - C)(1 - P_B) P_B^{j-1}$$

where C is the delay (counted in symbols) between the end of transmission of a block of N bits and the beginning of the retransmission of that same block whenever a negative acknowledgment is received. Referring to fig. (3.1), it is readily seen that j(N + C) - C is the time period (in symbols) during which block # i acts as a head block for transmission or retransmission; and consequently, (j - 1) is the exact number of times block # i is retransmitted as a head block.

We obtain then

$$\bar{C} = (1 - P_B)\left((N + C) \sum_{j=1}^{\infty} j P_B^{j-1} - C \sum_{j=1}^{\infty} P_B^{j-1}\right)$$

$$= (1 - P_B)\left((N + C) \frac{d}{dP_B}\left(\sum_{j=0}^{\infty} P_B^{j}\right) - C \sum_{j=0}^{\infty} P_B^{j}\right)$$

$$= (1 - P_B)\left((N + C) \frac{d}{dP_B}\left(\frac{1}{1 - P_B}\right) - C\left(\frac{1}{1 - P_B}\right)\right)$$

$$= \frac{N + CP_B}{1 - P_B} = \frac{N + TR_{un}P_B}{1 - P_B}$$

Since there are exactly RN = K information bits transmitted through the system during the average time period $\bar{t} = \bar{C}/R_{un}$ the throughput is given by

$$\eta = \frac{RN}{\bar{t}R_{un}} = \frac{RN}{\bar{\bar{C}}} = \frac{RN(1 - P_B)}{(N + TR_{un}P_B)} = \frac{K(1 - P_B)}{(N + TR_{un}P_B)} \tag{3.5}$$

Let us now examine the variation of the throughput η as a function of other system parameters (channel error characteristics, modem bit rate, propagation delay, code rate, block length, etc.). Several authors have assumed that the rate of the error detection code is constant and attempted to maximize the throughput η with respect to the block size N. With this assumption and in the case of a binary symmetric channel with bit error rate p, we have

$$\eta \simeq \frac{Ne^{-Np}}{N + C(1 - e^{-Np})} R \tag{3.6}$$

so that

$$\frac{d\eta}{dN} = R \frac{e^{-Np} \; (C(1 - e^{-Np}) - Np(N + C))}{(N + C\,(1 - e^{-Np}))^2} > 0 \text{ for all } N > 0$$

The optimum block length is thus obviously the minimum $N_{opt} = 1$, and the system throughput is therefore bounded by

$$\eta^* = \frac{R}{1 + Cp} = \frac{R}{1 + TR_{un}p} \tag{3.7}$$

Such a result can be used as a fairly good measure of the system efficiency, but it provides no help in determining operationally the block size because it is well known that the rate of error detecting code is effectively low whenever the block size is small.

For illustrative purposes, let us consider a code used by IBM in many of their computer-communications systems [8]. The code has K = 16 parity symbols and is capable of detecting all blocks with three or fewer errors or with burst error patterns spanning 16 or fewer bits, provided the block length is less than $2^{15} - 1$ bits. With a block length of 800 bits, the probability of having an undetected block error with this code has been pessimistically estimated at 10^{-8} [9] on a data link using telephone lines. If this error detection code is used over a binary symmetric channel with bit error rate p, we then have

$$\eta = \frac{Ne^{-Np}}{N + C\ (1 - e^{-Np})} \times \frac{N - K}{N} \tag{3.8}$$

where K = 16.

Consequently

$$\frac{d\eta}{dN} = \frac{e^{-Np}(C(1 - e^{-Np}) - Np\ (N + C - K) + k\ (1 + pC)}{(N + C(1 - e^{-Np}))^2}$$

and the optimum block length is a solution to the following equation

$$C(1 - e^{-Np}) - Np(N + C - K) + K(1 + pC) = 0 \tag{3.9}$$

In the case where Np is sufficiently small, i.e. so small that $1 - (1 - p)^N \cong 1 - e^{-Np}$, and Cp is much smaller than 1, we obtain an approximate optimum block length given by

$$\hat{N}_{opt} \simeq \frac{K}{2} + (\frac{K}{p})^{\frac{1}{2}} \tag{3.10}$$

For an ARQ system operating with the following parameters

modem signalling rate, $R_{un} = 10^4$ bps
propagation delay, $D = 0.512$ sec.
number of parity symbols per block, $K = 16$

The results of the computation of the approximate optimal block length N_{opt}, the corresponding system throughput η, as well as the upper bound η^* are represented as a function of the channel bit error rate p in table 3.4.

p	10^{-6}	10^{-5}	10^{-4}	10^{-3}
$\hat{N}_{opt}$	4000	1265	400	(135)
$\hat{\eta}$	.987	.927	.614	(.134)
η^*	.995	.951	.661	.163

Table 3.4 Variations in the values N_{opt}, η, η^* as a function of p

We have seen in the above illustrative example that it is logically necessary to take into account the coding rate of the chosen error detecti codes R(N) as a function of the block size N, in the process of determining the optimum block length through maximization of the system throughput. However, the user is usually not only concerned with the system throughput but also with the undetected block error probability given a value P_u for the probability of undetected block error acceptable to the user, it is theoretically possible to determine the appropriate code rate R(N) from the capability of the error detection code used and channel error characteristics. Unfortunately, this approach seems impractical since the amount of computations required is enormous. Consequently, in order to give rough ideas about the throughput of continuous ARQ systems, let us restrict ourselves to a specific example with the following characteristics:

propagation delay of 0.512 second

block size of 1023 bits

BCH (1023, 913) block code used for error detection

BSC type channel with bit error rate of 10^{-7}, 10^{-6}, 10^{-5}, 10^{-4} or 10^{-3}

System throughputs as functions of modem bit rate for various channel bit error rates are shown in Figure (3.5). In Figure (3.6) the throughputs of stop-and-wait and continuous ARQ system associated with a block size of 1023 bits and bit error rate of 10^{-4} are compared to their corresponding upper bounds given respectively by the expressions (3.4) and (3.7). The results show that for the satellite link, the ARQ systems give good performances characterized by high throughput and low undetected block error in cases of either very low bit error rates (10^{-6} or less) or reasonably low bit error rates (10^{-5} or 10^{-4}) and moderate modem bit rates (10^{4} or 10^{5} bps). For channels of rather poor quality (a bit error rate of 10^{-3} for example) the system throughput decreases drastically. The last condition may prevail if the data link uses a time multiplexed sub-channel on a high speed TDMA satellite channel.

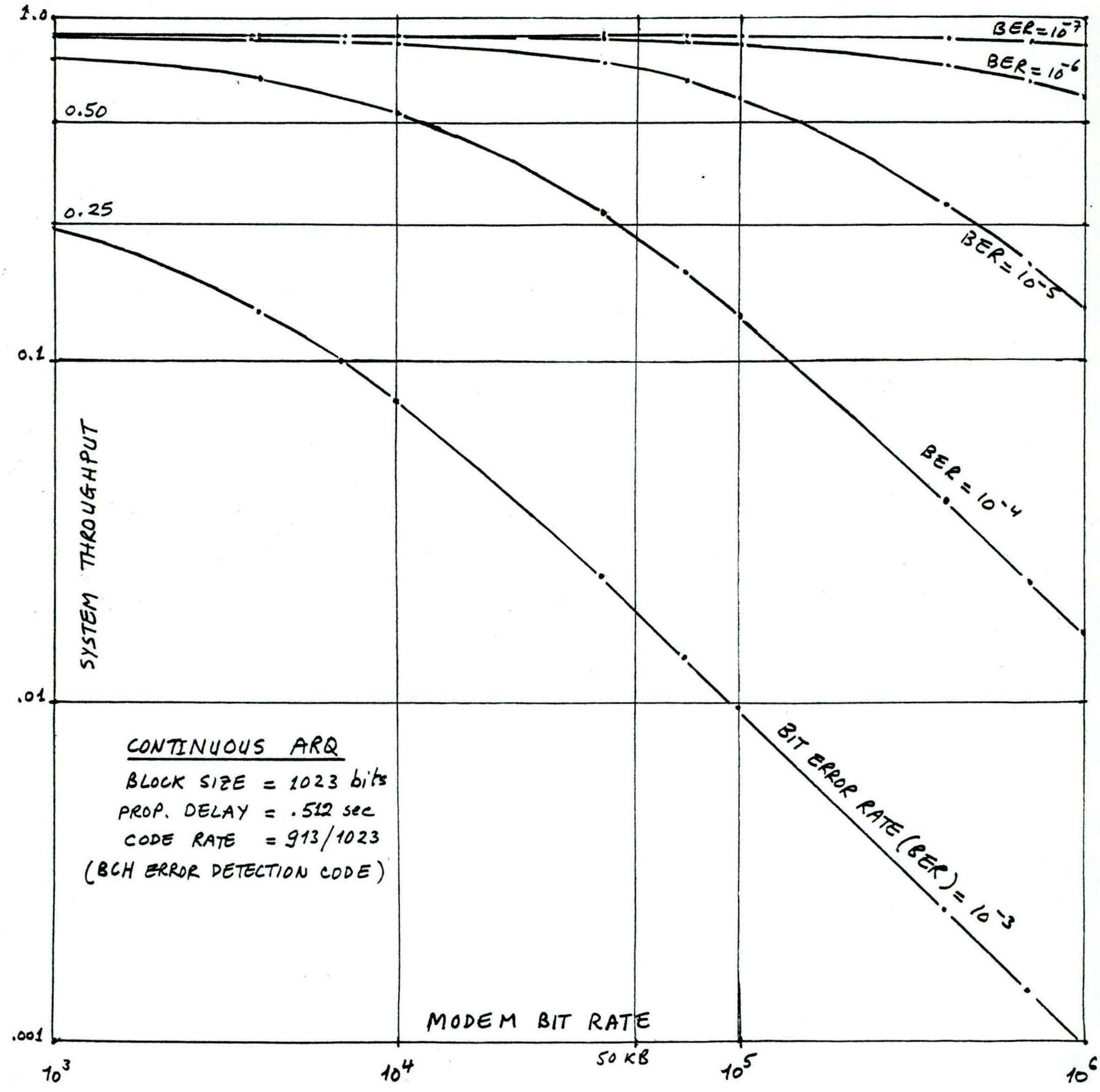


Figure 3.5 Throughput versus Modem Bit Rate for a Continuous ARQ System

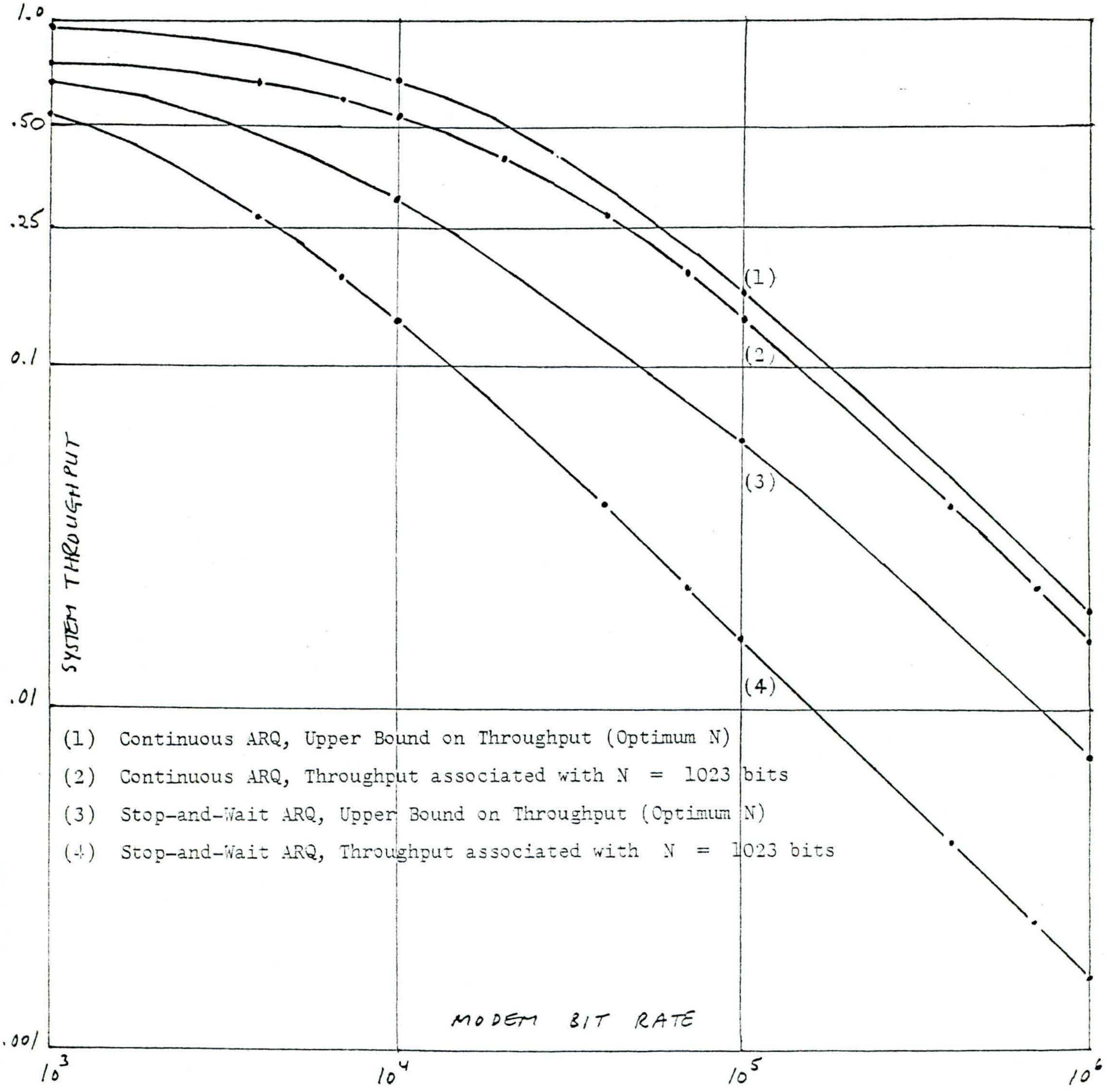


Figure 3.6 Throughputs and Upper Bounds versus Modem Bit Rates for an ARQ System with Bit Error Rate of 10^{-4}

III.4 Undetected Block Errors and Error Detecting Code Rates.

The probability of undetected block errors P_u depends on the error detection capability of the error detecting code used. Given a class of error detecting codes, the capability of error detection is a function of the coding rate R. There is consequently a relationship between the probability of undetected block errors P_u and the error detecting code rates R. Such a relation is a peculiar characteristic of error detecting codes and involves the block size N. However, it might be computationally a fantastic task to single out the required code rates R(N, P_u) as a function of the block size N and probability of having an undetected block error P_u.

For illustrative purposes, let us consider an ARQ system for data transmission via a satellite link (modelled as a BSC) using BCH error detection codes. Let us use the following upper bound for the probability of undetected block errors

$$P^*_u = 1 - \sum_{\ell = 0}^{t} \binom{N}{\ell} (1 - p)^{N-\ell} p^{\ell} \qquad (3.11)$$

where p = the channel bit error rate

t = the code detection capability, maximum number of erroneous symbols in a block resulting in an error detection

N = the block length

If P_u is given (say 10^{-8} or 10^{-10}), it is then possible to compute the required code detection capability t for a block of length N (or equivalently the ratio t/N). From this value, it is subsequently possible to choose an appropriate BCH code and obtain a corresponding coding rate R: [10, Table 9.1 and Appendix D].

The required code detection capability ratio t/N and code rate R are given in Fig. (3.7 and 3.8) for the values $P_u = 10^{-10}$, and $p = 10^{-3}$, 10^{-4}. The results show that for moderate block sizes (a few thousands of bits), the code rate R is quite insensitive to block size variations, and reasonably high (about 90%). Fig.(3.8) shows the variations of the maximum coding rate R compatible with given P_u with respect to the block size. Moreover, by using the Chernov bounding technique on the tail of the Bernouilli probability distribution.

$$B(k; N, p) = \binom{N}{k} p^k (1 - p)^{N - k}$$

it is possible to determine the asymptotic behaviour of the required code capability as the block size N increases indefinitely. In fact, let

$$Z = \sum_{i=1}^{N} z_i$$

where

$$z_i = \begin{cases} 1 & \text{with probability } p \\ 0 & \text{with probability } (1 - p) \end{cases}$$

then

$$\text{Prob}\ (Z \geqslant N\lambda) \leqslant 2^{-N(T_p(\lambda) - H(\lambda))}, \ \lambda > p \qquad (3.12)$$

where

$$\begin{aligned} H(\lambda) &= -\lambda \log_2 \lambda - (1 - \lambda) \log_2 (1 - \lambda) \\ T_p(\lambda) &= -\lambda \log_2 p - (1 - \lambda) \log_2 (1 - p) \end{aligned} \qquad (3.13)$$

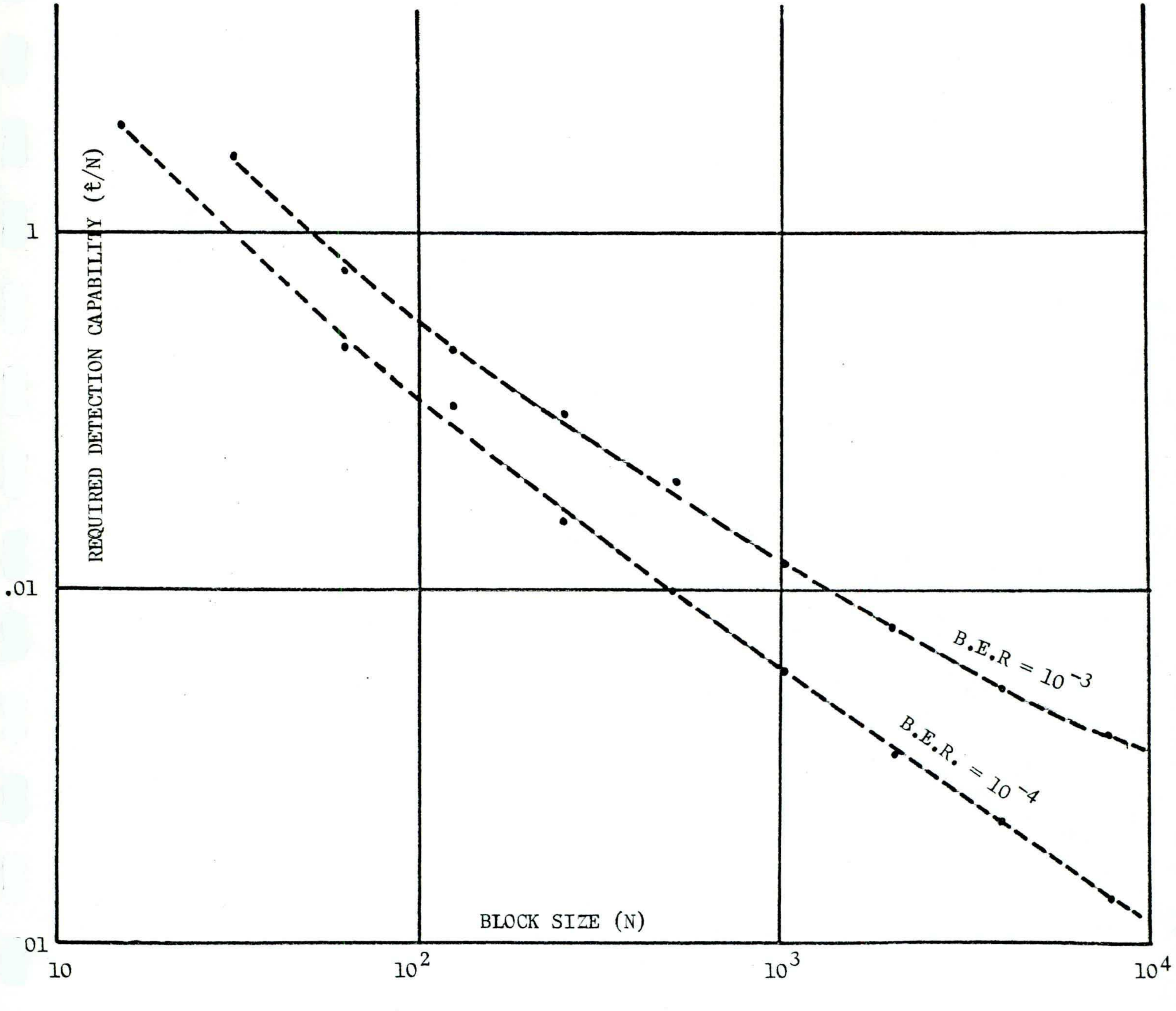


Figure 3.7: Required Code Capability versus Block Size for a Probability of Undetected Block Error of 10^{-10}

Thus, one would have, defining P^*_u as the Chernov upper bound on P_u:

$$2^{-N(T_p(\lambda) - H(\lambda))} = P^*_u$$

or

$$-N\ (T_p(\lambda) - H(\lambda)) = \log_2 P^*_u$$

$$T_p(\lambda) - H(\lambda) = \frac{1}{N} \log_2 \frac{1}{P^*_u}$$

$$\lambda \log_2 (\frac{\lambda}{p}) + (1 - \lambda) \log_2 (\frac{1 - \lambda}{1 - p}) = \frac{1}{N} \log_2 \frac{1}{P^*_u} \qquad (3.14)$$

Since λ must decrease monotonically to p as $N \to \infty$, the equation (3.14) can be further approximated by

$$(\ln 2) \quad (\frac{\lambda(\lambda - p)}{p} - \frac{(1 - \lambda)\ (\lambda - p)}{1 - p}) = \frac{1}{N} \log_2 \frac{1}{P^*_u}$$

which is the 1st order approximation obtained from

$$(\ln 2)(\lambda (\frac{\lambda - p}{p} + \frac{(\lambda - p)^2}{p^2} + \ldots)$$

$$+ (1 - \lambda) \quad (- \frac{\lambda - p}{1 - p} + \frac{(\lambda - p)^2}{(1 - p)^2} + \ldots)) = \frac{1}{N} \log_2 \frac{1}{P^*_u}$$

Then

$$\frac{t^*}{N} = \lambda \cong p + \sqrt{\frac{p(1 - p) \ln(1/P^*_u)}{N}} \to p \text{ as } N \to \infty$$

On the other hand, if the ratio $R = \frac{K}{N}$ is kept fixed, the lower bound on $\frac{t}{N}$ approaches zero as N becomes large. As a consequence, if one desires to deep R at a reasonable level and guarante a fixed limit on P_u the block size connot grow idefinitely.

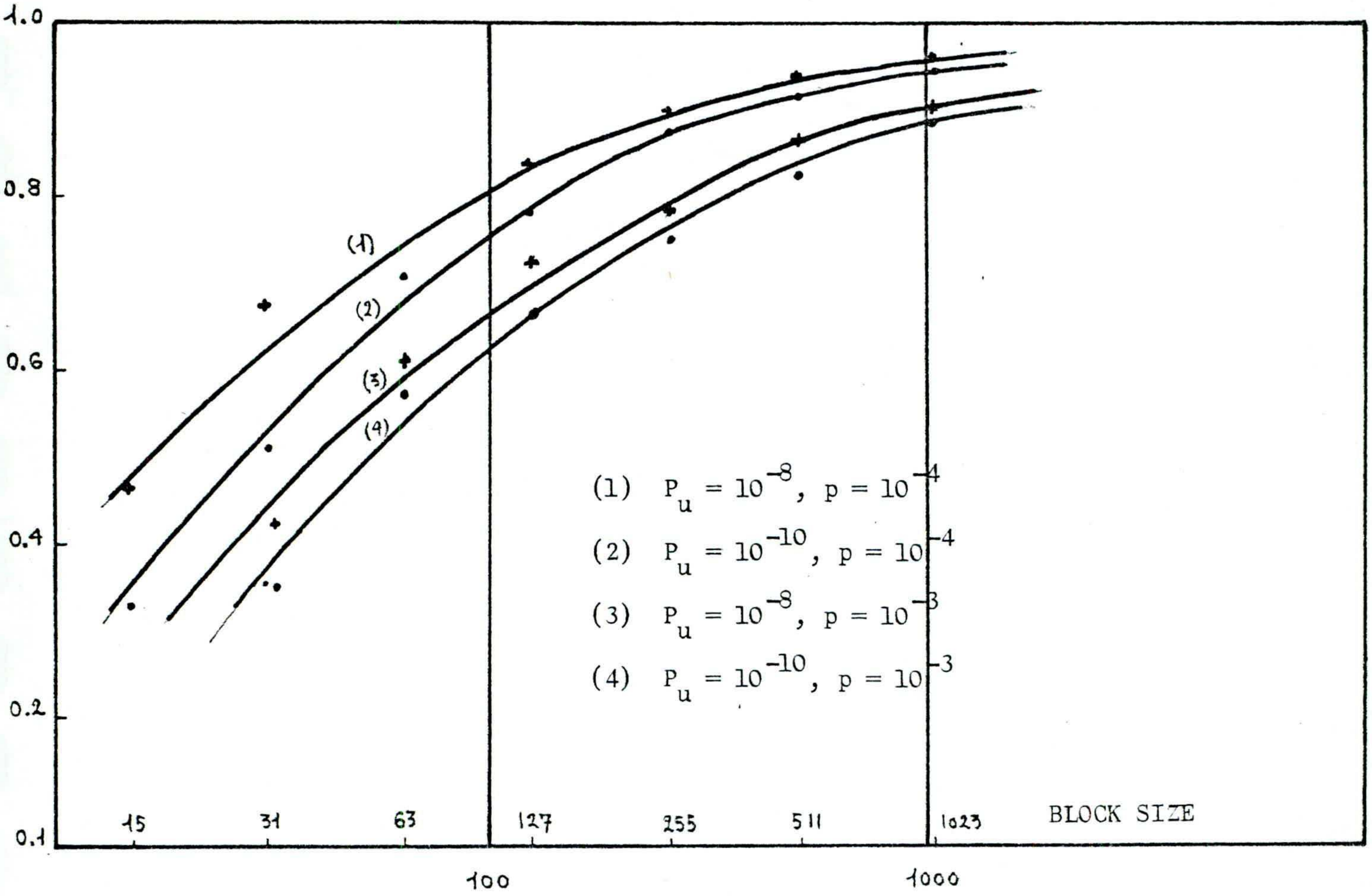


Figure 3.8: Required Code Rate versus Block Size.

III.5 Other Variations of ARQ Systems.

Continuous transmission, where applicable, is more efficient than stop-and-wait operation. In many cases however, an upper limit may be placed on the number of blocks of data that remain accessible for retransmission at the sending terminal. Hence, we may in fact have a continuous L-block (for $L \geqslant 2$) buffer system that operates as a continuous system as long as the buffer capacity is not exceeded. If the acknowledgment delay is sufficiently large to allow the buffer to fill, then the system stops and waits for the acknowledgment message.

Another variation of the continuous system, called selective repeat system, in which only the required block is retransmitted performs significantly better at higher error probabilities [11]. However, the blocks do not arrive at the receiver in serial order, requiring additional circuitry for rearrangement. Also, with this system, the acknowledgment message must be able to identify the block with which it is associated. The overall link control with such a scheme is quite involved and requires the full capability of minicomputers.

Recently, some improvements on the throughput efficiency of ARQ systems over satellite links under high error rate conditions have been obtained by Sastry [12] in the following manner. For stop-and-wait ARQ systems, whenever a request for repeat is received, a finite number $i \geq 2$ of retransmissions of the erroneous block are sent continuously before stopping and waiting for acknowledgment. In the continuous ARQ system, the required block is transmitted continuously until it is received correctly instead of sending after each repeat different blocks which are ignored at the receiving terminal. In this case, the throughput is significantly improved at high error rates and whenever the delay counted in symbols is much larger than the block length, i.e. whenever the modem bit rate is high. Unfortunately, the overall value of throughput can still become so small as to make the ARQ systems inefficient for very high speed data transmission.

III.6 Exploiting the Return Channel

It is a known fact that ARQ systems are highly reliable, i.e. the probability of undetected block error is very low. However, their use implies the availability of a return channel. This requirement should be accounted for in the process of determining the actual throughput of ARQ systems in order to shed more insights into the comparison between ARQ and FEC techniques.

Let us consider, for example, the case of full-duplex channels used in simplex or half-duplex transmission mode and where one terminal merely sends data and the other just acknowledges. Consequently, in simplex or half-duplex transmission the return channel is very poorly used. If the throughput value of the forward channel is η (as determined previously), then the overall system throughput should be defined as $\eta/2$ to reflect the poor utilization of the return channel. To remedy to such an inefficient use of a potentially full duplex channel, the simplex/half-duplex transmission scheme can be extended to full-duplex operation by allowing both terminals to simultaneously send data and confirmations.

To illustrate the extension of half-duplex transmission to full-duplex operation, let us assume that the block length is sufficiently large (or the delay is sufficiently short), so that the acknowledgment of the previous block is received by the sending terminal before the transmission of the current block terminates. In this case, a reliable half-duplex transmission would operate according to the following data link control procedure.

A sequence numbering binary symbol is attached to each block sent from A to B. After B receives the block it decides whether or not this block is error-free. It sends back to A an ACK/NACK message (possibly just a bit) indicating to A whether or not the immediately preceding A → B block was error-free. After A receives this acknowledgment, one among the three following possibilities holds: (1) the A → B block was correctly transmitted, (2) the A → B message was incorrectly received, (3) the ACK/NACK message was in error so that A does not know whether the A → B block has been correctly received or not. In case (2) and (3) A simply sends a copy of the same A → B block. In case (1), A transmits the next block with a sequencing symbol inverted. Whenever B receives a block that is not in error, it compares the sequence symbol of this new block to the sequence symbol associated with the most recent error-free block. If the sequence bits are equal the new block is rejected as being a copy of an already acknowledged block; otherwise, the new block is accepted.

From the above half-duplex transmission one obtains a full-duplex operation by performing the following modifications. Each terminal has its own sequencing state. As B sends the acknowledgment back to A, it also transmits a block with sequencing symbol. The A → B block that follows contains the acknowledgment information for the previous B → A transmission. Both simplex channels operate then essentially independently of each other.

The performance analysis of stop-and-wait, selective repeat and continuous ARQ systems for both half-duplex and fall duplex modes has been carried out by Benice and Frey [13] under the assumption stated previously that the time delay counted in bits was less than the block length. An interesting continuation of this work would be to extend the analysis to the case of satellite channels where the time delay is larger than the block length, sin this would be the case at very high transmission rates.

IV. PERFORMANCES OF FEC, ARQ SCHEMES ON SATELLITE CHANNELS COMPARISON WITH A NEW HYBRID SYSTEM

IV.1 INTRODUCTION

This chapter considers the use of direct FEC and ARQ error control techniques as applied to the transmission of data through a satellite link operating in the TDMA mode. As a working hypothesis, the data will always be assumed to be structured in blocks; the error performance being P_e/P_u the uncorrected/undetected block error probability (to be $\leq 10^{-10}$). On the other hand, only the earth station to earth station link is considered; the terrestrial network that links the actual users to the earth stations is not taken into account in this report. The choice of a data structuration in blocks is essentially dictated by the recent advent of computer data communication networks (ARPA, CYCLADES), in which blocks constitute the basic information entity. The parameter of interest will be the variation of the global throughput compatible with a given error performance as a function of the signaling rate on the satellite channel. It appears essentially, from the results which were obtained, that FEC systems do not provide a sufficient coding gain at the high signal upon noise ratio values compatible with the given error performance. ARQ systems, on the other hand are inherently limited in speed by the rapidly decreasing value of the throughput efficiency as the signaling speed on the satellite channel increases; for all ARQ systems except the selective type ARQ scheme, this effect is even more pronounced as the data rate of the individual users grows larger. It is however possible to consider a third system which combines the advantages of both FEC systems and ARQ schemes. Such a system will be called a hybrid ARQ-FEC scheme and consists of an outer ARQ system and an inner FEC system. The coding gain provided by the FEC system decreases the number of repetitions in the ARQ as the signaling speed on the satellite channel is increased, extending considerably the range of operation. Moreover, the system is virtually error free, since the error rate which is dictated by the ARQ system remains quite low and is independent of variations in the operating parameters of the system.

IV.2 FEC TECHNIQUE

In a digital communication system employing FEC technique, the data source generates binary information symbols at the rate R_s bits/sec. These information symbols are encoded for the purpose of error protection and the encoder output is a binary sequence of rate R_n symbols/sec. The code rate R in bits/symbols is then given by

$$R = R_s/R_n \tag{4.1}$$

Since $R < 1$, the transmission symbol speed is larger than the data speed delivered by the source. Equivalently, the introduction of an error control coding scheme requires a bandwidth expansion.

Let the received power at the earth station be P and let N_o be the spectral density of the channel noise. Then the signal-to-noise ratio per information bit

$$E_b/N_o = P/(N_oR_s) \tag{4.2}$$

serves as a figure of merit for different combinations of coding and modulation schemes. It is clear that a coding or modulation scheme which reduces the E_b/N_o required for a given error probability leads to an increase in allowable data rate and/or a decrease in the necessary P/N_o. The problem then is determining the system that will operate at the lowest E_b/N_o with a given quality. A lower bound exists on the available E_b/N_o. The value E_b/N_o min is derived in section II and found to be -1.6 dB for the infinite bandwidth white gaussian channel. Among all real time decoding systems, Viterbi maximum likehood decoding of convolutional codes with soft quantization (in practice 8 levels are sufficient) is known to yield the best performance in terms of bit error rate. In view of this fact, it was decided to evaluate the performances of FEC systems using rate $\frac{1}{2}$ convolutional codes of short constraint lenghth ($3 \leq k \leq 7$) and a Viterbi decoder. The complete system is represented on Figure (4.1). Consider the transmission of blocks of length N bits coming from different users, and let p_v be the bit error rate for this decoder. Although the errors delivered by the decoder appear in bursts, because of the natural bit interleaving provided by the TDMA scheme (see the Appendix A), for each user the errors will appear independently with the same bit error probability p_v. The block error probability is then

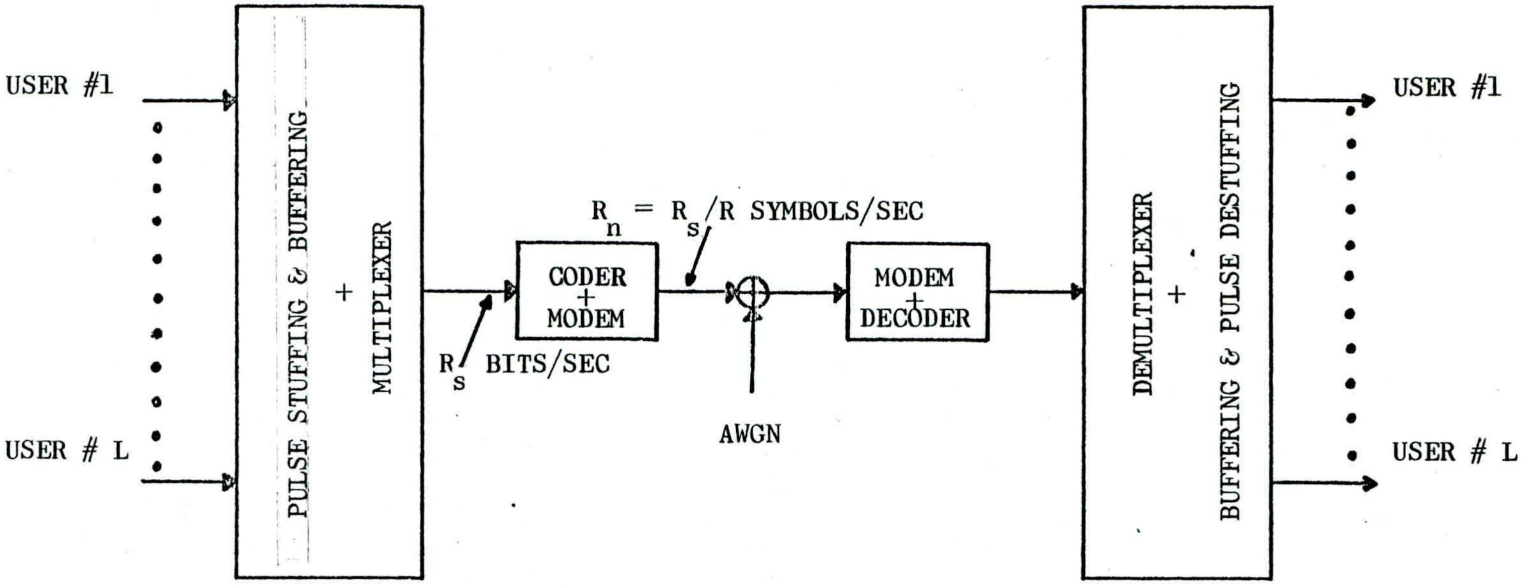


Figure 4.1: BLOCK DIAGRAM OF THE CODED TDMA CHANNEL SCHEME

(CONVOLUTIONAL ENCODING AND VITERBI DECODING)

$$P_B = 1 - (1 - p_v)^N \simeq Np_v, \qquad Np_v \ll 1 \qquad (4.3)$$

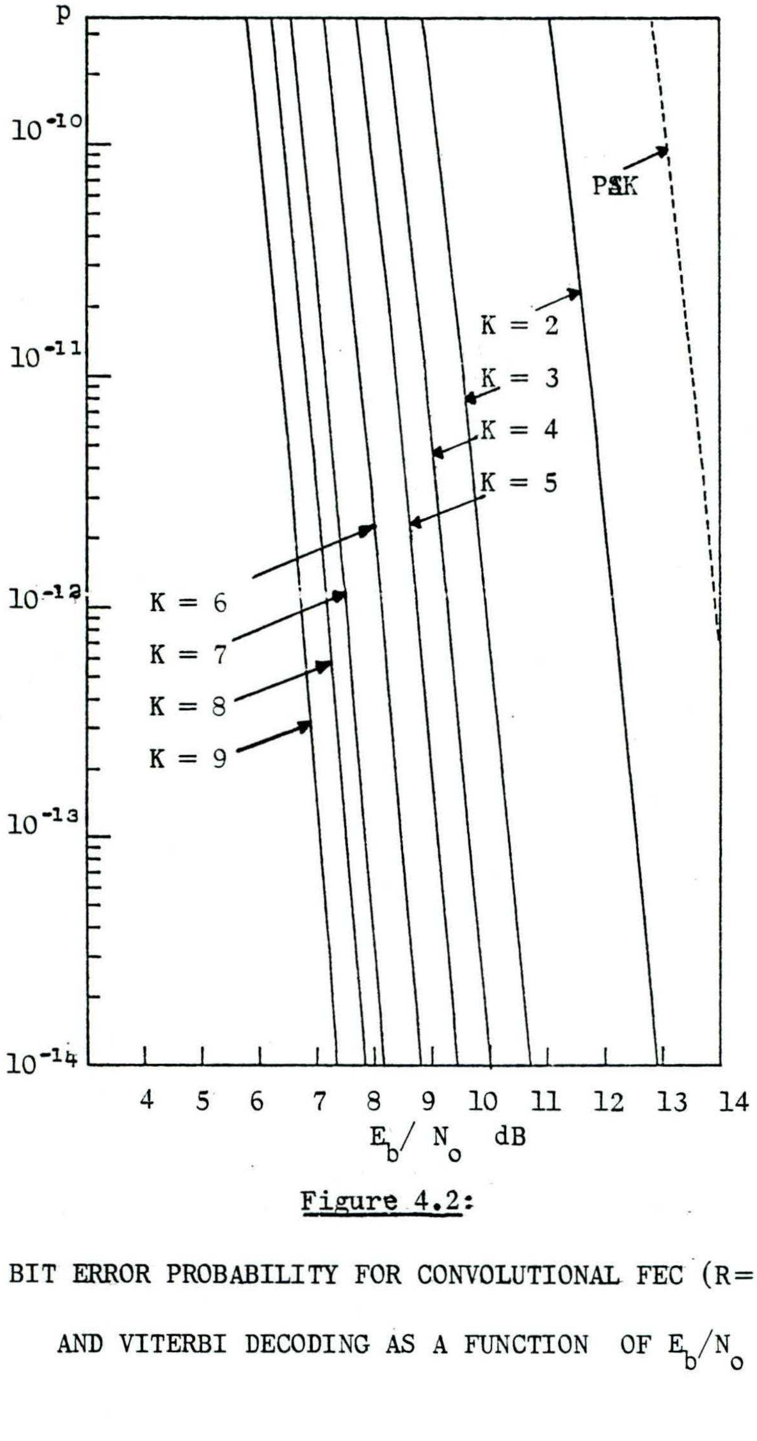
Consequently, for a user to have a block error probability $P_B = 10^{-10}$, the bit error rate must be $p_v = 10^{-13}$ for block lengths N = 1000 bits.

Error probability bounds vs E_b/N_o for Viterbi decoding and coherent PSK modulation can be evaluated by using the transfer function of the convolutional codes [4]. Using the optimal codes and their spectra [14] these bounds have been computed for a soft quantized Viterbi decoder (see Appendix B) and are plotted in Figure (4.2). The up-link being virtually error free, we consider only the down-link calculation as given in Appendix A to obtain

$$E_b/N_o = 47.6 + G/T - 10 \log_{10} R_s \qquad (4.4)$$

where R_s is the transmission rate in bits/sec and G/T the gain-tempara-ture ratio of the receiving earth station.

From Equation (4.4) and Figure (4.2), information rates R_s vs block error probability curves have been calculated for a station G/T = 40.7 and several constraint length codes (see Figure (4.3)). Results for the (23, 12), 3 errors correcting Golay code and coherent noncoded PSK are also given for comparison. The transmission rate for a given error performance increases with the constraint length of the code, but decreases monotonically as the error probability gets lower. Also, as the available energy per bit increases, coding becomes less attractive. However, in this case the advantages for using coding are quite apparent, since Figure(4.3)



<u>Figure 4.2</u>:

BIT ERROR PROBABILITY FOR CONVOLUTIONAL FEC ($R = \frac{1}{2}$)

AND VITERBI DECODING AS A FUNCTION OF E_b/N_o

[redacted] [redacted]

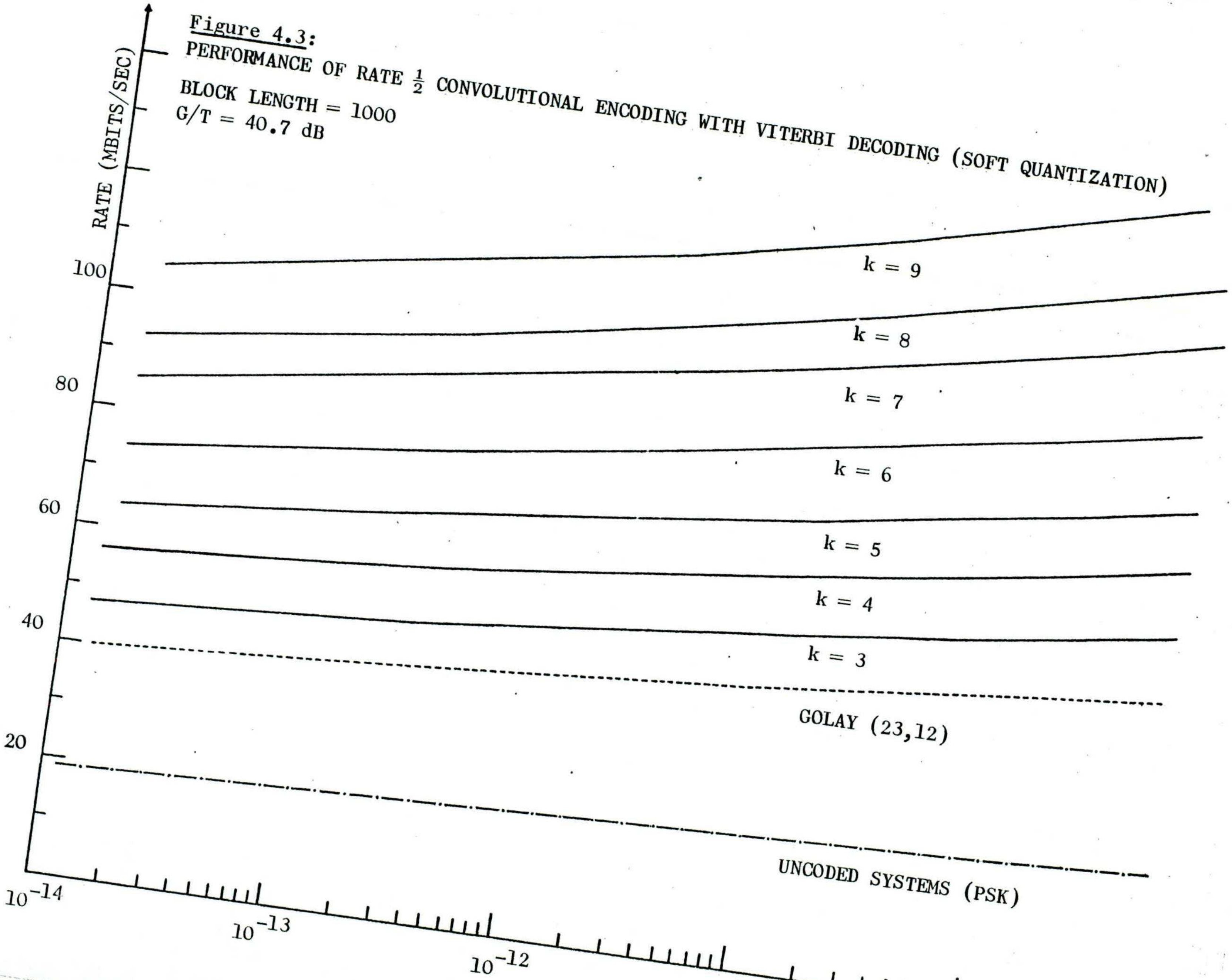


Figure 4.3:
PERFORMANCE OF RATE ½ CONVOLUTIONAL ENCODING WITH VITERBI DECODING (SOFT QUANTIZATION)
BLOCK LENGTH = 1000
G/T = 40.7 dB

shows that a transmission rate of 100 Mbits/sec can be achieved at $P_B = 10^{-10}$ with a code of constraint length K = 7, whereas the noncoded PSK modulation yields only a rate of 20 Mbits/sec. Naturally the actual signalling speed over the channel is $R_n = 2R_s$ symbols/sec for this code, and for very high transmission speeds, the buffering, speed translation and multiplexing of the individual user's data may become severe problems. All computations have been carried out using a value of G/T = 40.7 dB. For an actual $G/T = (G/T)_o$ dB, the available data rate R_s is obtained by multiplying the value obtained from Figure (4.3) by the constant factor B where

$$B = 10^{[(G/T)_o - G/T]/10} \qquad (4.5)$$

IV.3 ARQ TECHNIQUES

In ARQ systems block coding is employed but no error correction is performed by the decoder. Whenever an error is detected in a block, a retransmission of that block is requested through a return channel. A block is accepted by the user only after it appeared to be error free. Clearly then the important measures of performance are: the undetected block error probability P_u which is typically very small ($P_u < 10^{-10}$), and the throughput efficiency of the system. The throughput efficiency is defined as the ratio of the number of information bits transmitted by the forward channel to the signalling rate of the channel. A block diagram of a TDMA system using an ARQ scheme is shown in Figure (4.4).

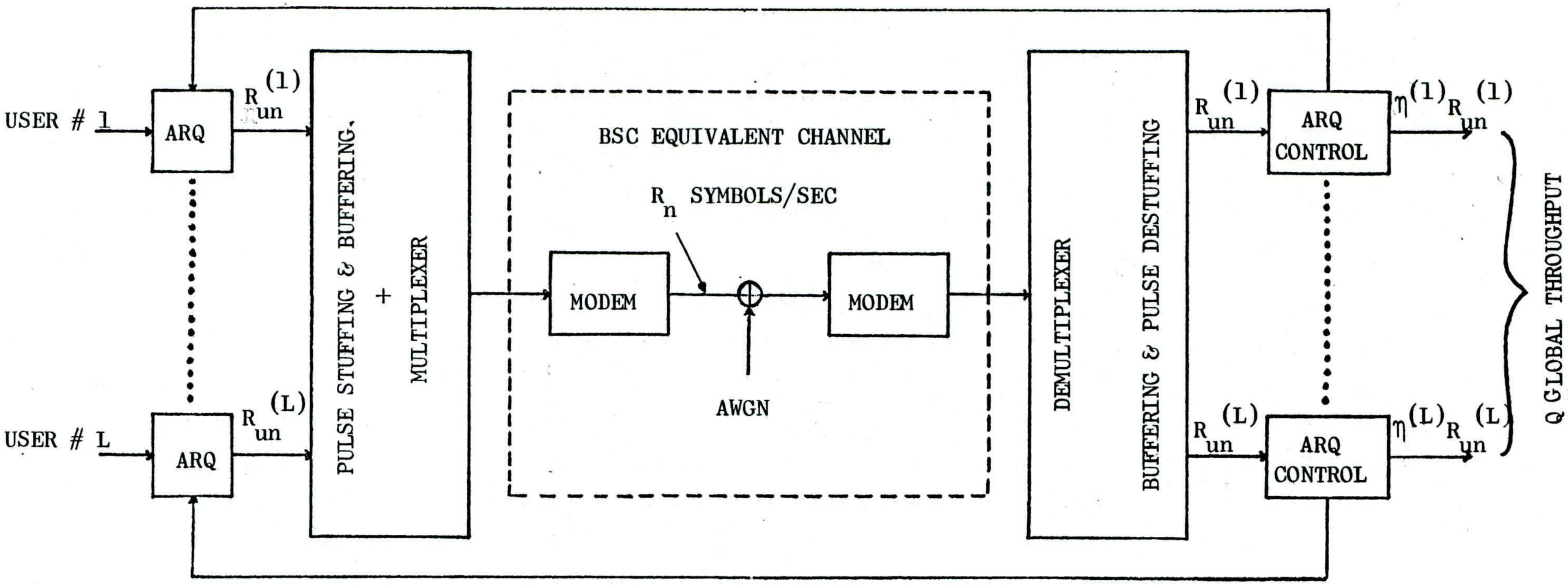


Figure 4.4: BLOCK DIAGRAM OF THE ARQ SCHEME

The most important variants of ARQ techniques are called "Stop and Wait", "Continuous" and "Selective". The Stop and Wait system is very inefficient for satellite channels and will not be considered here.

In a Continuous ARQ system, the blocks are transmitted consecutively, without any delay between blocks, as long as positive acknowledgements are received at the sending terminal. Whenever a negative acknowledgement is received, the sending station backs-up to the erroneous block and retransmits that block and all subsequent blocks. The throughput efficiency expression is derived in section II, and is reported here for later reference

$$\eta = \frac{K(1 - P_B)}{(N + R_{un}\ T\ P_B)} \quad \text{bits/symbol} \qquad (4.6)$$

where

K = information symbols per block

N = block length

R_{un} = user's signalling speed in symbols/sec

T = round trip delay + diverse service delays (Usually negligeable)

P_B= probability of block error.

In selective systems, only the particular block received in error is retransmitted. As a consequence the throughput efficiency becomes independent of the transmission delay, but the actual implementation of these systems is comparatively more complex. Using the same notation as for Equation (4.6), the throughput efficiency is

$$\eta_S = \frac{K}{N} (1 - P_B) \text{ bits/symbol} \qquad (4.7)$$

The computation of the global throughput for the system of Figure (4.4) is easily carried out as follows. For a given receiving station G/T, the symbol energy-to-noise ratio E_n/N_o corresponding to a given burst signalling rate R_n on the forward channel is given by the link equation

$$\frac{E_n}{N_o} = 47.6 + G/T - 10 \log_{10} R_n \qquad (4.8)$$

Assuming coherent PSK modulation, the transition probability p of the equivalent binary symmetric channel is then easily obtained [3]. Since the channel is memoryless, the block error probability is

$$P_B = 1 - (1 - p)^N \simeq pN, \qquad pN \ll 1 \qquad (4.9)$$

Refering to Figure (4.4), let $R_{un}^{(j)}$ symb/sec be the signalling speed of the $j^{\underline{th}}$ user, and let $\eta^{(j)}$ be its throughput efficiency. The throughput corresponding to this $j^{\underline{th}}$ user is

$$q^{(j)} = \eta^{(j)} R_{un}^{(j)} \text{ bits/sec} \qquad (4.10)$$

The global throughput for the L users is therefore

$$Q = \sum_{j=1}^{L} q^{(j)} \text{ bits/sec} \qquad (4.11)$$

or

$$Q = \sum_{j=1}^{L} \eta^{(j)} R_{un}^{(j)}, \text{ with } \sum_{j=1}^{L} R_{un}^{(j)} \leq R_n \qquad (4.12)$$

Let N, K, $R_{un}^{(j)}$, $\eta^{(j)}$ be the same for all L users. Then the global throughput for the entire system becomes

$$Q = \eta^{(j)} R_{un}^{(j)} \left\lfloor \frac{R_n}{R_{un}^{(j)}} \right\rfloor \simeq \eta^{(j)} R_n \text{ bits/sec} \quad (4.13)$$

where $\lfloor x \rfloor$ represents the integer part of x.

Curves giving the global throughput Q as a function of the TDMA burst rate R_n(i.e. the signalling speed on the channel) are given in Figure (4.5) for the continuous ARQ system. Users rates R_{un} varying from 1200 to 48000 symb/sec are considered for a receiving station G/T= 40.7, and for comparison purposes, results for 2400 symbols/sec with two smaller earth stations are also given. For the G/T = 40.7 station, we see that up to about 50 Msymb/sec, the system is very efficient for all users rates. However, for any user rate, as the signalling speed increases, the available E_n/N_o per transmitted symbol drops, leading to more errors. Hence retransmission requests become more frequent and the global throughput actually seen by the users starts decreasing; all throughput maximum occurs earlier for the high-speed users since, as the propagation time delay is constant, a retransmission request will affect a larger number of blocks as the users speed increases. For very large burst rates, the channel degrades so much that the system is almost continually retransmitting the same blocks leading to a drastic fall of the overall throughput. Clearly, beyond certain burst rate values, the ARQ system is indeed very inefficient and buffering of the incoming data becomes a severe problem. The evaluation of an upper bound on the probability of undetected error P_u can be easily carried out as follows. Let (N, K, d_{min})

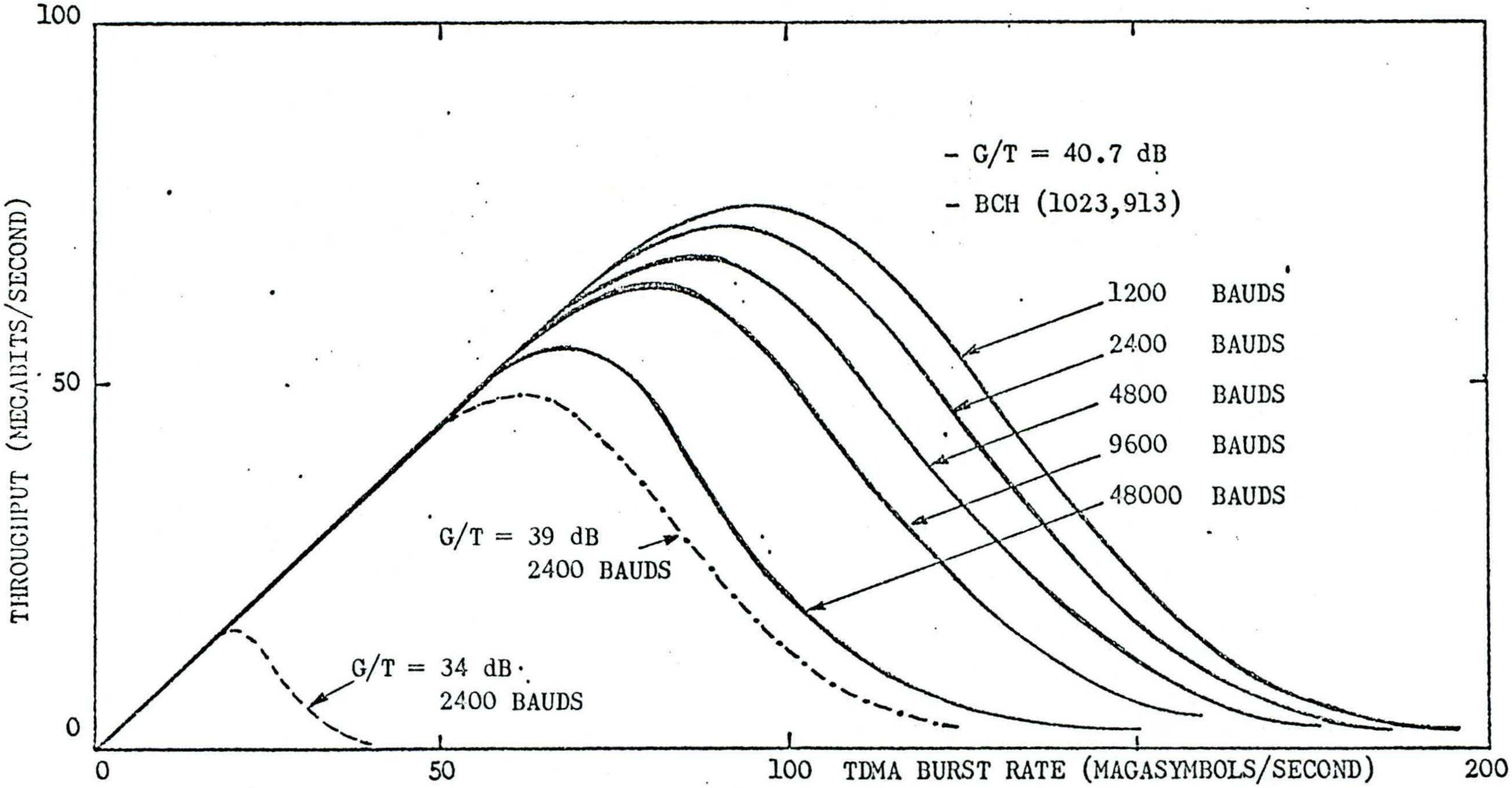


Figure 4.5: SYSTEM THROUGHPUT FOR CONTINUOUS ARQ WITH DIFFERENT USER RATES

be the characteristics of the error detecting code, where d_{min} represents as usual the minimum Hamming distance. The error correction capability of the code e is then related to d_{min} by

$$e = d_{min} - 1 \tag{4.14}$$

Denoting by p_v the cross over probability on the BSC equivalent to the TDMA channel, P_u can be bounded by

$$P_u \leq \sum_{p=e+1}^{N} p_v^{\,p}(1 - p_v)^{N-p} = \sum_{p=d_{min}}^{N} p_v^{\,p} (1 - p_v)^{N-p} \tag{4.15}$$

The summation on the right hand side of Equation (4.15), which represents the tail of binomial distribution of parameter p_v, can be bounded from above by a technique due to Chernov (see section II) to yield

$$P_u \leq \left[\frac{Np_v}{d_{min}}\right]^{d_{min}} x \left[\frac{1-p_v}{1-\frac{d_{min}}{N}}\right]^{N-d_{min}} \tag{4.16}$$

For the BCH code (1023, 913, 23), even for a value of $p_v = .1$, the bound of Equation (4.16), still yields a value for P_u of 1.06×10^{-21}.

IV.4 A HYBRID ARQ/FEC SCHEME

From the results obtained with FEC and ARQ techniques we observe the following:

The FEC technique provides a coding gain in the form of an increased available data rate at the expense of an increased required bandwidth.

However, as the level of performance becomes quite high, the effect of the coding gain is reduced and more powerful codes with more complex decoding schemes are necessary.

On the other hand, the ARQ technique provides an excellent error performance independently of the quality of the channel. However, as the channel becomes noisier the number of retransmissions increases, the same blocks get repeated over and over, leading to a very small effective throughput. Operation at the maximum throughput value is desirable, but clearly an ARQ system is rather inefficient if this maximum throughput occurs for a burst rate value lower than the maximum rate compatible with the available bandwidth of the link.

For a given users rate, improvement of the throughput will come from a reduction in the number of retransmission requests, that is by using a better quality channel. On a satellite link this could be achieved by a brute force increase of the repeater radiated power and/or of the receiving station G/T. Discarding this approach, we now propose a hybrid ARQ/FEC scheme which combines the advantage of both ARQ and FEC techniques. A block diagram of the scheme is shown in Figure (4.6). In this scheme the outer ARQ technique will provide a very high overall error performance for the system, but in order to have also a high throughput efficiency the raw channel quality is considerably improved by the introduction

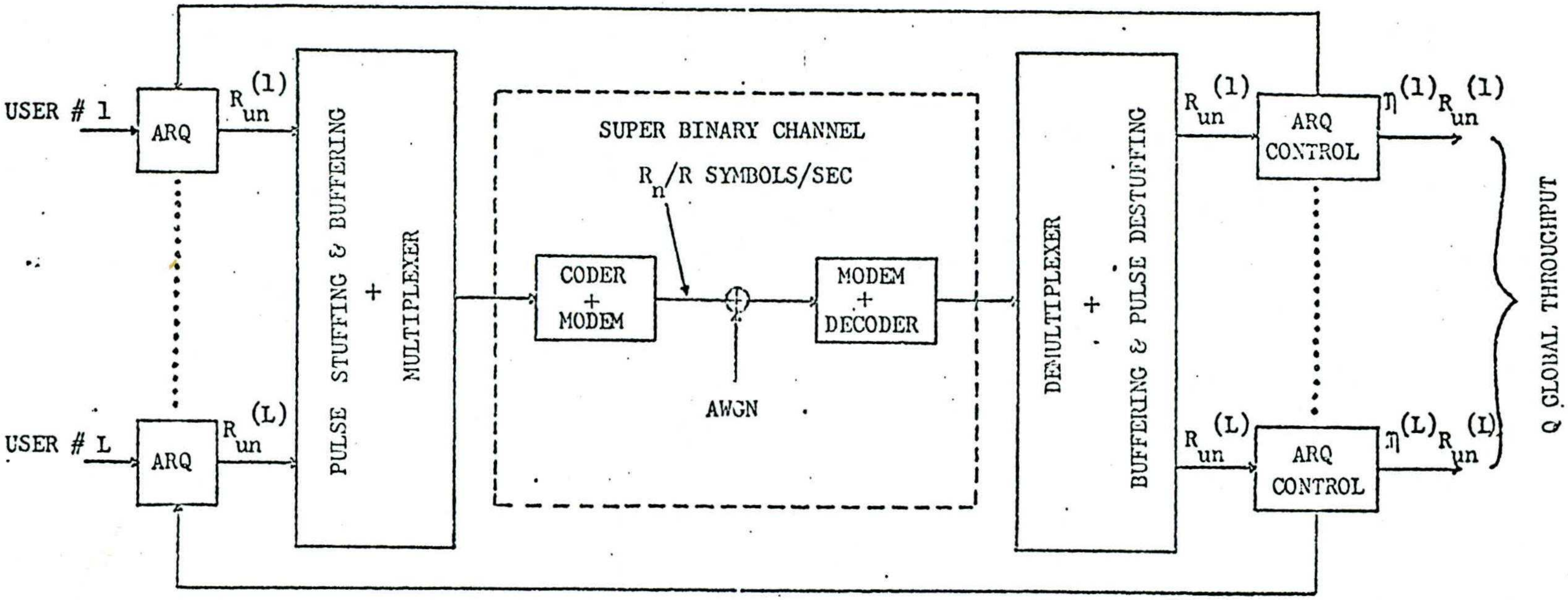


Figure 4.6 BLOCK DIAGRAM OF THE HYBRID ARQ/FEC SCHEME

of an inner FEC system. Consequently for the outer ARQ system the coder-raw channel-decoder may be regarded as a super binary channel whose transition probability becomes the bit error probability of the inner FEC system. Again, because of the natural interleaving of the TDMA technique, the super-channel appears to be memoryless, so that the same computation as for the ARQ system alone can be carried out.

Throughput results for a hybrid scheme using continuous ARQ as the outer scheme, with the powerful BCH (1023, 913, 23) as the error detecting code and rate $\frac{1}{2}$ convolutional coding with Viterbi decoding as the inner FEC system are shown in Figure (4.7). Results for only the 2400 symb/sec user rate is shown and for ease of comparison, the corresponding result for the ARQ system alone is also given. The hybrid scheme yields a considerably higher throughput than the ARQ scheme alone. The powerful effect of the inner FEC coding is amply demonstrated, for even with a relatively weak code, (K = 3 constraint length code), a doubling of the throughput is achieved. The straight line behaviour of the curves (up to approximately 50 Mbits/sec) indicates that few retransmissions are requested in this region: the super binary channel is virtually noiseless. However as the burst rate increases, the channel degrades and the error correcting capability of the inner code becomes exceeded. The errors given out by the FEC system are however detected by the ARQ system, which then requests retransmission. The operation of the entire system is thus that of an ARQ, and as expected with these

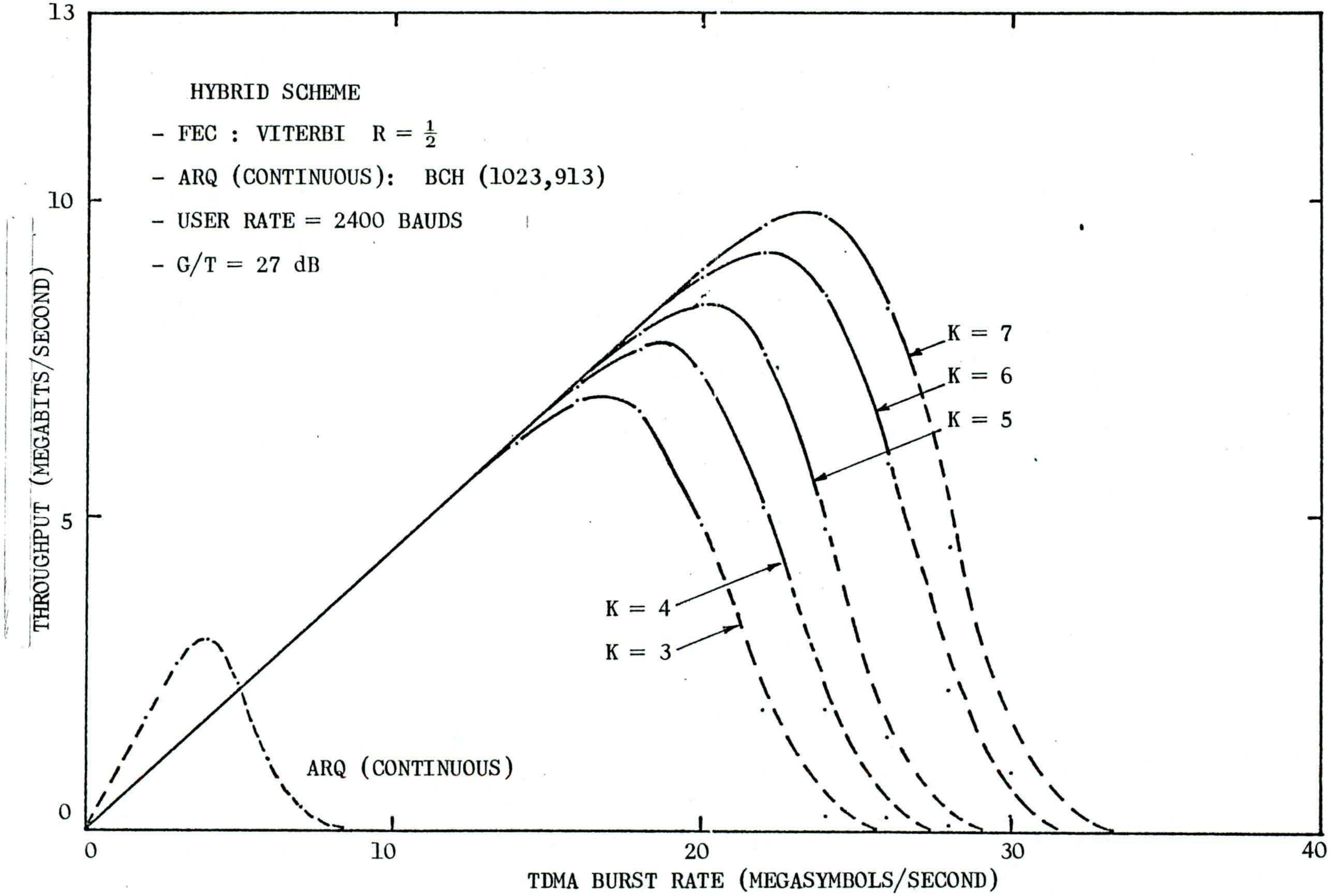


Figure 4.7 SYSTEM THROUGHPUT FOR ARQ/FEC HYBRID SCHEME

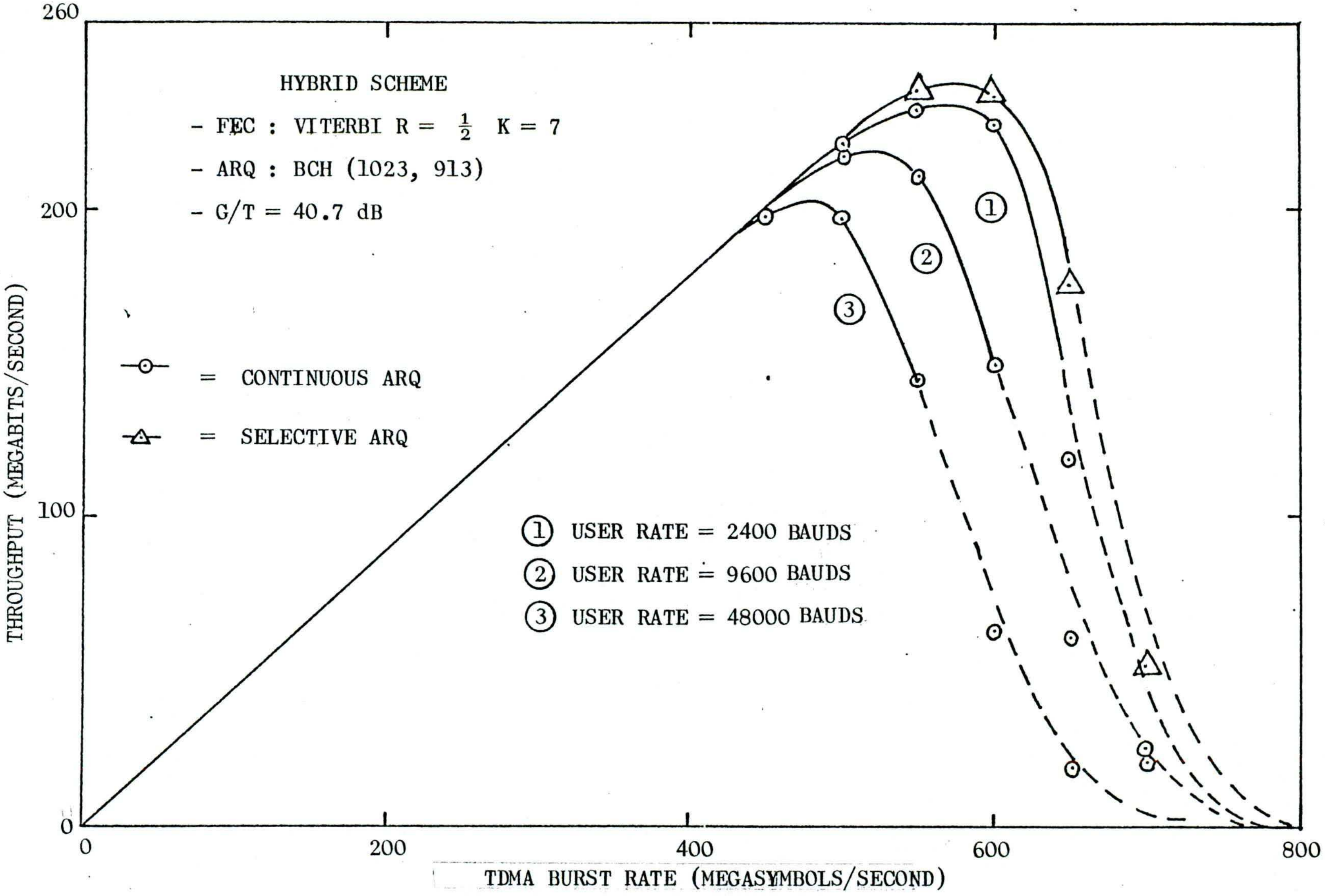


Figure 4.8 SYSTEM THROUGHPUT FOR ARQ/FEC HYBRID SCHEME

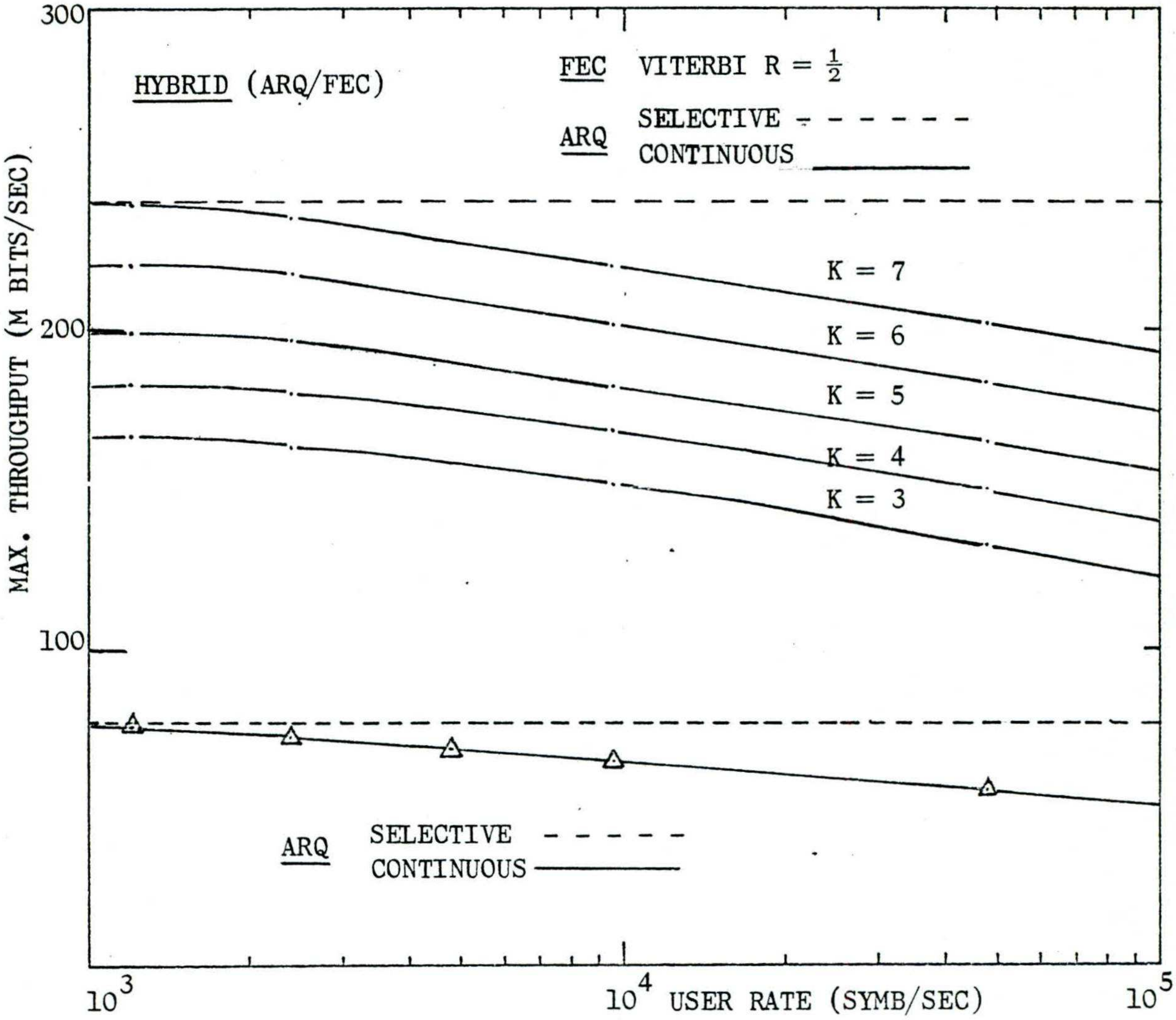


Figure 4.9: MAXIMUM THROUGHPUT AS A FUNCTION OF THE USERS RATES FOR THE ARQ AND HYBRID SCHEMES.

systems, the throughput reaches a maximum value then falls off rapidly. The maximum throughput that could be reached increases with the quality of the super channel, that is as more powerful inner coding is used. Figure (4.7) showns that a substantial improvement is obtained as the constraint length of the code increases from K = 3 to K = 7.

Results with several users rates using continuous ARQ and selective ARQ systems in conjunction with a rate $\frac{1}{2}$, K = 7 convolutional code are given in Figure (4.8). The curves illustrate the relative sensitivities of the these two ARQ schemes to the users rates together with the potential advantages of using the selective scheme. Finally, a comparison of the maximum throughput values achieved as a function of the users rates are given in Figure (4.9) for the ARQ alone and hybrid schemes. It shows that the concatenation of Viterbi decoding with the ARQ system allows (for the example treated here), a multiplication of the throughput of the ARQ system by a factor of 3. Of course this throughput gain must be weighed against the additionnal hardware requirement and bandwidth extension, but it represents a far better utilization of the satellite link than the ARQ or FEC techniques alone.

V. CONCLUSIONS

The analysis and results presented in this report demonstrate that for very reliable data transmission through a satellite link, the traditional FEC and ARQ techniques may not be the most efficient in terms of effective information rates. The new hybrid ARQ/FEC scheme takes advantage of both techniques and yields a considerable improvement in the achievable throughput. It must be noted that although the computations were carried out for a specific link with sufficient bandwidth using convolutional coding and the most powerful (Viterbi) decoding, the analysis may be applied to many other coding schemes. Moreover the results may be considered as being upper bounds since the traffic due to ACK/NACK information, control information and other overhead engineering services was not accounted for. Also, the so-called return channels were considered error-free, and therefore, the results should only be interpreted as illustrations of the power of the hybrid schemes. The model can certainly be extended to accomodate practical limitations.

The improvement obtained by the new hybrid scheme with the K = 3 code suggests investigating for the inner FEC system less powerful coding-decoding techniques that require a smaller bandwith expansion than the rate $\frac{1}{2}$ Viterbi decoding. For example, a rate 3/4 code with Viterbi decoding may yield an even higher throughput value at lower burst transmission rates than the rate $\frac{1}{2}$ code together with a smaller bandwidth. For bandwidth limited links with small earth stations, simpler FEC schemes such as rates 3/4 or 7/8 threshold decodable codes may be very promising.

From practical point of views, other aspects of the problem such as buffer requirements and management strategies must be considered. In the restricted framework of earth station to earth station transmission, the hybrid scheme presents no more buffer requirements nor buffer management complexity than the ARQ scheme alone. However, in general the users are located far away from the earth station. Terrestrial links are thus required to connect them to the earth stations. In this more generalized framework, mixing terrestrial and satellite links may not be the best approach to the problem because the overall compound channel may not be easily modelled. A simpler approach consists in separating the user-to-user compound channel into three segments: user to earth station, earth station to earth station and earth station to user. ARQ schemes would be used on the terrestrial segments and FEC or hybrid schemes on the satellite link. The segment coordination protocoles, buffer requirements and buffer management at the earth stations present interesting subjects for further studies.

APPENDIX A

TDMA SYSTEMS

In TDMA the station's transmitted information is separated into discrete time slots, with each station assigned a particular time slot or block of time slots. All earth terminals are transmitting through the same channel of the satellite repeater and they are using the same transmitter frequency. Thus, the transmissions from the earth terminals are interleaved in time. An advantage to such a system is that only one modem is needed for each earth terminal. Moreover TDMA systems do not suffer from intermodulation products which reduce the useful signal power as in FDMA systems.

Figure [A1] depicts a typical TDMA frame format. The following nomenclature is used:

Time slot: That portion of time assigned for transmission to a station.

Frame rate: Number of times per second a specific time assignment is repeated.

Burst rate: The data rate at which a station transmits in its assigned time slot. That is the stations transmit in bursts and the bursts are repeated at the frame rate. The input information rate, burst rate, frame rate and time slot duration are all interrelated.

Guard time: A small portion of time between successive time slots.

Preamble: A multipurpose digital word inserted at the beginning of each transmission. The preamble consists of known data patterns used to acquire and track carrier phase, and bit/symbol timing used in coherent demodulation.

The data portion of the bursts are generally composed of one or more sub-bursts. Each sub-burst may be directed toward a particular user. The transmitted data rate or burst rate need not be the same for all sub-bursts. In general the transmitted sub-bursts within a burst will be grouped together in order of ascending burst rates.

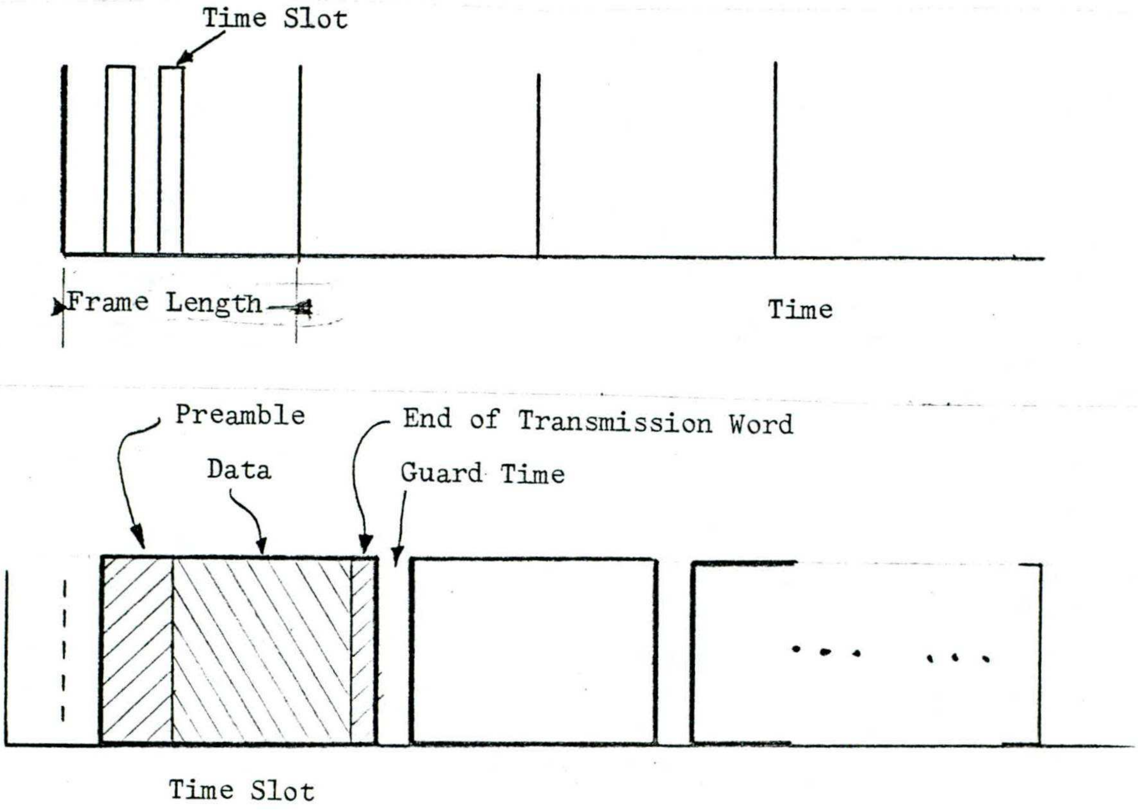


Figure A1

GENERAL TDMA FORMAT

At the sending station data arrives asynchronously from different users and enters a buffer where discrepancies between users clock and divided version of the TDMA clock are resolved. Following this interface a burst compression buffer stores enough data for one sub-burst. The process is repeated for all sub-bursts. The TDMA multiplexer controls the output of the compression buffers and combines them with the preamble to form a station burst. The burst so formed may be encoded, then used to modulate a carrier to form the transmitted signal.

At the receiver the same process is repeated in reverse. Demultiplexing is followed by a burst expansion buffer so that sub-bursts are converted in a continuous data stream.

The received power-to-noise ratio is given by

$$P/N_o = E\ RP + G/T - P_1 - k - M_s - M_i \qquad (A1)$$

where

$E\ RP$ = Effective radiated power (DbW)

G/T = Gain-temperature ratio for the receiving terminal(dB/°K)

P_1 = Path loss at the down-link frequency

k = Boltzmann's constant (-228.6 dBW/°K-Hz)

M_s = System margin (including rain margin)

M_i = Equipment implementation margin.

The examples presented here were computed using the following values:

$$E\ RP = 28 \text{ dBW}$$

$$P_l + M_i = 203 \text{ dB}$$

$$M_s = 6 \text{ dB}$$

Substitution in eq. (Al) yields

$$P/N_o = 47.6 + G/T \qquad \text{(A2)}$$

The received bit energy-to-noise ratio is then

$$\frac{E_b}{N_o} = \frac{P}{N_o} - 10 \log_{10} R_s \qquad \text{(A3)}$$

For convenience, the maximum information burst rate R_s was determined throughout by assuming only an ERP constraint with no bandwidth constraint.

APPENDIX B

This appendix considers both sequence error probability and symbol error rate in connection with convolutional encoding and Viterbi decoding. Upper bounds on "first error event" and bit error probabilities, based on a technique due to Viterbi [4], are obtained for a specific code and any binary input discrete memoryless channel. These bounds are further tightened for the additive gaussian noise channel to yield asymptotically accurate values which are used extensively in chapter IV of this report to evaluate the performance of both the FEC system with rate $\frac{1}{2}$ convolutional encoding and Viterbi decoding and the hybrid scheme introduced in section IV.4.

B.1 CONVOLUTIONAL ENCODING AND VITERBI DECODING

The concept of convolutional encoding was introduced in section II.2 and is briefly reviewed here in order to enforce the notation. A binary convolutional code of rate $R = 1/V$ bits/symbols and constraint length K is defined as any finite state linear (on the field GF(2)) sequential machine implying a memory of the K-1 past input symbols. An example of convolutional encoder of rate $\frac{1}{2}$ and constraint length K=3 is represented in Figure (B.1). For this example, at each clock stroke, 2 output binary symbols (also called a branch) are evaluated by scanning the output of the 2 modulo-2 adders and a new data symbol is fed into the shift register. The branch of 2 symbols is then transmitted through the channel. The encoder dynamic can be described through a state transition diagram by defining the state to be the (K-1)-tuple formed by the concatenation of the K-1 rightmost cells of the shift register. The state space is then defined to be the set of all 2^{K-1} (K-1)-tuples and the dynamic behavior of the encoder is then seen to be broken into two steps; a scan which computes the branch value and a shift leading to the new state. These two steps are described by the output and the next state transition maps relating the branch and the next state respectively to the input and present state values. The graphical representation of these two functions is known as the Mealy state transition diagram of the sequential machine equivalent to the encoder, and is represented on Figure (B.2) for the code considered in Figure (B.1). Of primary importance is the fact that the state characterises uniquely the past of the encoder, in the sense that if two different input sequences lead for the first time to the same state value at some step j and are identical thereafter, the corresponding output sequences must be the same starting at step j. Two sequences with this property are called reconvergent sequences at step j. Since the state value is composed of the past (K-1) inputs, in order for two sequences to be reconvergent for the first time at step j, it is necessary and sufficient that they have identical state at step j and different state at step j-1. Hence, these two sequences must agree on the input symbols at step j-1, j-2, ..., j- (K-1) and disagree on the input sumbol at step j-K, or else the reconvergence would have taken place at least at step j-1. This reconvergence property is the key to the trellis representation of a convolutional code. The basic trellis extension diagram combines in a graphical way the reconvergence property and the state transition diagram into two connected components: the first component which corresponds to all the possible states at step j and the second component which all the next states at step j. All states which are connected in a one step extension are then linked by the corresponding branch. The basic trellis extension diagram for the code of Figure(B.1) is represented in Figure (B.3a)). It is important to note that the structure of this graph is actually independent of the code connections, which only determine the branches values. The complete trellis associated with a code is then the graph obtained by successive concatenation of the trellis extension diagram as the depth index j takes on the values 0, 1, 2, ... For synchronization purposes, at step j=0, the machine is set in a predefined state (usually state "0"). Moreover, the depth index set can be finite or infinite, corresponding respectively to a so

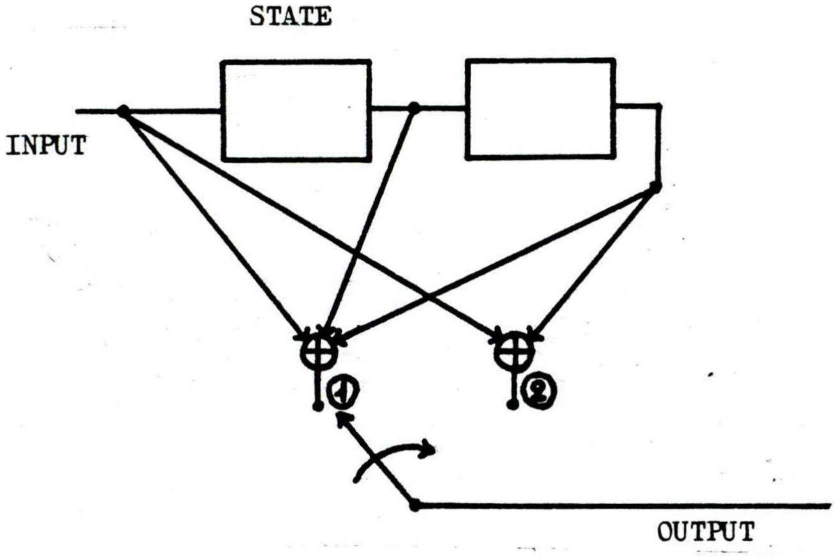
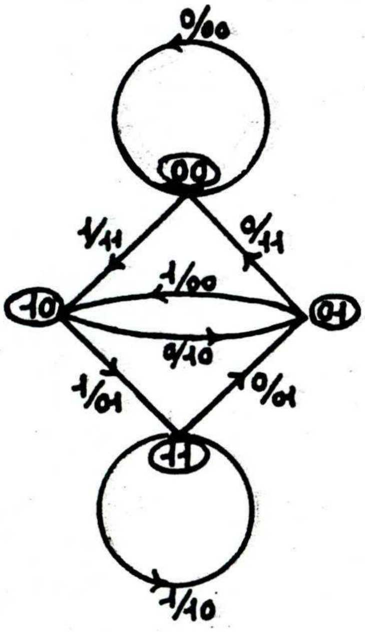


Figure B.1 Example of Rate $\frac{1}{2}$, K = 3 convolutional encoder.



KEY

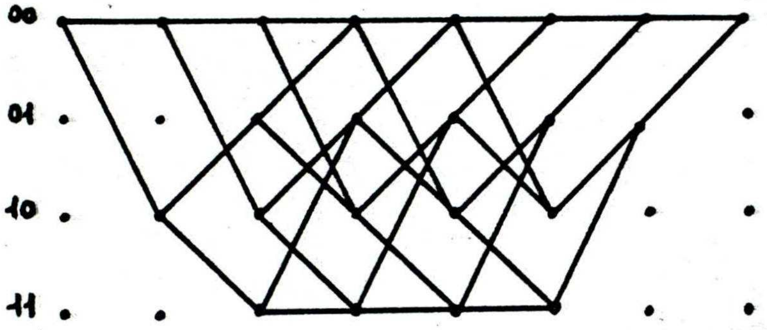
□ State value

i/j^k i input

(j^k) branch value

k = 1, 2, ..., V

Figure B.2 State transition diagram of the code of Figure (B.1).



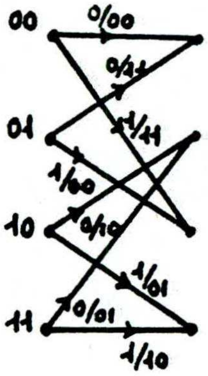


Figure B.3b) Terminated trellis (L=5) for the code of Figure B.1.

Figure B.3a) Basic trellis extension diagram for the code of Figure B.1.

called terminated or unterminated trellis. In the terminated trellis case, which should be considered when transmitting blocks of data, the state of the encoder is usually resynchronized to "0" by appending (K-1) trailing input 0's to the information sequence. This results in a lowering of the coding rate value by a factor $1 + \frac{K-1}{L}$ where L is the input block length, which is usually negligeable for large values of L with respect to K. The terminated trellis corresponding to a code with constraint length 3 and L = 5 is represented on Figure (B.3.b). The Viterbi algorithm for maximum likelihood decoding (MLD) of sequences operates on the trellis diagram as a repetitive procedure exploiting at each step the reconvergence property of the different paths.

Let $U_\ell \triangleq (u_o, u_1, u_2, \ldots, u_\ell)$be a path on the trellis linking successively nodes u_o = "0", $u_1, u_2, \ldots, u_\ell$; where u_m,$0 \leq m \leq \ell$ is the state value associated with the node at depth m on the path U_ℓ. A convenient notational expendency to extend a path will be to use the concatenation operator $U_{\ell+j} \triangleq U_\ell(u_{\ell+1}, u_{\ell+2}, \ldots, u_{\ell+j})$. We then note that the operation of the encoder is then described by a unique path in the trellis associated with the code and decoding is then tantamount, given the received sequence is $\tilde{Y}_\ell$, to find the path which is optimum given a criterion to measure "goodness". Assuming all paths are equally likely and the criterion is the probability of error on sequences; it is well known that an optimum decoder selects one sequence among the set of sequences such that the likelihood function is maximum. If we define the "metric" of the path U_ℓ as the log likelihood ratio

$$\Gamma(U_\ell) \triangleq \log\{\,\text{Prob}(\tilde{Y}_\ell/U_\ell)\,\} \qquad \text{(B.1.1)}$$

a maximum likelihood decoding algorithm (MLD), is then an algorithm which solves the following optimization problem for a terminated trellis of length L.

Determine U_L^* satisfying the initial and terminal conditions such that

$$\Gamma(U_L^*) = \max_{U_L}\{\Gamma(U_L)\} \qquad \text{(B.1.2)}$$

This problem is easily solved by a forward dynamic program in the case of a discrete memoryless channel. In this case we can write (B.1.1) as

$$\Gamma(U_\ell) = \Gamma(U_{\ell-1}) + \log\{\text{Prob}\,(\tilde{y}_\ell/u_{\ell-1},u_\ell)\} \qquad \text{(B.1.3)}$$

where the second term in (B.1.3) represents the branch metric associated with the transition from state $u_{\ell-1}$ to u_ℓ. If we define

$$\gamma_\ell(u_{\ell-1},u_\ell) \triangleq \log\{\text{Prob}(\tilde{y}_\ell/u_{\ell-1},u_\ell)\} \qquad \text{(B.1.4)}$$

(B.1.3) can be rewritten as

$$\Gamma(U_\ell) = \Gamma(U_{\ell-1}) + \gamma(u_{\ell-1},u_\ell) \qquad \text{(B.1.5)}$$

Let X_ℓ be the space of all possibly extended states of the machine associated with the convolutional encoder at depth ℓ; the optimization problem (B.1.2) can then be carried out at any depth ℓ with any terminal state belonging to X_ℓ. Let $U_\ell^{u_\ell}$ be a sequence which ends up at state $u_\ell \in X_\ell$ at depth ℓ, then

$$\max_{U_\ell^{u_\ell}} \{\Gamma^{(U_\ell^{u_\ell})}\} = \max_{U_\ell^{u_\ell}} \{\Gamma^{(U_{\ell-1})} + \gamma_\ell^{(u_{\ell-1},u_\ell)}\} \qquad \text{(B.1.6)}$$

If we define P_{u_ℓ} as the set of predecessors of u_ℓ in the fully extended trellis, (B.1.6) can be readily rewritten as

$$\max_{U_\ell^{u_\ell}}\{\Gamma^{(U_\ell^{u_\ell})}\} = \max_{U_{\ell-1}^{u_{\ell-1}},\ u_{\ell-1} \in P_{u_\ell} \cap X_{\ell-1}} \{\Gamma^{(U_{\ell-1}^{u_{\ell-1}})} + \gamma_\ell^{(u_{\ell-1},u_\ell)}\} \qquad \text{(B.1.7)}$$

$$= \max_{u_{\ell-1} \in P_{u_\ell} \cap X_{\ell-1}} \{\max_{U_{\ell-1}^{u_{\ell-1}}}\{\Gamma^{(U_{\ell-1}^{u_{\ell-1}})}\} + \gamma_\ell^{(u_{\ell-1},u_\ell)}\}$$

(B.1.7) is the basic recurrence formula required to solve the optimization problem (B.1.2). In words it says that among the two possible branches converging onto u_ℓ, only one, satisfying equation (B.1.7) needs to be retained. Evidently, if $X_{\ell-1} \cap P_{u_\ell}$ contains only one predecessor, the branch associated with the transition is kept.

Defining
$$F^\ell_{(u)} \triangleq \max_{U^u} \{\Gamma^{(U_\ell^u)}\} \qquad \text{(B.1.8)}$$

the complete forward dynamic program to solve problem (B.1.2), taking into account both initial and terminal conditions associated with a terminated trellis can be summarized as follows.

```
$\forall \ell = 1, 2, \ldots, L + K - 1$

set $X_\ell = \{u: P_u \cap X_{\ell-1} \neq \emptyset\}$

$\forall u \in X_\ell$

set $F^{\ell}(u) = \max_{u' \in P_u \cap X_{\ell-1}} \{F^{\ell-1}_{(u')} + \gamma_\ell^{(u',u)}\}$ †

select u*, such that $F^{\ell}_{(u)} = F^{\ell-1}_{(u*)} + \gamma_\ell^{(u*,u)}$, as the predecessor of u.

When $\ell = L + K - 1$, the decoded path is then formed as the
backward sequence of predecessors starting at state "0".
```

The program can be readily simplified by noting that $X_\ell \equiv X$ (the space of all possible states of the encoder) $\forall \ell \geq K - 1$, and moreover for any $\ell \leq K - 1$ no reconvergence occurs. If we set a bias value of $-\infty$ on every state that does not belong to X_ℓ, every time such a state is extended and compared with a normally extended state, this extension will

† In case of ties select one of the branches at random or whichever branch is first in a predefined ordering of the branches.

be discarded: moreover if two such states are extended, the corresponding final value of the metric will still be $- \infty$, no matter which path is retained as survivor. Taking into account these remarks and the fact that $F^{\ell}(u)$ does not need to be retained for every value of ℓ, since only the value at $\ell - 1$ is required for extension, the following simple algorithm (Viterbi Maximum Likelihood Sequence Decoder, VMLSD) suitable for both hardware and software implementation can be considered.

VMLSD Algorithm

```
Step 1.     Initialization

            DO  for every $u \in X$
                if $0 \in P_u$ set $F^1(u) = \gamma_1^{(0,u)}$, $p^1(u) = 0$
                else set $F^1(u) = - \infty$
            $\ell = 2$

Step 2.     (Basic trellis extension)

            If $\ell$ is even, set $i = 0$, $j = 1$
            Else set $i = 1$, $j = 0$
            DO  for every $u \in X$
                set $F^i(u) = \max_{u' \in P_u} \{F^j(u') + \gamma_\ell^{(u',u)}\}$, $p^\ell(u) = u^*$
                where $u^*$ is one among the $u'$ which maximize $\{F^j(u')+\gamma_\ell^{(u',u)}\}$
            If $\ell = L + K - 1$ go to Step 3.
            Else set $\ell = \ell \pm 1$ and go to Step 2.
```

Step 3. (Backward decoding of ML path in the trellis)

```
Set $u_{L+K-1} = 0$
DO  $j = 1, L+K-1$ by increment of 1
    $u_{L+K-1-j} = p^{L+K-j}(u_{L+K-j})$
The decoded path is then $[u_o, u_1, \ldots u_L]$, Stop
```

The global amount of storage for this algorithm includes

a) 2 tables F^j $j = 0, 1$ of size #[X] (where #[ ] stands for the cardinal number of a set); assuming a q bits quantization of the metric, this amounts to $2q \times \#[X]$ bits.

b) #[X] linking tables of size $L + K - 1$, $p^j(u)$ to store the surviving predecessor at depth j branching onto state u. For a binary input convolutional code, each entry in these tables is one bit, corresponding to the symbol shifted out during the transition.

Summing a) and b) for a code of constraint length K ($\#[X] = 2^{K-1}$) yields the total storage requirement

$$s = 2^{K-1}[L + K - 1 + 2q] \qquad \text{(B.1.10)}$$

which increases exponentially in K and linearly in L. As a consequence, the previous algorithm is only suitable to the decoding of finite sequences and as such can only be applied to terminated trellis.

Nevertheless, the procedure can be modified for real time applications, operating on an unterminated trellis as follows. Let Δ be a fixed positive integer, called the decoding lag; at each depth $\ell > \Delta$ the VMLSD can be set to select among all survivors a survivor U^*_ℓ which maximizes the metric Γ_{U_ℓ} and puts out as the decoded branch the branch at depth $\ell - \Delta$ on the path U^*_ℓ. With these modifications, the algorithm becomes a Real Time Viterbi Decoder (RTVD) as follows.

<u>RTVD Algorithm</u>

```
Step 1.     (Initialization)

            DO  for every $u \in X$
                If $0 \in P_u$ set $F^1(u) = \gamma_1^{(0,u)}$, $p^1(u) = 0$
                Else set $F^1(u) = -\infty$
                $\ell = 2$

Step 2.     (Basic trellis extension)

            If $\ell$ is even, set $i = 0$, $j = 1$
            Else set $i = 1$, $j = 0$
            DO  for every $u \in X$
                Set $F^i(u) = \max_{u' \in P_u} \{F^j(u') + \gamma_\ell^{(u',u)}\}$,
                Set $m = [\ell - 1, \text{mod.} \Delta] + 1$, $p^m(u) = u^*$
                where $u^*$ is one among the $u'$ which maximizes $\{F^j(u) + \gamma_\ell^{(u',u)}\}$
            Set $\ell = \ell + 1$
            If $\ell \leq \Delta$ go to Step 2.
            Else go to Step 3.
```

<u>Step 3.</u> (Real time decoding)

```
Set $u = u'$ where $u'$ satisfies $F^i(u') = \max_{u \in X} \{F^i(u)\}$
DO $j = 1, \Delta - 1$ by increment of 1
    If $m + 1 - j > 0$, set $n = m + 1 - j$
    Else set $n = \Delta + m + 1 - j$
    Set $u = p^n(u)$
If $m + 1 - \Delta > 0$, set $u' = \overset{m+1-\Delta}{p(u)}$
Else set $u' = p_{(u)}^{m+1}$
The decoded branch at depth $\ell - \Delta$ is the input branch corresponding to the state transition from $u'$ to $u$.
Go to Step 2.
```

It is readily seen that for this algorithm, the total storage requirement for a binary input convolutional code of constraint length K, using a q bit quantization scheme for the metric is

$$s = 2^{K-1}[\Delta + 2q] \qquad \text{(B.1.11)}$$

which still increases exponentially in K (a feature of all Viterbi type decoders) but is independent of L.

The actual implementation of the RTVD can be further simplified when the following property of the algorithm is recognized. All survivors at some step $\ell + \Delta$ are diverging from a common node at some depth $\ell + \Delta - n$, and hence are identical up to depth $\ell + \Delta - n$, where n is a random variable whose distribution is a function of the actual signal upon noise ratio. If Δ is sufficiently large, the probability that n will exceed Δ can be made very small and no significant degradation in performance will result if the branch decoding scheme is based on a majority decision over the set of all surviving branches at depth ℓ. With such a procedure, the backward pointers can be eliminated and the decoding can be performed by a simple shift of the branches followed by a majority logic decision. As such, the modified scheme is particularly efficient when hardware implementation is considered.

B.2 ERROR ANALYSIS OF VITERBI DECODERS

In the sequel we assume that the Viterbi decoder starting node is the all zero state. Let S_ℓ be the set of incorrect paths at step ℓ, which is defined as the set of all finite paths diverging from the correct path at some time $\ell - n$ and reconverging for the first time on the correct path at time ℓ. These paths represent the set of all potential adversaries to be considered in the Viterbi decoder, and will result in an error if and only if one of these paths is more likely than the correct one. This set can be compactly described by a generating function T(D,L,N) whose power series expansion with respect to

L leads to the characterization of each one of the adversaries in S_ℓ by a factor of the form $L^\ell D^d N^n$, where $\ell(\ell \geq K)$ is the number of branches between reconvergence, d and n are respectively the Hamming distance between the adversary and the correct path up to their convergence and the number of discrepencies in the input bit patterns generating these two paths. Since the code is linear, the actual distances, reconvergence lengths as well as bit patterns distributions are actually independent of the transmitted input message, which can then be chosen without any loss in generality as the all 0 bit pattern. The set S_6 relative to the code of Figure (B.1) is represented in Figure (B.4). The associated generating function T(D, L, N) is then easily seen to be the transfer function of the modified state transition graph obtained by opening the branches leaving and leading to state "0", removing the loop at state "0" and assigning to each branch a transfer function of the form $LD^w N^i$ where w is the Hamming weight of the branch and i = 0, 1 depending on whether the branch corresponds to a state transition induced **by** a 0 or a 1 input. For the code of Figure (B.1), we easily obtain from Figure (B.5)

$$T(D,L,N) = \frac{D^5 L^3 N}{1-DL(1+L)N} \qquad (B.2.1)$$

the power expansion of this function with respect to L leads to the weight spectrum of the code

$$T(D,L,N) = D^5L^3N \sum_{k=0}^{+\infty} L^k \sum_{n=\lceil\frac{k}{2}\rceil}^{k} \binom{n}{k-n} D^nN^n \qquad (B.2.2)$$

where $\lceil x\rceil$ stands for the smallest integer $\geq x$ which can be interpreted as follows.

The number of adversaries with a divergence lenght of $3 + k \quad k = 0,1,2,\ldots$ is

$$N_k = \sum_{n=\frac{k}{2}}^{k} \binom{n}{k-n}$$

Among these adversaries, $\binom{n}{k-n}$ have a weight of $5 + n$ and the corresponding input sequences have weight n, $n = \lceil\frac{k}{2}\rceil \;,\; \lceil\frac{k}{2}\rceil + 1, \ldots, k$.

For a terminated trellis, the summation in (B.2.2) must be truncated to take into account the fact that all paths must be starting from node "0" at time 0. With these preliminaries in mind, we are now in a position to start the analysis of errors in a Viterbi decoder. We first consider the terminated trellis case. Let L be the length of the message to which we append K-1 trailing 0's. The output of the encoder is then transmitted over a binary input memoryless channel whose transition matrix is p_{ij} $i = 0, 1; j = 0, 1, \ldots J - 1$ (see Figure (B.6)). Let $(u_o, u_1, u_2, \ldots u_\ell)$ be the sequence of transition states corresponding to the correct path up to length ℓ. A first error event ϵ_1^ℓ is said to occur at step $\ell(\ell \geq K)$

if the correct path is the surviving path at nodes u_i at step i $\forall$ i $< \ell$ but is not the surviving path at state u_ℓ, during the trellis extension at step ℓ. It is clear that ϵ_1^ℓ can be bounded as

$$P(\epsilon_1^\ell) \le \text{Prob}\{\text{for some } y' \in S_\ell \text{ and surviving at step } \ell\text{-1: } \Gamma^\ell(y') \ge \Gamma^\ell(y_o)\}^* \qquad \text{(B.2.3)}$$

where y_o stands for the correct path, and $\Gamma^\ell(y')$ and $\Gamma^\ell(y_o)$ stand for the likelihood ratios of y' and y_o limited to the branches in which they diverge.

Using the union bound, the right hand side of (B.2.3) can be overbounded to yield

$$P(\epsilon_1^\ell) \le \sum_{y' \in S_\ell} \text{Prob}\{\Gamma^\ell(y') \ge \Gamma^\ell(y_o) \,|\, y' \text{ being a survivor at step } \ell\text{-1}\} \qquad \text{(B.2.4)}$$

now Prob$\{\Gamma^\ell(y') | y'$ being a survivor at step ℓ-1$\}$ is the probability of error between two code words which diverge over $n_{y'}$ branches.

Let $d_{y'}$ be the Hamming distance between y' and y_o. Each term in B.2.4 can then be overbounded (Gallager [3 , page 129]) by

$$\text{Prob}\{\Gamma^\ell(y') \ge \Gamma^\ell(y_o) \,|\, y' \text{ being a survivor at step } \ell\text{-1}\}$$
$$\le \min_{0<s<1} \{(\sum_{j=0}^{J-1} p_{o_j}^{1-s}\, p_{1_j}^{s})\}^{d_{y'}} \qquad \text{(B.2.5)}$$

[*] This is indeed an overbound since $\Gamma(y') = \Gamma(y)$ does not always results in an error, depending on the tie resolving mechanism.

if we let $D_o \triangleq \min_{0<s<1} \left(\sum_{j=0}^{J-1} p_{o_j}^{1-s} \, p_{1_j}^{s} \right)$ (B.2.6)

we have

$$P(\epsilon^{\ell}_{1}) \leq \sum_{y' \in S_{\ell}} (D_o)^{d_{y'}} \qquad \text{(B.2.7)}$$

the Hamming distance distribution in S_{ℓ} is easily obtained as T

$$T^{\ell}_{w}(D) \triangleq T^{\ell}(D,1,1)$$

where $T^{\ell}_{w}(D,L,N)$ is the truncated generating function up to lenght ℓ, so that

$$P(\epsilon^{\ell}_{1}) \leq T^{\ell}_{w}(D_o) \qquad \text{(B.2.8)}$$

the right hand side of B.2.7 can be increased by adding more terms in the summation to yield the uniform bound on $P(\epsilon^{\ell}_{1})$

$$P(\epsilon^{\ell}_{1}) \leq T_{w}(D_o) \qquad \text{(B.2.9)}$$

where

$$T_{w}(D) \triangleq T(D,1,1)$$

It is now easy to overbound the probability of sequence error P_E for a terminated trellis of lenght $L + K - 1$. Let P_C be the probability of correct decoding of the sequence.

$$P_C = \prod_{\ell=K}^{L+K-1} (1 - P(\epsilon^{\ell}_{1})) \qquad \text{(B.2.10)}$$

hence

$$P_C \geq \prod_{\ell=K}^{L+K-1} (1 - T_w(D_o)) = (1 - T_w(D_o))^{L-1} \qquad \text{(B.2.11)}$$

and we can write

$$P_E = 1 - P_C \leq 1 - (1 - T_w(D_o))^{L-1} \qquad \text{(B.2.12)}$$

if $T_w(D_o)$ is small with respect to 1, we have the following approximation

$$P_E \overset{\sim}{\leq} (L-1)\ T_w\ (D_o) \qquad \text{(B.2.13)}$$

Now we consider the case of a very long trellis or an unterminated trellis. In this context, the probability of sequence error P_E is of no significance since as the message length L increases P_E will eventually tend to 1. The parameter of interest in this case is the bit error rate P_B which can be defined as

$$P_B = \lim_{L\to\infty} \frac{E\{\#\text{ of bit errors in a message of length L}\}}{L} \qquad \text{(B.2.14)}$$

Moreover, in this context the decoded path will diverge and then reconverge on the correct path many times. An error event ϵ of length M is said to occur at time ℓ if the decoded path diverges from the correct path at time ℓ and reconverges for the first time on the correct path M branches later. The importance of error events is due to the fact that errors in the very long or unterminated trellis case are made of disjoint error events which are then statistically independent if the channel is memoryless. It is readily seen that an error event will always result in one or more bit errors being generated by the decoder, resulting in the fact that the bit error sequence will have a bursty behavior within each of the error events. The analysis of bit errors within error events is fairly complicated, essentially due to the fact that the Viterbi decoder is not an optimum bit decoder. As such, decisions at the bit level depend not only on the past and present history of the decoder but also on the future behavior of the noise sequence. For a real time Viterbi decoder, the future to be considered is finite and limited to the decoding lag Δ. However, consistently with the idea that for an unterminated trellis the analysis should be carried per branch extension or unit time we define the probability of an error event per unit time at time ℓ as the probability that the state "0" surviving path at time ℓ-1 will not be the surviving path during the next extension at state "0". We can indeed assume without loss of generality that the all "0"s path is transmitted since error paths are always made of paths in the incorrect set S_m at some step m. Two cases must be considered depending on whether the path which is discarded is the correct path or an incorrect path. With the first assumption, the probability is simply the first error probability which has been uniformly bounded by (B.2.9). Under the second assumption, a path in S_ℓ is compared with a path in $S_{\ell-1}$ and results in a larger likelihood. Now the surviving path in $S_{\ell-1}$ must be more likely than the correct path up to time ℓ-1 or else it would not have survived at step ℓ-1. This implies that if y'' is the survivor at time ℓ, y' and y_o are respectively the survivor at step ℓ-1 and the correct path, their likelihood functions up to time ℓ must satisfy

$$\Gamma(y'') \geq \Gamma(y') \geq \Gamma(y_o)$$

and hence the probability that the likelihood of y'' exceeds the likelihood of y' is smaller than the probability that the likelihood of y'' exceeds the likelihood of y_o, which is upper bounded by (B.2.9). As a consequence the probability of an error event per unit time is always uniformly bounded by (B.2.9)

$$P_{\epsilon} \mid \text{unit time} \leq T_w(D_o) \qquad \text{(B.2.15)}$$

In order to bound the bit error rate, we also consider a per unit time event, namely the event P_ℓ of an error event terminating at time ℓ. Conditionnally on P_ℓ it is easy to compute a bound on the expected number of bit errors which will be put out by the decoder during this error event. By hypothesis, the decoded path belongs to S_ℓ so that the expected number of bit errors can be bounded by weighting each term in (B.2.7) by the number of bit errors $N_y{}'$ in the adversary.

$$E\{\# \text{ of errors } |P_\ell\} \leq \sum_{y' \in S_\ell} N_y{}' \, D_o^{d_y{}'} \qquad \text{(B.2.16)}$$

now let $T_{w,n}(D,N) \triangleq T(D,1,N)$ be the transfer function of the set S_ℓ including only the information relative to the weights and generating input bit sequences in S_ℓ. We can write:

$$T_{w,n}(D,N) \triangleq \sum_{k=k_{min}}^{+\infty} \sum_{m=1}^{m_k} a_{m,k} D^k N^m \qquad (B.2.17)$$

where $a_{m,k}$ represents the number of paths in S_ℓ of weight k and resulting from an input sequence of weight m

$$\frac{dT_{w,n}(D,N)}{dN} = \sum_{k=k_{min}}^{+\infty} \sum_{m=1}^{m_k} m \times a_{m,k} D^k N^m \qquad (B.2.18)$$

and

$$\left.\frac{dT_{w,n}(D,N)}{dN}\right|_{N=1, D=D_o} = \sum_{k=k_{min}}^{+\infty} b_k D_o^k \qquad (B.2.19)$$

where

$$b_k \triangleq \sum_{m=1}^{m_k} m \times a_{m,k} \qquad (B.2.20)$$

represents the total # of 1's in the sequences belonging to S_ℓ and of weight k. The right hand side of (B.2.18) is readily seen to be equivalent ot (B.2.17) through a reordering of the terms (The series in B.2.16 converges absolutely), yielding

$$E\{\text{# of bit errors} | P_\ell\} \leq \left.\frac{dT_{w,n}(D,N)}{dN}\right|_{N=1,\ D=D_o} \qquad (B.2.21)$$

since conditioning can only increase the expected value in (B.2.21), we can bound the expected number of errors/unit time as

$$E\{\#\text{ of bit errors}\mid\text{unit time}\} \leq \left.\frac{dT_{w,n}(D,N)}{dN}\right|_{N=1,D=D_o} \qquad (B.2.22)$$

the expected # of bit errors in a message of length L can then be overbounded by multiplying (B.2.22) by L. (In this process we indeed obtain an overbound by overcounting certain parts of bit error sequence which are common to S_ℓ, $S_{\ell+1}$, ...). The uniform bound on P_B results by including (B.2.22) multiplied by L in (B.2.14)

$$P_B \leq \left.\frac{dT_{w,n}(D,N)}{dN}\right|_{N=1,D=D_o} \qquad (B.2.23)$$

At this point it is easy to extend all preceding results to more general convolutional codes. A binary convolutional code of rate $R = \frac{b}{v}(b\geq 1, v>b)$ and constraint length K is defined as any linear finite state machine in which the input is any binary b - tuple , and the output is a binary v - tuple which depends linearly over GF(2) on the present input and the past (K-1) inputs. Figure (B.7) illustrates the concept for a code (R = 2/3, K = 2). These codes have a state transition diagram in which the state is defined as the content of the b(K-1) rightmost cells of the shift register [see Figure (B.8)]. The main difference with the rate

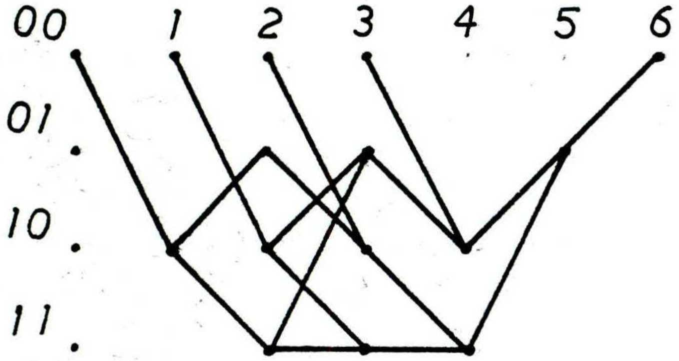


Fig. B.4 Incorrect set at depth 6 for the code of Fig.(B.1).

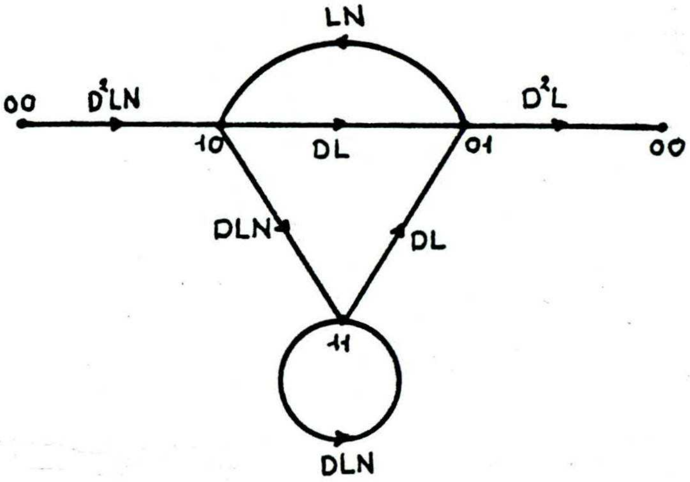


Fig. B.5 Graph for the computation of T(D,L,N) for the code of Fig.(B.1).

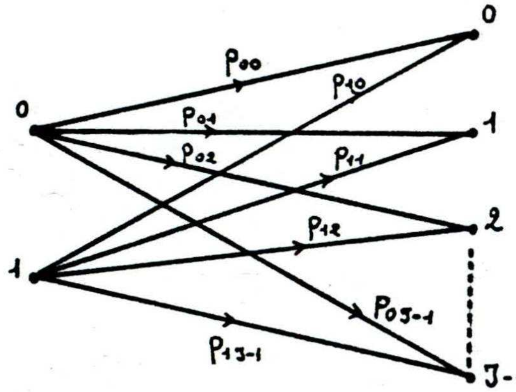


Fig. B.6 General binary input discrete memoryless channel.

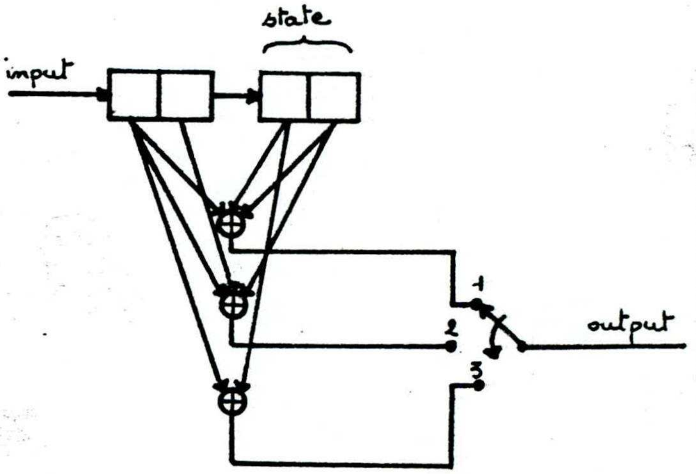


Fig. B.7. Binary convolutional code R = 2/3, K = 2.

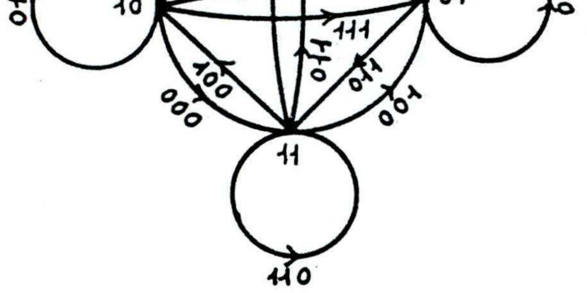


Fig. B.8 State transition graph for the rate 2/3 convolutional code of Fig.(B.7).

$\frac{1}{2}$ case resides in the fact that 2^b branches now diverge and reconverge at each state. A trellis diagram can also be defined from the state transition in a very similar to the rate $\frac{1}{2}$ case. All Viterbi decoders are indeed valid replacing input bits by input binary b - tuples. The bounds on P_{ϵ_1}, P_E are also applicable with the appropriate transfer function T(D,L,N). The only difference resides in the evaluation of P_B. Indeed for a message of length L branches, Lb input bits are transmitted over the channel, and as a consequence (B.2.14) must be divided by b. This modification results in the fact that P_B in (B.2.23) must also be divided by b.

B.3 APPLICATION OF THESE BOUNDS TO THE BINARY INPUT AWGN CHANNEL.

We will now illustrate the application of the bounds found previously to the Additive White Gaussian Noise (AWGN) channel with infinitely fine quantization, and bipolar signaling on the channel. It turns out that in this case the bounds can be further refined. Let E_S be the energy per transmitted channel symbol. If E_b is the energy per transmitted bit and N_o is the one sided spectral density of the noise, we can express the signal upon noise ratio $\frac{E_S}{N_o}$ as

$$\frac{E_S}{N_o} = \frac{R\,E_b}{N_o} \qquad (B.3.1)$$

where R is the code rate $R \triangleq b/v$.

In comparing two code words y_o and y' which differ in n_y' symbols, the probability of error can be exactly computed as (see Wozencraft and Jacobs [2])

$$P^{\ell}_{E} = Q\left(\sqrt{\frac{2n_y{}'E_S}{N_o}}\right) \qquad (B.3.2)$$

where

$$Q(x) \triangleq \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{-x} \exp - \frac{u^2}{2} du$$

this exact expression can be used in place of $D_o^{n_y'}$ in (B.2.7) and (B.2.16) to yield using the definitions (B.2.17) and (B.2.20)

$$P(\epsilon^{\ell}_1) \leq \sum_{k=k_{min}}^{+\infty} a_k \, Q\left(\sqrt{\frac{2kE_S}{N_o}}\right) \qquad (B.3.3)$$

and

$$P_B \leq \sum_{k=k_{min}}^{+\infty} b_k \quad Q\left(\sqrt{\frac{2kE_S}{N_o}}\right) \qquad (B.3.4)$$

where $a_k \triangleq \sum_{m=1}^{m_k} a_{m,k}$ is the number of paths in the incorrect set of weight k, and b_k has been defined as the total number of 1's in all the adversaries of weight k. Let the free distance of the code d_f be defined as the minimum weight of the paths in the incorrect set. Then k_{min} in (B.3.3) and (B.3.4) is equal to d_f. From the well known inequality (Odenwalder [14]),

$$Q(\sqrt{y + x}) \leq \exp - \frac{y}{2} \; Q(x) \qquad (B.3.5)$$

we can rewrite (B.3.3) and (B.3.4) as

$$P(\epsilon^{\ell}_{1}) \leq Q\left(\sqrt{\frac{2d_f RE_b}{N_o}}\right) \sum_{\ell=0}^{+\infty} a_{\ell+d_f} \exp\left(- \frac{2\ell RE_b}{N_o}\right) \qquad (B.3.6)$$

$$P_B \leq \frac{1}{b} \; Q\left(\sqrt{\frac{2d_f RE_b}{N_o}}\right) \sum_{\ell=0}^{+\infty} b_{\ell+d_f} \exp\left(- \frac{2\ell RE_b}{N_o}\right) \qquad (B.3.7)$$

or equivalently

$$P(\epsilon^{\ell}_{1}) \leq Q\left(\sqrt{\frac{2d_f RE_b}{N_o}}\right) \exp\left(\frac{2d_f RE_b}{N_o}\right) T_w(\exp - \frac{2RE_b}{N_o}) \qquad (B.3.8)$$

$$P_B \leq \frac{1}{b} \; Q\left(\sqrt{\frac{2d_f RE_b}{N_o}}\right) \exp\left(\frac{2d_f RE_b}{N_o}\right) \frac{dT_{w,n}(D_o, N_o)}{dN} \qquad (B.3.9)$$

where $N_o = 1, \quad D_o = \exp - \frac{2RE_b}{N_o}$

REFERENCES

[1] Shannon, C.E., "A Mathematical Theory of Communication", Bell System Technical Journal, Vol. 27, July and Oct. 1948.

[2] Wozencraft, J.M., and Jacobs, I.M., "Principles of Communication Engineering", John Wiley, N.Y. 1965.

[3] Gallager, R.G., "Information Theory and Reliable Communication", John Wiley, N.Y. 1968.

[4] Viterbi, A.J., "Convolutional Codes and Their Performance in Communication Systems", IEEE Trans. on Communication Technology, Vol. Com-19, pp. 751-772, Oct. 1971.

[5] Jacobs, I.M., "Sequential Decoding for Efficient communication from Deep Space", IEEE Trans. on Communication Technology, Vol. Com-15, pp. 492-501, August 1967.

[6] Heller, J.A., and Jacobs, I.M., "Viterbi Decoding for Satellite and Space Communication", IEEE Trans. on Communication Technology, Vol. Com-19, pp. 835-848, Oct. 1971.

[7] Massey, J.L., "Threshold Decoding", M.I.T. Press, 1963.

[8] IBM, "Binary Synchronous Communications", report GA27-3004.

[9] Martin, J., "Teleprocessing Network Organization", Prentice Hall, 1970.

[10] Peterson, W., Weldon, E.J. Jr., "Error Correcting Codes", 2nd Edition; the M.I.T. press, 1972.

[11] Balkovic, M.D., Muench, P.E., "Effect of Propagation Delay Caused by Satellite Circuits on Data Communication Systems that use Block Retransmission for Error Correction", pp. 29-36, ICC 1969, Boulder.

[12] Sastry, A.R.K., "Improving Automatic Repeat-Request (ARQ) Performance on Satellite Channels Under High Error Rate Conditions", IEEE Trans. on Communication Technology, Vol. Com-23, pp. 436-439, April 1975.

[13] Benice, R.J., Frey, A.H. Jr., "An Analysis of Retransmission Sy Systems", IEEE Trans. on Communication Technology, Vol. Com-12, pp. 135-145, Dec. 1964.

[14] Odenwalder, J.P., "Optimal Decoding of Convolutional Codes", Ph.D. Dissertation, Depart. of Elect. Eng., U.C.L.A., Jan. 1970.