AD 738213

AD No.____
DDC FILE COPY

[redacted]

LINKABIT CORPORATION

10453 ROSELLE STREET
UNIVERSITY INDUSTRIAL PARK
SAN DIEGO, CALIFORNIA 92121

[redacted]

PERFORMANCE STUDY OF VITERBI DECODING AS RELATED TO SPACE COMMUNICATIONS

By I.M. Jacobs and J.A. Heller

with contributions by
K.S. Gilhousen
J.E. Dunn

Final Technical Report
On Contract No. DAAB07-71-C-0148
(CLIN 0008) -- Sequence No. E002

Submitted to

United States Army
Satellite Communications Agency
Fort Monmouth, New Jersey 07703

August 31, 1971

Submitted by

LINKABIT CORPORATION
10453 Roselle Street
San Diego, California 92121

[redacted]

I

→ The following figures are deleted —

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF FIGURES

TABLE OF FIGURES (Continued).

TABLE OF FIGURES (Continued).

TABLE OF TABLES

1.0 INTRODUCTION

LINKABIT Corporation has examined a wide range of system problems and alternatives pertinent to the application of Viterbi decoding to TDMA satellite communication. Proper resolution of questions concerning the placement of the Viterbi decoder in the ground terminal required augmenting the breadth of the study to include burst buffer designs and TDMA control implementation. The broader viewpoint has provided a substantial payoff, yielding a powerful yet economic design for the TDMA-buffer-decoder-control subsystem. Details are provided in Section 5.0.

A number of significant parameters affect the performance of a Viterbi decoder in a TDMA satellite communication environment. These parameters, including constraint length, code rate, bit error probability, received data quantization, decoder delay, and details of decoder operation are examined in Section 2.0. Extensive analytic and computer simulation results are presented to provide the basis for a complete system design. Modem imperfections can cause significant degradations; decoder tolerance of AGC, bit timing, and carrier phase tracking errors are determined and operating margins specified.

Implementation of high speed, high performance Viterbi decoders involves a range of sophisticated design techniques. Various high speed decoder configurations are reviewed in Section 3 to set the stage for a meaningful determination of speed cost and parts count tradeoffs involving TTL and MECL logic family implementations. Time-sharing and paralleling of decoders is described; both techniques are required for cost-effective system realization. Reliability and maintainability aspects of the decoder implementations are considered for decoders of various rates. The high power dissipation requirements and parts count of high speed (greater than 30 Mbps) decoders raises significant problems in achieving high availability; such decoders are not recommended for the stage 2 system.

Section 6 deals with the problem of modifying a modem to provide soft quantized matched filter outputs. In particular, a circuit design is presented to adapt the Philco-Digi-Phase modem to provide an 8 level quantized output. Finally Section 7 presents study conclusions and recommendations regarding a cost-effective decoder configuration for TDMA operation on the DSCS.

This report is divided into two parts. Part I deals with the performance, complexity and other properties of Viterbi decoders - more or less independent of TDMA considerations. Areas discussed in Part II are pertinent to Viterbi decoder-TDMA system considerations.

PART I - VITERBI DECODING

2.0 VITERBI DECODER PERFORMANCE

In this section the theory of convolutional encoding and Viterbi decoding is briefly reviewed. Performance data based on simulation and tight upper bounds on error probability are presented. The effects on performance of varying certain cost sensitive decoder parameters is discussed. Finally code synchronization, AGC requirements and the degradation due to imperfect carrier phase reference is treated.

2.1 Convolutional Encoding

Fig. 2.1 shows a general binary-input-binary output convolutional coder. The encoder consists of a kK stage binary shift register and v mod-2 adders. Each of the mod-2 adders is connected to certain of the shift register stages. The pattern of connections specifies the code. Information bits are shifted into the encoder shift register k bits at a time. After each k bit shift, the outputs of the mod-2 adders are sampled sequentially yielding the code symbols. These code symbols are then used by the modulator to specify the waveforms to be sent over the channel. Since v code symbols are generated for each set of k information bits, the code rate, R_N, is k/v information bits per code symbol, where $k < v$. The constraint length of the code is K, since that is the number of k bit shifts over which a single information bit can influence the encoder output. The state of the convolutional encoder is the contents of the first $k(K-1)$ shift register stages. The encoder state together with the next k input bits uniquely specify the v output symbols.

As an example, a K=3, k=1, v=2 encoder is shown in Fig. 2.2a. The first two coder stages specify the state of the encoder; thus, there are 4 possible states. The codewords, or sequences of code symbols, generated by the encoder for various input information bit sequences is shown in the code "trellis" of Fig. 2.2b. The code trellis is really just a state diagram for the encoder of Fig. 2.2a. The four states are represented by circled binary numbers corresponding to the contents of the first two stages of the encoder. The lines or "branches" joining states indicate state transitions due to the input of single information bits. Dashed and solid lines correspond to "1" and "0" input information bits respectively. The trellis is drawn under the assumption that the encoder is in state 00 at time 0. If the first information bit were a 1, the encoder would go to state 10 and would output the code symbols 11. Code symbols generated are shown adjacent to the trellis branches. As an example, the input data sequence 101 ... generates the code symbol sequence 111000 ...

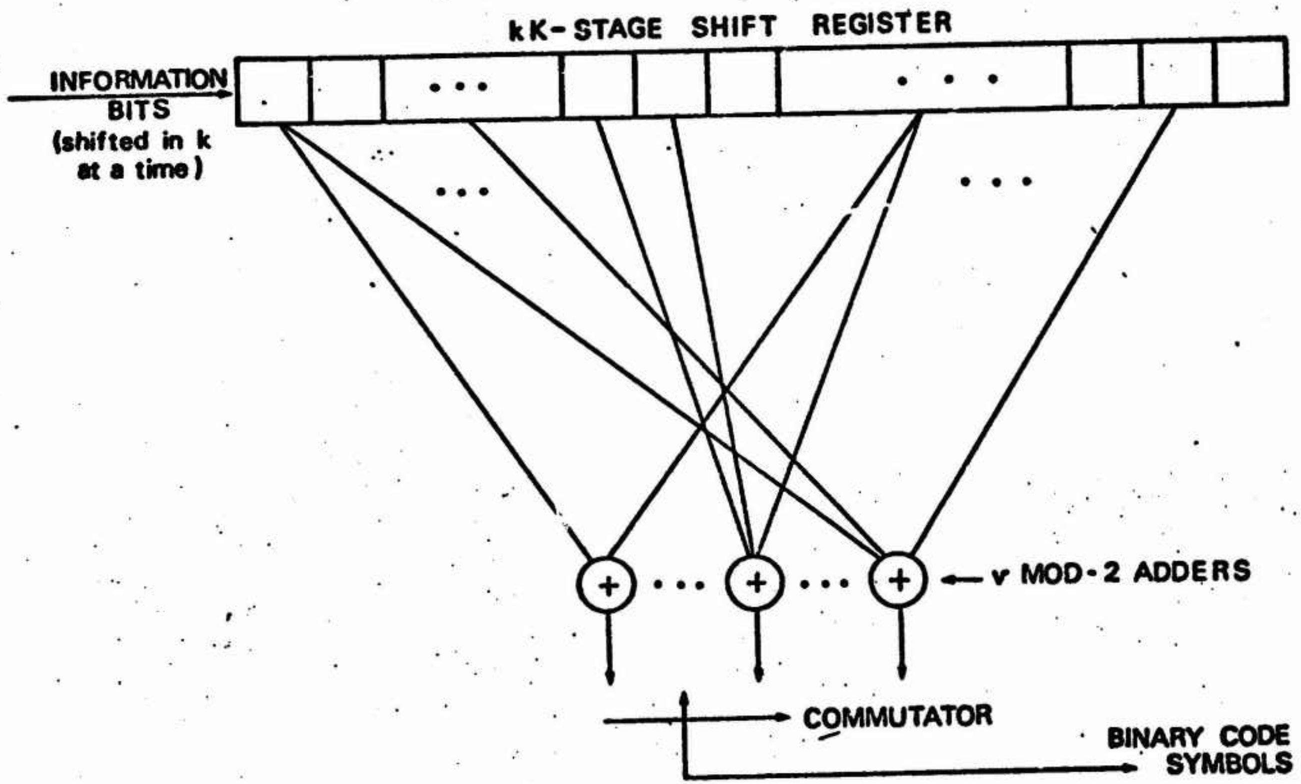
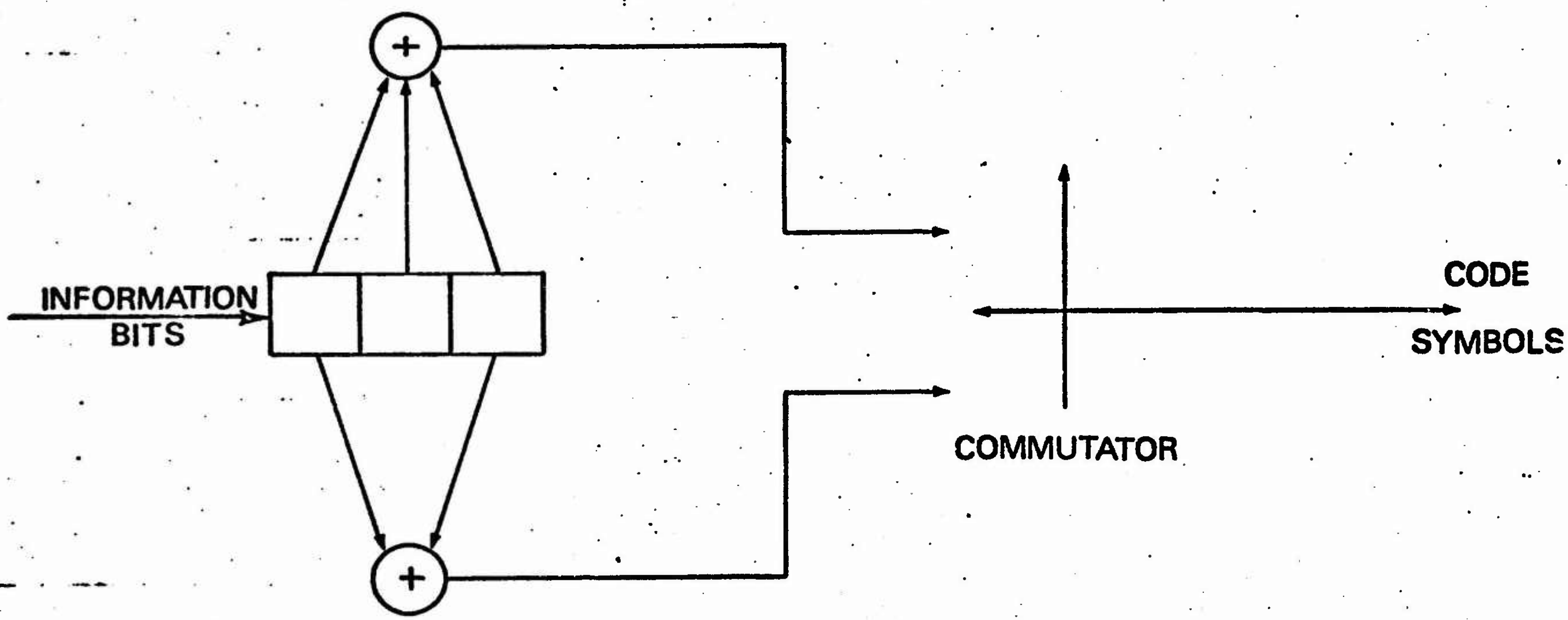
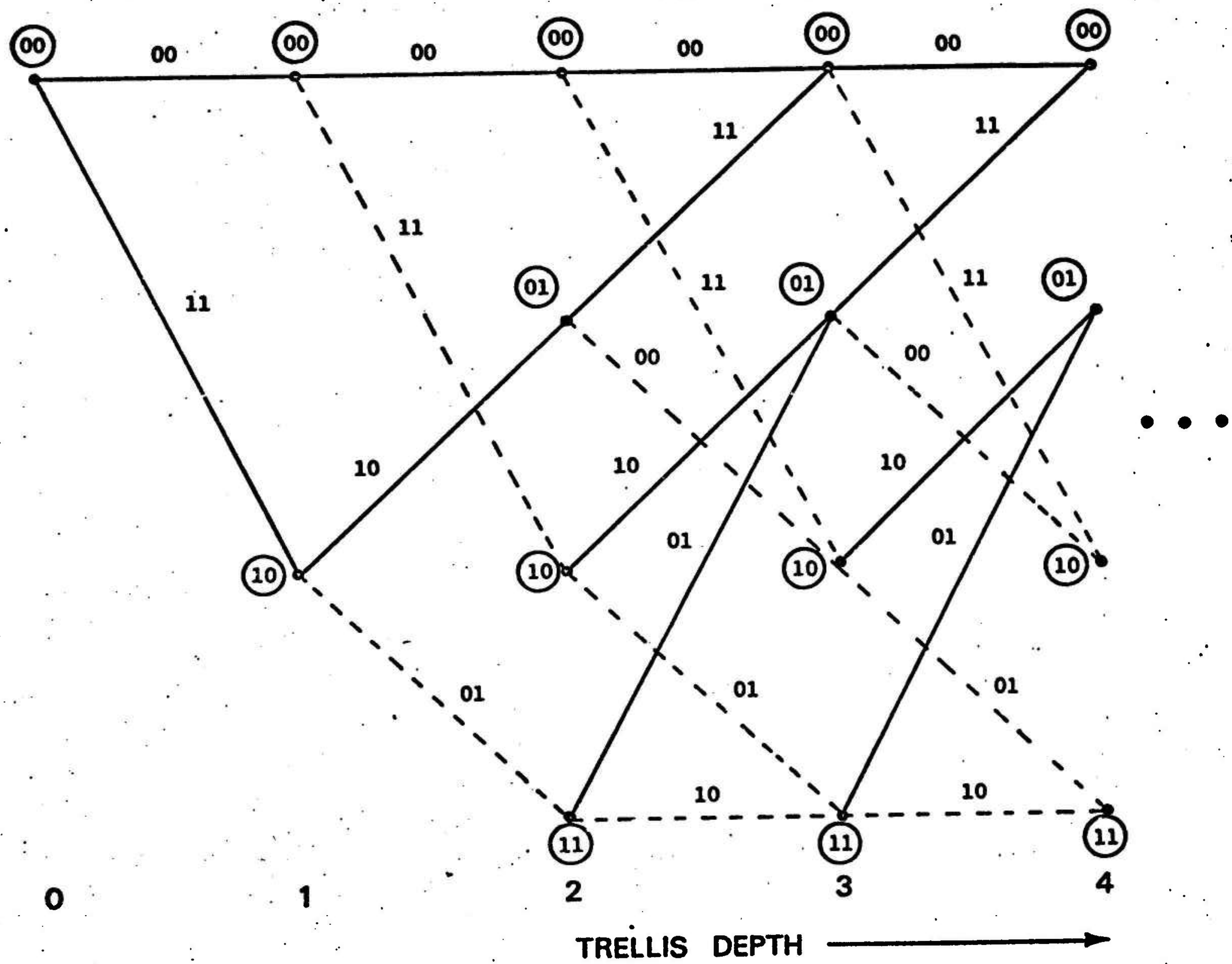


Fig. 2.1 - Rate k/n Convolutional Encoder.



(a)

Fig. 2.2a K=3, R_N=1/2 Convolutional Encoder.



(b)

Fig. 2.2b - Code Trellis Diagram

2.2 Viterbi Decoding

2.2.1 The Basic Algorithm

The maximum likelihood or Viterbi decoding algorithm was discovered and analyzed by Viterbi (Ref. 1) in 1967. Viterbi decoding was first shown to be an efficient and practical decoding technique for short constraint length code by Heller (Ref. 2, 3). The following paragraphs will briefly review the Viterbi decoding algorithm and elaborate on those features and parameters which bear on decoder performance and complexity on satellite channels.

Referring to the code trellis diagram of Fig. 2.2b, a brute force maximum likelihood decoder would calculate the likelihood of the received data for code symbol sequences on all paths through the trellis. The path with the largest likelihood would then be selected, and the information bits corresponding to that path would form the decoder output. Unfortunately, the number of paths for an L bit information sequence is 2^L; thus, this brute force decoding quickly becomes impractical as L increases.

With Viterbi decoding, it is possible to greatly reduce the effort required for maximum likelihood decoding by taking advantage of the special structure of the code trellis. Referring to Fig. 2.2b, it is clear that the trellis assumes a fixed periodic structure after trellis depth 3 (in general, K) is reached. After this point, each of the 4 states can be entered from either of two preceding states. At depth 3, for instance, there are 8 code paths - 2 entering each state. For example, state 00 at level 3 has the two paths entering it corresponding to the information sequences 000 and 100. These path are said to have <u>diverged</u> at state 00, depth 0 and <u>remerged</u> at state 00, depth 3. Paths remerge after 2 (in general $k(K-1)$) consecutive identical information bits. A Viterbi decoder calculates the likelihood of each of the 2^k paths entering a given state and eliminates from further consideration all but the most likely path that leads to that state. This is done for each of the $2^{k(K-1)}$ states at a given trellis depth; after each decoding operation <u>only one path remains leading to each state</u>. The decoder then proceeds one level deeper into the trellis and repeats the process.

For the K-3 code trellis of Fig. 2.2b, there are 8 paths at depth 3. Decoding at depth 3 eliminates 1 path entering each state. The result is that 4 paths are left. Going on to depth 4, the decoder is again faced with 8 paths. Decoding again eliminates 4 of these paths, and so on. Note that in eliminating the less likely paths entering each state, the Viterbi decoder will not reject any path which would have been selected by the brute force maximum likelihood decoder.

The decoder as described thus far never actually decides upon <u>one</u> most likely path. It always retains a set of $2^{k(K-1)}$ paths after each decoding step. Each retained path is the most likely path to have entered a given encoder state. One way of selecting a single most likely path is to periodically force the encoder into a prearranged state by inputting a K-k bit fixed information sequence to the encoder after each set of L information bits. The decoder can then select that path leading to the known encoder state as its (1 bit) output.

The great advantage of the Viterbi maximum likelihood decoder is that the number of decoder "operations" performed in decoding L bits is only $L2^{k(K-1)}$, which is linear in L. Of course, Viterbi decoding as a practical technique is limited to relatively short constraint length codes due to the exponential dependence of decoder operations per bit decoded on K. Fortunately, excellent decoder performance is possible with good short constraint length codes.

2.2.2 Path Memory

In order to make the Viterbi algorithm a practical decoding technique, certain refinements on the basic algorithm are desirable. First of all, periodically forcing the encoder into a known state by using preset sequences multiplexed into the data stream is not always operationally desirable. It can be shown (Ref. 4) that with high probability, the $2^{k(K-1)}$ decoder selected paths will not be mutually disjoint very far back from the present decoding depth. All of the $2^{k(K-1)}$ paths tend to have a common stem which eventually branches off to the various states. This suggests that if the decoder stores enough of the past information bit history of each of the $2^{k(K-1)}$ paths, then the oldest bits on all paths will be identical. If a fixed amount of path history storage is provided, the decoder can output the oldest bit on an arbitrary path each time it steps one level deeper into the trellis. The amount of path storage required, u, is equal to the number of states, $2^{k(K-1)}$ multiplied by the length of the information bit path history per state, h

$$u = h2^{k(K-1)}$$

Since the path memory represents a significant portion of the total cost of a Viterbi decoder, it is desirable to minimize the required path history length h. One refinement which allows for a smaller value of h is to use the oldest bit on the most likely of the $2^{k(K-1)}$ paths as the decoder output, rather than the oldest bit on an arbitrary path. It has been demonstrated theoretically and through simulation (Ref. 4) that a value of h of 4 or 5 times the code constraint length is sufficient for negligible degradation from optimum decoder performance. Simulation results showing performance degradation incurred with smaller path history lengths are presented and discussed in Section 2.3.

2.2.3 State and Branch Metric Quantization

The path comparisons made for paths entering each state require the calculation of the likelihood of each path involved for the particular received information. Since the channel is memoryless, the path likelihood function is the product of the likelihoods of the individual code symbols.

$$P(\underline{r}^*/\underline{x}^{\ell}) = \prod_j P(r_j^*/x_j^{\ell})$$

where $\underline{r}^* = (r_1^*, r_2^*, \ldots, r_j, \ldots)$ is the vector of quantized receiver outputs and $\underline{x}^{\ell} = (x_1^{\ell}, x_2^{\ell}, \ldots, x_j^{\ell}, \ldots)$ is the code symbol vector for the ℓ^{th} trellis path. In order to avoid multiplication, the algorithm of the likelihood is a preferable path metric.

$$M_{\ell} = \log P(\underline{r}^*/\underline{x}^{\ell})$$

$$= \sum_j \log P(r_j^*/x_j^{\ell}) \;\underline{\Delta}\; \sum_j m_j^{\ell} \;,$$

where M_{ℓ} is the metric of the ℓ^{th} path and m_j^{ℓ} is the metric of the j^{th} code symbol on the ℓ^{th} path. With this type of additive metric, when a path is extended by one branch, the metric of the new path is the sum of the new branch symbol metrics and the old path metric. To facilitate this calculation, the path metric for the best path leading to each state must be stored by the decoder as a state metric. This is in addition to the path information bit history storage required.

Viterbi decoder operation can then be summarized as follows, taking the K=3 case of Fig. 2.2 as an example.

a) The metric for the 2 paths entering state 00 are calculated by adding the previous state metrics of states 00 and 01 to the branch metrics of the upper and lower branches entering state 00 respectively.

b) The largest of the two new path metrics is stored as the new state metric for state 00. The new path history for state 00 is the path history of the state on the winning path augmented by a 0 or 1 depending on whether state 00 or 01 was on the winning path.

c) This add-compare-select operation is performed for the paths entering each of the other 3 states.

d) The oldest bit on the path with the largest new path metric forms the decoder output.

Since the code symbol metrics must be represented in digital form in the decoder, the effects of metric quantization come into question. Simulation has shown that decoder performance is quite insensitive to symbol metric quantization. In fact, use of the integers as symbol metrics instead of log likelihoods results in a negligible performance degradation with 2, 4 or 8-level receiver quantization

(Ref. 2, 4). Use of these symbol metrics implies that symbol metrics as well as the received symbols themselves may be represented by 1, 2, or 3 bits for 2, 4, and 8-level receiver quantization respectively.

2.2.4 Unknown Starting State

It has been assumed thus far that a Viterbi decoder has knowledge of the encoder starting state before decoding begins. Thus, in Fig. 2.2b, the starting state is assumed to be 00. A known starting state may be operationally undesirable since it requires that the decoder know when transmission commences. In reality, it has been found through simulation that a Viterbi decoder may start decoding at any arbitrary point in a transmission, if all state metrics are initially reset to zero. The first 3-4 constraint lengths worth of data output by the decoder will be more or less unreliable because of the unknown encoder starting state. However, after about 4 constraint lengths, the state metrics with high probability have values independent of the starting values and steady state reliable operation results.

2.3 Rate 1/2 Convolutional Codes and Viterbi Decoders

Computer simulation of Viterbi decoders is a useful technique for evaluating performance down to a bit error rate of about 10^{-4} to 10^{-5}, depending on code constraint length. Simulations at lower error rates require prohibitively long computer runs to obtain meaningful data. Fortunately, an upper bound on both event and bit error rates has been derived which is very tight for error rates of about 10^{-5} and lower. A combination of the simulations and the numerically evaluated upper bound presented here provides a complete picture of Viterbi decoder performance over a wide range of error rates.

2.3.1 Good Convolutional Codes

One obvious criterion for selecting codes is bit error probability. Unfortunately, obtaining bit error probability through simulation is too time consuming to be used as a method of sifting through a large number of convolutional codes. A much more useful measure of a code is its minimum free distance. As used in this report, the free distance between two code words is the Hamming distance between them from the state in the trellis at which they diverge (the point at which the information bits begin to differ), to the state where they remerge (after K-1 identical information bits). A set of large free distances between the correct code path and the competing incorrect paths is desirable with Viterbi decoding. This is because the greater the free distance, the more channel errors must occur in order for an incorrect path to look more likely than the correct path.

The minimum free distance, d_f, is the smallest value of free distance between the correct path and any other path. Since the codes under consideration are linear codes, the set of distances from any codeword to all other codewords is the same as the set of distances from the all zeros codeword to all other codewords. Thus, d_f is the minimum of the weight of all codewords from the point at which they diverge from, until the point at which they remerge to, the all zeros path. Often, but not always, the minimum weight path corresponds to an information sequence with a single 1 in it. The codeword associated with this sequence diverges from the all zeros path where the information 1 occurs, and remerges K-1 branches later. This, of course, is the shortest length over which two distinct paths can be diverged.

Using the algebraic properties of linear group codes, an upper bound on the minimum free distance of a convolutional code, as a function of constraint length, has been found (Ref. 2, 5). For rate 1/n nonsystematic codes, the bound is

$$d_f \leq \min_h \frac{2^{h-1}}{2^h-1} \left[(K+h-1)n \right]$$

This bound provides a target value of d_f which can be used when searching for good codes. If a code is found with a d_f which satisfies the bound with equality, it is immediately known that no code exists with a larger minimum distance.

Of course, maximizing minimum free distance does not necessarily minimize decoder error probability. The number of codewords having the minimum distance, as well as the distribution of codewords at distances somewhat greater than d_f, are also important. After preselecting codes based on minimum free distance, these other factors are useful in final code selection. Simulations and numerical code evaluation indicates that choosing codes with maximum minimum free distance, taking into account the number of paths at this distance, and if necessary, slightly larger distances yields codes with minimum error probabilities with Viterbi decoding.

The optimum rate 1/2 codes for K=3 through 8 were found by Odenwalder (Ref. 6). They are tabulated in Table 2.1. For each constraint length the table shows the optimum code generators, the actual d_f for the code, the number of errors n_e in all of the codewords at the minimum distance, and the upper bound value on minimum free distance d_f*.

2.3.2 Numerical Code Performance Bound

One of the two principal tools used in evaluating the performance of convolutional codes in this study has been an upper bound on error probability related to the convolutional code transfer function. The bound is extremely tight for high E_b/N_o (low decoder error rates) where computer simulation is impractical, due to the prohibitively long times

K	Code Generators	d_f	n_e	d_f^*
3	111 101	5	1	5
4	1111 1101	6	2	6
5	11101 10011	7	4	8
6	111011 110001	8	6	9
7	1111001 1011011	10	36	10
8	11111001 10100111	10	2	10

Table 2.1 Optimum Rate 1/2 Codes. d_f is the code minimum free distance, n_e is the number of bit errors in paths at distance d_f, d_f^* is the upper bound on minimum free distance.

required to collect significant data.

It has been shown (Ref. 7) that a union bound on the performance of a convolutional code on memoryless channels can be obtained from the directed-graph state diagram of the coder. For example, the optimum constraint length K=3, rate 1/2 coder is shown in Fig. 2.2a. The states correspond to the contents of all but the first stage of the coder register, when a new information bit has just entered the first stage. The exponent of D(0, 1, or 2) is the weight of the (two symbol) vector output at this time, and the exponent of N(0 or 1) indicates whether a 0 or 1 information bit has just entered the coder.

Regarding the all zeros node as both the input and output of the graph, the transfer function of any path through the tree is defined as the product of the branch transfer functions along that paht. For example, the transfer function of the path corresponding to the information sequence 10100 is

$$T_{10100} = (ND^2)\ (D)\ (N)\ (D)\ (D^2) = N^2D^6 \qquad (2.1)$$

The transfer function of the graph is the sum of the transfer function of all paths starting and ending in the all zeros state. The general form of this transfer function is

$$T(N,D) = D^{d_f} f_1(N) + D^{d_f+1} f_2(N) + \cdots + D^{d_f+i} f_{i+1}(N) + \cdots \qquad (2.2)$$

Here d_f is the minimum free distance of the code. Notice that the exponent in the path transfer function (Eq. 2.1) is the weight of the code symbols on the particular path through the graph of Fig. 2.3. Therefore, with N=1, the terms in the transfer function T(1,D) are of the form

$$D^{d_f+i} f_{i+1}(1)$$

where $f_{i+1}(1)$ is just the number of paths at distance d_f+i.

For the unquantized, additive white Gaussian noise channel with PSK modulation, the error probability between the all zeros (correct) path, and another path which diverges from and returns to the all zeros path, is bounded by (Ref. 8)

$$P_2 < e^{-dE_s/N_o}$$

where d is the weight of the competing path. E_s is the code symbol energy and N_o is the noise spectral density. For example, if the competing path corresponded to the information sequence 10100, the bound is obtained from Eq. (2.1)

$$P_2 < T_{10100} \Big|_{N=1,\ D=\exp(-E_s/N_o)} = e^{-6E_s/N_o}$$

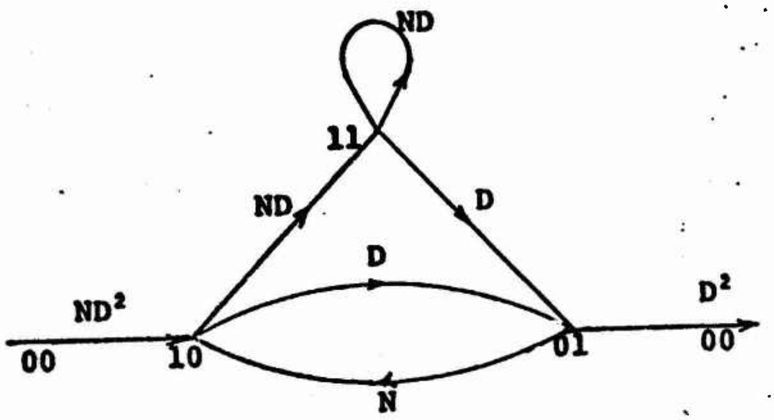


Fig. 2.3 - State Diagram for K=3 Code

Likewise, a union bound on first event error probability due to all paths competing with the all zeros path (all paths through the graph in Fig. 2.3 is

$$P_E < T(N,D) \Big|_{N-1,\ D=\exp(-E_s/N_o)} \tag{2.3}$$

In order to get a bound on bit error probability, we note that the exponent of N in a path transfer function is the number of information 1's (errors) on that path. A union bound on bit error probability would be obtained if the path transfer function were weighted by the number of bit errors on the path. One simple way of doing this is to take the derivative of T(N,D) with respect to N. This brings down the exponents of N -- the number of bit errors on a path -- into the coefficients. The bound on bit error probability is therefore

$$P_B < \frac{dT(N,D)}{dN} \Big|_{N=1,\ D=\exp(-E_s/N_o)} \tag{2.4}$$

For the Gaussian channel these bounds can be tightened somewhat

$$P_E < \text{erfc}\,(\sqrt{d_f E_s/N_o})\quad D^{-d_f} T(N,D) \Big|_{N=1} \tag{2.5}$$

$$P_B < \mathrm{erfc}\left(\sqrt{d_f E_s/N_o}\right) D^{-d_f} \left.\frac{dT(N,D)}{dN}\right|_{N=1} \qquad (2.6)$$

with $D = \exp(-E_s/N_o)$ in both cases.

The difficulty of this approach is that the number of states grows exponentially with K and consequently the tedium involved in direct computation is effectively insurmountable for K>4.

On the other hand, the calculation of the transfer function is equivalent to a matrix inversion. Taking into account the particular properties of a convolutional code transfer matrix, the transfer function can be evaluated numerically using an iterative technique. A computer program has been written to evaluate the transfer function bound as a function of K, code rate, and E_b/N_o. For rate 1/2 codes the performance bounds are presented in Section 2.3.3.

Decoder performance predicted by the bounds at around 10^{-5} bit error rate is quite close to simulation results, allowing for finite receiver quantization in the simulations.

2.3.3 Simulation and Numerical Performance Data*

* Some of the results reported in this subsection and subsections 2.3.4.2, 2.3.5, and 2.4 were obtained previously by LINKABIT Corporation and appear in the final report of contract NAS2-6024 NASA Ames Research Center, Moffett Field, CA.

2.3.3.1 General Performance Results

The principal results of the simulations and code transfer function bounds are shown in Figs. 2.4, 2.5, and 2.6. All of these figures show bit error rate vs. E_b/N_o for Viterbi decoders using the optimum rate 1/2 convolutional codes of Table 2.1. In all cases, the decoder state path length was 32 bits. In all simulation results in Figs. 2.4 and 2.5 are for soft (8-level) receiver quantization. Equally spaced demodulation thresholds are used at $\pm 1.5\sigma$, $\pm\sigma$, ± 0.5, and 0 where $\sigma^2 = N_o/2$ is the noise variance. This choice of 8-level quantizer thresholds is within a broad range of near optimum values, as will be shown presently. The transfer function bound is for infinitely finely quantized received data. Allowing for the 0.20 to 0.25 db loss usually associated with 8-level receiver quantization compared with infinite quantization, the transfer function bound curves are in excellent agreement with simulation results in the 10^{-4} to 10^{-5} bit error rate range.

Since the accuracy of the transfer function bound increases with E_b/N_o, decoder performance can be ascertained accurately in the 10^{-5} to 10^{-8} region even in the absence of simulations.

Ideally, the symbol metrics associated with each of the 8 quantization levels would be proportional to the log-likelihood of receiving the given level, given the hypoth-

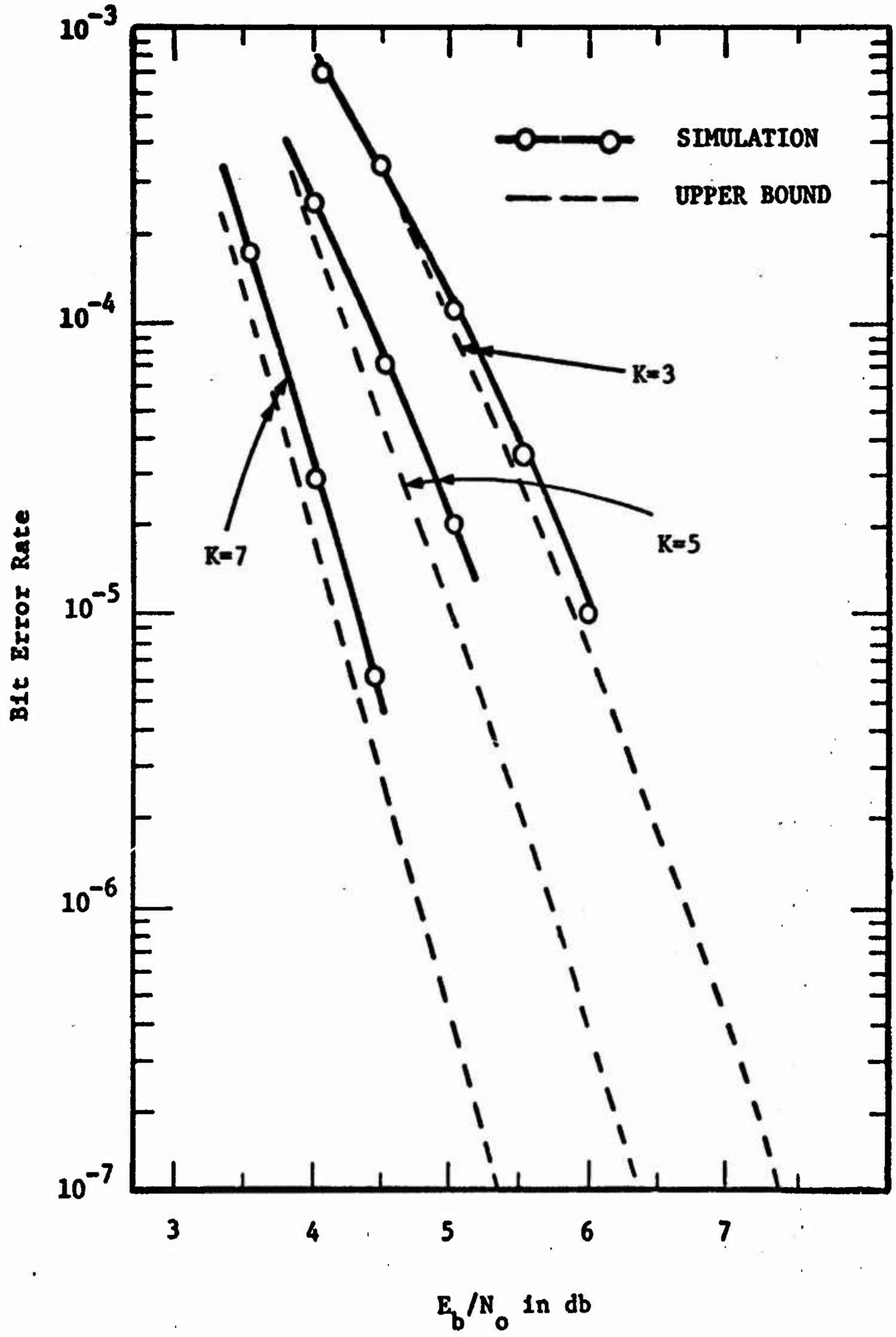


Fig. 2.4 - Bit error rate vs. E_b/N_o for rate 1/2 Viterbi decoding. Eight level quantized simulations with 32 bit paths, and the infinitely finely quantized transfer function bound, K=3, 5, 7.

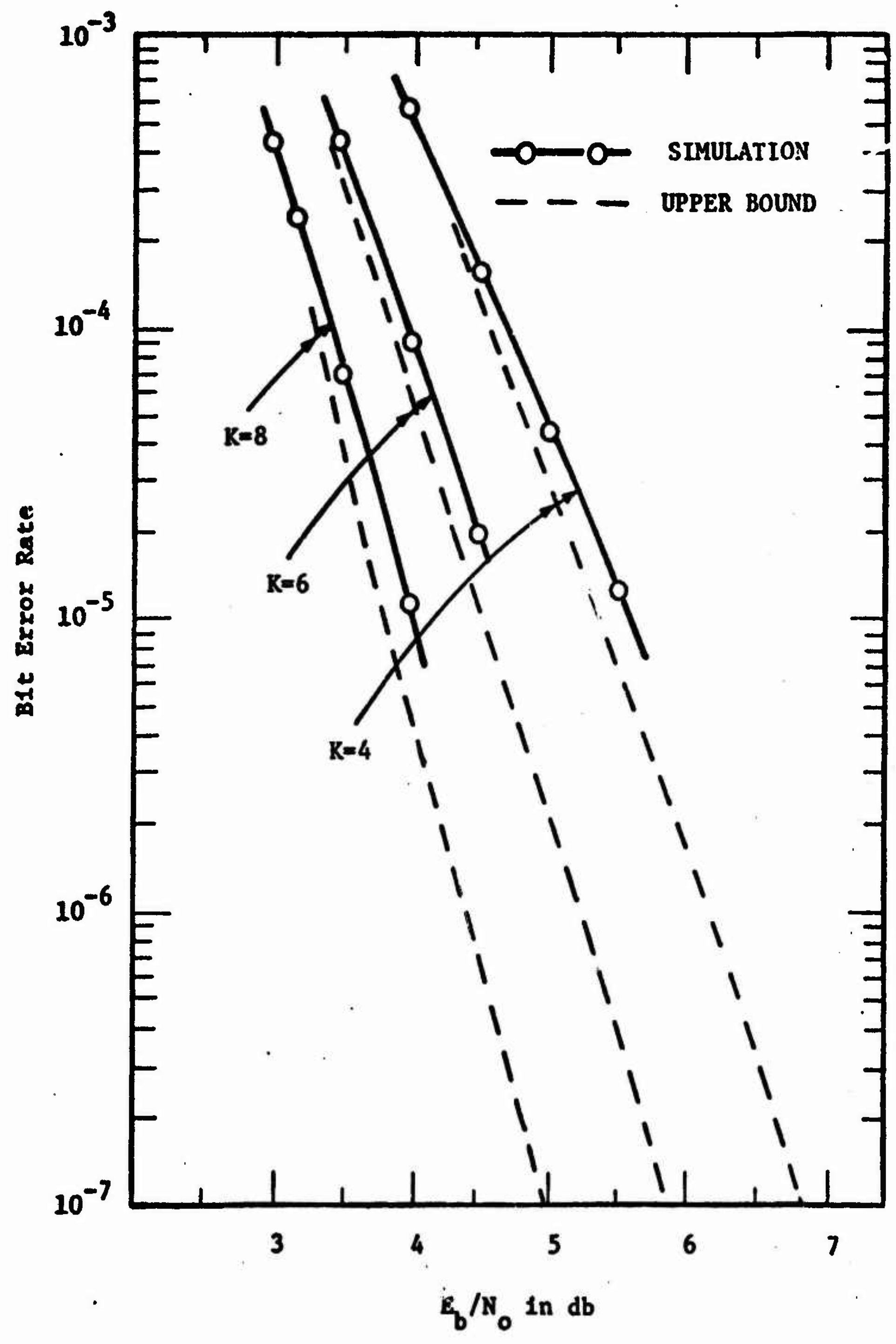


Fig. 2.5 - Bit error rate vs. E_b/N_o for rate 1/2 Viterbi decoding. Eight level quantized simulations with 32 bit paths, and the infinitely finely quantized transfer function bound, K=4, 6, 8.

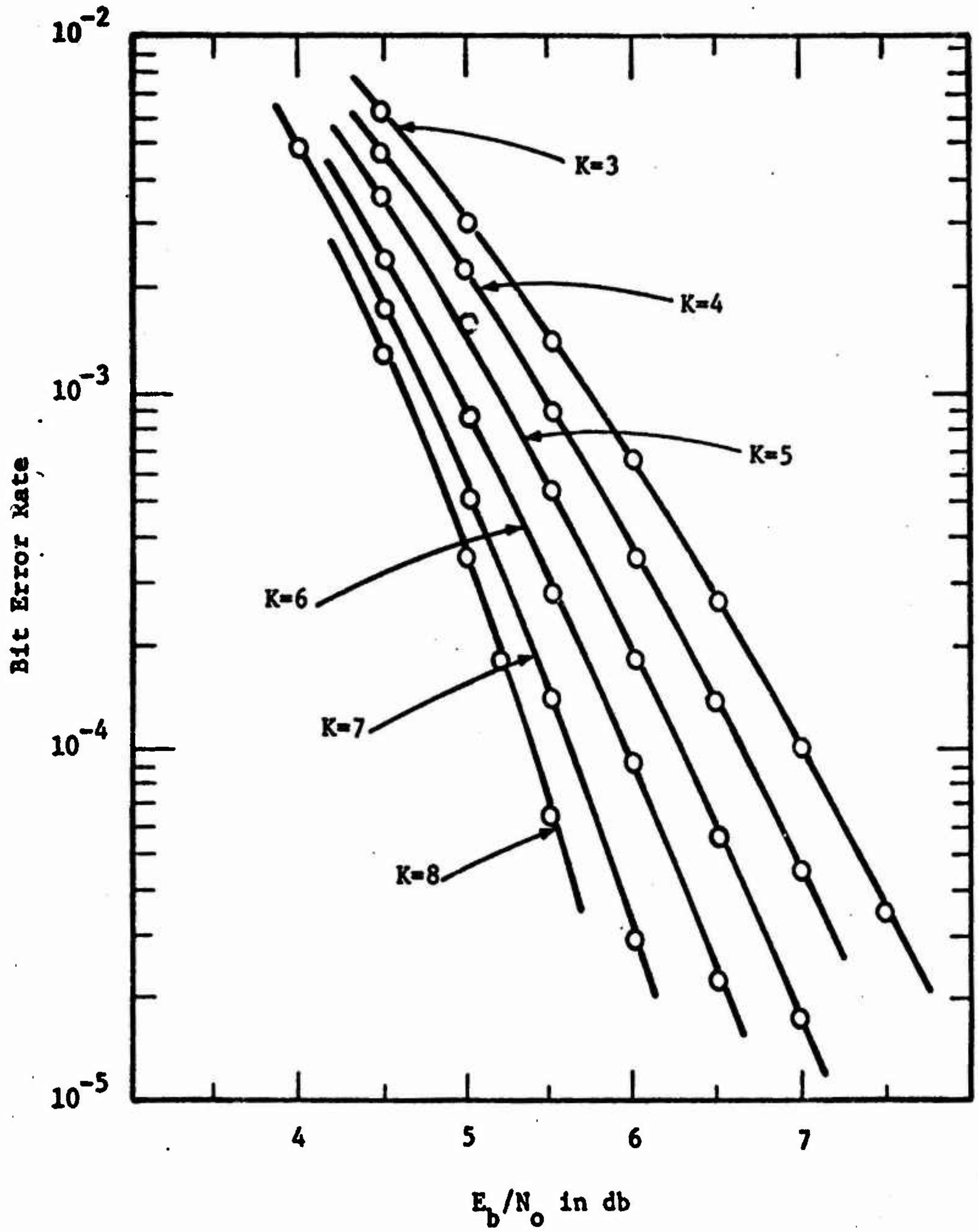


Fig. 2.6 - Bit error rate vs. E_b/N_o for rate 1/2 Viterbi decoding. Hard quantized received data with 32-bit paths K=3 through 8.

esis of a "0" of a "1" transmitted. In the interest of keeping the number of bits required to represent metrics to a minimum, it was shown (Ref. 2) that equally spaced symbol metrics, for instance, the number 0-7, could be used with negligible performance degradation. We have taken the compression of metric representation one step further. An additional bit in the state metric can be saved if levels symmetrically located about the zero threshold have symbol metrics which are the negatives of one another. Thus, for the simulations presented in Figs. 2.4 and 2.5, the eight symbol metrics used were 4, 3, 2, 1, -1, -2, -3, -4. These symbol metrics clearly do not change in equal increments; however, simulations have shown that system performance does not suffer significantly.

Fig. 2.6 gives the simulation results for Viterbi decoding with hard receiver quantization. The same optimum rate 1/2, K=3 through 8 codes were used here as in the 8-level quantized simulations.

Several points are obvious from the performance curves:

a) 2-level quantization is everywhere close to 2 db inferior to 8-level quantization.

b) Each increment in K provides an improvement in efficiency of something less than .5 db at a bit error rate of 10^{-5}.

c) Performance improvement vs. K increases with decreasing bit error rate.

2.3.3.2 Receiver Quantization

In order to observe the effects of varying receiver quantization more closely, simulation performance data is presented in Fig. 2.7 for the K=5, rate 1/2 code, with 2, 4, and 8-level receiver quantization. The 8-level thresholds and metrics are identical to those of Fig. 2.4. In fact, the 2 and 8 quantization level curves are taken from Figs. 2.6 and 2.4 respectively. The 4-level thresholds were set at 0 and $\pm\ \sigma$. The metrics were chosen to be 2, 1, -1, -2, for the same reasons which suggested the 8-level metrics.

2.3.3.3 Path Memory

The Viterbi decoder is a maximum likelihood decoder only when its decision path memories are infinitely long. That is, decoding delay is infinite. For practical purposes, it is desirable to use path memories as short as possible. There is a path memory for each state in a Viterbi decoder. Providing storage and managing decision paths is a significant part of any Viterbi decoder. It is therefore worthwhile to study the performance degradation vs. path length for Viterbi decoding.

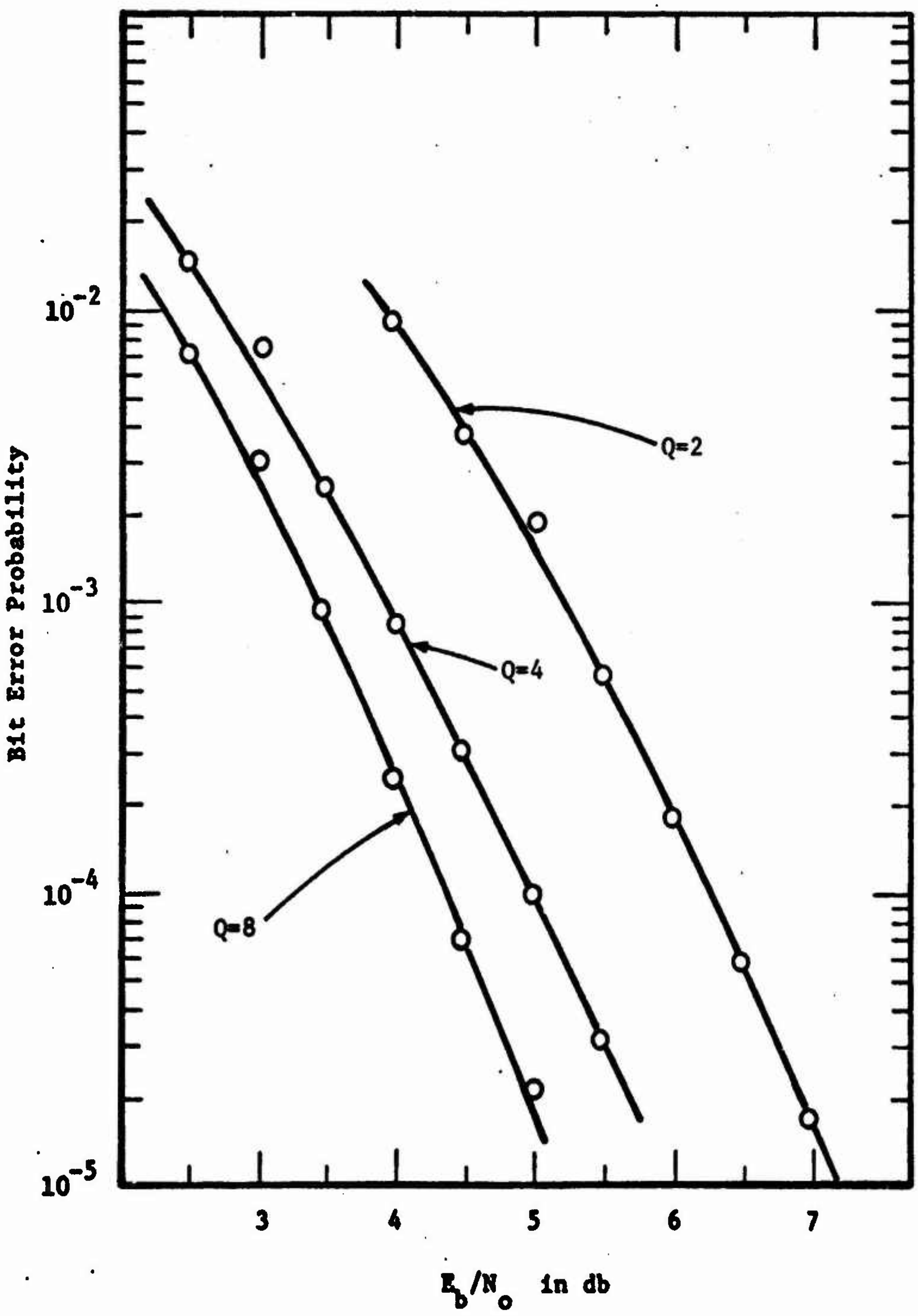


Fig. 2.7 - Performance comparison of Viterbi decoding using a rate 1/2, K=5 code with 2, 4, and 8 level quantization. Path length = 32 bits.

Fig. 2.8 shows bit error rate performance vs. E_b/N_o for three path lengths (8, 16 and 32) using the rate 1/2, K=5 code, for both 2 and 8-level received data quantization. The length 32 path curve is identical to the K=5 curve in Fig. 2.4. Performance with length 32 paths is essentially identical to that of an infinite path decoder. Even for a path length of only 16, there is only a small degradation in performance. Other simulations have shown that a path length of 4 to 5 constraint lengths is sufficient for other constraint lengths as well.

2.3.3.4 Decoder Output Selection

In a Viterbi decoder with finite path memories, it is possible that not all state paths are merged at the point at which a decoded bit must be output. Physically this means that the oldest bits in each of the state path memories may not always agree. The decoder must output a bit however and there must be a means for selecting which of the 2^{K-1} oldest path bits to output.

The optimum method for selecting output bits is to choose the bit corresponding to the path with the best metric. This selection rule is very complex to mechanize in a high speed decoder, where the pairwise state comparisons are done in parallel. This fact has lead to a study of simpler output selection schemes, the aim being to find one which does not degrade performance appreciably. One

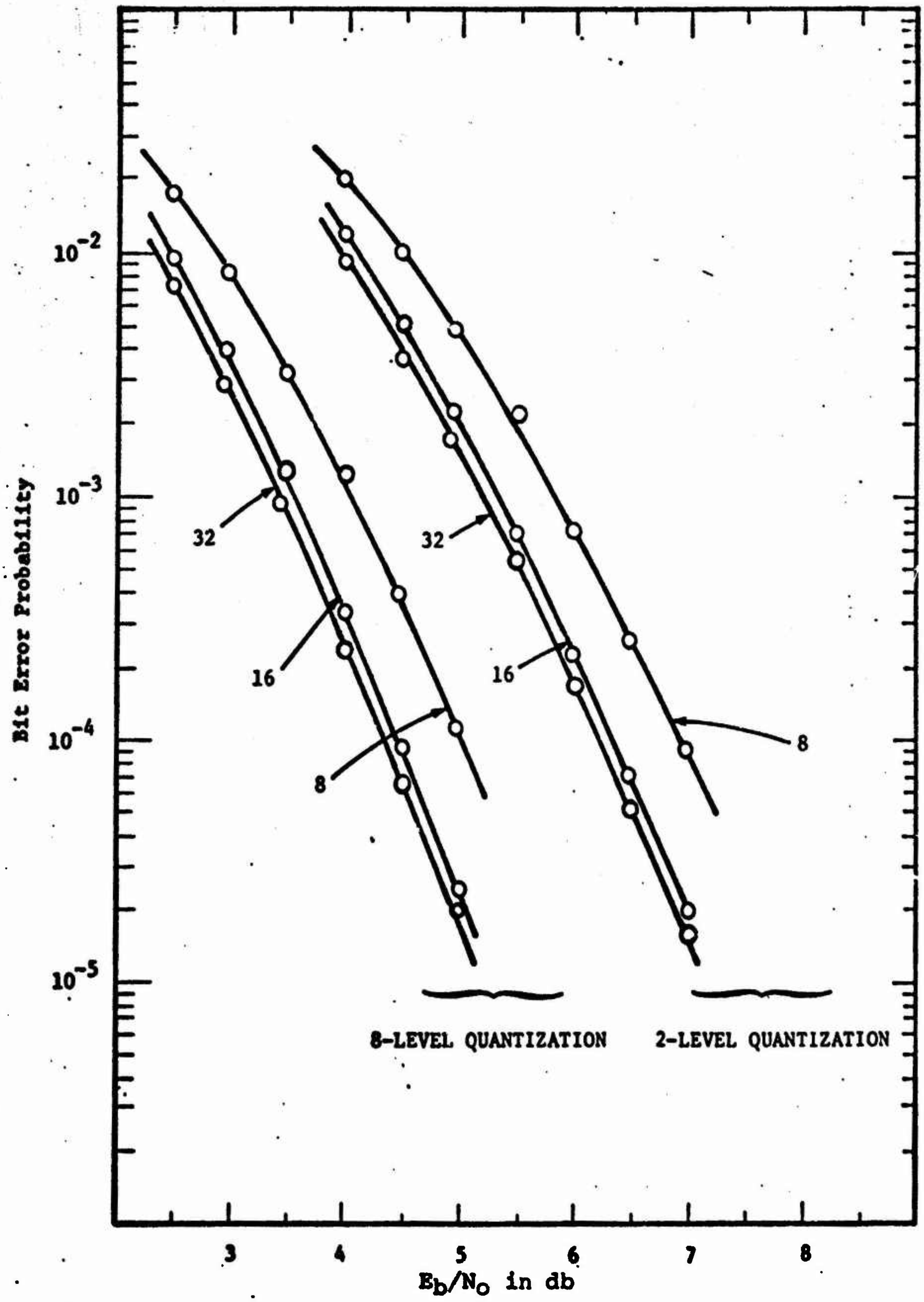


Fig. 2.8 Performance Comparison of Viterbi decoding using a rate 1/2, K=5 code with 8, 16, and 32 bit path lengths and 2 and 8 level quantization.

very simple scheme is to choose a path at random from which to output decoded bits (or always output bits from the same path). This scheme, however, has been found to significantly degrade performance. In fact, if a path memory of n bits is required for a given performance goal with maximum likelihood output selection, then simulation has shown that a memory of up to 2n bits is required for the same performance with arbitrary output selection.

Another method is to output a "0" if a majority of the 2^{K-1} paths have a "0" as their oldest bit; otherwise, output a "1". This scheme is somewhat simpler to implement than maximum likelihood selection.

An efficient yet simple to implement scheme, which we have devised, is to select the output from some state path whose metric is better than a certain threshold. This scheme is called "less than four" selection. Fig. 2.9 compares the performance of a) maximum likelihood selection, b) majority selection, and c) "less than four" selection. The comparison is made for a K=7, rate 1/2 code with 8-level quantization, and path length 32. On the other hand, the performance of a K=5 decoder with a 32 bit path memory was the same for all three output selection schemes.

As the path memory gets long relative to K, there is a larger probability that all state paths will be merged by the time a bit must be output. Thus the output selection mechanism has less of an effect on performance.

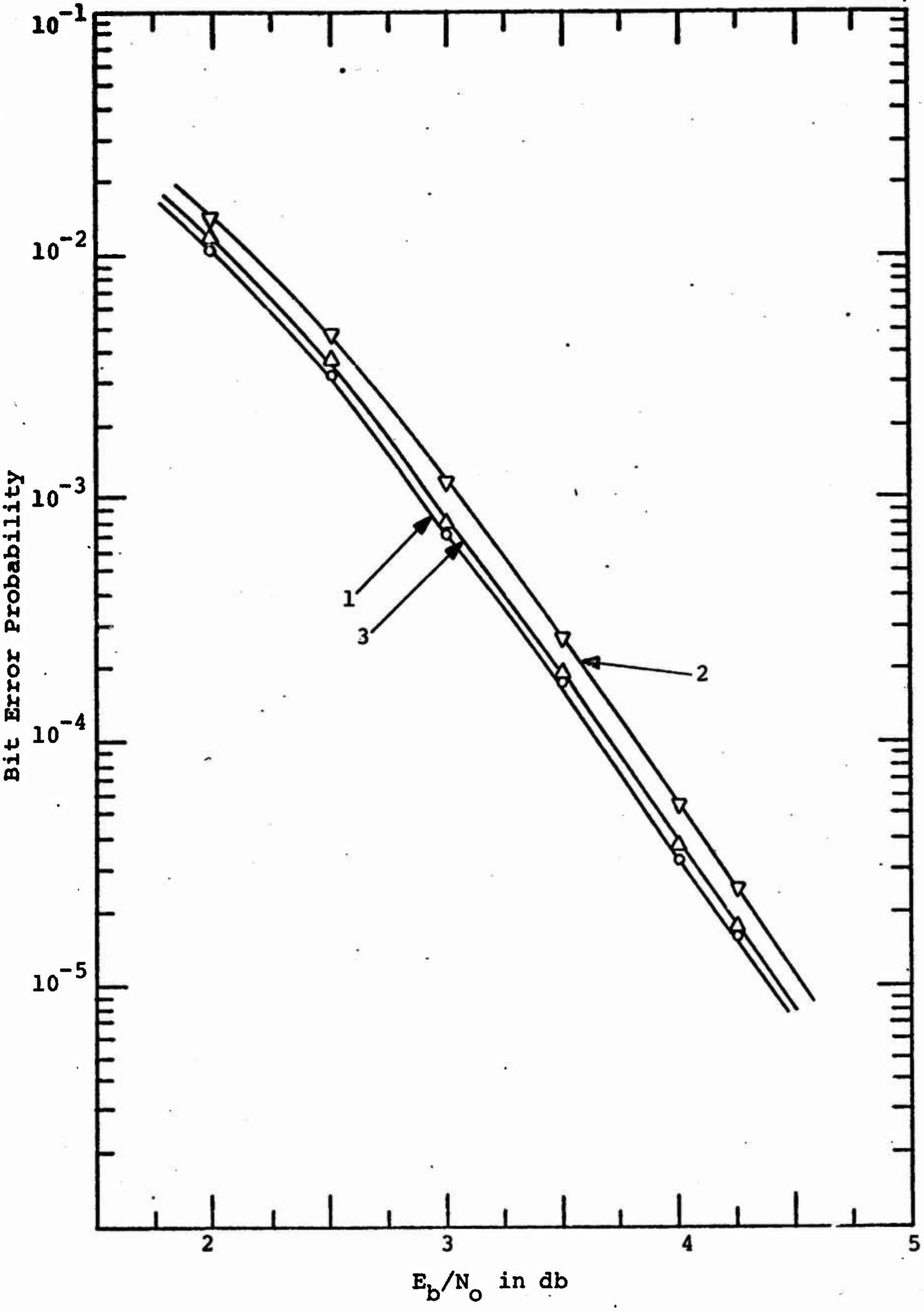


Fig. 2.9 Bit Error Rate vs. E_b/N_o for Viterbi Decoding of the K=7, Rate 1/2 Code, Using the Following Output Selection Mechanisms: 1) Maximum Likelihood, 2) Majority, and 3) "Less than Four". Q=8 Levels, path length = 32.

2.3.4 Code Synchronization and Channel Reliability

2.3.4.1 Node Synchronization and Phase Ambiguity Resolution

In a TDMA system there is no code synchronization problem beyond that of obtaining TDMA frame synchronization. Once frame sync is obtained it is a simple matter of counting bits from the start of frame to establish beginnings of bursts, sub-bursts, etc. Furthermore, no 90 or 180 degree phase ambiguity exists because phase tracking is done on the unmodulated reference contained in the preamble of each burst. The following discussion of code synchronization and phase ambiguity resolution is therefore pertinent to unframed data communication systems such as a pure FDMA channel with phase tracking performed on the QPSK or BPSK modulated signal.

Because of the inherent continuity involved in convolutional coding, code synchronization at the receiver is usually much simpler than in the case of block codes. For convolutional decoding techniques involving a fixed number of computations per bit decoded, such as Viterbi decoding, the decoder initially makes an arbitrary guess of the encoder state to start decoding. If the guess is incorrect, the decoder will output several bits or, at most, tens of bits of unreliable data before assuming steady state reliable operation. Thus, the block synchronization problem does not really exist. There remains the problem of node synchronization and, depending upon the modulation-demodulation technique used, the problem of 2 or 4 phase ambiguity resolution. For a rate 1/n code, there are n code symbols on each branch in the code tree. Node synchronization is obtained when the decoder has knowledge of which sets of n symbols in the received symbol stream belong to the same branch. In a purely serial received stream, this is a 1 in n ambiguity.

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

We have directed our efforts toward using the error <u>detection</u> capability of convolutional decoders, or the ability of the decoder to detect unsatisfactory operation of the system to detect and correct for incorrect node synchronization and the occasional phase flips in the phase tracking loop. Simple and effective techniques for maintaining node and phase synchronization <u>completely within the decoder itself</u> have been shown to be feasible.

For the purpose of ease of illustration, the rate 1/2, hard-quantized receiver output case will be considered here. Techniques generalize easily to soft quantization and with somewhat more complexity to other rates. A Viterbi decoder operating on hard quantized received data will use Hamming distance for state metrics. Relatively large metric values indicate a poorer match to received data than lower metric values. The smallest state metric at any time corresponds to the path with the best match to received data, and the metric itself is equal to the number of discrepencies between the received data and that path. Clearly, when the decoder is in correct node synchronization and the demodulator loop is locked properly, the path with the smallest Hamming distance will usually correspond to the correct path. The rate of increase of this path metric will depend on the channel error rate. For instance, if the channel error rate is $p = .02$ then, on the average, there will be an increase of 1 in the correct path metric for every 50 channel symbols. On the other hand, we intuitively expect that if node synchronization is lost, or if the phase lock loop locks onto an incorrect phase position, the match between the received data at the nearest codeword should be much poorer than 1 mismatch in 50 symbols. If, in fact, the best path metric increases more rapidly when out of node or phase synchronization, this can be used to detect these maladies.

In actual practice the best path metric is not always conveniently available. A more readily accessible measure of decoder performance is the state metric normalization signal (Ref. 9). This signal provides an indication whenever the best state metric (Hamming distance) increases by 4. It is used by the decoder to re-normalize path metrics by subtracting 4.

One simple technique would use an up-down counter to detect unreliable system operation. The counter counts up by k units (k>1) each time a metric normalization occurs, while it counts down by 1 for each data bit. The parameter k is chosen so that the average drift of the counter is downward when in proper node and phase synchronization and upward when out of either phase or node synchronization or both. The first condition requires that $k\frac{p}{4} < 1$ where p is the channel crossover probability; while, the second condition requires $k\frac{p'}{4} > 1$ where p' is the rate of increase of the best path metric with improper node or phase synchronization (It can be shown that for rate 1/2 codes $p' \approx .11$ regardless of p, k, or the particular code used). The factor of 1/4 is due to the fact that 4 apparent channel errors are needed to cause a metric normalization event. The count is not allowed to fall below zero. When the count exceeds a threshold value N, the assumed node synchronization or phase synchronization position is changed in the decoder according to a preset strategy.

Potentially effective methods of changing phase and node synchronization depend upon whether the system uses

BPSK or QPSK. For BPSK, both the problem of node synchronization and 180° phase ambiguity exist. The 180° phase ambiguity can be circumvented by using differential encoding of the information prior to convolutional encoding and differential decoding after the channel decoder. A transparent convolutional code is required, i.e., the all 1's data sequence maps into the all 1's code sequence. This technique is discussed in Section 2.3.4.2.

With a transparent rate 1/2 coded system, whenever the synchronization count exceeds N, the node synchronization position is changed between one of two positions. Likewise when QPSK modulation is used with a rate 1/2 code transparent to 180° phase changes, synchronization counter overflows are used to change the assumed phase state by 90° (this is accomplished by exchanging the received code symbol pairs and complementing one of them). Thus, in both the BPSK and QPSK cases the synchronization technique is called upon to resolve a one of two state ambiguity for proper decoder operation.

As mentioned previously, the factor k and the threshold N must be chosen such that the counter used to indicate data quality drifts upward when synchronization is incorrect and drifts downward when synchronization is correct. The actual values chosen for these parameters will determine:

a) the expected time to first passage over the threshold, and hence change of system state, when node or phase synchronization is incorrect. We will call the expected time to resynchronization, E_{rs}.

b) the expected first passage time when node and phase synchronization is correct. This is the expected time to false alarm, E_{fa}.

Obviously we want to make E_{rs} as short as possible and E_{fa} as large as possible. Analytical techniques have been successfully used to approximate E_{rs} and E_{fa} as a function of k, T and p. The analysis is based upon recognizing that the input to the up-down counter is a random walk with a reflecting boundary at zero and an absorbing boundary at N. This work is reported in detail in (Ref. 4).

In addition to the approximate analysis, simulations and experiments using the feasibility model Viterbi decoders have been performed to tie down E_{rs} and E_{fa} more precisely. The count-up rate, k, was chosen to be 48, because that value is nearly optimum. The rate 1/2, K=7 code was used with hard receiver quantization. Fig. 2.10 shows E_{fa} vs. the threshold T. This limited data supports the analytical results which state that E_{fa} rises exponentially with T. Naturally, for a given T, E_{fa} is larger for smaller channel crossover probabilities p.

Fig. 2.11 shows the other parameter E_{rs} as a function of T. E_{rs} also rises with T, but more slowly. In fact it is nearly linear with T, as predicted by theory. E_{rs} is not

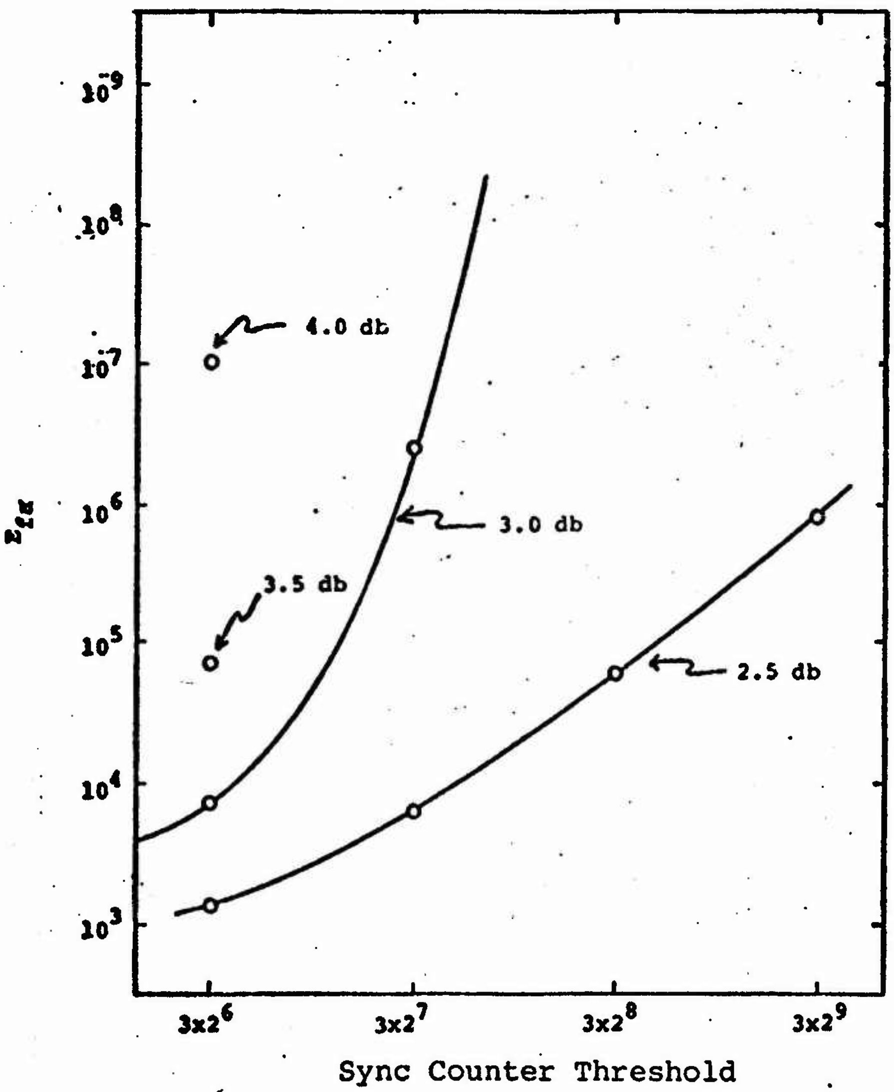


Fig. 2.10 Average number of bits decoded between false loss of sync event vs. sync counter threshold with E_b/N_o as a parameter - hard quantization, K=48.

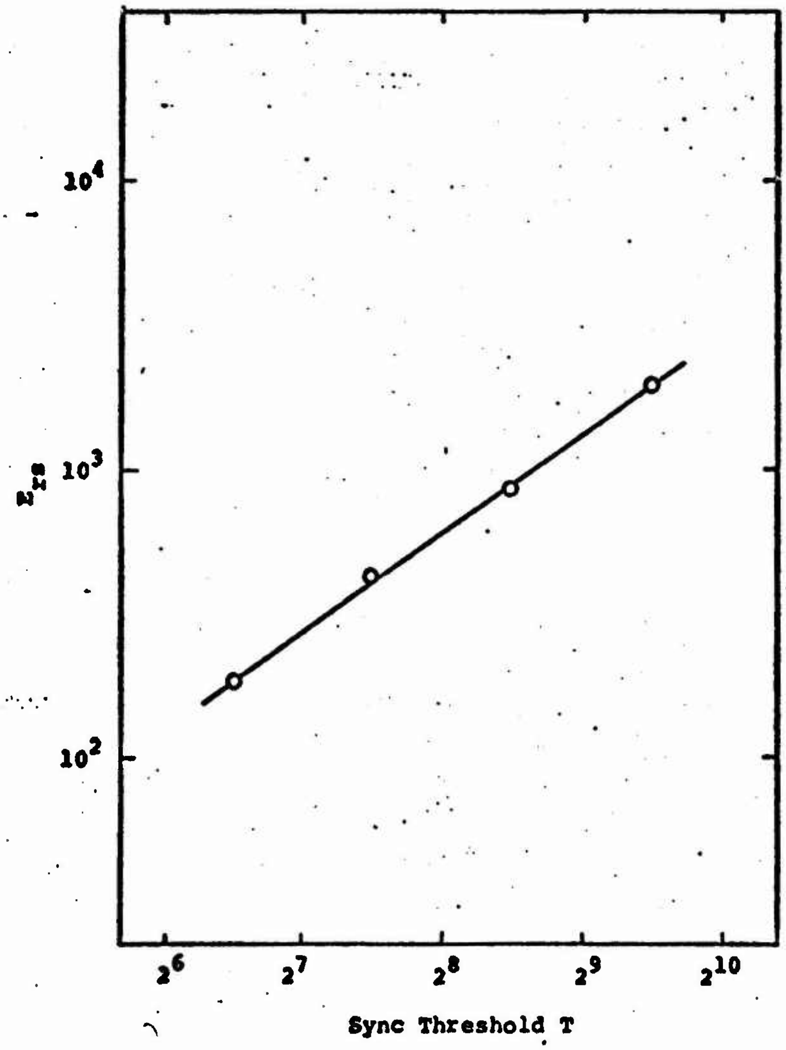


Fig 2.11 Average number of bits to recover after loss of sync vs. sync counter threshold - hard quantization, K=48.

a function of p because it is the expected time to resync when the decoder is out of sync. When out of sync, the received data has the same random statistics regardless of the noise.

These results show that a value of T can be selected to make E_{fa} truly negligible, while maintaining E_{rs}, the resync time, as low as several hundred to a thousand bits. The increase in system bit error rate due to false loss of sync is

$$P_L = E_{rs}/E_{fa}$$

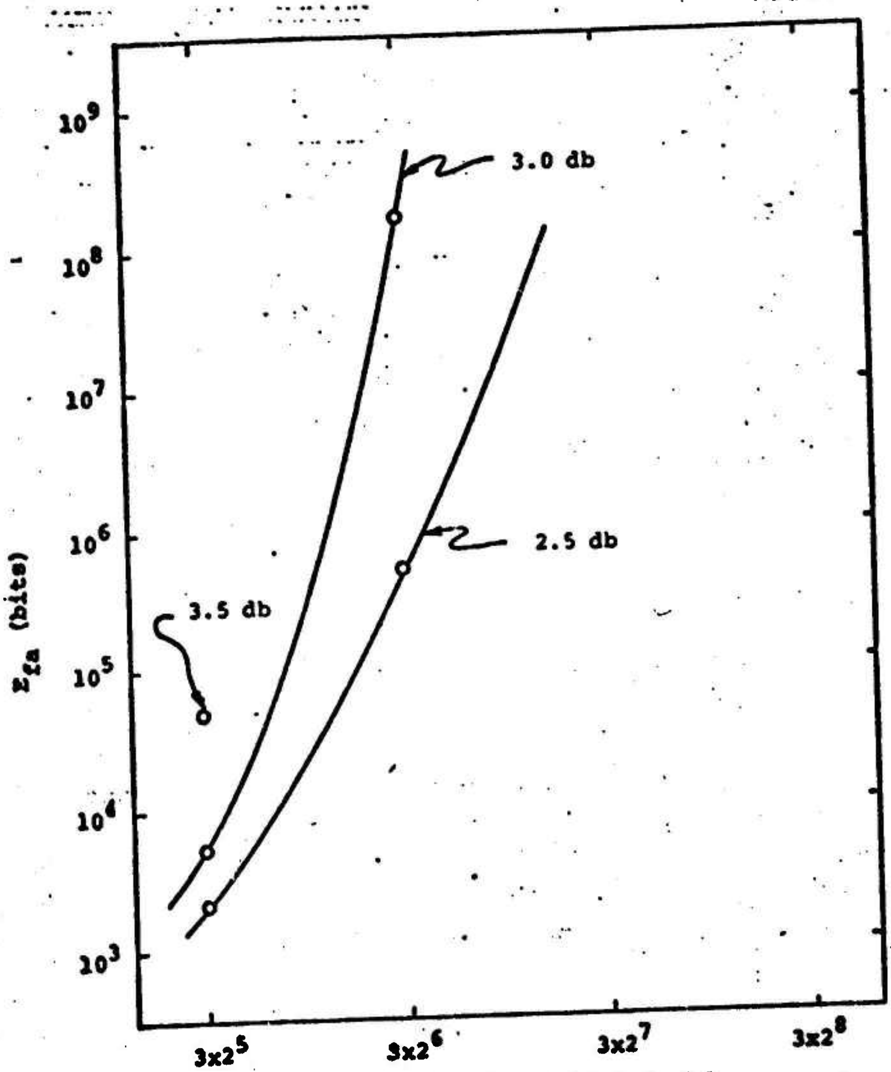
This is because, on the average, for E_{fa} bits decoded sync is lost once. It takes, on the average, E_{rs} to return to the proper sync state.

In the case of soft (8-level) receiver quantization, the same basic node-phase synchronization technique can be used. Now, however, it is possible for the best state metric to increase by up to 4 units in one bit time. This, coupled with the fact that a soft decision decoder operates at about a 2 db lower E_b/N_o for the same performance as a hard decision decoder, necessitates a lower value for k. If k=48 were used for soft quantization as well as hard, the increased occurrence of "soft errors" at low E_b/N_o would bias the sync counter upward even with the decoder in proper synchronization. A value of k=24 was chosen (for the feasibility models) as close to optimum for the range of E_b/N_o corresponding to decoder bit error rates of 10^{-3} and below. Figs. 2.12 and 2.13 show E_{fa} and E_{rs} vs. the threshold T for 8-level quantization using the rate 1/2, K=7 Viterbi decoder. Notice that in this case E_{rs} is somewhat dependent on E_b/N_o.

2.3.4.2 Transparent Codes

As mentioned in the previous section, one way to resolve 180° phase ambiguities is to use a code which is transparent to 180° phase flips, precode the data differentially and use differential decoding. A transparent code has the property that the bit-by-bit complement of a code-word is also a codeword. Such a code must have an odd number of taps on each of its encoder mod-2 adders. This insures that if a given data sequence generates a certain codeword, its complement will generate the complementary codeword.

If the received data is complemented due to a 180° phase reversal, it will still look like a codeword to the decoder, and will likely be decoded into the complement of the correct data sequence. Now decoding to the complement of the sequence input to the encoder is no problem if the



Sync Counter Threshold

Fig. 2.12 Average number of bits decoded between false loss of sync events vs. sync counter threshold with E_b/N_o as a parameter - 8 level quantization, K=24.

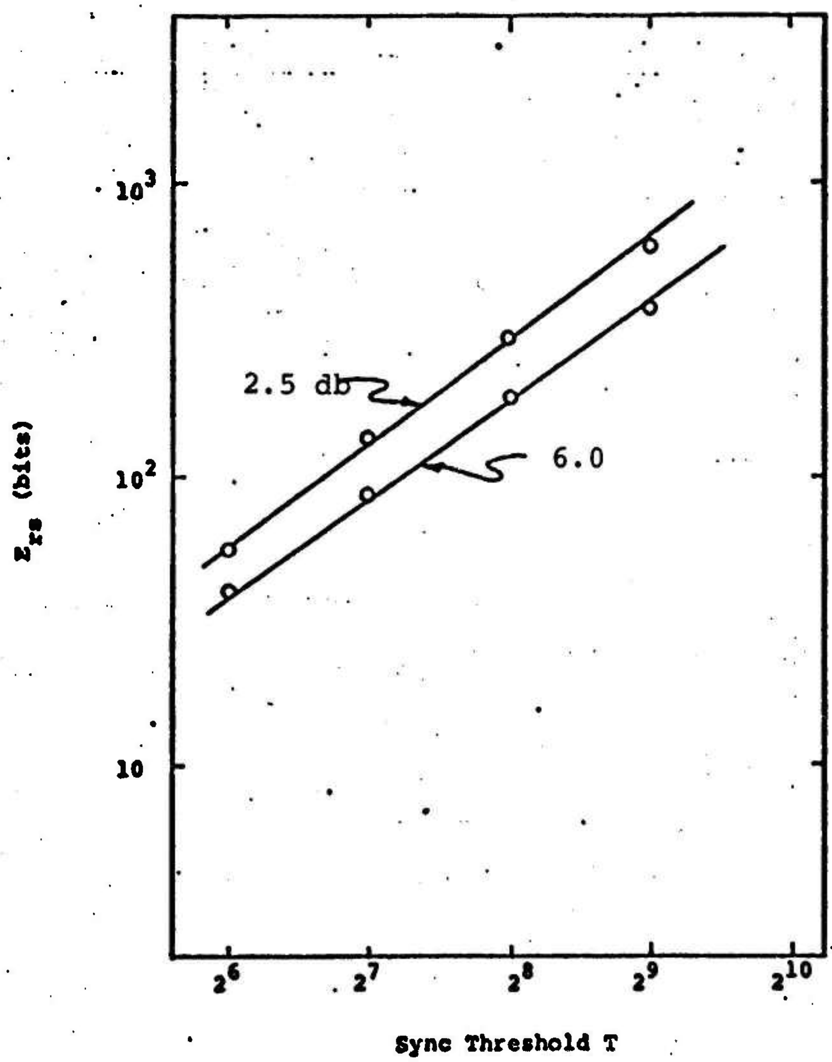


Fig 2.13 Average number of bits to recover after loss of sync vs. sync counter threshold with E_b/N_o as a parameter - 8 level quantization, K=24.

data was precoded differentially. This means that information is contained in the occurrence or non-occurrence of transitions in the encoded output sequence rather than the absolute sequence itself. These transitions occur in the same places even if the decoded sequence is complemented.

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

Fig. 2.14 shows bit error rate performance curves for the K=7, rate 1/2 code with and without differential encoding-decoding. The degradation in error probability is least at low E_b/N_o. Here decoder error bursts are relatively long (on the average one to two constraint lengths), so differential encoding-decoding loses very little. At higher E_b/N_o, bit error rate degradation is slightly larger--but nowhere near a factor of two. The

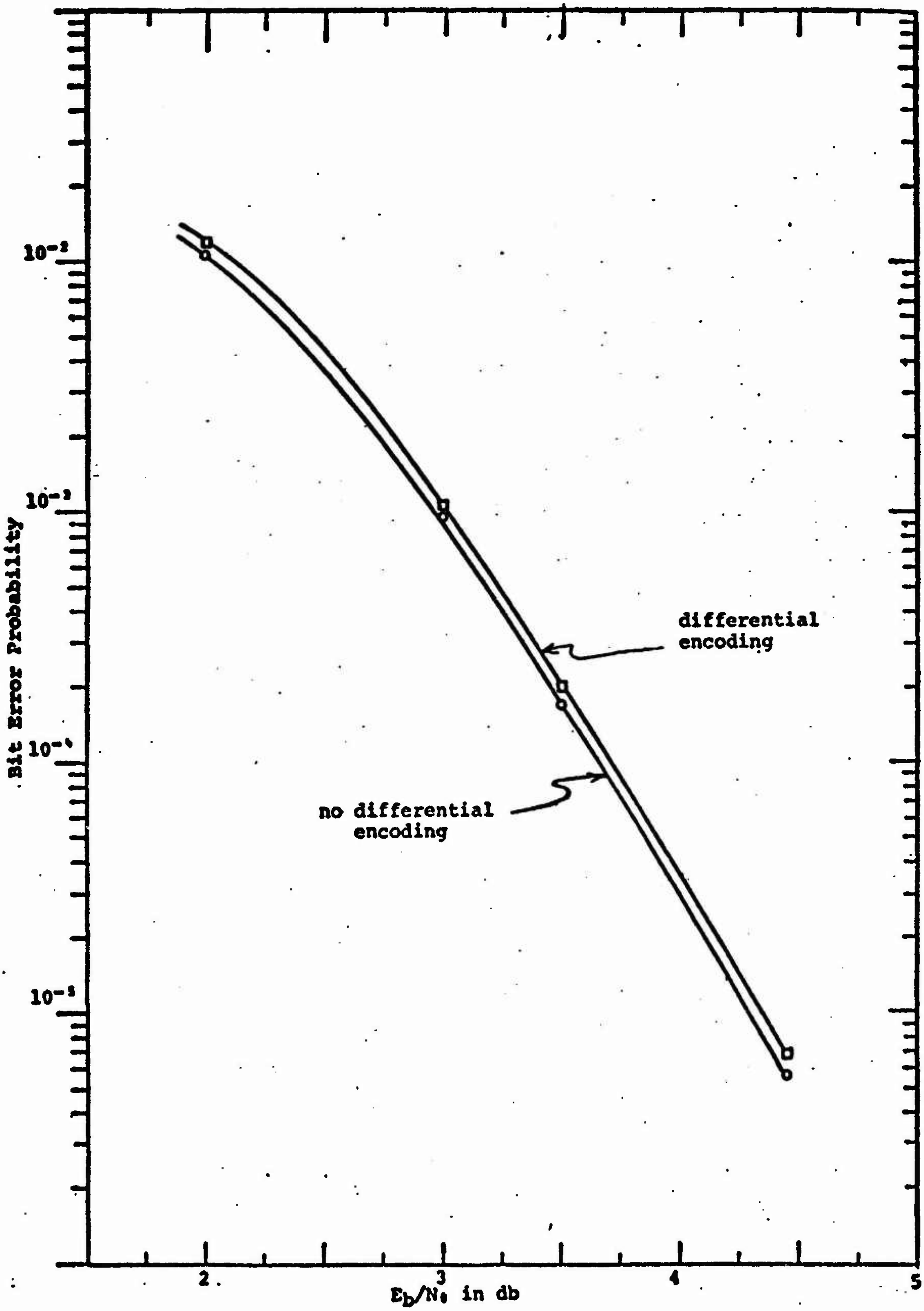


Fig. 2.14 Performance Comparison of Viterbi Decoding With and Without Differential Encoding-Decoding. K=7, Rate 1/2 Transparent Code used. Q=8 Levels.

worst bit error rate degradation factor is about 1.2 over the range shown. Using differential encoding-decoding, the E_b/N_o required increases by less than 0.1 db.

2.3.4.3 Channel Reliability Information

The up-down counter used for node and/or phase synchronization also provides a means for very sensitive measurement of channel performance. For hard decisions, in the absence of decoder errors, the counter counts up whenever four channel errors occur. Thus, the number of times the counter counts up per unit time is directly proportional to the channel error rate p. This will be true as long as the decoder output error rate is low, which corresponds to good system performance. If p becomes large due to a system failure or loss of code sync, the count will tend to remain above zero.

One method of monitoring system reliability is to sum the count over a number of bits which is large compared with 1/p, where p is the lowest channel error rate to be monitored. The integrator output will be stable and proportional to p when p is in the range corresponding to decoder error rates lower than about 10^{-2}. As p rises beyond this point, the integrator output will rise monotonically, but not proportionally. When p becomes greater than about .11, the integrator output will saturate. This is because the number of decoder corrections (metric increases) for a rate 1/2 code never exceeds .11 on the average as was discussed in the previous section.

Integration of the sync counter value over a fixed time window therefore provides a sensitive measure of p when the decoder is putting out useful data. It also is a good indicator of system failure. This is the method used to display approximate channel error rate on the feasibility model front panel meter. With soft decisions the integrated metric normalization signal is still roughly proportional to p when p is low. This is because most soft decision errors that occur have a metric of 1 associated with them. As p increases, however, the integrated count will rise somewhat faster than linearly. Thus, the channel error rate meter would not be as accurate with soft decisions as with hard, unless separately calibrated.

2.3.5 Sensitivity to AGC Inaccuracy

Coded systems which make use of receiver outputs quantized to more than two levels require an analog-to-digital converter at the modem matched filter output, with thresholds that depend on the noise variance. For instance, all of the 8-level quantized Viterbi decoder simulations reported on thus far have used level thresholds at 0, $\pm 0.5\sigma$, $\pm\sigma$, and $\pm 1.5\sigma$.

Since the level settings are effectively controlled by the automatic gain control (AGC) circuitry in the modem, it is of interest to investigate the sensitivity of decoder performance to an inaccurate or drifting AGC signal. Fig. 2.15 shows the decoder performance variation as a function of A-D converter level threshold spacing (in all cases the thresholds are uniformly spaced). These simulations used the K=5 rate 1/2 code, with E_b/N_o = 3.5 db. It is evident that Viterbi decoding performance is quite insensitive to wide variations in AGC gain. In fact, performance is essentially constant over a range of spacings from 0.5σ to 0.7σ. This allows for a variation in AGC gain of better than 20% with no significant performance degradation.

2.4 Performance of a Rate 3/4 Code

The preceding sections have concentrated on Viterbi decoding of rate 1/2 convolutional codes. Most of the results on performance fluctuation due to decoder parameter variation carry over qualitatively, if not quantatively, to other code rates.

Code rates less than 1/2 will buy improved performance at the expense of increased bandwidth expansion and more difficult symbol tracking due to decreased symbol energy to noise ratios. Rates above 1/2 conserve bandwidth but are less efficient in energy.

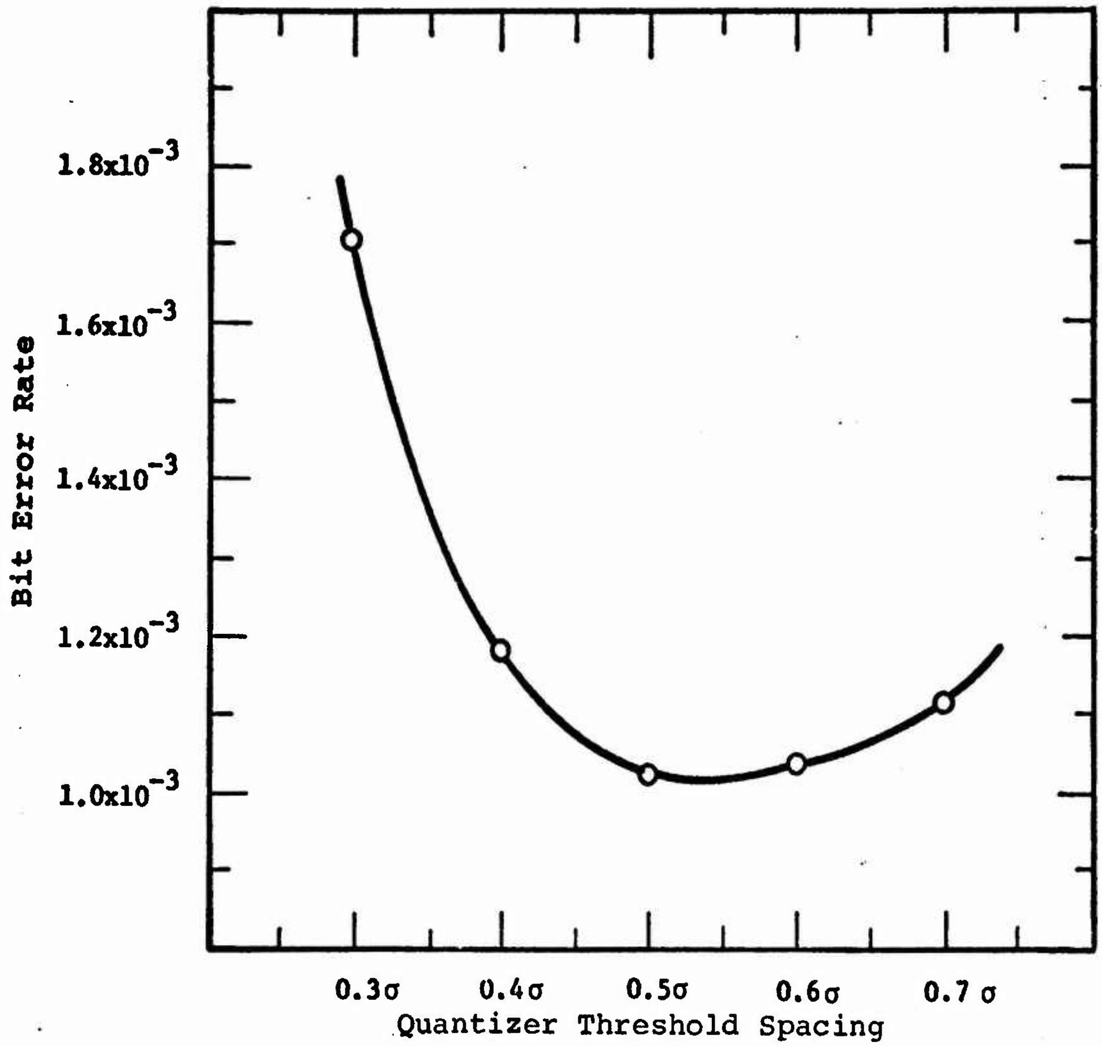


Fig. 2.15 - Viterbi decoder bit error rate performance as a function of quantizer threshold level spacing - K=5, rate 1/2, E_b/N_o=3.5 db, 8-level quantization with equally spaced thresholds.

Using a code search technique, the optimum K=6, rate 3/4 code has been found (the actual number of binary stage in this encoder shift register is 2 x 3 = 6 -- see Fig. 2.1). The encoder for the code is shown in Fig. 2.16. Table 2.2 shows the generators of this code along with the free distance d_f, the number of bit errors in paths at the free distance n_e, and the value of an upper bound on minimum free distance d_f^*, analogous to that obtained for rate 1/2 codes.

Fig. 2.17 shows numerical bound and simulation performance results for the optimum rate 3/4, K=6 code. Simulation curves are for 2 and 8-level quantization, while the numerical performance bound curves are, as usual, for infinitely fine receiver quantization.

2.5 Imperfect Carrier Phase Coherence

Thus far it has been assumed that carrier phase is known exactly at the receiver. In real systems, of course, oscillator instabilities and uncompensated doppler shifts necessitate closed loop carrier phase tracking at the receiver. Since the carrier loop tracks a noisy received signal, the phase reference it provides for demodulation will not be perfect.

An inaccurate carrier phase reference at the demodulator will degrade system performance. In particular with BPSK modulation a constant error, ϕ, in the demodulator phase will cause the signal component of the matched

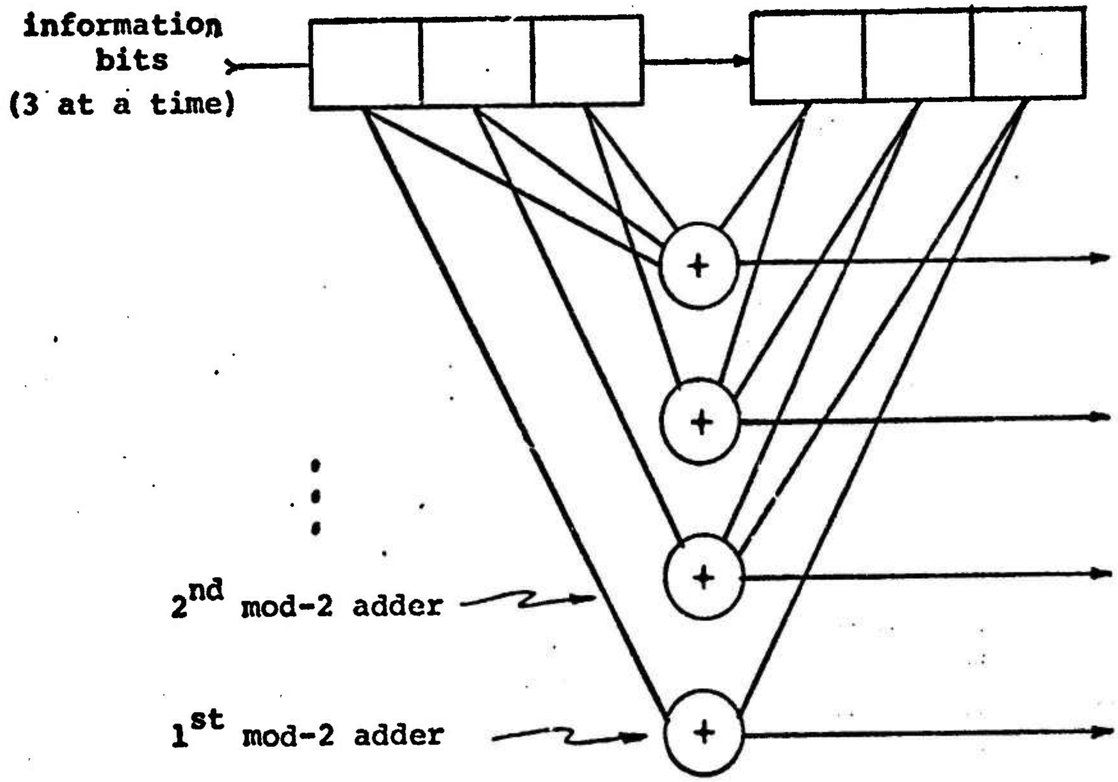


Fig. 2.16 - Optimum Rate 3/4,
K=6 Code Encoder.

K	Code Generators	d_f	n_e	d_f^*
6	100001 010011 001110 111101	4	40	4

Table 2.2 - Optimum Rate 3/4 Code.

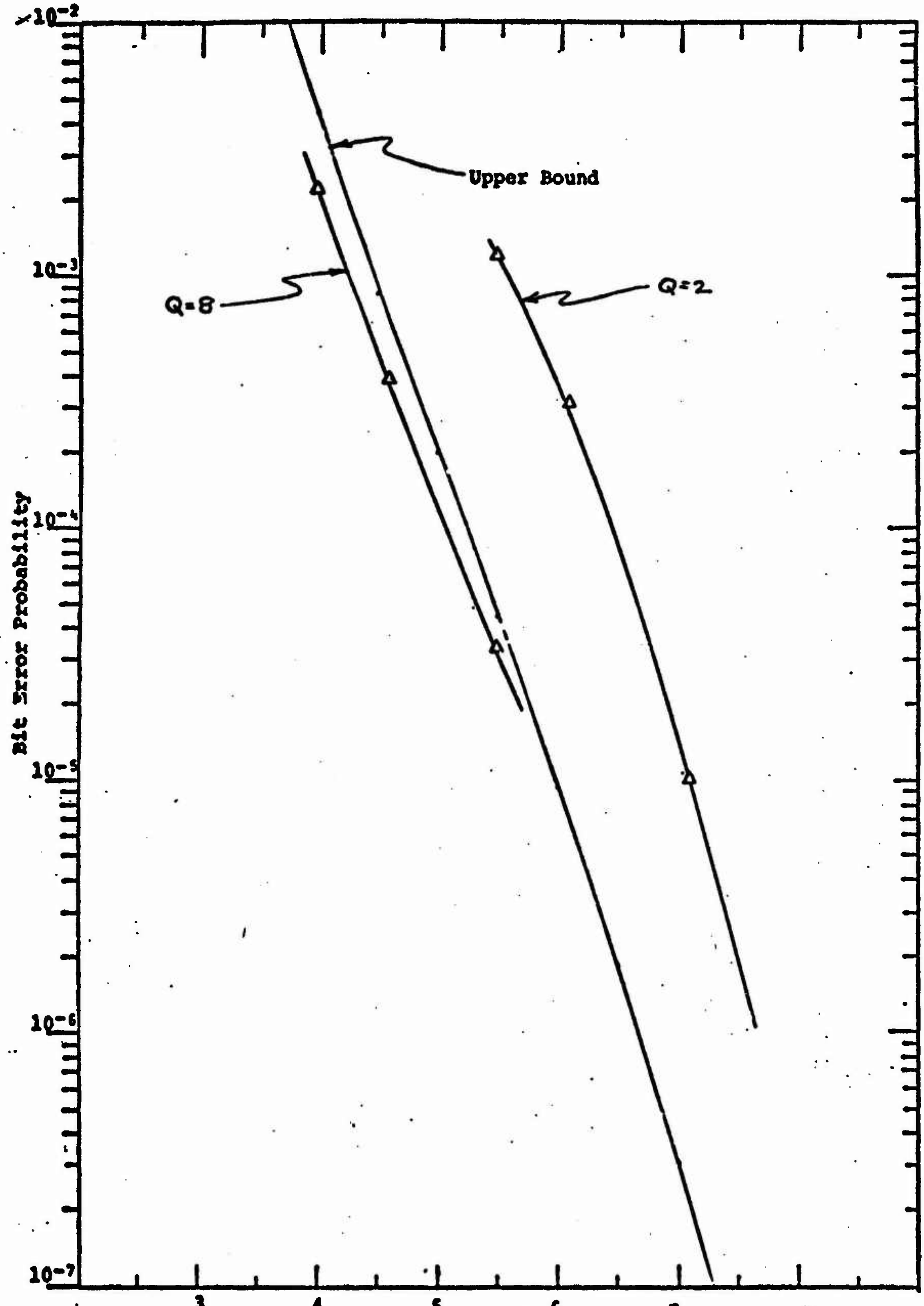


Fig. 2.17 Performance of a Rate 3/4, K=6 Code with Viterbi Decoding -- Numerical Bound and Simulation Results.

filter output to be suppressed by the factor cos ϕ (Ref. 8, Chapter 7).

$$r_j = \pm \sqrt{\frac{2E_s}{N_o}} \cos \phi + n_j \quad . \tag{2.7}$$

An imperfect carrier phase reference causes an apparent loss in received energy to noise ratio. Furthermore, unless care is taken in the design of the phase tracking loop, the phase error might be higher for the coded system than for an uncoded system, since loop performance may depend upon E_s/N_o, which is significantly smaller for coded than uncoded systems.

For convolutional coding with phase coherent demodulation and Viterbi decoding, exact analytical expressions for bit error rate, P_e, vs. E_b/N_o are not attainable. The simulation results of the preceeding section, however, defines a relationship between P_e and E_b/N_o which can be written formally as

$$P_e = f\left(\frac{E_b}{N_o}\right) \quad , \tag{2.8}$$

for a given code, receiver quantization, and Viterbi decoder. Since the carrier phase is being tracked in the presence of noise, the phase error ϕ will vary with time. To simplify analysis, we will assume the data rate is large compared to the carrier loop bandwidth so that the phase error does not vary significantly during perhaps 20 to 30 information bit times. Viterbi decoder output errors are typically several bits in length and are very rarely longer than 10 to 20 bits when the overall decoder bit error probability is less than 10^{-3}. Therefore, the phase error is assumed to be constant over the length of almost any decoder error. This being the case, the bit error probability for a constant phase error ϕ, can be written as

$$P_e(\phi) = f\left(\frac{E_b}{N_o} \cos^2\phi\right), \qquad (2.9)$$

from Eqs. (2.7) and (2.8). This result uses the fact that received signal energy is degraded by $\cos^2\phi$. If ϕ is a random variable with distribution $p(\phi)$, the resulting error probability averaged on ϕ is

$$P_e' = \int_{-\pi}^{\pi} p(\phi)\; P_e(\phi)\, d\phi \qquad (2.10)$$

For the second order phase-locked loop

$$P(\phi) = \frac{e^{\alpha\cos\phi}}{2\pi I_o(\alpha)}, \quad \alpha >> 1, \qquad (2.11)$$

where $I_o(\;)$ is the zeroth order modified Bessel function and α is the loop signal to noise ratio. Using this distribution and the P_e vs. E_b/N_o curve for the K=7, rate 1/2 code of Fig. 2.4, the P_e integral of Eq. (2.10) has been evaluated for several values of α. The results are shown in Fig. 2.18 as curves of P_e' vs. E_b/N_o with α as a parameter (the K=7, rate 1/2 simulation curve of Fig. 2.4 was extrapolated to get the high E_b/N_o results shown in this figure). These curves exhibit the same general shape as those for uncoded binary PSK modulation with phase coherence provided by a carrier tracking loop.

When QPSK modulation is used, analysis becomes more difficult. This is because the apparent energy to noise ratio on a received code symbol when there is a constant phase error, ϕ, is degraded by the factor

$$(\cos\phi \pm \sin\phi)^2$$

(Ref. 10) where the sign depends upon the code symbol modulating the quadrature dimension. The corresponding averaged expression for bit error probability cannot be obtained analytically because it depends on the particular sequence of code symbols generated.

Simulation provides another method for determining performance of Viterbi decoding with QPSK modulation and imperfect phase coherence. Simulations were performed using pseudo-randomly generated data and determining the sign of the signal-to-noise ratio degradation factor by observing the sequence of code symbols transmitted. Each simulation

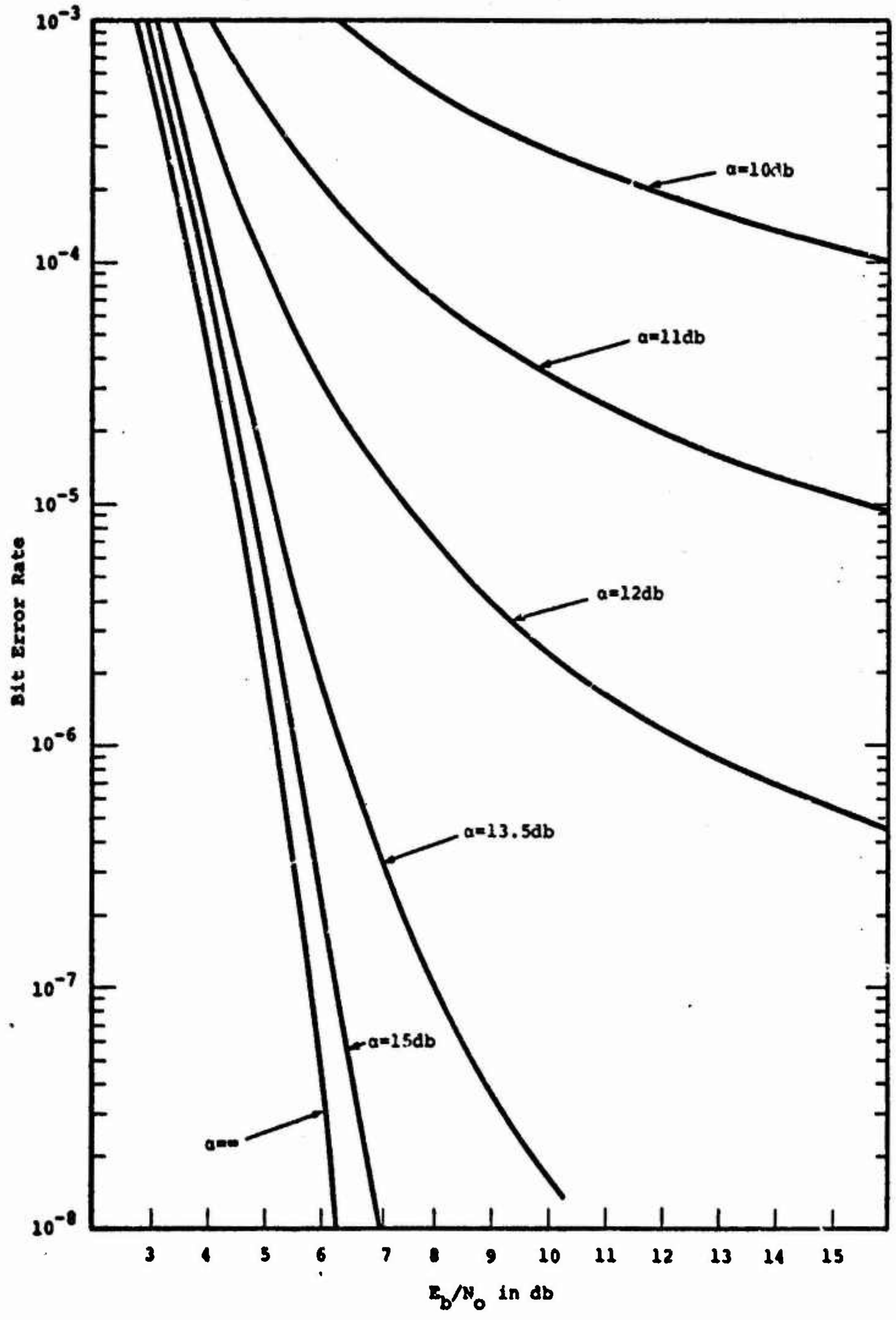


Fig. 2.18 - Performance curves for a rate 1/2, K=7 Viterbi decoder with BPSK modulation and 8-level quantization as a function of carrier phase tracking loop signal-to-noise ratio, α.

run was performed with a different but constant phase error. The resulting emperical curve of error probability vs. ϕ was averaged on ϕ using Eqs. (2.10) and (2.11). Before discussing the results, it is interesting to investigate the relationship between the signal-to-noise ratio degradation with and without coding.

When coding is used, the P_e vs. E_b/N_o curve is always steeper than with no coding. Suppose, for the moment, that both curves are linear in E_b/N_o over a significant range of P_e. That is

$$
\begin{aligned}
P_{eu} &\cong C_1 - C_2\, E_b/N_o \\
&\text{and} \qquad\qquad\qquad\qquad\qquad\qquad (2.12) \\
P_{ec} &\cong C_3 - C_4\, E_b/N_o
\end{aligned}
$$

If this is the case, it can be shown that the degradation in E_b/N_o is the same in both cases. The degradation in performance for the uncoded system has been obtained previously and is shown in Fig. 4-3.1 of Ref. 10. Assuming the same performance degradation, the performance of a K=7 rate 1/2 Viterbi decoder with QPSK modulation as a function of carrier loop signal-to-noise ratio is shown in Fig. 2.19. Two facts testify as to the accuracy of these curves in the region shown:

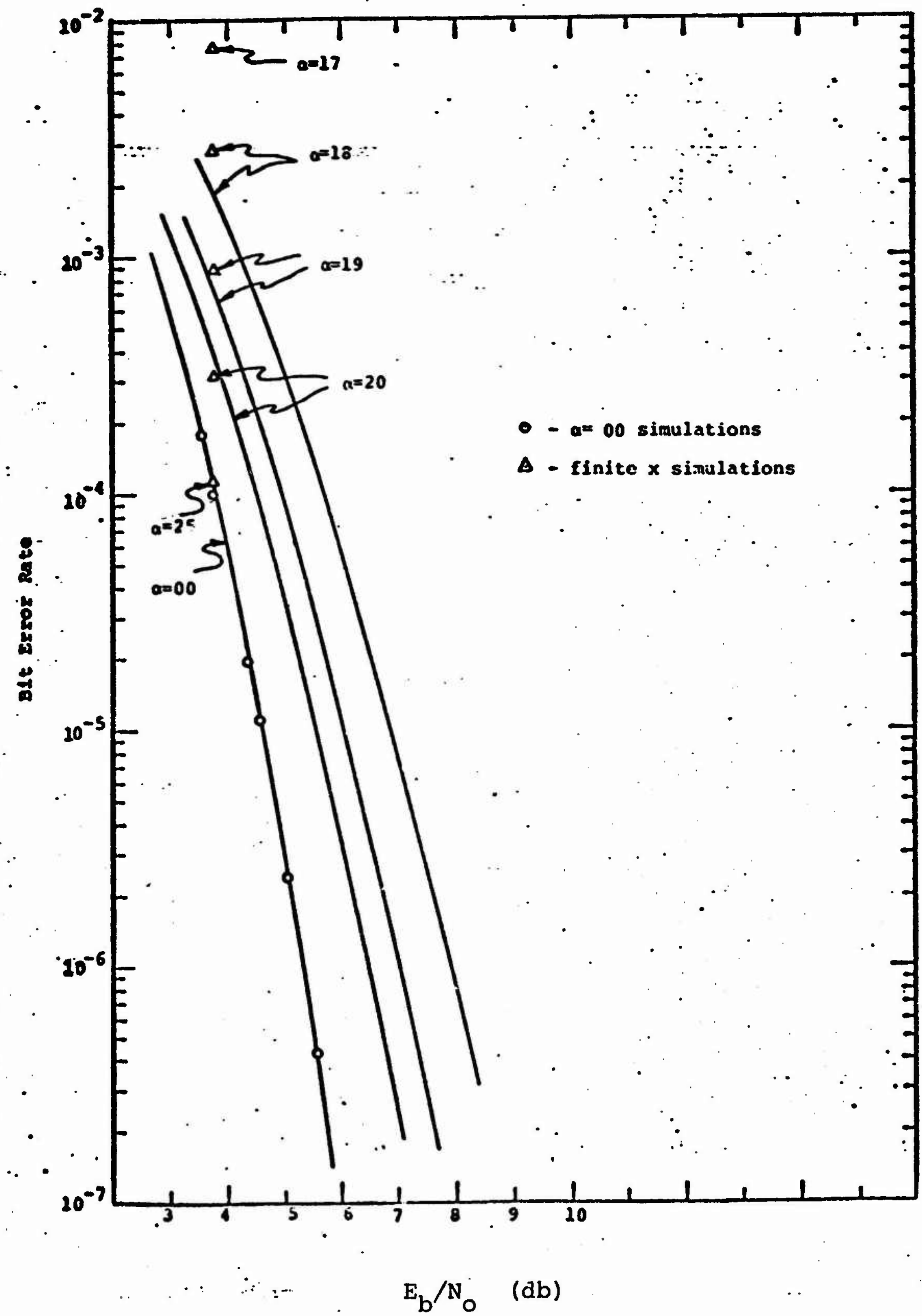


Fig 2.19 Performance curves for rate 1/2, K=7 Viterbi decoder with QPSK modulation and 8-level quantization as a function of carrier phase tracking loop signal-to-noise ratio, α.

1) The P_e vs. E_b/N_o curves appear linear in the 10^{-3} to 10^{-6} region for both the uncoded and K=7 coded cases.

2) The simulation results are in excellent agreement with these curves. The simulation results are also shown in Fig. 2.19. The only value of E_b/N_o for which simulation was performed is 3.7 db. This is because of the large amount of computer time required to collect data. Simulation must be performed for a range of phase error values for each choice of E_b/N_o. The actual and predicted QPSK performance will be in closer agreement at higher E_b/N_o; thus, the close agreement of the simulations with the curves of Fig. 2.19 is a convincing indication of the accuracy of the curves at higher E_b/N_o.

Observation of Figs. 2.18 and 2.19 shows that performance is degraded more with QPSK than BPSK modulation for any fixed loop signal-to-noise ratio α. This is expected because of the interplay between symbols modulated in quadrature when a phase error exists.

REFERENCES FOR SECTION 2

1. A. J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," IEEE Transactions on Information Theory, Vol IT-13, No. 2, April, 1967.

2. J. A. Heller, "Short Constraint Length Convolutional Codes," Jet Propulsion Laboratory, Space Programs Summary 37-54, Vol. III, pp. 171-177, October-November, 1968.

3. J. A. Heller, "Improved Performance of Short Constraint Length Convolutional Codes," Jet Propulsion Laboratory, Space Programs Summary 37-56, Vol. III, pp. 83-84, February - March, 1969.

4. LINKABIT Corporation, Final Report on a Coding Systems Study for High Data Rate Telemetry Links, Contract NAS2-6024, NASA Ames Research Center, Moffett Field, California.

5. R. M. McEliece and H.C. Rumsey, "Capabilities of Convolutional Codes," Jet Propulsion Laboratory, Space Programs Summary 37-50, Vol. III, 1968.

6. J. P. Odenwalder, Optimum Decoding of Convolutional Codes, Ph.D. Thesis, Systems Science Department, UCLA, Los Angeles, 1970.

7. A. J. Viterbi, "Convolutional Codes and Their Performance in Communication Systems," IEEE Transactions on Communication Technology, Vol. COM-19, No. 5, October, 1971.

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

9. LINKABIT Corporation, Theory, Operation and Maintenance Manual for the LINKABIT LV7026 Viterbi Decoder, Contract DAAB07-71-C-0148, USASATCOMA, Fort Monmouth, New Jersey.

10. Comsat Laboratories, The Conceptual Design of a Time-Division Multiple-Access System for the DSCS, Contract DAAB07-69-0392, USASATCOMA, Fort Monmouth, New Jersey.

3.0 VITERBI DECODER IMPLEMENTATION AND USAGE

The general performance of convolutional encoders and Viterbi or maximum likelihood decoders were described in Section 2.0 for a range of parameters and system conditions. We consider here the cost and complexity of implementing decoders over a range of speeds, utilizing TTL, MECL 10,000 and MECL 3 logic families but restricting attention to constraint length 7, soft-quantized, rate 1/2 decoders.

Total system cost can be reduced by time-sharing decoders. It is demonstrated in Section 3.2 that time-sharing can be accomplished without informing the decoder of transitions between users. No modifications of decoders are required nor do any control signals need to be supplied to the decoder. Similarly, system initialization may be accomplished by entering a short string of zero data, preferably equal to four constraint lengths, before beginning the initial decoding.

3.1 Cost-Speed Tradeoffs

The speed of a Viterbi decoder is principally limited by the requirements of performing 2^{K-1} add-compare-select (ACS) operations during each bit time. The arithmetic section of the decoder which carries out a single ACS operation is referred to as an ACS unit. For a constraint length 7 decoder, $2^6 = 64$ ACS operations are required for each bit decoded. These operations can be time-shared among several ACS units. Typically, we consider using 1 ACS unit for 32, 4, 2, or 1 calculations, since these choices permit convenient multiplexing. The number of ACS units required for a K=7 decoder is thus 2, 16, 32, or 64 respectively.

The speed of an ACS unit is controlled principally by the logic family from which it is constructed and to a smaller extent by the degree of sharing. Multiplexing the decoder among 4 calculations requires slightly more time than 2-way multiplexing, which itself requires more time than if no multiplexing is carried out.

A second decoder segment which limits speed is the decision memory. For speeds below 10 Mbps, relatively inexpensive LSI memory circuits can be used, providing storage of 64 bits per chip or greater. For higher speeds, it is necessary to utilize individual latches with 4 or fewer bits per chip. Thus, parts count and cost begin to escalate rapidly above 10-15 Mbps.

All of these factors have been examined and the cost and parts complexity of various decoders estimated. Results are presented in Table 3.1. Since actual cost depends upon production volume and environmental specifications, the cost shown in Table 3.1 is an estimated cost relative to that of the LV7026 decoder. Such relative costs are useful for evaluating different system configurations. For rough

DECODER	MAXIMUM DECODER SPEED (Mbps)	RELATIVE COST	APPROXIMATE PARTS COUNT	READY FOR ENGINEERING MODEL
TT32	0.2	0.65	170	No
TT4	2	1.0	350	Yes
TT2	4.2	1.5	500	Yes
MT4	7	1.7	380	Yes
TT1	8.5	2.0	700	Yes
MT2	14	2.5	530	No
MT1	30	5.0	1000	No
MM2	40	9.0	1400	No
MM1	80	20.0	3200	No
*LS4157	80 (Burst)	3.5	750	Yes

*The LS4157 is a hard decision sequential decoder included for comparative purposes.

Table 3.1 Relative Cost and Complexity of Viterbi Decoders budgetary purposes, unit cost is between $10,000 and $20,000, exclusive of non-recurring costs including an engineering development model.

In Table 3.1, the first two letters in the decoder designation indicate whether TTL (T) or MECL (M) logic is used in ACS units and decision memory respectively. The number indicates the number of ACS calculations performed by one ACS unit. The model denoted TT4 is the present LV7026 decoder using TTL throughout and time-sharing the ACS units among 4 ACS operations.

The parts count is a careful estimate of the required number of DIP's to implement the decoder using presently available components. If decoders are manufactured in quantity, parts count can be reduced and reliability increased by using hybrid technology. For example, two ACS units and associated multiplexors and metric storage might be placed in one package. Such savings are not reflected in the relative cost.

Development of the LV7026 has proved out the present LINKABIT design. Those Viterbi decoders for which a "Yes" is indicated under the heading "Ready for Engineering Model" are sufficiently similar to the present decoder to allow construction of an engineering model without further feasibility development.

3.2 Time-Sharing Viterbi Decoders

In Section 5, two system configurations, burst-rate decoding and aggregate-rate decoding, are considered in which one Viterbi decoder is time-shared among several users. In order to do so, each user must "tail-off" his transmissions by sending K-1 known bits, usually 0's, following the last information bit in the transmission. For a constraint length K=7 decoder, a tail of six 0's suffices. The tail serves to return the coder to the 0-state at the completion of each transmission.

Each new user, whether local or remote, must also begin his transmission with the encoder in a known state. We now show that if this known state is also the 0-state, it is unnecessary for the decoder to known where one user ends and the next begins. Thus, no change in decoder design whatsoever is required to permit time-shared operation. In particular, the LV7026 decoders delivered under this contract can be used in a time-shared mode.

First consider a decoder operating on data from a single source. Bits have some probability of being in error; in general, the errors occur in short bursts. In particular, suppose the binary sequence x_1, x_2, ..., x_j, 0, 0, ..., 0, x_{j+1}, ... x_m is encoded, transmitted, received, and decoded. The group of 0's in the middle of the sequence have some probability of being in error;

indeed, a decoder output error burst may span several of the 0's and x's. However, if the system is operating at an E_b/N_o which yields a bit error probability of 10^{-5}, all bits in the output have a uniform probability of individually being in error equal to 10^{-5}. No bit is favored over any other bit.

Next suppose that there are K-1 zeros between x_j and x_{j+1}, so that the encoder is in the zero state when x_{j+1} is encoded. As long as the decoder does not know of the presence of the zeros, it operates as above, providing a uniform bit error probability. Finally, assume that x_{j+1}, x_{j+2}, ... originates in a different encoder from that coding the tailed-off sequence x_1, x_2, ..., x_j, 0, ..., 0. As long as the different encoder is initially in the zero-state, the decoder need not know of the switch and decodes as before. Of course, before outputting the data, the decoded tail, which should be zeros but may contain some 1's as errors, must be discarded.

The above procedure permits time-sharing the decoder without its knowledge. This technique is recommended although it does not provide optimum performance. If the decoder were informed of the presence of the tail, and therefore based its decisions on knowledge that it must be in the zero state at the conclusion of the tail, bits on either side of the tail would have an error probability lower than average. The system advantage of providing a performance improvement for a few bits does not appear to justify the additional circuitry required in the decoder. The simpler and less costly technique of time-sharing the decoder without its knowledge is therefore assumed for use in Section 5.0.

3.3 Paralleling Decoders

Table 3.1 estimates the cost of implementing single decoders capable of achieving various operating speeds. Another possibility, that of paralleling decoders to achieve high burst data rates, at first glance appears more attractive at certain operating speeds. For example, three 30 Mbps decoders operating in parallel would together achieve a data rate of 90 Mbps at a cost of 3 x 5.0 = 15.0 in contrast to a single 80 Mbps decoder which has a cost of 20.0. The difficulty occurs during time-sharing. In order to use the three 30 Mbps decoders for burst-rate decoding, each subburst would have to consist of three interleaved subsequences separately encoded which could then be separately decoded. Unfortunately, each subsequence would require a separate six bit tail, thus imposing a significant penalty in terms of reduced frame efficiency (see Section 5.7). Single decoders avoid this penalty. In Section 5.8, parallel 7 Mbps decoders are used as aggregate-rate decoders. No penalty is involved here, since successive subbursts can be stored in different aggregate buffers and decoded in their entirety by one decoder. Only one tail is thus required, since interleaved sequences are avoided.

3.4 Viterbi Decoder Design Philosophy

A Viterbi decoder can be usefully broken into four sections: an input processor, an arithmetic processor, a state metric memory and a path decision memory (Ref. 1). Fig. 3.1 shows the interrelationships and signals that flow between the four major decoder sections.

As discussed in Section 2 of this report, the basic operation performed by a Viterbi decoder is to calculate the pair of path metrics corresponding to the two paths which enter a given state at a given time. These metrics are compared and the best is selected as the new metric of the given state. The new path history of this state is just the path history of the previous state shifted by one bit and augmented by the new comparison decision. The decoder output is taken to be the oldest bit on the path history of a state whose metric satisfied some criterion of goodness. For each bit decoded, the decoder must perform 2^{K-1} add-compare-select (ACS) operations - one for each encoder state. Also 2^{K-1} path metrics and path decision histories must be stored - again one for each state.

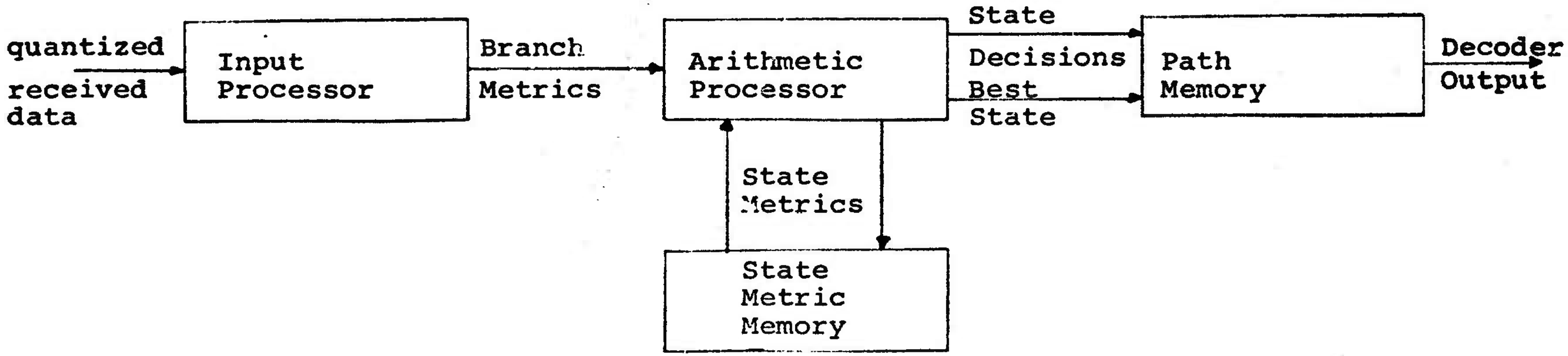


Fig. 3.1 - Block Diagram of a Viterbi Decoder

The Input Section

Fig. 3.2 shows a general portion of the code trellis or encoder state transition diagram for a rate 1/2 convolutional code. States x0 and x1 are the binary representations of two out of the 2^{K-1} states in the trellis at trellis depth ℓ. Here x is an arbitrary K-2 bit binary number. Note from Fig. 2.1 (with k=1) that if the encoder was in either state x0 or x1 at depth ℓ, then inputting a 0 will cause it to go to state 0x, whereas inputting a 1 causes it to go to state 1x. The pair of code symbols associated with the state transitions are also shown in Fig. 3.2. For example, inputting a 0 to the encoder when it is in state x0 causes it to go to state 0x and generate the code symbols c_1 and c_2. The binary pair of c_1c_2 can take on 4 possible combinations of values depending on x and the code used. The code symbols on the four branches of Fig. 3.2 are related in the manner shown in this figure. This is a result of the fact that the first and last stages of the encoder shift register are always connected to both mod-2 adders in good rate 1/2 codes.

Associated with each of the branches connecting states in Fig. 3.2 is a branch metric. This metric is a measure of the match between the code symbols on the branch and the corresponding pair of received (hard or soft

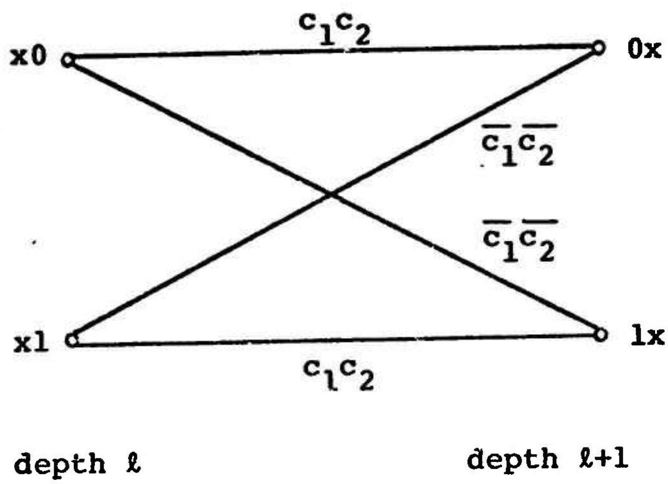


Fig. 3.2 - Representative Portion of Code Trellis for a Rate 1/2 Convolutional Code.

quantized) symbols. There are four branch metrics computed by the input processor for each bit time. These are denoted R_{00}, R_{01}, R_{10}, R_{11}, where the subscript denotes the pair of code symbols involved. The computed branch metrics are passed on to the arithmetic processor (Fig. 3.1).

The Path Metric Memory

The function of the path metric memory is to store each of the 2^{K-1} path metrics which result for the 2^{K-1} decoding steps associated with each bit time. These metrics are input to the arithmetic processor. The metric of state x0, for instance, is denoted M_{x0}.

The Arithmetic Processor

This part of the decoder performs the 2^{K-1} add-compare-select operations for each bit decoded. Each operation results in a binary decision as to which of the two paths entering a state is the most likely. For instance, if

$$M_{x0} + R_{c_1c_2} \leq M_{x1} + R_{\overline{c_1}, \overline{c_2}}$$

then D_{0x}=0 and the upper path (the one comi g through state x0) going to state 0x is selected. On the ner hand, if

$$M_{x0} + R_{c_1c_2} > M_{x1} + R_{\overline{c_1}, \overline{c_2}}$$

then $D_{1x}=1$; hence, the lower path is chosen (the convention is taken that smaller metrics indicate a higher likelihood).

The 2^{K-1} ACS operations may be done serially by one ACS unit or for increased speed, many ACS units may be used. In fact, for the highest operating rate, 2^{K-1} ACS units may be used - one for each state. In a high speed decoder containing many ACS units, the arithmetic processor represents a sizeable portion of the decoder circuitry. Doubling the maximum decoding rate of a decoder by doubling the number of ACS units thus results in a substantial increase in complexity. Sometimes it is more economical to use a different, faster logic family to achieve a higher decoding speed than to increase the number of ACS units with the same slower logic. This type of tradeoff was considered in obtaining the relative costs of various types of decoders shown in Table 3.1.

The LV7026 decoders have $2^6 = 64$ states. 16 ACS units are used in this decoder. Each one is used 4 times per bit decoded.

Path Decision Memory

The path decision memory stores the most recent 4 or 5 constraint lengths worth of decisions for each of the 2^{K-1} paths. New paths for each state are formed from one of the paths associated with the predecessor states augmented by the ACS decisions. For instance, if the upper path leading to state 0x in Fig. 3.2 is selected as most likely the new path for state 0x is the path associated with state x0 augmented by D_{0x}.

Each time the paths are re-stored in the path memory after a decoding step, the oldest bit on the paths are dropped off. The decoder selects as its output the oldest bit on a path which has a metric value below a fixed threshold.

3.5 The LV7026 Feasibility Model Viterbi Decoder

3.5.1 Encoder-Decoder Description and Performance

The LV7026 Viterbi decoders fabricated under this contract are constraint length K=7, hard or soft decision, rate 1/2 decoders. The theory of operation and detailed mechanical and electrical characteristics for these decoders is presented in a separate document (Ref. 2). We will present here the results of some tests to determine decoder performance and a summary of decoder specifications.

The encoder and decoder sections of the LV7026 are entirely independent, having separate controls and sharing only the power supplies, thus providing full duplex capability. The LV7026 is specifically designed to operate with binary-phase-shift keying (PSK) and quadri-phase-shift keying (QPSK) modems.

The system employs a code which is transparent to 180° phaseambiguities, and may operate in conjunction with differential encoding of the data source. When differential coding is not available in the associated modem, an internal differential decoder (in the Viterbi decoder section) can be switched into operation.

All necessary synchronization circuits are included in the decoder section. This includes node sync for serial PSK data and resolution of 90° phase ambiguities for parallel QPSK data.

The data rate at which the encoder/decoder system operates is completely determined by the frequency of the input clocks. The LV7026 is designed for operation at any data rate up to 1.8 megabits per second.

The logic design is optimized for the 1.8 Mbps maximum data rate based on currently available TTL medium-scale integrated circuits. Through the use of a number of LINKABIT developed techniques, the total number of integrated circuits necessary to implement the LV7026 is approximately 360, including both the encoder and decoder, interface circuits, and all other peripheral circuits.

The test configuration used in obtaining bit error rate vs. effective channel energy per bit to noise ratio is shown in Fig. 3.3. For the purposes of test, dummy data was generated using a length 255 PN sequence generator. Data is fed into the encoder portion of the LV7026, generating two parity streams (in the parallel output QPSK mode). These two streams feed the noise generator. The noise generator is an all digital system which generates a pair of 3 bit quantized numbers for each pair of parity bits input. These 3 bit numbers

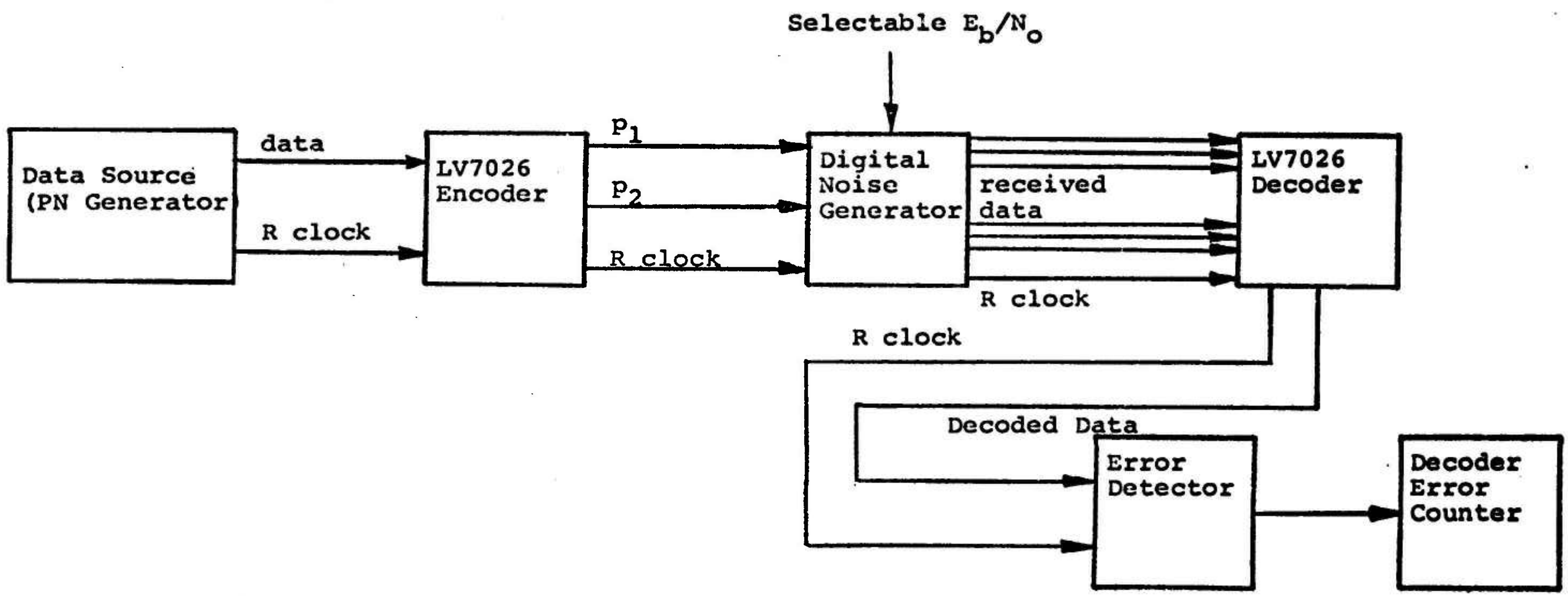


Fig. 3.3 - Test Configuration for LV7026 Encoder/Decoder

represent the soft decision demodulator outputs. For each parity symbol, the 8 possible demodulator outputs are assigned probabilities which are a function of E_b/N_0. The noise generator is implemented using 8 PN generators and 7 level comparators (Ref. 2). The set of comparison levels is selected by means of an E_b/N_o selector switch. Pairs of 3 bit quantized data are sent to the LV7026 decoder. Finally, the decoder output is compared with the original data sequence in the Error Detector and resulting errors are counted using an event counter.

The collection of performance data for hard quantized demodulator data uses the same test set-up shown in Fig. 3.3 except that only the most significant of the 3 soft decision output bits produced by the noise generator is used by the decoder. Tests were also made in the serial encode/decode mode, which is representative of BPSK operation. Here the encoder output is a single interleaved 2R parity stream. The decoder input consists of one code symbols worth of hard or soft decision data at a time. Both test configurations result of course in identical decoder performance.

Fig. 3.4 shows a comparison of the P_e vs. E_b/N_o performance of the LV7026 and a computer simulated decoder using the same K=7 rate 1/2 code. Results for both hard and soft (8-level) receiver quantization are shown. The agreement is excellent for hard quantization. The somewhat better than expected soft quantization results are due to two factors:

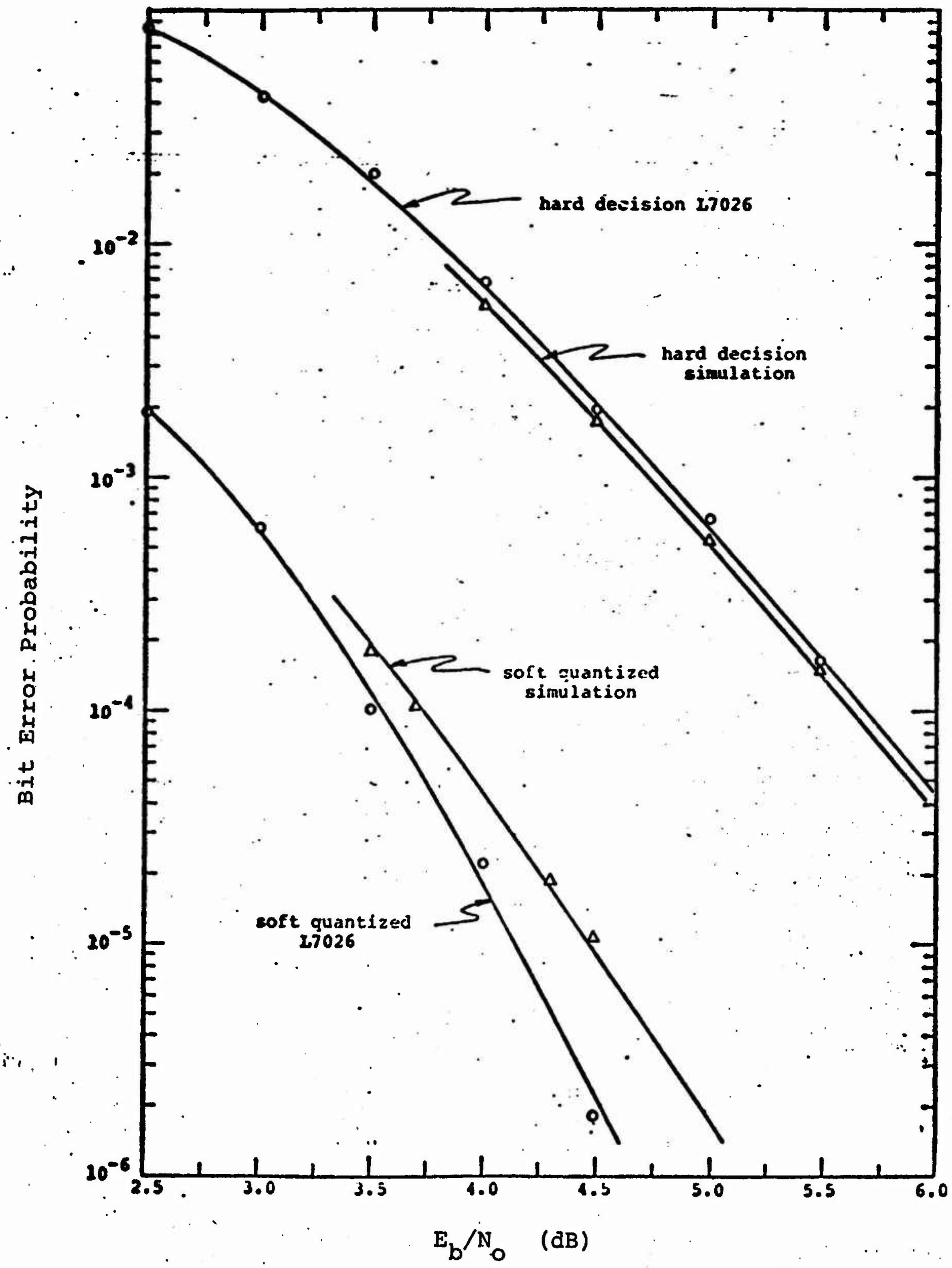


Fig 3.4 Performance of LV7026 compared with computer simulation data.

a) Quantization error in the digital noise generator. Soft quantized data was generated by means of a comparison of a random 8-bit word with a fixed level. Ties were resolved in favor of higher E_b/N_o.

b) The effective LV7026 path delay is somewhat longer than that of the computer simulation.

The decoder node synchronization (BPSK) and 90° phase ambiguity resolution (QPSK) technique used in the LV7026 is discussed in detail in Section 2.3.4. The observed values of the expected time to synchronization, E_{rs}; and expected time to falsely change of sync, E_{fa}, are shown in Figs. 2.10-2.13. The sync counter threshold values used in the LV7026 were 3072 for hard decisions and 1536 for soft decisions. The synchronization system parameters chosen were adequate to insure that degradation in performance due to false sync position changes is negligible for bit error rates less than 10^{-3}. Threshold values may be changed by means of a jumper wire as described in Ref. 2.

All tests performed on the LV7026 convolutional encoder - Viterbi decoders confirmed our previous theoretical and computer simulation results. The LV7026 encoder-decoders thus prove the feasibility of Viterbi decoding as a technique for improving satellite communication system efficiency.

3.5.2 Mechanical Description

The LINKABIT Model 7026 Encoder/Decoder is packaged in a 19 inch rack mountable chassis having a height of 7.9 inches and a depth of 22 inches. A photograph of the system appears in Fig. 3.5.

The chassis is constructed as a tiltable drawer with removable top and bottom covers allowing access to all portions of the system without removal from the rack.

The system is modular in construction, consisting of two power supplies, a tilt-out card cage containing 12 circuit boards, and a front and back panel.

There are no solder connections between the modules. All power cables are terminated by means of screw-down terminals, and all signal and control lines are implemented with two 34-conductor flat cables with plug-in connectors on both ends. The circuit boards plug into the card cage, thus providing easy removal and insertion.

One of the circuit boards, the driver board containing mostly discrete components, is fabricated as a printed circuit board. The remaining eleven boards and the back-plane are fabricated using a stitch wire technique. Power distribution is obtained on these boards through the use of a ground plane on the wire side and a V_{CC} plane on the component side, and signal inputs and outputs are accomplished be means of 100-pin edge connectors. The integrated circuits are mounted in the

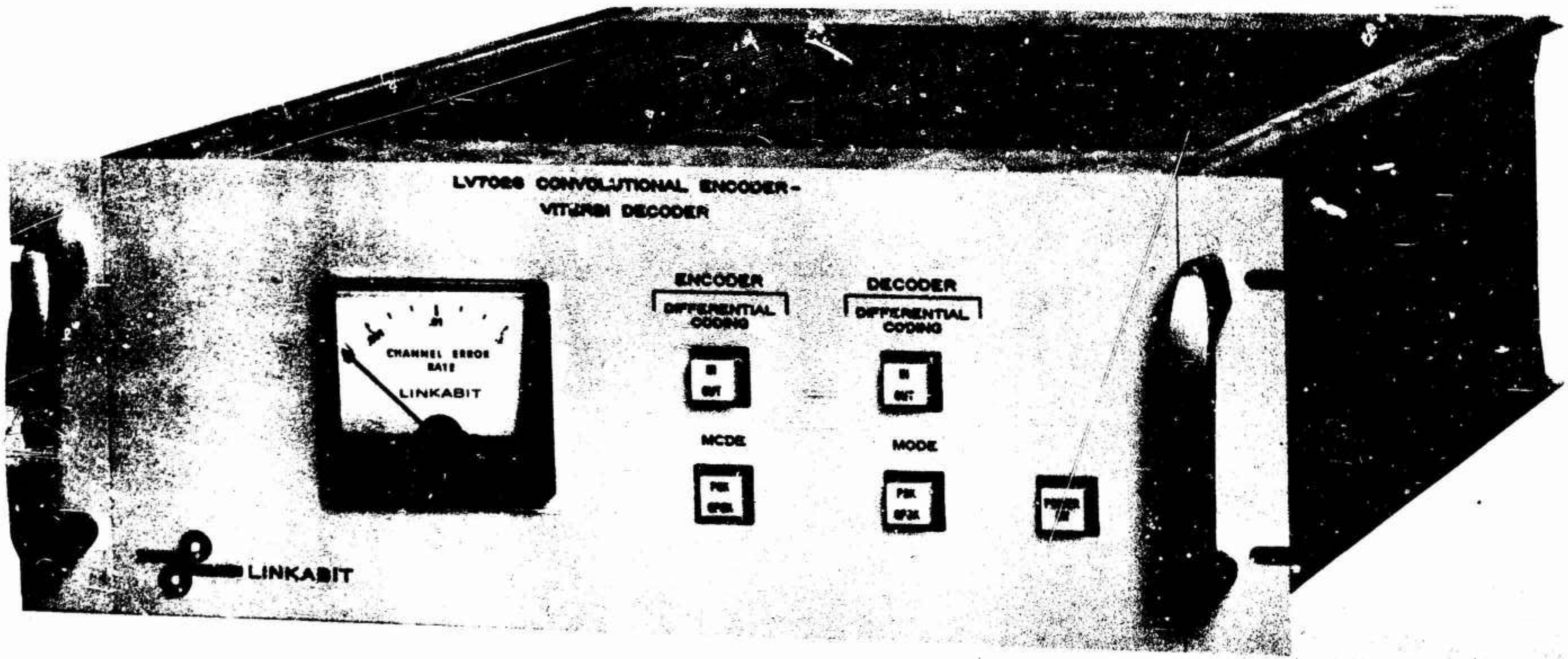


Fig. 3.5 LV 7026 CONVOLUTIONAL ENCODER-VITERBI DECODER

"dead-bug" position and are held in place by nylon clips. Decoupling capacitors are soldered in place and plate-through holes are used for this purpose. Other discrete components are soldered to the gold-plated steel terminals. Mechanical construction is to best commercial practice. The mechanical specifications are tabulated in Table 3.2.

TABLE 3.2

MECHANICAL SPECIFICATIONS

Dimensions	
Height	7.9 inches
Width	19 inches
Length	22 inches
Weight	78 pounds
Finish	
Front Panel	Painted aluminum, engraved
Back Panel	Aluminum, silk screened
Chassis	Aluminum panels, extrusions
Cooling	Forced air by means of fan on back panel.

3.5.3 Electrical Specifications

The LINKABIT Model LV7026 Encoder/Decoder is fabricated primarily from standard TTL integrated circuits. A single 5 volt power supply is provided for this portion of the system.

The interface circuits require a ±6.5 volt power supply which is provided as a single dual-output supply.

Switch control signals are carried to the card cage by means of 34-conductor flat cables. Data and clock signals are also carried on these flat cables, and to reduce cross-talk they are arranged with two grounds between pairs of adjacent signals.

Table 3.3 lists the electrical specifications for the system.

All signal and clock inputs and outputs are by means of BNC connectors located on the rear panel.

TABLE 3.3

ELECTRICAL SPECIFICATIONS

AC Power	
Line Voltage	105-132 VAC
Line Frequency	57-63 Hertz
Maximum Power Consumption	200 Watts
DC Power	
Primary power (TTL logic)	5 Volts ± 5% @ 17 Amps
Interface power (Driver board)	+6.5 Volts @ 300 ma -6.5 Volts @ 300 ma
Logic levels (not including interface signals)	
Logic 0	<0.8 Volt
Logic 1	>2.4 Volts

3.5.4 Interface Specifications

The interface circuits for the LV7026 comply with Annex I to USASATCOMA Technical Requirements SCA-2106.

The fundamental specifications for the transmitter and receiver circuits are outlined in Table 3.4. These specifications apply whether a particular transmitter or receiver is used for data or clock signals.

3.5.5 Organization of the LV7026

The logic circuitry for the LINKABIT LV7026 is organized into 12 cards of 7 different types. Table 3.5 lists each card by number, type, and a short description of its functions. Board types which appear in more than one board location are identical, e.g., boards numbered 4 and 5 are both ACS boards and are identical.

TRANSMITTER	
Open Circuit Voltage	
Logic 0	-6 ± 1 volt
Logic 1	$+6\pm1$ volt
Source Impedance	75 ohms $\pm$ 10%
Rise and Fall Times (with 75 ohm resistive load)	50 ns, nominal
RECEIVER	
Input impedance	75 ohms $\pm$ 10%
Shunt capacitance	< 10 pf
Sensitivity	
Min voltage to cause logic 1 output	+100 mv
Max voltage to cause logic 0 output	-100 mv

TABLE 3.4

Interface Transmitter and Receiver Specifications

Number	Name	
1	Driver Board	Line drivers for encoder and decoder data and clock outputs
2	Interface Board	Line receivers for encoder and decoder data and clock inputs Convolutional encoder Differential encoder and decoder
3	Input/Clock/ Sync Board	Branch metric computations Main timing generation Sync
4	ACS Board	Add-Compare-Select board Pairwise decisions
5	ACS Board	
6	ACS Board	
7	ACS Board	
8	Metric Memory	State metric storage 128 bits of memory per board
9	Metric Memory	
10	Best State/Normalization Board	Best State selection for use by output circuitry Normalization generation to maintain metric magnitudes within specified limits Decision Memory addressing and control Output circuitry
11	Decision Memory	Storage for decision bits D_{Ox}, D_{1x} generated by ACS boards 2048 bits of memory per board
12	Decision Memory	

TABLE 3.5

Tabulation of Logic Boards in LV7026

REFERENCES FOR SECTION 3

1. LINKABIT Corporation, "A Very High Speed Viterbi Decoder for Convolutional Codes," Unsolicited Proposal Submitted to USASATCOMA, Fort Monmouth, N.J., September, 1969.

2. LINKABIT Corporation, "LV7026 Convolutional Encoder - Viterbi Decoder Instruction Manual," Contract No. DAAB07-71-C-0148, USASATCOMA, Fort Monmouth, N.J.

PART II - VITERBI DECODER - TDMA CONSIDERATIONS

Previous .udies[1, 2] have sought to optimize TDMA system parameters such as frame rate, preamble length, etc., with respect to projected network requirements of the late 1970's and beyond. While coding was considered in these efforts, it was given, for the most part, a cursury treatment. What is more important, the use of Viterbi decoding with soft receiver quantization, which is a relatively recent development, has not been considered previously at all.

It now appears that Viterbi decoding is the most cost-effective signal processing technique for reducing the required energy-per-bit to noise ratio, E_b/N_o, by about 5 db at a 10^{-5} bit error rate, for data rates up to several tens of megabits per second. The remainder of this report will be concerned with a thorough analysis of the interrelation of Viterbi decoding and the rest of the TDMA system. As a point of departure, the recommendations of the COMSAT report[1] will be used (suitably modified to conform to recent changes in system requirements). Fortunately, many of the TDMA system parameters are relatively insensitive to the type of coding-decoding used.

The use of soft decision Viterbi decoding does have a major impact on the burst compression-expansion buffer subsystem however. This is because there is a factor of six increase in required buffer size if decoding is done at the user vs. the burst rate due to the necessity of storing rate 1/2 soft quantized received data. The burst buffer Viterbi decoder optimization problem will be treated in Section 5 of this report.

One other area which deserves re-examination is the performance degradation due to imperfect phase coherence in a system which relies on burst to burst coherence. The low duty cycle (.001) carrier loop gate recommended results in significant performance degradations for burst rates less than 10 Mbps with QPSK modulation. This topic will also be treated in detail in this section.

4.0 REVIEW OF TDMA SYSTEM ASPECTS RELATED TO VITERBI DECODING

4.1 Frame, Burst and Sub-Burst Format

In the TDMA system under consideration, the frame repetition rate will be 1200 frames/second. In each frame, each active transmitting station is allotted a unique time slot during which that station and only that station may transmit. Thus, the stations transmit in bursts at the rate of 1200 bursts per second.

The bursts from each station are composed of a preamble followed by data directed to one or more receiving stations. The preamble consists of known data patterns used to acquire and track carrier phase, and bit/symbol timing used in coherent demodulation. The modulation used will be quadriphase-shift-keying (QPSK) or possibly binary phase shift keying (BPSK). The data portion of the bursts are composed of one or more sub-bursts. Each sub-burst is directed toward a particular user.

The transmitted data rate (burst rate) may vary from one sub-burst to another depending on the class of receiving station to which it is directed. Stations with higher G/T will naturally be capable of receiving higher burst rates than lower G/T stations, and hence make more efficient use of the TDMA system. The transmitted sub-bursts within a burst will be grouped together in order of ascending burst rates. The convention that bits/second always refers to <u>information</u> rate will be used in this report. When referring to the transmission of encoded data, the notation symbols/second will be used. Since rate 1/2 codes will be emphasized, the code symbol rate will usually be twice the information rate.

The maximum burst rate presently under consideration is 80 Mbps (160 mega symbols/second). This upper limit is imposed by system constraints which include the following:

a) The largest G/T station is capable of supporting 80 Mbps with the coding gain that a K=7 Viterbi decoder can provide,

b) The particular DSCS phase II satellite channel being addressed in this report (earth coverage-earth coverage) has a nominal bandwidth of 125 MHz. Even at a signalling rate of 160 mega-symbols/second, there will be a substantial system performance degradation due to inter-symbol interference (perhaps 2 to 3 db) (Ref. 3). This problem may necessitate limiting the maximum burst rate to a value somewhat below 80 Mbps.

Transmitting user data rates must be of the form

$$C_d \times 75 \times 2^{nd} \text{ bits/second}$$

where C_d is an odd integer. Since the frame rate is 75 x 24 = 1200 frames/second, each user will generate an integral number of bits per frame. For instance, a $75 \times 2^7 = 9600$ bps user subburst would consist of 8 bits in each frame. The total number of bits that can be transmitted in a frame depends on the burst rate(s) used. As a point of reference at the nominal burst rate of 80 Mbps (in actuality, 75×2^{20} bps), there would be

$$75 \times 2^{20}/1200 = 65{,}536 \text{ bits/frame}$$

Since all subbursts are not sent at 80 Mbps and guard spaces are needed between bursts, the total system capacity is always less than 65,536 bits/frame. For convenience, the time duration of a bit transmitted at 80 Mbps will be designated a basic time unit (BTU). There are 65,536 BTU's per frame. A bit is communicated in 1 BTU at a burst rate of 80 Mbps, in 2 BTU's at 40 Mbps, etc.

4.2 TDMA Station Configurations

The several G/T class TDMA stations along with link calculations will be reviewed in the following subsection. Here the typical TDMA portion of a station will be discussed. Refer to Figs. 4-27A, B, C, and 4-28 A, B, C of Ref. 1 in the following discussion. For the present, the encoder and decoder will not be considered.

At the TDMA transmitter, data arrives from one or more users (each user stream may in fact be a multiplexed bit stream from several smaller sources) asynchronously.

Each user stream is input to an asynchronous interface consisting of a pulse stuffing buffer. Data is clocked in with the user clock and clocked out using a divided down version of the TDMA system clock. A dummy bit stuffing strategy solves the problem of discrepancies between the clock rates. The asynchronous interface is followed by a burst compression buffer. This buffer stores sufficient user data for one subburst and outputs it at the burst rate when called upon to do so by the TDMA multiplexer. The multiplexer combines all user subbursts and a preamble to form a station burst. The burst so formed is then used to QPSK (or BPSK) modulate a carrier to form the transmitted signal.

At the receiver, the same process is repeated in reverse. Coherent demodulation is followed by demultiplexing and a burst expansion buffer for each user. The expansion buffer converts a received subburst to a continuous user data stream. This stream passes through a pulse destuffing buffer to remove dummy stuff bits and provide a useful user data stream.

This system description is presented as a point of departure. When soft quantized Viterbi decoding enters the picture, some system changes are required to minimize costs. This is especially true of the burst expansion buffers as will be shown in Section 5.0.

Master timing for the TDMA system is distributed to stations via a separate timing and control channel. This channel will use a small fraction of the total available power and will not be considered further in this report.

4.3 Carrier Phase Recovery - Burst to Burst Coherence

In Section 2.5, the performance of Viterbi decoding with an imperfect carrier phase reference was discussed. In particular, bit error rate vs. E_b/N_o curves were obtained for operation with QPSK and BPSK modulation with the tracking loop signal-to-noise ratios available for several G/T class stations, and the corresponding E_b/N_o losses will be studied.

From the receiving station G/T, and the effective radiated power, we may compute the received power-to-noise ratio using

$$P/N_o = ERP + G/T - k - \text{path loss} - M - m \qquad (4.1)$$

where

ERP = effective radiated power in dbW

k = Boltzmans constant (dbW/Hz - °K)

M = rain margin (6 db)

m = demodulation implementation margin (2 db)

A synchronous orbit path loss of 201 db will be assumed. All calculations are based upon earth coverage - earth coverage antennas with ERP = 28 dbW.

Substitution yields

$$P/N_o \quad = \quad 47.6 + G/T \qquad (4.2)$$

Now the proposed TDMA system will perform carrier tracking using only the unmodulated portion of the burst preamble. As many phase locked loops are needed at a station as there are subbursts of information directed toward that station. The received signal is observed by each loop only during the unmodulated portion of the preamble of that burst position being tracked by that loop. The duty cycle of the gating signal is about .001 of a frame time. Thus, the effective signal-to-noise ratio in the loop bandwidth B_L in db will be

$$\alpha \quad = \quad \frac{P}{B_L N_o} + 10 \log_{10}(.001) \quad = \quad \frac{P}{B_L N_o} - 30 \text{ db} \qquad (4.3)$$

With an anticipated loop bandwidth of B_L = 240 Hz, we get from eq. (4.2)

$$\alpha \quad = \quad (-6.2 + G/T) \text{ db} \qquad (4.4)$$

α and P/N are tabulated for several station classes in Table 4.1. The values in this table represent worst cases since they include a 6 db rain margin.

Clearly, for values of G/T in the R3 class and lower, the performance degradation due to imperfect coherence becomes significant when QPSK modulation is used.

4.4 Link Coding Requirements

The currently planned burst rates for the R1, R2, and R3 class stations are 80 Mbps, 40 Mbps and 10 Mbps respectively. We will now consider the coding gain necessary to support these rates.

The available E_b/N_o as a function of G/T is

$$E_b/N_o = P/N_o - 10 \log_{10} R$$

$$= 47.6 + G/T - 10 \log_{10} R \tag{4.5}$$

where R is the burst rate. The resulting available E_b/N_o for each station class is shown in Table 4.3.

Station Class	G/T	Burst Rate (R)	E_b/N_o
R1	39	80 Mbps	7.6 db
R2	34	40 Mbps	5.6 db
R3	27	10 Mbps	4.6 db

Table 4.3 - Worst Case E_b/N_o at each of three station classes receiving at the indicated burst rates.

The E_b/N_o required for a 10^{-5} bit error rate with Viterbi decoding is shown for constraint lengths K=5, 6, and 7 in Table 4.4.

K	E_b/N_o required for bit error rate of 10^{-5}
5	5.2 db
6	4.8 db
7	4.4 db

Table 4.4 - E_b/N_o require for bit error rate of 10^{-5} with Viterbi decoding vs. code constraint length K.

Clearly, a K=7 Viterbi decoder is needed for the R3 class of station receiving at a burst rate of 10 Mbps (20 mega-symbols/second). For R_2 and R_3 class stations, it appears at first glance that a smaller amount of coding gain will be sufficient (uncoded PSK provides a 10^{-5} bit error rate with E_b/N_o = 9.6 db). However, losses due to intersymbol interference at the higher burst rates will bring the actual effective E_b/N_o for these stations much closer to that of the R3 station. The coding gain afforded by a K=7 Viterbi decoder will therefore be necessary for all of the considered station classes.

Actually, the available E_b/N_o for the R3 class station will be further reduced by the imperfect coherence loss shown in Table 4.1 if QPSK modulation is used. This will bring the required coding gain out of the range of even the K=7 decoder. Since the available bandwidth (125 MHz) is large compared to the 10 Mbps receiving burst rate for R3 class stations, one possible way of almost eliminating the E_b/N_o loss due to imperfect coherence is to use BPSK modulation. QPSK could still be used in transmissions to stations with sufficient G/T to support burst rates over 10 Mbps; while, BPSK would be used for lower G/T stations.

REFERENCES FOR SECTION 4

1. Comsat Laboratories, "The Conceptual Design of a Time-Division Multiple-Access System for the Defense Satellite Communication System," Final Report on Contract DAAB07-69-0392, USASATCOMA, Fort Monmouth, N.J.

2. Philco-Ford Corporation, "TDMA Modem for Communication Satellite Links," Final Report on Contract DAABC7-68-C-0351, USASATCOMA, Fort Monmouth, N.J.

3. M. Jones, <u>Private Communication</u>

5.0 DECODER AND TDMA SYSTEM CONFIGURATIONS

Three alternative configurations have been examined for locating the Viterbi decoder within the TDMA system. These configurations, referred to as burst-rate decoding, aggregate-rate decoding, and user-rate decoding, differ in their impact on decoder speed, TDMA buffer organization and size, TDMA control configuration, frame efficiency, and system reliability and flexibility. All three alternatives provide the same communication efficiency.

For each alternative, the decoder may be viewed as a fixed length, non-volatile, clocked shift register. The clock need not be periodic. Each clock time, the decoder accepts two 3-bit numbers on six parallel lines as inputs and delivers a single bit on one line as output. Pairs of 3-bit numbers are denoted on Figs. 5.1-5.3 as 6X data, and represent soft quantized decisions from the inphase and quadrature QPSK detectors. The output line is denoted 1X data. During periods when no clock pulses are input, the decoder contents remain unchanged.

In Sections 5.1-5.3, we consider burst-rate, aggregate-rate, and user-rate decoding in detail, noting buffer, timing, and control requirements. Prior to evaluating these configurations, we present a DCS network model and consider details and costs of a new buffer design. In Section 5.7, the impact on frame efficiency of the three configurations is examined.

In Section 5.8, relative system costs for various decoder mixes in the three system configurations are tabulated, providing a firm basis for system recommendations.

5.1 Burst-Rate Decoding

A ground terminal receiver for terminal j of a TDMA system utilizing burst-rate decoding is illustrated in Fig. 5.1a. The terminal receives information from several transmitters, with the information appearing in subbursts of rate R_s bits per second (rate $2R_s$ check digits per second for the rate 1/2 code). Each subburst is part of a transmitted segment containing all of the rate-R_s data transmitted during the frame by the transmitter. Each constant-rate segment is terminated by a 6-bit tail, as shown in Fig. 5.1b.

The box marked TDMA in Fig. 5.1a accepts burst-modulated IF carrier from the Satellite Link Terminal Receiver and timing from the USC-28. It sends to the decoder 6X data and R_s clock during the rate-R_s segments of each transmitted burst which contains data for the present station. No R_s clock is sent to the decoder except when appropriate 6X data is available. The decoder thus operates in bursts, decoding only data contained in rate-R_s portions of each frame.

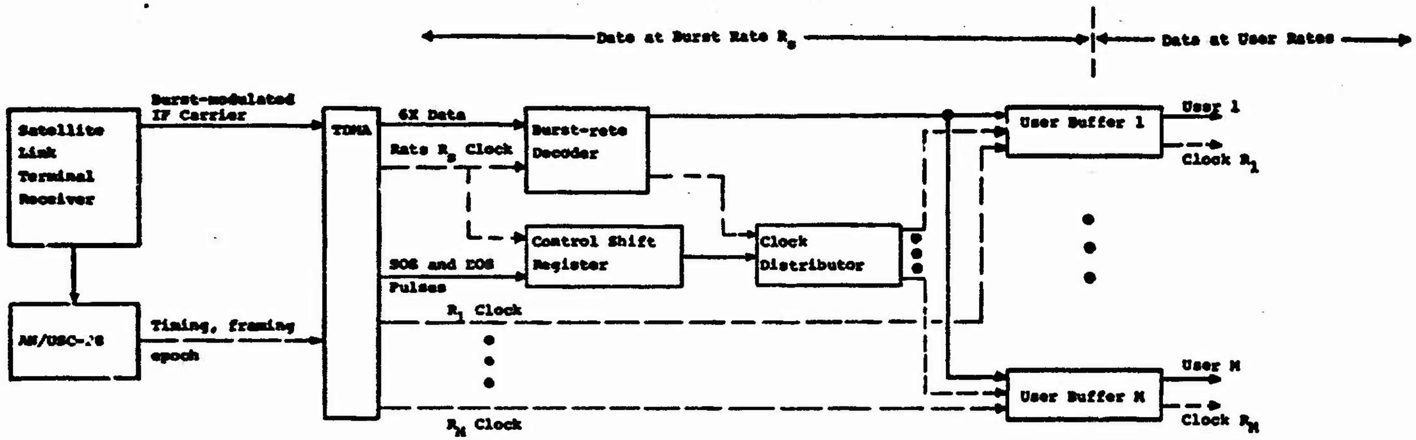


Fig. 5.1a - Burst-Rate Decoder Receiver Configuration

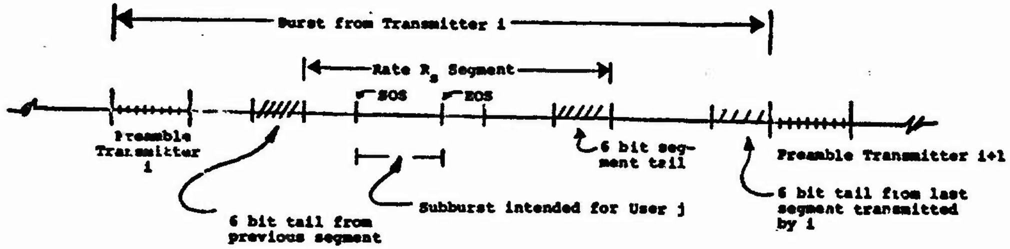


Fig. 5.1b - Burst Details for Burst-Rate Decoding

The use of 6-bit tails to terminate the fixed rate segments of each burst permits the decoder to effectively return to the zero-state at the completion of each fixed-rate segment decoded, thus permitting time-shared decoder operation (over successive rate-R_s segments) as discussed in Section 3.3. No start-of-segment nor end-of-segment signals need be sent to the decoder.

The TDMA also delivers to a control shift register two marker pulses, a "1" indicating start-of-subburst (SOS) and a second "1" marking end-of-subburst (EOS) for each sub-burst intended for station j. The input to the control shift register is zero otherwise. The control shift register is clocked by the same R_s clock that clocks the decoder and has a delay identical to that of the decoder. Thus, the SOS bit, a "1", arrives at the output of the control shift register at the same time the first bit destined for a new user is output by the decoder. The "1" enables the clock distributor, causing clock R_s to be gated to the appropriate user buffer. When clocked, the user buffer accepts decoded data from the decoder. The next "1" from the control shift register, the EOS indicator, disables the clock and then advances the clock distributor to the next user buffer line. The distributor remains dormant until the next SOS.

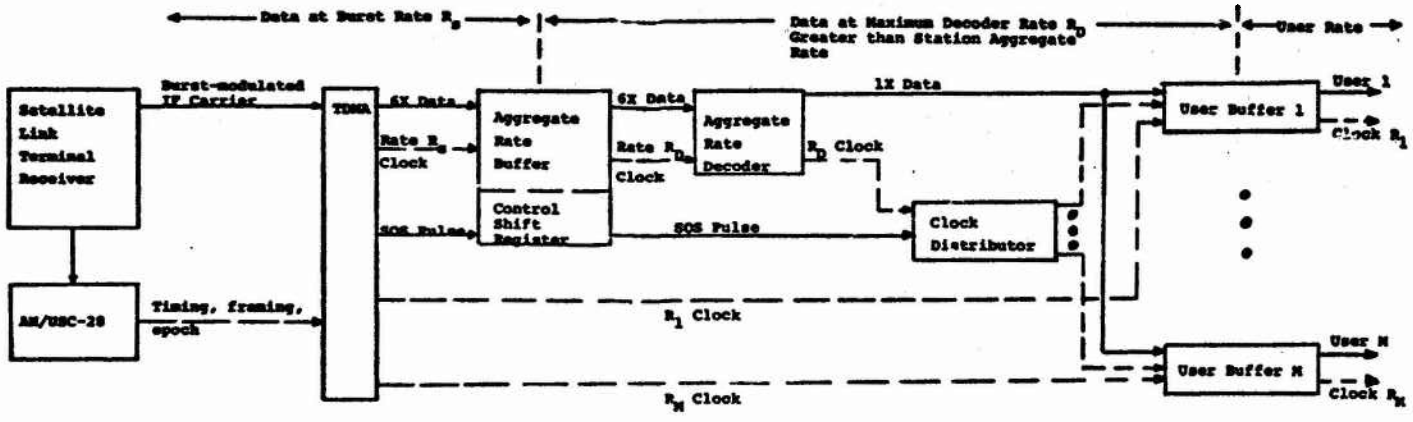
User buffers are emptied to the ICF at the user rates R_1, R_2, ..., R_M where R_i is the frame rate multiplied by the number of bits per frame for each user. The rate R_i is derived in the TDMA and supplied to the user buffers.

5.2 Aggregate-Rate Decoding

Aggregate-rate decoding is illustrated in Fig. 5.2. The aggregated rate, R_A, for a station is defined as the total information rate directed to the station from all other transmitters in the network. It is equally well described as the sum of the user rates for the station, $R_A = R_1 + R_2 + \ldots + R_M$.

The box marked TDMA in Fig. 5.2a accepts burst-modulated IF carrier and timing and provides 6X data and R_s clock to an aggregate-rate buffer during subbursts intended for the station. No clock is sent to the buffer except during such subbursts. Thus, only data intended for station j is accepted by the aggregate-rate buffer. Since data reaches the aggregate-rate buffer prior to decoding, the aggregate buffer must accomodate 6X data; that is, the number of bits buffered each frame is six times the number of information bits directed to station j per frame, due to the soft decision and rate 1/2 code.

The aggregate-rate buffer contains two alternating storage buffers. Details are given in Section 5.5. As soon as one storage buffer is full, input data is directed to the other buffer. The full buffer then empties at rate



demonstrating
SOS - Start of Subburst

Fig. 5.2a - Aggregate-Rate Decoder Receiver Configuration

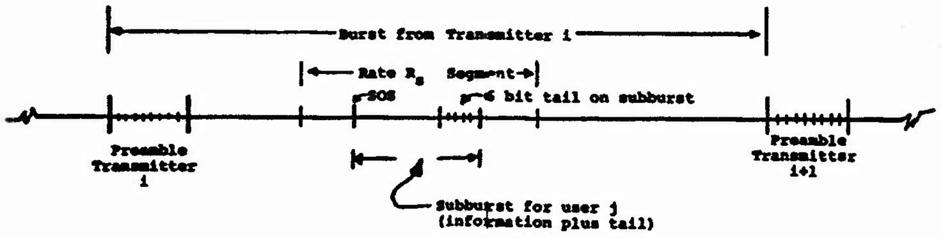


Fig. 5.2b - Burst details for Aggregate-Rate Decoding

R_D into the decoder. The rate R_D must be greater than the station aggregate-rate R_A so that the decoder does not fall behind. Otherwise, R_D need not be related to any other clock in the system and may be chosen for convenience. A particularly suitable choice for R_D is the maximum rate at which the decoder operates reliably, since this choice maximizes the acceptable aggregate rate.

A control shift register is again used to control a clock distributor which directs the output of the decoder into appropriate user buffers in a manner similar to that utilized in burst-rate decoding. It is convenient to implement this shift register as part of the aggregate-rate buffer, since the buffer delay must be included in the shift register delay and the appropriate clocks are available. A control simplification is possible in the aggregate rate configuration. Since only data contained in subbursts intended for station j are decoded by the decoder, rather than entire rate-R_s segments, only the start of subburst (SOS) is required for control. When the SOS pulse is received by the clock distributor, the distributor is advanced to the next user buffer clock line.

The SOS bit is used to serve a second function. As shown in Fig. 5.2b, for aggregate-rate decoding, each subburst in the frame is terminated with a 6-bit tail, permitting the decoder to be time-shared among subbursts.

The impact on frame efficiency is discussed later. By making the control shift register 6 bits short, the SOS bit is received at the clock distributor just at the time that the decoder begins to output the tail from the previous subburst. The tail is then simply deleted, by disabling the clock for 6 bit times. While the clock is disabled, the clock distributor is advanced.

5.3 User-Rate Decoding

User-rate decoding is illustrated in Fig. 6.3. The start of subburst (SOS) signal is used by the clock distributor to gate 6X data from successive subbursts into successive user buffers. Again, the R_s clock is on only during subburst intended for station j. Data is clocked out of user buffers into parallel decoders periodically at the user rates R_1, R_2, ..., R_M. The decoded data and clocks are fed to the ICF. Since decoders are not time-shared in the user-rate configuration, no tails are necessary.

Although this configuration is quite simple, a multiplicity of decoders is required. Furthermore, the numerous user rate buffers must accept 6X data at the burst rate. Such buffers are expensive to implement, as noted in Section 5.5.

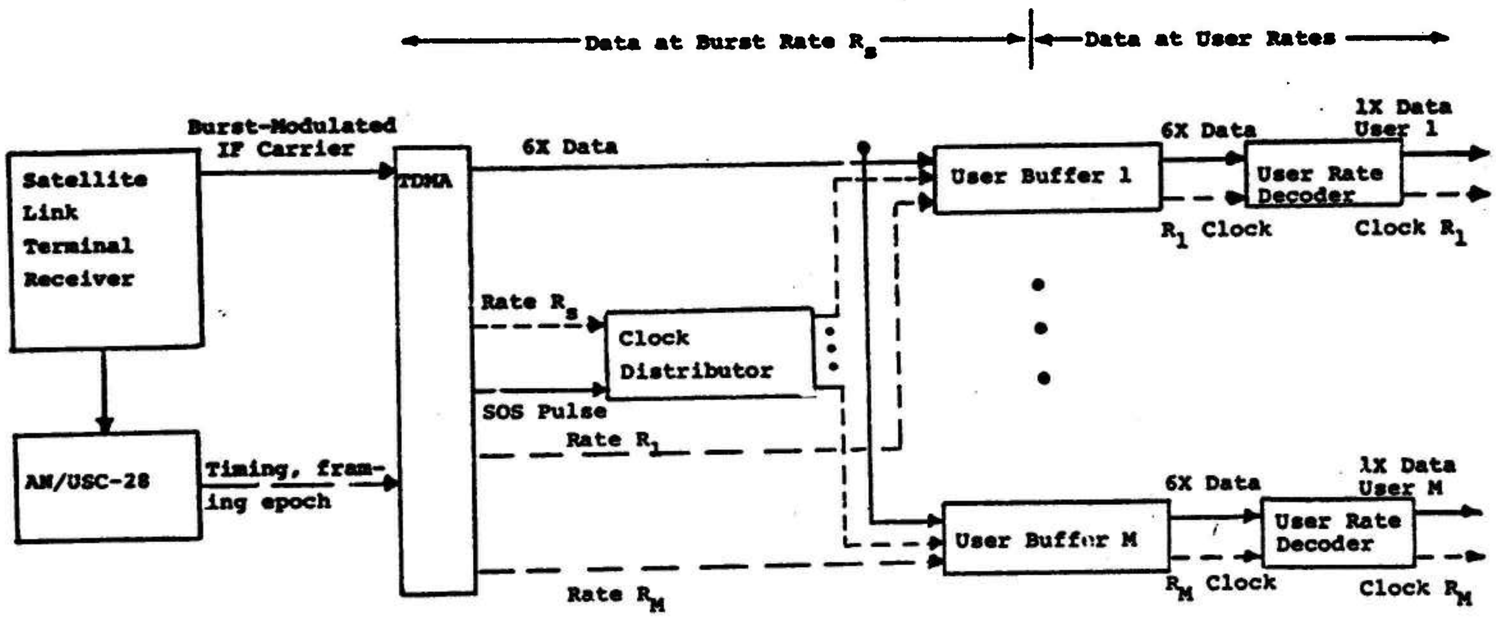


Fig. 5.3 - User-Rate Decoder Receiver Configuration

5.4 Burst Mode Sequential Decoding

The burst-rate Viterbi decoder in Fig. 5.1a might be replaced with a sequential decoder. A 100 mega-computation per second rate 1/2, hard decision sequential decoder, the LS4157, embodying a very sophisticated decoding algorithm, is being built by LINKABIT for NASA under Contract No. NAS2-6411, with delivery scheduled for February, 1972. This decoder could be used as a burst-rate decoder with minor modifications.

Although accepting data in high rate bursts, the sequential decoder achieves a system performance determined by the station aggregate rate. The integral buffer in the sequential decoder smooths the bursts so that the decoder need only keep up with the aggregate rate. The worst speed factor (ratio of computation rate to average information rate) for the networks considered in Section 5.5 is $100/13.3 = 7.5$ for receiving station 1 of network C. All other speed factors are 15 or greater.

The sequential decoder performance in terms of E_b/N_o is roughly equal to that of Viterbi decoding at an average error probability of 10^{-5} and is a few tenths of a dB better at 10^{-6}, assuming ideal channel conditions. Carrier-phase and bit-tracking loop errors cause significantly greater degradation in sequential as opposed to Viterbi decoding because of impact on computational variability. On the other hand, AGC fluctuations do not disturb the hard decision sequential decoder but might affect Viterbi decoding if sufficiently severe.

Because sequential decoding is being used at the burst rate and hence time-shared among users, tails are required. For the constraint length 41 sequential decoder, the tail can be compressed into 20 bits. For effective aggregate-rate decoding, a 20-bit tail must be appended to each subburst, significantly impacting frame efficiency. Tails might be added only to equal-rate segments, but the sequential decoder would then be required to decode all data arriving at the station burst rate, decreasing the speed factor and imposing up to a 0.5 dB degradation in performance. A reasonable compromise could be achieved by inserting a small number of tails in long constant-rate segments.

Another significant difficulty is introduced by the use of sequential decoding. Most decoding errors occur in long bursts occasioned by buffer overflows. Care in system design would be necessary to minimize the impact of the long bursts. In particular, bit integrity must be maintained despite the bursts. Of course, bursts need not extend beyond tailed-off subbursts.

One last factor should be noted. The sequential decoder provides different coding gains to different users.

If carrier-phase and bit-tracking errors are controlled, the value of E_b/N_o required for a given bit error probability would be smaller for low aggregate-rate receivers due to larger speed factors. Moreover, the short information blocks between tails causes a decrease in overflow probability, although decoder modifications would be necessary to fully utilize this tail information. The relative cost of sequential decoding, based on the cost of the LS4157 indicated in Table 3.1, is included in tabulations presented in Section 5.8.

5.5 Buffer Design

The system configurations of Figs. 5.1, 5.2, and 5.3 utilize buffers for rate conversion. For example, the aggregate-rate system which requires but one decoder per station does require a buffer to convert data from the burst rate R_s to the maximum decoder rate R_D, as well as additional user buffers to convert from rate R_D to the individual user rate R_1, R_2, ..., R_M. The conversion from rate R_s to rate R_D is potentially expensive, since it involves both high burst rates, up to 80 Mbps, and 6X data. Expensive buffering would tend to rule out the use of the aggregate-rate configuration.

Fortunately, inexpensive buffers can be designed, although the approach differs markedly from that recommended by IBM[3]. (It should be noted that IBM was not considering soft-quantized 6X data.) The present design is based on the use of dynamic MOS shift register technology, presently available in 16-pin dual-in-line ceramic packages from several vendors, including Texas Instrument and Intel. Two sizes in interesting price ranges are presently offered; a quad 80 bit shift register (TM 3409) costing $8 in quantity or 2.5¢ per bit, and a quad 256 bit shift register (TMS 3412) costing $12 in quantity or 1.2¢ per bit. Based on past experience, prices can be expected to decrease dramatically, since both chips are relatively new. Both size chips have a conservatively rated maximum shifting rate of 5 Mbps.

A buffer capable of 80 Mbps operation is shown in Fig. 5.4. Data is fed serially to a 16 bit, high-speed shift register. After 16 bits are loaded, data is transferred in parallel to the input of 16 MOS shift registers. These 16 registers then shift data right at 80/16 = 5 Mbps. (Burst rates of 40 Mbps can be accomodated with 8 parallel registers, etc.) At the output, a 16 bit shift register accepts data in parallel from the MOS register outputs and converts the data back to serial form. The design is straightforward. One potentially expensive element, the 16 stage high-speed input shift register can be replaced by two or four slower shift registers together with an 80 Mbps switch or a 4-stage, high-speed shift register, if cost effective.

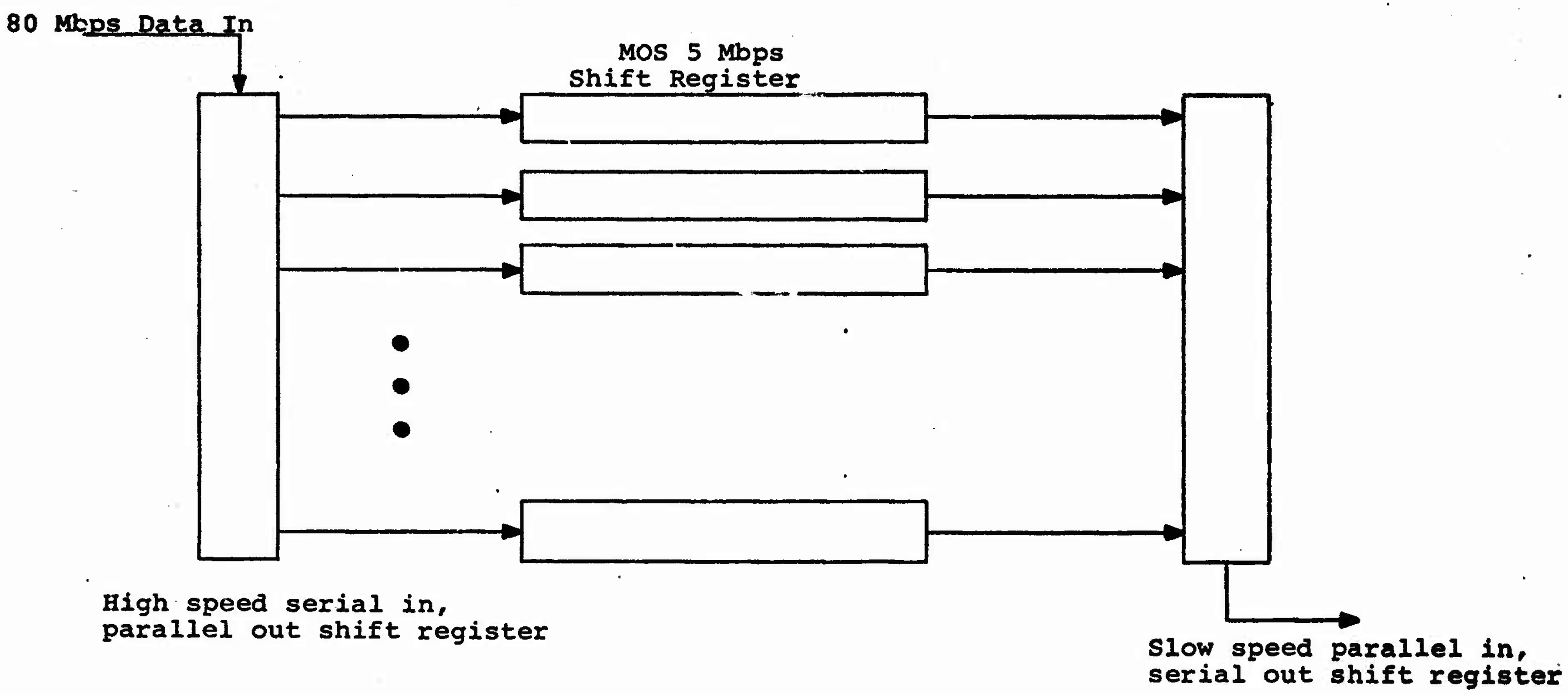


Fig. 5.4 Storage Buffer Configuration for Burst Rate to Decoder Rate Conversion.

Counters may be used to keep track of the number of bits shifted into a register. Alternatively, the shift register may be zeroed prior to shifting data in, except for an initial "1" in the first stage. The fully-loaded condition is detected when the first "1" is output from the register.

Since it is necessary to shift data into the burst buffer at one rate and shift data out of the buffer at a second rate, two storage buffers must be used alternately. Data is input to one of the buffers. When this buffer is filled, the input is switched to the second buffer, and the first buffer is emptied. It should be noted that two buffers do not necessarily double cost since the total capacity of the two buffers need be no greater than the total number of bits entered into the buffer in one frame, even if the input is one short burst and the output is periodic.

Control of the buffers is also straightforward. For the aggregate-rate decoder, the buffer full indicator, generated in the buffer by a leading "1" or a counter as noted above, causes the buffer to output through the decoder in a burst of rate R_D, continuing until the buffer is empty.

It is informative to consider the maximum continuous output data rate, R_{max}, which can be achieved by the minimum size buffer composed of either 80 bit or 128 bit shift registers. We assume a frame repetition rate of 1200 frames per second. At a burst rate of 80 Mbps, a minimum of 16 parallel 5 Mbps registers are required. If each register holds 80 bits, one storage buffer holds 16 x 80 = 1280 bits. Since a burst buffer is composed of two storage registers, the 80 Mbps burst buffer has a total storage capacity in the 80 bit MOS shift registers of 2 x 1280 = 2560 bits. A continuous output data rate can thus be guaranteed if each frame contains at most 2560 bits. At a frame repetition rate of 1200 frames per second, this corresponds to a maximum continuous rate out of the buffer of

$$R_{max} = 1200 \times 2560 = 3.072 \text{ Mbps.}$$

Any continous output rate less than R_{max} can also be accomodated. The value of R_{max} can be increased by using 256 bit shift registers and further increased by using 512 or 1024 bit shift registers, which are also available.

Achievable values of R_{max} for 80 bit and 256 bit shift registers are summarized in Table 5.1 along with the number of MOS DIP's required. The relative cost of the buffer with respect to the cost of an LV7026 decoder is also included for configuration evaluation purposes.

In the aggregate rate configuration, the control shift register must provide a delay equal to that of the burst buffer and the decoder. This control shift register is most

Burst Rate	No. of Shift Registers in Parallel	R_{max} (Mbps)		No. of MOS DIP's		Relative Cost		
		80 bit	256 bit	1X	6X	1X 80 bit	6X 80 bit	6X 256 bit
80 Mbps	16	3.072	9.8304	8	48	.1	.23	.25
40 Mbps	8	1.586	4.9152	4	24	.08	.15	.16
20 Mbps	4	.793	2.4576	2	12	.07	.12	.12
10 Mbps	2	.793 (1X)	1.2288 (6X)	2	6	.07	.1	.1
5 Mbps	1	.793 (1X)	.6144 (6X)	1	4	.05	.08	.08

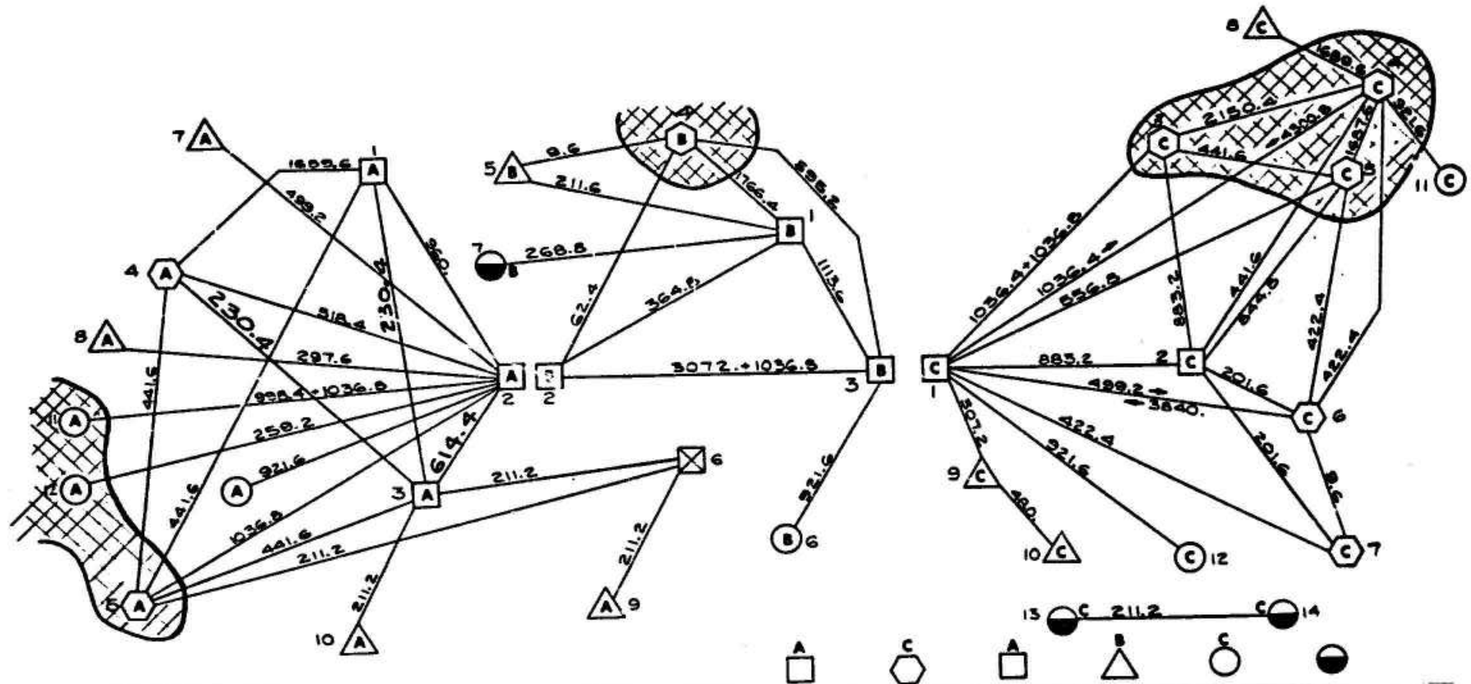
Table 5.1 - Maximum Continuous Rates and Relative Costs of Burst Buffers. (The LV7026 decoder is taken as relative cost 1.0.)

easily made integral with the aggregate rate burst buffer as indicated in Fig. 5.2a. Cost of the control shift register is included in the estimate of the 6X data buffer.

5.5 Network Configurations

Quantitative comparison of the three alternative decoder placements in an operating TDMA system depends significantly upon network configurations and user rates. For purposes of this study, four network configurations are used as models. Three, denoted A, B, and C, were provided by M. Jones of CSC and are reproduced as Fig. 5.5. These networks represent estimates of the future utilization of West Pacific, East Pacific, and Atlantic satellites respectively, assuming 1 satellite in each area. The fourth, D, is the model from Ref. 2.

A computer program was prepared by LINKABIT to permit rapid evaluation of data concerning these networks. Typical printouts for the four networks are given as Tables 5.2 - 5.5. Each table contains a matrix, the ij^{th} element of which contains 3 items of information pertaining to transmission from station i to station j. The top item is the user data rate in Kbps. The second item is the number of bits used per frame at 1200 frames per second. The third item is needed for buffer comparisions and is the minimum number of bits of storage required for each 5 Mbps dynamic



SATELLITE
A = W. Pac.
B = E. Pac.
C = Atlantic

TERMINALS	HT	MT	FSC-9	MSC-46	TSC-54	MASCOT
G/T(dB)	39	27	38	34	26.5	31
BW (MHz)	125	125	40	125	10	?
Burst Rate M Symbols/ Sec.	80	10	80	40	10	20

Fig. 5.5 - Typical Phase II Network

FROM/TO	1 HT EC	2 HT EC	3 HT EC	4 PT EC	5 PT NO	6 FSC EC	7 MSC EC	8 MSC EC	9 FSC EC	10 MSC EC	11 TSC NR	12 TSC NM	13 TSC FC
1 HT	0.0 0. 0.0	960.0 800. 50.0	230.4 192. 12.0	1689.6 1408. 70[illegible].0	441.6 368. 23.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0
2 HT	960.0 800. 50.0	0.0 0. 0.0	614.4 512. 32.0	518.4 431. 216.0	1036.8 863. 54.0	0.0 0. 0.0	499.2 416. 52.0	297.6 248. 31.0	0.0 0. 0.0	0.0 0. 0.0	2035.2 1696. 106.0	259.2 215. 13.5	921.6 768. 304.0
3 HT	230.4 192. 12.0	614.4 512. 32.0	0.0 0. 0.0	230.4 192. 96.0	441.6 368. 23.0	211.2 176. 11.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	211.2 176. 22.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0
4 PT	1689.6 1408. 88.0	518.4 431. 27.0	230.4 192. 12.0	0.0 0. 0.0	441.6 368. 23.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0
5 PT	441.6 368. 23.0	1036.8 863. 54.0	441.6 368. 23.0	441.6 368. 184.0	0.0 0. 0.0	211.2 176. 11.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0
6 FSC	0.0 0. 0.0	0.0 0. 0.0	211.2 176. 11.0	0.0 0. 0.0	211.2 176. 11.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	211.2 176. 22.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0
7 MSC	0.0 0. 0.0	499.2 416. 26.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0
8 MSC	0.0 0. 0.0	297.6 240. 15.5	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0
9 FSC	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	211.2 176. 11.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0
10 MSC	0.0 0. 0.0	0.0 0. 0.0	211.2 176. 11.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0
11 TSC	0.0 0. 0.0	2035.2 1696. 106.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0
12 TSC	0.0 0. 0.0	259.2 215. 13.5	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0
13 TSC	0.0 0. 0.0	921.6 768. 48.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0
TOTAL P/1.2 /BR/5	3321.6 2768. 173.0	7142.4 5949. 372.0	1939.2 1616. 101.0	2880.0 2399. 1200.0	2572.8 2143. 134.0	633.6 528. 33.0	499.2 416. 52.0	297.6 248. 31.0	211.2 176. 22.0	211.2 176. 22.0	2035.2 1696. 106.0	259.2 215. 13.5	921.6 768. 304.0

BTU TOTAL FOR THIS NETWORK IS 44764.

PREAMBLE LENGTH IS 160 BTU
FOR TAIL LENGTH OF 0 BITS, AND TOTAL TAIL LENGTH OF 0 BTU, NETWORK EFFICIENCY IS 95.81 PERCENT
FOR TAIL LENGTH OF 6 BITS, AND TOTAL TAIL LENGTH OF 462 BTU, NETWORK EFFICIENCY IS 94.82 PERCENT
FOR TAIL LENGTH OF 20 BITS, AND TOTAL TAIL LENGTH OF 1540 BTU, NETWORK EFFICIENCY IS 92.11 PERCENT

FOR TOTAL TAIL LENGTH OF 806 BTU, VITERBI BURST-RATE EFFICIENCY FOR THIS NETWORK IS 94.65 PERCENT

Table 5.2 - Network A Details

FROM/TO	1 FT EC	2 HT EC	3 HT EC	4 PT MP	5 PSC EC	6 TSC EC	7 MAS EC
1 FT	0.0	364.8	1113.6	1766.4	211.2	0.0	268.8
	0.	304.	928.	1472.	176.	0.	224.
	0.0	19.0	58.0	92.0	22.0	0.0	56.0
2 HT	364.8	0.0	4108.8	62.4	0.0	0.0	0.0
	304.	0.	3424.	52.	0.	0.	0.
	19.0	0.0	214.0	3.2	0.0	0.0	0.0
3 HT	1113.6	4108.8	0.0	595.2	0.0	921.0	0.0
	928.	3424.	0.	456.	0.	767.	0.
	58.0	214.0	0.0	31.0	0.0	383.7	0.0
4 PT	1766.4	62.4	595.2	0.0	9.6	0.0	0.0
	1472.	52.	456.	0.	8.	0.	0.
	92.0	3.2	31.0	0.0	1.0	0.0	0.0
5 PSC	211.2	0.0	0.0	9.6	0.0	0.0	0.0
	176.	0.	0.	8.	0.	0.	0.
	11.0	0.0	0.0	[illegible]	0.0	0.0	0.0
6 TSC	0.0	0.0	921.0	0.0	0.0	0.0	0.0
	0.	0.	767.	0.	0.	0.	0.
	0.0	0.0	47.9	0.0	0.0	0.0	0.0
7 MAS	268.8	0.0	0.0	0.0	0.0	0.0	0.0
	224.	0.	0.	0.	0.	0.	0.
	14.0	0.0	0.0	0.0	0.0	0.0	0.0
TOTAL	2724.3	4536.0	6738.6	2437.6	220.8	921.0	268.8
B/1.2	2104.	3780.	5615.	2028.	184.	767.	224.
/BR/5	194.9	236.3	351.0	[illegible]	23.0	383.8	56.0

BTU TOTAL FOR THIS NETWORK IS 23047.
PREAMBLE LENGTH IS 160 BTU
FOR TAIL LENGTH OF 0 BITS, AND TOTAL TAIL LENGTH OF 0 BTU, NETWORK EFFICIENCY IS 95.14 PERCENT
FOR TAIL LENGTH OF 6 BITS, AND TOTAL TAIL LENGTH OF 192 BTU, NETWORK EFFICIENCY IS 94.45 PERCENT
FOR TAIL LENGTH OF 20 BITS, AND TOTAL TAIL LENGTH OF 640 BTU, NETWORK EFFICIENCY IS 92.57 PERCENT

FOR TOTAL TAIL LENGTH OF 138 BTU, VITERBI BURST-RATE EFFICIENCY FOR THIS NETWORK IS 94.57 PERCENT

Table 5.3 - Network B Details

FROM/TO	1 MT EC	2 MT EE	3 MT MB	4 MT MB	5 MT MB	6 MT EC	7 MT EE	8 MSC EC	9 MSC EC	10 MSC EC	11 TSC EC	12 TSC EC	12 MOS EE	14 MOS EE
1 MT	0.0	883.2	2073.2	1036.8	556.8	499.2	422.4	0.0	307.2	0.0	0.0	921.6	0.0	0.0
	0.	736.	1727.	864.	464.	416.	352.	0.	256.	0.	0.	768.	0.	0.
	0.0	46.0	107.9	53.9	29.0	208.0	176.0	0.0	32.0	0.0	0.0	384.0	0.0	0.0
2 MT	883.2	0.0	883.2	441.6	844.8	201.6	201.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	736.	0.	736.	368.	704.	168.	168.	0.	0.	0.	0.	0.	0.	0.
	46.0	0.0	46.0	23.0	44.0	84.0	84.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3 MT	2072.2	883.2	0.0	2150.4	441.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	1727.	736.	0.	1792.	368.	0.	0.	0.	0.	0.	0.	0.	0.	0.
	107.9	46.0	0.0	112.0	23.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4 MT	4300.8	441.6	2150.4	0.0	1689.6	422.4	0.0	1689.6	0.0	0.0	921.6	0.0	0.0	0.0
	3584.	368.	1792.	0.	1408.	352.	0.	1408.	0.	0.	768.	0.	0.	0.
	224.0	23.0	112.0	0.0	88.0	176.0	0.0	176.0	0.0	0.0	384.0	0.0	0.0	0.0
5 MT	556.8	844.8	441.6	1689.6	0.0	422.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	464.	704.	368.	1408.	0.	352.	0.	0.	0.	0.	0.	0.	0.	0.
	29.0	44.0	23.0	88.0	0.0	176.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6 MT	3840.0	201.6	0.0	422.4	422.4	0.0	9.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	3200.	168.	0.	352.	352.	0.	8.	0.	0.	0.	0.	0.	0.	0.
	200.0	10.5	0.0	22.0	22.0	0.0	4.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7 MT	422.4	201.6	0.0	0.0	0.0	9.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	352.	168.	0.	0.	0.	8.	0.	0.	0.	0.	0.	0.	0.	0.
	22.0	10.5	0.0	0.0	0.0	4.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8 MSC	0.0	0.0	0.0	1689.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.	0.	0.	1408.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
	0.0	0.0	0.0	88.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9 MSC	307.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	480.0	0.0	0.0	0.0	0.0
	256.	0.	0.	0.	0.	0.	0.	0.	0.	400.	0.	0.	0.	0.
	16.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	50.0	0.0	0.0	0.0	0.0
10 MSC	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	480.0	0.0	0.0	0.0	0.0	0.0
	0.	0.	0.	0.	0.	0.	0.	0.	400.	0.	0.	0.	0.	0.
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	50.0	0.0	0.0	0.0	0.0	0.0
11 TSC	0.0	0.0	0.0	921.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.	0.	0.	768.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
	0.0	0.0	0.0	48.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12 TSC	921.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	768.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
	48.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
13 MOS	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	211.2
	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	176.
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	44.0
14 MOS	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	211.2	0.0
	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	176.	0.
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	44.0	0.0
TOTAL	13305.2	3456.0	5548.4	8351.6	3955.2	1555.2	631.0	1689.0	787.2	480.0	921.6	921.6	211.2	211.0
B/1.2	11087.	2880.	4623.	6959.	3296.	1296.	528.	1408.	656.	400.	768.	768.	176.	176.
/BR/S	693.0	180.0	289.0	435.0	206.0	440.0	264.0	176.0	82.0	50.0	384.0	384.0	44.0	44.0

BTL TOTAL FOR THIS NETWORK IS 64802.
PREAMBLE LENGTH IS 100 BTU
FOR TAIL LENGTH OF 0 BITS, AND TOTAL TAIL LENGTH OF 0 BTU, NETWORK EFFICIENCY IS 96.91 PERCENT
FOR TAIL LENGTH OF 6 BITS, AND TOTAL TAIL LENGTH OF 720 BTU, NETWORK EFFICIENCY IS 95.89 PERCENT
FOR TAIL LENGTH OF 20 BITS, AND TOTAL TAIL LENGTH OF 2520 BTU, NETWORK EFFICIENCY IS 92.07 PERCENT

FOR TOTAL TAIL LENGTH OF 450 BTU, VITERBI BURST-RATE EFFICIENCY FOR THIS NETWORK IS 95.84 PERCENT

Table 5.4 - Network C Details

FROM/TO	12 MSC EC	21 MT EC	34 PT EC	44 MT EC	53 PSC EC	64 PT EC	73 MSC EC	85 PT EC	94 PT EC	5 MT EC	15 MT EC
12 PSC	0.0 0. 0.0	393.6 328. 20.5	201.6 160. 64.0	201.6 168. 84.0	201.6 168. 21.0	201.6 160. 84.0	201.6 160. 21.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0
21 PT	393.6 328. 41.0	0.0 0. 0.0	460.8 384. 152.0	259.2 216. 108.0	316.0 264. 33.0	268.8 224. 112.0	508.8 424. 53.0	201.6 168. 84.0	201.6 168. 84.0	76.0 64. 32.0	249.6 208. 104.0
34 PT	201.6 168. 21.0	316.8 264. 16.5	0.0 0. 0.0	9.6 8. 4.0	9.6 0. 1.0	9.6 8. 4.0	201.6 160. 21.0	0.0 0. 0.0	9.6 0. 4.0	0.0 0. 0.0	0.0 0. 0.0
44 PT	201.6 168. 21.0	253.2 211. 13.1	9.6 8. 4.0	0.0 0. 0.0	201.6 160. 21.0	201.6 168. 84.0	201.6 160. 21.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0
53 PSC	201.6 168. 21.0	316.8 264. 16.5	9.6 8. 4.0	4201.6 3501. 1750.6	0.0 0. 0.0	9.6 8. 4.0	201.6 168. 21.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0
64 PT	201.6 168. 21.0	316.8 264. 16.5	9.6 8. 4.0	201.6 168. 84.0	9.6 8. 1.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0
73 PSC	201.6 168. 21.0	508.8 424. 26.5	211.6 160. 64.0	201.6 168. 84.0	201.6 160. 21.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0
85 PT	0.0 0. 0.0	2200.8 1840. 115.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0
94 PT	0.0 0. 0.0	201.6 160. 10.5	9.6 8. 4.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0
5 PT	0.0 0. 0.0	76.8 64. 4.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0
15 PT	0.0 0. 0.0	249.6 208. 13.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0	0.0 0. 0.0
TOTAL B/1.2 /BR/5	1401.6 1168. 146.0	4842.0 4035. 252.2	902.4 752. 376.0	5875.2 4896. 2114.7	940.8 784. 98.0	691.2 576. 288.0	1315.2 1096. 137.0	201.6 168. 84.0	211.2 176. 88.0	76.8 64. 32.0	249.6 208. 104.0

BIT TOTAL FOR THIS NETWORK IS 41276.
PREAMBLE LENGTH IS 160 BTU
FOR TAIL LENGTH OF 0 BITS, AND TOTAL TAIL LENGTH OF 0 BTU, NETWORK EFFICIENCY IS 97.12 PERCENT
FOR TAIL LENGTH OF 6 BITS, AND TOTAL TAIL LENGTH OF 1560 BTU, NETWORK EFFICIENCY IS 95.00 PERCENT
FOR TAIL LENGTH OF 20 BITS, AND TOTAL TAIL LENGTH OF 4560 BTU, NETWORK EFFICIENCY IS 90.39 PERCENT

FOR TOTAL TAIL LENGTH OF 828 BTU, VITERBI BURST-RATE EFFICIENCY FOR THIS NETWORK IS 96.89 PERCENT

Table 5.5 - Network D Details

MOS shift register element, when sufficient MOS shift registers are paralleled to provide a buffer capable of accepting data at the station burst rate. Each matrix column is summed with totals appearing at the bottom. Since a column sum represents an accumulation over all users directed to a given station, the column totals may be identified with station aggregate rates.

The duration of one QPSK symbol at the 80 Mbps maximum burst rate is referred to as a basic time unit, denoted BTU. (Only 1 information bit is conveyed by each QPSK symbol since a rate 1/2 code with 2 check bits per information bit is assumed.) Lower burst rates are achieved by lengthening QPSK symbols, always using integer multiples of a BTU. Thus, a QPSK symbol at a burst rate of 20 Mbps occupies 4 BTU's. The exact number of BTU's per frame at 1200 frames per second is

$$75 \times 2^{20}/(75 \times 2^{4}) = 2^{16} = 65{,}536 \text{ BTU's/frame.}$$

The total BTU's used by a network is given below the matrix. Notice that Network C is very nearly used to capacity.

5.7 Frame Efficiency

Frame efficiency, defined as the ratio of the number N_I of BTU's per frame carrying information to the sum of

N_I plus the number of BTU's per frame devoted to preambles and tails is an important system parameter. Efficiency is highly dependent on the length and number of tails per frame. Table 5.6 tabulates frame efficiencies assuming a preamble (including guard time) of 160 BTU's preceding each transmission burst. The numbers are taken from Tables 5.2-5.5 Note that all configurations using Viterbi decoding, including aggregate-rate decoding, have frame efficiencies greater than 94.3% and hence are acceptable. Burst-rate decoding with sequential decoding, which requires a 20 bit tail with every subburst, has efficiencies below 93% for each of the networks, and is marginal from the viewpoint of frame efficiency.

5.8 Decoder and Buffer Costs for Various Configurations

The relative cost of outfitting all stations in Networks A, B, and C of Fig. 5.5 with decoders and spares can be calculated using the buffer costs of Table 5.1 and the decoder costs of Table 3.1. A spares policy is adopted of including one spare decoder and one spare buffer, each of the highest required speed, with each station. Results are tabulated in Table 5.7.

Note first that the cost of burst-rate decoding with either Viterbi or sequential decoding is considerably greater than aggregate or user-rate decoding with Viterbi decoders.

CONFIGURATION	NETWORK			
	A	B	C	D
User-Rate	95.3	95.1	96.5	97.1
Aggregate-Rate	94.3	94.4	95.4	95.0
Burst-Rate (Viterbi)	94.7	94.6	95.8	96.3
Burst-Rate (Sequential)	92.1	92.6	92.3	90.4

Table 5.6 - Summary of Frame Efficiencies (%).

Class	System Configuration	Number of Decoders (All K=7 Viterbi, except * indicates sequential)														
		Speed: / Cost:	0.2 / .65	2 / 1.0	4.2 / 1.5	7 / 1.7	8.5 / 2.0	14 / 2.5	30 / 5.0	40 / 9.0	80 / 20.0	80* / 3.5	Total Decoders	Decoder Cost	Buffer Cost	TOTAL COST
1.	Burst Rate-Viterbi							14	6	16	32		68	850	13	863
2.	Burst Rate-Sequential											68	68	238	13	251
3.	Aggregate Rate		10	32	14	8	2	2					68	82	22	104
4.	Aggregate Rate			42		22		4					68	89	22	111
5.	Aggregate Rate			42		28							70	90	23	113
6.	Aggregate Rate					70							70	119	23	142
7.	User Rate		22	99	17								138	139	27	166
8.	User Rate			121	17								138	147	27	174

Table 5.7 - Relative Cost of Decoders and Buffers for Several System Configurations and Decoder Speed Mixes.

Notes:
1) One spare decoder and one spare buffer of the maximum required speed is included per station.
2) Decoder costs from Table 3.1.
3) Buffer costs from Table 5.1. Aggregate rate buffers use 6X, 256 bit buffers. All user buffers are 80 bit buffers of appropriate burst rate and size (1X for burst rate; 6X for user rate).
4) Mixes 5 and 6 assume 7 Mbps decoders are paralleled in two stations.

Furthermore, development and maintenance costs are unreasonably increased when a large variety of decoders are included in the system. It appears that only classes 5, 6, and 8 deserve serious consideration, since each utilizes at most two different speeds of decoders without incurring significant cost penalties.

Both classes 5 and 6 use two 7 Mbps decoders in parallel to handle two high-aggregate-rate stations, one of 13.3 Mbps and one of 8.35 Mbps, both in Network C. The paralleling of decoders in this case does not necessitate including additional tails, as would be the case in paralleling at the burst rate (see Section 3.2), since complete subbursts can be decoded in each decoder. A pair of aggregate buffers is required, however, both to buffer different subbursts with minimal control requirements as well as to achieve the required data rates; the maximum buffer speed is seen to be 9.83 Mbps in Table 5.1. The additional buffers are included in the cost estimates of classes 5 and 6 in Table 5.7.

REFERENCES FOR SECTION 5

1. R.K. Townley and R.C. Davis, "TDM/TDMA Synchronization", Eastcon 70, pgs. 260-265.

2. G.J. Goubeaud and J.E. McGovney, "Time Division Multiple Access Computer Simulation Report," USASATCOMA, 1 May 1970.

3. IBM Buffer Study, Expansion and Compression Buffers for the Phase II DSCS TDMA Final Report, February, 1971, Federal Systems Division, International Business Machines Corporation, Gaithersburg, Maryland 20760

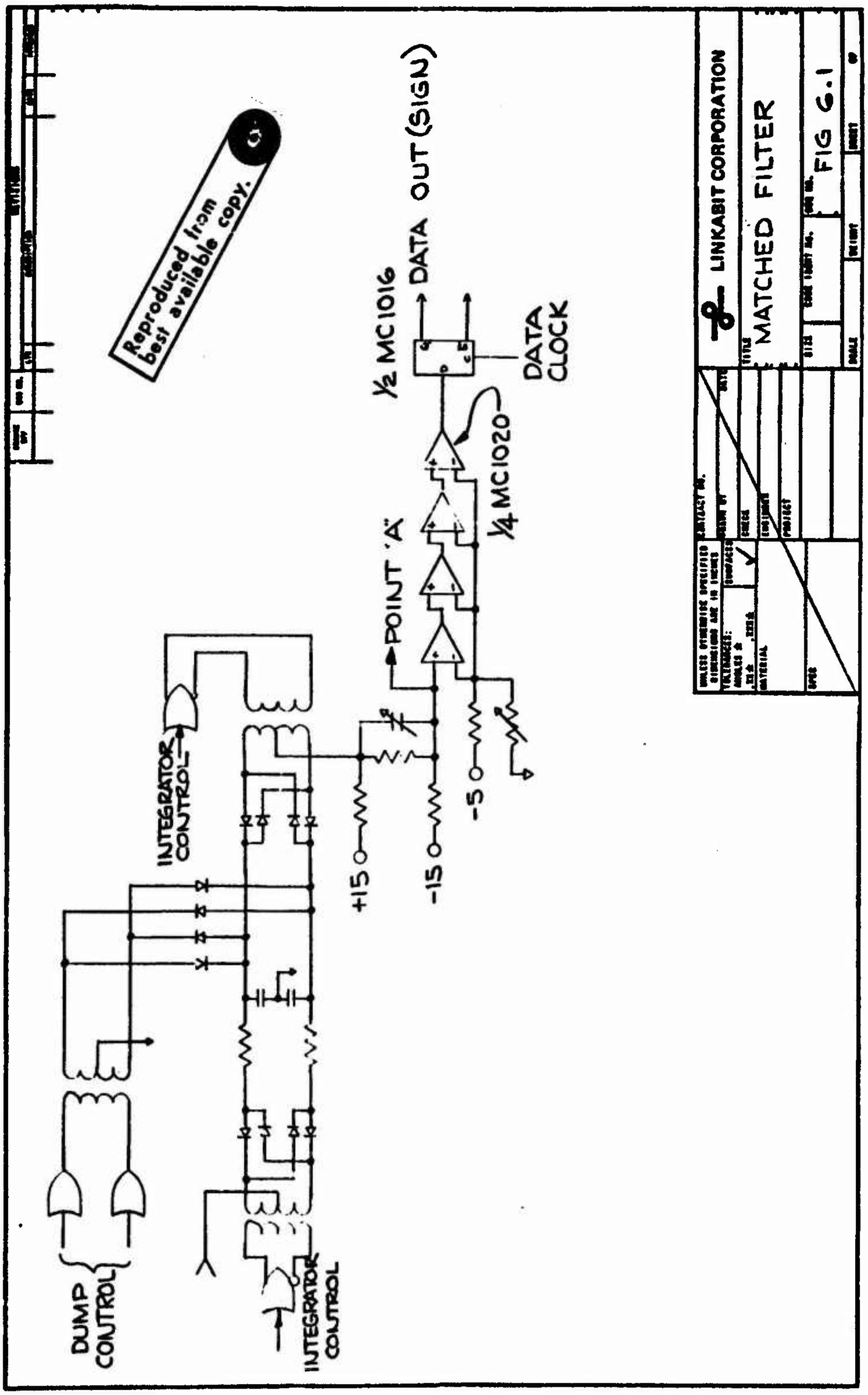
6.0 SOFT DECISIONS FOR QUADRIPHASE MODEM

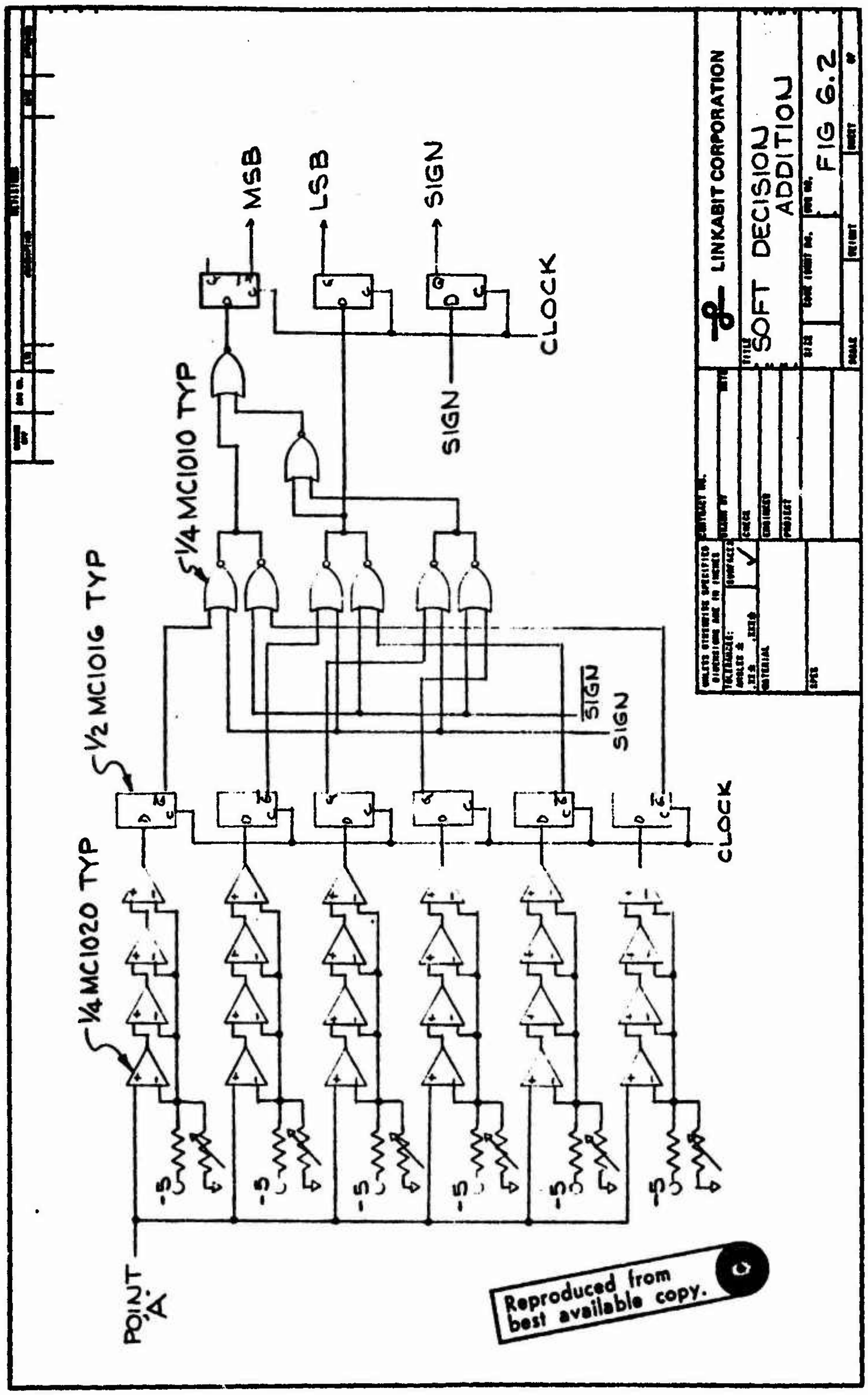
This section deals with the addition of soft decisions to an existing quadriphase modem. The modem under consideration is the Philco Digi-Phase IV Quadriphase Modem. The Digi-Phase IV variable data rate quadriphase modem is capable of operating at 2 1/2, 5, 10, 20, and 40 megabits per second rate. Bit detection is performed by a matched filter followed by a hard decision circuit. The matched filter card for this modem contains two matched filters and decision elements, one for each of the two channels of the modem. The signal input to the filter is plus or minus .1 volts peak, centered around ground, for signal in the absence of noise.

The matched filter consists of a pair of RC integrators which operate on alternate bits. During one bit time, one of the integrators integrates the input signal while the other one is being discharged. On the next bit time, the integrator that was previously discharged integrates the input signal while the other integrator is being discharged. The input is switched between the two integrators by a diode-transformer multiplexer. The outputs of the two integrators are recombined in a similar diode-transformer multiplexer. The output of the matched filter is then fed to a decision element made from a MECL II quad line receiver, the MC 1020. The four line receivers are cascaded to provide sufficient gain. The MC 1020 has a differential input. The output from the matched filters is fed into one input and a reference voltage that is established by a resistor voltage divider is fed into the other input. This reference voltage sets the threshold point for the hard decision operation. The output of the MC 1020 which is a MECL II logic level signal, is clocked into an MC 1016 latch. A simplified schematic of the matched filter is shown in Fig. 6.1.

The simpliest way to add soft decisions to this matched filter would be to add six additional MC 1020 differential line receivers to each matched filter. The output of the integrator would drive all seven of the differential line receivers. Each of the differential line receivers would be provided with its own reference voltage divider which would establish the seven decision boundries required for soft-decisions. The seven outputs from the MC 1020's would then be clocked into seven MC 1016 latches. The output of the latches would then be encoded into a three bit sign-magnitude representation. The result could then be buffered and passed on to the external decoder. A simplified schematic of the addition to the matched filter modified for soft decisions is shown in Fig. 6.2.

This modification would require the addition of approximately 22 MECL II IC's to the modem. It appears that there is insufficient room on the existing match


Reproduced from best available copy.
DATA OUT (SIGN)
½ MC1016
DATA CLOCK
¼ MC1020
POINT 'A'
+15
-15
-5
INTEGRATOR CONTROL
DUMP CONTROL
INTEGRATOR CONTROL
LINKABIT CORPORATION
MATCHED FILTER
FIG 6.1


POINT 'A'
-5
1/4 MC1020 TYP
1/2 MC1016 TYP
1/4 MC1010 TYP
SIGN
SIGN
CLOCK
CLOCK
MSB
LSB
SIGN
SIGN
LINKABIT CORPORATION
SOFT DECISION ADDITION
FIG 6.2

filter card to add an additional 22 IC's. However, there appears to be plenty of space above the matched filter card. The easiest way to add the additional cicruits required is to mount them on an additional circuit board and mount this circuit board above the existing matched filter card using standoffs.

A rough budgetary estimate on the cost of modifying and testing a pair of modems would be $15,000 to $20,000.

7.0 DECODER RELIABILITY PREDICTIONS

Tables 7.1 and 7.2 give the predicted mean-time-between-failures for both the 1.8 MBPS and 7.0 MBPS LINKABIT Viterbi decoders. Data for these tables was taken from MIL-HDBK-217A (paragraphs 5, 6, 7). It should be noted that while these tables show all integrated circuits used to have an expected 40 failures per 10^8 hours of use, data from the principal manufacturers of integrated circuits indicates that a failure rate of 10 per 10^8 hours is justified. Were such a figure to be used the MTBF for both the LV7026 and the 7 MBPS version would be well above 13,000 hours.

LINKABIT LV7026 ENCODER/DECODER

MEAN-TIME-BETWEEN-FAILURE-PREDICTION

Equipment Ambient = 25°C
Part Ambient = 45°C

COMPONENT PART NUMBER	MANUFACTURER	NUMBER USED (N)	FAILURE PER 10^8 HOURS (λ)	K-FACTOR (K_F)	TOTAL ($N\lambda$)	TOTAL ($N\lambda K_F$)
			SOURCE: MIL-HDBK-217A			
7400	Texas Instruments	5	40	1.0	200	200
7402	Texas Instruments	8	40	1.0	320	320
7404	Texas Instruments	13	40	1.0	520	520
7408	Texas Instruments	1	40	1.0	40	40
7487	Texas Instruments	15	40	1.0	600	600
7430	Texas Instruments	2	40	1.0	80	80
7440	Texas Instruments	1	40	1.0	40	40
7474	Texas Instruments	3	40	1.0	120	120
7486	Texas Instruments	7	40	1.0	280	280
7489	Texas Instruments	64	40	1.0	2560	2560
7495	Texas Instruments	92	40	1.0	3680	3680
74107	Texas Instruments	1	40	1.0	40	40
74155	Texas Instruments	2	40	1.0	80	80
74163	Texas Instruments	2	40	1.0	80	80
74166	Texas Instruments	2	40	1.0	80	80
74150	Texas Instruments	4	40	1.0	160	160

Page 1 of 3

TABLE 7.1 LINKABIT LV7026 Encoder/Decoder
Mean-Time-Between-Failure Prediction

LINKABIT LV7026 ENCODER/DECODER

MEAN-TIME-BETWEEN-FAILURE PREDICTION

Equipment Ambient = 25°C
Part Ambient = 45°C

COMPONENT PART NUMBER	MANUFACTURER	NUMBER USED (N)	FAILURE PER 10^8 HOURS (λ)	K-FACTOR (K_F)	TOTAL	
			SOURCE: MIL-HDBK-217A		($N\lambda$)	($N\lambda K_F$)
74164	Texas Instruments	1	40	1.0	40	40
74180	Texas Instruments	2	40	1.0	80	80
74181	Texas Instruments	48	40	1.0	1920	1920
74193	Texas Instruments	8	40	1.0	320	320
74198	Texas Instruments	2	40	1.0	80	80
7407	Texas Instruments	2	40	1.0	80	80
74H11	Texas Instruments	2	40	1.0	80	80
74H30	Texas Instruments	1	40	1.0	40	40
74H55	Texas Instruments	16	40	1.0	640	640
74H74	Texas Instruments	1	40	1.0	40	40
74H103	Texas Instruments	1	40	1.0	40	40
74500	Texas Instruments	16	40	1.0	640	640
9318	Fairchild	2	40	1.0	80	80
9322	Fairchild	20	40	1.0	800	800
9602	Fairchild	1	40	1.0	40	40
9615	Fairchild	6	40	1.0	240	240

Page 2 of 3

TABLE 7.1 continued

LINKABIT LV7026 ENCODER/DECODER

MEAN-TIME-BETWEEN-FAILURE PREDICTION

Equipment Ambient = 25°C
Part Ambient = 45°C

COMPONENT PART NUMBER	MANUFACTURER	NUMBER USED (N)	FAILURE PER 10^8 HOURS (λ)	K-FACTOR (K_F)	TOTAL ($N\lambda$)	TOTAL ($N\lambda K_F$)
			SOURCE: MIL-HDBK-217A			
HROM-0542	Harris Semiconductor	2	40	1.0	80	80
899-1-R470	Beckman	9	15	.03	135	4
¼ Watt Resistor	Allen-Bradley	26	.35	6	9.1	54
½ W. Resistor	Allen-Bradley	12	.35	6	4.5	27
.01 μf cap.	Centralab	150	.2	1.0	30	30
2.2 μf cap.	Centralab	40	.2	1.0	8	8
LV-E-5-0V	Lambda Power Supply	1	1100	1.0	1100	1100
LCD-A-H	Lambda Power Supply	1	1100	1.0	1100	1100
Miscellaneous Connections		7483	.19	1.0	1422	1422
					TOTAL	17865
					MTBF	5598

Page 3 of 3

Table 7.1 continued

7 MBPS K=7 VITERBI DECODER

MEAN-TIME-BETWEEN-FAILURE PREDICTION

Equipment Ambient = 25°C
Part Ambient = 45°C

COMPONENT PART NUMBER	MANUFACTURER	NUMBER USED (N)	FAILURE PER 10^8 HOURS (λ)	K-FACTOR (K_F)	TOTAL	
			SOURCE: MIL-HDBK-217A		($N\lambda$)	($N\lambda K_F$)
7400	Texas Instruments	2	40	1.0	80	80
7404	Texas Instruments	4	40	1.0	160	160
7408	Texas Instruments	1	40	1.0	40	40
7409	Texas Instruments	2	40	1.0	80	80
7430	Texas Instruments	2	40	1.0	80	80
7437	Texas Instruments	6	40	1.0	240	240
7486	Texas Instruments	4	40	1.0	160	160
7489	Texas Instruments	64	40	1.0	2560	2560
74107	Texas Instruments	3	40	1.0	120	120
74150	Texas Instruments	4	40	1.0	160	160
74156	Texas Instruments	5	40	1.0	200	200
74175	Texas Instruments	7	40	1.0	280	280
74193	Texas Instruments	9	40	1.0	360	360

Page 1 of 3

TABLE 7.2 7 MBPS K=7 Viterbi Decoder
Mean-Time-Between-Failure Prediction

7 MBPS K=7 VITERBI DECODER

MEAN-TIME-BETWEEN-FAILURE PREDICTION

Equipment Ambient = 2
Part Ambient = 4

COMPONENT PART NUMBER	MANUFACTURER	NUMBER USED (N)	FAILURE PER 10^8 HOURS (λ)	K-FACTOR (K_F)	TOTAL ($N\lambda$)	TOTAL ($N\lambda K_F$)
			SOURCE: MIL-HDBK-217A			
74180	Texas Instruments	2	40	1.0	80	80
9300	Fairchild	30	40	1.0	1200	1200
9309	Fairchild	14	40	1.0	560	560
9318	Fairchild	2	40	1.0	80	80
9322	Fairchild	6	40	1.0	240	240
HROM-05-12	Harris Semiconductor	2	40	1.0	80	80
MC10102	Motorola	44	40	1.0	1760	1760
MC10105	Motorola	44	40	1.0	1760	1760
MC10141	Motorola	64	40	1.0	2560	2560
MC10181	Motorola	48	40	1.0	1920	1920
MC1268	Motorola	8	40	1.0	320	320
Resistors (¼W)	Allen-Bradley	10	.35	6	3.5	21
.01μf cap.	Centralab	150	.2	1.0	30	30
2.2μf cap.	Centralab	40	.2	1.0	8	8

TABLE 7.2 continued

7 MBPS K=7 VITERBI DECODER

MEAN-TIME-BETWEEN-FAILURE PREDICTION

Equipment Ambient = 25°C
Part Ambient = 45°C

COMPONENT PART NUMBER	MANUFACTURER	NUMBER USED (N)	FAILURE PER 10^8 HOURS (λ)	K-FACTOR (K_F)	TOTAL ($N\lambda$)	TOTAL ($N\lambda K_F$)
			SOURCE: MIL-HDBK-217A			
Power Supply (LV-E-5-OV)	Lambda	1	1100	1.0	1100	1100
Power Supply (LCD-A-11)	Lambda	1	1100	1.0	1100	1100
Connections		7400	.19	1.0	1406	1406
					TOTAL	19065
					MTBF	5245

Page 3 of 3

TABLE 7.2 continued

8.0 CONCLUSIONS

The DCS Stage II TDMA satellite communication system should use constraint length 7, soft decision, rate 1/2 Viterbi decoding in an aggregate-rate configuration to achieve a 5 dB advantage over uncoded QPSK. Based on the results of Section 5.0, class 5 aggregate rate systems utilizing both 2 and 7 Mbps decoders and MOS shift register buffers is recommended. Total decoder and buffer cost, together with controls, appear quite moderate when compared to the modem costs. Assuming 2 Mbps decoders are procured for use in the FDMA stage Ib system, the aggregate rate approach permits a transition to Stage II without necessitating any changes in the decoders.

Use of aggregate rate decoding as opposed to user rate decoding (Classes 5 and 8 respectively of Table 5.7) has two significant advantages. First, the total system cost of 113 (in relative units) for class 5 is 35% less than the cost of 174 for class 8. Secondly, since only one decoder is required per station, the spare decoder provides a fully redundant capability. A simple circuit can be used to monitor the system performance indicator and to automatically switch output to the spare whenever degraded performance is observed. This capability would be more difficult to realize in the user-rate configuration, since only one spare is available to replace several decoders.

Both systems require two different speed decoders. It is recommended that the second decoder be the 7 Mbps decoder, both because it provides increased capability over the 4.2 Mbps decoder and, more importantly, since it requires fewer parts (see Table 3.1) and hence has greater potential reliability.

The aggregate rate buffer and control circuits described in Section 5 appear quite elegant in their simplicity. The control shift register is easily incorporated in the aggregate rate buffer and provides a simple but flexible means of adapting to different subbursts. The TDMA need only provide a start of subburst indicator at the beginning of each subburst intended for the station; no additional bookkeeping is required. The clock distributor and user buffers of Fig. 5.2 appear equally straightforward. An engineering model of the entire receiver could be undertaken at the present time with low risk using this approach.

The connection between the user buffers and the ICF was not a subject of this study. The possibility should be examined, however, of dispensing with the user buffers and transmitting user data over the ICF in bursts, with appropriate markers. If the ICF is capable of accepting data at rate R_D, say 2 Mbps for the slower decoders, the decoder output could be transmitted directly to the TDM multiplexer at the TCF. It is not known what, if any, modifications in the AN/GSC-24 would be required to accept data in bursts; the modifications might be sufficiently complex to render the scheme impractical. The cost of user buffers is not sufficiently great to justify large efforts unless the entire system is simplified.

It appears that the gated carrier tracking loops will provide sufficient reference signal-to-noise ratios with the suggested .001 duty cycle to achieve less than 1 db degradation due to incoherence losses, for burst rates above 10 Mbps. For lower burst rates it is recommended that sufficient bandwidth be provided in the low rate stations to permit binary-phase-shift-keying rather than quadriphase-shift-keying for these stations.

The impact of intersymbol interference on communication efficiency when transmitting 80 Mbps at rate 1/2 through a 125 MHz channel is still unresolved. LINKABIT studied the simulation results provided by CSC, but no quantitative statements are yet justified. Continuing effort is indicated.

Certain DCS users desire substantially lower error probabilities than 10^{-5}. Such probabilities can be attained, subject of course to low probability system failures, by concatenation of a short interleaved convolutional code and feedback decoder, such as the LINKABIT L1011. Encoding of data prior to multiplexing by the AN/GSC-24 and decoding following demultiplexing would provide protection through the TCF, ICF, and satellite link. Cost of this protection is a doubling of required user data rate, since a second rate 1/2 code is utilized.