General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.

- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.

- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.

- This document is paginated as submitted by the original source.

- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

FINAL REPORT

THE STUDY OF SEQUENTIAL DECODING TECHNIQUES FOR SPACECRAFT TELEMETRY SYSTEMS

(12 JANUARY 1968 – 12 JUNE 1968)

PREPARED FOR
GODDARD SPACE FLIGHT CENTER
GREENBELT, MARYLAND

UNDER
CONTRACT NO. NAS5-11503

C&S INCORPORATED
communications & systems, incorporated
6565 Arlington Boulevard, Falls Church, Va. 22046

Final Report

for

THE STUDY OF SEQUENTIAL DECODING TECHNIQUES FOR SPACECRAFT TELEMETRY SYSTEMS

(12 January 1968 - 12 June 1968)

Contract No.: NAS5-11503

Goddard Space Flight Center

Contracting Officer: J. C. Comstock

Technical Monitor: John Sos

Prepared by:

Communications & Systems, Incorporated

Falls Church, Virginia

Project Manager: Howard Lefkowitz

for

Goddard Space Flight Center
Greenbelt, Maryland

ABSTRACT

The object of this report is to summarize the results of a study performed on the application of convolutional encoding-sequential decoding to spacecraft telemetry systems. This report deals with the application of this coding technique to both near earth and deep space telemetry missions.

The primary area of investigation for near earth missions involved a consideration of the IMP PCM telemetry system. This investigation resulted in the conclusion that there are no apparent problem areas in the design of this system. In investigating the channel characterizing the IMP telemetry system it was noted that the tape recorder equipment may give rise to burst type errors. However, the IMP sequential decoder can correct these errors at the expense of computer time. In order to provide insight into the treatment of burst errors simulation results are presented which demonstrate the effect of burst errors, of length 2-6, on a sequential decoder.

In considering the design of a coherent deep space telemetry link, using convolutional encoding-sequential decoding, an important limitation is the ability to provide a suitable phase reference. Errors in the estimation of this phase reference will seriously degrade the system performance. This degradation manifests itself as an inability to achieve the predicted energy per symbol per noise density ratio (E_S/N_0) in the sequential decoder. The techniques involved in determining the degradation and the means for maximizing the decoder performance are presented in this report.

Several parameters are of particular importance in the design of a space telemetry system which will utilize sequential decoding. In the encoder area some preliminary work is presented which indicates that the use of a nonsystematic code can provide a better undetected error rate (with shift register lengths about one half of that of a systematic code) than the systematic code from which it was derived. In the area of code rate selection it was found that a basic limitation exists due to the spacecraft oscillator stability. The results of this work imply that an optimum code rate exists for a specified oscillator stability, data rate and carrier frequency. Sequential decoding using quantized channel outputs is analyzed; theoretically optimum threshold settings are derived and practical implementation is discussed.

In the area of synchronization some preliminary work is presented which provides insight into the following:

(a) the determination of mean-squared timing error given the structure of a symbol synchronizer

(b) the structure and performance of an optimum or near-optimum symbol synchronizer and

(c) the system parameters effect on a symbol synchronizer.

The techniques applicable to the digital implementation of a coherent receiver are presented. This analysis leads to the development of a ground station which can utilize a general purpose computer for performing the functions of phase locked loops, bit synchronizers and mixers. In this situation the parameters are binary words inserted into the permanent

computer memory. Evidence indicates that a digitally implemented decision-directed feedback sequential decoder represents an effective approach to minimizing the deterioration in performance which is present in a system operating at high symbol error rates.

The potential advantages, in terms of equipment simplicity and cost, to using threshold decoding for near earth missions as a "quick-look" device are presented. A simplified implementation to determine feasibility was configured and evaluated, concurrent with computer simulation, for systematic convolutional codes.

The concept of concatenation in conjunction with sequential decoding was reviewed as to its applicability to spacecraft telemetry systems. Recommendations were made relative to the possible advantages which might be obtained utilizing this coding technique and future study areas which could result in a viable deep space telemetry system.

The techniques developed for simulating a sequential decoder on the IBM 1130 computer are presented. In addition, the impulse method of computing parity checks for convolutional codes is discussed along with the simplification which can be realized using this technique.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Continued)

LIST OF ILLUSTRATIONS (Continued)

LIST OF TABLES

Section 1

INTRODUCTION

The fundamental limit to the performance of a communication system is set by the channel capacity theorem which states that, for a given signal-to-noise ratio, the probability of error in a digital communication system can be made arbitrarily small providing the information rate is kept below a maximum rate called the channel capacity. In practice, it has been difficult to develop implementable system that performs within 8 or 9 dB of channel capacity.

With the advent of sequential decoding (Reference 2), it is possible to increase the information rate to within 3 or 4 dB of channel capacity. In the past the equipment necessary to implement this technique has proved to be sufficiently complex so that its use in communications systems has been limited. The rapid development of microminiature components and integrated circuit technology during the past few years has resulted in the ability to construct reasonably sized encoding and decoding equipments. With the corresponding improvements in integrated circuit costs and reliability, convolutional encoding-sequential decoding can now be used in space applications to provide dramatic improvements in communication efficiency. Indeed, parallel developments in computer technology have brought the state-of-the-art to the point where, in many applications, there exists a choice of either building special purpose equipment or using small, third-generation, general purpose digital computers.

This report describes the work accomplished in the performance of a "Study of Sequential Decoding Techniques for Spacecraft Telemetry Systems", under contract NAS5-11503. The report presents the results of an advanced study effort and is not intended to provide the basic principle of convolutional encoding-sequential decoding. For descriptions of this coding technique, including the performance enchancement obtainable, the reader is referred to References 1 through 4.

The major study tasks performed during this investigation were as follows:

a. An investigation of the IMP PCM Telemetry System was performed to determine the channel characteristics effects on the sequential decoding system performance.

b. The design of sequential decoding systems for both near earth and deep space missions was investigated in order to provide basic information relative to the modulation method, encoding technique, decoding considerations and synchronization.

c. A special study was performed regarding the digital implementation of ground station receivers by utilizing state variable techniques to synthesize the digital equivalent of analog devices.

d. The decision-directed feedback (DDF) method of sequential decoding was studied for its potential application to deep space missions where high symbol error rates are present.

e. The possibility of utilizing threshold decoding as both a real time decoder and a "quick look" device, when adequate E_b/N_o margin is available, was investigated for telemetry applications in near earth missions.

f. The application of algebraic codes to space telemetry missions was investigated.

Section 2 presents the summary and conclusions for the entire study. Section 3 summarizes the recommendations resulting from the analytical and simulation analyses presented in the main body of the report.

The analysis of the IMP PCM telemetry system indicated that the system will perform satisfactorily. In characterizing the channel it was noted that short burst errors might be generated by the tape recording system. The effects of burst errors on the IMP telemetry system performance is presented in Section 4.

Section 5 discusses the application of sequential decoding to both near earth and deep space missions. Considerations in the choice of optimum modulation technique and the effects of a noisy phase reference are discussed. To assist in the selection of coding system parameters the following topics are discussed:

a. Comparison of systematic and non-systematic codes

b. Selection of the code rate

c. Choice of code constraint length

d. Quantization level and location

e. Synchronization with high symbol error rates

Section 6 presents a short discussion on the selection of earth terminal sequential decoding equipment.

Section 7 discusses several special topics which are applicable to the design of Space Telemetry Systems which utilize coding techniques. This section presents the following study areas:

a. Digitally implemented ground terminal receivers using computer techniques.

b. Discision-directed feedback sequential decoders

c. Threshold decoding

d. Concatenated convolutional/algebraic codes

e. Simulation programming techniques for the IBM 1130 or 1800 computers.

Section 2

RESULTS AND CONCLUSIONS

2.1 IMP TELEMETRY SYSTEM

A review of the IMP system indicated that it would perform satisfactorily under all conditions. It was noted that the communications channel contained a mechanism which could cause short burst errors (on the order of 2 to 10 bits). Since the coding system design was based on a random error channel the possibility of having dependence among received digits which are in error would increase the sequential decoder computational time. This will not measurably effect the IMP system performance. However, because of the possibility of having burst errors in future systems this area was treated in depth. In considering the operational aspects of the IMP system two major areas become apparent:

a. Much of the time available E_b/N_o will be far in excess of that required for sequential decoding.

b. The proposed "quick look" method of data reduction may not provide the user with sufficiently reliable information digits.

It, therefore, seems reasonable to consider the possibility of alternate and less expensive techniques for decoding the bulk of the received data and to reserve sequential decoding for data for which adequate margins do not exist. This leads to the consideration of using a threshold decoder for the "quick look" capability. This type of decoder could provide excellent performance, for this operational mode, as well as indicating when sequential decoding is required. The use of threshold decoding can provide approximately a 4 dB advantage over decoding only data bits, at decoded error rates in the vicinity of 10^{-5}.

In analyzing the effects of short burst errors on sequential decoders the following results were obtained:

a. Burst errors of very short length (i.e., on the order of two to six) cause an increase in the computational load imposed upon the sequential decoder.

b. Theoretical considerations lead to the conclusion that for both the hypothesized burst channels and the binary symmetric channel, the computational behavior is bounded by $p_r [C > L] \sim L^{-\rho}$. Further, the Pareto exponent ρ, for the burst channel, was shown to be greater than that for the corresponding random error channel. This leads to the conclusion, in principle, sequential decoding over short bursts may not be as serious as problem as the simulation results demonstrate if appropriate modifications were to be made.

c. The simulation results were intentionally performed with parameters appropriate to a random error channel. That is, use of the likelihood computation, code generating polynomial, and Fano algorithm implementation all presupposed a random channel. The modifications referred to in (b) would encompass changes in one or possibly all of these factors. As presently envisioned the sequential decoder in no way uses the information that errors tend to cluster if a burst channel is postulated.

d. The consideration of theoretical bounds and practical implementations, appropriate to sequential decoding, on channels with memory represents an area about which relatively little is known. The models used in this analysis are simple, but do provide certain insight into the more general problem.

e. The analytical results presented tend to lead to the conclusion that scrambling is not the best approach to burst channel decoding.

2.2 SEQUENTIAL DECODING FOR SPACE MISSIONS

The use of sequential decoding techniques for space telemetry missions requires careful attention to all aspects of the entire system. This implies that the performance benefits inherent in the use of this powerful decoding technique can only be fully exploited through an optimum integration of all the system parameters. This includes such parameters as data rates, transmitter power, antenna gain, orbital parameters, required error performance, real time data reduction needs, etc. The optimum use of this coding technique may actually require some changes in the telemetry system design philosophy presently employed by GSFC. This is particularly evident in the degree to which the total communications channel must be defined. The satisfactory application of sequential decoding to future space missions may take the form of optimizing the channel performance independent of the experimenters needs with the resulting data capacity being apportioned to the various users on an adaptive need basis.

For certain types of missions, particularly deep space applications, the performance needs lead to the choice of binary phase reversal PSK as the most desirable modulation technique. This modulation technique is compatible with the encoder, provides high communication efficiency, and reduces the required received bit energy.

2.2.1 Encoding

Preliminary simulation results indicated that the nonsystematic codes (with shift register length about one half that of the systematic codes) showed a better uncorrected error performance than the systematic codes from which they were derived. The average computational load for the non-systematic code was slightly higher than for the systematic codes-indicating the non-systematic code essentially "works harder" but does a superior job of correcting errors. The same trend was noted for the non-systematic code derived from a length 32 systematic code which was suggested by GSFC for the IMP system. From this preliminary work it appears that non-systematic codes offer some attractive features - especially at low redundancies (e.g., rate 1/2).

In the consideration of code rate, the analytical results indicated that an optimum code rate exists for a specified oscillator stability, data rate, and carrier frequency. In a pracal sense, data rates should exceed one bps in order to realize efficient phase coherent communication. Convolutional code rates of less than 1/4 offer a negligible reduction of E_b/N_o with all but the most stable oscillators. With a code rate of 1/3, E_b/N_o is increased by less than 0.2 dB compared with a code rate of 1/4. Operation near $E_b/N_o = 2.5$ dB is considered feasible.

In considering the choice of constraint length as L is increased, while α is held fixed, the decoded bit error probability diminishes while the average number of computations per bit required to achieve this performance increases to an asymptotic value. It would appear

from the limited study results that very long codes would require about 2.4 and 4 computations per bit at $\alpha = 0.867$ and 0.956 respectively. Theoretically, when α approaches one and L approaches infinity, the average number of computations per bit will also approach infinity.

2.2.2 Decoding

In considering the number of quantization levels and their locations, the following results were obtained:

a. A reduction in E_b/N_o on the order of 1.5 dB over the "hard decision" case may be realized by using four-level quantization. Additional improvement on the order of 0.4 dB may be achieved by using eight-level quantization.

b. Eight-level quantization approaches to within approximately 0.2 dB of the theoretical minimum E_b/N_o for binary antipodal signaling with continuous output (i.e., infinite level quantization). Hence, little justification exists to consider levels of quantization higher than eight.

c. Because of the relatively constant behavior of threshold setting as a function of Video SNR, nearly optimum performance may be obtained using preset level divisions. Performance could not be greatly enhanced by using adaptive techniques. This result is most useful when considering practical detector design.

d. A single counter can be used with three- and four-level quantization schemes to ascertain whether the thresholds are set properly. The counter can be used to automatically adjust the thresholds to compensate for AGC drift and similar phenomena.

2.2.3 Symbol Synchronizer

A preliminary analysis was developed to provide insight into the following synchronizer problem areas:

a. The determination of the mean-squared timing error of a given symbol synchronizer.

b. The determination of the structure of an optimum or near-optimum symbol synchronizer and its performance characteristics.

c. The influence of the systems parameters on an optimum symbol synchronizer.

2.3 EARTH TERMINAL SEQUENTIAL DECODING EQUIPMENTS

Using external parity checks and a third generation computer, having the following parameters;

Memory Cycle:	1 Microsecond
Memory Size:	4096 words
Word Length:	18 bits

the possibility exists of achieving decoding rates on the order of a few Kilobits/sec. If, for certain applications, channels with data rates of lesser magnitude are required then the computer can be time shared and perform other functions simultaneously with sequential decoding. For NASA sites which do not possess a computer, the question arises as to the desirability of obtaining a general purpose computer. One function of the general purpose computer would be to perform sequential decoding, or to obtain a specific device for sequential decoding. This question is related to the total system functions to be served by the site. It is important to recognize that both possibilities exist.

2.4 DIGITALLY IMPLEMENTED COHERENT RECEIVER UTILIZING SEQUENTIAL DECODING WITH DECISION-DIRECTED FEEDBACK

The advantages of digital implementation are substantial when the ground station is used to process data for a variety of missions with different signaling characteristics. For such an application the receiver circuits following the IF amplifier would be reduced to a digitizer which converts the IF amplifier output to a binary bit stream followed by the digital processing circuitry which can be either a general purpose computer or special digital modules. These modules replace the customary receiver components, such as phase locked loops, bit synchronizers and mixers. The parameters of the particular mission are binary words inserted into a permanent memory.

Although analytical proof is not yet available, there is substantial evidence that the DDF method of decoding represents an effective approach to eliminating the deterioration in performance which is present in system operating at high symbol error rates. The difficulties are particularly evident in systems utilizing sequential decoding since these have the lowest energy/symbol/noise density ratio of all binary signaling methods. Furthermore, as the rate (information bits/symbol) is decreased the problem becomes more acute, because the binary modulation on the received signal represents an uncertainty, which causes deterioration in the tracking performance of the carrier tracking phase locked loop and the bit synchronizer. Of course, the higher the uncertainty, i.e., the symbol error rate, the greater its negative effect. This is reflected in increased mean-squared tracking errors and corresponding deterioration in performance. The degree of deterioration depends on the system data rate. A high data rate system requires power which is far in excess of that required in the aforementioned tracking circuits. In these systems substantial effective power loss caused by symbol uncertainty can be afforded without loss in tracking accuracy. In low data rate systems the available power may barely be sufficient for tracking, and any uncertainty might cause substantial tracking errors. From the above considerations it is clear that DDF is effective in low data rate systems.

This area of the study treated five major topics. First, the DDF approach is explained and contrasted with conventional sequential decoding. Next the basic concepts associated with digital processing, such as state variables and optimum recursive filtering and prediction, are presented. Subsequently a digital DDF sequentially decoding receiver is described, and its major component blocks are developed in detail. Next a comparative analysis of DDF versus conventional sequential decoder efficiency as a function of oscillator stability is given. Finally, a digital coherent sequentially decoding receiver without DDF is briefly described.

2.5 THRESHOLD DECODING

It was illustrated that threshold decoding of convolutional codes can provide a real time decoding capability relatively inexpensively with minimum equipment requirements. Although

the error correcting capability of a threshold decoder is not as great as that of a sequential decoder, it was shown that two specific codes, one of length 7 and one of length 32, which are optimum for use with a sequential decoder also perform well with a threshold decoder. Specifically, it was found that the threshold decoding of both of these codes provides correction of all single and double error patterns and a sizeable fraction of the triple error patterns.

Also, it was shown that the addition of a single modulo-two adder and a pulse counter to a threshold decoder provides the capability of accurately estimating the probability of error of the received symbols.

Based upon these results, the proposed use of a threshold decoder as a "quick-look" technique is firmly justified.

2.6 CONCATENATED CONVOLUTIONAL/ALGEBRAIC CODES

As briefly outlined in this report, the work to date regarding concatenation of codes has been largely theoretical in nature and oriented toward a fundamental understanding of the properties and performance of these codes. The promising results favor further investigation of the potential of this approach for space telemetry problems. The following problem areas, applicable to telemetry using convolutional encoding/sequential decoding, are ameniable to solution using this technique.

a. Capacity Limitations - For deep space missions in particular, it is important to utilize all available power to the maximum extent practicable. It is well known that such a channel may be represented as an additive white gaussian noise channel for which $(E_b/N_o)_{min} = -1.6$ dB. For a single, convolutional encoder/sequential decoder the minimum theoretical E_b/N_o for a rate 1/3 code and eight-ary output channel is about 2.2 dB. When practical allowances are made for noisy phase references, resynchronization and other implementation difficulties, another 1-2 dB loss may be incurred. The work of Falconer has shown an approach which is capable of considerably enhanced performance. Thus, the first area in which the concatenation scheme might aid is that of increased capacity (or longer range in the case of a deep space probe).

b. Decoder Overflow - A well recognized problem in conjunction with sequential decoding is that of decoder overflow. Concatenation could be used in a number of possible ways to reduce the effect of overflow. One such way is to increase the Pareto exponent which will result in a decrease in the overflow probability.

An alternate approach would be to permit overflow to occur with some probability and then treat blocks for which overflow occurred as erasures and employ burst correction codes to correct these erasures. These could either be block or convolutional codes.

c. Implementation Simplification - A third area deserving attention is the question of implementation. A prime motivation behind the whole concept of concatenation is that of less difficult hardware implementation. Thus encoder reliability could be enhanced by some concatenated approach and decoder complexity might be reduced. For example, it might be reasonable to consider shortened constraint length convolutional encoders in conjunction with algebraic and sequential decoding. While the undetected error rate of the sequential decoder without algebraic decoding would be high, the outer (block) coder could reduce the error probability. Allowing more undetected errors would significantly reduce computational requirements.

2.7 SIMULATION PROGRAM FOR SEQUENTIAL DECODERS

To assist GSFC in the use of an IBM 1800 computer for simulating a sequential decoder the techniques, developed by C&S for an IBM 1130 computer, are presented. It is C&S's understanding that the main line program as well as the subroutines provided may be used directly on the IBM 1800, with the possible requirement that they be recompiled. To this end C&S has furnished the appropriate computer card decks.

An approach to more efficient programming, known as the impulse method of computing parity checks for convolutional codes, was developed. This approach is particularly well suited for digital computer applications, and may also be useful for some types of analytical work.

Section 3

RECOMMENDATIONS

Based upon the study results presented in this report C&S recommends that the following courses of action be considered.

3.1 IMP TELEMETRY SYSTEM

a. A threshold decoder is recommended as a "quick look" technique. During those periods of time when the system margin exceeds 7 dB, the output of the threshold decoder will be essentially error-free (less than one bit error per year at 400 bps).

b. The small additional circuitry required to estimate the channel error rate should be added to the threshold decoder. This will provide a reliable measure of which tapes are sufficiently noisy to require sequential decoding.

c. Considering future telemetry systems, for channels on which short burst errors may occur, this should be taken into account in the sequential decoder design. Tradeoff studies should be made among the various possible corrective measures which could be taken. Such measures might include: scrambling of the transmitted sequence, incorporation of burst error correcting codes as part of the total channel coding problem and exploiting the nature of the error dependence within the decoding algorithm.

d. Additional theoretical work is required to develop better and more meaningful, representations of channels with memory. This work would have the objective of providing direction for the most efficient practical implementation of sequential decoding on channels with memory.

3.2 SEQUENTIAL DECODING FOR SPACE MISSIONS

a. In the area of nonsystematic codes further analytical and simulation work should be undertaken to confirm the preliminary results obtained. This is particularly important for long constraint length codes.

b. Future study effort should be devoted to computer simulation of lower rate codes, in particular rate 1/3 and possibly rate 1/4.

c. The theoretical analysis of symbol synchronizers should be extended to determine

1. Detailed structure of the optimum synchronizer
2. Performance curves relating the mean square timing error to the carrier power-to-noise density ratio, data rate, and oscillator stability.
3. Development of a systematic approach to the evaluation of typical symbol synchronizers.

4. Derivation of curves as in (2) for the suboptimum synchronizers, and comparison with the optimum curves.

Based upon the theoretical results an implementation program involving the use of a computer should be initiated.

3.3 EARTH TERMINAL SEQUENTIAL DECODING EQUIPMENT

a. The computers presently installed at STADAN stations should be examined in detail to estimate the sequential decoding capability.

b. NASA should take a general system oriented view of the receiving stations with regard to coded telemetry. For example, it may be much more economical to use a decoding technique less powerful than sequential decoding for on-line operation and reserve sequential decoding as a completely off-line capability.

3.4 DIGITALLY IMPLEMENTED COHERENT RECEIVER UTILIZING SEQUENTIAL DECODING WITH DECISION-DIRECTED FEEDBACK

Based upon the potential system improvement described in this report, a systematic study effort involving the replacement of a ground station receiver, is recommended.

A suggested breakdown of the recommended activity into a number of tasks to be performed in sequence, follows.

Step 1. Development of a Costas type phase-locked loop computer program, to be run with simulated signals generated inside the computer. These signals will be generated under the assumption of ideal symbol sync and nonencoded PSK modulation.

Step 2. Development of a symbol synchronizer program. The input signals will be simulated, i.e., generated inside the computer, but will be different from Step 1 in that ideal symbol synchronization is not assumed. They will be of the type obtained when the received PSK signal is asynchronously sampled at a rate on the order of ten samples per symbol. The PSK signal is not encoded.

Step 3. Combining Step 1 and 2, a program is obtained which is capable of demodulating PSK signals and which has as its output the demodulated binary bits. At this stage the input signal to the program is still simulated inside the computer as described in Step 2.

Step 4. Design and construction of a sampling circuit which will operate with a noisy PSK signal.

Step 5. Feed PSK signal to sampling circuit and feed digital output in real time to computer. Run receiver program and evaluate performance (i.e., count errors). If the noisy PSK signal is from a tape, this might create problems; if the noisy PSK signal is generated in a test set, the actual information bits may be fed to the computer and the symbol errors can be counted.

Step 6. Combine the phase-locked loop/synchronizer program with an independently developed sequential decoder program. No quantization should be included in the decoding;

in this manner the program of Steps 1, 2 and 3 may be used with hardly any modifications. Run program and decode computer-simulated signals.

Step 7. Modify the combined sequential decoder/digitized receiver program for DDF. At this point the signals are still generated internally (simulated in the computer).

Step 8. Build test set which generates noisy PSK convolutionally encoded signals. Feed these to sampling circuit of Step 4.

Step 9. Inject in real time into computer. Run program and decode.

The activity outlined could be subjected to considerable modifications. For example the computer simulation of PSK signals might be omitted and real time injection into the computer might be implemented as the first step. Alternately it might be desirable to implement a system without DDF.

3.5 THRESHOLD DECODER

a. The preliminary work on the potential use of a threshold decoder in conjunction with a sequential decoder should be continued.

b. An implementation program should be initiated, in particular with regard to the IMP telemetry system.

c. A study effort should be initiated regarding the use of non-systematic codes with threshold decoders.

3.6 CONCATENATED CONVOLUTIONAL/ALGEBRAIC CODES

The concept of concatenation of convolutional encoding/sequential decoding with algebraic (or perhaps convolutional) codes should be investigated for space telemetry. The following specific objectives should be considered:

a. Possible hardware simplification and/or increase in reliability

b. Improvement in performance in regard to overflows and undetected errors

c. Decoding for deep space channels at rates in excess of the computational cutoff rate.

Section 4

IMP PCM TELEMETRY SYSTEM

4.1 DESCRIPTION

The Interplanetary Measurements Probe (IMP) telemetry system, as designed by GSFC, utilizes a PCM-PM modulation system. The phase modulator operates with an index of ±1 radian thus providing sufficient carrier power so that it may be tracked with a phase-lock receiver. It is C&S's understanding that the original design provided a 6 to 7 dB margin for a 400 bps data rate at the design range. The addition of a convolutional encoder-sequential decoder to the IMP system was based upon two factors:

a. The requirement, by GSFC, to increase the telemetry data rate to 1600 bits/sec and maintain the margin.

b. The desire to evaluate, under actual flight conditions, a convolutional encoder-sequential decoder system.

The use of a sequential decoder will provide a telemetry system which has an inherent capability of providing a near zero error rate. This will afford an excellent comparative evaluation with the standard GSFC telemetry system during the IMP mission.

For this study C&S was asked to review the telemetry system design from the standpoint of the coding system performance. A preliminary analysis revealed that there are no major problem areas. As discussed in Paragraph 4.2, a mechanism is present which can result in short burst errors (defined as errors occurring in groups of from two to ten bits). Since the design of coding systems is usually based upon a channel having random errors, an analysis was performed to evaluate the effects of short burst errors on a sequential decoder. This analysis is presented in Paragraph 4.4.

4.2 CODING SYSTEM

In reviewing the IMP telemetry system design a mechanism was defined in the communications channel, which could cause burst errors. This is due to the possibility of having a tape recorder dropout.

A typical tape dropout is on the order of 5×10^{-3} inches. The width probability distribution and the dropout rate are not known. However, at a recording speed of 3-1/2 inches per second and a data rate of 3,200 symbols per second (1600 bits per second at a rate 1/2 code) the number of symbols lost is approximately five (1.09×10^{-3} inches/symbol). At 800 symbols per second (4.37×10^{-3} inches per symbol), the loss per dropout is one symbol. Thus tape dropouts will cause errors to occur in groups at the high data rates.

4.3 OPERATIONAL CONSIDERATIONS

The analysis of the proposed IMP telemetry system indicated that the incorporation of sequential decoding involves the following two, separate, but somewhat interrelated, areas:

a. Much of the time the available E_b/N_o will be far in excess of that required for sequential decoding

b. The proposed "quick look" method of data reduction may not provide the user with sufficiently reliable information digits.

The NASA design concept is intended to provide a 6-7 dB margin in E_b/N_o over that required for sequential decoding. During periods of operation in which such a margin actually exists, it does not seem reasonable to insist upon sequentially decoding all blocks of data. This would not aid in demonstrating sequential decoding but would still require large amounts of computer time. (Regardless of the margin, sequential decoding cannot proceed with less than one computation per decoded information bit.) It then seems reasonable to consider the possibility of alternate, and less expensive, techniques for decoding the bulk of the received data and reserve sequential decoding for that data for which adequate margin does not exist.

One possibility is to use the information digits directly and discard the parity digits. This approach possesses two limitations in that it will not work with nonsystematic codes and there is no way of determining if information digits are in error (except by examining data which is already known prior to reception).

With the increased interest in redundancy removal and on board data processing, future systems will not be able to rely upon "known" data as a means of verifying correct system operation.

This leads to the second area, which involves the provision for an on-site "quick look" facility. The concept of using only information bits as a "quick look" capability makes inefficient use of energy and provides the user with no indication of bit errors. As an alternate possibility, the use of a threshold decoder should be considered for use at the receiving stations. This would have the following significant advantages:

a. Threshold decoding, even with the convolutional code generators, used for sequential decoding, can provide good performance at low values of E_b/N_o. For example, the threshold decoder discussed in Paragraph 7.2 could provide an output error rate of 10^{-5} with $E_b/N_o = 5.2$ dB.

From the GSFC computed power budgets for the IMP at 400 bps the received E_b/N_o without margin is approximately 11.5 dB which, for a rate 1/2 code corresponds to an E_b/N_o of 8.5 dB. For the proposed modulation scheme this corresponds to a symbol error probability of approximately 8×10^{-5}. This results in a calculated bit error probability for the threshold decoder of approximately 5×10^{-11} which corresponds to less than one decoded bit error per year at 400 bps. Thus, threshold decoding of the total symbol data stream is seen to provide essentially error-free data bits. The same value of E_b/N_o, decoding of only the information bits by conventional "quick-look" means would only provide marginal results.

b. Threshold decoding could provide an excellent indicator of the presence of a high input error rate. Hence, threshold decoding could be used to identify those blocks which should be sequentially decoded.

c. Such a decoder would be very inexpensive and should be capable of being easily implemented into existing receiving stations.

d. Unlike sequential decoding, threshold decoding proceeds in real time.

e. Threshold decoding may be useful with non-systematic codes. This area needs careful investigation to determine theoretical and practical performance limitations.

Two performance curves showing the potential of threshold decoding were computed. The Figure 1 considers performance with a convolutional systematic code of length 7 and Figure 2 shows performance with a code of length 32. (The reason for the choice of these codes and specific tap connections are discussed in Paragraph 5.3.1). The curves illustrated in Figures 1 and 2 provide performance data for the following cases:

a. Decoding of data bits only

b. Transmission of uncoded information

c. Threshold decoding of the convolutional code

d. Sequential decoding (hard decision) of the convolutional code.

The first three represent computed results, and the sequential decoding curve represents simulated results. It is important to realize that the curves for threshold decoding apply to convolutional codes which were developed for sequential decoding.

Inspection of Figures 1 and 2 reveals that threshold decoding has approximately a 4 dB advantage over decoding of only data bits at error rates in the vicinity of 10^{-5}. Hence, it may be reasonable to consider threshold decoding for much of the received data and reserve sequential decoding for those times during which the available E_S/N_O is less than about 8 dB. (The threshold decoder could be a good indicator of the necessity for sequential decoding).

4.4 EFFECTS OF SHORT BURST ERRORS ON CHANNELS WITH MEMORY

The computational behavior of sequential decoders on random error channels is relatively well known and has been extensively investigated. Additionally, it has been generally recognized that any dependence among received digits which are in error will adversely affect the sequential decoder. It is toward the consideration of this characteristic of sequential decoders that this section is directed.

The major purposes of this analysis were threefold:

a. To demonstrate by computer simulation the effect of short burst errors upon the computational behavior of a sequential decoder. The particular source of these errors for the IMP system is discussed in Paragraph 4.2.

b. To present some theoretical arguments pertaining to capacity and computational behavior for sequential decoding on the short burst channel and relate these results to the computer simulations.

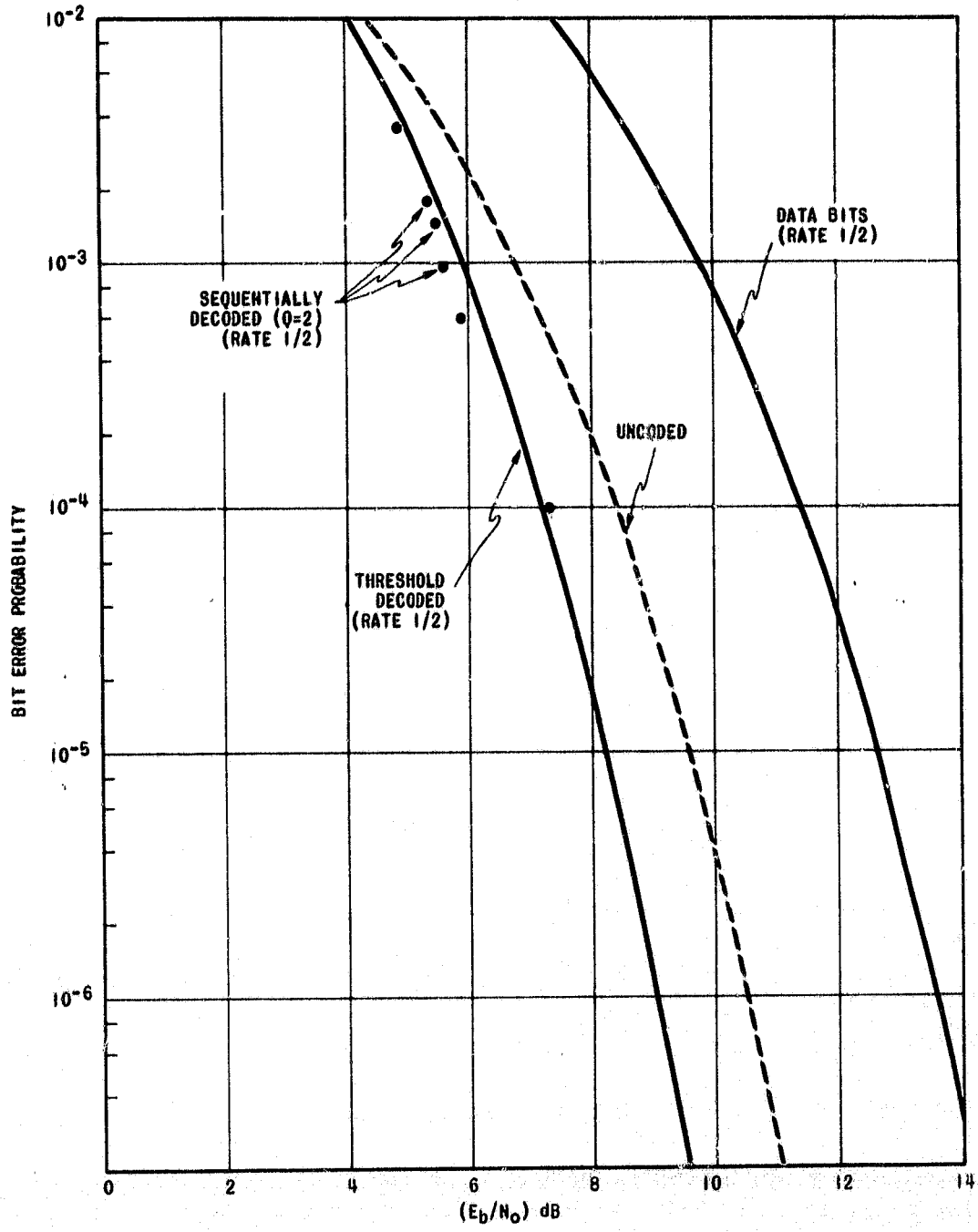


FIGURE 1. Quick Look Decoding for a Length 7 Code

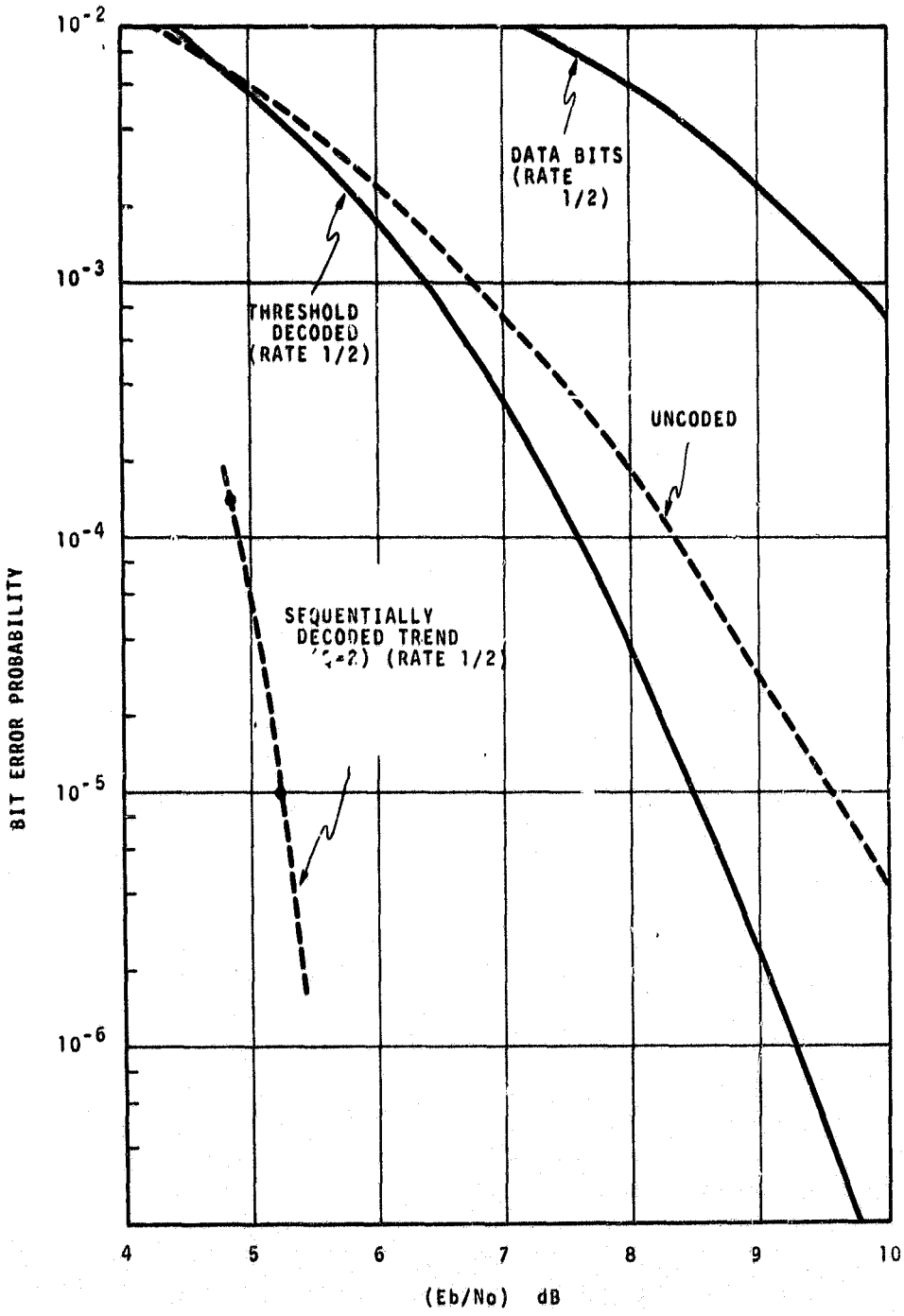


FIGURE 2. Quick Look Decoding for a Length 32 Code

c. To outline further effort necessary to investigate the effect of burst errors and the possibility of taking corrective action for future telemetry systems.

Each of these objectives is discussed below. It is pertinent to emphasize that the aim of this particular section is not to provide specific solutions to a particular channel but rather to provide insight into the nature and magnitude of the problem.

4.4.1 Computer Simulation Results

The effect of short burst errors on the computational behavior of a sequential decoder was simulated on an IBM 1130 digital computer. The purpose of these simulations was to demonstrate the magnitude of the computational problem when error dependencies are introduced; complete solutions to this problem constitute a considerable effort and are not a part of this report.

A binary input channel using hard decisions was postulated. The input burst errors of length B were simulated as follows: given that an input error occurred, the following B-1 received digits were in error with probability one half. A rate one half, systematic code was used. The encoder was a 32-stage shift register with tap connection 35454506423 (octal). The Fano algorithm was simulated and the metric was calculated as for a binary symmetric channel.

Results were obtained for average probabilities of error of 0.01 and 0.03. (The concept of average probability of error on burst channels is discussed in Paragraph 4.4.2. For the binary symmetric channel with rate one half, these correspond to the ratio R/R_{comp} of 0.68 and 0.87 respectively. The results of the simulations are plotted in Figures 3 and 4. Curves are shown for burst lengths of 1 (binary symmetric channel), 2, 4, and 6. In each case, the curves are based upon 50,000* decoded information bits which were decoded in blocks of length 250 information bits with a tail of length 32. The maximum number of computations per bit was set at 100.

Figures 3 and 4 clearly demonstrate the nature of the computational problem with dependent errors. Consider first, Figure 3. An average error probability of 0.01 corresponds to an R/R_{comp} of about 0.68 for the rate 1/2 code. For the random error channel, computation proceeds rather smoothly with the probability of exceeding three computations per bit becoming rather small. As the burst length increases, the computational load on the sequential decoder is increased significantly. For example, the number of computations required for the B=4 case is from one to two orders of magnitude worse than for the corresponding binary symmetric channel.

As the input error probability is increased, things become somewhat worse. For the case of B=4, ten percent of the blocks now require more than about 60 computations per decoded digit. Also, as the rate approaches R_{comp}, the effect of paired errors become highly significant. (Note that paired errors would be worse than the case B=2 because for the burst cases, the second digit is correct with probability one half.)

Finally, the somewhat peculiar shape of the curves deserves comment. In order to keep the average error probability constant, the number of bursts per block will be a variable.

---

*12,500 for the case B = 6, p = .03.

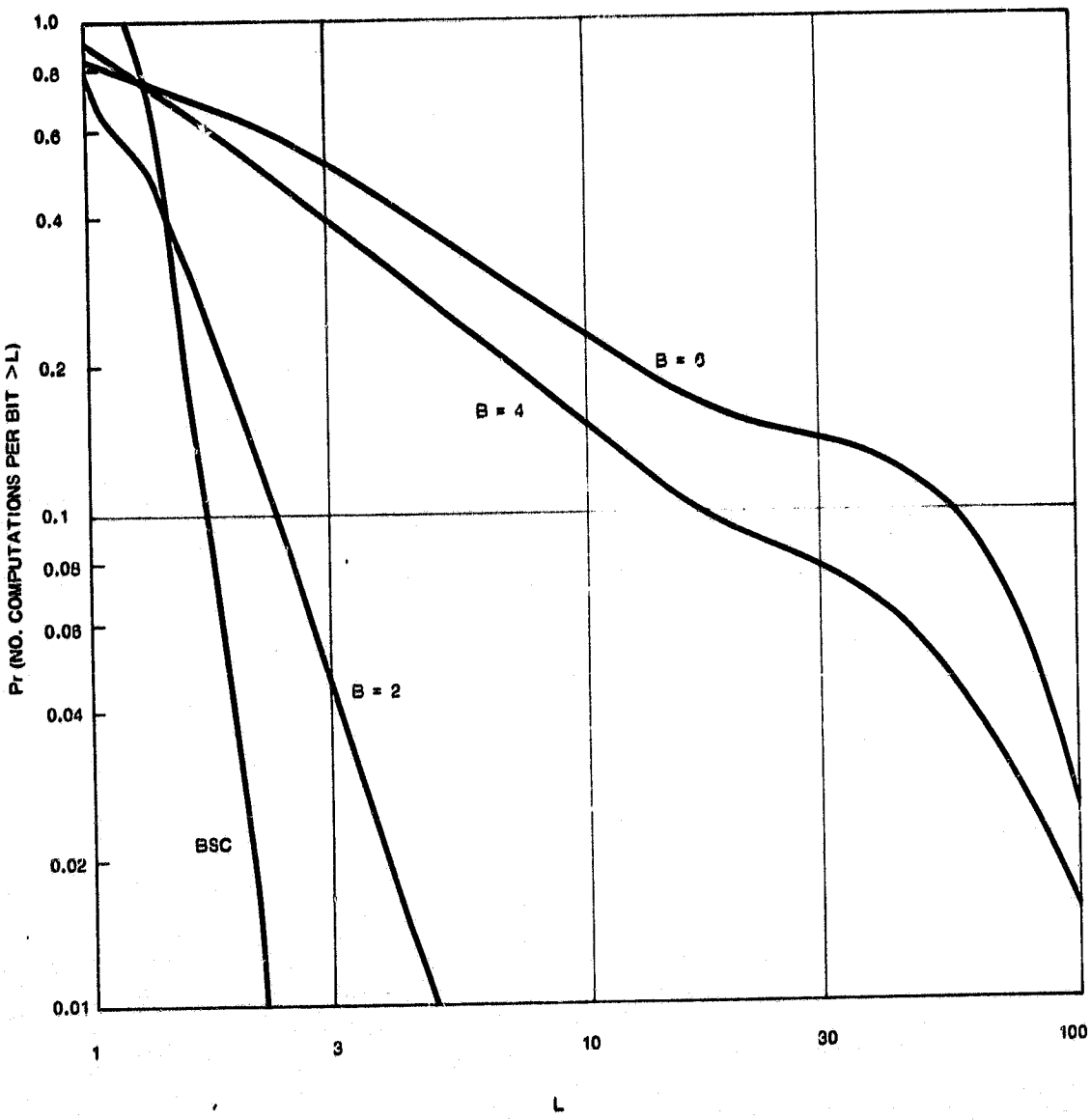


FIGURE 3. Computational Requirement with Dependent Errors for Average Probability of Error of 0.01

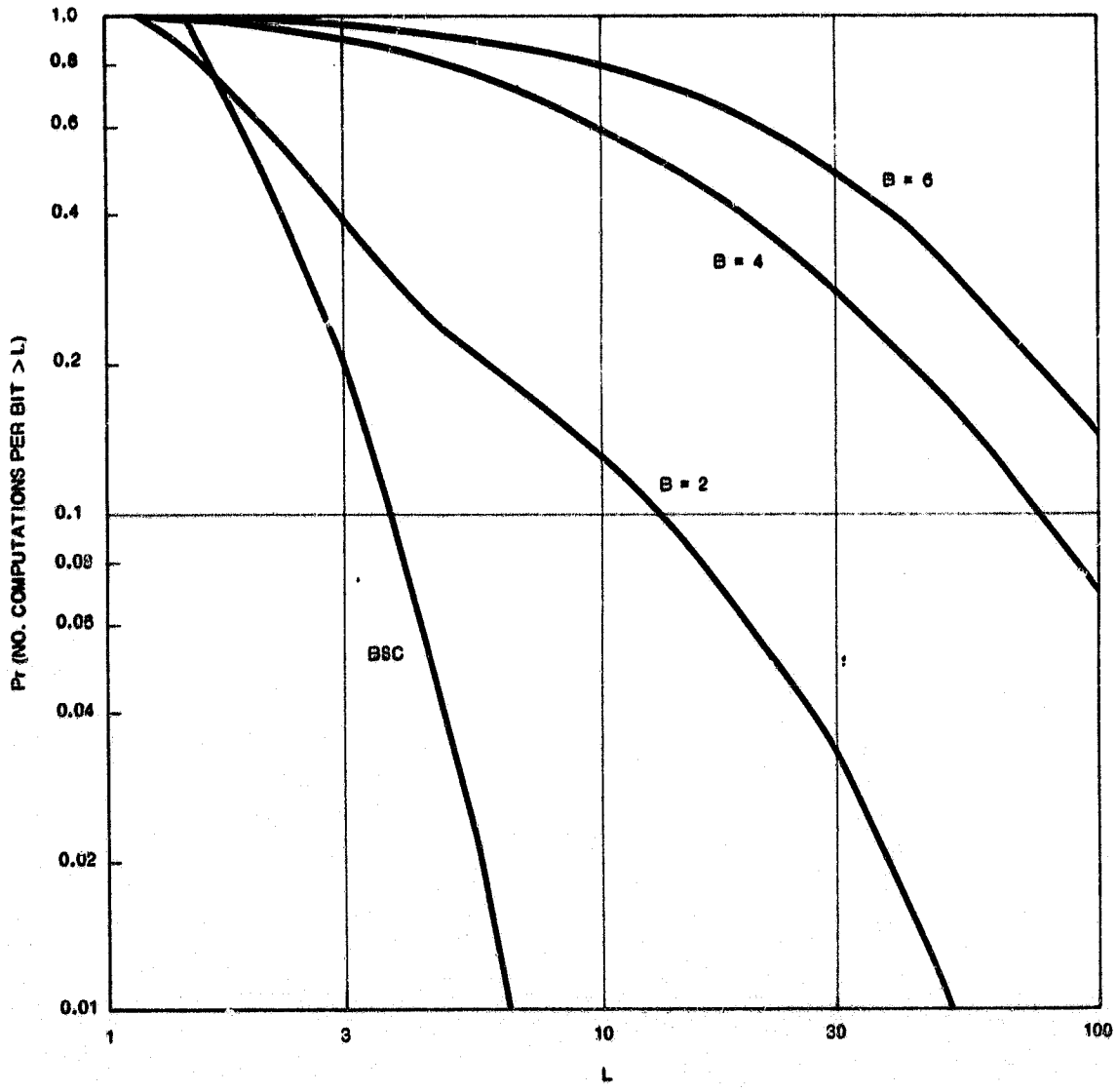


FIGURE 4. Computational Requirement with Dependent Errors for Average Probability of Error of 0.03

Reflection upon this should convince the reader of the reasonableness of the "multi-slope" behavior of the curves. Further, the probability of blocks possessing no errors becomes significant, hence, the burst simulations tend to require less computations in the upper percentiles.

4.4.2 Theoretical Considerations

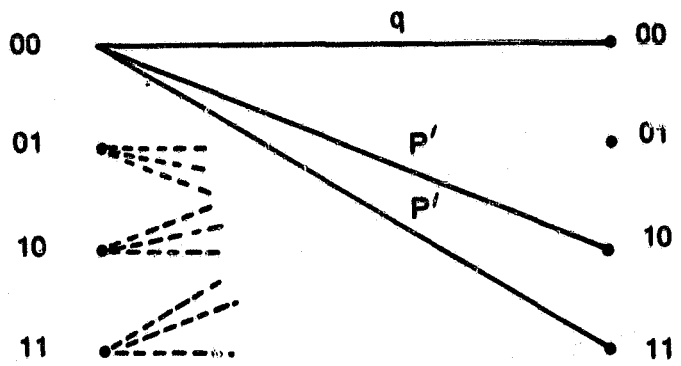
The purpose of this paragraph is to present some theoretical results which pertain to computational behavior and capacity of a sequential decoder when used on a channel with memory. The general solution to sequential decoding performance on channels with memory is an unsolved problem. The objective is to develop some insight into the nature of the performance bounds.

Two general channel models were analyzed. The first represents the short burst channel with binary symbols as a discrete memoryless channel with an M-ary input and M-ary output. The second approach uses a first order Markov model of the channel with memory. This concept was first introduced by Savage, Reference 5.

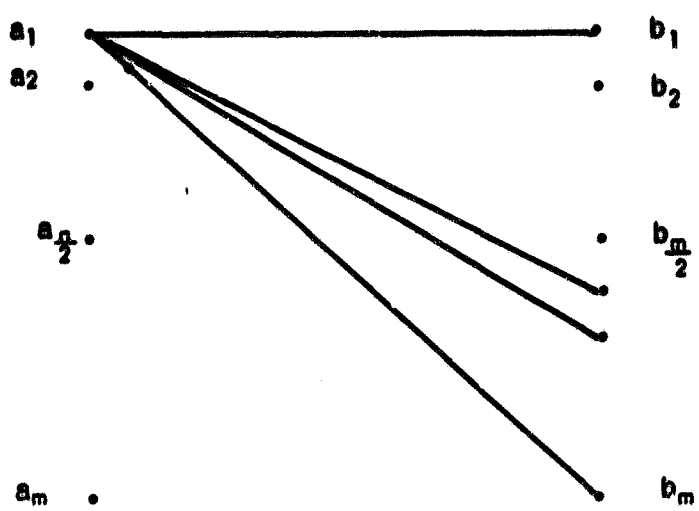
The following paragraphs present calculations of the channel capacity, computational cutoff rate (R_{comp}) and the error bound parameter $E_o(\rho)$ for these models. Relationships for the dynamic computational behavior of a sequential decoder are also shown.

A Discrete Memoryless Channel

Consider the following channel:



where the dotted lines indicate a complete symmetry. This channel may be viewed as a simple approximation to a burst channel where the bursts are of length two. That is, given that an error occurred in the reception of the first digit, the second digit will be in error with probability one half. The extension of this model to bursts of length B is straightforward. The resulting channel will have $M = 2^B$ inputs and M outputs as shown below.



Such a model has some recognized shortcomings as a burst error model. For example, bursts may begin only every B digits. However, for an inter-burst interval, much greater than the burst length B, the channel model will still be instructive, particularly as regards the decoder computational behavior. Also, an objection may be raised based upon the fact that the burst length B is fixed.

Before presenting the results of the theoretical calculations it is important to derive a basis of comparison with a "corresponding" binary symmetric channel. Two channels will be considered "corresponding" if the average probability of symbol error over both channels is the same.

Let p be defined as the average probability of a symbol error. (For the binary symmetric channel, p is simply the transition probability.) Next, p may be related to the parameters of the burst channel model as follows:

$$p_r \text{ (symbol error} \mid \text{burst error)} = \frac{B+1}{2B}$$

$$p_r \text{ (burst error)} = 1 - q = 2^{B-1} p'$$

Then,

$$p = \left(\frac{B+1}{2B}\right) 2^{B-1} = 2^{B-2} \left(\frac{B+1}{B}\right) p'$$

With the basis for comparison defined, the first consideration is that of the channel capacity for a channel described above compared to that for the "corresponding" binary symmetric channel. Channel capacities for the binary symmetric channel and the burst channel are plotted in Figure 5. Derivation of the equations used to plot this figure may be found in Appendix I. From Figure 5 it is immediately apparent that the burst channel has a higher capacity than the "equivalent" binary symmetric channel. This result could have been predicted since for the burst channel the ideal receiver possesses a greater knowledge of the location of the errors; hence, a higher capacity.

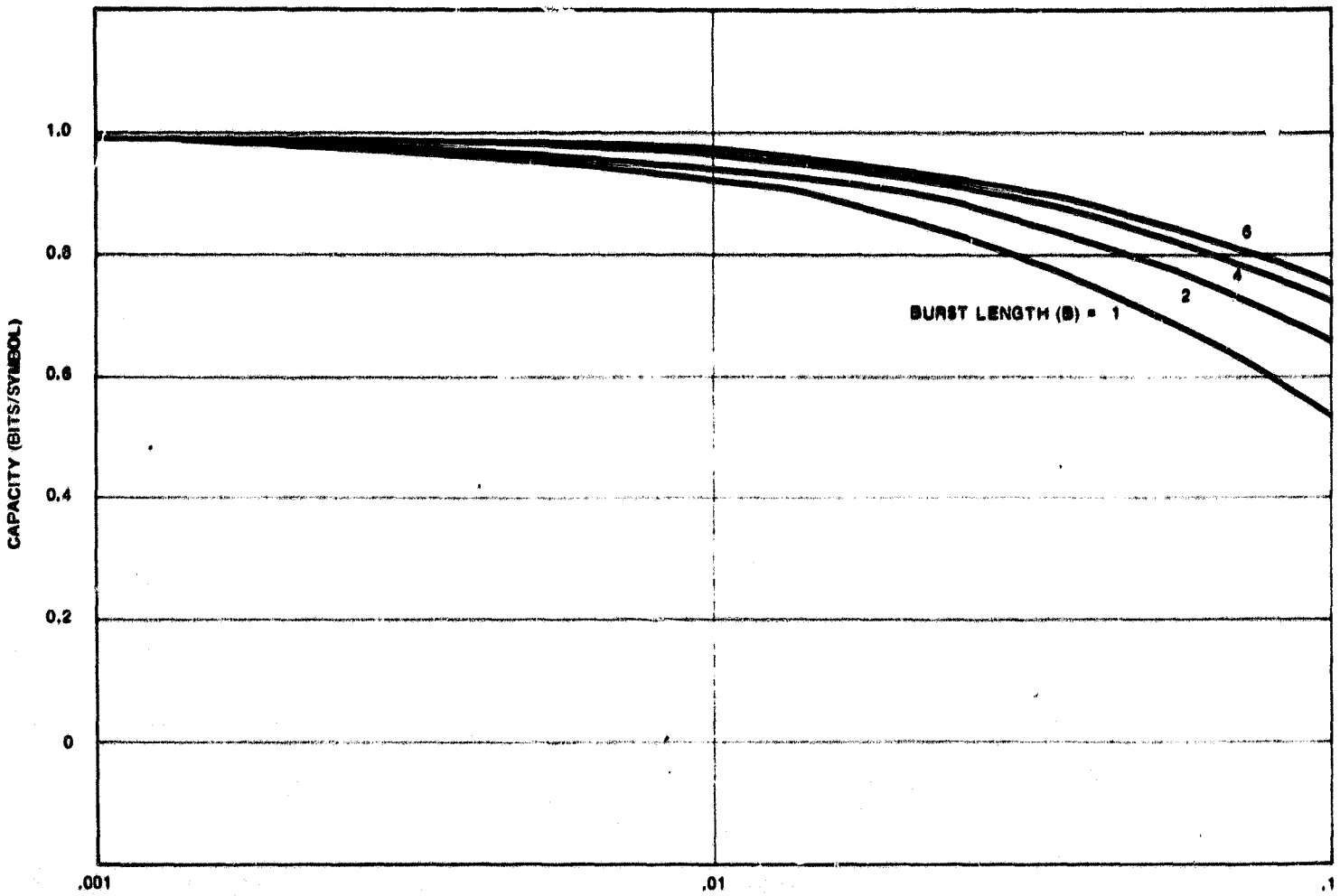


FIGURE 5. Channel Capacity for Binary Symmetric and Burst Channels

For sequential decoding considerations, channel capacity is not as significant as some other parameters. Specifically, consider the following functions which were originally presented by Gallager (Reference 6):

$$E_o(\rho, \vec{P}) = -\ln \sum_{j=1}^{J} \left(\sum_{K=1}^{K} p_k q_{kj}^{\frac{1}{1+\rho}} \right)^{(1+\rho)} \quad 0 < \rho < \infty \tag{1}$$

$$E_o(\rho) = \max_{\vec{P}} E_o(\rho, \vec{P}) \tag{2}$$

where

$\vec{P} = (p_1, p_2, \ldots, p_k)$

K = input alphabet size

J = output alphabet size

Further, define $\hat{E}_o(\rho)$ as the concave hull of $E_o(\rho)$. Using these definitions and the lower bounds on error probability given by Shannon, Gallager, and Berlekamp (Reference 7), the following lower bound relating computational behavior to the parameter ρ has been obtained by Jacobs and Berlekamp (Reference 8.)

$$F_N(L) \geq L^{-\rho} \exp\{-0(\sqrt{\ln L})\} \tag{3}$$

where

$F_N(L)$ = the probability that at least L computations are performed at a distance N from the origin of the tree

$0(\sqrt{\ln L})$ = a term which is of the order of $\sqrt{\ln L}$

In their result, Jacobs and Berlekamp showed that ρ is lower bounded by the constraint that $R \geq R_\rho$ where

R = the rate of the code (nats | symbol)

$$R_\rho \triangleq \hat{E}_o(\rho)/\rho$$

With these concepts in mind a lower bound to the distribution of the computational behavior (Pareto exponent) for any discrete memoryless channel may be calculated. The function $(E_o(\rho \vec{p})$ may be obtained by straightforward substitution of the channel transition probabilities into Equation 1. For this channel it may be shown that all ρ, $E_o(\rho, \vec{P})$ maximizes at $p_1 = 1/M$ and $E_o(\rho)$ is itself concave; hence, $\hat{E}_o(\rho) = E_o(\rho)$. The function $\hat{E}_o(\rho)/\rho$ was calculated for various burst lengths and average probabilities of error, p, of 0.01 and 0.03. The results are plotted in Figures 6 and 7.

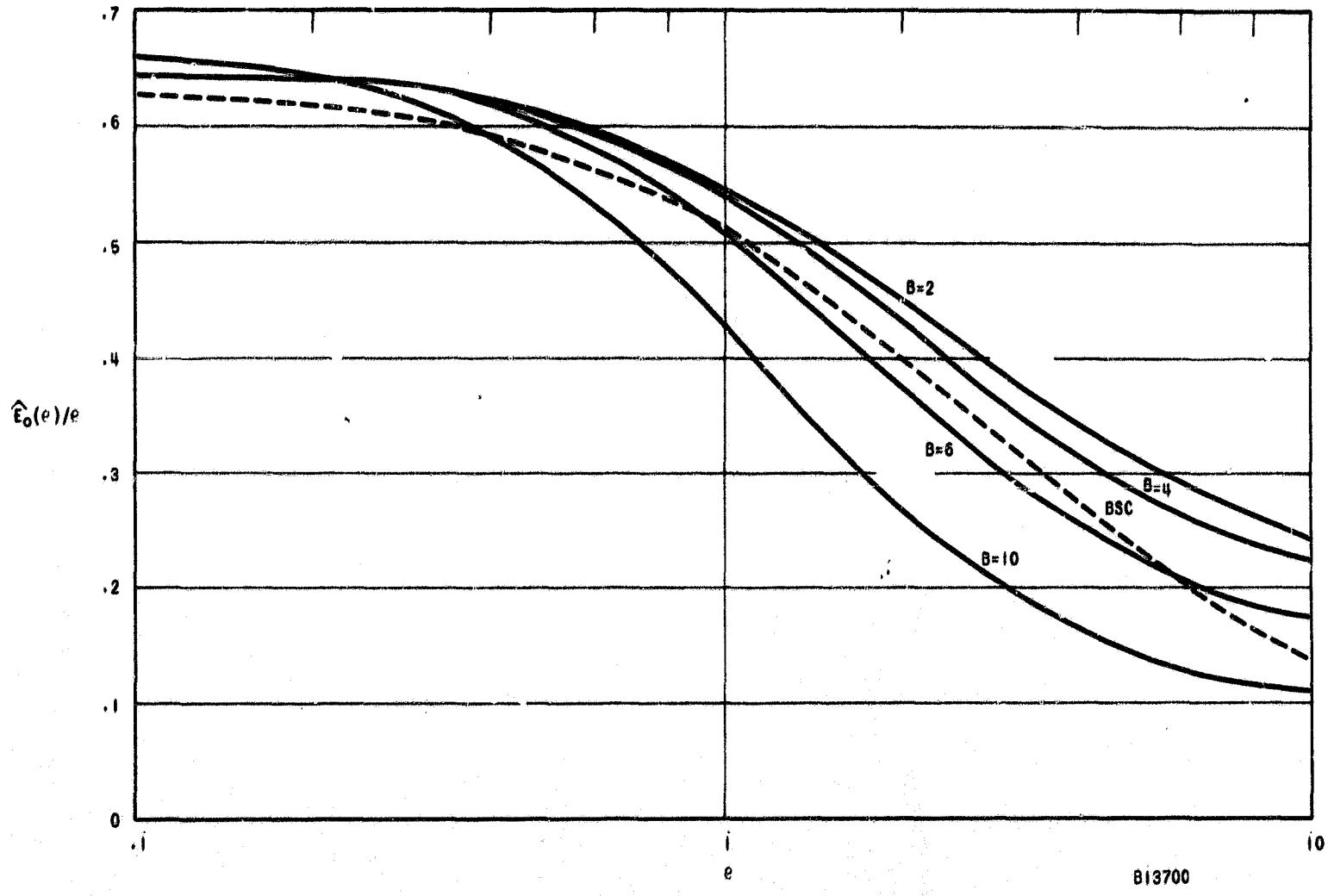


FIGURE 6. Theoretical Pareto Exponents for the Discrete Memoryless Approximation to the Burst Channel with Average Probability of Error = 0.01.

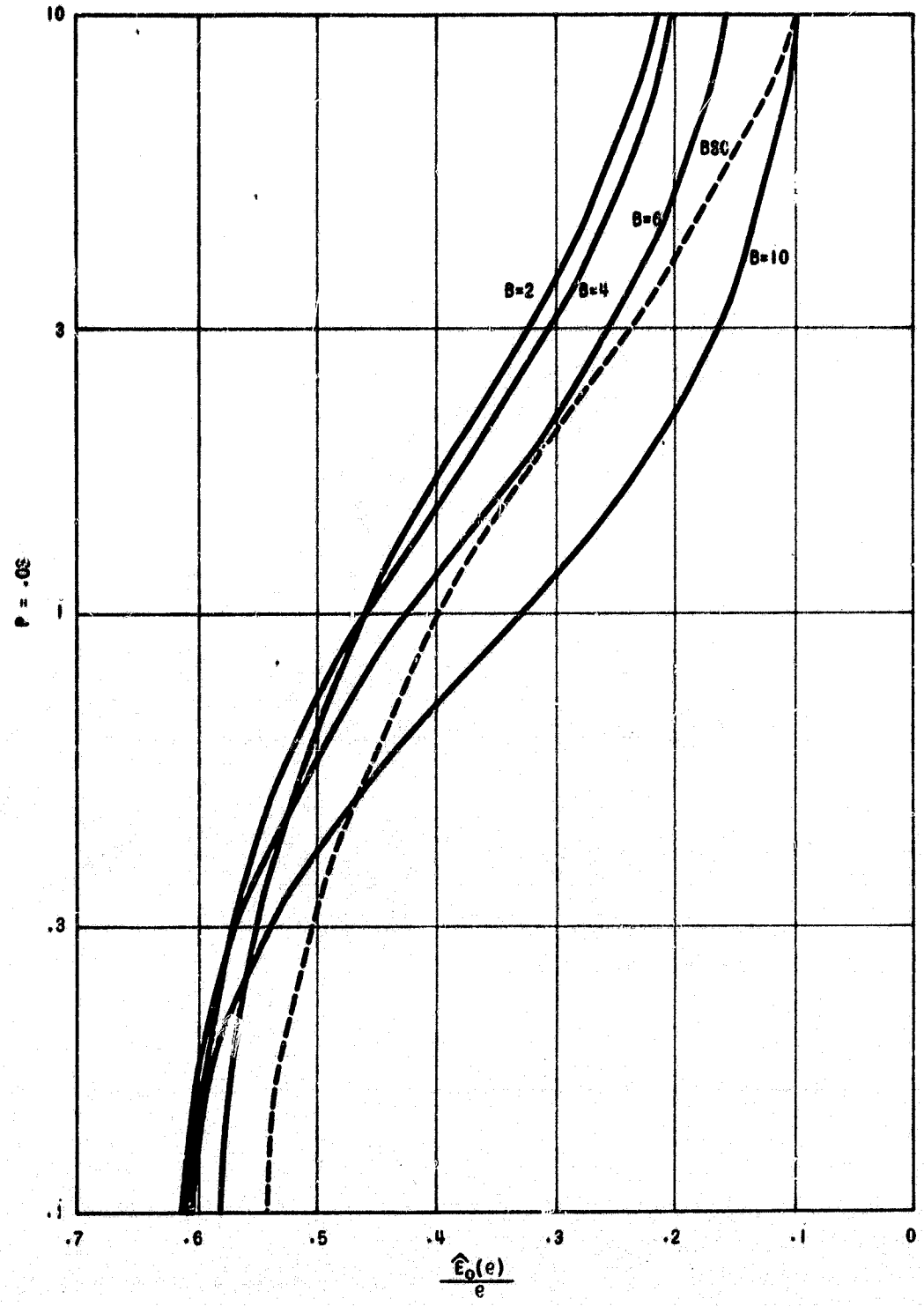


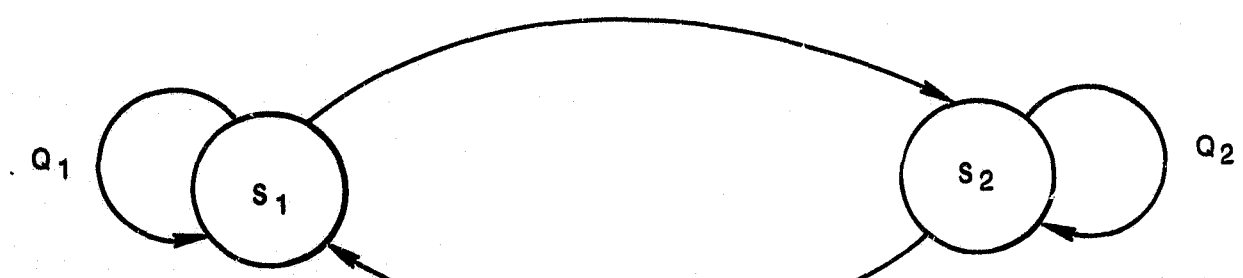
FIGURE 7. Theoretical Pareto Exponent for the Discrete Memoryless Approximation to the Brust Channel with Average Probability of Error = 0.03

In addition to calculating the more general bound parameter $E_o(\rho)$, calculations of the parameter $R_{comp} = E_o(1)$ were performed. Results are shown in Figure 8 in which the ratio of R_{comp} for the burst channel to that for the corresponding binary symmetric channel is plotted as a function of the average probability of symbol error.

Figures 6 and 7 reveal several interesting phenomena. For short bursts, as defined by this particular model, the Pareto exponent is greater than that of the corresponding binary symmetric channel. For example, consider bursts of length four and rate one-half bits per X symbol code (= 0.347 nats | symbol). For the binary symmetric channel, $\rho \approx 1.4$, whereas for the channel with $B = 4$, $\rho \approx 2.3$. This indicates that, in principle, the probability of exceeding some specified large number of computations per decoded digit on the short burst channel should be less than that for the corresponding binary symmetric channel. This result is in contrast to the actual simulation results which were described in the previous section.

Another method for considering the case of sequential decoding on channels with memory is to introduce dependence by a Markov model and to review the simplest such model which was first introduced by Savage (Reference 5). It must be recognized at the outset that this particular model perhaps bears less resemblance to a physical channel than the discrete memoryless approximation. However, it is important to consider the Markov approach for two major reasons. First, the method suggested here may be subsequently generalized and extended to more representative channels. Secondly, this approach, while completely different from the discrete memoryless approximation, tends to substantiate the conclusions derived from that approach.

A representation of the two state, first order Markov process is shown below. The two states are defined as follows:



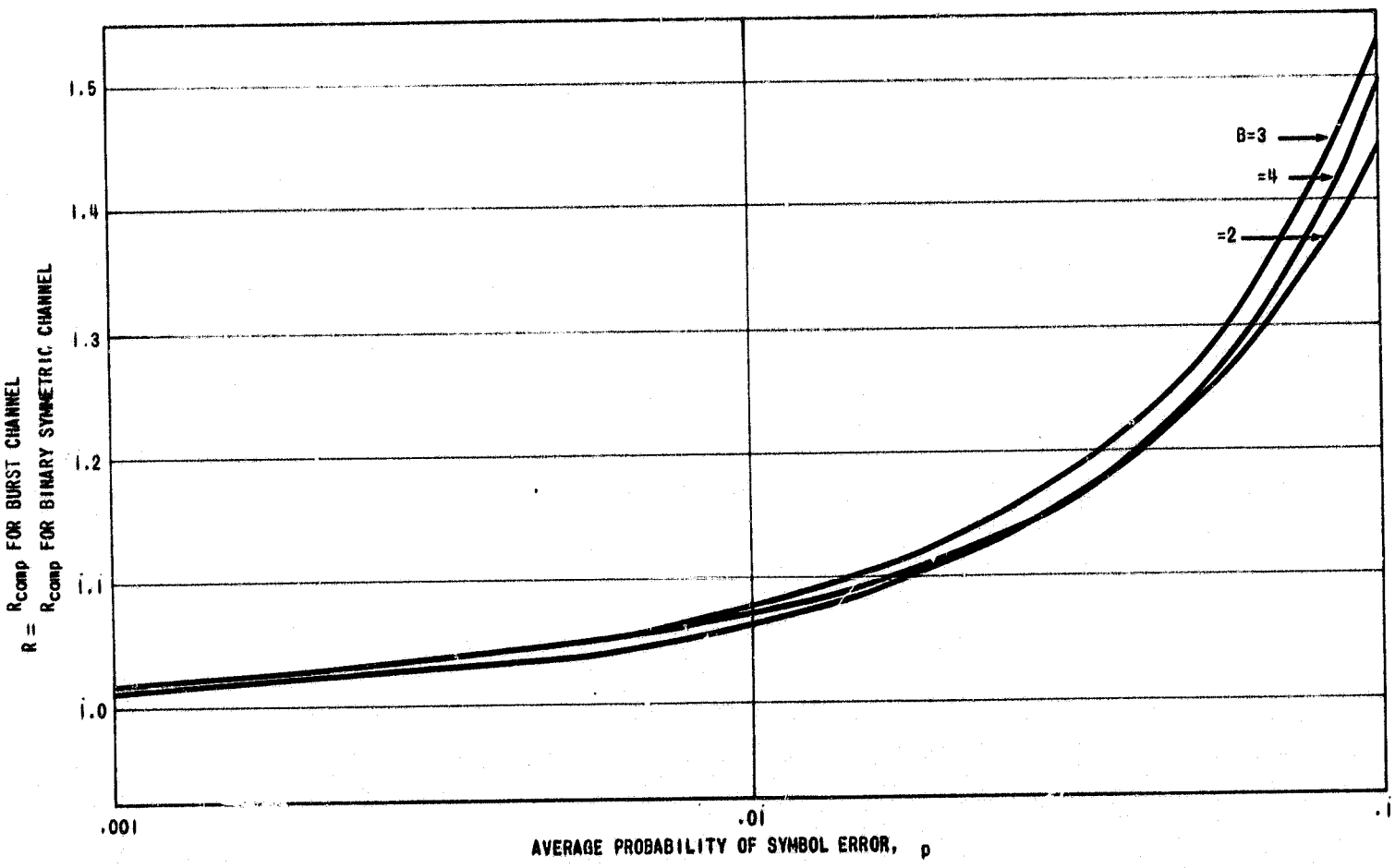


FIGURE 8. R_{comp} for the Discrete Memoryless Approximation to the Burst Channel

S_1 is the noise free or errorless state (i.e., binary digits are received correctly) and S_2 is the error state (binary digits are received incorrectly). P_1, P_2, Q_1, and Q_2 are the state transition probabilities. The stationary absolute probabilities may be shown to be:

$$P(S_1) = \frac{P_2}{P_1 + P_2} \tag{4}$$

$$P(S_2) = \frac{P_1}{P_1 + P_2}$$

For the Markov sequence, it may be shown, Appendix II that for arbitrarily large block length:

$$R_{comp} = 1 - 2 \log_2 \left\{ \frac{\sqrt{Q_1} + \sqrt{Q_2} + \sqrt{Q_1 + Q_2 - 2\sqrt{Q_1 Q_2} + 4\sqrt{p_1 p_2}}}{2} \right\} \tag{5}$$

This result agrees with that stated by Savage. Using the definition that the average probability of symbol error, p, is identical to the stationary probability of being in the error state $[p(S_1)]$, R_{comp} may be computed as a function of p. The results are plotted in Figure 9.

Consider the results shown in Figure 9. For a fixed transition probability from the noiseless state to the error state, p_1, as p increases, the error bursts tend to become longer. This follows from the unique relationship among $p(S_2) = p$, p_1, and p_2 as shown in Equation (4). Notice, in general, R_{comp} tends to increase as p becomes larger (i.e., longer bursts). Again R_{comp} is observed to exceed that for the corresponding binary symmetric channel over an important range. A note of caution should be inserted at this point. The bursts of this model are different from those in the previous model in that digits are always received in error during a burst as opposed to being in error with probability one half. Hence, any physical interpretation of the curves for large p becomes somewhat obscure.

4.4.3 Discussion of Results

a. Burst errors of very short length (i.e., on the order of two to six) cause a serious increase in the computational load required of a sequential decoder. For example, using a rate one half code and an average symbol error probability of 0.01, the probability of exceeding about 2.2 computations per bit is only 0.01 on a binary symmetric channel; on the corresponding channel of burst length four, the probability of exceeding 100 computations per bit is in excess of 0.01.

b. Theoretical considerations lead to the conclusion that for both the hypothesized burst channel and the binary symmetric channel, the computational behavior is bounded by $P_r [C > L] \sim L^{-\rho}$. Further, the Pareto exponent ρ, for the burst channel, was shown to be greater than that for the corresponding random error channel. This leads to the conclusion that, in principle, sequential decoding over short bursts may not be as serious a problem as the simulation results demonstrate, if appropriate modifications were to be made.

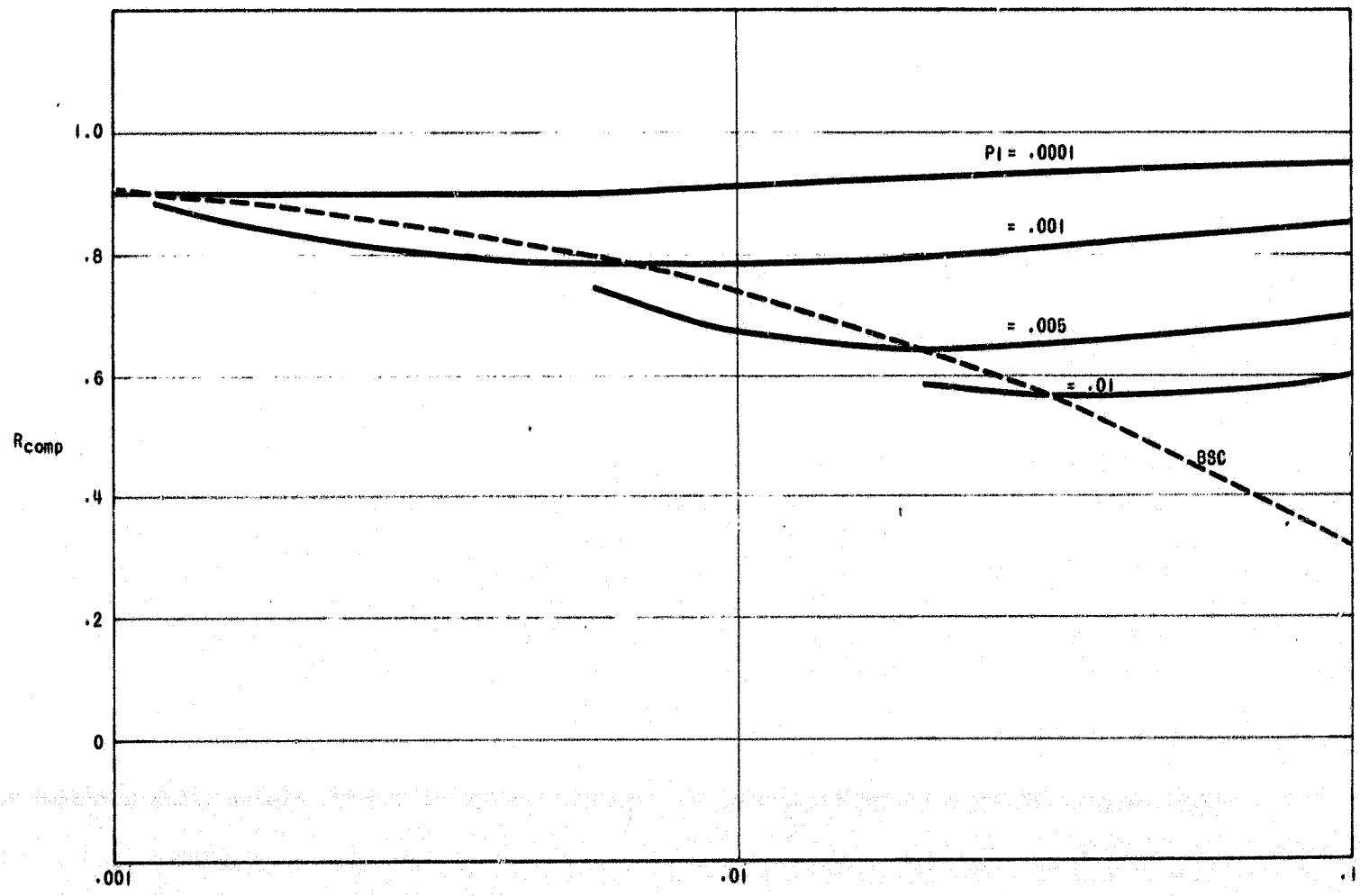


FIGURE 9. R_{comp} for the Markov Approximation to the Burst Channel

c. The simulation results were intentionally performed with parameters appropriate to a random error channel. That is, the likelihood computation, code generating polynomial, and Fano algorithm implementation all presupposed a random channel. The modifications referred to in (b) would encompass changes in one or possibly all of these factors. As presently implemented the sequential decoder in no way uses the information that errors tend to cluster if a burst channel is postulated.

d. The consideration of theoretical bounds and practical implementations, appropriate to sequential decoding, on channels with memory represents an area about which relatively little is known. The models used here are acknowledged to be simple, but do provide certain insight into the more general problem.

e. The analytical results presented tend to lead to the conclusion that scrambling is not the best approach to burst channel decoding. Additional theoretical effort is required as discussed below to substantiate this.

4.4.4 Recommendations

a. For channels on which short burst errors may occur, it is imperative that this be taken into account in the sequential decoder design. Tradeoff studies should be made among the various possible corrective measures which could be taken. Such measures might include;

1. Scrambling of the transmitted sequence.
2. Incorporation of burst error correcting codes as part of the total channel coding problem
3. Exploiting the nature of the error dependence within the decoding algorithm.

b. Additional theoretical work is required to develop better, and more meaningful, representations of channels with memory. This work would have the objective of providing direction for the most efficient practical implementation of sequential decoding on channels with memory.

Section 5

SEQUENTIAL DECODING FOR SPACE MISSIONS

5.1 REQUIREMENTS

In order to define the communications requirements for any particular mission, specific values of data rate, transmitter power, antenna gains, orbital parameters, required error performance, and other system information must be available. Since much of the required information is not yet available for the anticipated missions, the following subsections deal primarily with the important coding system parameters that should be considered in designing a telemetry system. Emphasis is placed upon the impact of various system parameters in so far as they influence the performance of sequential decoders.

The general philosophy, which led to some of the results presented in this section, is based upon recognition of the fact that the primary purpose of using sequential decoding in the near future, for near earth missions, is to prove feasibility as to error correction capability and to flight qualify the required encoding circuitry. Logically then, one should use relatively long codes (relatively complex encoder) for very good error performance which, at the same time, places a relatively large burden upon the encoder reliability in its spaceborne environment. Later near earth missions will probably be oriented towards obtaining larger quantities of data with acceptable rather than near-zero error rates. To accomplish this with a minimum spacecraft effective radiated power (lower or nonexistant margin) and/or minimum RF bandwidth will require that closer attention be paid to some of the "secondary" factors that are involved in designing a system with sequential decoding; e.g., number of demodulator quantization levels and their spacings, improved carrier modulation (converting to $\pm \pi$ rather than $\pm$ 1 radian PSK,) improved carrier tracking techniques, and methods for protecting against clustered errors. It is expected that the acceptable error rate will remain rather low (10^{-5} or less) due to the possible future use of on-board computers to perform data reduction such that only non-redundant data is available to the encoder prior to transmission.

For deep space missions, careful attention must be given to all aspects of the system design. In addition to the factors mentioned above, code rates less than 1/2 should be considered as a means of reducing the required bit energy by a few tenths of a db, and some care should be exercised in using highly (short-term) stable oscillators in order to more fully realize the potential advantage of antipodal signals.

In the following subsections, general consideration is given to modulation, encoding, and decoding. Where applicable, specific recommendations are provided for both near earth and deep space missions based upon the broader philosophy of requirements discussed above.

5.2 MODULATION

Binary phase reversal PSK is suggested as the most desirable modulation technique for use with sequential decoding of space telemetry signals. A number of considerations that support this choice are briefly discussed below.

a. Compatibility with convolutional encoder - since the output of a convolutional encoder is a sequence of binary symbols, the transmission of some M-ary alphabet would require a binary-to-M-ary conversion. This additional spaceborne system complexity is circumvented through the use of a direct binary modulation of the carrier by the encoder output.

b. Relatively high communication efficiency - The restriction of transmitted waveforms to binary causes a negligible increase in the required E_b/N_o when operating at low values of E_s/N_o (energy per symbol to noise density ratio). Justification for this statement (valid for E_s/N_o less than 2 or 3 dB) can be found in Figure 5.18 of Wozencraft and Jacobs (Reference 9).

c. Minimum E_b - Use of antipodal rather than orthogonal binary transmission reduces the required received bit energy by 3 dB. As discussed in paragraph 5.3.2, practical carrier tracking techniques permit operation close to that predicted by theory.

5.2.1 Theoretical Performance Limits

It is frequently said that sequential decoding is 3 dB inferior than Shannon's limit. This statement is true only in the limit of an infinite bandwidth channel. In this paragraph, sequential coding of antipodal (unquantized) signals is compared with three theoretical limits:

- A binary input, binary output channel
- A binary input, continuous (unquantized) output channel
- A continuous input, continuous output channel.

In all cases, a channel bandwidth of watts Hz is assumed and, when required, it is assumed that the received carrier phase can be accurately estimated. The only source of disturbance is a flat Gaussion noise with one-sided spectral density N_o watts/Hz.

Binary Input/Binary Output channel

The capacity of a B. S. C. with symbol error probability p is $C = [1 + p \log_2 p + (1-p) \log_2 (1-p)]$ bits per symbol provided the received symbols are independently corrupted. Since 2 TW independent, bandlimited, symbols can be transmitted in time T, $C = 2W [1 + p \log_2 p + (1p) \log_2 (1-p)]$ bits per second. (6)

When transmitting at R=C bps,

$$\frac{E_s}{N_o} = \frac{R}{2W} \cdot \frac{E_b}{N_o} \quad .$$

The probability of symbol error is given by

$$p = \frac{1}{2} \operatorname{erfc} \sqrt{\frac{E_s}{N_o}} = \frac{1}{2} \operatorname{erfc} \sqrt{\frac{R}{2W} \cdot \frac{E_b}{N_o}} \qquad (7)$$

where

$$\operatorname{erfc} Z = 1 - \frac{2}{\sqrt{\pi}} \int_0^Z e^{-t^2}\, dt.$$

Using Equation (6) with R substituted in place of C and Equation (7) substituted in place of p, there results a transcendental equation giving R/W as a function of both R/W and E_b/N_o. Numerical solutions were obtained on a computer and the results are shown in Figure 10, Curve 1. For the infinite bandwidth limit (R/W→O), Appendix IV shows that

$$\frac{E_b}{N_o} = \frac{\pi}{2} \ln 2 = 0.3695 \text{ dB}; \quad W \to \infty. \tag{8}$$

Binary Input/Continuous Output Channel

The mutual information for a continuous output channel can be expressed as

$$I(Y;X) = H(X) - H(X \mid Y) \tag{9}$$

where H(X) is entropy (average uncertainty) of the channel output and H(X | Y) is the entropy of the output after knowing the input. If the channel output is the sum of the input signal plus a statistically independent Gaussian noise, then H(X | Y) is equal to the entropy of the additive noise which is given by

$$H(X \mid Y) = \ln \sqrt{2\pi e \sigma^2} \tag{10}$$

Channel capacity is the maximum value of I(Y;X) when maximized with respect to the input symbol probabilities. Since H (X|Y) is not a function of the input signal statistics, I(Y;X) will maximize when H(X) is maximized. This occurs (because the noise is symetric) when the binary input symbols are chosen with equal probability to be ±1. Then,

$$H(X) = -\int_{-\infty}^{+\infty} p(X) \ln [p(X)]\, dx \tag{11}$$

where

$$p(X) = \frac{1}{2} (2\pi\sigma^2)^{-1/2} \left[\exp\left\{ \frac{-(x-1)^2}{2\sigma^2} \right\} + \exp\left\{ \frac{-(x+1)^2}{2\sigma^2} \right\} \right] \tag{12}$$

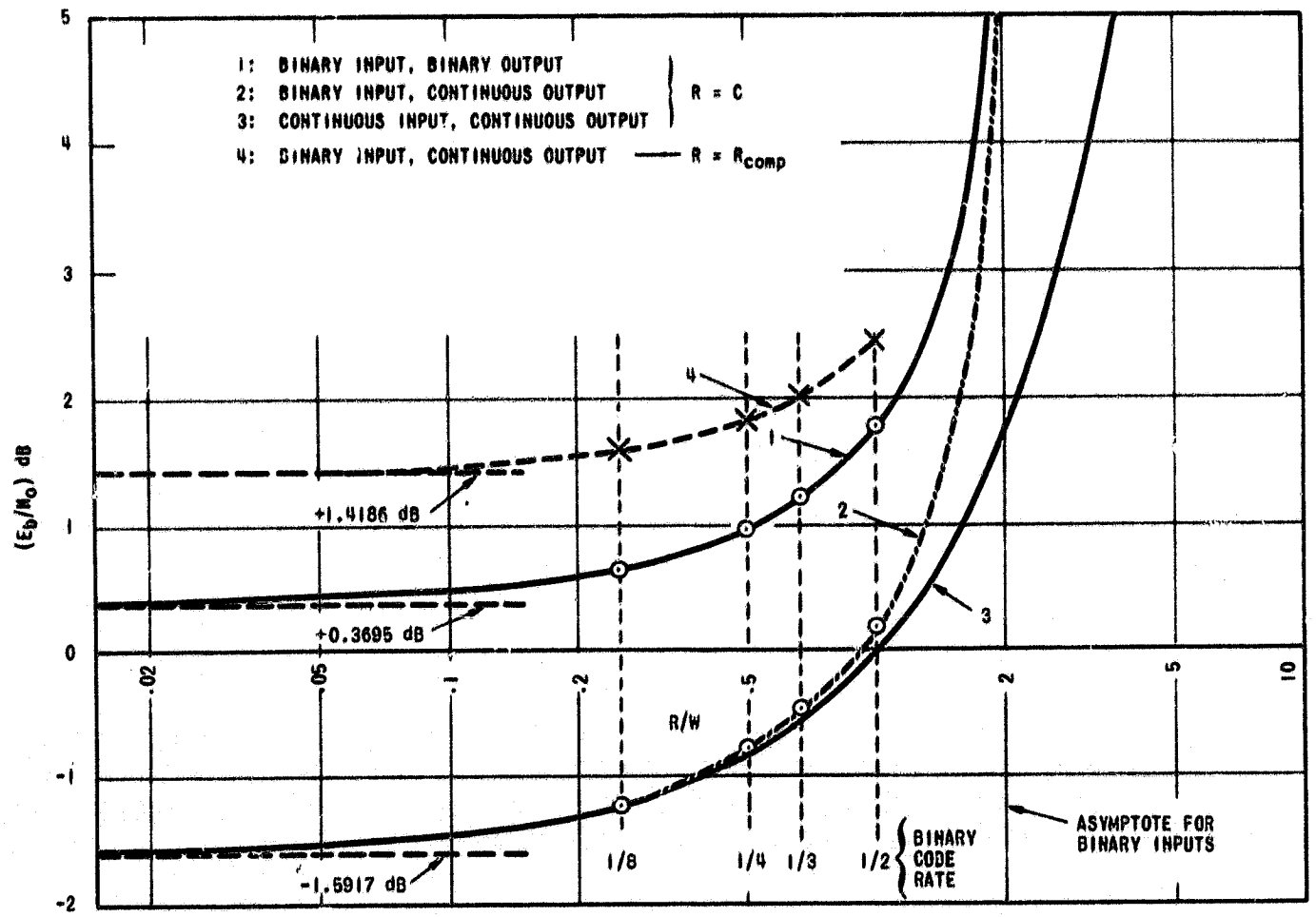


FIGURE 10. Performance Bounds of Ideal Systems Versus Sequential Decoding

and where

$$\frac{1}{\sigma^2} = \frac{E_b}{N_o} \cdot \frac{R}{W} \tag{13}$$

The relation between $\frac{E_b}{N_o}$ and $\frac{R}{W}$ (with R=C) is

$$\frac{R}{W} = \frac{2\, I\,(Y;X)}{\ell n\, 2} \quad \text{b. p. s./Hz} \; . \tag{14}$$

By combining Equations (9) through (14) there results a transcendental equation giving R/W as a function of both R/W and E_b/N_o. Numerical solutions were obtained on a computer and the results are shown in Figure 10 Curve 2.

Continuous Input/Continuous Output Channel

The capacity under an average signal power constraint is given by Shannon's equation.

$$C = W \log_2 (1+P/NoW) \quad \text{bits per second.} \tag{15}$$

where P is the average signal power. Setting C=R and $P=E_bR$, it follows immediately from Equation (15).

$$\frac{R}{W} = \log_2 \left(1+\frac{R}{W} \cdot \frac{E_b}{N_o}\right) . \tag{16}$$

Upon solving for E_b/N_o, one obtains

$$\frac{E_b}{N_o} = \frac{2^{R/W}}{R/W} - 1 \tag{17}$$

This result is shown in Figure 10 Curve 3.

For $R/W << 1$;

$$2^{R/W} \approx 1 + \frac{R}{W} \ell n\, 2 + \frac{1}{2} \left[\frac{R}{W} \ell n\, 2\right]^2$$

and thus, from Equation (17)

$$\frac{E_b}{N_o} \approx \ell n\, 2 \left[1 + \frac{1}{2}\left(\frac{R}{W}\right) (\ell n\, 2)\right] \; ; \; \frac{R}{W} << 1 \; .$$

In the infinite bandwidth limit,

$$\lim_{\left(\frac{R}{W} \to o\right)} \left(\frac{E_b}{N_o}\right) = \ln 2 = -1.5917 \text{ dB.} \tag{18}$$

It should be noted that this is the lowest value of E_b/N_o that could possibly be approached by any errorless communication scheme when operating in the presence of additive white Gaussian noise.

Antipodal Signals Continuous Output/Sequential Decoding

The quantity R_{comp} is that value of data rate in bits per binary symbol above which the average number of sequential decoder computations per decoded bit tends to infinity as the code constraint length is made infinite and below which the average number of computations per bit remains finite. Since the probability of decoded bit error approaches zero as the constraint length is made infinite, R_{comp} can be considered the highest error free signaling rate obtainable with sequential decoding. It is of interest, therefore, to compare R_{comp} with channel capacity. As shown in paragraph 5.4.1, it is possible to calculate R_{comp} with antipodal signals and an unquantized (continuous) demodulator output. The results for this case are also shown on Figure 10, Curve 4. For the infinite bandwidth case, it is seen that sequential decoding requires exactly twice the energy per bit than do the ideal systems with a continuous output when operating at capacity (Curves 2 and 3). When considering systems operating at rate 1/2, the following results are obtained:

Sequential Decoding	$E_b/N_o = 2.45$ dB
Ideal Binary Input/Continuous Output	$E_b/N_o = 0.18$ dB
Ideal Continuous Input/Continuous Output	$E_b/N_o = 0.00$ dB

It should be noted that the small (0.18 dB) degradation caused by the restriction of ideal binary transmission over ideal continuous transmission, vanishes at lower code rates.

From these results it should also be noted that sequential decoding is only 2.45-0.18= 2.27 dB inferior to the ideal binary input system when operating at rate 1/2. As discussed earlier, this difference approaches (approximately) 3 dB at very low code rates (the infinite bandwidth case).

In this paragraph, consideration was given only to sequential decoding with an unquantized demodulator output. The indicated performance can be closely approached (within 0.2 dB) with 8-level quantization.

5.2.2 Effects of Noisy Phase Reference

It is known that antipodal waveforms require 3 dB less energy per bit than do orthogonal binary waveforms in order to achieve the same level of performance. In order to realize this 3 dB advantage, however, demodulation of antipodal waveforms must be done coherently with a local oscillator which is precisely phase locked to the received carrier. In practice, such

an ideal reference oscillator is not available. The degradation in performance when sequentially decoding antipodal waveforms with a reference oscillator exhibiting phase jitter with respect to the received carrier has been evaluated in References 10 and 11.

Factors which contribute to phase jitter and design criteria for minimizing the effects of phase jitter are discussed in detail in paragraph 5.3.2 and in Appendix III.

5.3 ENCODING

5.3.1 Systematic and NonSystematic Codes

A systematic code is one in which the information sequence is transmitted unaltered and the parity symbols are obtained from a linear combination of the information sequence. In general, sequential decoding has been studied in conjunction with systematic codes; performance with non-systematic codes is not generally reported in the literature. Two major advantages of systematic codes are often quoted:

a. For cases in which the signal to noise ratio is high, decoding may be accomplished by observing only the information sequence and discarding the parity sequence.

b. For the same length shift register, the systematic code may be simpler to generate.

The purpose of this investigation was to take a somewhat broader view of the problem to determine if any significant advantage might be gained through the use of nonsystematic codes.

The approach used was to take an optimum (in the maximum-minimum distance sense) systematic code generator and derive from this nonsystematic generators with the same distance property. The performance of these codes was then simulated on a digital computer and the results compared. The parameters of interest in the simulations were probability of undetected error and computational behavior. The objective was to determine if the performance of a rate 1/2 systematic code of shift register length L was exceeded in any sense by a rate 1/2 nonsystematic code with identical distance properties having shift register length approximately L/2.

Relatively short shift register lengths were chosen for two reasons. First, the undetected error probability is higher and hence, the number of simulations to achieve reasonably indicative error statistics is reduced. Secondly, the maximum length optimum generator tabulated by Bussgang (Reference 12) has length 16. In addition however, the code of length 32 reported by GFSC was used to derive nonsystematic generators of length 15 and 17 and results were also obtained for these codes.

The following two paragraphs discuss the generation of nonsystematic codes and comparative computer simulations.

Determining Nonsystematic Code Subgenerators

The purpose of this paragraph is to describe an exact method of selecting nonsystematic code subgenerators requiring shorter shift registers than equivalent systematic code generators. Complete details are carried out for two rate 1/2 codes of constraint lengths 13 and 16, which are best generators (in the maximum-minimum distance sence) determined by the exhaustive search procedure given by Bussgang (Reference 12).

Let (I, P) be a canonical generator of rate 1/2, where P is an odd N-tuple. The equivalent nonsystematic generator may be constructed in the following manner: Let (Q, R) denote the equivalent nonsystematic code, where Q and R = Q*P are odd subgenerators. Then, the entries q_i (q_1=1) are determined by the system of equations.

$$\begin{bmatrix} p_{N_e} & p_{N_e-1} & \cdots & p_3 & p_2 \\ p_{N_e+1} & p_{N_e} & \cdots & p_4 & p_3 \\ p_{N_e+2} & p_{N_e+1} & \cdots & p_5 & p_4 \\ p_{N-1} & p_{N-2} & \cdots & p_{N-N_e+2} & p_{N-N_e+1} \end{bmatrix} \begin{bmatrix} q_2 \\ q_3 \\ q_4 \\ q_{N_e} \end{bmatrix} = \begin{bmatrix} p_{N_e+1} \\ p_{N_e+2} \\ p_{N_e+3} \\ p_N \end{bmatrix}$$

For N odd, $N_e = (N+1)/2$, and the above matrix is square. A unique set of q's may be found if the determinant of the matrix is not zero. For N even, N_e may be reduced further by taking one more equation than there are unknowns, provided that the determinant of the augmented matrix is zero. In this case $N_e = N/2$.

Optimum systematic generators have been found by Bussgang, using computer methods, up to N=16. The calculations leading to the equivalent nonsystematic generators of reduced coding constraint length N_e=8 are reproduced below.

Consider, first, the case (I, P)

I = (1, 0, . . . , 0)

P = (1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0)

The augmented matrix of the system of equations is:

$$\begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Adding the first row to the third and sixth, the determinant of this matrix may be written

$$\begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Erasing the first row and column, and adding the first row of the resulting seven-by-seven determinant to the third and sixth rows respectively, we obtain the determinant,

$$\begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Again, erasing the first row and column, and adding the last row of the resulting six-by-six determinant to the third and fourth rows respectively, we obtain

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

This process is repeated, yielding successively,

$$\begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix}, \quad \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}, \quad \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

Since the first two rows of the final determinant are zero, the determinant of the original augmented matrix is zero.

The system of equations may be written,

$$\begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \\ q_7 \\ q_8 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad (20)$$

or,

$$\begin{array}{llllllll} 1) & q_2 + & +q_4 & +q_5 & +q_7 & +q_8 & = & 0 \\ 2) & q_3 & +q_5 & +q_6 & & +q_8 & = & 1 \\ 3) & q_2 & +q_4 & & +q_6 & +q_7 & = & 0 \\ 4) & q_2 & & +q_5 & +q_7 & +q_8 & = & 0 \\ 5) & & q_4 & & +q_6 & +q_8 & = & 1 \end{array} \qquad (21)$$

6) $q_2 + q_5 + q_7 = 0$

7) $q_3 + q_6 + q_6 = 0$

8) $q_4 + q_7 = 0$

Equations (1), (4), and (8) $\longrightarrow q_4 = 0$

$\therefore q_7 = 0$ (Equation 8)

4) and 6) $\longrightarrow q_8 = 0$

$\therefore q_2 = q_3 = q_5 = q_6 = 1$

$\therefore Q = (1,1,1,0,1,1,0,0,0....0)$

R may be found from the convolution equation

$$r_j = \sum_{m=1}^{j} p_m q_{j-m+1} \qquad (j=1,...8)$$

$(r_1=1)$

This gives,

$R = (1,0,1,0,1,001,0...0)$

Summarizing,

$I = (1,0,.....0)$

$P = (1,1,1,0,1,1,0,1,0,1,0,0,1,0,0,0)$ } Simulation Code #5

$Q = (1,1,1,0,1,1,0,0,.....0)$

$R = (1,0,1,0,1,0,0,1,0,.....0)$ } Simulation Code #1

The nonsystematic code (Q, R) is equivalent to the systematic code (I, P).

Similarly, for Bussgang's 'best' generator (I, P), of length 16, where

$P = (1,1,0,1,0,1,0,1,1,0,0,1,0,1,1,1)$

the augmented matrix may be written,

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$

The following sequence of determinants is obtained, as before, by row operations and expansion of each determinant by the elements of the column containing the only 1.

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}, \quad \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix},$$

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix},$$

Since the final determinant is zero, the determinant of the augmented matrix is also zero.

The system of equations may be written,

$$1)\quad q_2 + q_4 + q_6 + q_8 = 1$$

$$2)\quad q_2 + q_3 + q_5 + q_7 = 0$$

$$3)\quad q_3 + q_4 + q_6 + q_8 = 0$$

$$4)\quad q_4 + q_5 + q_7 = 1 \qquad (23)$$

$$5)\quad q_2 + q_5 + q_6 + q_8 = 0$$

$$6)\quad q_3 + q_6 + q_7 = 1$$

$$7)\quad q_2 + q_4 + q_7 + q_8 = 1$$

$$8)\quad q_2 + q_3 + q_5 + q_8 = 1$$

$q_4 + q_5 = 1$ from 1) and 5)

$q_7 = 0$ from 4)

$q_8 = 1$ 2) and 8)

$q_2 + q_4 = 0$ 7)

$q_6 = 0$ 1)

$q_3 = 1$ 6)

$q_4 = 0$ 3)

$q_2 = 1$ 1)

$q_5 = 1$ (any equation involving q_5)

$\therefore Q = (1,0,1,0,1,0,0,1,0,\ldots\ldots 0)$

From the convolution equation

$$R = (1,1,1,0,0,1,1,0,0,\ldots\ldots 0)$$

Summarizing,

$$I = (1,0,\ldots\ldots\ldots 0)$$
$$P = (1,1,0,1,0,1,0,1,1,0,0,1,0,1,1,1) \quad \left.\right\} \text{ Simulation Code \#6}$$

$$Q = (1,0,1,0,1,0,0,1,0,\ldots\ldots 0)$$
$$R = (1,1,1,0,0,1,1,0,0,\ldots\ldots 0) \quad \left.\right\} \text{ Simulation Code \#2}$$

Computer Simulation of Systematic and NonSystematic Codes

Eight distinct codes were simulated on the IBM 1130. A binary symmetric channel was used with transition probabilities of 0.03 and 0.04. For rate 1/2 codes these correspond to ratios of R/R_{comp} of 0.867 and 0.956 respectively. $5(10)^4$ bits were decoded for each case and no deletions were observed.

For convenience, the codes are summarized in Table 1. Codes 1 and 2 represent the nonsystematic codes derived from codes 5 and 6 respectively. (As noted in the preceding paragraph, codes 5 and 6 are the optimum generators of length 16 determined by Bussgang. Codes 3 and 4 are the optimum systematic generators of length 8 reported by Bussgang. Finally, code 7 is the nonsystematic code of equivalent maximum-minimum distance derived from the length 32 systematic code shown as code 8.

The results of the simulations are presented in Table 2 and are discussed below. In all cases, identical noise sequences were used to provide a better comparison.

TABLE 1. Systematic and Non-Systematic Codes Simulated

Code #	Structure
1	G_1 = 1 1 1 0 1 1 G_2 = 1 0 1 0 1 0 0 1
2	G_1 = 1 1 1 0 0 1 1 G_2 = 1 0 1 0 1 0 0 1
3	G = 1 1 0 1 1 1
4	G = 1 1 1 0 0 1 1
5	G = 1 1 1 0 1 1 0 1 0 1 0 0 1
6	G = 1 1 0 1 0 1 0 1 1 0 0 1 0 1 1 1
7	G_1 = 1 1 0 1 1 0 0 0 0 0 1 1 0 0 1 G_2 = 1 0 1 0 1 0 0 0 1 0 0 1 1 0 1 1 1
8	G = 1 1 0 1 0 1 0 1 1 0 0 1 0 1 1 1 0 0 0 1 1 1 0 1 1 1 1 0 1 1 0 1

TABLE 2. Simulation Results

			$P_s = 0.03$		$P_s = 0.04$	
			$\alpha = 0.867$		$\alpha = 0.956$	
Code #	TYPE	L_1, L_2	$\overline{B}$	P_e	$\overline{B}$	P_e
1	N	6,8	2.26	2.0(-4)	3.14	9.8(-4)
2	N	7,8	2.36	1.0(-4)	3.52	5.2(-4)
3	S	1,6	1.99	2.3(-3)	2.58	5.3(-3)
4	S	1,7	2.00	1.5(-3)	2.64	3.6(-3)
5	S	1,13	2.17	4.8(-4)	3.06	1.4(-3)
6	S	1,16	2.30	1.0(-4)	3.31	9.8(-4)
7	N	15,17	2.33	N.A.	3.66	N.A.
8	S	1,32	2.33	N.A.	3.66	1.0(-4)

N Non Systematic
S Systematic
L_1, L_2 Generator lengths
$\overline{B}$ Average number of computations per decoded information bit
P_s Binary symmetric channel transition probability
P_e Undetected error probability
α R/R_{comp}
N.A. No errors observed in $5(10)^4$ decoded bits

Discussion of Results

Perhaps the most significant observation from these results may be obtained by comparing nonsystematic codes 1 and 2 which were derived from systematic codes 5 and 6. In general, the nonsystematic codes (with shift register length about half of that of the systematic code) showed a better undetected error performance than the systematic codes from which they were derived. The average computational load for the nonsystematic code was slightly higher than that for the systematic codes - indicating that the nonsystematic code essentially "works harder" but does a superior job of correcting errors. It might be conjectured that the increased structure of a rate 1/2 nonsystematic code limits the undetected errors to a lower value than that possible with the systematic code. Additional simulations as well as theoretical work would be required to substantiate this.*

This same trend, although on much weaker statistical grounds, was obtained for the nonsystematic code derived from a length 32 systematic code. Whereas no errors were observed for the former, errors were observed for the latter (The same noise sequence was used for both simulations).

*A recent investigation by Codex Corporation (High-Speed Sequential Decoder Study, Contract DAAB07- 68-C-0093, 15 April 1968) was received after this work was performed. The work tends to verify the possible advantage of nonsystematic codes.

Finally, for the same shift register length, the nonsystematic code always showed superior error rate performance. This result may not be too surprising, but the magnitude of the observed improvement may be of interest.

For certain missions, then, the choice of systematic vs. nonsystematic code may be determined by encoder implementation considerations. For example, it is pertinent to contrast the reliability of a shift register of length L with one modula 2 adder to that of length L/2 with two modulo 2 adders.

In summary, it appears that nonsystematic codes offer some attractive features, especially at low redundancies (e.g., rate 1/2.) C&S suggests that some further investigation be undertaken to confirm the ideas offered in this paragraph to further explore the properties and implementation of nonsystematic codes.

5.3.2 Code Rate

When using a convolutional code in which ν binary digits are transmitted for each data bit, the energy per data bit which is ideally required for successful operation diminishes as ν is increased. With antipodal transmission, a phase coherent reference oscillator is required for demodulation. Due to imperfect oscillator stabilities, additive received thermal noise and the presence of binary modulation on the received carrier, the accuracy with which the carrier phase can be estimated diminishes as ν is increased. Thus, in practice, the demodulating reference oscillator will exhibit more phase jitter with respect to the received carrier phase for larger values of ν. The effect of phase jitter is to increase the required bit energy above that which would be required if a perfectly coherent reference oscillator were available.

Choice of a "best" code rate ($1/\nu$), therefore, is based upon trading off the reduced bit energy required of highly redundant codes (large ν) against the increase in bit energy which is required to overcome the effects of phase jitter. For a specified short term oscillator stability, S_T, carrier frequency, f_o, and transmitted data rate, R bits per second, one can define an oscillator stability factor $K = S_T (f_o/R)$. It will be shown that there exists an "optimum" value of ν for a specified value of K in the sense that the required energy per data bit to noise density ratio is minimized. Results are obtained for two types of carrier tracking circuits; theoretical optimum, and a second order loop with "Costas" carrier reinsertion. It is assumed that $R = R_{comp}$ where R_{comp} is the sequential decoder computational cutoff rate. It is further assumed that the demodulator output is unquantized. If 8-level quantization were used, the calculated values of E_b/N_o presented here would have to be increased by about 0.2 dB.

The criterion of minimizing required bit energy is considered to be the most meaningful for relatively low data rate and relatively low signal level situations representative of deep space telemetry systems.

Interrelationships Among E_b/N_o, R_{comp}, Loop SNR, and RMS Phase Error.

The probability density function of the phase error θ has been derived by Viterbi (Reference 10) and applied by Tenkhoff (Reference 11) to obtain R_{comp} for a memoryless*

---

* To realize a memoryless channel in practice might require symbol interleaving.

channel using antipodal modulation and a continuous output. Results were presented as required E_b/N_o (when operating at $R = R_{comp}$) versus R_{comp} with the loop S/N as a parameter. As shown by Viterbi,

$$\sigma_\theta^2 = \frac{1}{S/N} \tag{24}$$

where σ_θ^2 is the rms phase error and S/N is the signal to (one-sided) noise power in the carrier-tracking loop.

Degradation of Loop Performance Caused by Binary Modulation.

It is possible to track the carrier phase by adding an unmodulated carrier to the transmitted signal. This approach reduces the amount of energy available for the modulated signal, however, and it has been found advantageous (Reference 13) to utilize the modulated signal directly for extracting carrier phase information. Frenkel, Appendix III, has shown that loops designed to track the carrier on a binary antipodal signal are equivalent to a phase-locked loop which tracks an unmodulated, but phase disturbed, carrier which is imbedded in white noise enhanced by a factor δ. Results which show δ versus E_s/N_o (energy per received symbol to one-sided noise density ratio) for various types of tracking circuits are shown in Appendix III. Note that

$$(E_s/N_o) = (1/\upsilon)(E_b/N_o) \tag{25}$$

Frenkel also shows that

$$\sigma_\theta^2 \approx 0.68\ S^{2/3}\ \text{rad.}^2 \text{ - optimum tracking filter} \tag{26}$$

$$\sigma_\theta^2 \approx 1.46\ S^{2/3}\ \text{rad.}^2 \text{ - best second order tracking filter} \tag{27}$$

where

$$S = \frac{2\pi f_o N_o \delta S_T}{P} \tag{28}$$

with P used to denote the total received signal power.

Method of Analysis

A sample calculation is performed here to illustrate the manner in which the final results were determined. Consider the use of second order "Costas" loop with $\upsilon = 4$. Since $R = 1/\upsilon$ bits per symbol, $R = R_{comp}$ at $\upsilon = 4$ provides $R_{comp} = 1/4$. Enter Figure 3 in Reference 11 at $R_{comp} = 0.25$. Using the curve labeled S/N = 10 dB, read $E_b/N_o = 2.3$ dB. From Equation 24 is seen that S/N = 10 = 10 dB corresponds to $\sigma\theta^2 = 0.1$ radian^2. From Equation 27, calculate S = 0.0179, and from Equation (25) calculate $E_s/N_o = -3.7$ dB. From Figure III-1, Appendix III read $\delta = 3.4$ dB for the "Costas" circuit. Now, define a normalized oscillator stability factor

$$K = S_T(f_o/R) = \frac{S}{2\pi\delta}\left(\frac{E_b}{N_o}\right) \quad (29)$$

Numerical values have previously been obtained for S, δ, and E_b/N_o so one can calculate K = 0.0022 from the right hand side of Equation 29. This procedure can be repeated by choosing other values of S/N on Figure 3 in Reference 11 to obtain a curve of E_b/N_o versus K for $\upsilon = 4$. Similarly, a family of curves for different values of υ can be obtained. Having obtained a family of curves of E_b/N_o versus υ with K as a parameter, one can determine E_b/N_o versus υ with K as a parameter. The final results of this procedure are illustrated and discussed below.

The results for a second order loop with a "Costas" carrier reinsertion circuit are illustrated in Figure 11 as E_b/N_o versus ν with several different values of oscillator stability factor, K. It can be seen that a "flicker-noise-free" system (characterized by K=O) permits sequential decoding limit of E_b/N_o=1.419 dB to be approached as ν is made large. With increasing values of K, the required E_b/N_o is seen the increase. Furthermore, with any non-zero value of K, there exist an optimum value of ν for which the required E_b/N_o is minimum for a chosen value of K.

Similar results are illustrated in Figure 12 for an "optimum" loop which contains a Weiner filter that minimizes σ_θ^2 and an "optimum" carrier reinsertion circuit which minimizes δ for a given E_s/N_o. The major difference between the results shown in Figures 11 and 12 is that the "optimum" tracking configuration can tolerate a value of S which is 3.15 times as large than that required of the second order "Costas" configuration to achieve a specified value of E_b/N_o at the same value of ν. A less noticeable difference is also present. Close inspection of Figures 11 and 12 will reveal that if K is fixed, the optimum value of ν, call it ν_o, is larger with the "optimum" configuration than with the "Costas" configuration. This phenomenon is clearly illustrated in Figure 13 where ν_o is plotted as a function of K for both types of tracking circuits.

The minimum attainable E_b/N_o for a specified value of K is shown in Figure 14 for both types of tracking circuits. With K = 0.01, for example, it can be seen that the "optimum" circuit requires E_b/N_o = 2.38 dB and the "Costas" circuit requires E_b/N_o = 3.11 dB. To attain these levels of performance, Figure 13 shows that $\nu_o \approx 5.5$ and 4.6 respectively. Owing to the fact that curves of E_b/N_o versus ν for a fixed K are very flat in the vicinity of their minima, Figure 11 shows that with ν = 3 and 4 with the second order "Costas" configuration, E_b/N_o is increased to 3.20 and 3.12 dB respectively with K = 0.01.

Lowest Useable Data Rate

The presence of "flicker" noise on the transmitted signal imposes a lower limit on the transmitted data rate that can be used efficiently with a phase coherent transmission link. From Equation (29) it can be seen that

$$R = S_T f_o/K$$

A value for short-term oscillator stability S_T cannot be accurately specified (or measured). However, the best quartz crystal oscillators have been found "flicker" limited to at least 3 parts in 10^{13} of instability(14). A lower limit for spaceborn oscillators might be of the order

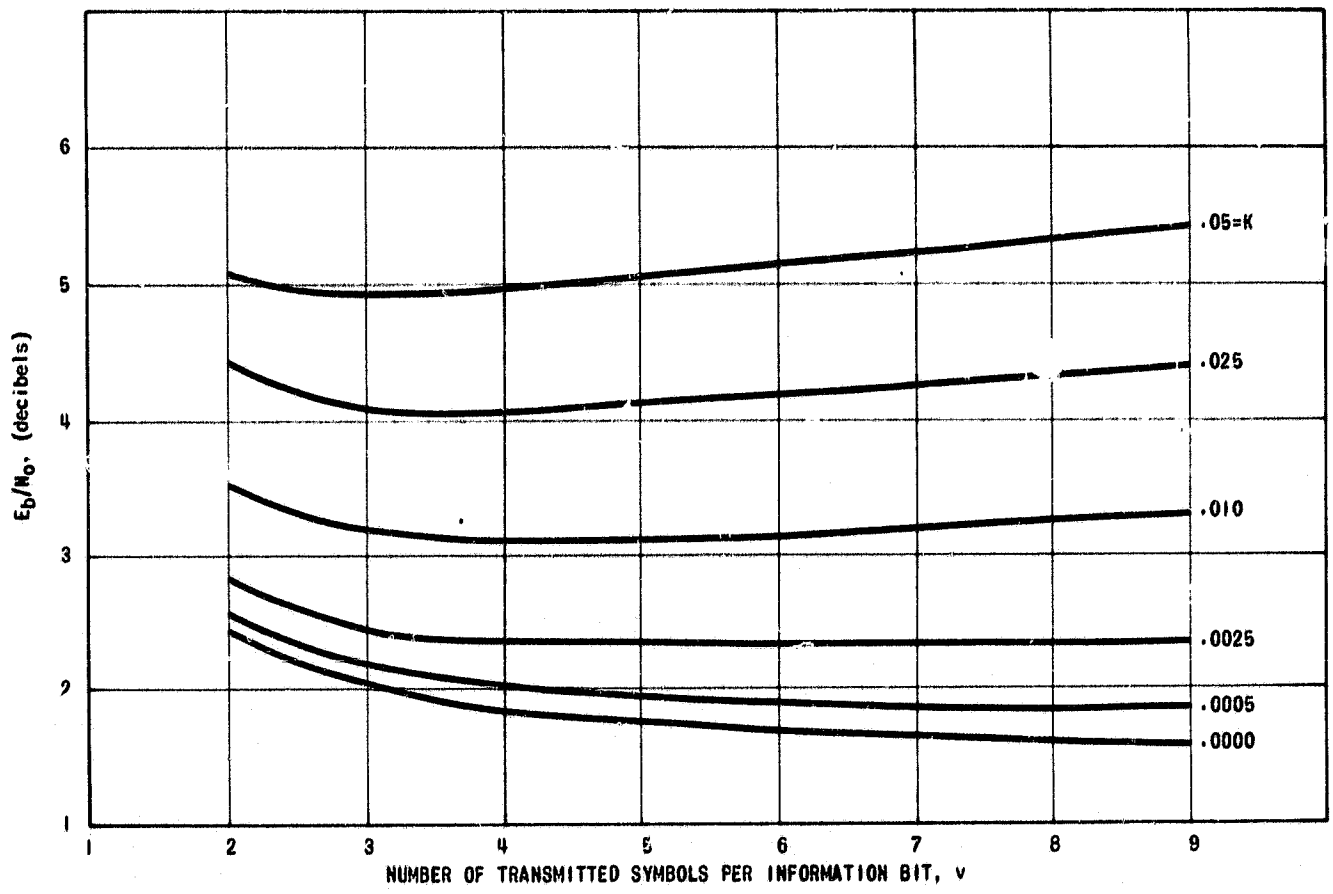


FIGURE 11. E_b/N_o versus ν for Second Order Loop

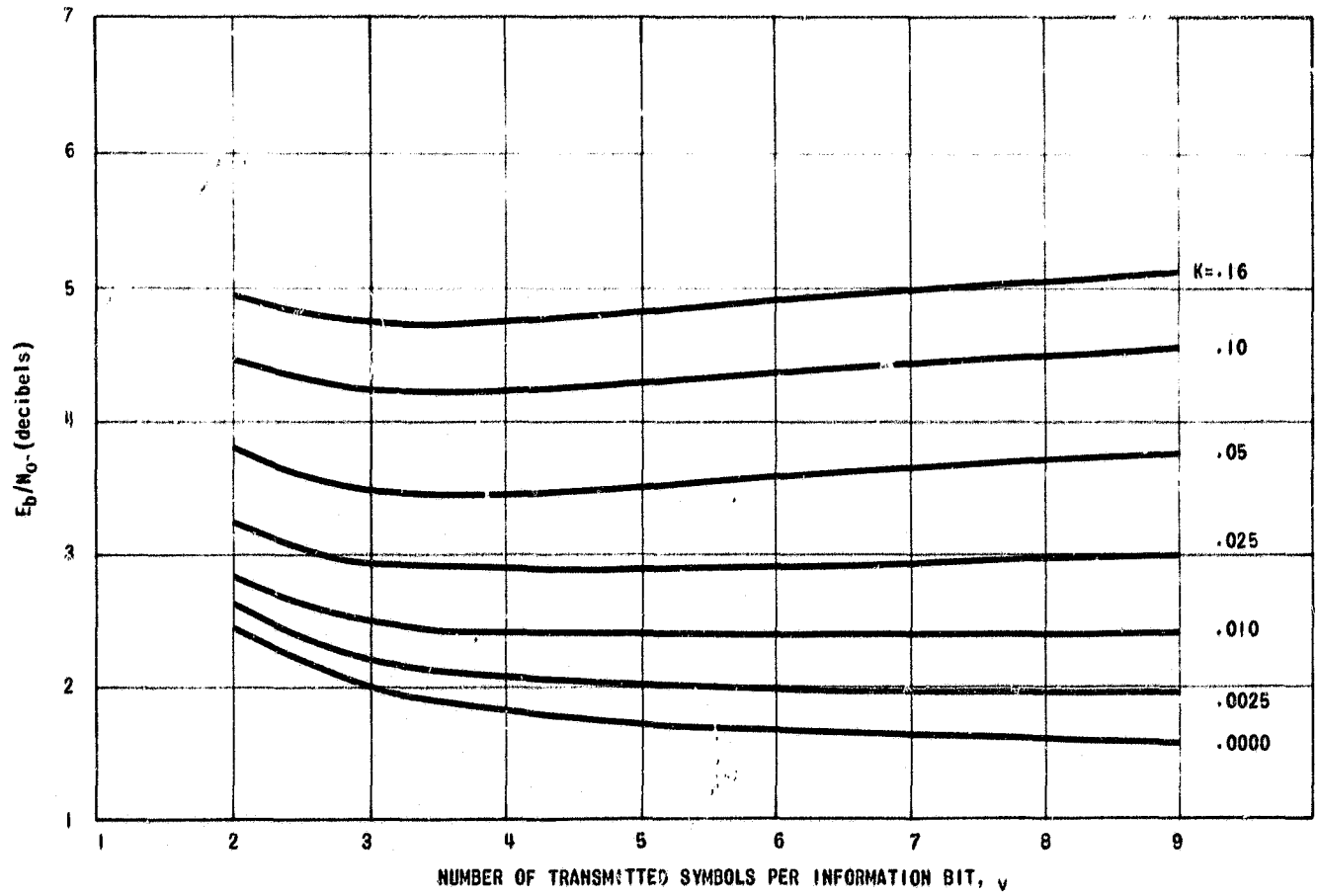


FIGURE 12. E_b/N_o versus ν for Optimum Loop

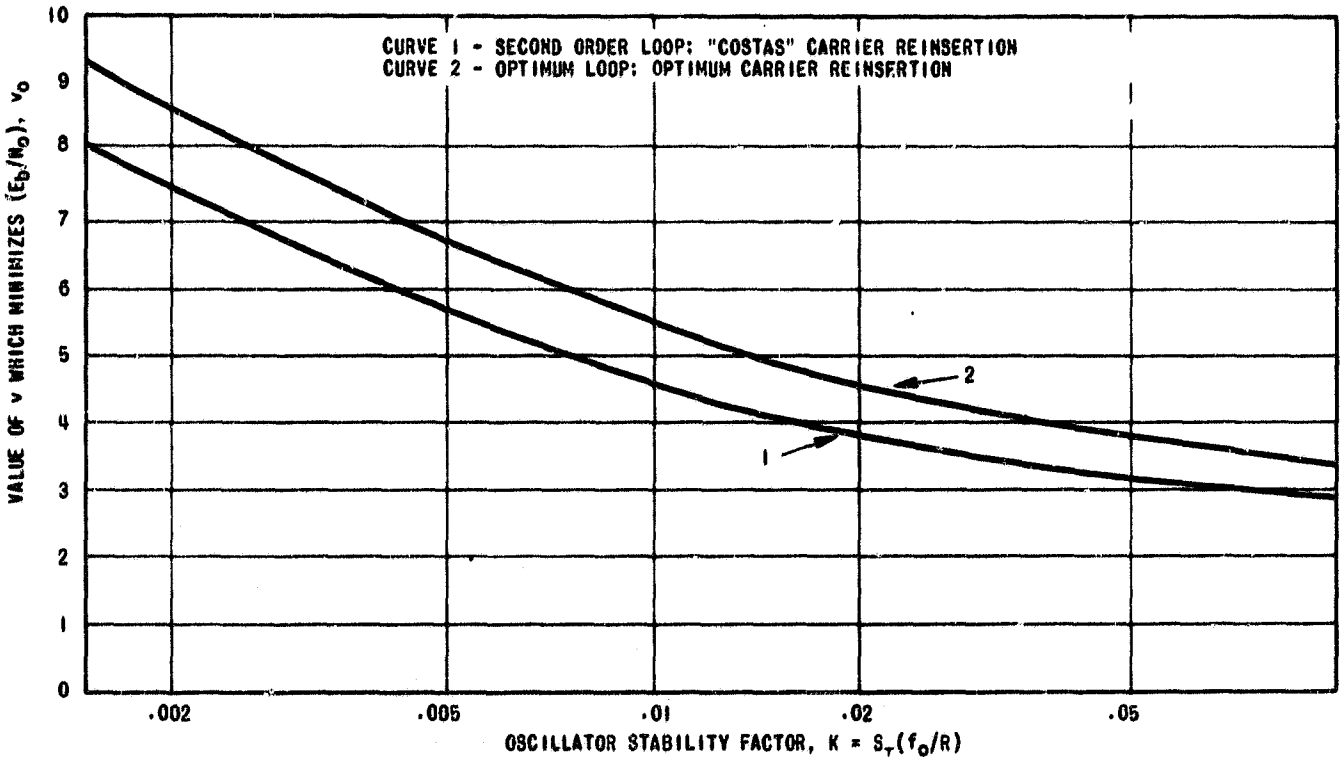


FIGURE 13. Optimum Value of ν versus Oscillator Stability Factor

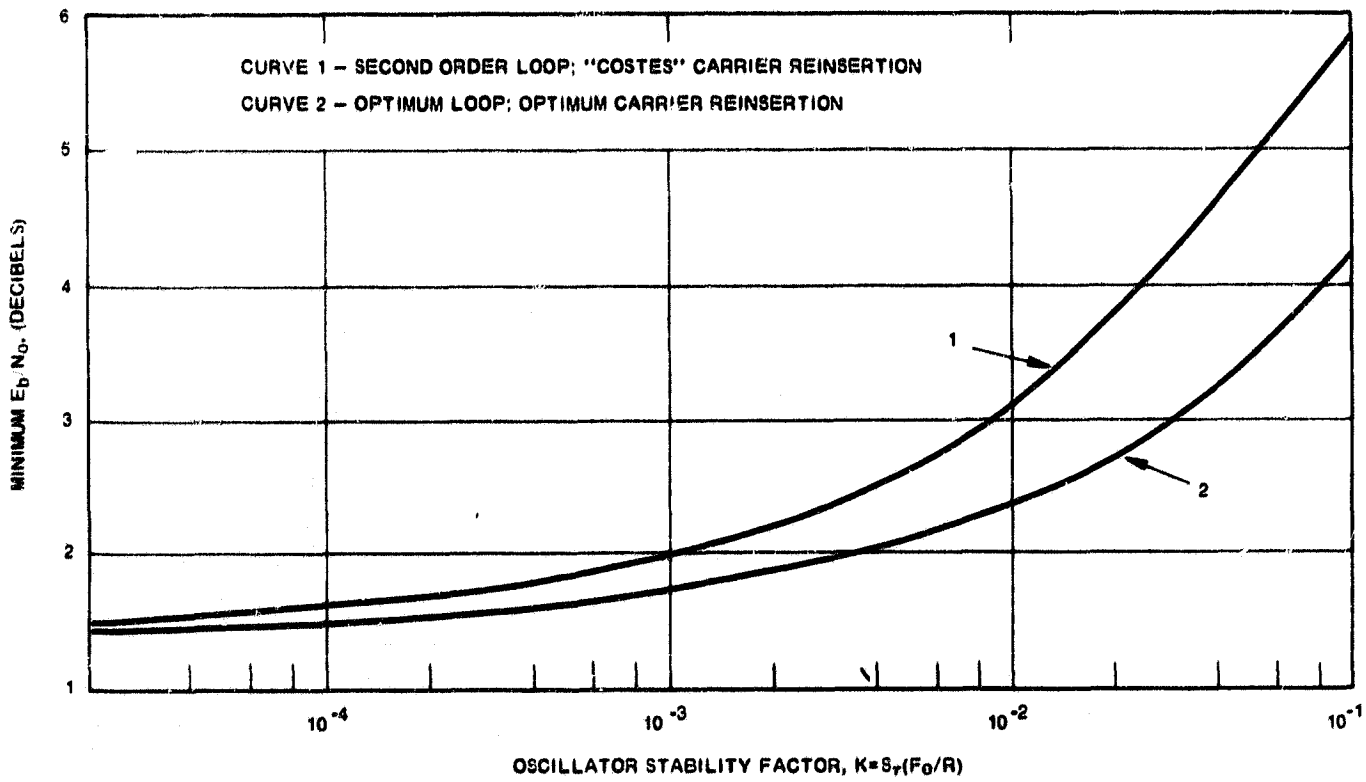


FIGURE 14. Minimum E_b/N_o versus Oscillator Stability Factor

10^{-12}. For post 1970 missions, a carrier frequency in the vicinity of 2.3 GHz is expected. Choosing $S_T = 10^{-12}$ and $f_o = 2.3 \times 10^9$, R = 0.0023/K bps. The value of K will be determined by the choice of carrier tracking circuit and the penalty in E_b/N_o that one is willing to accept in return for an unstable oscillator. Assume a one db sacrifice in E_b/N_o is acceptable. Then, Figure 14 shows that for E_b/N_o = 2.42 dB (one dB above the lower limit for sequential decoding), K = 0.011 and 0.0032 for the "optimum" and "Costas" tracking loops, respectively. Since these values of K differ only by a factor of about 3.5, the more conservative value of K = 0.0032 will be chosen since refined circuitry techniques cannot be expected to provide a substantial improvement. Since a larger value of K implies a larger value of E_b/N_o, the transmitted data rate can be bounded by

$$R > 0.0023/.0032 \approx 1 \text{ bps}$$

for

$$E_b/N_o \leq 2.42 \text{ dB}$$

Conclusions

It has been shown that an optimum code rate ($1/\nu_o$) exists for a specified oscillator stability, data rate and carrier frequency. In a practical sense, data rates should exceed 1 bps in order to realize efficient phase coherent communication. Convolutional code rates of less than 1/4 offer a negligible reduction of E_b/N_o with all but the most stable oscillators. With a code rate of 1/3, E_b/N_o is increased by less than 0.2 dB compared with a code rate of 1/4. Operation near E_b/N_o = 2.5 dB is achievable.

Near Earth Missions

For R>1000 bps, $f_o \approx 2.3$ GHz, and $S_T \approx 10^{-12}$, Equation (29) shows that $K < 2 \times 10^{-6}$. From Figure 11 and/or Figure 13 it can be seen that the corresponding ν_o is impractically large. With a second order loop and "Costas" carrier reinsertion, Figure 11 can be used with K = O to estimate E_b/N_o for various values of ν. A brief table of results is presented below.

ν	E_b/N_o
2	2.45
3	2.03
4	1.84
7	1.63

Since the IMP mission anticipates about 7 dB of margin above E_b/N_o = 4.5 dB* at a range of 150,000 miles (a rate 1/2 code is assumed), there is little reason to increase the complexity by going to a rate 1/3 code. As the table above shows, the potential advantage would be 2.45-2.03 = 0.42 dB. This improvement might prove worthwhile, however, in other near earth missions. To increase ν beyond 3, however, provides very little potential improvement.

* As provided by GSFC

Deep Space Missions

For $1<R<1000$ bps, $f_o \approx 2.3$ GHz, and $S_T \approx 10^{-12}$, Equation 29 shows that $2x10^{-6} < K < 2x10^{-3}$. From Figure 13 it can be seen that the optimum value of ν with $K = 2x10^{-3}$ (corresponding to R = 1 bps) is about 7.5 with a second order Costas loop. From Figure 11, the following values are obtained at $K = 2.5x10^{-3}$ which approximates the performance expected at R = 1 bps

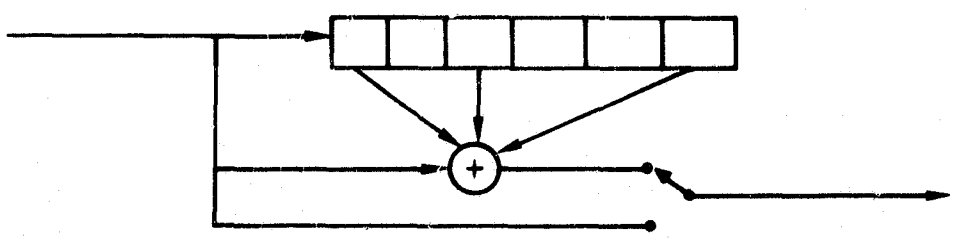
ν	E_b/N_o
2	2.81
3	2.46
4	2.35
optimum ≈ 7	2.32

Therefore, at low data rates, use of $\nu = 3$ or 4 provides a reduction of E_b of 0.35 or 0.46 dB respectively compared with use of $\nu = 2$. With $\nu = 3$ the performance is only 0.14 dB inferior to that which is obtained with $\nu = 7$. At very low data rates $\nu = 3$ is probably the best practical choice. In fact, it $S_T > 10^{-11}$, $\nu=3$ represents (essentially) the optimum value for R≈1 bps. On the other hand, if R > 100 bps and $S_T \approx 10^{-12}$, $K < 2x10^{-5}$ in which case the table of values given under Near Earth Missions can be used with accuracy.

5.3.3 Code Constraint Length

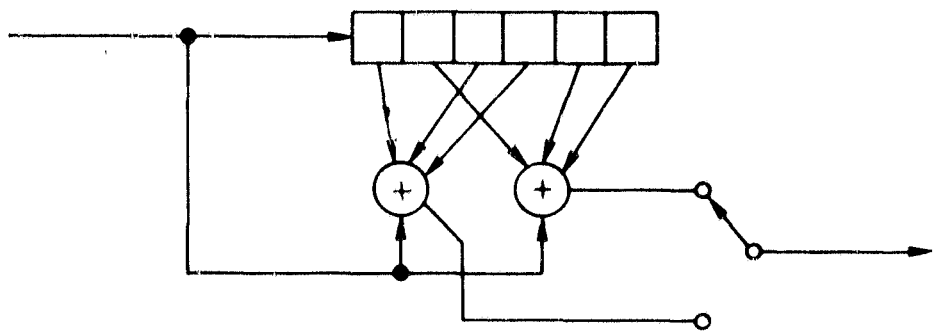
When dealing with convolutional codes, there are several ways in which "constraint length" can be defined. In this paragraph the encoder connection set described by the generating polynominal 1101001 for example, will be considered to yield a systematic code with a constraint length of L = 7. In more generality, the constraint length will be defined here as the length of the generating polynominal from the first binary one to the last binary one, inclusive.

Although this simple definition is at variance with some other mathematical definitions which have been used in the literature*, it provides a direct measure of the required length of the encoder shift register. To generate a systematic code with constraint length L requires a shift register with as few as (L-1) stages. The diagram below illustrates this point for the 1101001 generating polynominal mentioned above. When dealing with non-systematic codes, two lengths are given. For example, consider the nonsystematic code generated by the



*For example, the generating polynominal 11010 is tabulated by Bussgang as an optimum polynominal for a mathematical constraint length of N = 5 even though the last stage of the encoder shift register is not used.

sub-generators 11011 and 1010011. This code is herein defined by $L_1 = 5$ and $L_2 = 7$. It can be generated by a single shift register with $[\max \{L_1, L_2\} - 1]$ stages (7-1=6 stages in this example) as illustrated below:



The extension of this definition for constraint length to lower rate codes is obvious.

Choice of constraint length can be made on the basis of required decoded bit error probability. Results of simulations at C&S of five good systematic codes are illustrated in Figure 15 as decoded bit error probability versus constraint length. Details of the simulation can be found in paragraph 7.4. As is well known, the decoded error probability is seen to decay (approximately) exponentially with constraint length. Since each simulation point was obtained with a single code rather than an average of many codes, the results for a fixed value of α cannot be expected to fall exactly on a straight line (exponential decay). Also, it is seen that operation close to R_{comp} requires longer codes for a given level of performance. Quantitatively, these results indicate that L must be increased about 1.8 fold in order to hold the decoded bit error probability fixed while α is increased from 0.867 to 0.956. This range of α is equivalent to input symbol error probabilities ranging from 3 to 4 percent.

The improved error correcting capability that results from sequentially decoding long codes as opposed to short codes is paid for in two ways. First, and most obvious, is that a longer spaceborne shift register is required to encode the data. Consequently, increased weight, size, power consumption and unreliability are imposed upon the spaceborne encoder. All or none of these factors, may be inconsequential when considering a choice of L between 32 and 48. These considerations have not been studied here.

Secondly, in order to correct more errors (with larger L) one should expect the decoder to do more work. This is illustrated in Figure 16 which shows the average number of decoder computations per decoded bit versus constraint length. These results were determined from the same C&S simulations discussed earlier. As L is increased, α is held fixed. Figure 15 showed that the decoded bit error probability diminishes whereas Figure 16 shows that the average number of computations per bit required to achieve this performance increases to an asymptotic value. It would appear from these limited results that very long codes would require about 2.4 and 4 computations per bit at $\alpha = 0.867$ and 0.956 respectively. Note that these results were obtained with a zero detection rate. Theoretically, when α approaches 1 and L approaches infinity, the average number of computations per bit will also approach infinity.

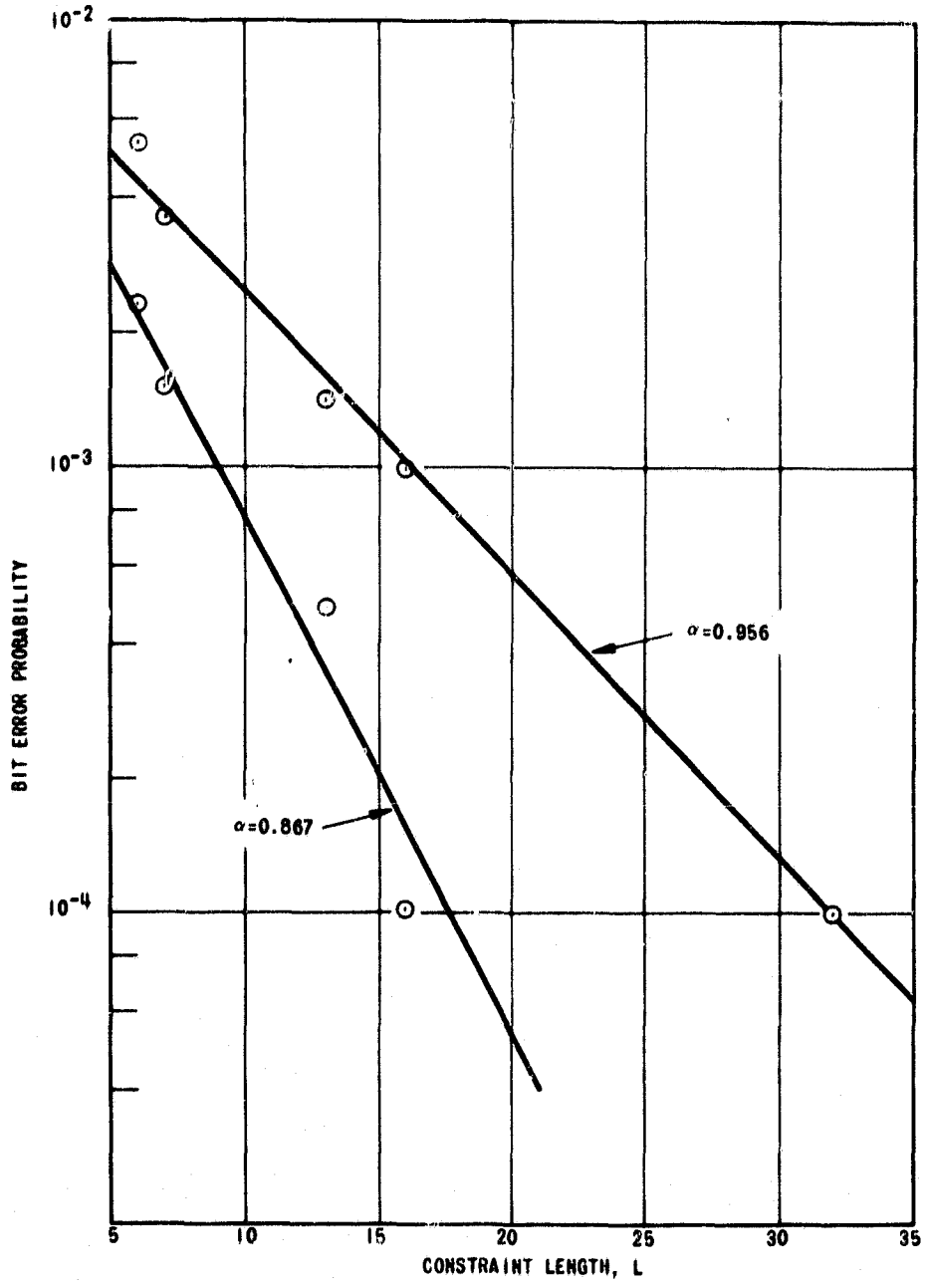


FIGURE 15. Bit Error Probability Versus Constraint Length

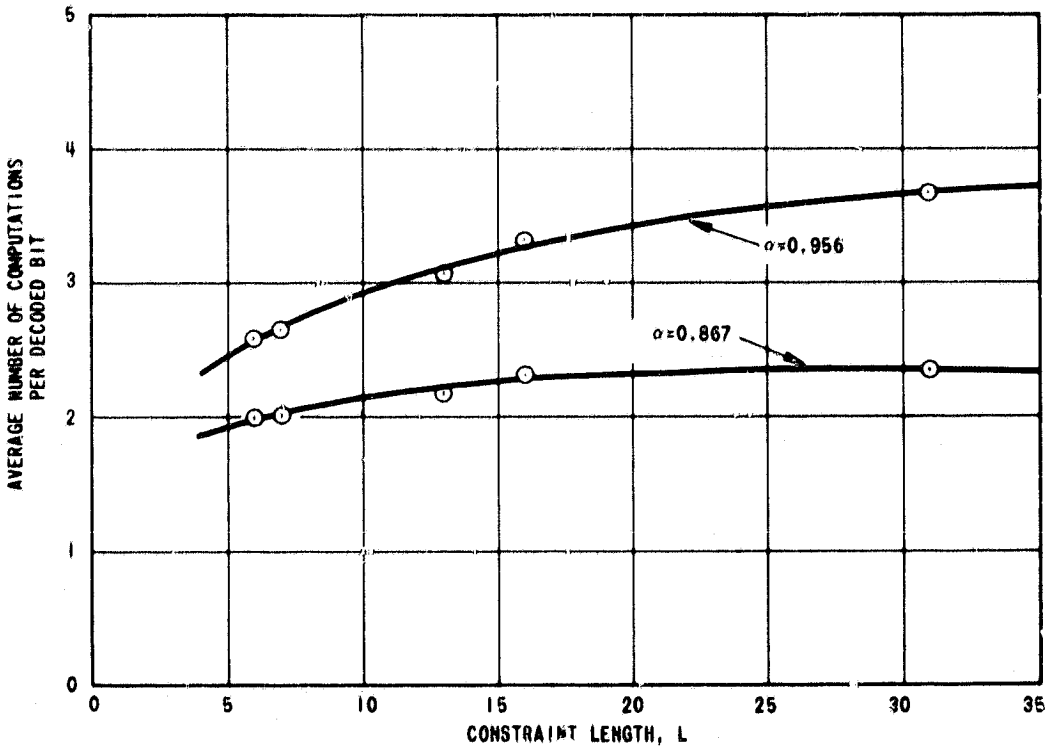


FIGURE 16. Average Number of Computations Versus Constraint Length

5.4 DECODING

5.4.1 Introduction

Most detection techniques for binary modulation use some form of absolute decision on each received bit (i.e., a decision that it is a "0" or a "1"). This decision is made by observing some voltage and comparing it with a threshold setting. Such a detection method is known as a "hard decision." Among others, Jordan[15] realized that signaling efficiency would be improved for binary modulation if additional information regarding the received signal was to be retained. One way of achieving this is through multilevel amplitude quantization.

The purpose of this paragraph is to investigate amplitude quantization with binary antipodal signaling and sequential decoding. Binary antipodal signaling was chosen since this represents the most efficient form of binary modulation. The measure of signaling efficiency used here is E_b/N_o where E_b is the received energy required for the transmission of one bit of data and N_o is the noise power density. Hence

$$\frac{E_b}{N_o} = \frac{E_s}{N_o} \quad \frac{1}{R} \tag{30}$$

where

E_s is the received energy per binary symbol
R is the data rate in bits per binary symbol.

There is a limit to how large R can be. This limit is imposed by the sequential decoder, since successful decoding can only be accomplished if the received waveform has sufficient redundancy to allow the decoder to correct received errors. A high error rate will require large redundancy (small R). Theoretical considerations have revealed that the average number of decoder computations per decoded data bit approaches infinity when $R = R_{comp}$ where the theoretical number R_{comp} is called the computational cutoff rate. In practice, computer simulation studies reveal that for $R < 0.8\ R_{comp}$, decoding proceeds smoothly with less than approximately two decoder computations per decoded data bit. At $R = R_{comp}$ this number rises to about 15. At still higher values of R, the average number of computations continues to rise very rapidly. In the following analysis, it will be assumed that $R = R_{comp}$, since this represents both a practical and a theoretical limit on the largest value of R. The overall system efficiency when $R < R_{comp}$ is discussed by Jordan,[16].

5.4.2 Bounds On E_b/N_o

Reiffen (Reference 17) has shown that for discrete, memoryless channels,

$$R_{comp} = \max_{\{p_k\}} \left\{ -\log_2 \sum_{j=1}^{J} \left[\sum_{k=1}^{K} p_k \sqrt{p(j \mid k)} \right]^2 \right\} \tag{31}$$

where

K = number of input levels

J = number of output levels

p_k = a-priori probabilities of channel input levels

$p(j/k)$ = channel transition probabilities $(= p_{jk})$

For the case of binary input channels with J outputs, Equation (31) reduces to

$$R_{comp} = 2 - \log_2 \sum_{j=1}^{J} \left[\sqrt{p(j|1)} + \sqrt{p(j|2)} \right]^2 \tag{32}$$

and for the binary symmetric channel, Equation (32) further simplifies to

$$R_{comp} = 1 - \log_2 \left[1 + 2 \sqrt{p\,(1-p)} \right] \tag{33}$$

where

p is the channel probability of error.

For phase coherent channels with binary antipodal modulation,

$$p = \frac{1}{\sqrt{2\pi}} \int_{\sqrt{\frac{2E_s}{N_o}}}^{\infty} e^{-x^2/2} \, dx \tag{34}$$

Using Equations (30), (33), and (34) curves relating E_b/N_o, E_s/N_o, and R_{comp} may be obtained for the case of a "hard decision". These results were presented by Jordan and are included in Figures 17, 18, and 19 for reference purposes. The "hard decision" curves, then, represent an upper bound (on E_b/N_o) against which the quantization results may be compared. Upper bound means that it is to be expected that quantization into $J > 2$ output levels will result in an E_b/N_o less than that for a hard decision $(J = 2)$.

A lower bound for E_b/N_o was also derived by Jordan (Reference 18) by assuming a continuous output, using Equation (31) and replacing the sum by an integral. The result is:

$$R_{comp} = \log_2 \frac{2}{1 + \exp\,(-E_s/N_o)}$$

Relations among E_b/N_o, E_s/N_o, and R_{comp} for the case of continuous output can then be obtained using Equations 30 and 35. These are shown in Figures 17, 18 and 19 by the curves labeled "continuous output."

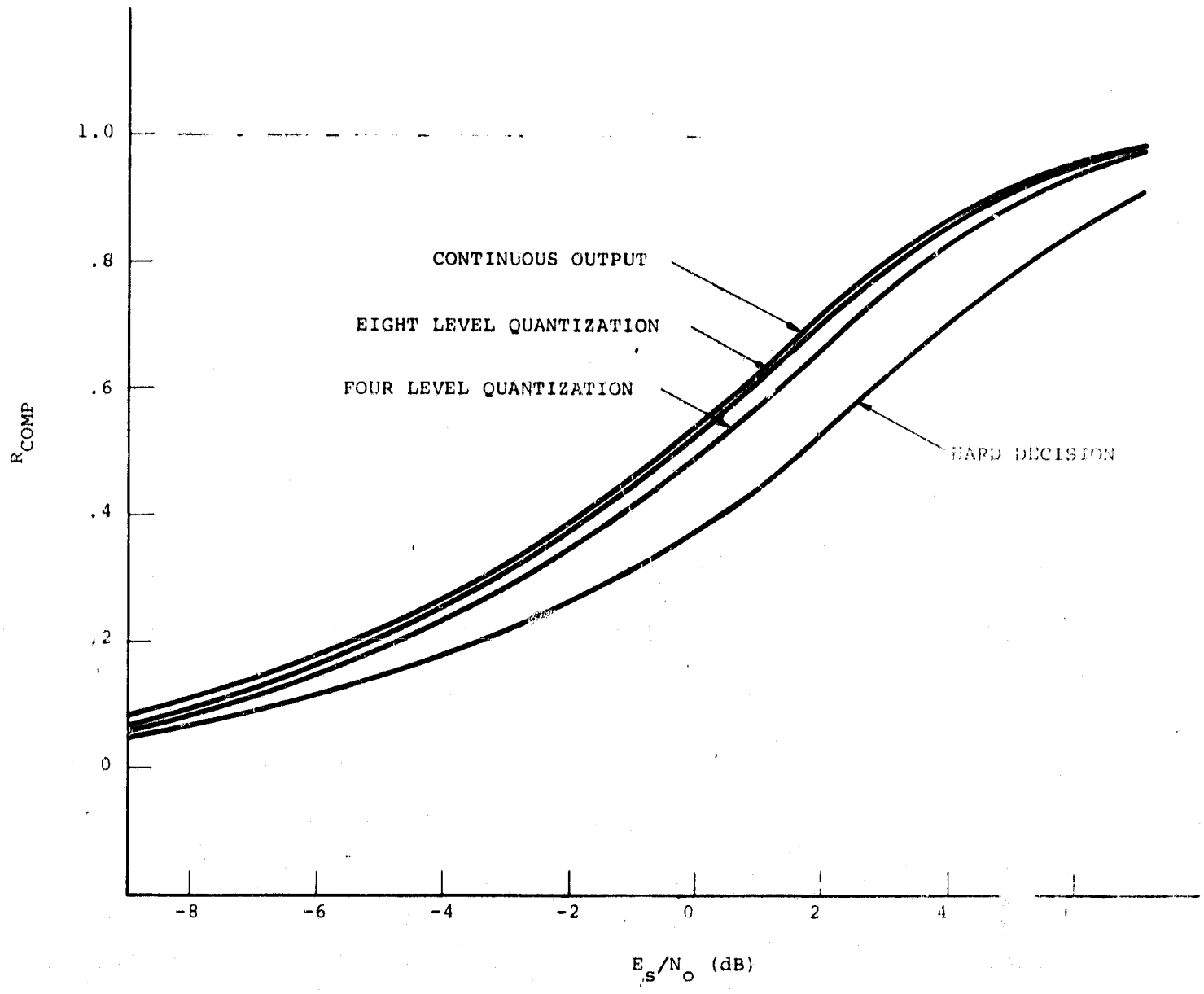


FIGURE 17. R_{comp} as a Function of E_s/N_o

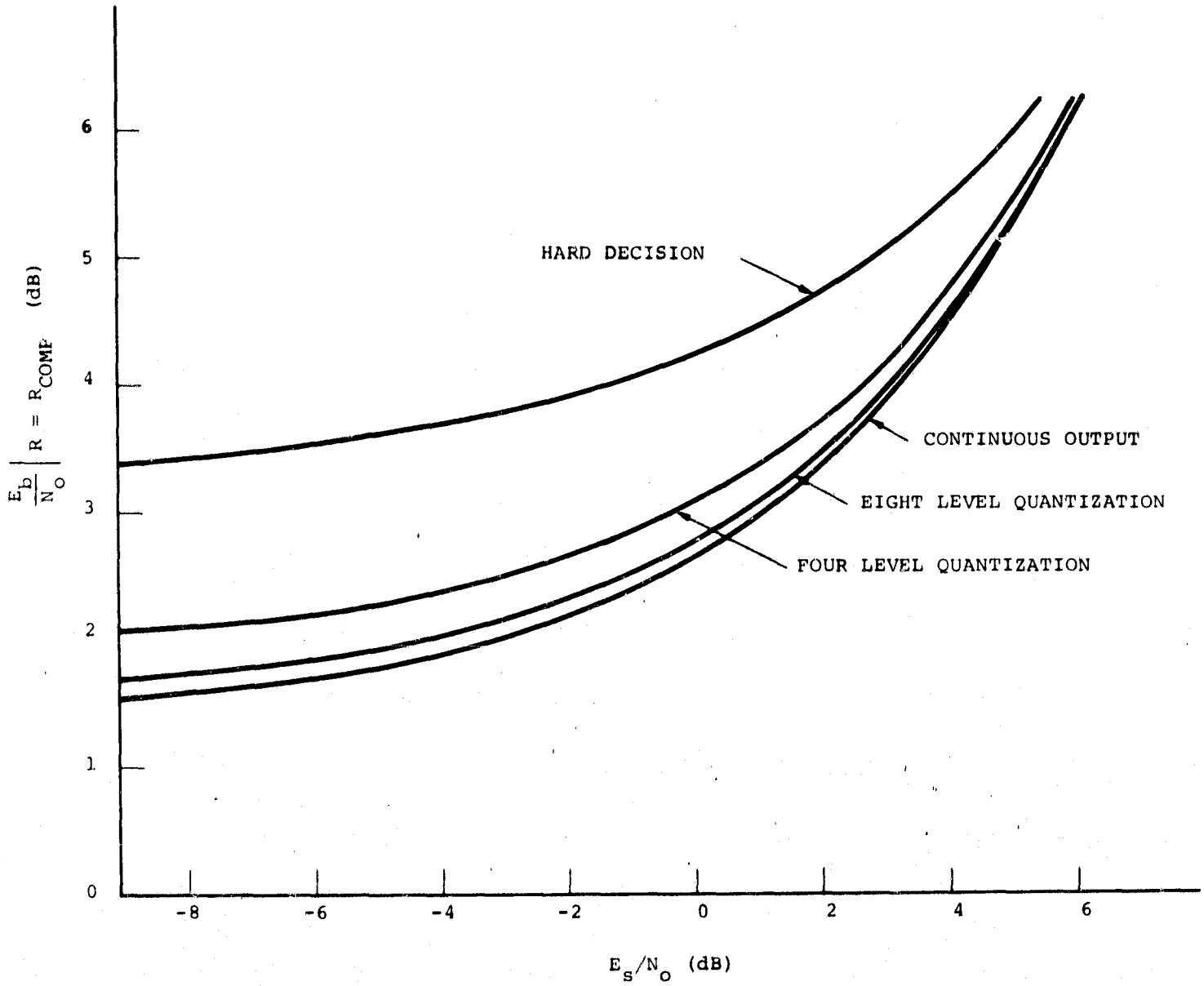


FIGURE 18. E_b/N_o as a Function of E_s/N_o

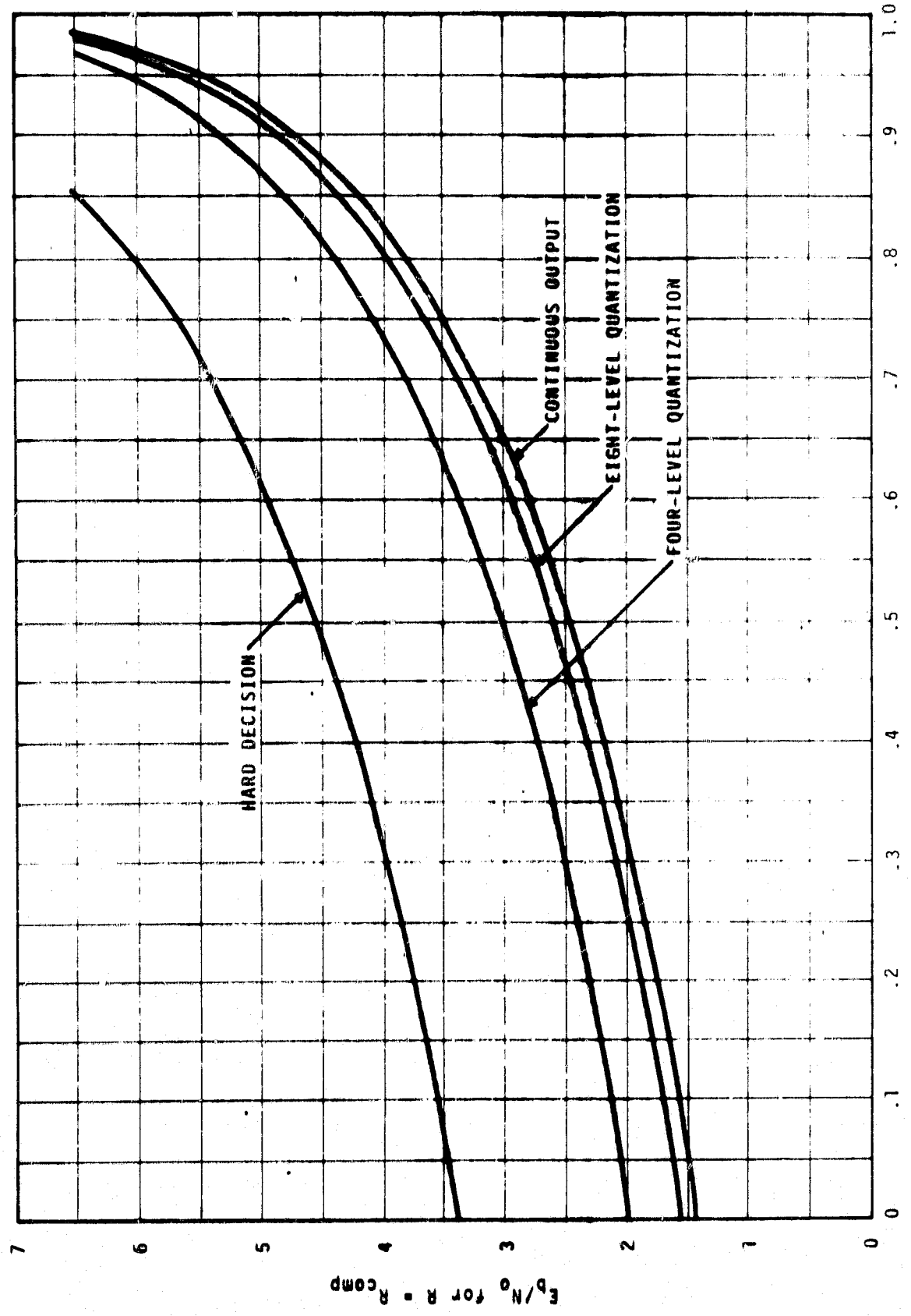


FIGURE 19. E_b/N_o as a Function of R_{comp}

Hence, the curves labeled "hard decision" and "continuous output" represent upper and lower bounds to realizable values of E_b/N_o for binary antipodal signaling. Inspection of these curves reveals that significant improvement over the hard decision case is possible.

5.4.3 N Level Amplitude Quantization

The probability density functions at the detector output for phase reversal (binary antipodal) signaling are well known and are given by the expressions:

$$p(y|x_1) = \frac{1}{\sqrt{2\pi}} \quad e^{-\frac{(y-a)^2}{2}}$$

$$p(y|x_2) = \frac{1}{\sqrt{2\pi}} \quad e^{-\frac{(y+a)^2}{2}} \qquad (36)$$

where

$$a = \sqrt{2E_s/N_o}$$

The received signal space may be divided into N regions as shown in Figure 20.

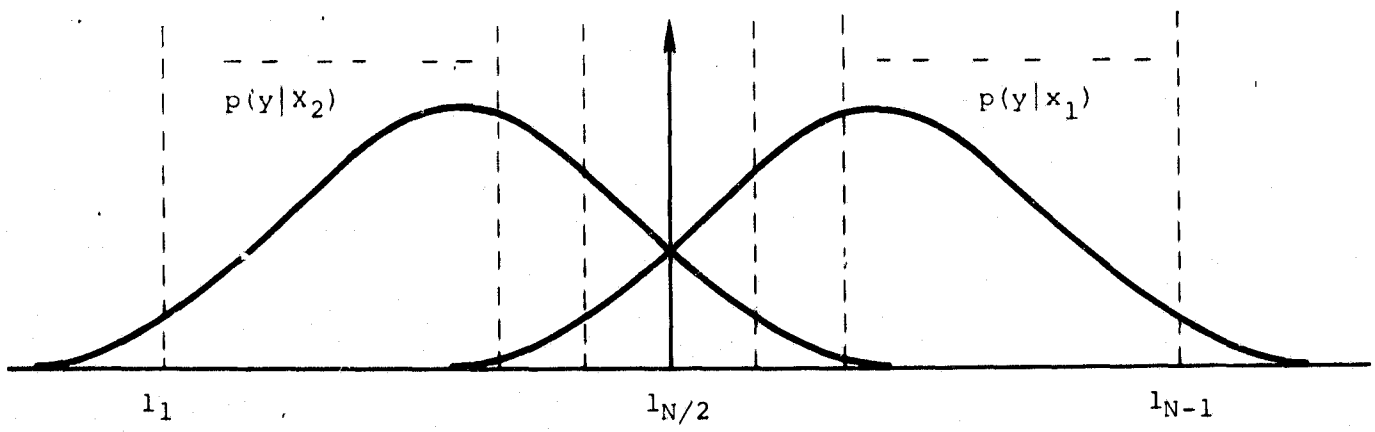


FIGURE 20. N Level Quantization

Expressing the channel transition probabilities conditioned on x in terms of the quantization levels and E_s/N_o (or more conveniently the parameter a), N equations are obtained of the form

$$p_{j1} = \frac{1}{\sqrt{2\pi}} \int_{1_{j-1}}^{1_j} e^{-\frac{(x-a)^2}{2}} dx \qquad \begin{array}{l} j = 1, 2, \ldots N \\ N \text{ an even integer} \end{array} \qquad (37)$$

The symmetry of the density functions leads to the consideration of symmetrical quantization divisions and the following relationships holds

$$p(j\,|1) = p(N-j+1\,|2) \qquad j = 1, 2, \ldots N \qquad (38)$$

Using Equations (32) and (38), R_{comp} may now be determined for the channel with N level quantization. This will be done for the specific cases of four- and eight-level quantization.

Four-Level Quantization

Using Equations (32) and (38) the expression for R_{comp} for the case of four-level quantization is

$$R_{comp} = 1 - \log_2 \left[\left(\sqrt{p_1} + \sqrt{p_4} \right)^2 + \left(\sqrt{p_2} + \sqrt{p_3} \right)^2 \right] \qquad (39)$$

For notational convenience, the second index has been dropped from the channel transition probabilities (i.e., $p_{11} = p_{42} = p_1$).

The symmetry properties of the channel permit analysis using only one of the two possible conditional density functions, as shown in Figure 21, where T is the threshold setting (normalized to the rms noise). It can be seen that

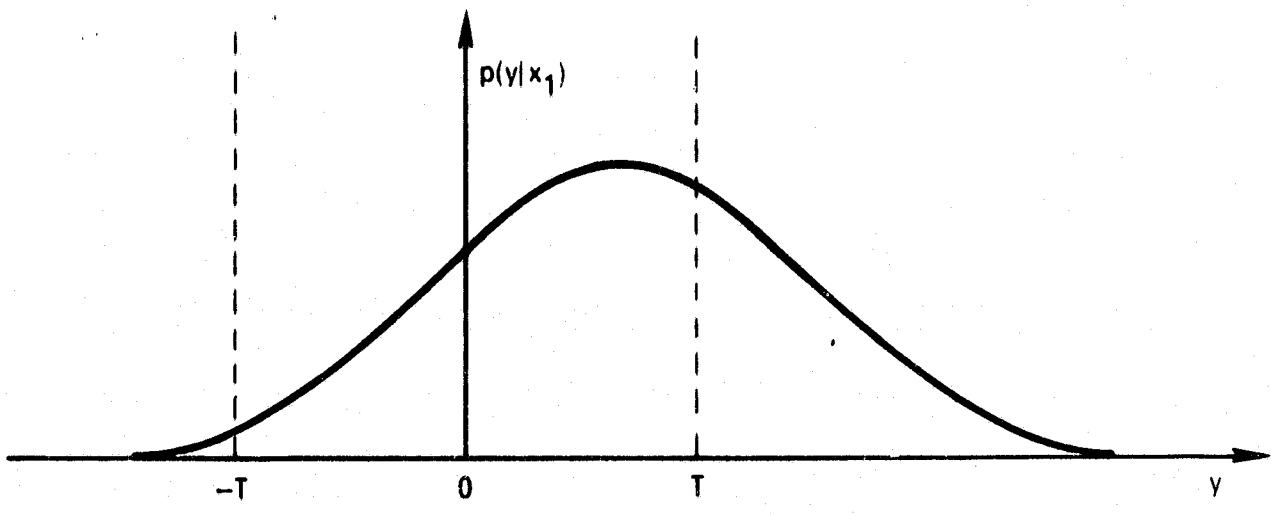


FIGURE 21. Four-Level Quantization

$$p_1 = \int_{-\infty}^{-T} f(y)dy$$

$$p_2 = \int_{-T}^{o} f(y)dy$$

$$p_3 = \int_{o}^{T} f(y)dy$$

$$p_4 = \int_{T}^{\infty} f(y)dy$$

where

$$f(y) = \frac{1}{\sqrt{2\pi}} \; e^{-\frac{(y-a)^2}{2}}$$

The next step is to determine the value of T such that R_{comp} is maximized (i.e., E_b/N_o minimized). The value of T which results will be called the optimum normalized threshold setting, T_o. From Equation 39, define

$$\psi = (\sqrt{p_1} + \sqrt{p_4})^2 + (\sqrt{p_2} + \sqrt{p_3})^2 \tag{41}$$

Since the logarithm is a monotonic function, maximizing R_{comp} is the same as minimizing ψ. Writing ψ as a function of T and differentiating with respect to T yields

$$\frac{d\psi}{dT} = (\sqrt{p_1} + \sqrt{p_4})\left[\frac{1}{\sqrt{p_1}} \frac{dp_1}{dT} + \frac{1}{\sqrt{p_4}} \frac{dp_4}{dT}\right] + (\sqrt{p_2} + \sqrt{p_3})$$
$$\left[\frac{1}{\sqrt{p_2}} \frac{dp_2}{dT} + \frac{1}{\sqrt{p_3}} \frac{dp_3}{dT}\right] \tag{42}$$

$$\frac{d\psi}{dT} = (\sqrt{p_1} + \sqrt{p_4})\left[\frac{f(-T)}{\sqrt{p_1}} + \frac{f(T)}{\sqrt{p_4}}\right] + (\sqrt{p_2} + \sqrt{p_3})\left[\frac{f(T)}{\sqrt{p_3}} + \frac{f(-T)}{\sqrt{p_2}}\right] = 0 \qquad (43)$$

Equation (43) may now be solved for $T = T_o$.

Inspection of Equation (43) shows that the unknown, T, appears both in the exponential and the limits of integration of the normal variable. Hence, a closed form solution cannot be obtained. Equation (43) was solved numerically using a computer. A plot of T_o vs. E_s/N_o is shown in Figure 22. The important observation from Figure 22 is that T_o is a slowly varying function of E_s/N_o. This behavior is important when practical receiver implementation is considered.

Using the solution obtained for T_o, R_{comp} may now be computed as a function of E_s/N_o and related to E_b/N_o. The results are shown in Figures 17, 18 and 19 by the curves labeled "Four Level Quantization." Relations among R_{comp}, E_b/N_o and E_s/N_o were also obtained by setting $T = 1.0$. The results were essentially identical to those for $T = T_o$, the difference occurring only in the fourth or fifth significant digit of R_{comp}. Hence, the optimum results may be approached very closely with a fixed threshold design.

Eight-Level Quantization

The eight-level case was analyzed to determine how much additional improvement in performance might be achieved. The analysis parallels that for the four-level case with the new expression for R_{comp} being

$$R_{comp} = 1 - \log_2 \sum_{i=1}^{4} (\sqrt{p_i} + \sqrt{p_{9-i}})^2 \qquad (44)$$

Now there are three threshold settings to be made as shown in Figure 23.

For this case, three independent partial differential equations with three unknowns may be written. After some simplifications these become:

$$\frac{\partial \psi}{\partial T_1} = -(\sqrt{p_3}+\sqrt{p_6})\left[\frac{f(-T_1)}{\sqrt{p_3}} + \frac{f(T_1)}{\sqrt{p_6}}\right] + (\sqrt{p_4}+\sqrt{p_5})\left[\frac{f(-T_1)}{\sqrt{p_4}} + \frac{f(T_1)}{\sqrt{p_5}}\right] = 0$$

$$\frac{\partial \psi}{\partial T_2} = -(\sqrt{p_2}+\sqrt{p_7})\left[\frac{f(-T_2)}{\sqrt{p_2}} + \frac{f(T_2)}{\sqrt{p_7}}\right] + (\sqrt{p_3}+\sqrt{p_6})\left[\frac{f(-T_2)}{\sqrt{p_3}} + \frac{f(T_2)}{\sqrt{p_6}}\right] = 0 \qquad (45)$$

$$\frac{\partial \psi}{\partial T_3} = -(\sqrt{p_1}+\sqrt{p_8})\left[\frac{f(-T_3)}{\sqrt{p_1}} + \frac{f(T_3)}{\sqrt{p_8}}\right] + (\sqrt{p_2}+\sqrt{p_7})\left[\frac{f(-T_3)}{\sqrt{p_2}} + \frac{f(T_3)}{\sqrt{p_7}}\right] = 0$$

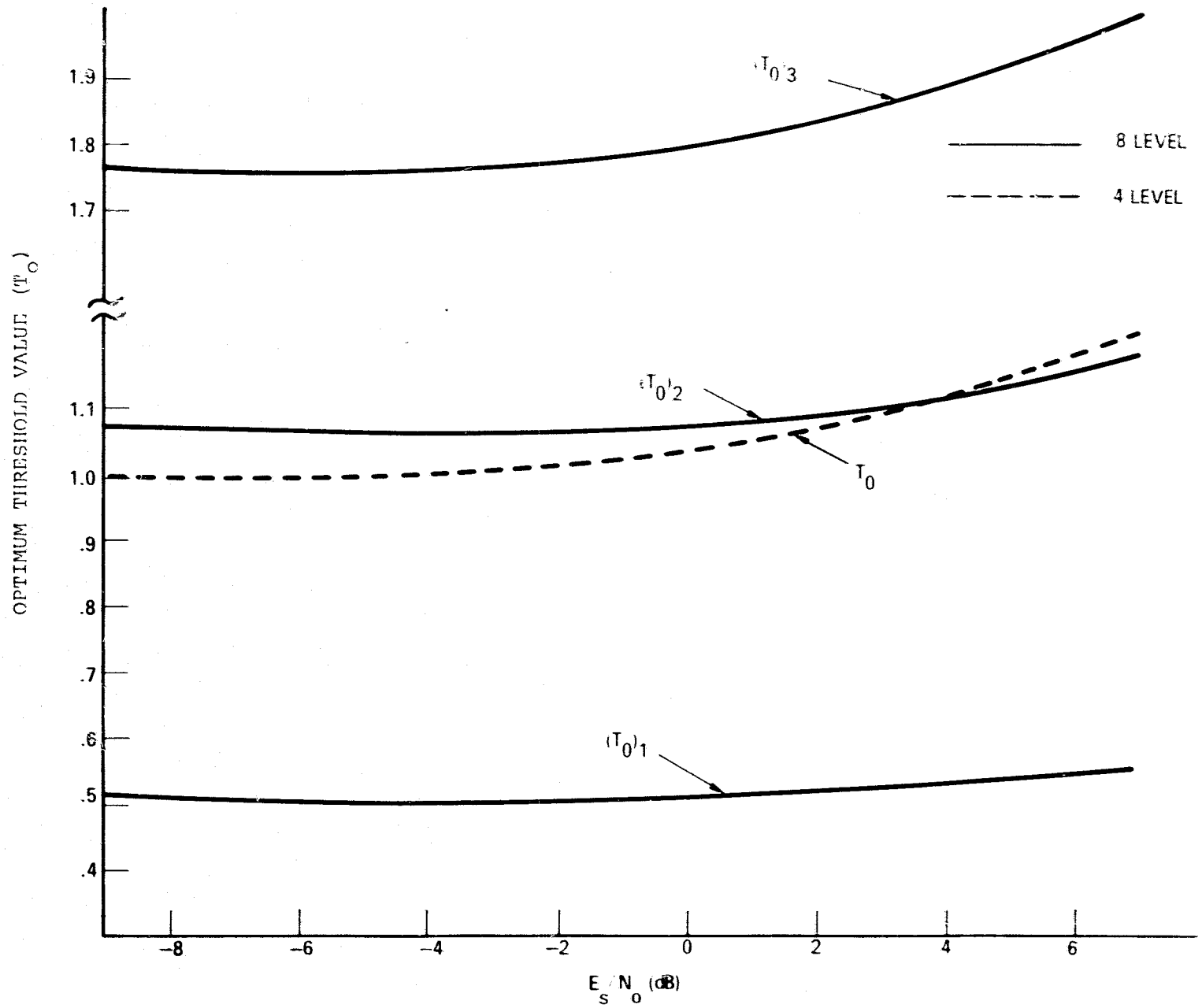


FIGURE 22. Optimum Threshold Level Settings for Four and Eight Level Quantization

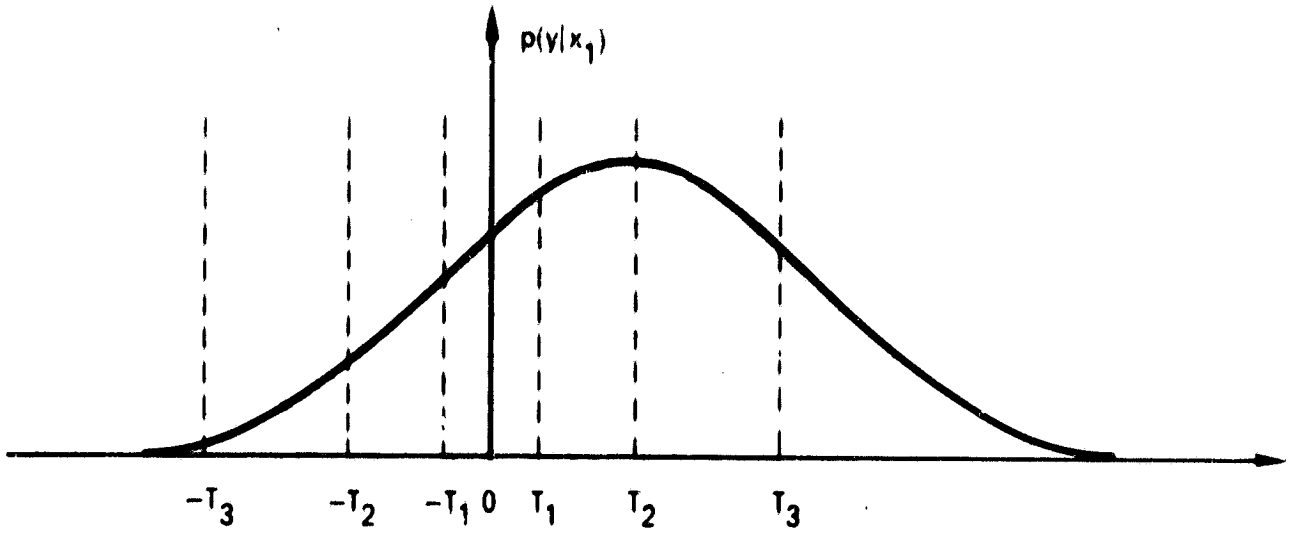


Figure 23. Eight-Level Quantization

These simultaneous equations were again solved utilizing a computer with the results shown in Figure 22. As in the case of four-level quantization, the optimum normalized threshold settings are slowly varying functions of E_s/N_o. Using the optimum threshold values obtained from Equation (45), $(T_o)_1$, $(T_o)_2$, $(T_o)_3$, and Equations (30), (37), and (44), relations among R_{comp}, E_s/N_o, and E_b/N_o were calculated. These same relations were calculated after setting $T_1 = 0.505$, $T_2 = 1.060$, and $T_3 = 1.780$. Again, the results coincided to at least three decimal places in R_{comp} with those for optimum (adaptive) threshold settings.

5.4.4 Practical Implementation Considerations

This paragraph discusses the practical aspects of multilevel quantization. Consideration will be limited to three- and four-level quantization schemes which are most amenable to implementation and subsequent adjustment, since only one independent threshold setting is required. The results which are discussed below were obtained from minor alterations of the analysis previously presented.

In a practical sense, it is most convenient to investigate the performance in terms of baseband voltages which can readily be measured. This leads to the following definitions; (a) $\sqrt{N}$ is the rms baseband noise voltage (N is the total baseband noise power measured in a one ohm resistor) and (b) $\pm t$ is the threshold setting measured in volts. The video signal to noise power ratio SNR $= a^2 = 2(E_s/N_o)$ at the output of a filter which is matched to the received binary symbols. The manner in which R_{comp} varies as t is changed is shown in Figures 24 and 25 for three- and four-level quantization, respectively. With three-level quantization (often referred to as null-zone reception), Figure 24 indicates that the optimum threshold settings, t_o, are near 0.6 N for any video SNR from -9 to +9 dB. As t approaches

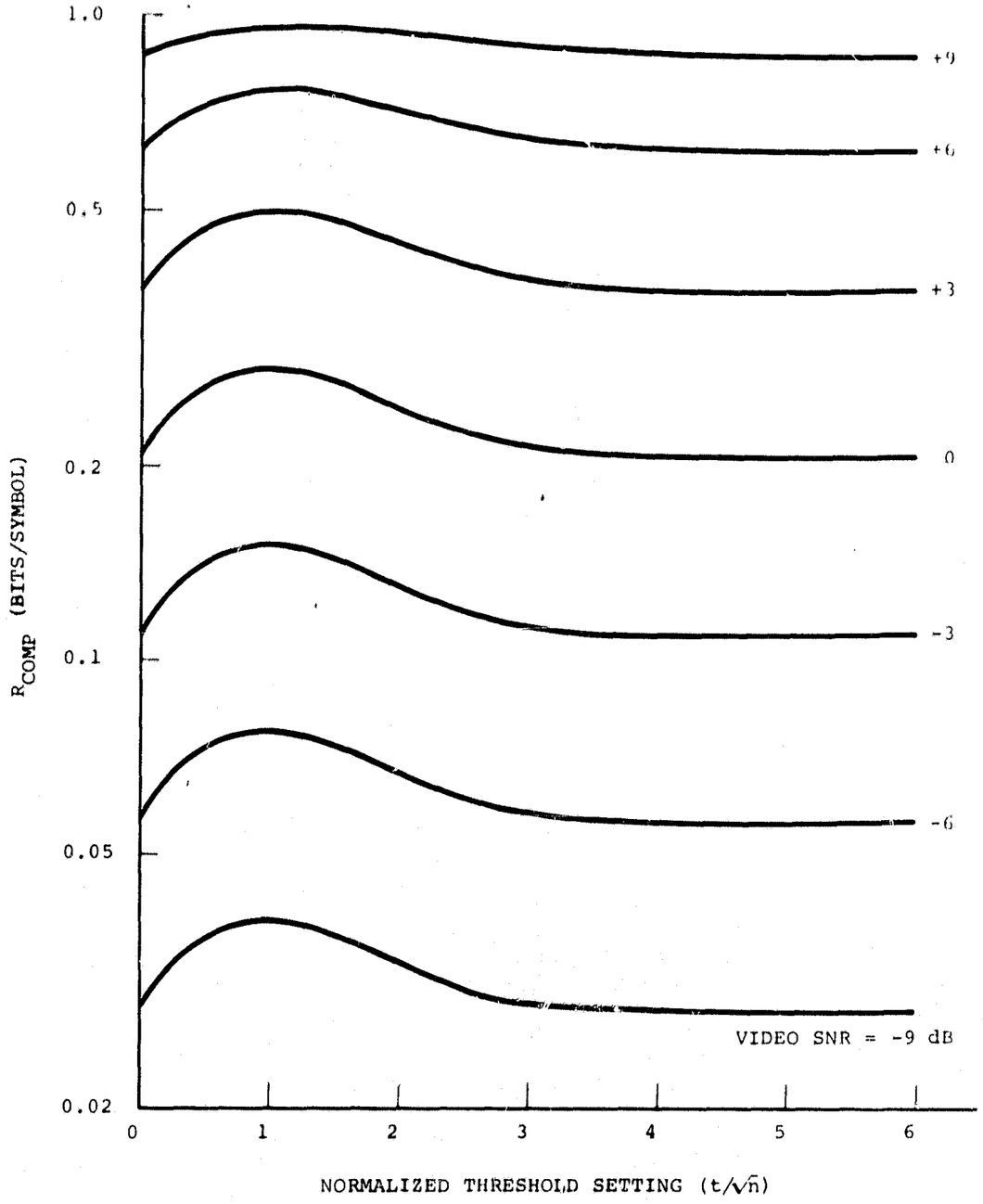


FIGURE 25. R_{comp} as a Function of Normalized Threshold Setting for Four Level Quantization

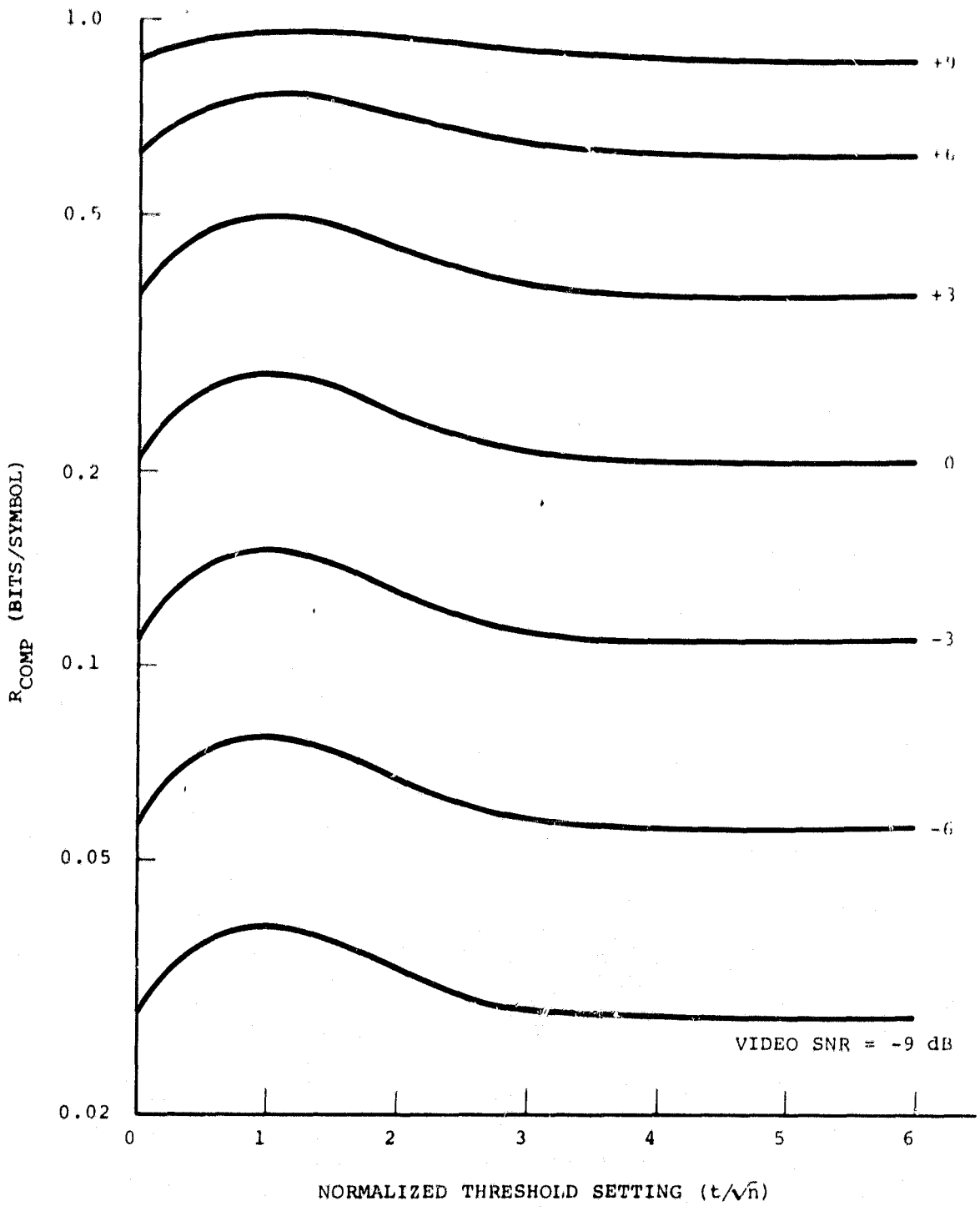


FIGURE 25. R_{comp} as a Function of Normalized Threshold Setting for Four Level Quantization

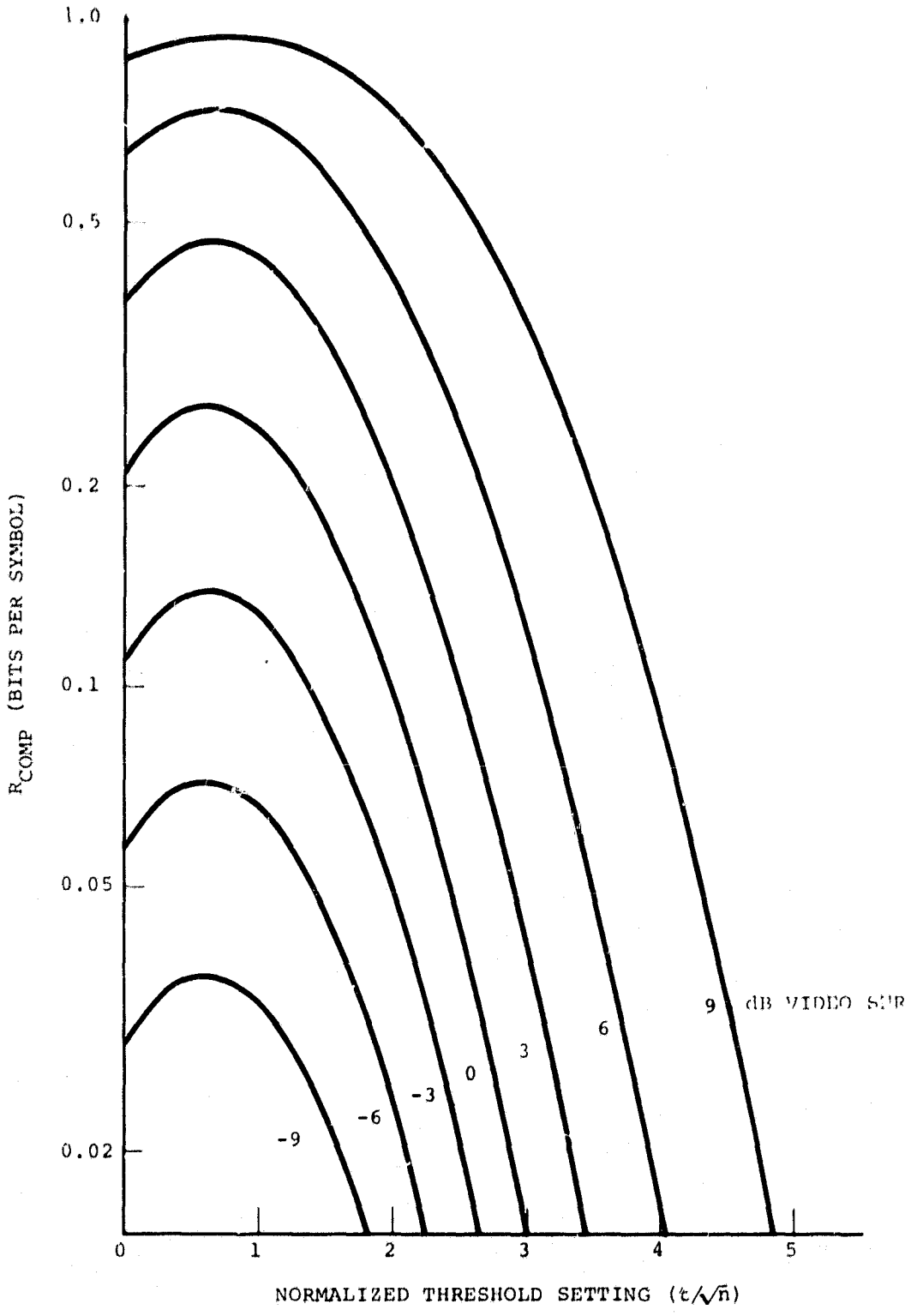


FIGURE 24. R_{comp} as a Function of Normalized Threshold Setting for Three Level Quantization

0 the performance degrades to that which is obtained with a "hard decision". If t is made too large, however, R_{comp} is seen to diminish rapidly. This is to be expected since most of the received symbols will fall into the null zone and the decoder is frequently asked to "guess" which symbol has been transmitted. With four-level quantization, Figure 25 indicates that t_o is near $1.0\sqrt{N}$ for any SNR from -9 to +9 dB. In this case, t can vary from zero to a very large value without catastrophic results; since in either case performance degrades to that which is obtained with a "hard decision". Practically speaking, optimized four-level quantization will be obtained when the quantization regions are $-\infty$ to $-\sqrt{N}$, $-\sqrt{N}$ to zero, zero to $+\sqrt{N}$, and $+\sqrt{N}$ to $+\infty$.

An accurate determination of the optimum threshold settings can be made from Figures 26 and 27 with two- and four-level quantization, respectively. Figures 24 and 25 reveal that the curves of R_{comp} versus $t/\sqrt{N}$ become relatively flat when SNR is large. Thus, it is concluded that if a fixed threshold setting is used (non-adaptive as SNR varies) t should be chosen from Figure 26 or 27 to be optimum at or near the expected minimum SNR. Although the same value of t will not be optimum at larger values of SNR, the performance degradation which results will be relatively small. By a fixed (non-adaptive) threshold it is meant that $t/\sqrt{N}$ is held. An adaptive threshold is defined as one in which $t/\sqrt{N}$ is a function of the video SNR, as illustrated by Figures 26 and 27.

Since it would be difficult to accurately measure $\sqrt{N}$ when signal plus noise is present, the following method of threshold adjustment is suggested. Corresponding to an optimum threshold setting with three-level quantization, there exists an optimum probability of quantizing the received signal into each of the three levels. Call the three quantization regions

a. Reliable spaces

b. Null symbols

c. Reliable marks.

Since the "reliable marks" and the "reliable spaces" occur with the same probability, one need only measure (or estimate) the probability of quantizing the received signal into a "null symbol" to determine if t is properly set. Such a measurement is easily performed by counting the number of "null symbols" presented to the decoder. For example, with a data rate of 500 bps and a rate R = 1/2 code, the received symbols arrive at the rate of 1000 per second. A counter at the null symbol point at the quantizer output, for example, indicates 92 pulses per second which provides the estimate of 0.092 as the probability of "null symbol" quantizer output. As shown in Figure 28, this probability is optimum for a SNR of +6 dB. As mentioned previously, little will be sacrificed by setting t as the optimum for the lowest expected SNR. If the counter reads too high, t should be decreased until the estimated probability is approximately correct for the value of minimum SNR which was chosen. In this manner, the effects of AGC drift and associated anomalies can be minimized. As long as the counter reads approximately the correct value, one is assured that the thresholds are set in a near optimum fashion for any SNR.

A similar procedure can be used with four-level quantization where the quantized symbols can be referred to as

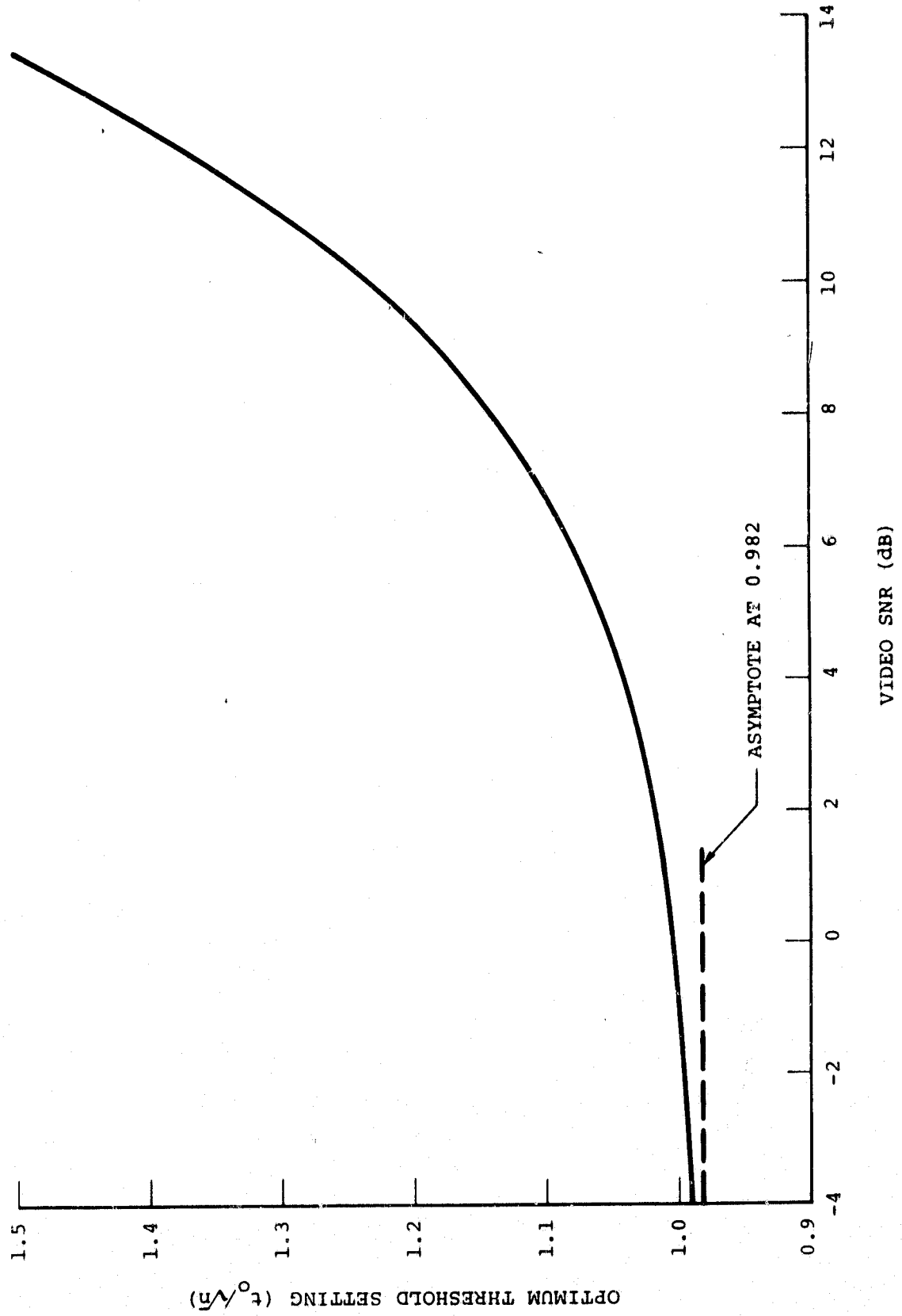


FIGURE 26. Optimum Threshold Setting for Three-Level Quantization

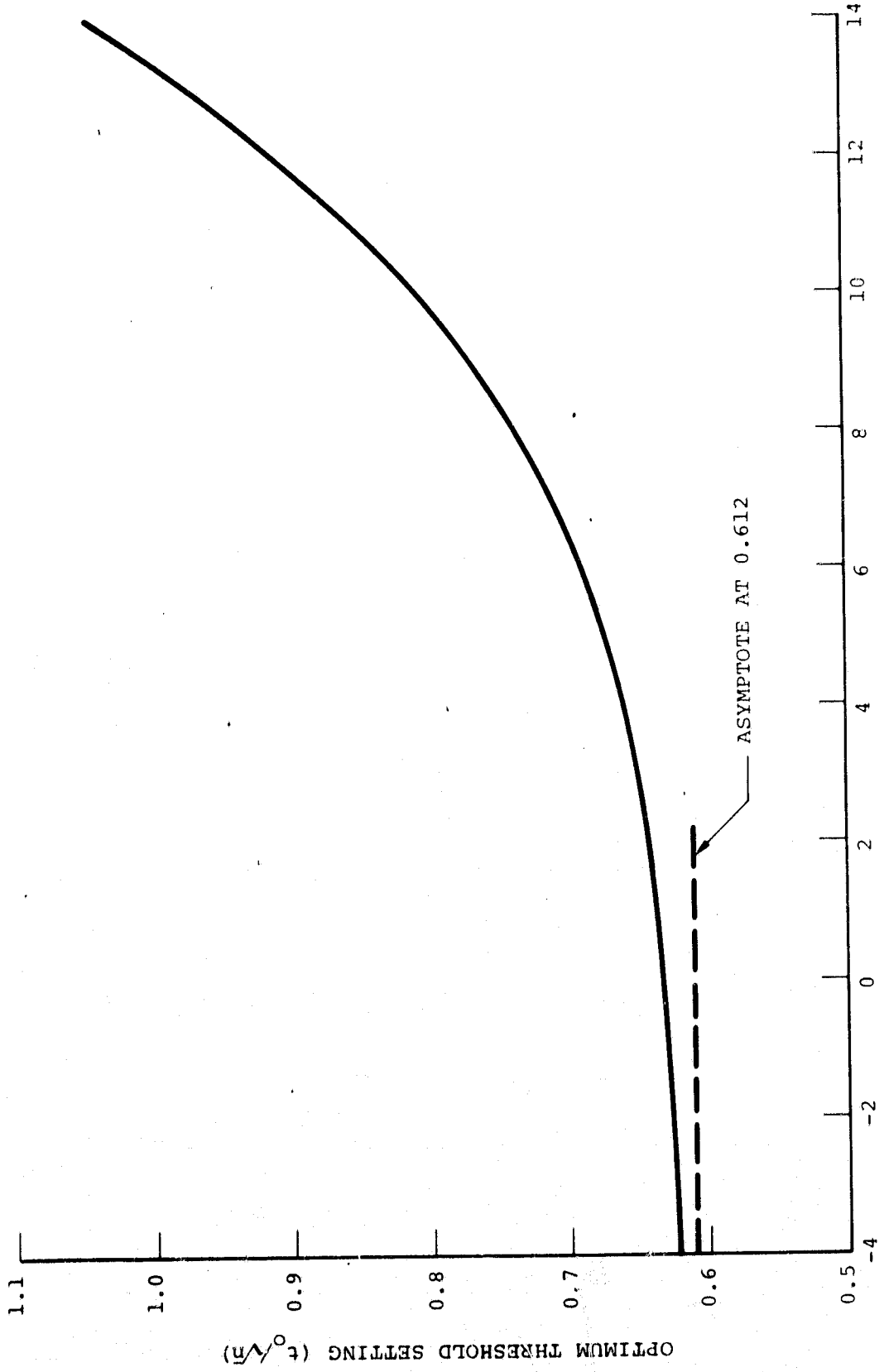


FIGURE 27. Optimum Threshold Setting for Four-Level Quantization

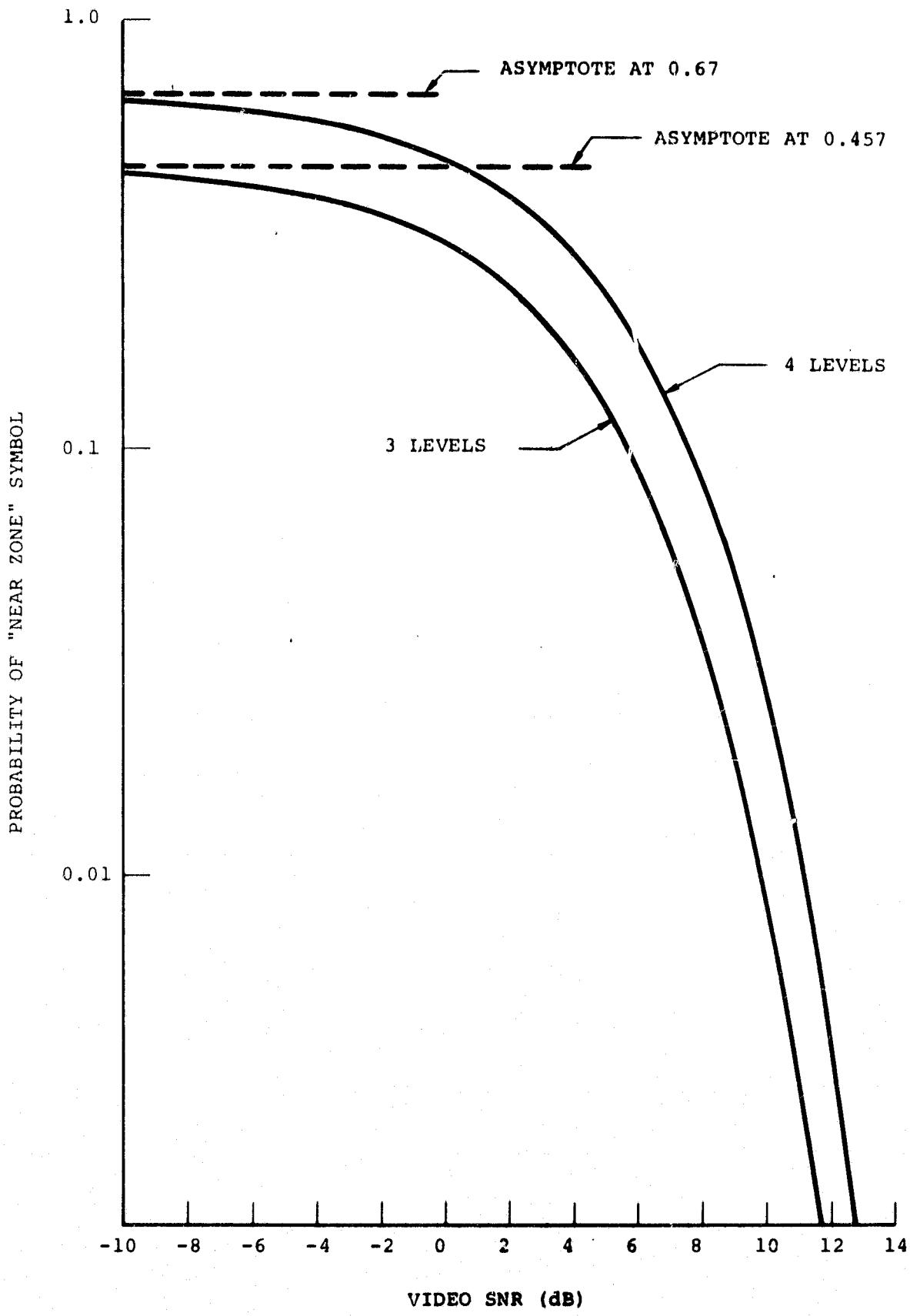


FIGURE 28. Probability of "Near Zone" Symbol as a Function of Video SNR

a. Reliable spaces

b. Unreliable spaces

c. Unreliable marks

d. Reliable marks.

Because of symmetries, it is only necessary to measure the probability of "unreliable" symbols (cases b and c). The probability of quantizing into an "unreliable" symbol can be called the proability of "near-zone" symbol reception since the sampled video voltage was nearer to zero than t. The same applies to the case of three-level quantization. Thus, the ordinate in Figure 28 is labelled "Probability of Near Zone Symbol" and the optimum value of this quantity is plotted versus video SNR for both three- and four-level quantization schemes. For example, suppose that four-level quantization is employed and the minimum video SNR for which the system is designed is 4 dB. For this case, Figure 28 indicates that the optimum probability of receiving a "near zone" symbol is about 0.29. Thus, a counter which counts the number of "unreliable marks" and "unreliable spaces" out of the quantizer should read about 290 per second with a received binary symbol rate of 1,000 per second. If the count tends to become significantly smaller than this value, t should be increased and vice versa. As long as the count is maintained near 290, near optimum performance is assured for any SNR near or above the selected minimum value for t.

Future work will more closely investigate the case of eight-level quantization if it is deemed advisable. That case, however, will require the use of three counters, and optimum threshold adjustment will be complicated by the interaction of the threshold settings.

Conclusions

a. A reduction in E_b/N_o on the order of 1.5 dB over the "hard decision" case may be realized by using four-level quantization. Additional improvement on the order of 0.4 dB may be achieved by using eight-level quantization.

b. Eight-level quantization approaches to within approximately 0.2 dB of the theoretical minimum E_b/N_o for binary antipodal signaling with continuous output (i.e., infinite level quantization). Hence, little justification exists to consider levels of quantization higher than eight.

c. Because of the relatively constant behavior of threshold setting as a function of Video SNR, nearly optimum performance may be obtained using preset level divisions. That is, performance could not be greatly enhanced by using adaptive techniques. This result is most useful when considering practical detector design.

d. A single counter can be used with three- and four-level quantization schemes to ascertain whether the thresholds are set properly. The counter can be used to automatically adjust the thresholds to compensate for AGC drift and similar phenomena.

e. The work presented here provides insight into some theoretical and practical problems associated with sequential decoding used in conjunction with level amplitude quantization. It should be noted that the criteria has been based upon maximizing the computational cutoff rate, R_{comp}. The effect of non-optimum (i.e., non-adaptive) thresholds on the

dynamic computational behavior of the sequential decoder has not been investigated. This should be the subject of a separate study.

5.4.5 The Design and Performance of a PSK Symbol Synchronizer Operating at High Symbol Error Rates

The timing sources, commonly referred to as symbol synchronizers, utilized in demodulating binary signals, exhibit deterioration in timing accuracy with symbol error rate. In systems operating at high symbol error rate, this can become a source of degradation. The problem becomes particularly acute in convolutionally encoded signaling methods which as a result of both encoding redundancy and signaling efficiency operate at very high symbol error rates. These considerations lead to three areas of investigation as follows:

a. Determination of the mean-squared timing error of a given symbol synchronizer

b. Structure of an optimum or near-optimum symbol synchronizer and its performance characteristics, and

c. Influence of the systems parameters on the aforementioned synchronizers.

The purpose of this paragraph is to suggest a systematic approach toward the solution of these problems.

Structure and Major Components of Symbol Synchronizers

Figure 29 presents the block diagram of a symbol synchronizer. Although synchronizers vary considerably in structure, most of them reduced to the equivalent structure of Figure 29, having the following four major components:

a. An error detector whose output has a signal component proportional to the timing error which is modulated by a waveform dependent on the binary modulation. A noise component is also present.

b. A circuit which will be called a symbol decorrelator. Its role is to remove the binary modulation from the timing error signal. Its two inputs are the error detector output and the output of the integrate and dump circuit used for binary decisions. Its output is the timing error signal corrupted by noise.

c. A filter which determines the closed-loop response of the synchronizer.

d. A timing source such as a voltage controlled square-wave generator.

Some factors to be considered in designing these four components and their effect on the synchronizer performance are discussed in the following paragraphs.

In designing an error detector, a convenient point of view is to assume that the transmitted binary sequence is known exactly and to design the detector accordingly. The binary errors will introduce some deterioration in the detector performance; however, as will be shown in the next paragraph, the two problems; generate optimum error signals and minimize the degradation due to equivocation, can be separated.

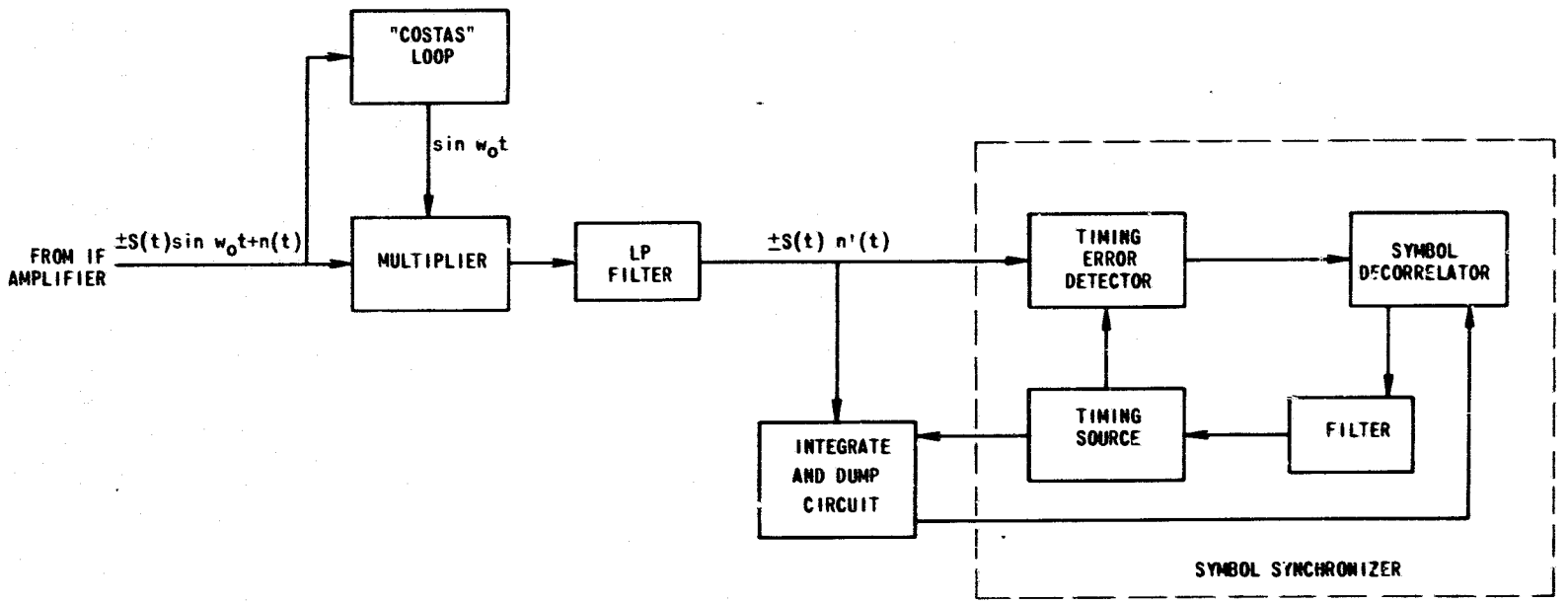


FIGURE 29. Typical Symbol Synchronizer

In practice the closed loop bandwidth of symbol synchronizers is much smaller than the symbol rate. As a result the error signal may be assumed constant during the symbol period, and the problem of generating a continuous error signal may be reduced to one which is more tractable, namely that of generating a discrete sequence of errors, one for each time segment of a duration equal to the symbol period. Depending on the approach used, the time segment is chosen as either coincidental with the estimated symbol period or half a symbol displaced.

In a practical system, which is limited to a bandwidth dictated by the symbol rate, the binary symbol video wave form is a smooth function. (A typical example used in practice is the raised cosine.) Under these conditions if the mean-squared timing error is not too large, i.e., if a linearized model is applicable, an optimum error detector can be derived. Let the timing error during a time segment T_i coincidental with the i^{th} symbol be ϵ_i. Let the error detector output generated for this period be $\epsilon_i + \eta_i$ where η_1 is a noise term. The optimum error detector is the one for which the variance of η_i is smallest. It will be shown presently that the optimum detector in this configuration of Figure 29 correlates the low-pass filter output with the first derivative of the binary video waveform corresponding to a +1.

Referring to Figure 29, the receiver IF amplifier output is converted to video through multiplication by the local carrier, generated in a phase-locked loop. For convenience the loop phase error is assumed negligible. Ignoring the double frequency terms, the result of the mixing process is the time function ± s(t) imbedded in noise n'(t) which will be assumed white and of power density N_o' watts/Hz. The ± sign denotes the binary modulation; thus, s(t) is the signal component if the transmitted binary sequence is all 1's. As mentioned previously, the binary sequence is assumed known, and for simplicity will be assumed to consist of all ones. This is equivalent to the assumption that the signal component is s(t). Since there is a time uncertainty in the system, which is to be resolved, more accurately, the multiplier output is $s(t + \epsilon) + n'(t)$, where ϵ is to be estimated. Under assumptions of small ϵ, a linear model may be used and the known function $s(t + \epsilon)$ expanded in a power series with all but the first two terms neglected, yielding for the receives signal

$$y_1'(t) = s(t) + \epsilon(t)\, \dot{s}(t) + n'(t)$$

Taking the symbol period associated with the i^{th} symbol as T_i and subtracting the known wave form s(t) from $y_1'(t)$:

$$y_1'(t) - s(t) = \epsilon_i \dot{s}(t) + n'(t) \quad t \epsilon T_i \tag{46}$$

and the task is to get the best estimate of ϵ_i from the right hand side function over the time period T_i. Under the assumption of no change in ϵ over the symbol period, ϵ_i is a constant. The continuous function on the right hand side may be replaced by a discrete set if T_i is broken down into k small segments of duration T/k during each of which s(t) may be taken as constant and over which the right hand side function is integrated, yielding the sequence

$$h_{1i} = \epsilon_i\, \dot{s}_1 \frac{T}{k} + n_{1i}; h_{2i} = \dot{\epsilon}_i\, \dot{s}_2 \frac{T}{k} + n_{2i}, \ldots . h_{ki} = \epsilon \dot{s}_k \frac{T}{k} + n_{ki}$$

where $\dot{S}_j$ is the value of s(t) during the j^{th} interval. n'(t) being white and of density N_o' the n_{ji}'s are independent and of variance $\frac{T}{k} N_o'$ where $\frac{T}{k}$ is the integration time. ϵ_i is obtained as a linear combination of the h_{ji}:

$$\hat{\epsilon}_i = \sum_{j=1}^{k} \ell_j h_{ji} \tag{47}$$

where ℓ_j is chosen to minimize $\sigma^2_{\Delta\epsilon}$, the mean-squared error in estimating ϵ_i:

$$\sigma^2_{\Delta\epsilon} = E\,(\hat{\epsilon}_i - \epsilon_i)^2 = E\left[\sum_{j=1}^{k} (\ell_j \epsilon_i \frac{T}{k} \dot{s}_j + \ell_j n_{ji}) - \epsilon_i\right]^2 =$$

$$\sigma_\epsilon^2 \left[\sum_{j=1}^{k} (\ell_{jk} \frac{T}{k} \dot{s}_j - \frac{1}{k}\right] + \frac{T}{k} N_o' \cdot \sum_{j=1}^{k} \ell_j^2$$

Differentiating with respect to ℓ_r and setting equal to zero yields:

$$\frac{j\sigma^2_{\Delta\epsilon}}{j\,\ell_r} = 2\sigma^2_\epsilon \left[\sum_{j=1}^{k} (\ell_{jk} \frac{T}{k} s_j - \frac{1}{k})\right] \cdot \frac{T}{k}\, \dot{s}_r + \frac{T}{h} N_o' \ell_r = O$$

or

$$\ell_r = \text{constant} \cdot \dot{s}_r$$

Increasing the number k of subintervals into which the symbol period was subdivided yields in the limit as the summation in Equation (47) becomes an integration, and with the constant taken as unity for convenience.

$$\hat{\epsilon}_i = \int_{T_i} \left[(y_1'(t) - s(t))\right] \dot{s}\ (t)\ dt$$

When the video pulse s(t) is symmetric around its midpoint (which is usually the case), its product with $\dot{s}(t)$ integrates to zero over the symbol period, and it is not necessary to subtract s(t) from $y_1'(t)$; hence:

$$\hat{\epsilon}_i = \int_{T_i} y_1'(t)\ \dot{s}(t)\ dt$$

$\hat{\epsilon}_i$ is the minimum variance estimate of timing error extracted from the i^{th} symbol under the assumption that this symbol is +1. It may be shown that if this assumption is correct, $\hat{\epsilon}_i$ is an unbiased estimate of ϵ_i, i.e., the expectation of $\hat{\epsilon}_i$ is ϵ_i, the timing error.

If the symbol assumption is not correct, the signal component of $y_1'(t)$ is $-s(t)$ instead of $s(t)$ and the expectation of $\hat{\epsilon}_1$ is $-\epsilon_1$. Thus a wrong guess at the binary symbol results in an error signal which will drive the loop away from a null.

An interesting special application of this derivation is represented by $s(t)$ being a sine wave. $\dot{S}(t)$ is a cosine wave, i.e., in quadrature and Equation (49) is the expression for the phase detector. Thus, the phase-locked loop phase detector is a minimum mean-squared error, timing error estimator and the quadrature signal is the best signal that can be used for tracking.

There might be instances in which the assumptions of this derivation are not valid. For example the linear approximation is not valid if S(t) is assumed to be a square wave. In such a case other techniques must be used in deriving the optimum error detector. In any event once an error detector is designed Equation (49) may be used to evaluate its performance. Specifically, the error detector specified by Equation (49) estimates the timing error with a variance which may be computed if S(t) and N_0 are known and linearity (small timing errors) is assumed. The Cranner/-Rao theorem gives a lower bound to the variance of the estimate of any parameter (in this case ϵ) in terms of the likelihood function. When the Cramer'-Rao bound is computed, it may be shown analytically that it is identical to the variance of the error detector given by equation (49) under the assumption of linearity. The ratio of the variance of the error estimate in any other detector to the Cramer'-Rao bound gives the signal-to-noise deterioration of that error detector. A ratio larger than one indicates a nonoptimum detector.

Symbol Decorrelator

It was shown in the previous paragraph that the output of the error detector is $\epsilon' = \pm \epsilon$, where the $\pm$ sign is associated with the ± 1 transmitted binary symbol respectively. If the transmitted symbol were known, multiplying ϵ' by it would remove the uncertainty. Since this is not the case, the problem of degradation due to equivocation may be approached from a number of points of view. First, the question of minimizing the degradation due to equivocation was solved in References 19 and 20 for the carrier tracking circuit, and there is every indication that the same approach is effective in this case. Briefly, since the best estimator of the timing error is its expectation, under symbol uncertainty this expectation is given by:

$$E(\epsilon) = E(\epsilon/+1)\, p(+1) + E(\epsilon/-1)p\,(-1) \tag{50}$$

where $E(\epsilon/\pm 1)$ are the expectations of ϵ derived under the hypothesis that ± 1 respectively was transmitted. As shown in the previous paragraph $E(\epsilon/\pm 1) = \epsilon'$ and $E(\epsilon/-1) = -\epsilon'$ of Figure 29; $p(\pm 1)$ is the a-posteriori probability of ± 1 symbol and is obtained from the binary decision circuit (in-phase correlator). Thus Equation (50) becomes

$$E(\epsilon) = \epsilon' p(+1) - \epsilon' p(-1) = \epsilon' \left[p(+1) - p(-1)\right] \tag{51}$$

It was shown in References 19 and 20 that the expression in the square bracket is the hyperbolic tangent of the output of the integrator used for binary decision at the end of the symbol period. If in order to avoid the function generator deriving the hyperbolic tangent, a simpler structure is used, its performance may be compared to the one defined by Equation (51) which being optimum serves as a convenient standard. In most cases a degradation factor may be computed by using a spectral convolution technique which yields the noise generated by the equivocation.

In a decision-directed feedback sequential decoder, all the unknown parameters, symbol timing included, are computed as functions of the symbol sequence hypothesized by the decoder. For such a system, the equivocation is removed, and the symbol decorrelator operating in conjunction with the optimum error detector discussed previously will have as its output:

$$\hat{\epsilon}_i = b_{oi} \int_{T_i} y_1'(t)\, \dot{s}(t)\, dt \qquad (52)$$

where $b_{oi} = \pm 1$ is the i^{th} symbol hypothesized by the sequential decoder.

Filter

The choice of the appropriate filter does not present any particular problem. Given the spectrum of the instability of the oscillator used for symbol-timing in the transmitter, a Wiener filter may be designed. If such a filter is too complex, one might compromise by accepting one with fewer poles. The mean-squared error may be readily derived in either case if the transfer characteristics of the error detector/symbol decorrelator combination is known.

Evaluation of Symbol Synchronizer Performance

In the previous paragraph it was shown that the various building blocks of a symbol synchronizer may be designed separately. When they are put together, the resultant performance in terms of mean-squared timing error is readily derived as follows:

Referring to Figure 29, the error detector output may be shown in most cases to be equivalent to a signal proportional to the timing error, imbedded in noise. The symbol decorrelator enhances this noise by some factor δ caused by the symbol error. The resultant equivalent circuit is shown in Figure 30 and given the oscillator instability spectrum, the mean-square timing error is readily derived. It will depend on the following factors:

S_γ = instability of the oscillator to be tracked

k = transfer characteristic of the error detector. The larger R, the better the error detectors.

δ = noise enhancement caused by symbol decorrelator

H(w) = transfer function of the filter.

Various configurations defined by different values of k_ϵ and H(w) result in different mean-squared timing errors, which may be used for comparison.

Recommendations

C&S recommends that the design and evaluation of symbol sychronizers be continued from both a theoretical investigation and hardware implementation standpoints.

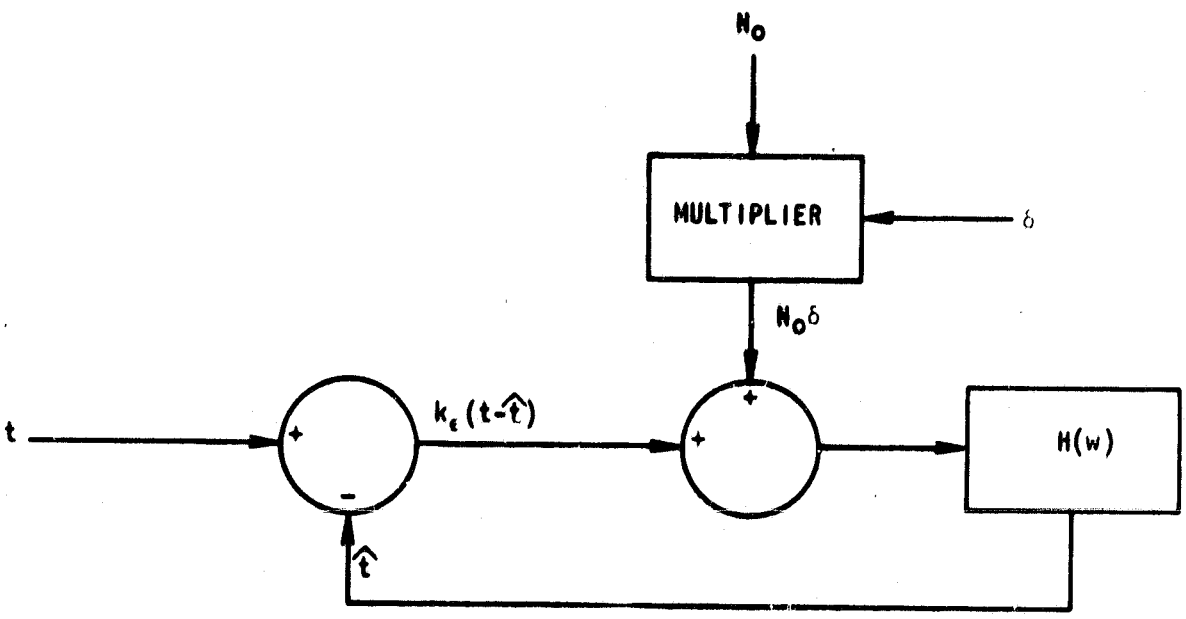


FIGURE 30. Equivalent circuit of Symbol Synchronizer.

Theoretical Investigation - Successful completion of this study should yield the following results:

a. Detailed structure of the optimum synchronizer.

b. Performance curves relating the mean square timing error to the carrier-power-to-noise density ratio, data rate and oscillator stability.

c. Development of a systematic approach to the evaluation of typical symbol synchronizers.

d. Derivation of curves as in (b) for the nonoptimum synchronizers, and a comparison with the optimum curves.

Implementation - Based on the findings in the theoretical studies, one or more of the following devices should be implemented.

a. The optimum synchronizer.

b. One or two of the suboptimum synchronizers, depending on the complexity of the optimum synchronizer and the departure from optimality of the suboptimum structures.

c. A computer implementation of the optimum synchronizer, consisting of a computer program and real-time injection and processing of digitized PSK signals.

Section 6

EARTH TERMINAL SEQUENTIAL DECODING EQUIPMENT

The purpose of this section is to discuss briefly, and in rather broad generality, earth terminal sequential decoding equipment. This discussion is primarily intended for those readers who are not familiar with sequential decoding but who may be interested in decoding hardware.

Equipment to perform the sequential decoding function might be divided in two distinct types: special devices and general purpose computers. The relative merit of each of these deserves some attention.

It is C&S's understanding that certain of the STADAN sites presently include a third generation general purpose computer. Some previous studies at C&S indicated that such a computer is definitely suited to the task of sequential decoding. While investigation of the specific computers existing at NASA sites is outside the scope of this report, some general comments may be of interest.

To consider the applicability and potential of third generation computers a survey was made of existing models. Based upon this survey, the following parameters were chosen as representative.

Memory Cycle: 1 microsecond
Memory Size: 4096 words
Word Length: 16 bits

A logical first question of interest to those presently possessing a third generation computer is, "Could this computer be used without additional hardware for sequential decoding?" The major problem area is that of performing parity check calculations. The decoder contains a replica of the convolutional encoder (i.e., a shift register of length on the order of 30-50) with the output of n modulo-2 sums of a specified subset of the register contents. Determining the modulo-2 sum of two words in a digital computer generally requires a large number of operations and severely restricts decoding speed. This becomes especially acute when the encoder length exceeds the computer word size. For this reason it is advantageous to perform parity checks externally. Assuming external parity checks and the parameters shown above, a cursory examination indicated the possibility of achieving decoding rates on the order of a few kilobits/sec.

An interesting possibility exists for channels with data rates of a lesser magnitude. The computer could be time shared and perform other functions simultaneously with sequential decoding. Details of this would be dependent upon the particular computer presently in use.

For sites which do not possess a computer, the question arises as to the desirability of obtaining a general purpose computer, one function of which would be to perform sequential

decoding, or to obtain a specific device for sequential decoding. This question is related to the total system functions to be served by the site.

Recommendations

a. The computers presently installed at STADAN stations should be examined in detail to estimate their sequential decoding capability.

b. NASA should take a general system oriented view of their receiving stations with regard to coded telemetry. For example, it may be much more economical to use a decoding technique less powerful than sequential decoding for on-line operation and reserve sequential decoding as a completely off-line capability.

Section 7

SPECIAL TOPICS

7.1 DIGITALLY IMPLEMENTED COHERENT RECEIVER UTILIZING SEQUENTIAL DECODING WITH DECISION-DIRECTED FEEDBACK

The two major subjects of this section are the methods whereby ground station receivers can be implemented digitally and the decision-directed feedback (DDF) method of sequential decoding.

The techniques of digital implementation were investigated because they are a prerequisite to a DDF system. However, during the study it became evident that digital implementation offers advantages even when considered by itself. These advantages are substantial when the ground station is used to process a variety of missions with different signaling characteristics. For such an application the receiver circuits following the IF amplifier would be reduced to a digitizer which converts the IF amplifier output to a binary bit stream followed by the digital processing circuitry which can be either a general purpose computer or special digital modules. These modules replace the customary receiver components, such as phase locked loops, bit synchronizers and mixers. The parameters of the particular mission are binary words inserted into a permanent memory.

Although analytical proof is not yet available, there is substantial evidence (References 21 and 22) that the DDF method of decoding represents an effective approach to eliminating the deterioration in performance which is present in system operating at high symbol error rates. The difficulties are particularly evident in systems utilizing sequential decoding since these have the lowest energy/symbol/noise density ratio of all binary signaling methods. Furthermore, as the rate (information bits/symbol) is decreased the problem becomes more acute, because the binary modulation on the received signal represents an uncertainty, which causes deterioration in the tracking performance of the carrier tracking phase locked loop and the bit synchronizer. The higher the uncertainty, i.e., the symbol error rate, the greater its negative effect. This is reflected in increased mean-squared tracking errors and corresponding deterioration in performance. The degree of deterioration depends on the system data rate. A high data rate system requires power which is far in excess of that required in the aforementioned tracking circuits. In these systems substantial effective power loss caused by symbol uncertainty can be afforded without loss in tracking accuracy. In low data rate systems the available power may just barely be sufficient for tracking, and any uncertainty might cause substantial tracking errors. From the above considerations it is clear that DDF is effective in low data rate systems. This is fortunate because the memory-size requirements in sequential decoders using DDF are about tenfold those of conventional methods. In a low data rate system where the required core memory is small, this increase is not too troublesome.

While DDF dictates digital implementation, the converse is not true. Digitizing a sequential decoding receiver is worthwhile, even if DDF is not used. In this latter case the core memory required is approximately twice that needed in an analog receiver.

This paragraph is subdivided into five major topics. First, the DDF approach is explained and contrasted with conventional sequential decoding. Next the basic concepts associated with digital processing, such as state variables and optimum recursive filtering and prediction, are presented. Subsequently a digital DDF sequentially decoding receiver is described, and its major component blocks are developed in detail. Next a comparative analysis of DDF versus conventional sequential decoder efficiency as a function of oscillator stability is given. Finally, a digital coherent sequentially decoding receiver without DDF is briefly described.

7.1.1 Sequential Decoder With Decision-Directed Feedback (DDF)

Figures 31 and 32 show a conventional and a DDF coherent sequentially decoding receiver respectively. It should be stressed that the implementation in actual decoders vary considerably from the ones shown in Figures 31 and 32. In particular, holding operations such as insertions into memory were omitted. However, the signals present in all implementations are substantially those indicated. The circuits are shown for illustrative purpose and are not necessarily the best available techniques.

In Figure 31 in-phase and quadrature-phase references are derived in a "Costas" circuit. The output of the in-phase multiplier $y_1'(t)$ is used for symbol synchronization as well as for deriving a metric. In order to derive an error signal ϵ for symbol tracking $y_1'(t)$ is integrated for the second half of the symbol period, held for half a period, and the absolute value is subtracted from the absolute value of the integral over the first half of the next symbol period. The result is the error signal associated with the particular transition. It is held for the duration of one symbol until the next error signal is generated, and it is applied to the square-wave generator through a filter which determines the synchronizer closed-loop response.

The signal $y_1'(t)$ is also integrated over one symbol period, and the resultant is applied to a function generator, whose output depends on the method used for generating a metric (hard decision or n-level quantization for example). The timing signals for the various integrations are obtained from the symbol synchronizer square-wave generator.

The salient feature of this implementation is that the carrier-tracking loop and the symbol synchronizer operate independently of the hypothesis. In the case of the carrier tracking loop, it is shown in Reference 23 and Appendix III that the operation is equivalent to a loop tracking the same carrier from which the binary modulation was removed, imbedded in white noise enhanced by a factor δ. Factor δ is an inverse function of the energy/symbol/noise density ratio, hence a direct function of the symbol error rate and reflects the deterioration in tracking ability caused by the symbol uncertainty.

In the case of the symbol synchronizer, the tracking mechanism is fundamentally the same. No analysis is yet available but all considerations (See Paragraph 5.4.5) point to the same results, namely a noise enhancement factor δ which is a monotonic function of the symbol error rate.

The operation of the DDF sequential decoder of Figure 32 is different in a number of ways. It is important to realize that the operations are performed digitally in non-real time, and the blocks shown in Figure 32 are merely equivalents of these computations. In particular, holding operations are not shown, and the sequence in which the computations are performed is important.

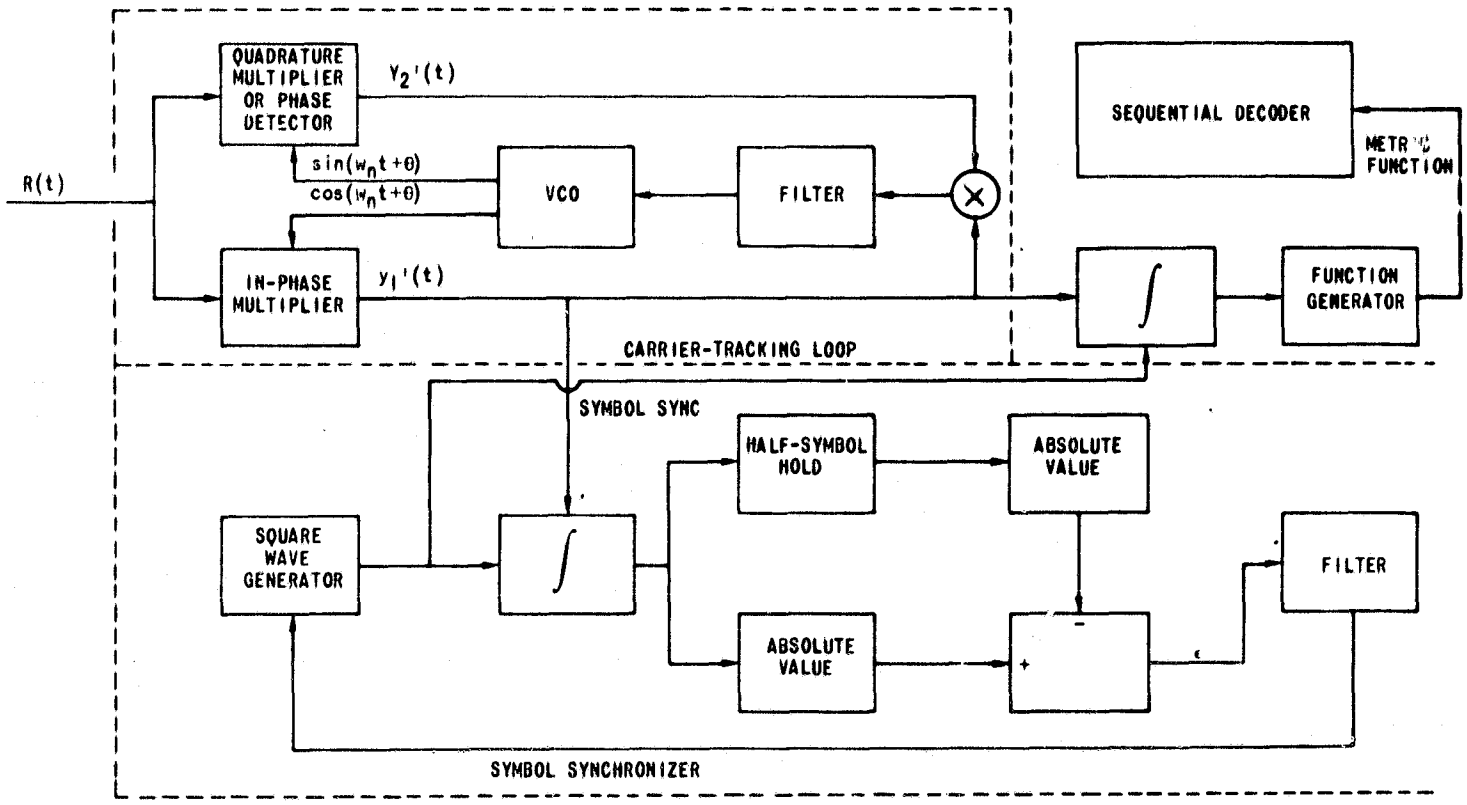


FIGURE 31. Conventional Coherent Receiver

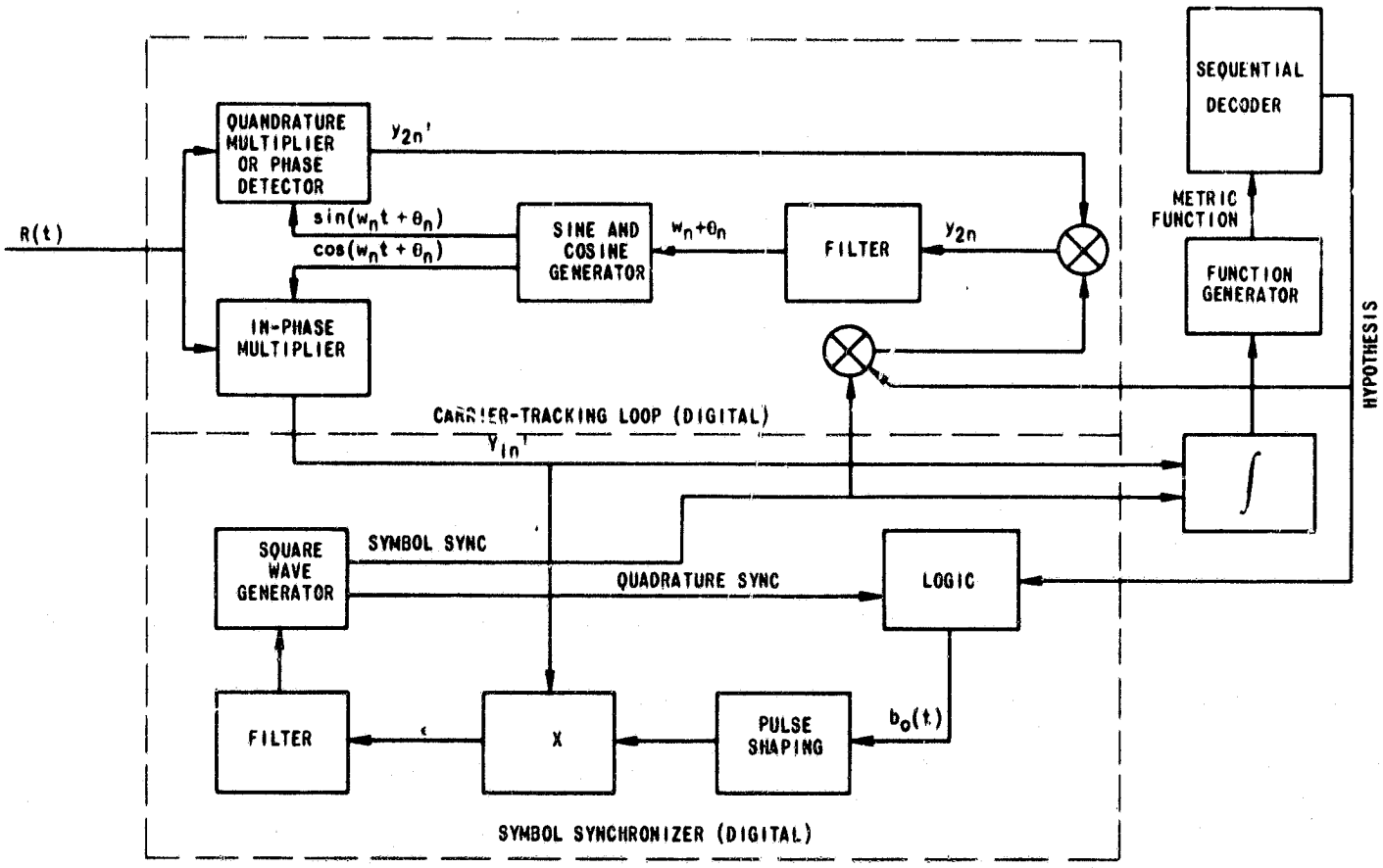


FIGURE 32. Decision-Directed Feedback Receiver

w_n and θ_n represent the phase-locked loop estimates of frequency and phase at the beginning of the n^{th} symbol corresponding to a node. They were derived utilizing the received signal R(t) up to that instant and are used to generate ($\cos w_n t + \theta_n$) during the time period of the next branch. The phase-locked loop is "coasting" on past information, and R(t) received during the time associated with the branch does not enter into generating $\cos(w_n t + \theta_n)$. This function is used to generate y_{1n}' the digital equivalent of y'(t), and hence the metric functions. One function per symbol is generated; the total number of functions equals the number of symbols per branch. The decoder receives these functions and selects a path, i.e., a sequence of symbols. The square-wave generator output is convolved with the hypothesized symbols. The result is a signal which is the decoder hypothesis of the binary antipodal modulation on R(t) for that branch. It multiplies y_{2n}' and the result is to remove the binary modulation from the quadrature multiplier output, if the hypothesis is correct. The signal y_{2n} thus obtained is used to update w_n and θ_n to their value at the next node. The hypothesis is also used in a similar manner to update the symbol synchronizer. The output of the logic in the symbol synchronizer is a square wave of one symbol duration, centered on the transitions, i.e., half a symbol out of phase with the symbol sync signals. The amplitude b_o of these waves equals +1 if the hypothesized symbols it straddles are +1 and -1, -1 if they are -1 and +1, and the amplitude is zero if the two symbols are the same $b_o(t)$ after some pulse shaping multiplies $y_1'(t)$ and the resultant is applied after filtering as a control voltage to the square-wave generator. The signal $b_o(t)$ insures that if the hypothesis is true, the expectation of the error signal ϵ is of the proper polarity to reduce the synchronizer timing errors regardless of the transmitted sequence.

By contrast with the conventional receiver of Figure 31, the operation of the two tracking circuits of Figure 32 depends on the hypothesis. If the hypothesis is correct, it follows from the foregoing that the degradation caused by the symbol error rate is absent. If the hypothesis is incorrect, the circuits do not track at all. The error voltages entering the filters are uncorrelated with the actual errors (they are keyed by a random symbol stream of ±1) and the loops drift. It can be argued that when the hypothesis is wrong, small tracking errors are not needed since the mean value of the integrator output utilized for deriving the metric is zero regardless of phase error or bit sync error.

According to References 21 and 22 there is substantial evidence that the performance of the DDF system is identical to the performance of a "clairvoyant" system, i.e., one in which the phase-tracking circuits knows the transmitted sequence of symbols.

Because a sequential decoder modifies its hypothesis in the process of search, the circuits of Figure 32 must be digital. They must be capable of retracing their steps to a "state" corresponding to the one they were in at some past node toward which the decoder is retracing its steps. From there they can then proceed to operate in a fashion dictated by the new hypothesis. The inputs R(t) must also be in a digital form so that inputs associated with past branches will be available. The whole operation may be likened to trick films of the silent era, e.g. the Keystone Cops chase the robbers (information bits). At some point they decide that at the previous intersection they took the wrong turn. In the happy never-never land of DDF they can turn the frame back to the one in question and proceed with the improved maneuver.

The following paragraphs describe how these circuits can be reduced to a set of digital logic operations.

7.1.2 Application of State Variable Techniques to the Synthesis of the Digital Equivalent of Analog Devices

The performance of all linear networks may be characterized by linear differential equations. The mathematician faced with the problem of determining the output of such a network at a given time t_1, must solve the associated differential equation. In order to do so he needs two basic elements of information: the initial conditions at some previous time t_o, and the forcing function or input f(t), defined at all times $t_o \leq t \leq t_1$. The output or result will have two components, the general solution caused by the initial conditions, and the particular solution caused by f(t); $t_o \leq t \leq t_1$. If the initial conditions and f(t) for $t_o \leq t \leq t_1$ are known the output at time t_1 is uniquely defined. The number of initial conditions associated with an electrical network equals the number of components - capacitors and inductors—having memory. It also equals the number of poles of the transfer function. Thus the set of initial conditions may be represented as a vector $\underline{X}_n$* In absence of further inputs to the network, the vector $\underline{X}_n$ associated with time t_n, uniquely defines the output at any future time. Furthermore since any one of the components of X_n may be defined as the output, the state of the network X_{n+1} at time t_{n+1} is uniquely defined by the state $\underline{X}_n$. The vector $\underline{X}_n$ which is a function of time is called the state variable. It is convenient to take as one of its components the output of the network, although this is not absolutely necessary, since any other variable — which by necessity is a unique function of the state variable vector — may be defined as the output.

In discrete implementations — ignoring for the moment the problem of the sampling rate necessary to insure negligible deterioration in performance — it is convenient to replace input signals, which in communication systems occupy a bandwidth much smaller than the carrier, by a sequence of discrete vectors having the components y_{11}, y_{21}, y_{12}, $y_{22} \ldots y_{1n}$, y_{2n} at times t_1, $t_2 \ldots t_n$ respectively. With this redefinition of symbols, the performance of the network may be expressed as:

$$\underline{X}_{n+i} = \underline{X}_{n+i,\,g}(\underline{X}_n) + \underline{X}_{n+i,\,s}(Y_{1,n+1}Y_{2,n+1};Y_{1,n+2},Y_{2,n+2}\cdots Y_{1,n+i}Y_{2,n+i} \qquad (53)$$

where each component of $\underline{X}_{n+i,\,g}$ is the general solution of the differential equation expressing that component considered as an output in terms of the initial conditions vector $\underline{X}_n$. Similarly the components of $\underline{X}_{n+i,\,s}$ are the special solutions of the same differential equations and are functions of the discrete representation of the input or forcing function. For i = 1, Equation (53) becomes:

$$\underline{X}_{n+1} = \underline{X}_{n+1,\,g}(\underline{X}_n) + \underline{X}_{n+1,\,s}(Y_{1,n+1}Y_{2,n+1}) \qquad (54)$$

The networks being linear so are the differential equations as well as their solutions. Denoting the components of the p-dimensional vector $\underline{X}_{n+1}$ by $X_{1,n+1}$, $X_{2,n+1} \ldots X_{p,n+1}$, the vector Equation (54) is given by:

$$X_{1,n+1} = \varphi_{11} X_{1,n} + \varphi_{12} X_{2,n} + \cdots \varphi_{1p} X_{pn} + \psi_{11} Y_{1,n+1} + \psi_{12} Y_{2,n+1}$$

---

*A dash under a symbol indicates a vector or matrix.

$$X_{2,n+1} = \varphi_{21} X_{1,n} + \varphi_{22} X_{2,n} + \cdots \varphi_{2p} X_{pn} + \psi_{21} Y_{1,n+1} + \psi_{22} Y_{2,n+1}$$

$$\vdots$$

$$X_{p,n+1} = \varphi_{p1} X_{1,n} + \varphi_{p2} X_{2,n} + \cdots \varphi_{pp} X_{pn} + \psi_{p1} Y_{1,n+1} + \psi_{p2} Y_{2,n+1}$$

or in matrix notation

$$\underline{X}_{n+1} = \underline{\varphi}\, \underline{X}_n + \underline{\psi}\, \underline{Y}_{n+1} \tag{55}$$

where $\underline{X}$ is a px1 vector, $\underline{\varphi}$ and $\underline{\psi}$ are pxp and px2 matrices respectively, and $\underline{Y}_{n+1}$ is a two-component vector. If the filter is low pass instead of bandpass the same expression results with $\underline{\psi}$ and $\underline{Y}_{n+1}$, of order p and 1 respectively. The matrices $\underline{\varphi}$ and $\underline{\psi}$ can be derived from the transfer function of the network.

Equation (55) is a recursive equation which tells how the state of the network is updated from the previous state and the next input. From a computational point of view it leads to economy in computations because in order to obtain the updated state variable $\underline{X}_{n+1}$ one need only know the last state variable $\underline{X}_n$ and the next input sample $\underline{Y}_{n+1}$. This is to be contrasted with computations based on the convolution integral;

$$X_r(t) = \int_{-\infty}^{t} Y(t-\tau)\, h_r(\tau)\, d\tau$$

which yields the discrete equivalent

$$X_{r,n+1} = \sum_{i=1}^{n} Y(t - \tau_i)\, h_r(\tau_i) \tag{56}$$

In Equation (56) one needs to know n samples of the input in order to compute a given state variable $\underline{X}_{n+1}$, and n can be a large number. In Equation (55) all information about the past and the effect of the past on future states resides in the state variable $\underline{X}_n$, having p components, and past inputs $\underline{Y}_n$, $\underline{Y}_{n+1}$, $\underline{Y}_{n-2}$, ... may be discarded without inpunity in computing future states $\underline{X}_{n+1}$, $\underline{X}_{n+2}$, etc. Herein resides the computational advantage as well as the reduced memory requirements of the state variable approach.

7.1.3 Application of State Variable Concepts to Filtering

It is convenient to discuss the problem of optimum linear filtering in terms of sampled representation:

Given a transmitted sequence of signals:

$$X_n, X_{n+1} \cdots\cdots\cdots\cdots X_{n-i} \cdots$$

and the corresponding received sequence

$$Y_n, Y_{n-1} \cdots\cdots\cdots\cdots Y_{n-i} \cdots$$

where $Y_i = X_i + n_i$, and the n_i are assumed to be samples of a white normal process, an estimate of X_n is sought as a linear combination of the received sequence:

$$\hat{X}_n = \sum_{i=0}^{\infty} a_i Y_{n-i}$$

such that the mean-squared error $E\,\epsilon^2$ is minimized:

$$E\,\epsilon^2 = E(X_n - \sum_{i=0}^{\infty} a_i Y_{n-i})^2 = E\,X_n^2 + \sum_{i,j=0}^{\infty} a_i a_j E(Y_{n-i} \cdot Y_{n-j}) + \sum_{i=0}^{\infty} a_i E(X_n Y_{n-i})$$

The various variables are assumed to have zero mean, yielding;

$$E\,\epsilon^2 = \sigma_{x_n}^2 + \sum_{i,j=0}^{\infty} a_i a_j \operatorname{cov}(Y_{n-i}; Y_{n-j}) + \sum_{i=0}^{\infty} a_i \operatorname{cov}(X_n Y_{n-i}) \qquad (57)$$

The problem consists of determining the constants a_i which minimize $E\,\epsilon^2$. According to Equation (57) these constants will be functions of the covariances associated with the various variables X_i and Y_i. It is noteworthy that regardless of what the joint distribution of the variables is, as long as these distributions yield the same covariances between the variables, they lead to the same mean-squared error and the same linear filter. It is convenient to assume that the variables stem from the multivariate normal distribution for a number of reasons. First, since the only statistical parameters of interest are the convariances, and since the covariances uniquely define the multivariate normal distribution, the distribution is uniquely defined by the relevant parameters. Second, the multivariate normal distribution was extensively studied and there are many results, which can be used to advantage. Third, the majority of the processes to be filtered, stem from the multivariate normal distribution.

The covariances associated with the transmitted sequence are given by:

$$\operatorname{Cov}(X_i, X_j) = E(X_i \cdot X_j) = E\,[X(t_i)\,X(t_j)] = \varphi(t_i - t_j)$$

where the last equality is based on the assumption that the process is stationary and $\varphi(t_i - t_j)$ is the autocorrelation function of the process X(t) whose sampled representation is the sequence X_i. The Fourier transform of $\varphi(t_i - t_j)$ is the power spectrum of the process.

A linear filter having the same power spectrum, excited by white noise, would have as its output a process X(t) having the covariance $\varphi(t_i - t_j)$. Since the filter is linear, it has a state-variable vector X associated with it, as discussed in the previous section, and according to Equation (55) the sequence of state variable is given by:

$$\underline{X}_{n+1} = \underline{\varphi}\, X_n + \underline{\psi}\, N_{n+1} \tag{58}$$

where the sequence $N_1, N_2 \ldots N_{n+1}$ of uncorrelated normal variates with zero mean is the discrete representation of the white noise input to the filter. The first component of the state variable vector will be taken as the transmitted sequence. Expression (58) is the discrete state-variable representation of the random process X(t). It follows from the foregoing, that any normal random process may be expressed in this form. The matricies $\underline{\varphi}$ and $\underline{\psi}$ are obtainable from the power-spectrum of the process (References 24 and 25). A continuous representation of the form

$$\dot{\underline{X}}(t) = \underline{M}\underline{X}(t) + \underline{P}N(t)$$

where M and P are pxp and 1xp matrices respectively (X is a p component vector) is also obtainable (Reference 26.)

The structure of the minimum mean-squared error linear filter may be derived by the application of the following property of multivariate normal distributions (Reference 27):

Given a sequence of variables $z_1, z_2, \ldots z_n$, having the multivariate normal distribution, the function $g(z_2, z_3, \ldots z_n)$ minimizing the variance of $[z_1 - g(z_2, z_3, \ldots z_n)]$ is the conditional expectation of z_1, i.e. $E(z_1/z_2, z_3, \ldots z_n)$. Furthermore, the conditional expectation is a linear combination of $z_2, z_3, \ldots z_n$.

A direct result of the above theorem is the minimum mean-squared error estimate $\hat{X}_{in}$ of any one of the components X_{in} at time t_n of the state variable vector $\underline{X}_n$ associated with the transmitted signal (included its first component X_1, which is the transmitted signal itself) is the conditional expectation of that component given the received sequence $Y_n, Y_{n-1} \cdots$

$$\hat{X}_{in} = E(X_{in}/Y_n, Y_{n-1} \ldots); \; i = 1, 2, \ldots p$$

where p is the number of components of X. These p equations may be expressed as a single vector equation:

$$\hat{\underline{X}}_n = E(\underline{X}_n/Y_n, Y_{n-1} \ldots) \tag{59}$$

The expression for the optimum filter given by Equation (59) may be extended to include optimum predictors. Assuming that $\hat{\underline{X}}_n$ in some configurations was obtained at time t_n from $Y_n, Y_{n-1}, \ldots$ as in Equation (59), the optimum filter would update $\hat{\underline{X}}_n$ and obtain $\hat{\underline{X}}_{n+1}$ on

the basis of the next element of information Y_{n+1} obtained at time t_{n+1}. By contrast, the optimum predictor would estimate $\underline{X}_{n+1}$ before Y_{n+1} is obtained, at time t_n. Taking expectations of both sides of Equation (58):

$$E(\underline{X}_{n+1}/Y_n, Y_{n-1} \cdots) = \varphi E(\underline{X}_n/Y_n, Y_{n-1} \cdots) + \underline{\psi} E(N_{n+1}/Y_n, Y_{n-1} \cdots) \tag{60}$$

The conditional expectation of N_{n+1} is zero prior to the reception of Y_{n+1} thus the output of the optimum predictor is obtained by substituting (7):

$$\hat{\underline{X}}_{n+1/n} = E(\underline{X}_{n+1}/Y_n, Y_{n-1} \cdots) = \underline{\varphi}\, \hat{\underline{X}}_n \tag{61}$$

where the subscript denotes that $\underline{X}$ at time n+1 is estimated at time n. Repeated utilizations of the procedure yields:

$$\hat{\underline{X}}_{n+1/n} = \varphi^i \hat{\underline{X}}_n \tag{62}$$

The optimum estimate of the state variable vector at some future time is obtained applying the present estimate, as initial conditions to the filter matrix $\underline{\varphi}^i$ representing the signal process and assuming that all future noise excitations are absent.

The estimates of the components $X_{i,n+1}$ of X_{n+1} may be obtained by considering the following equality in multivariate normal distribution (Reference 28):

$$E(X_{i,n+1}/Y_{n+1}, Y_n \cdots) = E(X_{i,n+1}/Y_n, Y_{n-1} \cdots) + a_i [Y_{n+1} - E(Y_{n+1}/Y_n, Y_{n-1} \cdots)]$$

where the constant a_i may be obtained from the covariances.

Since $Y_{n+1} = X_{1,n+1} + n_{n+1}$ where the samples n_i, n_{i+1}, ... are uncorrelated and have zero mean and $X_{1,n+1}$ is the first component of the state-variable vector, or the signal:

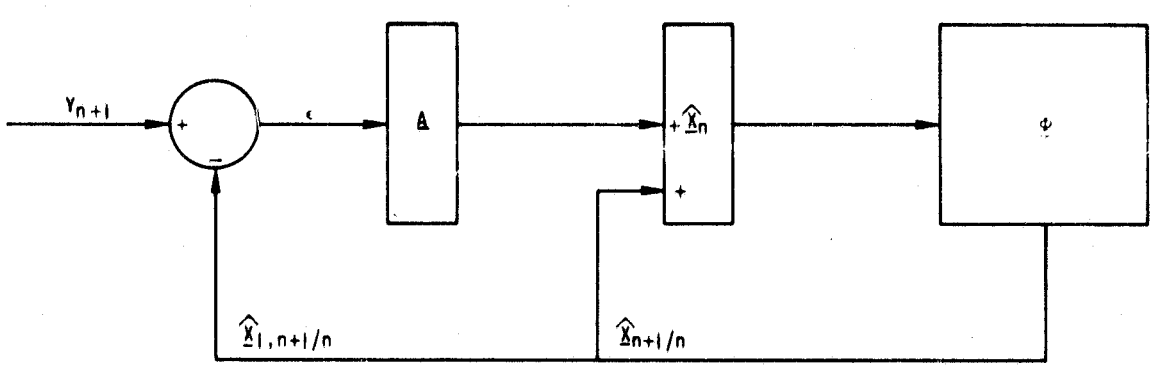
$$E(Y_{n+1}/Y_n, Y_{n-1} \cdots) = E(X_{1,n+1}/Y_n, Y_{n-1} \cdots) = \hat{X}_{1,n+1/n} \tag{63}$$

Equation (63) may be extended to all components of the state variable vector, hence the vector equation:

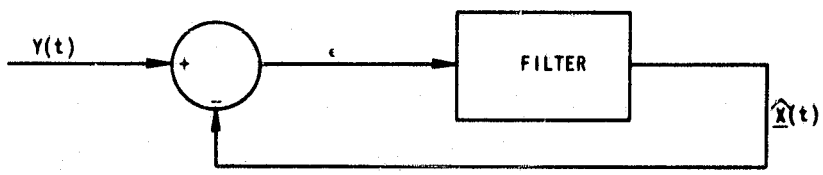
$$E(\underline{X}_{n+1}/Y_{n+1}, Y_n \cdots) = E(\underline{X}_{n+1}/Y_n, Y_{n-1} \cdots) + \underline{A}[Y_{n+1} - E(Y_{n+1}/Y_n, Y_{n-1} \cdots)]$$

where $\underline{A}$ is a p component vector obtainable from the covariance matricies. Substituting Equations (63) and (61):

$$\hat{\underline{X}}_{n+1} = \underline{\varphi}\, \hat{\underline{X}}_n + \underline{A}\, [Y_{n+1} - \hat{X}_{1,n+1/n}] \tag{64}$$



a) DISCRETE



b) CONTINUOUS

FIGURE 33. Recursive Filter

Equation (64) is the fundamental equation of the recursive or Kalman-Bucy filter (Reference 26) in its discrete form. The manner in which it operates is shown in Figure 33. The state variable vector is updated at each sampling time as follows: First, the last estimate $\hat{X}_n$ is multiplied by the matrix $\underline{\varphi}$, yielding the first term in Equation (64) which is the optimum prediction. The first component $X_{1,n+1/n}$ of this vector is the predicted value or expectation of Y_{n+1}, the received signal. This value is subtracted from the received signal, yielding the error in prediction, which when multiplied by $\underline{A}$ yields the second component in Equation (64). If the signal were turned off, the circuit would "coast". This is achieved by setting the error equal to zero in Equation (64), yielding Equation (62) which is the best prediction of X_{n+1}, X_{n+2} etc. based on signals up to Y_n.

As the sampling speed is increased the implementation tends toward the continuous shown in Figure 33 (b). In fact the discrete filter could have been obtained by deriving the optimum filter as a Wiener filter, obtaining a feedback representation which has the same closed-loop response as in Figure 33, and deriving its discrete equivalent in the manner described in Paragraph 7.1.2. The two approaches although different, yield the same result.

Noting that $X_{1,n+1/n} = \varphi_1 \underline{X}_n$ where $\underline{\varphi}_1$ is the first row of φ Equation (64) becomes

$$\hat{\underline{X}}_{n+1} = [\underline{\varphi} - \underline{A}\,\varphi_1]\,\underline{X}_n + \underline{A}Y_{n+1} \tag{65}$$

which is identical in form with Equation (55). Similarly, the response of any filter may be expressed in a feedback form. Thus Equation (55) may be expressed as

$$\hat{X}_{n+1} = [\varphi - \psi\,\varphi_1]\,X_n + \psi\,\epsilon_{n+1} \tag{66}$$

where $\underline{\varphi}_1\underline{X}_n$ is the predicted value of the first component of the state variable.

7.1.4 Analog-To-Digital Conversion

Figure 34 shows a typical network suitable for translating the received signal into digital words. The signal, which is received from either the IF amplifier or from tape, is at a nominal carrier frequency w_o. For each time interval T the vector components of the received signal are obtained at the end of the interval by correlating with the quadrature outputs of the reference oscillator. These components are digitized; thus, the digital network receives a continuous stream of pairs of words, $Y''_{1,m}$, $Y''_{2,m}$; $Y''_{1,(m+1)}$, $Y''_{2,(m+1)}$; $Y''_{1,(m+2)}$, $Y''_{2,(m+2)}$; ... where

$$Y''_{1,m} = \int_{(m-1)T}^{mT} R(t)\,\mathrm{Cos}\,w_o t\,dt;\quad Y_{2,m} = \int_{(m-1)T}^{mT} R(t)\,\mathrm{Sin}\,w_o t\,dt \tag{67}$$

The rate at which these pairs of words are generated — which will be referred to as the sampling rate — will depend on the subsequent digital circuitry. In the case of the sequential decoding receiver, it is dictated by the bit synchronizer. A resolution of one tenth of a symbol in the bit synchronizer results in a maximum loss of 10 log (1 - 1/20) $\approx$ 0.2 dB which is

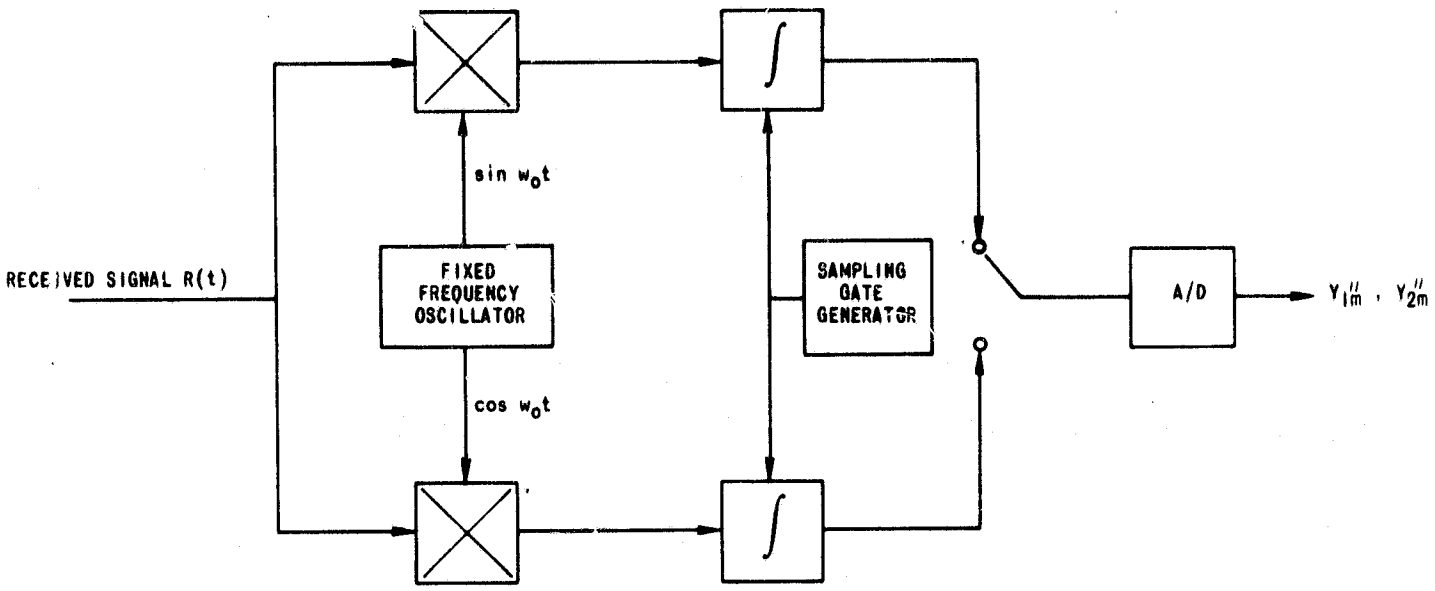


FIGURE 34. Analog-to-Digital Conversion

is acceptable. Thus ten samples per symbol seem adequate. All other circuits have bandwidths which permit much smaller sampling rates. For example the phase locked loop if it is designed to have a signal-to-noise ratio of 10 dB in the loop noise bandwidth, will have a bandwidth given by:

$$B_L = \text{Bit rate} \times \frac{E_b}{N_o} \times \frac{1}{10}$$

yielding for $E_b/N_o = 4$ dB for example and a rate 1/4 code

$$B_L = \frac{\text{Bit rate}}{4} = \frac{\text{Symbol rate}}{16}$$

resulting in a sampling rate which is 160 times the loop noise bandwidth.

7.1.5 Digitally Implemented Receiver Configuration

Figure 35 represents a block diagram of the digital DDF coherent sequentially decoding receiver shown in Figure 31 and described in Paragraph 7.1.1. This paragraph will merely describe the function of each block of Figure 35 with the detailed analysis and operation of these blocks discussed in Paragraph 7.1.6.

The sample vectors Y_{1m}'', Y_{2m}'' are generated by the A/D converter (Paragraph 7.1.4) at a nominal rate of ten vectors/symbol, and are inserted into the core memory. From there they are extracted as a function of the decoder search operation. Each time the decoder selects a branch (moving forward or backward) the corresponding sets of vectors are applied to the mixer. Here two things happen: First, the mixer takes each vector Y_{1m}'', Y_{2m}'' and rotates Y_{1m}'' Y_{2m}'' so that Y_{1m}'', Y_{2m}'' is expressed in terms of the phase-locked loop frequency reference w_n instead of the fixed-frequency oscillator. The resultant vectors Z_{1m}, Z_{2m} are the ones that would have been obtained if the reference oscillator used for sampling would have been the phase-locked loop VCO with in-phase and quadrature components used as in Figure 34 to derive the vector components. The sets of vectors Z_{1m}, Z_{2m} associated with a symbol are supplied to the symbol synchronizer. Here the vectors are used to derive the updated bit synchronization estimate consisting of the frequency estimate—how many vectors/symbol—and a phase estimate—a grouping of vectors belonging to the same symbol. The synchronizer is a tracking circuit based on the recursive filtering techniques of Paragraph 7.4.1. The error signal for updating the estimate is derived from the input vector sequence Z_{1m}, Z_{2m}, and is a function of the hypothesis, i.e., the assumed symbol sequence hence of the assumed type of symbol transition (0 to 1 or 1 to 0, etc.). The vectors Z_{1m}, Z_{2m}, belonging to any given symbol are also summed, and the resultant Y_{1n}', Y_{2n}' is applied to the correlator. This is a multiplier which takes each vector Y_{1n}', Y_{2n}' associated with a symbol and multiplies it with $H_n = +1$ or -1 for the hypothesis $+1$ or -1 for that symbol. If the hypothesis is correct, this operation has the effect of removing the antipodal modulation, and the resultant vector Y_{1n}, Y_{2n} is the one that would have been obtained if the transmitted signal would have been an unmodulated carrier.

The phase-locked loop is a digital recursive filter updating its state variable estimate once per binary symbol. Two components of this estimate are the carrier frequency w_n and phase θ_n.

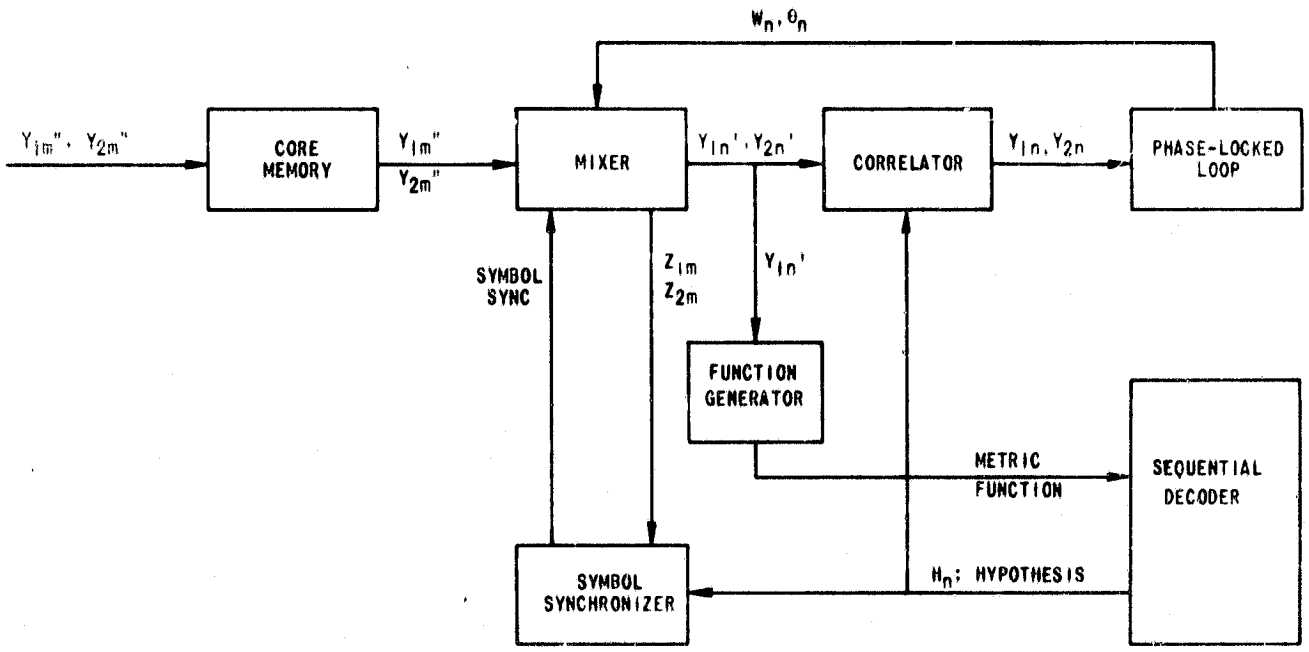


FIGURE 35. Digital Coherent Receiver with Decision-Directed Feedback Sequential Decoder

The sequential decoder selects a hypothesis on the basis of a set of metric functions (one for each branch of a node). These are computed as functions of the sequence Y_{1n} associated with that branch and of the sequences of binary symbols associated with each possible branch. The latter sequence is generated by the decoder. Having selected a branch on the basis of the metric, the associated sequence H_n of binary symbols is fed to the tracking circuits.

It is important to note a few salient features of the receiver:

a. For each possible branch the sequential decoder receives back a metric.

b. The selected branch "drives" the symbol synchronizer and phase locked loop forward or backward depending on the sense of the search operation.

c. The symbol synchronizer and phase-locked loop state variable estimates are functions of the hypothesis.

7.1.6 The Digital Equivalent of Some Typical Components of a Coherent Receiver

In Paragraphs 7.1.3 and 7.1.4 it was shown that the response of networks may be computed recursively in discrete form in terms of the state variable $\underline{X}_n$, and the next input Y_{n+1}, through the expression:

$$\underline{X}_{n+1} = \underline{\varphi}\,\underline{X}_n + \underline{\psi}\,\underline{Y}_{n+1} \qquad i = 2 \text{ for bandpass}$$

$$\text{px1} \quad \text{pxp} \quad \text{px1} \quad \text{pxi} \quad \text{ix1} \qquad i = 1 \text{ for low-pass}$$

where $\underline{\varphi}$ and $\underline{\psi}$ are matrices whose terms may be computed from the transfer function of the network. In this paragraph it will be assumed that the transfer function and thus $\underline{\varphi}$ and $\underline{\psi}$ are known. It will be shown that the concepts developed previously yield relatively simple digital implementations for the decision-directed feedback sequential decoder of Figure 35. For convenience a definition of the key symbols as well as their analog counter parts is given in Table 3.

Sine and Cosine Function Generator

In many networks such as correlators and phase detectors, the function sin θ and cos θ are utilized, where θ is itself the result of digital operations and is computed recursively. Such a recursive operation would yield expressions of the form:

$$\theta_i = \theta_{i-1} + \Delta\theta_i$$

In such an instance — since $\Delta\theta_i$ is small, cos θ_i and sin θ_i can also be computed recursively, and thus the use of memory tables is avoided (Reference 6):

$$\begin{aligned} \text{Cos } \theta_i &= \text{Cos } [\theta_{i-1} + \Delta\theta_i] \approx \text{Cos } \theta_{i-1} - \Delta\theta_i \text{ Sin } \theta_{i-1} \\ \text{Sin } \theta_i &= \text{Sin } [\theta_{i-1} + \Delta\theta_i] \approx \text{Sin } \theta_{i-1} + \Delta\theta_i \text{ Cos } \theta_{i-1} \end{aligned} \qquad (68)$$

One would begin by assuming some value for θ, e.g., $\theta = 0°$, Cos $\theta = 1$, Sin $\theta = 0$, and updating these values are more accurate estimates of θ are made available.

The operation is shown in Figure 36. Cos θ and Sin θ are in a memory. They are multiplied by the value of $\Delta\theta$ and the results added to the memory content, thus updating Cos θ and Sin θ.

Mixer

In Paragraph 7.1.7 it was shown that the sampling rate is dictated by the bit synsynchronizer resolution, and the remaining circuitry (e.g., the phase-locked loop) can operate at a much lower sampling rate. In order to reduce the computation requirements, it is convenient to combine the samples belonging to the same symbol. This is done in the mixer (Figure 35), and the resultant is applied to the phase-locked loop in which the lower sampling rate is acceptable. The components of the new sample vector associated with the nth symbol may be expressed as in Equation (67) by:

$$Y_{1n}' = \int_{(n-k)T}^{nT} R(t) \cos (w_n t + \theta_n) dt = \sum_{r=1}^{k} \int_{(n-r)T}^{(n-r+1)T} R(t) \cos (w_n t + \theta_n) dt \qquad (69)$$

$$Y_{2n}' = \int_{(n-k)T}^{nT} R(t) \sin (w_n t + \theta_n) dt = \sum_{r=1}^{k} \int_{(n-r)T}^{(n-r+1)T} R(t) \sin (w_n t + \theta_n)\, dt$$

At the beginning of each time segment of duration kT a frequency w_n is available from the digital loop. This frequency is used to derive Y_{1n}' and Y_{2n}'. If the frequency w_n is not derived from the phase-locked loop but is constant from period to period, the following derivations are still valid and the operations are the digital equivalent of a mixer, translating the sample vectors from a carrier frequency w_o to a carrier frequency w_n. For Y_{2n}' the transformation Equation (69) may be expressed as:

$$Y_{2n}' = \sum_{r=1}^{k} \int_{(n-r)T}^{(n-r+1)T} R(t) \sin (w_n t + \theta_n) dt = \sum_{r=1}^{k} \int_{(n-r)T}^{(n-r+1)T}$$

$$R(t) \sin [(w_n - w_o + w_o)t + \theta_n] dt =$$

$$\sum_{r=1}^{k} \int_{(n-r)T}^{(n-r+1)T} \{R(t) \sin w_o t \cos [(w_n - w_o)t + \theta_n)]$$

$$+ R(t) \cos w_o t \sin [(w_n - w_o)t + \theta_n]\} dt$$

TABLE 3. SIGNIFICANT DIGITAL PROCESSING PARAMETERS

Symbol	Analog Equivalent	Description
y_{1m}''	$\int_T R(t) \cos w_o t \, dt$	A/D converter outputs; sample vector components generated at rate 1/T or k vectors per binary symbol.
y_{2m}''	$\int_T R(t) \sin w_o t \, dt$	
Z_{1m}	$\int_T R(t) \cos (w_n t + \theta_n) dt$	Components of sample vector y_{1m}'', y_{2m}'' after transformation in order to express it in terms of w_n instead of w_o
Z_{2m}	$\int_T R(t) \sin (w_n t + \theta_n) dt$	
Y_{1n}'	$\int_{kT} R(t) \cos (w_n t + \theta_n) dt$	Sample vector components generated at the rate of one vector/binary symbol. Sum of k vectors Z_{1m}, Z_{2m} associated with a binary symbol.
Y_{2n}'	$\int_T R(t) \sin (w_n t + \theta_n) dt$	
Y_{1n}	$H_n \cdot Y_{1n}'$	Y_{1n}', Y_{2n}' modified by the hypothesis. If hypothesis is correct this is the vector that would exist if a string of all 1's were transmitted.
Y_{2n}	$H_n \cdot Y_{2n}'$	
k	T x symbol rate	Number of vectors out of A/D converter per binary symbol
τ	1/symbol rate	
T		Reciprocal of rate at which A/D converter generates vector samples.
w_o		Frequency of the fixed-frequency oscillator used to generate the vector samples from the received analog signal R(t).
w_n; θ_n		Frequency and phase estimates out of digital phase-locked loop. They are updated once per binary symbol.
R(t)		Received signal at output of IF amplifier. The noise is included.
H_n		A number equal to +1 or -1. The sequential decoder hypothesis that the n'th symbol equals H_n.

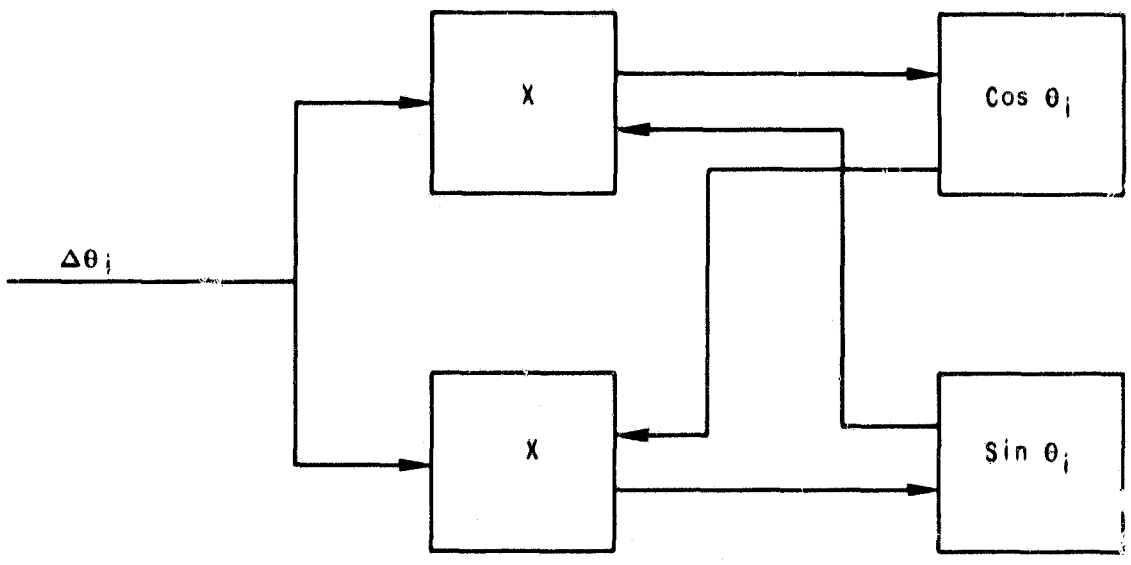


FIGURE 36. Sine and Cosine Function Generator

If $w_n - w_o \ll \frac{1}{T}$, $\cos (w_n - w_o)t$ may be assumed constant during the integration time, yielding

$$Y_{2n}' = \sum_{r=1}^{k} \left\{ \cos \left[(w_n - w_o)(n-r)T + \theta_n \right] \int_{(n-r)T}^{(n-r+1)T} R(t) \sin w_o t \, dt \right.$$

$$\left. + \sin \left[(w_n - w_o)(n-r)T + \theta_n \right] \int_{(n-r)T}^{(n-r+1)T} R(t) \cos w_o t \, dt \right\}$$

Substituting Equation (67)

$$Y_{2n}' = \sum_{r=1}^{k} \left\{ Y_{2,n-r+1}'' \cos \left[(w_n - w_o)(n-r)T + \theta_n \right] \right.$$

$$\left. + Y_{1,n-r+1} \sin \left[(w_n - w_o)(n-r)T + \theta_n \right] \right\} \tag{70}$$

Similarly

$$Y_{1n}' = \sum_{r=1}^{k} \left\{ Y''_{1,n-r+1} \cos \left[(w_n - w_o)(n-r)T + \theta_n \right] - Y_{2,n-r+1} \sin \right.$$

$$\left. \left[(w_n - w_o)(n-r)T + \theta_n \right] \right\} \tag{71}$$

Or

$$Y_{1n}' = \sum_{r=1}^{k} Z_{1,n-r+1}, \quad Y_{2n}' = \sum_{r=1}^{k} Z_{2,n-r+1} \tag{72}$$

where

$$Z_{1,n-r+1} = Y''_{1,n-r+1} \cos \left[(w_n - w_o)(n-r)T + \theta_n \right] - Y''_{2,n-r+1} \sin$$

$$\left[(w_n - w_o (n-r)T + \theta_n \right] \tag{73}$$

$$Z_{2,n-r+1} = Y''_{2,n-r+1} \cos \left[(w_n - w_o)(n-r)T + \theta_n \right] + Y''_{1,n-r+1} \sin$$

$$\left[(w_n - w_o)(n-r)T + \theta_n \right]$$

The trigonometric terms in Equation (73) may be generated recursively, as previously described.

Phase-Locked Loops

A number of authors (References 29 and 30) have pointed out that phase-locked loops tracking phase variations which are random processes, when operating in the linear region may be regarded as filters having as their signal input the phase process. Thus the phase modulation $\theta(t)$ of the transmitted signal (be it deliberate, i.e. carrying information, or caused by oscillator instabilities), may be regarded as originating from a network excited by random noise in a manner described in Paragraph 7.1.3, and has a state-variable vector θ associated with it:

$$\underline{\theta}_{n+1} = \underline{\varphi}\,\underline{\theta}_n + \underline{\psi}\,N_n$$

where the first component of θ will be assumed to be the transmitted phase.

The phase-locked receiver regarded as a filter will have a state variable $\underline{X}$ associated with it and a response similar to Equation (66):

$$\underline{X}_{n+1} = \underline{C}X_n + \underline{D}\epsilon_{n+1} \qquad (74)$$

where $\underline{C}$ and $\underline{D}$ are pxp and px1 matricies, p is the number of poles of the closed-loop response, and ϵ_{n+1} is the phase detector output.

Equation (74) is valid even if the loop response is not matched to the phase disturbance If the loop is a matched filter, $\underline{C}$ and $\underline{D}$ are obtained from the process matrices $\underline{\varphi}$ and $\underline{\psi}$. (For details see Reference 24). In this latter case the phase-locked loop is an optimum estimator and if the output is removed it is also an optimum predictor of all state variable components (phase, frequency, etc).

In paragraph 7.1.6, it is pointed out that the sample vector components Y_{1n}, Y_{2n} entering the digital phase-locked loop are expressed in terms of frequency estimate w_n derived in the loop. For the sake of generality and ease of presentation, it will be assumed in the following discussion that the sample vector is expressed in terms of a carrier frequency which is independent of the loop, such as the frequency w_o of the fixed frequency oscillator of Figure 24. The manner in which the operations are modified when the mixer uses the phase-locked loop output frequency w_n to combine the samples of a symbol is described at the end of this paragraph.

The operations to be performed on each sample vector $Y_{1,n+1}$, $Y_{2,n+1}$ are shown in Figure 32. A permanent memory holds the p^2 words of the $\underline{C}$ matrix and the p words of the $\underline{D}$ matrix. These matrices uniquely determine the loop response. A temporary memory holds the p words of the state variable $\underline{X}$. The first component of $\underline{X}$ is the phase θ. In addition it holds the two words of Sin θ and Cos θ and the two components Y_1 and Y_2

of the next received signal vector. For each sample the following operations are performed:

a. Utilizing the last value $\underline{X}_n = X_{n1}, X_{n2}, \ldots X_{np})$ from the temporary memory and C from the permanent memory:

$$U_1 = \sum_{i=1}^{p} C_{1i} X_{ni}; \quad U_2 = \sum_{i=1}^{p} C_{2i} X_{ni}; \ldots \quad U_p = \sum_{i=1}^{p} C_{pi} X_{ni}$$

or

$$\underline{U} = \underline{C}\underline{X}_n$$

is computed. The first component $U_1 = \hat{\theta}_{n+1/n}$

b. $\Delta\hat{\theta}_{n+1/n} = \hat{\theta}_{n+1/n} - \hat{\theta}_n$ is computed; where $\hat{\theta}_n$ is the first component of $\underline{X}_n$.

c. $\text{Sin}\ \hat{\theta}_{n+1/n} = \text{Sin}\ \hat{\theta}_n + \Delta\hat{\theta}_{n+1/n} \cdot \text{Cos}\ \hat{\theta}_n$

$$\text{Cos}\ \hat{\theta}_{n+1/n} = \text{Cos}\ \hat{\theta}_n + \Delta\hat{\theta}_{n+1/n} \cdot \text{Sin}\ \hat{\theta}_n$$

where $\text{Sin}\ \hat{\theta}_n$ and $\text{Cos}\ \hat{\theta}_n$ are obtained from the temporary memory. The left-hand sides are the sines and cosines at times t_{n+1} if there were no error voltage (predicted values).

d. $$\text{Sin}\ \epsilon_{n+1} \approx \epsilon_{n+1} = \text{Sin}\ \hat{\theta}_{n+1/n}\, Y_{1,n+1} + \text{Cos}\ \hat{\theta}_{n+1/n}\, Y_{2,n+1} \tag{75}$$

ϵ_{n+1} is the sampled and quantized phase-detector output.

e. Utilizing $\underline{D} = (D_1, D_2, \ldots D_p)$ from the permanent memory:

$$V_1 = \epsilon_{n+1} D_1; \quad V_2 = \epsilon_{n+1} D_2 \ldots V_p = \epsilon_{n+1} D_p$$

or

$$\underline{V} = \underline{D}\epsilon_{n+1}$$

is computed.

f. $$X_{1,n+1} = U_1 + V_1; \quad X_{2,n+1} = U_2 + V_2 \ldots X_{p,n+1} = U_p + V_p$$

or

$$\underline{X}_{n+1} = \underline{C}\underline{X}_n + \underline{D}\epsilon_{n+1}$$

is computed.

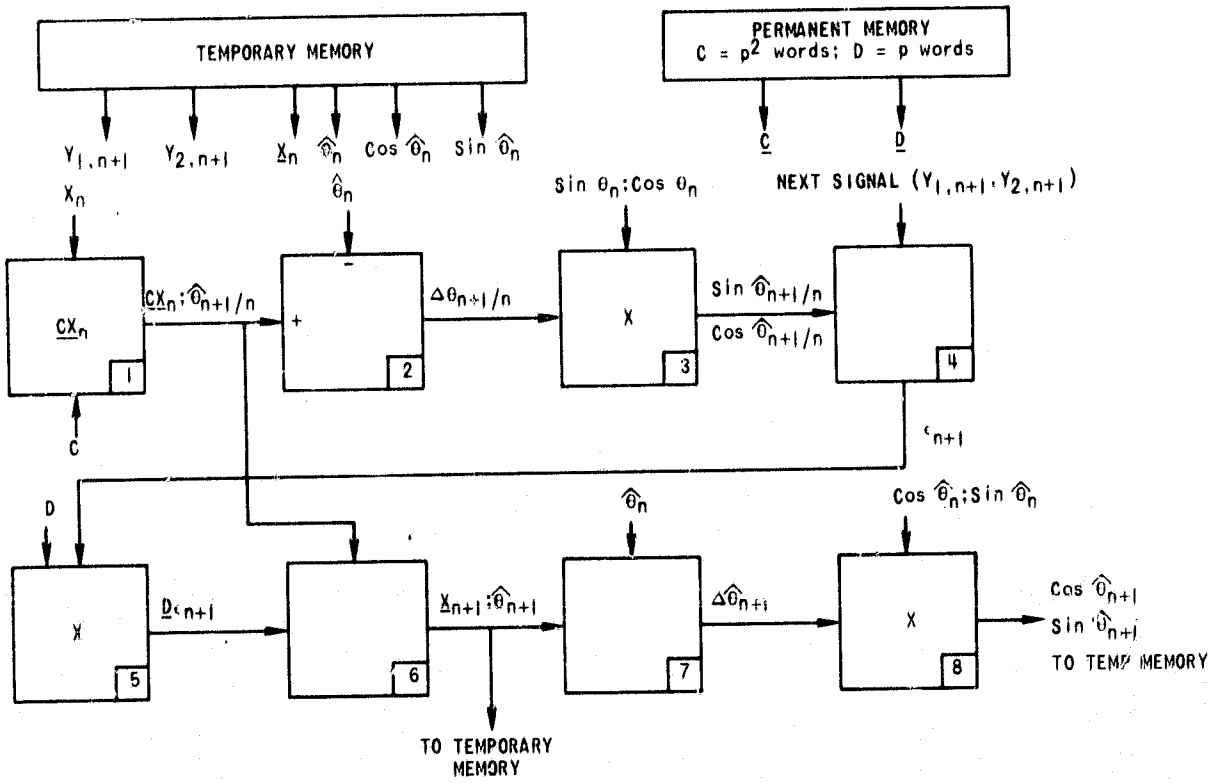


FIGURE 37. Digitized Phase-Locked Loop

This is the updated state variable, which is inserted into the temporary memory, the first component is $X_{1,n+1} = \theta_{n+1}$.

g. The phase increment is computed:

$$\Delta \hat{\theta}_{n+1} = \hat{\theta}_{n+1} - \hat{\theta}_n$$

h. The associated new values of the sine and cosine are obtained:

$$\text{Sin } \hat{\theta}_{n+1} = \text{Sin } \hat{\theta}_n + \Delta\hat{\theta}_{n+1} \text{ Cos } \hat{\theta}_n$$

$$\text{Cos } \hat{\theta}_{n+1} = \text{Cos } \theta_n + \Delta\theta_{n+1} \text{ Sin } \hat{\theta}_n$$

and inserted into the temporary memory.

Since the vector input to the phase locked loop is relative to the phase of the reference oscillator of frequency w_o used for quantization (See Paragraph 7.1.1), the angle $\hat{\theta}$ is an estimate of the signal phase relative to the phase of the same reference oscillator. Expressing the signal component of the receiver input prior to quantization in terms of the reference oscillator by:

$$s(t) = S \cos [w_o t + \theta(t)]$$

where the reference oscillator outputs are Cos ($\omega_o t$) and Sin ($\omega_o t$), the sequence $\hat{0}_n$, 0_{n+1}... is the sample representation of the estimate of $\theta(t)$.

From Equation (69) it is seen that when the mixer utilizes the phase-locked loop frequency w_n, in order to combine the samples of a symbol, Y_{2n}' is the sampled phase-detector output, and identical to ϵ_{n+1}. Under these conditions steps (c) and (d) become superfluous and are replaced by the computation indicated by Equation (70).

Correlator

Referring to Equation (70) it is seen that the phase-locked loop error voltage is computed from $Y_{1,n+1}$ and $Y_{2,n+1}$. These are given by:

$$\begin{aligned} Y_{1,n+1} &= H_{n+1} Y'_{1,n+1} \\ Y_{2,n+1} &= H_{n+1} Y'_{2,n+1} \end{aligned} \qquad (71)$$

where $H_{n+1} = \pm 1$ is the hypothesis generated by the decoder, that the $n+1^{th}$ symbol is +1 or -1. The operations in Equation (71) are the discrete equivalent of obtaining y_{2n} from y'_{2n} in Figure 38 (b).

Symbol Synchronizer

The method of designing a symbol synchronizer which performs well at high symbol-error rates is discussed in Paragraph 5.4.5. This paragraph merely outlines how a symbol synchronizer is digitized once its structure is given.

Figure 38(a) shows a typical (nonoptimum) symbol synchronizer which has the salient elements of all such devices. The received signal is multiplied with the in-phase output of the phase-locked loop VCO. The resultant is integrated for the first half of the symbol period, held for half a period and the absolute value subtracted from the absolute value of the integral over the second half of the symbol period. The result of the subtraction is the error associated with the particular symbol. The timing gates for the intergration are derived from the square-wave generator which generates two gates for each symbol period. An error voltage is developed at the end of each symbol period, as described, and is held for one symbol period. The sequence is applied to the square-wave generator through a filter.

The circuit is a closed-loop tracking system whose response is determined by the filter response. As such it has a digital realization (Figure 38(b)) based on estimating the associated state variables previously described.

According to equations (69) and (72) Z_{1m} is the sampled version of $Y_1'(t)$ and is generated at the sampling rate, which is of the order of ten samples/symbol. Two components of the state variable vector $\hat{X}_n$ are frequency expressed as the number of samples/symbol and phase, expressed as a grouping of the Z_{1m}'s belonging to the same symbol. The group is subdivided into two subgroups over which the summations of Figure 38(b) are performed. These summations are the digital equivalents of the intergrations in Figure 38(a). The matrices A and 0 are the digital equivalents of the filter, as discussed in Paragraph 7.1.4.

The reason for taking absolute values in the two circuits of Figure 38 is that the sequence of binary symbols is not known, and this operation removes the ambiguity. In Section 5 it was shown that in a decision directed feedback system, the optimum error signal associated with a symbol is given by:

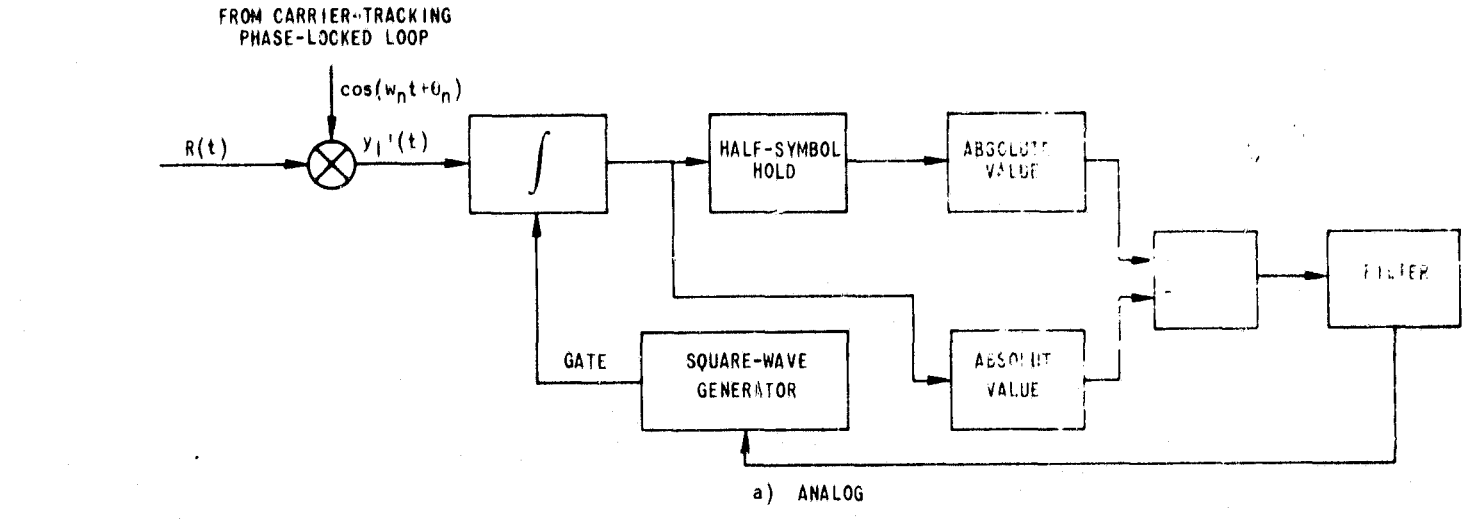
$$\epsilon = b_o \int_{\tau} y_1'(t)\,\dot{S}(t)\,dt \tag{72}$$

where $b_o = \pm 1$ is the binary symbol hypothesized by the decoder, τ is the symbol period, and $\dot{S}(t)$ is the first derivative of the symbol video pulse shape.

The digital equivalent of this expression is

$$\epsilon = b_o \sum_{i=1}^{k} \dot{S}_i\, Z_{1i} \tag{73}$$

where the summation is over the samples derived from a particular symbol. Equation (73) represents the computations replacing in a DDF system the circuits used in Figure 38(b) for generating ϵ.



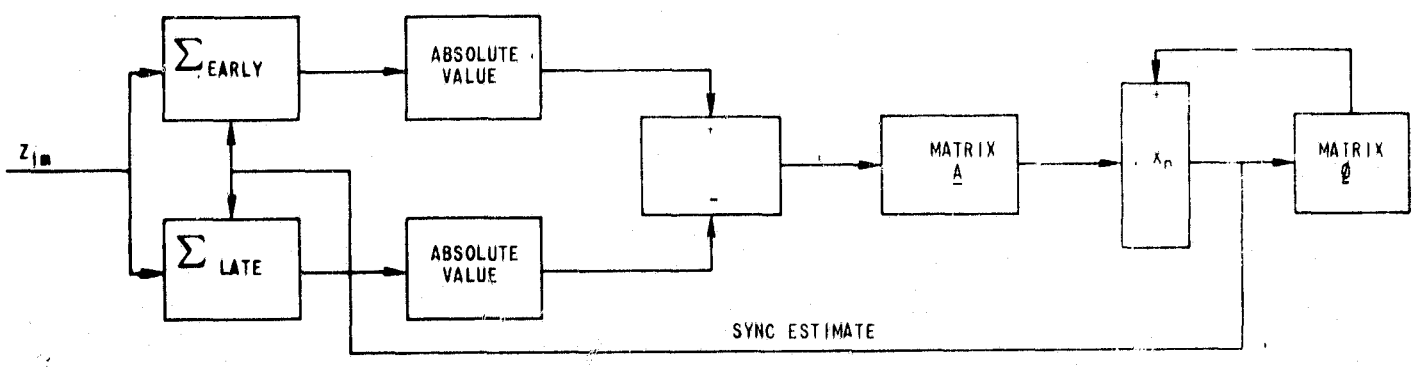


FIGURE 38. Bit Synchronizer and Its Digital Equivalent

7.1.7 Performance of the Digitally Implemented Coherent Receiver Utilizing DDF Decoding

The receiver implementation described in previous paragraph has five major areas of potential improvement relative to more conventional systems:

a. Reduction in phase and symbol timing errors through DDF

b. Reduction in phase and symbol timing errors through matched filter design of the tracking circuits

c. The elimination of symbol ambiguity

d. Optimization through adaptive operation

e. Optimization of carrier-reinsertion circuits.

Improvements (a) and (c) are a direct result of DDF implementation; the other three are present in a digitally implemented system, even if DDF is not employed.

The problem of evaluating the effect of oscillator instabilities on efficiency was discussed in Appendix III. To recapitulate the salient findings, the effect of symbol error (or symbol uncertainty) on the phase tracking circuit performance was obtained in terms of a noise enhancement factor δ as a function of E_S/N_O (energy per symbol/ noise density) in Reference 20. The resultant phase error $\sigma_{\varphi}{}^2$ was derived in Appendix III as a function of factor S, which contains δ, given by:

$$S = \frac{S_\tau \, w_o}{\frac{P}{N_o \delta}} \tag{74}$$

where

S_τ = oscillator instability. A discussion of its causes and method of measurement is given in Paragraph 7.1.9.

w_o = carrier frequency associated with S (rad/sec)

P = carrier power, watts

N_o = thermal noise density, watts/Hz, single-sided

δ_2 = noise-enhancement factor

Factor $\sigma_{\varphi}{}^2$ was derived as a function of S for both a matched filter (optimum) loop and a second order loop. Reference 11 derived the value of R_{comp} for various E_S/N_O and $\sigma_{\varphi}{}^2$.

Combining these three sets of curves yields a plot of signal efficiency expressed in E_b/N_o (energy/information bit/noise density ratio) as a function of

$$K = S_\tau \frac{2\pi w_o}{R_b} = S_\tau \frac{f_o}{R_b} \tag{75}$$

where R_b = information bit rate.

The result is shown as the two conventional curves of Figures 39 and 40. There are two significant conclusions that can be drawn from these curves. First, the performance deterioration can be substantially reduced through the use of higher-order loops. Second, the effect of oscillator instabilities (high value of K) are more severe for the rate 1/4 code than for the rate 1/2 code. This may be attributed to higher symbol error rate as well as lower signal power requirements associated with the rate 1/4 code.

As mentioned previously, there is experimental evidence (References 21 and 22) that DDF results in a mean-squared phase error equal to the one in a "clairvoyant" system, namely a system from which the symbol uncertainties were removed. The resultant performance may be obtained by the procedure described previously, except that the deterioration factor δ representing the symbol uncertainty is taken equal to 1 (no uncertainty). The result is shown as the two DDF curves each of Figures 39 and 40.

It must be stressed that the evidence of References 21 and 22 on which these curves are based is not conclusive, and these results should be verified experimentally. With this reservation comparing the DDF and conventional implementations, it may be concluded that for rate 1/2 codes DDF is hardly worthwhile. For rate 1/4 codes (and higher) DDF may result in appreciable improvement for high values of K. It is also noteworthy that implementing higher order loops results in more improvement than DDF implementation. If the system parameters are such that they cause timing errors in the symbol synchronizer and loss in efficiency, DDF will result in further improvement in this area, too. This is because the whole mechanism of synchronization as well as the effect of symbol uncertainties is analogous for the phase and symbol tracking loops.

The curves showing the performance of the conventional systems assume that the carrier tracking loop is a Costas circuit. These circuits have a phase ambiguity of 180°; namely the closed loop response has two stable points 180° apart. As a result even if the loop is at the correct stable point, it can "flip over" as a result of thermal noise; and unless special measures are taken, a symbol is undistinguishable from its complement. All special measures result in loss of efficiency, and for some of these the loss would be particularly severe for convolutionally encoded signaling. For example, differentially coherent PSK is approximately 3 dB less efficient than PSK at $E_S/N_o \approx 0$ dB. DDF eleiminates the symbol ambiguity by eliminating the need for a carrier tracking circuit with two stable points. The need for a carrier tracking circuit with two stable points. The resultant saving in power depends on the method that would have been used to remove the ambiguity.

In addition to the sources of improvement just discussed, there are a number of areas of potential improvement which are made possible by digital implementation. All these have one feature in common, namely that analog implementation is possible on paper; but actual

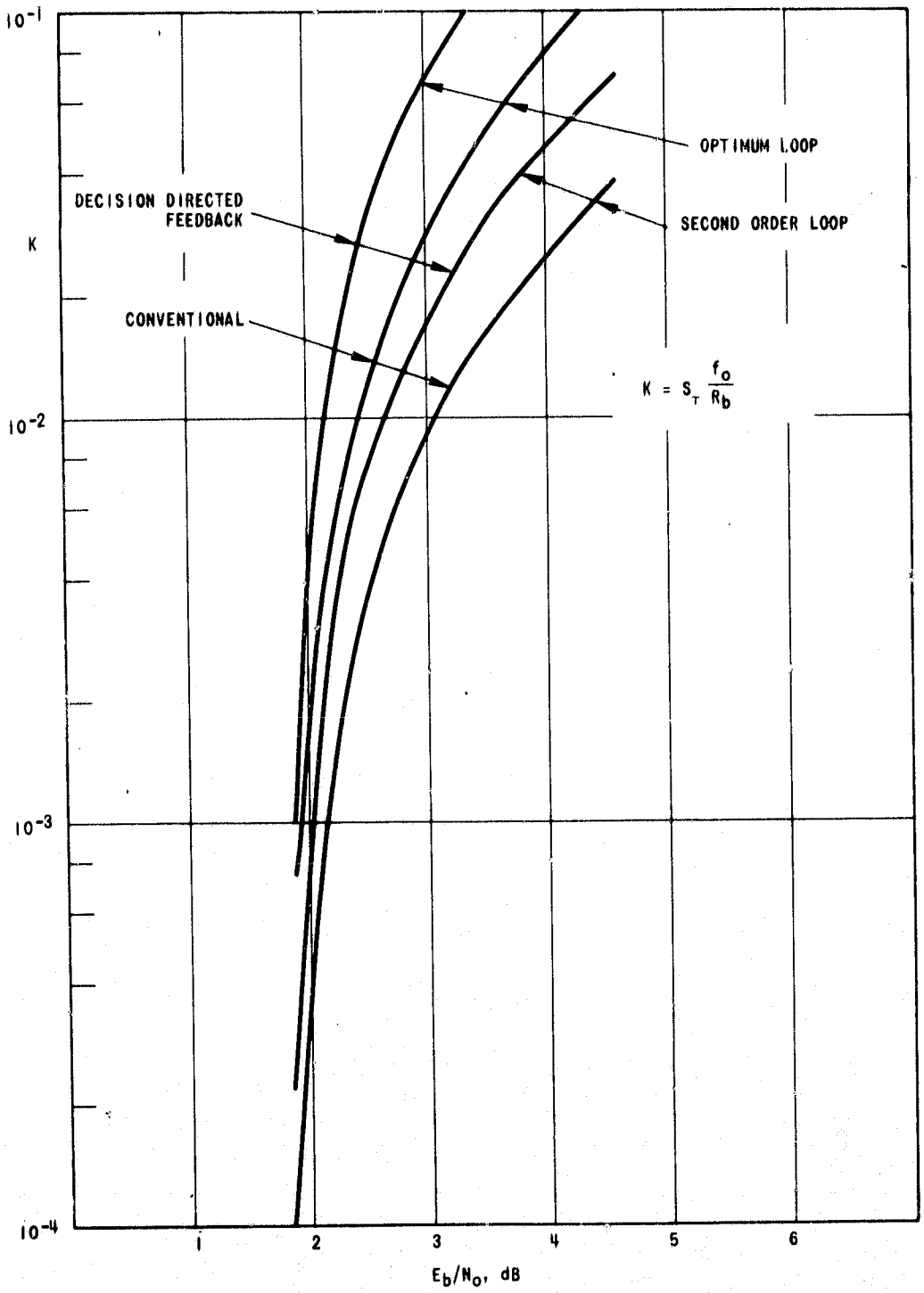


FIGURE 39. Loss of Efficiency vs. K. Rate = $R_{comp} = 0.25$.

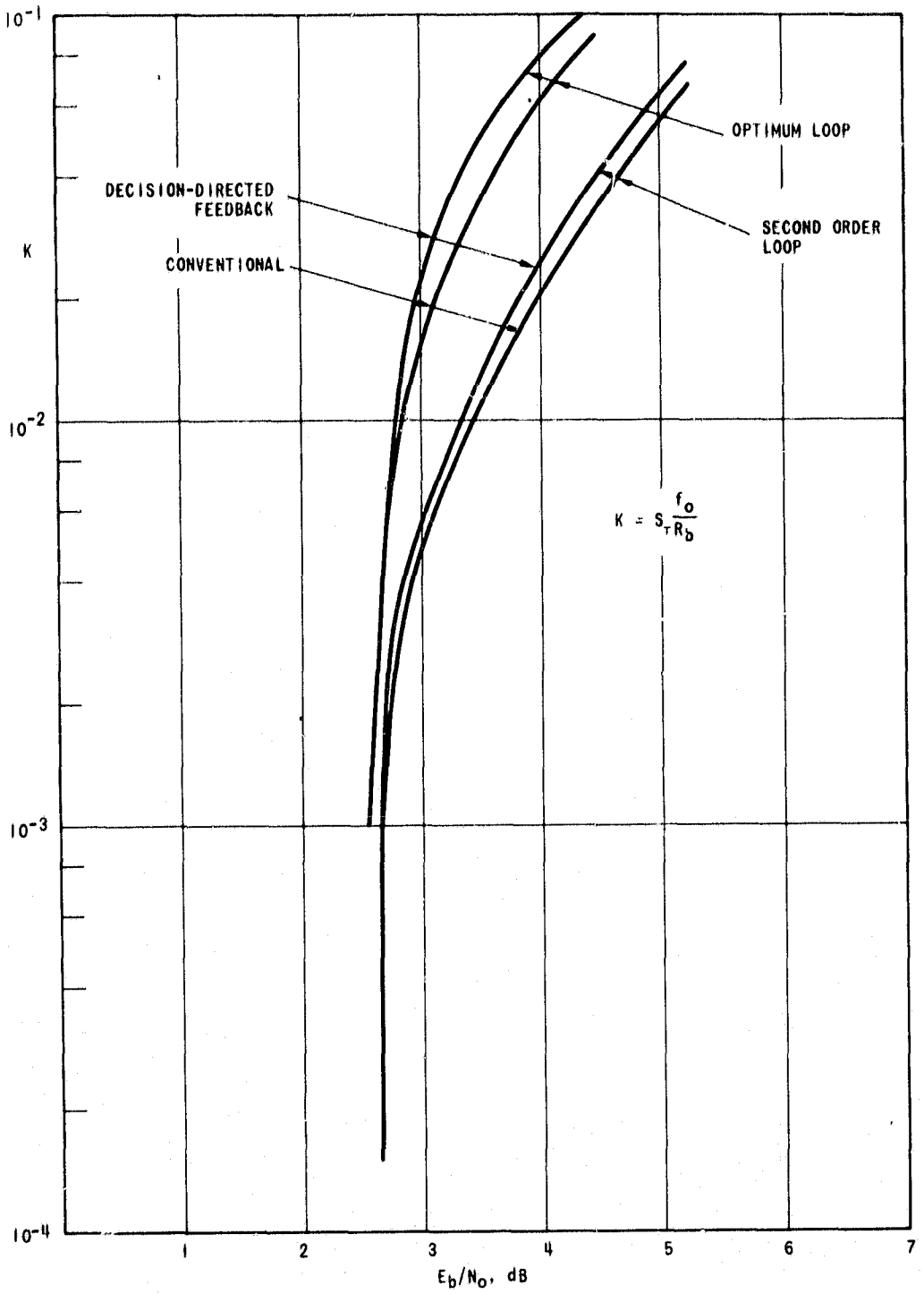


FIGURE 40. Loss of Efficiency vs. K. R_{comp} = Rate = 0.5.

realization is either unattractive because of complexity or not feasible because of second order effects. To name only a few:

a. The realization of loops with a large number of poles.

b. Adaptive features such as utilization of metrics which are functions of signal-to-noise ratio estimates.

c. The elimination of intersymbol interference effects through adaptive compensation.

d. Optimum bit synchronizer and carrier tracking loops based on nonlinear function generators. An example is the optimum carrier tracking loop for PSK which is based on generating a hyperbolic tangent function.

In a receiver where the operations are performed digitally, many of the complex operations associated with (a) through (d) become realizable. More than that, whereas implementations of a solution were often discouraged in the past by the complexities of the proposed approach, the digital approach implies merely a subroutine and the effectiveness of the proposed implementation can be ascertained without additional hardware.

7.1.8 Oscillator Instability and Its Various Causes

The following is a summary of some of the basic concepts associated with oscillator instability. For a detailed discussion see Reference 31. The discussion must necessarily be sketchy since the available literature leaves many questions unanswered, the whole field being in a state of flux. The output of an oscillator may be represented by:

$$v(t) = A \sin \left[\omega_o t + \phi(t)\right]$$

where

ω_o = basic frequency of the oscillator

$\phi(t)$ = phase disturbance of zero mean causing the instability.

The normalized phase disturbance accumulated during a time period τ is given by

$$\langle \dot{\phi} \rangle_{t,\tau} = \frac{\phi(t + \tau/2) - \phi(t - \tau/2)}{\tau} \tag{76}$$

It is a random variable having the variance

$$\sigma^2 \left[\langle \dot{\phi} \rangle_{t,\tau}\right] = \left[E \langle \dot{\phi} \rangle_{t,\tau}^{\ 2}\right]$$

The instability of an oscillator over τ seconds is defined by

$$S_\tau = \frac{\sigma \left[\langle \dot{\phi} \rangle_{t,\tau}\right]}{\omega_o} \tag{77}$$

When the experimenter is faced with the task of computing $\mathcal{L}_\tau$ given by equation (77), he must usually do so in a number of steps dictated by the fact that ω_o is not known exactly:

a. Measure the elapsed phase φ_T during a relatively long time interval T and estimate the oscillator frequency by $\omega_o = \varphi_T/T$.

b. Take a large number of small subintervals τ during T and measuring the elapsed phase φ_τ during each of these, compute

$$\phi(t + \tau/2) - \phi(t - \tau/2) = \varphi_\tau - \omega_o\tau$$

where ω_o has the value obtained in step a.

c. From equation (76) obtain $\langle\dot{\phi}\rangle_{t,\tau}$, through mean-squared averaging over all time intervals τ, compute $\sigma^2[\langle\dot{\phi}\rangle_{t,\tau}]$ and from (77) obtain S_τ.

It may be shown in reference 31 that the above-described averaging over T seconds has the effect of a low-pass filter of bandwidth 2/T on the phase instability $\phi(t)$.

The oscillator instability, S, has three major sources:

a. Additive noise. Caused by the noise of the narrow band amplifies utilized to boost the output of the basic oscillator circuit to the proper power level.

b. Noise that perturbs the oscillation. A random walk in oscillator output phase caused by thermal effect in the oscillator tank circuit.

c. Flicker noise. Caused by frequency drifts, for which $\langle\dot{\phi}\rangle_{t,\tau}$ has a power spectral density given by reference 31:

$$H_{\langle\dot{\phi}\rangle} = \frac{C_1}{|\omega|} \tag{78}$$

According to reference 32, it was verified experimentally that in narrow bandwidth phase-locked loops the contribution of flicker noise predominates. The manner in which oscillator stability measurements are made requires that T be made large enough so that the aforementioned equivalent filter bandwidth 2/T being small, its effect on the first two kinds of oscillator noise becomes negligible. The flicker noise has effectively infinite phase density at zero frequency and as shown in reference 31, the stability S_τ is a function of the ratio T/τ given by

$$S_\tau^2 = \frac{2C_1}{\pi w_o^2}\left(1.04 + 1/2 \log \frac{T}{2\tau}\right) \tag{79}$$

7.1.9 Digitally Implemented Coherent Sequential Decoder Without Decision-Directed Feedback

The performance curves of Figures 39 and 40 show that with the small values of K, corresponding to stable oscillators or high data rates, DDF does not result in increased

signaling efficiency through reduction in phase error. In such a case, the increased memory-size requirements of DDF systems make it unattractive. Digital implementation without DDF does still present substantial advantages. The implementation, with some modifications, is quite similar to that described in Paragraphs 7.1.5 through 7.1.7 and is shown in Figure 41.

The sample vectors Y_{1m}", Y_{2m}" are generated by the A/D converter as previously at a nominal rate of the order of ten/symbol. They are applied to the mixer which is identical to the one described in Paragraph 7.1.7. The frequency and phase estimates are obtained from the phase-locked loop where these estimates are, however, not a function of the hypothesis; rather they are derived in a digital "Costas" circuit. The tracking loop design of such a circuit is based on the recursive filtering approach. The additional feature of a "Costas" loop which is an in-phase correlator and multiplier can be readily implemented digitally. The input to the loop is a vector Y_{1n}', Y_{2n}' which is the same as the one in Figure 35; namely, it is generated at a rate of one vector per symbol. It is obtained as a combination of those vectors Y_{1m}", Y_{2m}" which the symbol synchronizer estimates to belong to the same symbol. The symbol synchronization is obtained in a recursive implementation of which Figure 38(b) is a typical example.

As shown in Table 3, Y_{1n}' is the output of the correlator in which the signal is correlated with the VCO in-phase component, the correlation time being over one symbol. This is precisely the signal used to generate the metric, and the resultant is inserted into the core memory. It is to be noted that the words enter the core memory at the rate of one per symbol; thus the core memory requirement is the same as for analog systems. This reduction from the required DDF memory is achieved because the digital "Costas" and symbol synchronizer loops operate in real time and independently of the sequential decoder.

Recommendations

Based on the potential for improvement described in Paragraph 7.1.8 a systematic program aiming at replacing most of the ground-station receiver by a computer is recommended. A suggested breakdown of the activity into a number of tasks to be performed in sequence follows.

Step 1. Development of a Costas type phase-locked loop computer program, to be run with simulated signals generated inside the computer. These signals will be generated under the assumption of ideal symbol sync and nonencoded PSK modulation.

Step 2. Development of a symbol synchronizer program. The input signals will be simulated, i.e., generated inside the computer, but will be different from Step 1 in that ideal symbol synchronization is not assumed. They will be of the type obtained when the received PSK signal is asynchronously sampled as in Figure 34 at a rate of the order of 10 samples per symbol. The PSK signal is not encoded.

Step 3. Combining Steps 1 and 2, a program is obtained which is capable of demodulating PSK signals and which has as its output the demodulated binary bits. At this stage the input signal to the program is still simulated inside the computer as described in Step 2.

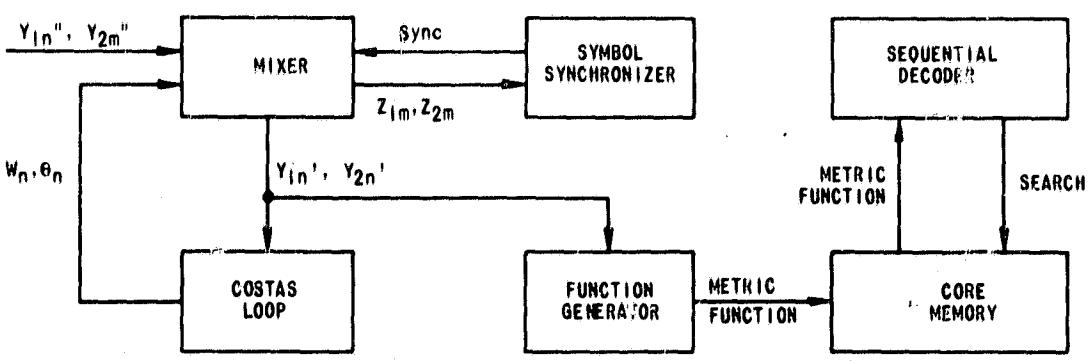


FIGURE 41. Digitally Implemented Coherent Receiver without Decision-Directed Feedback

Step 4. Design and construction of a sampling circuit, such as shown in Figure 34. Generation of a noisy PSK signal either by building a simple test set or from a real signal recorded from an operational system, if available.

Step 5. Feed PSK signal to sampling circuit and feed digital output in real time to computer. Run receiver program and evaluate performance (i.e., count errors). If the noisy PSK signal is from a tape, this might create problems; if it is generated in a test set, the actual information bits may be fed to the computer and the symbol errors can be counted.

Step 6. Combine the phase-locked loop/synchronizer program with an independently developed sequential decoder program. No quantization should be included in the decoding; in this manner the program of Steps 1, 2 and 3 may be used with hardly any modifications. Run program and decode computer-simulated signals.

Step 7. Modify the combined sequential decoder/digitized receiver program for DDF. At this point the signals are still generated internally (simulated in the computer).

Step 8. Build test set which generates noisy PSK convolutionally encoded signals. Feed these to sampling circuit of Step 4.

Step 9. Inject in real time into computer. Run program and decode.

The activity outlined in Steps 1 through 9 can be subjected to considerable modifications. For example the computer simulation of PSK signals might be omitted and real time injection into the computer might be implemented as the first step. Alternately it might be desirable to implement a system without DDF. Such a configuration would be much simpler, while many of the benefits described in Paragraph 7.1.8 would be retained.

7.2 THRESHOLD DECODING

Threshold decoding is a simple algebraic technique for decoding convolutional codes. Its use as a "quick-look" device was suggested in Paragraph 4.3. The purpose of this paragraph is to briefly familiarize the reader with the technique by means of a specific example. Massey's treatment of this subject is complete and illuminating. Much of the following material is derived from material in his book (Reference 33.)

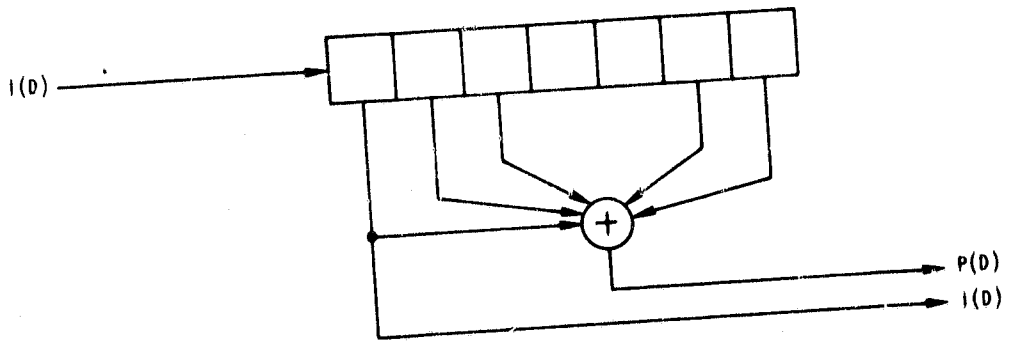
Block diagrams of a convolutional encoder and a compatible threshold decoder are shown in Figure 42. These circuits are designed to generate and decode the (shift register) length seven code which has been referred to as code number four in Tables 1 and 2. Each square box in the diagrams represents a unit (one information bit) delay and each + represents a modulo two adder. Note that the box labelled "Threshold Element" has four inputs and one output. In general, its function is to generate an output which represents the majority vote among its inputs. In this specific example, the threshold element contains four inputs. If zero, one, or two of the inputs are "ones", its output is a "zero". If three or four of its inputs are "ones", its output is a "one". Note that this decoder requires only 13 unit delays*, 8 modulo two adders, and one threshold element. A similar threshold decoder was recently built at C&S using available S.D.S. logic cards. In fact, using only portions of 3 type FT-12 and 4 type IT-11 boards an encoder, decoder, output error sensor, timing clock, and other associated circuitry were configured. The total cost of the seven boards is about $675 and the encoder/decoder thus obtained is capable of operating at a data rate above one megabit per second. It should be noted that similar decoders can be built for other than rate 1/2 codes. Codes with greater constraint lengths can also be threshold decoded using the same basic building blocks: unit delays, modulo two adders, and a threshold element. The total number of blocks required, however is greater for longer codes.

With the addition of a single modulo two adder (assuming use of a rate 1/2 code) to the basic threshold decoder, it becomes possible to obtain a reliable estimate of the symbol error rate at the decoder input. Refer to the decoder shown in Figure 42 (b). The quantity labeled S(D) is called the syndrome. When the syndrome is a binary 1, the input information digit produces a parity digit which does not agree with the received parity digit. (This assumes that the previous six information digits were correctly received.) The output of the threshold element is (with high probability) a 1 when the decoded information bit was received in error. As discussed by Massey, the modulo-two sum of E(D) (a delayed and modified syndrome which accounts for corrected information bit errors) and the threshold element output is (with high probability) a 1 when the parity bit associated with the decoded information bit was received in error. Thus, the simple additional circuitry shown below, serves to count the total number of received symbol errors.

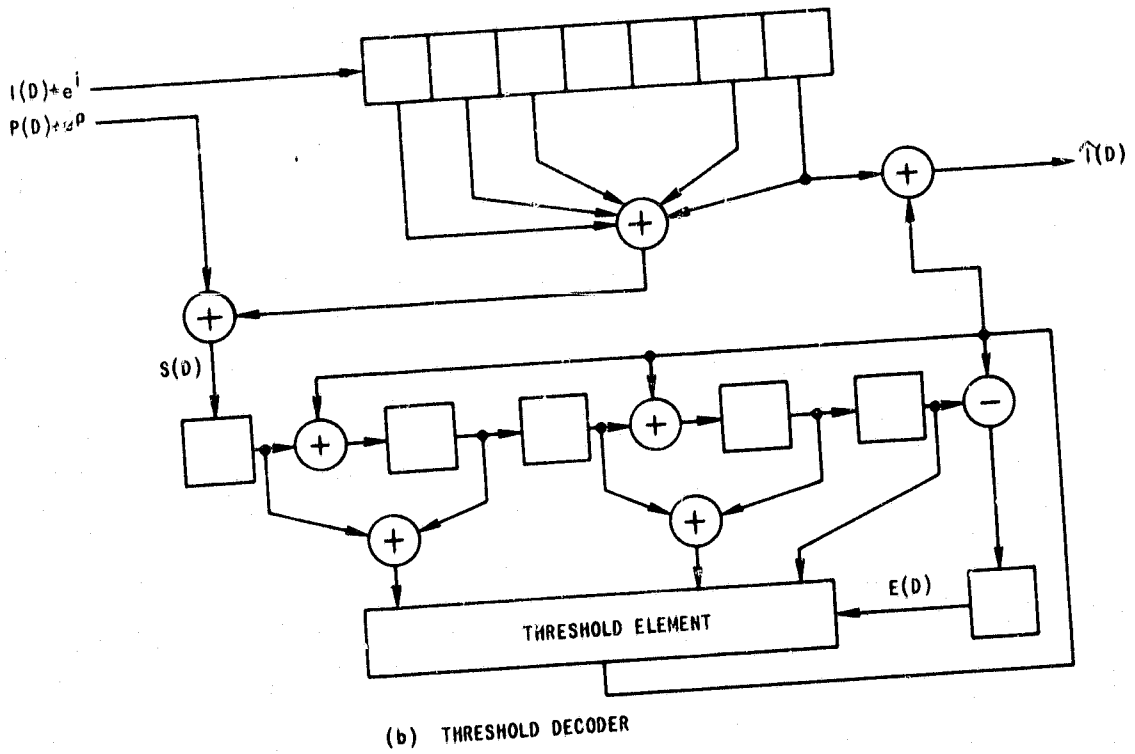
In the preceding discussion, "with high probability" can be replaced by "with probability 1" when only a single isolated symbol error is received. Such is the case when adjacent symbol errors are spaced by at least twice the length of the encoder shift register. For example, when considering a length 48, rate 1/2, systematic convolutional

---

*Actually only 12 unit delays since the shift register needs only 6 stages and not 7 as illustrated.

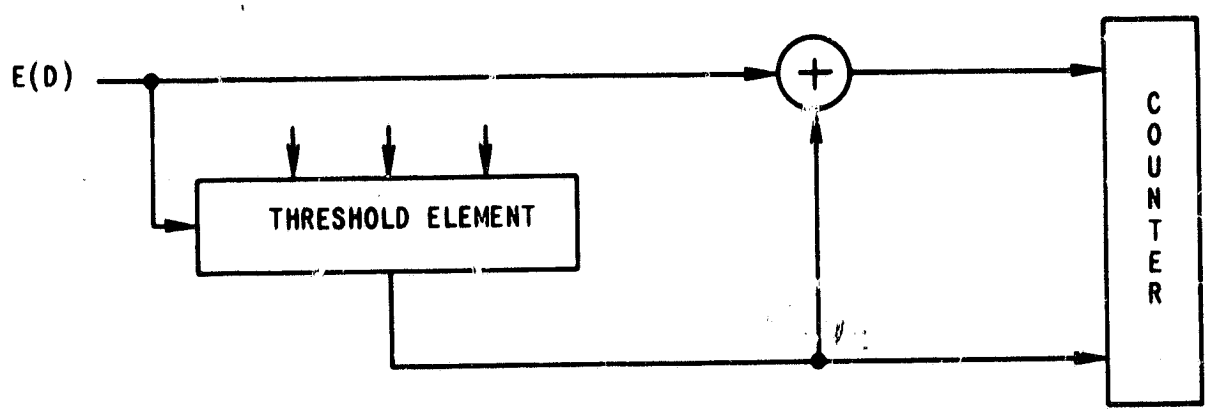


(a) CONVENTIONAL ENCODER



(b) THRESHOLD DECODER

FIGURE 42. Convolutional Encoder - Threshold Decoder



code, symbol error rates of up to about one precent can be accurately measured in the manner described above.

The probability of decoded bit error can be calculated as a function of received symbol error probability. Using the simple code illustrated in Table 1 (Code #4)

$$G(D) = 1 + D + D^2 + D^5 + D^6. \tag{80}$$

Following Massy's procedure for this code, one finds

$$P_1(e) = (1-p_o)\, C_T + p_o\, C_+ \tag{81}$$

where

p_o = probability of symbol error on a binary symmetric channel = $p_r\{e_o^i = 1\} = p_s$

$$C_T = p_1p_2p_3p_4 + q_1p_2p_3p_4 + p_1q_2p_3p_4 + p_1p_2q_3p_4 + p_1p_2p_3q_3$$

$$C_+ = C_T + q_1q_2p_3p_4 + q_1p_2p_4q_3 + p_1q_2p_4q_3 + q_1p_2p_3q_4 + p_1q_2p_3q_4 + p_1p_2q_3q_4$$

and

$$p_1 = p_o$$

$$p_2 = 2p_o(1-p_o)$$

$$p_3 = 3p_o - 6p_o^2 + 4p_o^3$$

$$p_4 = 4p_o - 12p_o^2 + 16p_o^3 - 8p_o^4$$

The quantity $p_1(e)$ is the probability of a decoded bit error when six previous bits are correctly decoded. For low error probabilities, this is essentially the decoded bit error probability. For symbol error probabilities below 1 percent, the results can be simplified to

$$p_1(e) \approx 85\, p_s^3 \approx p_b. \tag{82}$$

The implication of this asymptotic performance is that the decoder corrects all single and double symbol errors occuring within the effective constraint length* of 11 bits and roughly half of the triple error patterns. Numerical results of this analysis are shown in Figure 43. Since primary interest lies in longer codes to obtain negligible error rates when sequentially decoded, the preceding analysis was repeated for a length 32, rate 1/2, systematic convolutional code (previously referred to as Code #8). The calculated threshold decoded performance of this code is illustrated in Figure 44. For this code, it has been found that J=4 and n_e=18. It corrects all single or double errors within 18 symbols and a sizeable fraction of the triple error patterns.

Although Massey's work was devoted only to the threshold decoding of systematic convolutional codes, it is entirely reasonable to expect that comparable levels of performance can be achieved with non-systematic codes. The implementation of threshold decoders for use with non-systematic codes deserves future consideration since the technique of threshold decoding offers advantages as a quick-look scheme and non-systematic codes offer advantages over systematic codes when sequentially decoded.

7.3 CONCATENATED CONVOLUTIONAL/ALGEBRAIC CODES

The concept of concatenation, or linking together, of codes was first set forth by Forney (Reference 34.) He uses this approach to construct long codes out of shorter ones, but reduces the required computation over that associated with a single long code. He further demonstrates that concatenation of a finite number of codes yields an error exponent inferior to that of one single code but still remaining non zero for all rates below capacity. While Forney dealt exclusively with block codes, the concept of concatenation could apply equally well to convolutional codes.

Falconer (Reference 35) investigated in detail the concatenation of algebraic codes with convolutional encoding/sequential decoding. He analyzed and simulated a hybrid algebraic/sequential scheme whereby information was transmitted over N parallel

---

*The effective constraint length, n_e, and the number of orthogonal parity check equations, J, are bounded by

$$n_e \geq 1/2\, J^2 + 1/2\, J + 1; \text{ for rate } 1/2 \text{ codes.}$$

For the code discussed here, J has been found to equal 4 and n_e = 11. Since the above equality is satisfied, this code is relatively efficient.

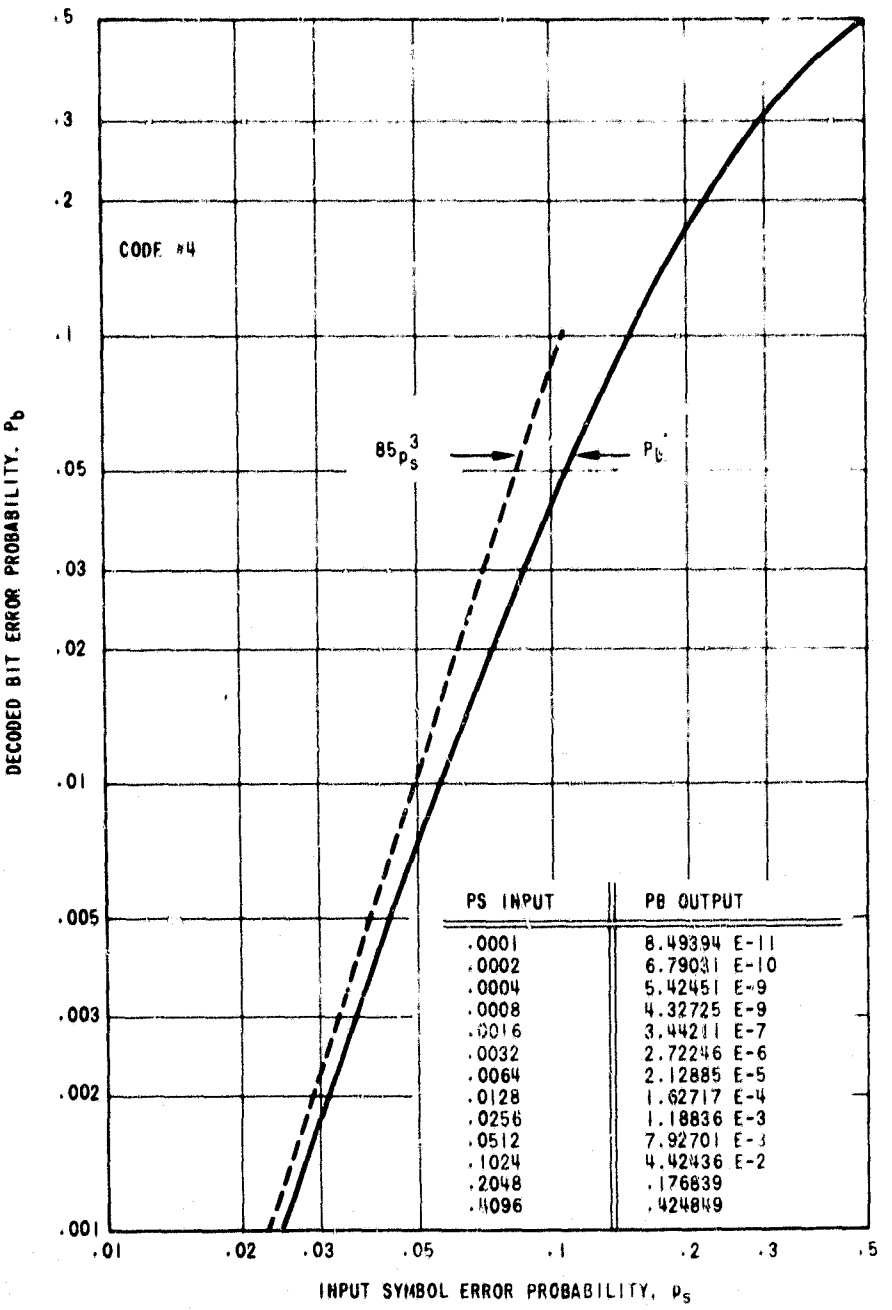


FIGURE 43. Threshold Decoder Performance For Code 4

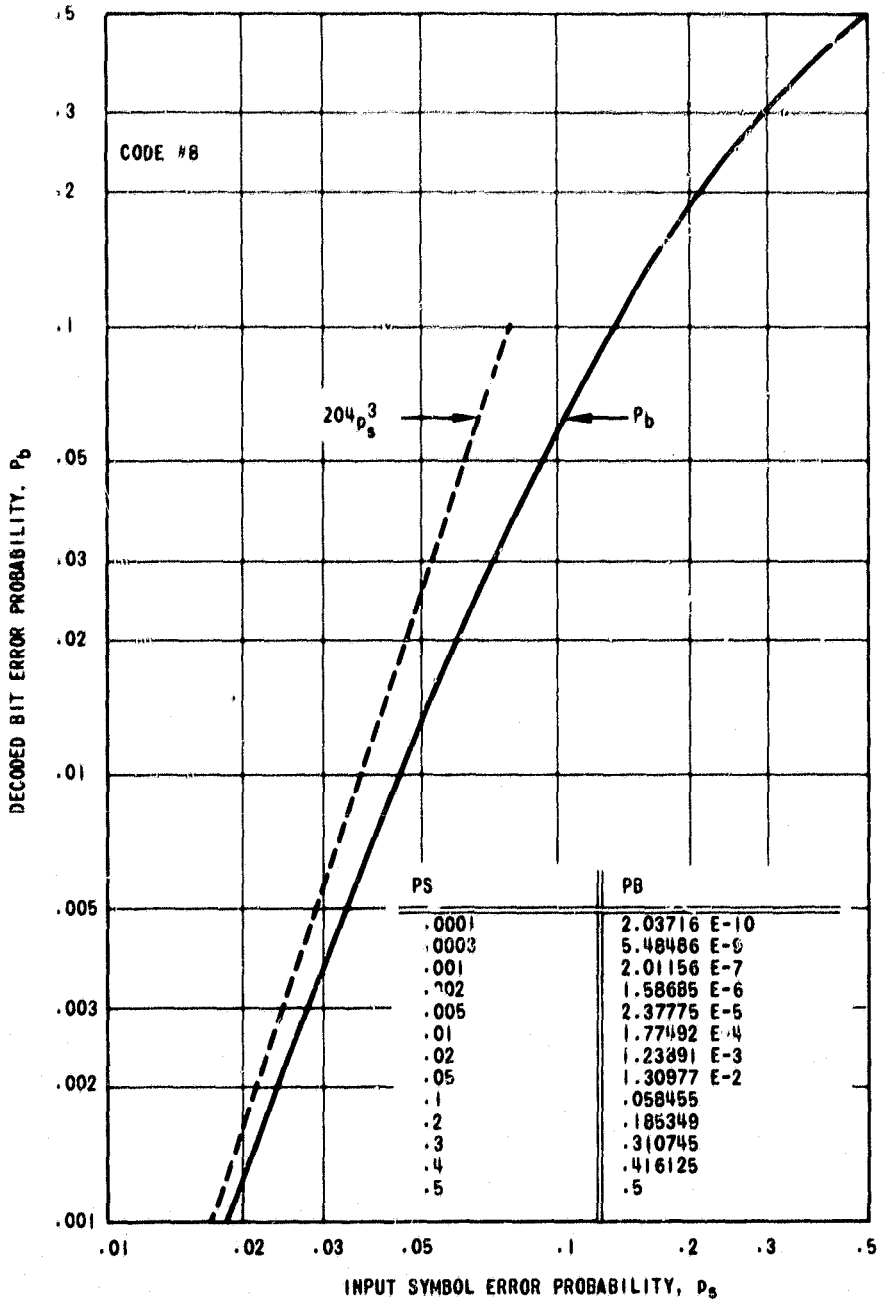


FIGURE 44. Threshold Decoder Performance For Code 8

convolutionally encoded/sequentially decoded systems. Each of these streams was not independent, however; an algebraic encoder formed code words of length N and rate R from a subset of the original information stream. This concept is outlined in Figure 45. In his dissertation, Falconer considered both Reed Solomon and simple partiy check codes for the algebraic portion of the concatenation scheme. Falconer's method generally works as follows: Each of the N sequential decoders proceeds independently. When a particular decoder (or decoders) encounters difficulty, the symbols in question are treated as erasures and are corrected by the algebraic code. If the algebraic code has minimum distance d, then up to d - 1 erased symbols may be corrected per block.

Analytical and simulation results presented by Falconer of greatest interest are that the hybrid scheme permits decoding at rates between R_{comp} and capacity with a finite mean number of computations per decoded bit, and the effective Pareto exponent for the hybrid scheme is the product of the Pareto exponent for a single unaided sequential decoder and the minimum distance of the algebraic code. For example, his simulation results for a convolutional code of rate 1/7 concatenated with a simple parity check code of rate 9/10 operating at E_b/N_O of 1.4 dB (3 dB from capacity for the infinite bandwidth channel) showed a sample mean computation per bit of approximately 50 and a Pareto exponent of 1.2. (A binary input, eight-ary output channel was simulated.) A rate 1/7 code for this channel is operating in excess of R_{comp} at E_b/N_O = 1.4 dB.

Potential Advantages

As briefly outlined, the work to date regarding concatenation of codes has been largely theoretical in nature and oriented toward a fundamental understanding of the properties and performance of these codes. The promising results favor further investigation of the potential of this approach for space telemetry problems. Consider, briefly, the following problem areas applicable to telemetry using convolutional encoding/sequential decoding.

a. Capacity Limitations. For deep space missions in particular, it is important to utilize all available power to the extent practicable. It is well known that such a channel may be represented as an additive white gaussian noise channel for which $(E_b/N_O)_{min}$ = -1.6 dB. From Figure 19 it may be seen that for a single, convolutional encoder/sequential decoder the minimum theoretical E_b/N_O for a rate 1/3 code and eight-ary output channel is about 2.2 dB. When practical allowances are made for noisy phase references, resynchronization and other implementation difficulties, another 12 dB loss may be incurred. The work of Falconer has shown an approach which is capable of considerably enhanced performance. Thus, the first area in which the concatenation scheme might aid is that of increased capacity (or longer range in the case of a deep space probe).

b. Decoder Overflow. A well recognized problem in conjunction with sequential decoding is that of decoder overflow. Concatenation could be used in a number of possible ways to reduce the effect of overflow. One such way is that described above whereby the Pareto exponent is increased and the overflow probability therefore reduced.

An alternate approach would be to permit overflow to occur with some probability and then treat blocks for which overflow occurred as erasures and employ burst correction codes to correct these erasures. These could either be block or convolutional

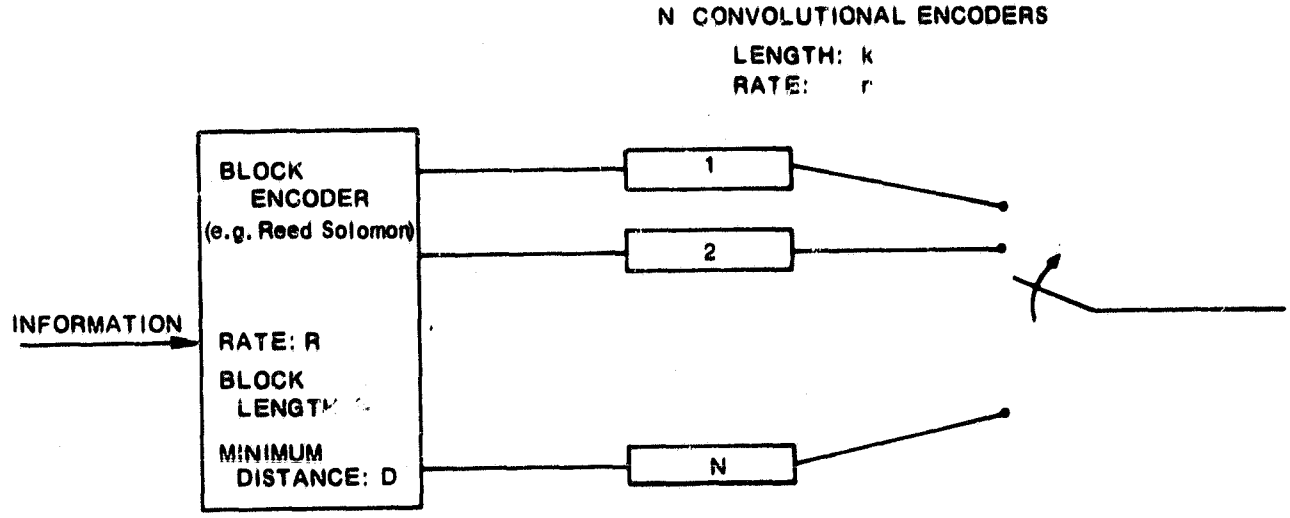


FIGURE 45. Hybrid Algebraic/Convolutional Encoder

codes. For example, suppose decoding proceeds by blocks of length L. Assume, for the present, that at least KL error free blocks follow an erasure of a single block. Berlekamp (Reference 36) has proven for a model such as this that there exists a unique convolutional code which is capable of correcting a burst of length E followed by a quiet interval of length Q such that

$$\frac{Q}{Q+E} \geq R \tag{83}$$

For the case described, then

$$\frac{K}{K+1} \geq R \tag{84}$$

But for rates of the form R = n/n+1, n an integer, Berlekamp has shown an algorithm for finding a code which satisfies Equation (83) with equality.

Assuming that the overflow probability is p, then for p small, the probability of more than one overflow in k blocks is approximately $(Kp)^2$. For example, if K = 10 $p = 10^{-5}$, the probability of 2 or more overflows in K blocks is 10^{-8}. Hence, for most overflows, the burst code could completely reconstruct the entire block (and thereby also reduce to negligible proportions the effect of clustered errors within the block). For the small events in which two overflows occur in less than the design quiet interval, the incoming data could be presented to the user and the total probability of error or erasure made extremely small. Convolutional burst codes of the type considered above have been described by Massey (Reference 37.)

The approach suggested is included only for illustrative purposes to demonstrate some of the extensive possibilities of concatenation applied to a problem other than increasing capacity. It is not suggested that this represents a particularly good approach from an implementation standpoint. This would require further investigation.

c. Implementation Simplification. A third area deserving attention is the question of implementation. A motivation behind the whole concept of concatenation is that of less difficult hardware implenetation. This notion should be explored to see if encoder reliability could be enhanced by some concatenated approach and b if decoder complexity could be reduced. For example, it might be reasonable to consider shortened constraint length convolutional encoders in conjunction with sequential and algebraic decoding. While the undetected error rate of the sequential decoder without algebraic decoding would be high, the outer (block) coder could reduce the error probability. Allowing more undetected errors would significantly reduce computational requirements.

Recommendations

The concept of concatenation of convolutional encoding/sequential decoding with algebraic (or pherhaps even other convolutional) codes should be investigated for space telemetry. The following specific objectives should be considered:

a. Possible hardware simplification and/or increase in reliability

b. Improvement in performance with regard to overflows and undetected errors

c. Decoding for deep space channels at rates in excess of the computational cutoff rate.

The major goal would be to consider the practical implementation of these theoretical concepts.

7.4 SIMULATION PROGRAM FOR SEQUENTIAL DECODERS

A program to simulate a sequential decoder on the IBM 1130 computer (8192 word memory) was written for this contract. The program was based upon previous work by C&S on other computers. Because of the restricted time available, the program was written in FORTRAN IV with parity check calculations performed in an assembler language subroutine.

The objective of this computer program was to provide a flexible tool to supplement analytical effort in the area of convolutional encoding/sequential decoding for space missions. No attempt was made to optimize the program for maximum computer efficiency. Such an optimization would have been too time consuming. Optimization would be required before extensive simulation work could be undertaken and would probably require conversion of the entire program to IBM assembler language with the possible addition of external hardware.

Several types of channels were simulated, including the binary symmetric channel, binary input/eight-ary output channel, and a discrete memoryless approximation to the short burst channel (described in detail in Paragraph 4.4). For the purposes of this report, only the binary symmetric channel program is described; the others merely involve appropriate changes in the generation of input data and the decoding metric.

Overall Description of the Program

Primarily for purposes of simplicity in programming, an all zero information sequence is transmitted; hence all transmitted symbols (e.g., information symbols and parity symbols for the case of systematic codes) are zero. Transmitted symbols are received in error with probability p. This is achieved by using a random number generator to simulate errors. For channels of greater complexity, the same general approach is used. That is, the random number generator is used in conjunction with the specified channel transition probabilities to characterize the received symbol.

An input sequence of length N information bits is thus generated. This sequence is followed by a tail of zeroes of length L. Both N and L may be varied and, in general, $L \ll N$.

This input sequence is then sequentially decoded using the Fano algorithm. Since this algorithm is assumed known to the reader, it will not be discussed here. The programming is relatively straightforward; a listing is included in Appendix I.

For this contract, systematic and nonsystematic codes of rate 1/2 were simulated. Encoder constraint length was set at 32 for programming simplicity. (The IBM 1130 is a 16-bit word machine.) Extension to codes of greater redundancy and length could be accomplished by relatively simple extension.

While the programs are somewhat flexible, the following inputs are generally used:

a. Channel transition probabilities

b. Block length

c. Number of blocks to be decoded

d. Maximum allowable computations per block

e. Length of block "tail"

f. Metric constant

g. Threshold separation

h. Code tap configuration.

The following outputs were specified:

a. Input errors per block

b. Output undetected errors

c. Number of computations per block

d. Number of computations per information digit for each block

e. Average number of computations for all blocks

The program simulated the channel and decodes at a rate of approximately 250,000 information bits per hour on the IBM 1130. (This particular installation has a 3.2 microsecond core memory cycle). Two of the major limitations to speed are the use of the IBM furnished random number generator which is written in FORTRAN and is rather slow and computation of parity checks. Each of these areas would require considerable work to improve decoding speed.

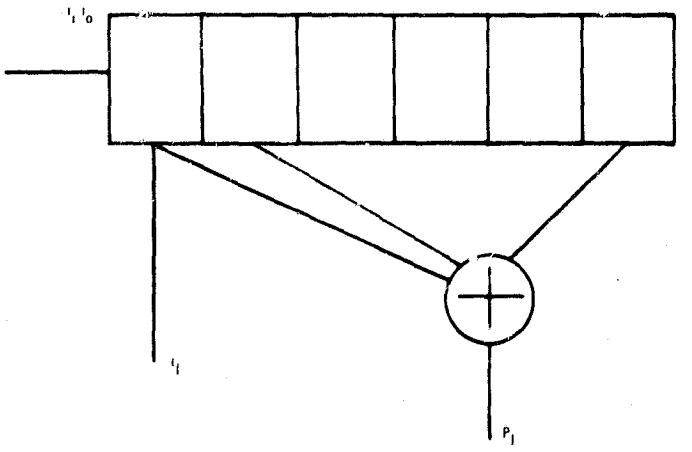
Digital Computer Calculation of Parity Calculations for Convolutional Codes

The calculation of parity checks represents one of the most time consuming operations for a digital computer. This arises because of the large number of operations required, even when assembler language instructions are used. The contents of the encoder must be added (exclusive or, EOR) to the tap configuration and the result checked for an odd or even number of ones. This latter check requires rotating the result of the EOR operation in 32 discrete steps. While some programming shortcuts were used, the problems is compounded by the short word length of the IBM 1130. Two solutions which might be considered are calculation of parity digits in external hardware and use of software methods which do not require the extensive number of operations as that previously described. Because of the short duration of this contract, neither of these methods were implemented for this contract. However, an approach to more efficient programming, known as the impluse method, is suggested.

Impulse Method

The impulse method is a simplified method of computing parity checks for convolutional codes. This approach is particularly well suited for digital computer applications, and may also be useful for some types of analytical work.

A systematic, rate 1/2 convolutional encoder of length k = n + 1 is illustrated below:



Using the notation introduced by Reference 33, the input information sequence may be represented by the polynomial

$$I(D) = i_0 + i_1 D + i_2 D^2 + \ldots . \tag{85}$$

where i_k is the information symbol entering the encoder at time k and D is the unit delay operator. For the rate 1/2 systematic code illustrated, the transmitted parity sequence, P(D), may be expressed as

$$P(D) = G(D)\ I(D) \tag{86}$$

where G(D) is the code generating polynomial (i.e., $G(D) = g_0 + g_1 D + g_2 D^2 + \ldots\ G_n D^n$).

The unit impulse information sequence can be defined as a binary "one" followed by an infinite sequence of zeroes. In terms of the above notation, then,

$$I(D) = 1$$

Thus,

$$P(D) = G(D)$$

and the transmitted parity sequence is identical to the tap configuration (i.e., the code generating polynomial). From this observation and the analogy to linear circuits, the name "impulse method"* follows. The parity sequence thus generated is known as the impulse sequence.

The general transmitted parity sequence may be generated from the modulo 2 addition of the shifted impulse sequences. From Equation 86, P(D) can be derived as follows:

$$\begin{aligned} P(D) &= G(D)\, I(D) \\ &= i_0 g_0 + D(i_0 g_1 + i_1 g_0) \\ &\quad + D^2 (i_0 g_2 + i_1 g_1 + i_2 g_0) \\ &\quad + D^n (i_0 g_n + i_1 g_{n-1} + \ldots + i_n g_0) \\ &\quad + D^{n+1} (\quad i_1 g_n + \ldots + i_{n+1} g_0). \end{aligned} \tag{87}$$

The parity digit at time j, for example, is

$$p_j = \sum_{k=j-n} i_k\, g_{j-k}$$

where all operations are performed modulo 2. Hence p_j, the parity check at time j, is the shifted sum of a subset of the coefficients of the generator polynomials. The particular subset is determined by the associated value of the information digit, i_k.

The parity checks may be accomplished in a digital computer within a single register of length equal to the convolutional encoder length. The required operations are noted below (assuming some working register, R, is initialized to zero). Let G be defined as a computer word with binary "ones" in those locations in which the encoder taps are corrected (i.e., g_k=1).

Case I: Hypothesized digit is a "zero"

a. Test left most digit of R for zero or one. (this is the parity digit.)

b. Shift left one.

*This concept of an impulse approach to parity calculations was originally suggested by Dr. Irwin Jacobs in a private communication.

Case II: Hypothesized digit is a "one"

a. R = R+G, where the addition is a modulo 2.

b. Performs steps (1) and (2) of Case I.

It should be apparent that this method is significantly faster for digital computer applications than the more conventional methods which require counting ones to determine even or odd parity. This time saving is important for large scale simulations.

For example, consider the six-stage shift register shown above and assume the following information sequence: 10011001

i.e., $I(D) = 1 + D^3 + D^4 + D^7$

$G(D) = 1 + D + D^5$

then the following table may be constructed:

Hypothesized Digit	Initial State of R	State of R at Check for Parity	Final State of R
1	000000	110001	100010
01	100010	100010	000100
001	000100	000100	001000
1001	001000	111001	110010
11001	110010	000011	000110

It can be easily verified that the left digit in column three is the correct parity digit. Obviously, right shifts and testing of the right hand digit could be used equally well.

Section 8

NEW TECHNOLOGY

In accordance with the New Technology clause in the contract, Communications & Systems, Incorporated has undertaken the following activities:

a. The C&S Project Manager met periodically with the C&S Contracts Administrator to discuss possible new technology which was developed under Contract NAS5-11503.

b. The C&S Project Manager was encouraged at all times to identify any new technology by review of engineering notes, notebooks, documents and the various technical reports developed under Contract NAS5-11503.

c. At the completion of engineering activity under Contract NAS5-11503, a final new technology meeting was held with all personnel concerned with the project for possible identification of any new technology.

None of these activities have resulted in the identification of any new technology under Contract NAS5-11503.

BIBLIOGRAPHY

1. J. M. Wozencraft and B. Reiffen, Sequential Decoding, MIT Press and Wiley, New York, 1961.

2. R. M. Fano, "A heuristic discussion of probabilistic decoding," IEEE Trans. on Information Theory, Vol. IT-9, pp. 64-74, April 1963.

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

4. C&S Inc., Staff, "Convolutional Encoding second sequential decoding," C&S memorandum report, June 1967.

5. J. F. Savage, "Sequential Decoding on a Channel with Simple Memory," Lincoln Laboratory Group Report 1964-5, 2 April 64.

6. R. G. Gallager, "A Simple Derivation of the Coding Theorem and Some Applications," IEEE Transactions on Information Theory, Volume IT-11, No. 1, January 1965, pp. 3-18.

7. C. E. Shannon, R. G. Gallager, and E. R. Berlekamp, "Lower Bounds to Error Probability of Coding on Discrete Memoryless Channels," Information and Control, 1967.

8. I. M. Jacobs and E. R. Berlekamp, "A Lower Bound to the Distribution of Computation for Sequential Decoding," IEEE Transactions on Information Theory, Volume IT-13, No. 2, April 1967, pp. 167-174.

9. I. M. Jacobs, "Sequential Decoding for Efficient Communication from Deep Space," IEEE Transactions on Communication Technology, Volume COM-15, pp. 492-501, August 1967.

10. A. J. Viterbi, "Phase-Locked Loop Dynamics in the Presence of Noise by Fokker-Planck Techniques," Proc. IEEE, Volume 51, pp. 1737-1753, December 1963.

11. P. A. Tenkhoff, "Performance of Binary Communication with a Noisy Phase Reference Using Sequential Decoding," Proc. IEEE, Volume 55, pp. 1643-1645. September 1967.

12. J. J. Bussgang, "Some properties of binary convolutional code generators," IEEE trans. on Information Theory, Volume IT-11, pp. 90-100, January 1965.

13. H. L. VanTrees, "Optimum Power Division in Coherent Communication Systems," MIT Lincoln Lab., Lexington, Mass., Tech. Report 30, February 1953.

14. D. Halford, "A General Mechanical Model for $|f|^{\alpha}$ Spectral Density Random Noise with Special Reference to Flicker Noise $1/|f|$," Proc. IEEE, Volume 56, pp. 251-258, March 1968.

15. K. L. Jordan, "The Performance of Sequential Decoding in Conjunction with Efficient Modulation," IEEE Transactions on Communication Technology, Volume COM-14, Number 3, June 1966, p. 291.

16. Ibid, p. 286.

17. B. Reiffen, "Sequential Decoding for Discrete Input Memoryless Channels," IRE Transactions on Information Theory, Volume IT-8, April 1962.

18. Jordan, op. cit., p. 290.

19. J. Stiffler, "A Comparison of Several Methods of Subcarrier Trackling," Jet Propulsion Laboratory, Space Programs Summary No. 37-37, Volume IV, February 28, 1966, pp. 268-275.

20. G. Frenkel, "The Optimum Carrier-Tracking Circuit in Binary Coherent Communications Systems," C&S Internal Memorandum, 17 July 1967.

21. D. Haccoun, "Simulated Communication with Sequential Decoding and Phase Estimation," SM Thesis, MIT, September 1966.

22. P. M. Gray, "Simulation of Sequential Decoding with Decision-Directed Channel Measurement," SM Thesis, MIT, June 1966.

23. G. Frenkel, "Carrier-Tracking Loops in Binary Coherent Communications Systems," IEE Conference Publication No. 31. Conference on Frequency Generation and Control of Radio Systems 22-24, May 1967, London.

24. W. E. Larimore, "Synthesis of Digital Phase-Locked Loops," C&S Internal Memorandum, 12 January 1968.

25. D. P. Lindorff, "Theory of Sampled-Data Control Systems," John Wiley & Sons, 1965.

26. R. E. Kalman, "New Methods in Wiener Filtering," Proc. First Synposium on Engineering Applications of Random Function Theory, Purdue University, November 1960 (New York: Wiley, 1963) pp. 270-388.

27. T. W. Anderson, "Introduction to Multivariate Statistical Analysis," Section 2.5.2, John Wiley & Sons, Inc., 1958.

28. Ibid, Section 2.5.3

29. H. L. VanTrees, "Detection and Continuous Estimation: The Fundamental Role of the Optimum Realizable Linear Filter," Wescon/66 Technical Papers, Session 7, August 13-26, 1966.

30. D. Snyder, "A Theory of Continuous Nonlinear Recursive Filtering with Applications of Optimum Analog Demodulation," Wescon/66 Technical Papers, Session 7, August 13-26, 1966.

31. L.S. Cutler and C.L. Seale, "Some Aspects of the Theory and Measurement of Frequency Fluctuations in Frequency Standards," Proc. IEEE, Volume 54, pp. 136-154, February 1966.

32. R. Sydnor and J.J. Caldwell, "Frequency Stability Requirements for Space Communications and Tracking Systems," Proc. IEEE, Volume 54, pp. 231-236, February 1966.

33. J.L. Massey, "Threshold Decoding," MIT Press, 1963

34. G.D. Forney, Concatenated Codes, Technical Report 440, MIT Research Laboratory of Electronics, Cambridge, Massachusetts, December 1, 1965. (Also submitted to MIT as a Ph.D. Dissertation, March 31, 1965.)

35. D.D. Falconer, A Hybrid Sequential and Algebraic Decoding Scheme, Ph.D. Dissertation, MIT, February, 1967.

36. E.R. Berlekamp, "A Class of Convolution Codes," Information and Control, Volume 6, pp. 1-13, 1963.

37. J.L. Massey, "Implementation of Burst-Correcting Convolutional Codes," IEEE Trans. on Information Theory, Volume IT-11, No. 3, July, 1965.

Appendix I

CHANNEL CAPACITY CALCULATIONS

This appendix outlines the calculations for the capacity of a channel as illustrated in Figure 5. The general procedure is shown for a model of burst length three with extensions to the general channel.

Consider the transition matrix P where the elements p_{ij} represent the probability that the jth output letter was received given that the ith input letter was transmitted.

$$P = \begin{bmatrix} q & o & o & o & p' & p' & p' & p' \\ o & q & o & o & p' & p' & p' & p' \\ o & o & q & o & p' & p' & p' & p' \\ o & o & o & q & p' & p' & p' & p' \\ p' & p' & p' & p' & q & o & o & o \\ p' & p' & p' & p' & o & q & o & o \\ p' & p' & p' & p' & o & o & q & o \\ p' & p' & p' & p' & o & o & o & q \end{bmatrix} \qquad \text{(I-1)}$$

The mutual information of A and B may be expressed as

$$I(A;B) = H(B) - H(B|A) \qquad \text{(I-2)}$$

where

$$H(B|A) = \sum_{A,B} P(a,b) \log \frac{1}{P(b|a)}$$

$$H(B|A) = \sum_{A,} P(a) \sum_{B} P(b|a) \log \frac{1}{P(b|a)}$$

$$\sum_{B} P(b|a_i) \log \frac{1}{P(b|a_i)} = q \log \frac{1}{q} + 4p \log \frac{1}{p}$$

$$H(B|A) = \sum_{A} P(a_i) \left[q \log \frac{1}{q} + 4p' \log \frac{1}{p'} \right] = q \log \frac{1}{q} + 4p' \log \frac{1}{p'}$$

$$P(b_1) = P(a_1)\,q + p'\left[P(a_5) + P(a_6) + P(a_7) + P(a_8)\right]$$

$$P(b_2) = P(a_2)\,q + p'\left[P(a_5) + P(a_6) + P(a_7) + P(a_8)\right]\text{; etc. for } P(b_3) \text{ and } P(b_4)$$

$$P(b_5) = P(a_5)\,q + p'\left[P(a_1) + P(a_2) + P(a_3) + P(a_4)\right]$$

$$P(b_6) = P(a_6)\,q + p'\left[P(a_1) + P(a_3) + P(a_2) + P(a_4)\right]\text{; etc. for } P(b_7) \text{ and } P(b_8).$$

Note that H(B|A) does not depend upon the $P(a_i)$ distribution. Thus, I(A;B) will be maximum when H(B) is maximum w.r.t. the $P(a_i)$ distribution. We know that H(B) will be maximum when

$$P(b_i) = 1/8 \text{ for all } i = 1, 2, \ldots, 8.$$

Note that $q + 4p' = 1$; $p' = (1-q)/4$. Thus,

$$P(b_1) = qP(a_1) + \left(\frac{1-q}{4}\right)\left[P(a_5) + P(a_6) + P(a_7) + P(a_8)\right]\text{; etc.}$$

$$P(b_5) = qP(a_5) + \left(\frac{1-q}{4}\right)\left[P(a_1) + P(a_2) + P(a_3) + P(a_4)\right]\text{; etc.}$$

Suppose we let $P(a_i) = 1/8$ for all $i = 1, 2, \ldots, 8$.

Then, $P(b_1) = \frac{1}{8}q + \frac{1}{8}(1-q) = \frac{1}{8} = P(b_2) = \ldots = P(b_8)$.

Then H(B) will maximize when all input symbols are equipable. Max $H(B) = \log_2 8 = 3$ bits/input symbol = 1 bit/binary digit.

$$C = \max I(A;B) = 3 + q \log q + 4p' \log p' = 3 + q \log q + (1-q) \log \left(\frac{1-q}{4}\right) P(a_i)$$

$$= 3 + q \log q + (1-q) \log (1-q) - 2(1-q)$$

$$\boxed{\therefore C = 1 + 2q - H(q)} \qquad \text{(I-3)}$$

For $q = 1$; (no errors); $C = 3$ bits/input symbol. In general; $C = \frac{1}{3}\left[1 + 2q - H(q)\right]$ bits/binary digit. Notice that for $q = 1$, $C = 1/3$ bit/binary digit and thus, the channel can never become totally random. This might represent a weakness in the model.

In more generality, consider bursts of length b or less. The case treated above was for $b = 3$. Following through the above analysis, it becomes clear that

$$C = b + q \log q + (1-q)\log(1-q) - (1-q)\log 2^{b-1}$$

$$C = b - (1-q)(b-1) - H(q)$$

$$\boxed{C = 1+q(b-1) - H(q)} \quad \text{bits|symbol} * \tag{I-4}$$

$$\boxed{C = \frac{1}{b}\left[1 + q(b-1) - H(q)\right] \quad \text{bits|binary digit}} \tag{I-5}$$

Finally, the average probability of error, P, may be related to the transition probability, p.

$$p' = (1-q)/2^{b-1}$$

$$\text{prob.}\left\{\text{binary error|burst error}\right\} = \frac{b+1}{2b}$$

Average binary error probability on a BSC is

$$p = \frac{1-q}{2}\left(\frac{b+1}{b}\right) = \frac{2^{b-1}p'}{2}\left(\frac{b+1}{b}\right)$$

$$\boxed{p = 2^{b-2}p'\left(\frac{b+1}{b}\right)} \tag{I-6}$$

---

*Note: For any value of b, $c \simeq 1$ bit|symbol for $q \simeq o$. This result is correct since the observation of an output symbol specifies which half of the input alphabet was used. Also, note that for $b = 1$, $C = 1 - H(q)$ which is the correct result for a BSC.

Appendix II

ERROR BOUND FOR MARKOV CHANNELS

The procedure followed by Gallager (Reference 1) for memoryless channels is paralleled here for channels in which errors occur as a first order Markov process. As shown by Gallager, the average probability of error can be upper bounded by a function containing the sum

$$\sum_{y \epsilon Y_N} \left[\sum_{x \epsilon x_N} \Pr(y \mid x)^{1/2} \right]^2 \qquad \text{(II-1)}$$

$$\vec{x} = (x_1 \ldots x_N)$$

$$\vec{y} = (y_1 \ldots y_N)$$

Define,

$$P\binom{x_1}{y_1} = P_r\,(X_1 \rightarrow y_1) \qquad \text{(II-2)}$$

Let

$$P\binom{0}{0} = P\binom{1}{1} = P_L \qquad \text{(L = "like")}$$

$$P\binom{0}{1} = P\binom{1}{0} = P_U \qquad \text{(U = "unlike")}$$

Write,

$$Z = Z_1 \ldots . Z_N$$

where

$$Z_i = U \text{ if } \binom{x_i}{y_i} = \binom{0}{1} \text{ or } \binom{1}{0}$$

$$Z_i = L \text{ if } \binom{x_i}{yi} = \binom{0}{0} \text{ or } \binom{1}{1}$$

The generation of the sequences is governed by the conditions for a simple Markov Chain:

$$P_r(L \mid L) = q_1 \quad , \quad P_r(U \mid L) = p_1$$

$$P_r(L \mid U) = p_2 \quad , \quad P_r(U \mid U) = q_2$$

$$p_1 + q_1 = 1 \quad , \quad p_2 + q_2 = 1.$$

Since the sequences x are equally likely, it is sufficient to consider one fixed sequence y since it is only the relationship of x_i to y_i that matters, i.e., whether it is like or unlike. Thus we may write,

$$\sum_{y \epsilon Y_N} \left[\sum_{x \epsilon X_N} P_r(y \mid x)^{1/2} \right]^2 = 2^N \left[\sum_{x \epsilon X_N} P_r(y \mid x)^{1/2} \right]^2 \qquad (\text{II-3})$$

where y on the right is a fixed sequence.

In considering P(y x), we may distinguish two cases according as the initial state Z_1 is L or U, i.e., according as $x_1 = y_1$ or $x_1 \neq y_1$. Let Prob. $(Z_1 = L) = p_L$, Prob. $(Z_1 = U) = P_U$ as above.

Consider the case $Z_1 = L$ first. Let,

$$\sum_{\substack{x \epsilon X_N \\ Z_1 = L, Z_N = L}} \left[P_r(y \mid x)^{1/2} \right] = F_N$$

$$\sum_{\substack{x \epsilon X_N \\ Z_1 = L, Z_N = U}} \left[P_r(y \mid x)^{1/2} \right] = G_N$$

Then, the following pair of simultaneous difference equations is seen to hold:

$$F_{N+1} = q_1^{1/2} F_N + p_2^{1/2} G_N \qquad (\text{II-4})$$

$$G_{N+1} = p_1^{1/2} F_N + q_2^{1/2} G_N \qquad (\text{II-5})$$

with "initial" conditions,

$$F_2 = (p_L q_1)^{1/2}, \quad G_2 = (p_L p_1)^{1/2}$$

For the case $Z_1 = U$, we have the same pair of difference equations with different "initial" conditions.

$$F_2^* = (p_U p_2)^{1/2} \quad , \quad G_2^* = (p_U q_2)^{1/2}$$

where now,

$$\sum_{\substack{x \epsilon X_N \\ Z_1 = U, Z_N = L}} [P_r(y \mid x)^{1/2}] = F_N^*$$

$$\sum_{\substack{x \epsilon X_N \\ Z_1 = U, Z_N = U}} [P_r(y \mid x)^{1/2}] = G_N^*$$

The simultaneous difference equations may be solved by the method of generating functions.

Let, $F(X) = \sum_{N=2}^{\infty} F_N X^N$, $G(X) = \sum_{N=2}^{\infty} G_N X^N$

where X is a "dummy" variable in this context.

Then,

$$\frac{1}{X} \sum_{N=2}^{\infty} F_{N+1} X^{N+1} = q_1^{1/2} \sum_{N=2}^{\infty} F_N X^N + p_2^{1/2} \sum_{N=2}^{\infty} G_N X^N$$

or,

$$\frac{1}{X} [F(X) - F_2 X^2] = q_1^{1/2} F(X) + p_2^{1/2} G(X)$$

Similarly,

$$\frac{1}{X} [G(X) - G_2 X^2] = p_1^{1/2} F(X) + q_2^{1/2} G(X)$$

Solving for F(X) and G(X) in the first and second equations respectively,

$$F(X) = \frac{F_2 X^2}{1 - q_1^{1/2} X} + \frac{p_2^{1/2} X G(X)}{1 - q_1^{1/2} X}$$

$$G(X) = \frac{G_2X^2}{1-q_2^{1/2}X} + \frac{P_1^{1/2}XF(X)}{1-q_2^{1/2}X}$$

Therefore,

$$F(X) = \frac{F_2X^2(1-q_2^{1/2}X) + G_2p_2^{1/2}X^3}{(1-q_1^{1/2}X)\ (1-q_2^{1/2}X)-(p_1p_2)^{1/2}X^2}$$

$$G(X) = \frac{G_2X^2(1-q^{1/2}X) + F_2p_1^{1/2}X^3}{(1-q_1^{1/2}X)\ (1-q_2^{1/2}X)-(p_1p_2)^{1/2}X^2}$$

or,

$$F(X) = \frac{p_L^{1/2}\{q_1^{1/2}X^2(1-q_2^{1/2}X) + (p_1p_2)^{1/2}X^3\}}{X^2\{(q_1q_2)^{1/2}-(p_1p_2)^{1/2}\}-X(q_1^{1/2}+q_2^{1/2})+1}$$

$$G(X) = \frac{p_L^{1/2}\{p_1^{1/2}X^2(1-q_1^{1/2}X) + (p_1q_1)^{1/2}X^3\}}{X^2\{(q_1q_2)^{1/2}-(p_1p_2)^{1/2}\}-X(q_1^{1/2}+q_2^{1/2})+1}$$

then,

$$G(X)+F(X) = \frac{p_L^{1/2}\{[(p_1p_2)^{1/2}-(q_1q_2)^{1/2}]X^3+[p_1^{1/2}+q_1^{1/2}]X^2\}}{X^2\{(q_1q_2)^{1/2}-(p_1p_2)^{1/2}\}-X(q_1^{1/2}+q_2^{1/2})+1}$$

Similarly, for the case Z_1=U with "initial conditions $F_2^* = (p_up_2)^{1/2}$, $G_2^* = (p_uq_2)^{1/2}$

$$G^*(X)+F^*(X) = \frac{p_U^{1/2}\{[(p_1p_2)^{1/2}-(q_1q_2)^{1/2}]X^3+[p_2^{1/2}+q_2^{1/2}]X^2\}}{X^2\{(q_1q_2)^{1/2}-(p_1p_2)^{1/2}\}-X(q_1^{1/2}+q_2^{1/2})+1}$$

$$G(X)+F(X)+G^*(X)+F^*(X) =$$

$$\frac{[(p_1p_2)^{1/2}-(q_1q_2)^{1/2}]X^3(p_L^{1/2}+p_U^{1/2})+[p_L^{1/2}(p_1^{1/2}+q_1^{1/2})+p_U^{1/2}(p_2^{1/2}+q_2^{1/2})]X^2}{X^2\{(q_1q_2)^{1/2}-(p_1p_2)^{1/2}\}-X(q_1^{1/2}+q_2^{1/2})+1}$$

The denominator may be written,

$$X^2\{(q_1q_2)^{1/2} - (p_1p_2)^{1/2}\} - X(q_1^{1/2} + q_2^{1/2}) + 1 =$$

$$[(q_1q_2)^{1/2} - (p_1p_2)^{1/2}](X-r_1)(X-r_2)$$

$$r_1 = \frac{q_1^{1/2} + q_2^{1/2} + \sqrt{q_1+q_2+4(p_1p_2)^{1/2} - 2(q_1q_2)^{1/2}}}{2\{(q_1q_2)^{1/2} - (p_1p_2)^{1/2}\}}$$

$$r_2 = \frac{q_1^{1/2} + q_2^{1/2} - \sqrt{q_1+q_2+4(p_1p_2)^{1/2} - 2(q_1q_2)^{1/2}}}{2\{(q_1q_2)^{1/2} - (p_1p_2)^{1/2}\}}$$

$\sum_{x\epsilon X_N} P_r(y|x)^{1/2}$ is the coefficient of X^N in

$G(X) + F(X) + G^*(X) + F^*(X)$.

The coefficient of X^m in $\frac{1}{X-r}$ is $\frac{-1}{r^{m+1}}$

Therefore, the coefficient of X^m in

$$\frac{1}{X-r_1} - \frac{1}{X-r_2} \text{ is } \frac{r_1^{m+1} - r_2^{m+1}}{(r_1r_2)^{m+1}} = (r_1^{m+1} - r_2^{m+1})[(q_1q_2)^{1/2} - (p_1p_2)^{1/2}]^{m+1}$$

Furthermore,

$$\frac{1}{(X-r_1)(X-r_2)} = \frac{1}{r_1 - r_2}\left[\frac{1}{X-r_1} - \frac{1}{X-r_2}\right] =$$

$$\frac{[(q_1q_2)^{1/2} - (p_1p_2)^{1/2}]}{\sqrt{q_1+q_2+4(p_1p_2)^{1/2} - 2(q_1q_2)^{1/2}}}\left\{\frac{1}{X-r_1} - \frac{1}{X-r_2}\right\}$$

$$\sum_{x \epsilon X_N} P_r(y|x)^{1/2} = \frac{\left[(p_1 p_2)^{1/2} - (q_1 q_2)^{1/2}\right]^{N-1}}{\sqrt{q_1 + q_2 + 4(p_1 p_2)^{1/2} - 2(q_1 q_2)^{1/2}}}$$

$$\left\{(p_L^{1/2} + p_U^{1/2})(r_1^{N-2} - r_2^{N-2}) + \{p_L^{1/2}(p_1^{1/2} + q_1^{1/2}) + p_U^{1/2}(p_2^{1/2} + q_2^{1/2})\}(r_1^{N-1} - r_2^{N-1})\right\}$$

Note that $|r_1 r_2| > 1$, $|r_1| > 1$, $|r_2| < 1$.

Thus for large N, r_2^N may be omitted.

$\left[\sum_{X \epsilon X_N} P_r(y|X)^{1/2}\right]^2$ may be written

$\left[\lambda_o(\frac{1}{2})\right]^{2N_D}$

where

$$\lambda_o(\tfrac{1}{2}) = \frac{q_1^{1/2} + q_2^{1/2} + \sqrt{q_1 + q_2 - 2\sqrt{q_1 q_2} + 4\sqrt{p_1 p_2}}}{2}$$

and D is independent of N.

Finally,

$$\overline{P}_{em} \leq (M-1)^e \sum_{y \epsilon Y_N} \left[\sum_{X \epsilon X_N} P(X) P_r(y|X)^{1/1+e}\right]^{1+e}$$

Let e=1 and upperbound M-1 by $M=2^{NR}$, then, for large N,

$$\overline{P}_{em} < 2^{-N}\left[-R+1-2\log_2\{\lambda_o(\tfrac{1}{2})\}\right]$$

which is identical to a previously known result (Reference 2).

References

1. "A Simple Derivation of the Coding Theorem and Some Applications," Robert M. Gallager, IEEE Trans. on Information Theory, vol. IT-11, pp3-18, January, 1965.

2. "Sequential Decoding on a Channel with Simple Memory", J. E. Savage, Group Report 1964-5, Lincoln Laboratory, MIT, April 2, 1964.

Appendix III

THE EFFECT OF PHASE ERRORS ON COHERENT DEEP SPACE TELEMETRY SYSTEMS UTILIZING SEQUENTIAL DECODING

III. 1. INTRODUCTION

In most coherent binary communication systems, the data rate and the required error rate dictate a ratio of carrier power to thermal noise density which is sufficiently high to allow the tracking of the carrier phase with negligible error. In low data rate communication systems, usually required for deep-space missions, signal conditions are often less favorable and the loss of correlation may cause appreciable deterioration in performance. When sequential decoding is used, the problem is aggravated for two reasons. First, since sequential decoding is an extremely efficient binary signaling method, the received signal power requirements are further reduced. Second, as will be shown subsequently, the various circuits (such as the Costas and squaring loop used for carrier-tracking in PSK systems) exhibit a noise enhancement which is inversely related to the received energy per symbol per noise density ratio. Since this ratio is low in receivers using sequential decoders, the problem of obtaining small mean-squared phase errors is further aggravated.

The primary cause of the phase errors is the instability of the oscillators utilized in the system. Thus, the problem reduces to one of a tradeoff between loss of system efficiency (or capacity) and increased oscillator stability requirements. This tradeoff is the subject of this technical note.

III. 2. THE THERMAL NOISE ENHANCEMENT IN THE CARRIER-TRACKING CIRCUIT CAUSED BY THE PSK MODULATION

Phase-shift keying is a suppressed carrier modulation technique which requires circuitry that will reinsert the carrier, in conjunction with the carrier-tracking phase-locked loop generating the coherent reference. The four types of circuits which can be used for this purpose are the "Costas" loop (Reference 1), the squaring loop (Reference 2), the decision-directed-feedback loop (Reference 3), and the optimum loop (References 3 and 4).

It was shown in Reference 3 that all these loops are equivalent to a phase-locked loop which tracks an unmodulated, but phase disturbed, carrier imbedded in white noise which is enhanced by a factor δ. For the Costas loop and squaring loop, assuming a predetection bandwidth equal to the bit rate, δ is given by:

$$\delta = 1 + \frac{1}{2\left(\frac{E_s}{N_o}\right)} \qquad \text{(III-1)}$$

where

$$\frac{E_s}{N_o} = \text{Energy/symbol/noise density ratio}$$

For the delayed decision feedback loop

$$\delta = \frac{1}{(1-2P_e)^2} \qquad \text{(III-2)}$$

where P_e = Binary error rate.

For the optimum loop, δ was computed in (III-2). The value of δ for the various loops is shown in Figure 1, reproduced from Reference 3.

III.3. PHASE ERRORS IN CARRIER-TRACKING FILTERS

In most practical systems there are three sources of phase error:

a. The instability of the oscillators used to generate a carrier

b. The thermal noise of the receiver

c. The phase perturbations associated with the Doppler effect caused by the relative motion of the system nodes.

In many instances, the phase error caused by the Doppler effect is negligible. An example, fairly common in practice, is the case in which the effects of all but a finite number of the derivatives of transmitter motion are negligible, in which case a loop of sufficiently high order will track the Doppler disturbance with a phase error, which, after the transients have subsided, is negligible. Another example occurs when the radial acceleration component is small, or it is known in advance with sufficient accuracy to permit cancellation of its effects. When this is the case, the phase error may be obtained in an extremely convenient form.

As pointed out in Reference 5 it has been verified experimentally that in tracking filters the "flicker" noise component of phase error predominates. The one-sided power spectrum of flicker noise is given by Reference 6.

$$H_{<\phi>}(w) = \frac{C_1}{|w^3|} \quad \text{rad}^2/\text{rad} \qquad \text{(III-3)}$$

It is related to the oscillator stability S_τ by

$$S_\tau^2 = \frac{2C_1}{\pi w_o^2} \left(1.04 + 1/2 \log \frac{T}{2\tau}\right) \qquad \text{(III-4)}$$

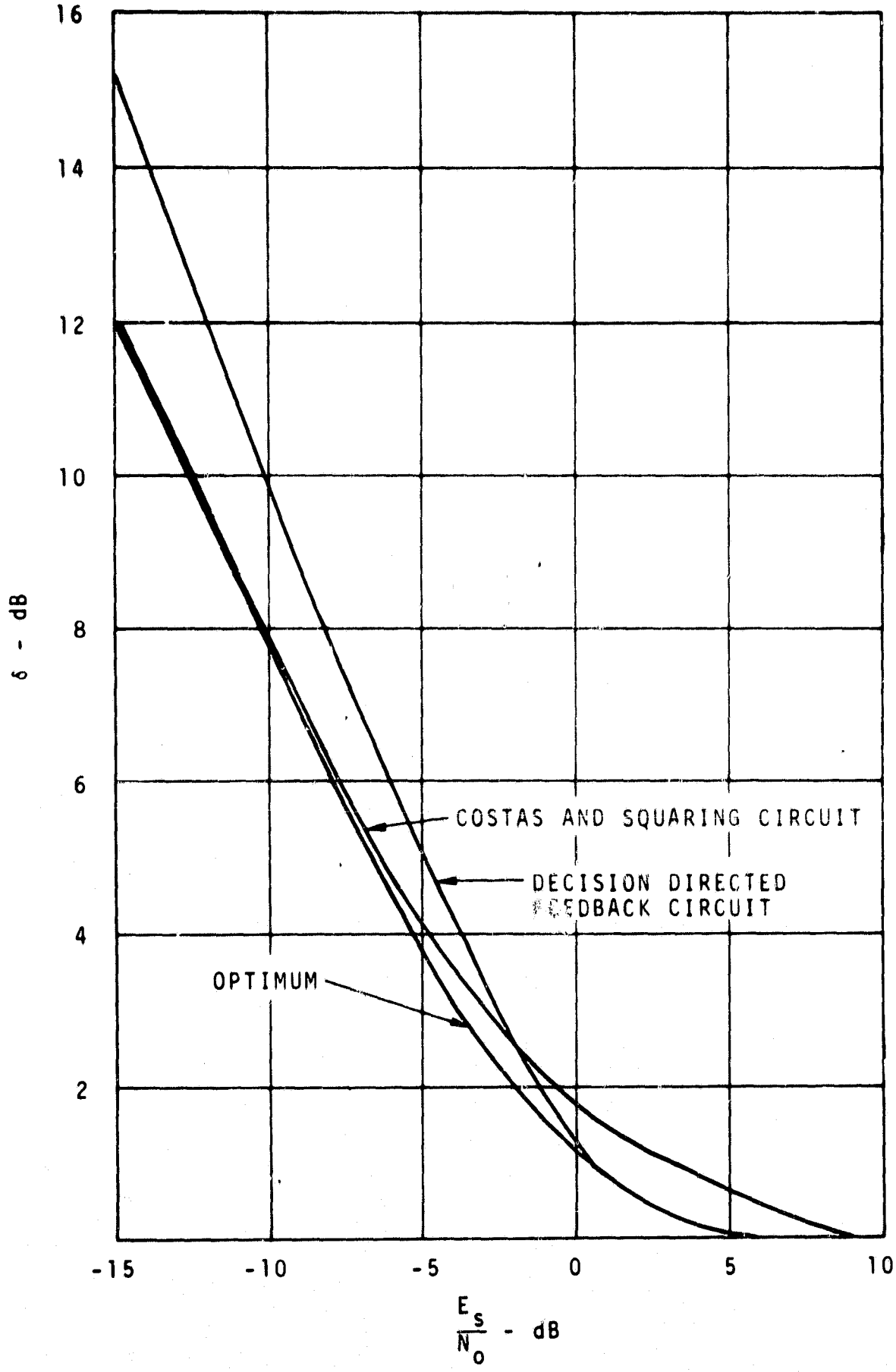


FIGURE III-1. Noise Enhancement δ vs. Energy/Symbol/Noise Density for Various Tracking Circuits

where

w_o = Carrier frequency radians/sec

τ = Time over which the short term stability S_τ is specified

T = Long-term averaging used in frequency measurement

A value of $\frac{T}{\tau} = 100$ is typical in laboratory stability measurements hence

$$C_1 = \frac{w_o^2 S_\tau^2}{1.91} \qquad \text{(III-5)}$$

The thermal noise causes a phase disturbance of one-sided power spectral density:

$$N'(w) = \frac{N_o \delta}{4\pi P} \text{ rad}^2/\text{rad} \qquad \text{(III-6)}$$

where

P = Received signal power, watts

N_o = One-sided noise power density, watts/Hz

δ = Noise enhancement described in the previous paragraph.

The problem of designing a matched filter to track the phase disturbance in the presence of white phase noise given by Reference 6 is complicated by the fact that the flicker-noise spectrum has an odd number of poles, which leads to a nonrealizable filter. However, it may be approximated to a specified degree of accuracy with a realizable filter, and thus, at least in principle, an optimum, i.e., minimum mean-squared, error filter can be synthesized. As shown in References 7 and 8, when the process to be estimated has the power spectrum $S_m(w)$ and is embedded in white noise of single-sided power density N', (watts/rad), the minimum mean-squared error is given by:

$$\sigma_\epsilon^2 = N' \int_0^\infty \ln \left[1 + \frac{2S_m(w)}{N'}\right] dw; \text{ watts} \qquad \text{(III-7)}$$

Substituting Equations (III-3) (III-5) and (III-6) for $S_m(w)$, C_1 and N' respectively,

$$\sigma_\epsilon^2 = \frac{N_o \delta}{4\pi P} \int_0^\infty \ln \left[1 + \frac{8\pi w_o^2 S_\tau^2 P}{1.91 N_o \delta |w|^3}\right] dw, \text{ rad}^2 =$$

Introducing the change of variables

$$x = w\left(\frac{1.91\, N_o \delta}{8\pi w_o^{\,2} S_\tau^{\,2} P}\right)^{1/3}$$

then

$$\sigma_\epsilon^2 \doteq \frac{N_o \delta}{4\pi P}\left(\frac{8\pi w_o^{\,2} S_\tau^{\,2} P}{1.91\, N_o \delta}\right)^{1/3} \int_o^\infty \ln\left(1+\frac{1}{x^3}\right)\, dx.$$

Introducing the change of variables

$$u = 1+\frac{1}{x^3};\quad x^3 = \frac{1}{u-1};\quad x = \frac{1}{(u-1)^{1/3}};\quad dx = -1/3\,(u-1)^{-1/3}\, du$$

then

$$\int_o^\infty \ln\left(1+\frac{1}{x^3}\right) dx = \int_o^\infty \ln u \cdot \frac{1}{(u-1)^{1/3}}\, du = \frac{\ln u}{(u-1)^{1/3}}\Bigg|_o^1$$

$$- \int_\infty^1 \frac{du}{(u-1)^{1/3} u}$$

From Reference 9, with a = u, b = 1, v = 1/3

$$\int_1^\infty \frac{du}{-u(u-1)^{1/3}} = \pi\, \mathrm{Cosec}\,\frac{\pi}{3} = -\frac{2\pi}{\sqrt{3}}$$

hence

$$\int_o^\infty \ln\left(1+\frac{1}{x^3}\right) dx = \ln \frac{u}{(u-1)^{1/3}}\Bigg|_\infty^1 + \frac{2\pi}{\sqrt{3}}$$

for $u_o = \infty$ or 1

$$\lim_{u \to u_o} \frac{\ln u}{(u-1)^{1/3}} = \lim_{u \to u_o} \frac{\frac{d}{du} \ln u}{\frac{d}{du}(u-1)^{1/3}} = \lim_{u \to u_o} \frac{3(u-1)^{2/3}}{u} = 0$$

thus

$$\int_0^\infty (1 + \frac{1}{x^3})\, dx = \frac{2\pi}{\sqrt{3}}$$

Substituting above

$$\sigma_\epsilon^2 = 0.68\, S^{2/3}\ \text{rad}^2 \qquad \text{(III-8)}$$

where

$$S = \frac{S_{\tau_o} w}{\frac{P}{N_o \delta}} \qquad \text{(III-9)}$$

For the second order tracking filter of single-sided noise bandwidth B_L, the mean-squared phase error caused by flicker noise is given by (Reference 5)

$$\sigma_\theta{}^2{}_M = \frac{0.884 C_1}{B_L{}^2} \qquad \text{(III-10)}$$

and the mean-squared phase error caused by thermal noise of single-sided power density N_o is

$$\sigma_\theta{}^2{}_N = \frac{N_o \delta}{2P} B_L \qquad \text{(III-11)}$$

Defining

$$\sigma_\theta{}^2 = \sigma_{\theta_M}{}^2 + \sigma_{\theta_N}{}^2 \qquad \text{(III-12)}$$

then substituting Equations (III-10) and (III-11) and solving for B_L in

$$\frac{\delta \sigma_\theta{}^2}{\delta B_L} = 0$$

with C_1 given by Equation (III-5) results in

$$\sigma_\theta^2 = 1.46 S^{2/3} \text{rad}^2 \qquad \text{(III-13)}$$

where S is defined in Equation (III-9).

The mean-squared phase error as a function of the parameter S for both the optimum and second order tracking filter is shown in Figure 2. As seen from Equations (III-8) and (III-13) the potential reduction is mean-squared phase error from second order to more complex filters is 3.4 dB.

III.4. DEGRADATION OF THE SEQUENTIAL DECODER PERFORMANCE CAUSED BY THE NOISY PHASE REFERENCE

The degradation caused by phase error on the sequential decoder performance was analyzed in Reference 10. The results obtained in this reference are directly applicable, assuming that the derivations of the previous paragraph are estimates based on a linearized phase-locked loop model. Assuming that the phase errors are small and the linear model is valid, the mean-squared phase error caused by thermal noise alone is given by

$$\sigma_\theta^2 = \frac{1}{2\frac{P}{N_o B_L}} \qquad \text{(III-14)}$$

where $\frac{P}{N_o B_L}$ is the signal to noise ratio in the loop noise bandwidth. In Reference 10 the degradation in performance caused by a noise reference is given in terms of the signal to noise ratio in the loop noise bandwidth. Equation (III-14) permits the direct utilization of the results of Equation (III-10) when σ_θ^2 is computed.

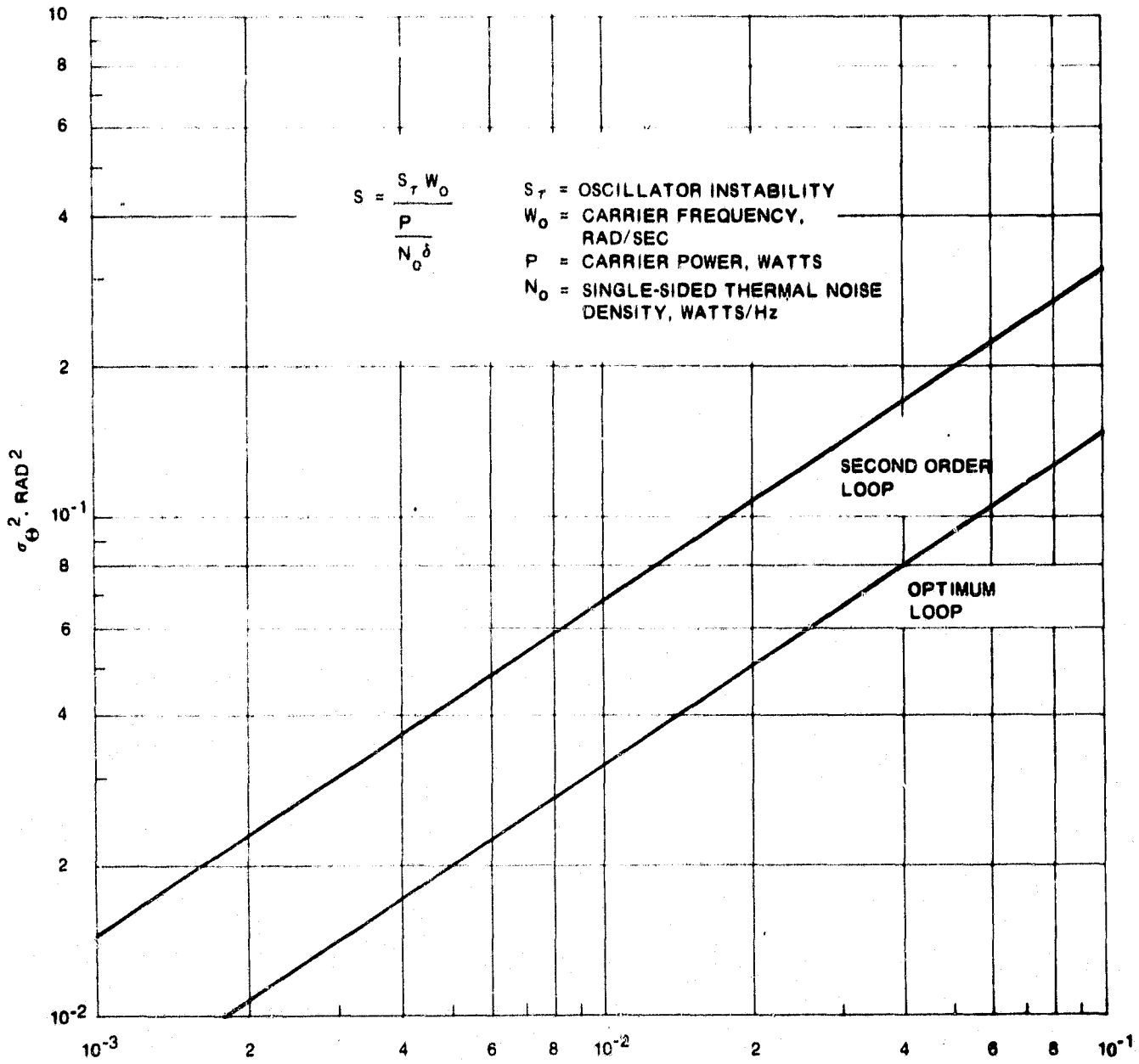


FIGURE III-2. Mean-Squared Phase Error Versus S.

REFERENCES

1. Costas, J. P., "Synchronous Communications," Proc. IRE, December 1956.

2. Lindsey, W. C., "Phase Shift Keyed Signal Detection with Noisy Reference Signals," IEEE Transactions on Aerospace and Electronic Systems, July 1966.

3. Frenkel, G., "The Optimum Carrier-Tracking Circuit in Binary Coherent Communication Systems," Communications Systems Incorporated Internal Report, August 1967.

4. J. Stiffler, "A Comparison of Several Methods of Subcarrier Tracking," JPL Space Programs Summary, No. 37-37 Vol. IV., for the period December 1, 1965, to January 31, 1966. February 28, 1966.

5. Sydnor, R., Caldwell, J. J., and Rose, B. E., "Frequency Stability Requirements for Space Communications and Tracking Systems," Proc. IEEE, February 1966.

6. Cutler, L. S., and Searle, C. L., "Some Aspects of the Theory and Measurement of Frequency Fluctuations in Frequency Standards," Proc. IEEE, February 1956.

7. Yowits, M. C. and Jackson, J. L., "Linear Filter Optimization with Game Theory Considerations," IRE Nat. Conv. Record p/4, March 1955.

8. Viterbi, A. J., "Principles of Coherent Communications," McGraw-Hill Book Company, 1966.

9. I. S. Gradshteyn and I. M. Ryzhik, "Table of Integrals, Series, and Products," Sec. 3. 1974, Academic Press, New York and London, 1965.

10. P. A. Tenkhoff, "Coherent Communication with a Noisy Phase Reference," Communications Systems Incorporated Interoffice Correspondence, 10 February 1967.

Appendix IV

EVALUATION OF B.S.C. CAPACITY AT SMALL R/W

From Paragraph 5.2.1

$$\frac{R}{W} = \left[2 \quad 1 + p \log_2 p + (1-p) \log_2 (1-p)\right] \tag{IV-1}$$

where

$$p = \frac{1}{2} \operatorname{erfc} \sqrt{\frac{R}{2W} \cdot \frac{E_b}{N_o}} \tag{IV-2}$$

Note that when $R/W \rightarrow O$, $p \rightarrow 1/2$. Thus, define

$$p = \frac{1}{2}(1-\epsilon) \tag{IV-3a}$$

$$1-p = \frac{1}{2}(1+\epsilon) \tag{IV-3b}$$

$$\log_2 p = -1 + [\ell n\,(1-\epsilon)/\ell n\ 2]$$

$$\log_2 (1-p) = -1 + [\ell n\,(1+\epsilon)/\ell n\ 2]\,.$$

Use of

$$\ell n\,(1+\epsilon) = \epsilon - \frac{1}{2}\epsilon^2 + \frac{1}{3}\epsilon^3 - \frac{1}{4}\epsilon^4 + \ldots, \quad -1 < \epsilon \leq 1$$

in the above provides

$$1 + p \log_2 p + (1-p) \log_2 (1-p) \approx \frac{1}{2 \ln 2} \left(\epsilon^2 + \frac{1}{6}\epsilon^4\right) \tag{IV-4}$$

$$\approx \epsilon^2/(2 \ln 2)$$

after terms of order E^4 and higher are dropped. From Equation (IV-2)

$$p = \frac{1}{2} \operatorname{erfc} Z = \frac{1}{2}\left[1 - \frac{2}{\sqrt{\pi}} \int_o^Z e^{-t^2} \, dt\right]$$

where $Z = \sqrt{\frac{R}{2W} \cdot \frac{E_b}{N_o}}$. For small values of Z (small R/W),

$$p \approx \frac{1}{2}\left[1 - \frac{2}{\sqrt{\pi}}\int_0^Z (1 - t^2)\,dt\right] = \frac{1}{2}\left[1 - \frac{2}{\sqrt{\pi}}\left(Z - \frac{1}{3}\,Z^3\right)\right] .$$

$$\approx \frac{1}{2}\left[1 - (2Z/\sqrt{\pi})\right]$$

Compare this result with Equation (IV-2) and it is seen that

$$\epsilon = \frac{2Z}{\sqrt{\pi}} = \sqrt{\frac{2}{\pi} \cdot \frac{R}{W} \cdot \frac{E_b}{N_o}} \quad . \qquad \text{(IV-5)}$$

Combining Equations (IV-1)(IV-4) and (IV-5) yields

$$\frac{R}{W} \approx \frac{\epsilon^2}{\ell n\ 2} \approx \frac{2}{\pi} \cdot \frac{R}{W} \cdot \frac{E_b}{N_o} \cdot \frac{1}{\ell n\ 2}$$

Solving for E_b/N_o,

$$\frac{E_b}{N_o} = \frac{\pi}{2}\ \ell n\ 2 \qquad \text{(IV-6)}$$

which becomes an exact result in the limit as R/W→O.

Appendix V

IBM 1130 DIGITAL COMPUTER SIMULATION OF A SEQUENTIAL DECODER

V.1 IDENTIFICATION OF VARIABLES USED IN SIMULATION PROGRAM

BSUM	Total number of decoded information digits
CAVG	Average number of computations per information bit per block
CSAVG	Average number of computations per information bit for N blocks
CSUM	Total number of computations to decode all information bits in N blocks
I1, I2, I3	Metric Values
II, IIMUL, IMULT	Parameters to extend period of random number generator
INDEX	Worst node indicator
INFO	Received information symbols
IRTHR	Value of running threshold
ISPAC	Threshold spacing
Ix	Initial value for random number generator
JPAR	Received parity symbols
JTAP	Code tap connection
JX	Hypothesized information symbols
KPOIN	Pointer position in tree
KTEST	Node currently being hypothesized
KTHET	Parameter θ in Fano algorithm
LMBD	Incremental metric
MET	Value of total metric

METSB	Value of metric at threshold comparison
NBLOC	Number of information digits per block
NCOMP	Running total of number of computations per block
NERR	Number of undetected errors per block
NIN	Number of input symbol errors per block
NMAX	Maximum number of computations allowed per block
PERR	Binary symmetric channel probability of error

V.2 LISTING OF FORTRAN SEQUENTIAL DECODING PROGRAM

Following is a listing for the sequential decoding program used for rate 1/2 systematic codes on the binary symmetric channel. Extension to other channels, rates, and nonsystematic codes is straight forward. Subroutines are listed in the following paragraphs.

```
* A"   SEGDE                                                                    SEGDE002
      DIMENSION INDEX(40 ),L"RD(4 ,2),I"FO(400),JPAR(M 1),JTAP( ,)             SEGDE003
     1,JX(400),II"UL(30)                                                       SEGDE004
      READ (2,1) PERR,ISPAC                                                    SEGDE005
      READ (2,7) I1,I2,I3                                                      SEGDE006
      READ (2,8) NBLOC,LBLOC,"AX,IX,LTAIL                                      SEGDE007
      READ (2,3) II"UL                                                         SEGDE008
      READ (2,2) JTAP                                                          SEGDE009
      WRITE (3,15)                                                             SEGDE010
      WRITE (3,16) PERR,LTAIL,ISPAC,NBLOC,LBLOC,"AX,IX                         SEGDE011
      WRITE (3,13)                                                             SEGDE012
      P1=32767.*PERR+1.                                                        SEGDE013
      IPERR=P1                                                                 SEGDE014
      LTOT=LBLOC+LTAIL                                                         SEGDE015
      LTOTP=LTOT+31                                                            SEGDE016
      LBLKP=LBLOC+31                                                           SEGDE017
      CALL MSKPK (JTAP)                                                        SEGDE018
      CSUM=0.                                                                  SEGDE019
      BSUM=0.                                                                  SEGDE020
      NN=0                                                                     SEGDE021
      NERRT=0                                                                  SEGDE022
      II=1                                                                     SEGDE023
C        INITIALIZE ENCODER                                                    SEGDE024
      DO 200 J=1,32                                                            SEGDE025
  200 JX(J)=0                                                                  SEGDE026
C        GENERATE ENCODER TAIL                                                 SEGDE027
      DO 210 J=LBLKP,LTOTP                                                     SEGDE028
      INFO(J)=0                                                                SEGDE029
  210 JPAR(J)=0                                                                SEGDE030
C        BEGIN DECODER OPERATION                                               SEGDE031
      DO 10 NN=1,NBLOC                                                         SEGDE032
C        INITIALIZE PARAMETERS                                                 SEGDE033
      "ET=0                                                                    SEGDE034
      IRTHR=0                                                                  SEGDE035
      KTHET=0                                                                  SEGDE036
      NCOMP=0                                                                  SEGDE037
      NERR=0                                                                   SEGDE038
      NIN=0                                                                    SEGDE039
      I"ULT=II"UL(II)                                                          SEGDE040
      KPOIN=31                                                                 SEGDE041
C        GENERATE RECEIVED SY"BOLS                                             SEGDE042
      DO 220 N=32,LPLKP                                                        SEGDE043
      CALL RECSY (IX,IY,I"ULT)                                                 SEGDE044
      IX=IY                                                                    SEGDE045
      IF (IX-IPERR) 230,230,240                                                SEGDE046
  230 INFO(N)=1                                                                SEGDE047
      GO TO 250                                                                SEGDE048
  240 INFO(N)=0                                                                SEGDE049
  250 CALL RECSY (IX,IY,I"ULT)                                                 SEGDE050
      IX=IY                                                                    SEGDE051
      IF (IX-IPERR) 260,260,270                                                SEGDE052
  260 JPAR(N)=1                                                                SEGDE053
      GO TO 220                                                                SEGDE054
  270 JPAR(N)=0                                                                SEGDE055
  220 CONTINUE                                                                 SEGDE056
C        BEGIN FANO ALGORITHY                                                  SEGDE057
C        ADVANCE NODE - CALCULATE PARITY
```

```
  100 KTEST=KPOI,+1                                                     SE.JDE058
      INDEX(KTEST)=1                                                    SEJDE059
      JX(KTEST)=INFO(KTEST)                                             SEJDE060
      KCK=KTEST-31                                                      SEJDE061
      CALL XSAND (JX(KCK),JPARC)                                        SEJDE062
      JTEST=JPAR(KTEST)-JPARC                                           SEJDE063
      IF (JTEST) 610,620,610                                            SEJDE064
  610 LMBD(KTEST,1)=I2                                                  SEJDE065
      LMBD(KTEST,2)=I2                                                  SEJDE066
      GO TO 305                                                         SEJDE067
  620 LMBD(KTEST,1)=I1                                                  SEJDE068
      LMBD(KTEST,2)=I3                                                  SEJDE069
C        ENTER FORWARD SEARCH                                           SEJDE070
  305 IK=INDEX(KTEST)                                                   SEJDE071
      METSB=MET+LMBD(KTEST,IK)                                          SEJDE072
      NCOMP=NCOMP+1                                                     SEJDE073
      IF (NCOMP-NMAX) 310,500,500                                       SEJDE074
  310 IF (METSB-IRTHR) 400,320,320                                      SEJDE075
  320 KPOIN=KPOIN+1                                                     SEJDE076
      IF (KTHET) 330,330,340                                            SEJDE077
  330 MET=METSB                                                         SEJDE078
C        TIGHTEN RUNNING THRESHOLD                                      SEJDE079
  350 INC=IRTHR+ISPAC                                                   SEJDE080
      IF (MET-INC) 120,355,355                                          SEJDE081
  355 IRTHR=INC                                                         SEJDE082
      GO TO 350                                                         SEJDE083
  340 INC=IRTHR+ISPAC                                                   SEJDE084
      IF (MET-INC) 370,360,360                                          SEJDE085
  360 IF (METSB-INC) 370,380,380                                        SEJDE086
  380 MET=METSB                                                         SEJDE087
      GO TO 120                                                         SEJDE088
  370 KTHET=0                                                           SEJDE089
      GO TO 330                                                         SEJDE090
C        ENTER BACK SEARCH                                              SEJDE091
  400 KTHET=1                                                           SEJDE092
  410 IF (KPOIN-31) 420,460,420                                         SEJDE093
  420 KTEST=KPOIN-1                                                     SEJDE094
      IK=INDEX(KPOIN)                                                   SEJDE095
      METSB=MET-LMBD(KPOIN,IK)                                          SEJDE096
      NCOMP=NCOMP+1                                                     SEJDE097
      IF (NCOMP-NMAX) 430,500,500                                       SEJDE098
  430 IF (METSB-IRTHR) 460,440,440                                      SEJDE099
  440 KPOIN=KPOIN-1                                                     SEJDE100
      MET=METSB                                                         SEJDE101
      KTEST=KPOIN+1                                                     SEJDE102
      IF (INDEX(KTEST)-2) 450,410,410                                   SEJDE103
  450 INDEX(KTEST)=INDEX(KTEST)+1                                       SEJDE104
      JJT=JX(KTEST)-1                                                   SEJDE105
      JX(KTEST)=IABS(JJT)                                               SEJDE106
      GO TO 305                                                         SEJDE107
  460 IRTHR=IRTHR-ISPAC                                                 SEJDE108
  120 IF (KPOIN-LTOTP) 100,110,110                                      SEJDE109
  110 CONTINUE                                                          SEJDE110
C        TABULATE INPUT AND OUTPUT ERRORS                               SEJDE111
  500 DO 560 J=32,LBLKP                                                 SEJDE112
      IF (JX(J)) 520,520,510                                            SEJDE113
  510 NERR=NERR+1                                                       SEJDE114
```

```
      IF (INFO(J)) 530,540,530                                          SEQDE115
  530 NIN=NIN+1                                                         SEQDE116
  540 IF (JPAR(J)) 550,560,550                                          SEQDE117
  550 NIN=NIN+1                                                         SEQDE118
  560 CONTINUE                                                          SEQDE119
      XNCOM=NCOMP                                                       SEQDE120
      XBLOC=LTOT                                                        SEQDE121
      NERRT=NERRT+NERR                                                  SEQDE122
      CAVG=XNCOM/XBLOC                                                  SEQDE123
      WRITE (3,9) NN, NIN, NERR, NCOMP, CAVG                            SEQDE124
      CSUM=CSUM+XNCOM                                                   SEQDE125
      BSUM=BSUM+XBLOC                                                   SEQDE126
      NNN=NNN+1                                                         SEQDE127
      IF (NNN-15) 10,10,31                                              SEQDE128
   31 NNN=0                                                             SEQDE129
      II=II+1                                                           SEQDE130
   10 CONTINUE                                                          SEQDE131
      CSAVG=CSUM/BSUM                                                   SEQDE132
      WRITE (3,14) CSAVG                                                SEQDE133
      XNBLK=NBLOC                                                       SEQDE134
      XBLOK=LBLOC                                                       SEQDE135
      XBLOK=XBLOK*XNBLK                                                 SEQDE136
      XNER=NERRT                                                        SEQDE137
      PUNDE=XNER/XBLOK                                                  SEQDE138
      WRITE (3,12) PUNDE                                                SEQDE139
    1 FORMAT (F10.0,I5)                                                 SEQDE140
    2 FORMAT (32I1)                                                     SEQDE141
    3 FORMAT (15I5)                                                     SEQDE142
    7 FORMAT (3I5)                                                      SEQDE143
    8 FORMAT (5I5)                                                      SEQDE144
    9 FORMAT (3X,I5,8X,I3,10X,I5,15X,I5,12X,F6.2)                       SEQDE145
   12 FORMAT (1H0,35HPROBABILITY OF UNDETECTED ERROR =  ,F10.7)         SEQDE146
   13 FORMAT (1H0,76HBLOCK NUMBER  INPUT ERRORS  OUTPUT ERRORS    NO. COSEQDE147
     1MPUTATIONS   AVG NO. COMP)                                        SEQDE148
   14 FORMAT (1HC,34HAVERAGE NUMBER OF COMPUTATIONS =  ,F6.2)           SEQDE149
   15 FORMAT (1H1,2X,42HPERR  LTAIL  ISPAC  NBLOC  LBLOC  NMAX  IX)     SEQDE150
   16 FORMAT (F7.4,6I7)                                                 SEQDE151
      CALL EXIT                                                         SEQDE152
      END                                                               SEQDE153
                                                                        SEQDE154
```

V.3 PARITY CHECK SUBROUTINES (MSAND, MSKPK)

Two separate subroutines were written in assembly language as a one double entry module for the IBM 1130. The purpose of the subroutines was to implement more rapid parity check calculations than would be available with FORTRAN. Each of these is considered below.

MSKPK This subroutine packs 32 16-bit words, each being a binary "0" or "1" into a single 32-bit word. MSKPK is used to represent the tap connection set as a single 32-bit word.

MSAND This subroutine performs three functions. First, it uses the bit packing routine of MSKPK to pack the 32 words representing the hypothesized contents of the convolutional encoder into a single 32-bit word. Secondly, this double word is logically ANDed with the double word and stored in MSKPK (top configuration). Thirdly, the resultant word is examined for even or odd parity.

A listing is shown below:

```
        ENT     MSAND                                           MSK32-001
        ENT     MSKPK                                           MSK32-001
*                                                               MSK32-001
*                                                               MSK32-001
*                                                               MSK32-001
MSAND   DC      0     INDIRECT ADDR OF MATRIX.                  MSK32-001
        LD      MSAND     LOAD IND ADDR INTO ACC.               MSK32-001
        STO     LOAD1+1   STORE IND ADDR IN LOAD1+1.            MSK32-001
        A       ONE  LOCATE ADDR TO STORE PARITY                MSK32-001
        STO     HERE+1    STORE PARITY ADDR IN HERE+1           MSK32-001
        A       ONE  LOCATE MAINLINE RETURN ADDR.               MSK32-001
        STO     XIT2+1    STORE IT AT XIT2+1.                   MSK32-001
DUMY2   LD    L ZOT  SETUP FOR BYPASSING MSKPK RETURN           MSK32-001
        LD      DUMY2+1   TO MAINLINE BY PLACING ADDR           MSK32-001
        STO     XIT1+1    OF ZOT IN XIT1+1.                     MSK32-001
DUMY3   LD    L TEMP SETUP STORAGE FOR WORD TO BE                MSK32-001
        LD      DUMY3-1   ADDED WITH MSK BY PLACING             MSK32-001
        STO     STORE+1   ADDR OF TEMP IN STORE+1.              MSK32-001
        B       LOAD1           GO TO PACK ROUTINE.             MSK32-001
*                                                               MSK32-001
*                                                               MSK32-001
*                                                               MSK32-001
MSKPK   DC      0     INDIRECT ADDR OF MSK ARRAY.               MSK32-001
        LD      MSKPK     LOAD IND ADDR INTO ACC.               MSK32-001
        STO     LOAD1+1   STORE IND ADDR IN LOAD1+1.            MSK32-001
        A       ONE  LOCATE MAINLINE RETURN ADDR.               MSK32-001
        STO     XIT1+1    STORE IT AT XIT1+1.                   MSK32-001
DUMY1   LD    L MSK  SETUP STORAGE FOR PACKED MASK BY           MSK32-001
        LD      DUMY1-1   STORING ADDR OF MSK IN                MSK32-001
        STO     STORE+1   STORE+1.                              MSK32-001
LOAD1   LD    L *-*  LOAD ADDR OF HI-CORE WORD OF 32            MSK32-001
        S       THRT1     WORD MATRIX. LOCATE ADDR OF           MSK32-001
        STO     LOOP+1    LO-CORE WORD. STORE AT LOOP           MSK32-001
        LDD     ZERO      +1 AND ZERO ACC AND Q REGS.           MSK32-001
        STO     PRCNT     ZERO PARITY-COUNT STORAGE.            MSK32-001
LOOP    OR    L *-*  OR MATRIX WORD WITH ACC.                   MSK32-001
        RTE     1     ROTATE ACC AND Q 1 BIT.                   MSK32-001
        STD     TEMP STORE ACC AND Q IN TEMP.                   MSK32-001
        MDX   L LOOP-1,1  ADD 1 TO MATRIX WORD ADDR.            MSK32-001
        LD    I LOAD1+1   LOAD ADDR OF HI-CORE WORD.            MSK32-001
        S       LOOP+1    SUBTR ADDR OF CURRENT WORD.           MSK32-001
        BSC   L EORTN,Z+  GO TO EORTN IF 32 WORDS ARE           MSK32-001
        LDD     TEMP      PACKED. ELSE RESTORE ACC              MSK32-001
        B       LOOP      AND Q WITH TEMP, LOOP AGAIN           MSK32-001
EORTN   LDD     TEMP LOAD 32 PACKED WORDS, EXCHANGE             MSK32-001
        XCH           ACC AND Q, AND STORE AT EITHER            MSK32-001
STORE   STD   L *-*   MSK OR TEMP. IF MSAND, GO TO ZOT         MSK32-001
XIT1    B     L *-*   IF MSKPK, RETURN TO MAINLINE.            MSK32-001
ZOT     LDD     MSK   LOAD MSK INTO ACC AND Q.                  MSK32-001
        AND     TEMP AND TEMP WITH ACC PART OF MASK.            MSK32-001
        XCH           SETUP FOR AND WITH Q PART.                MSK32-001
        AND     TEMP+1    AND TEMP+1 WITH OLD Q.                MSK32-001
        XCH           PUT THEM BACK IN PROPER SEQUENCE          MSK32-001
LSHFT   SLT     1     START COUNTING 1-BITS IN ANDED            MSK32-001
        BC      COUNT     WORDS.                                MSK32-001
        BSC   L TEST,+-   IF ACC=0, GO TO Q=0 TEST.             MSK32-001
        B       LSHFT     ELSE GO ON COUNTING.                  MSK32-001
```

```
TEST  XC-          SETUP Q FOR NEG TEST. IF ACC AND  MSK32-001
      BSC  L EQROD,+- Q=0, GO TO EQROD. ELSE RE-  MSK32-001
      XCH          STORE ACC AND Q.               MSK32-001
      B      LSHFT     GO ON COUNTING.            MSK32-001
COUNT MDX  L PRCNT,1   A 1-BIT ADDS 1 TO PARITY   MSK32-001
      B      LSHFT     COUNT. GO BACK FOR MORE.   MSK32-001
EQROD LD     PRCNT     LOAD 1-BIT COUNT. IF COUNT MSK32-001
      BSC  L ODD,E     IS ODD GO TO ODD,ELSE LOAD MSK32-001
      LD     ZERO ZERO IN ACC AND STORE IT AT ADDR MSK32-001
HERE  STO  I *-*   IN MAINLINE (SEE MSAND+4).     MSK32-001
      B      XIT2      BRANCH TO MAINLINE RETURN. MSK32-001
ODD   LD     ONE   PARITY WAS ODD. LOAD 1 IN ACC  MSK32-001
      B      HERE AND BRANCH TO HERE.             MSK32-001
XIT2  B    L *-*   RETURN TO MAINLINE.           MSK32-001
ZERO  DEC      0   STORAGE FOR 2-WORD CONSTANT=0  MSK32-001
TEMP  DEC      0     TEMP STORAGE FOR MISC WORK   MSK32-001
MSK   DEC      0     STORAGE FOR 2-WORD MASK      MSK32-001
ONE   DC      /0001      STORAGE FOR HEX CONSTANT=1 MSK32-001
THRT1 DC      /001F      STORAGE FOR HEX CONSTANT=31MSK32-001
PRCNT DC       0     STORAGE FOR PARITY-COUNT(1-BITS)MSK32-001
      END                                         MSK32-001
```

V.4 RANDOM NUMBER GENERATOR (RECSM)

Two important modifications were made to the IBM supplied random number generator. The first was to remove the floating point divide and use uniformly distributed integers in lieu of uniform values on the (0, 1) interval. This provides a significant speed increase. The second, and more important change is to extend the period of the sequence generated by the IBM routine which is 2^{13} or 8192. This is achieved by varying the multiplier in the power residue method. For a theoretical explanation see "Random Number Generation and Testing," IBM Manual C20-8011.

A listing of RECSM is shown below:

```
      SUBROUTINE RECSM (IX,IY,IMULT)
      IY=IX*IMULT
      IF (IY) 5,6,6
    5 IY=IY+32767+1
    6 RETURN
      END
```

V-5 IBM 1800 COMPATIBILITY

It is our understanding that the main line program (SEQDE) as well as the subroutines may be used directly on the IBM 1800, with the possible requirement that they be re-compiled. Card decks will be furnished to GSFC under separate enclosure.