BELLCOMM, INC.
1100 SEVENTEENTH STREET, N.W. WASHINGTON, D.C. 20036

COVER SHEET FOR TECHNICAL MEMORANDUM

TITLE- Analogue to Digital Conversion Requirements in Partial Digital Detection of Coded Phase Coherent Transmissions

TM- 68-2034-3

DATE- March 15, 1968

FILING CASE NO(S)- 900

AUTHOR(S)- L. Schuchman

FILING SUBJECT(S)- (ASSIGNED BY AUTHOR(S)- Coded Phase-Coherent Communications - Modulation Techniques

ABSTRACT

The performance of coded phase coherent systems when the receiver is designed so that nearly all of the detector can be realized by a digital computer has been studied. In particular the analogue to digital operation has been investigated with the hope of simplifying this operation so that longer phase coded message alphabets (which achieve better system performance) could be realized.

Large alphabet phase coherent codes are of interest in space communication applications because they trade bandwidth for improved system performance. Thus we find coded phase coherent systems being used for the Pioneer-Mariner 69 Programs and in addition they appear applicable to future manned space flight programs.

The results obtained demonstrate that simple A/D converters can be used in nonoptimum detectors to achieve near optimum performance.

SEE REVERSE SIDE FOR DISTRIBUTION LIST

BELLCOMM, INC.
1100 Seventeenth Street, N.W. Washington, D.C. 20036

SUBJECT: Analogue to Digital Conversion Requirements in Partial Digital Detection of Coded Phase Coherent Transmissions - Case 900

DATE: March 15, 1968

FROM: L. Schuchman

TM-68-2034-3

TECHNICAL MEMORANDUM

The advent of the space age has lead to the development of new communication techniques since space communication links have different constraints than are usually found in more conventional communication systems. Thus where space systems are normally power limited while having relatively large bandwidths the reverse is true for Earth to Earth linked systems. Therefore coding techniques which improve performance at the expense of bandwidth are receiving a good deal of attention. One such group of codes is M'ary orthogonal and bi-orthogonal binary codes transmitted by bi-phase modulating a carrier. This is commonly referred to as coded phase-coherent transmissions and has been discussed in great detail by Viterbi.[1,2,3*]

M'ary orthogonal codes are codes which have the following property. Let $\{S_i(t): i=1,2,\ldots,M\}$ be a set of M orthogonal code words of length MT. Then

$$\frac{1}{T}\int_0^{MT} S_i(t)S_j(t)dt = \begin{cases} M & i=j \\ 0 & i\neq j \end{cases} \tag{1}$$

There are several ways of generating orthogonal codes. We restrict our interest to those that are constructed from binary sequences. In particular we concern ourselves with those codes that can be generated from Rademacher-Walsh functions, as illustrated in Figure 1 for M=2 and M=4.** It can be seen that these

---

*This is to be distinguished from MPSK which trades performance for bandwidth conservation.

**Given that our code set for M=V is $S_1(t)\ \ldots\ S_V(t)$, all of which are Rademacher-Walsh functions, then for M=2V our code set is $S_1(t) + S_1(t-VT_{2V})$, $S_1(t) + \overline{S}_1(t-VT_{2V})$, $S_2(t) + S_2(t-VT_{2V})\ \ldots\ S_V(t) + \overline{S}_V(t-VT_{2V})$ where T_M is the chip width of any symbol in the M'ary code set and $\overline{S}_M(t)$ is the complementary signal $S_M(t)$.

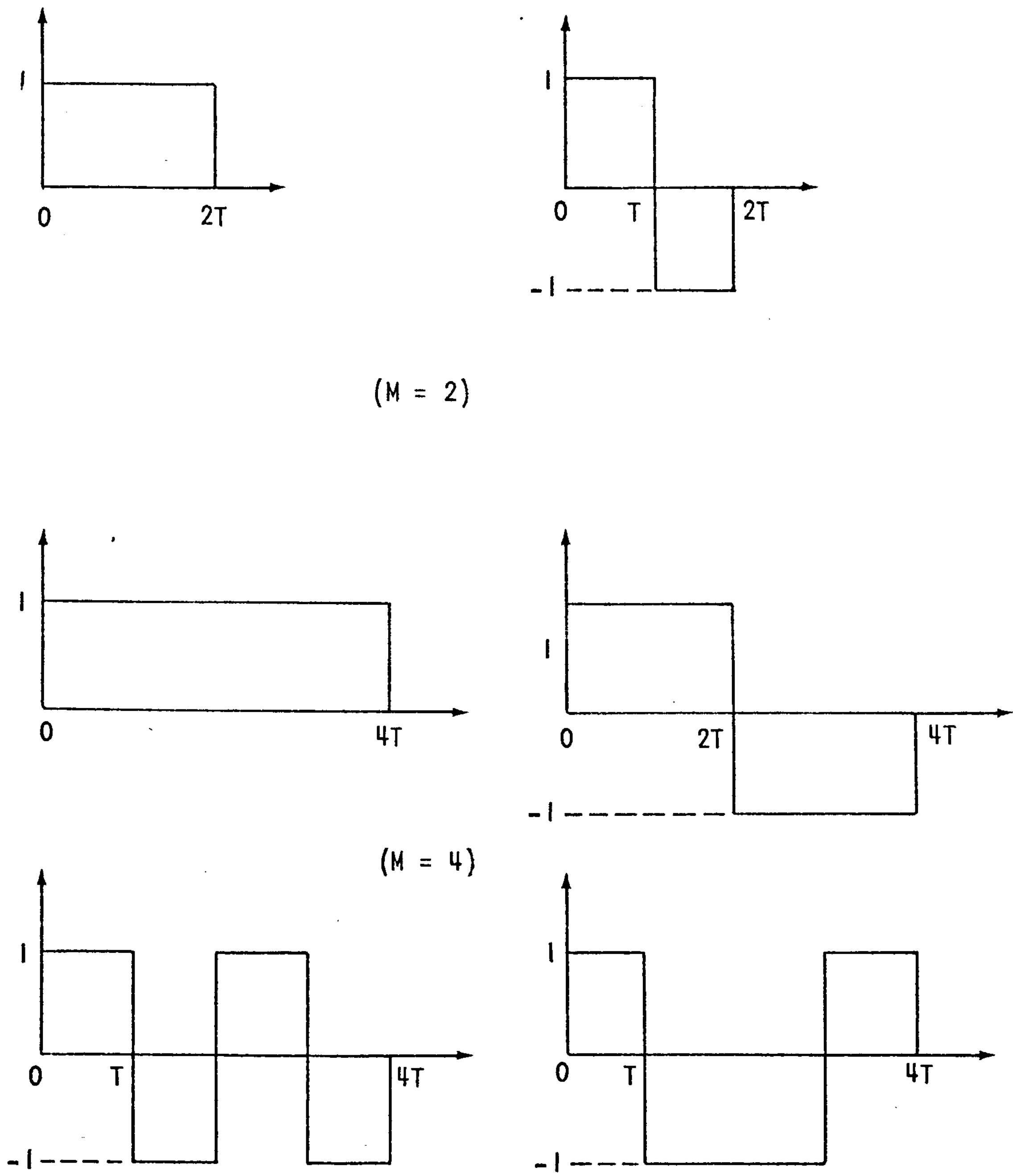

FIGURE 1 - M'ARY ORTHOGONAL BINARY SEQUENTIAL CODES (RADEMACHER-WALSH FUNCTIONS) FOR M = 2 AND M = 4

illustrated codes satisfy the orthogonality condition given in equation (1).

The basic communication system model for the transmission of orthogonaly coded phase-coherent signals is given in Figure 2. The transmitted signal $S_j(t)$ (one of M possible signals), is distorted by additive Gaussian noise, n(t) while in transmission. The received signal is then passed through a carrier detector which coherently detects each of the binary digits of the transmitted message. The resultant video signal z(t) (M binary digits long) then represents the best estimate of the transmitted message $S_j(t)$. z(t) is passed through a message detector, which is designed optimally from a statistical decision theory point of view with the result that the signal z(t) is correlated with the M $S_k(t)$ possible transmitted signals to form the set of M energy measures $\{U_k\}$. If the energy measure U_j is not greater than all other U_k $(k \neq j)$ an error occurs.

The advantage of transmitting orthogonal signals by a coded bi-phase modulated M'ary sequence over say an M'ary extension of PSK is that the receiver can be realized with only one radio frequency carrier matched filter. For MFSK M such filters are required.* Such a requirement is difficult to realize physically for reasonably large values of M (M>32).[3] In addition coded phase-coherent transmissions allow one to code biorthogonally which performs at least as well in half the bandwidth as orthogonally coded systems. It has been shown[2] that when a receiver such as described by Figure 2 is realized for orthogonally coded (or biorthogonally coded) systems the probability of bit error P_e, for a fixed transmitter power and data rate, decreases as M increases while the bandwidth increases by a factor $\frac{M}{\log_2 M}$ for orthogonal systems and $\frac{1}{2}\,\frac{M}{\log_2 M}$ for biorthogonal systems.[2] The performance of M'ary orthogonal systems is illustrated in Figure 3.

In looking at the phase-coherent receiver as described in Figure 2 it is to be noted that the M message correlations are performed at baseband and therefore the use of a digital processor (computer) to realize the Message Maximum likelihood detector becomes a possibility if the distortion due to analogue to digital conversion can be kept within tolerable limits. This paper is concerned with the design of such an analogue to digital converter.

---

*The advantage of MFSK is that it can be realized non-coherently so that it can be used in channels where coherence cannot be maintained.

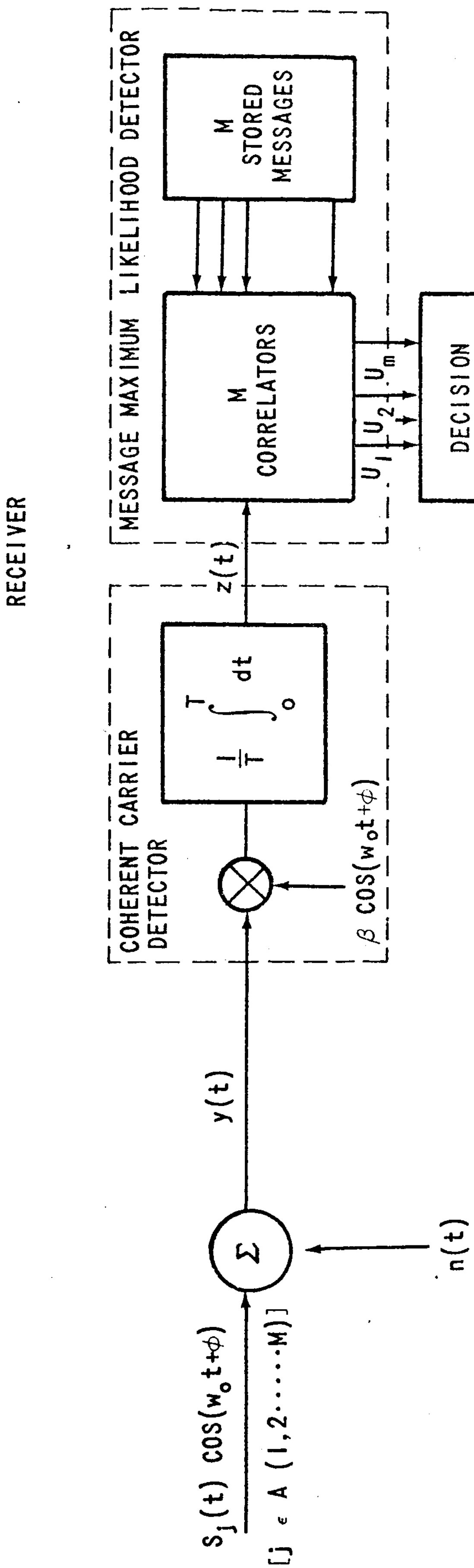

FIGURE 2 - ORTHOGONAL CODED PHASE COHERENT COMMUNICATION SYSTEMS

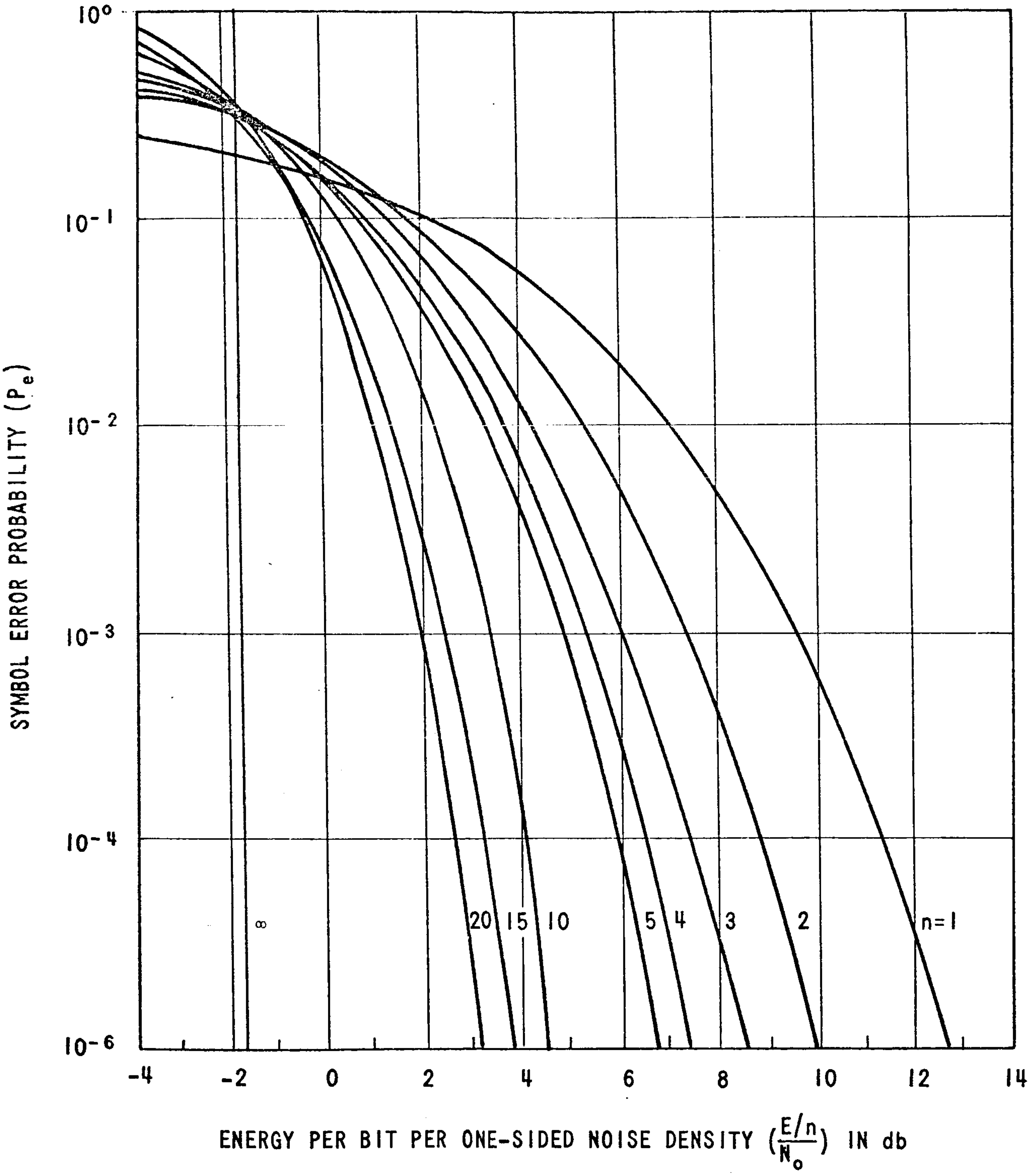

FIGURE 3 - PROBABILITY OF ERROR FOR COHERENT DETECTION OF ORTHOGONAL M'ARY CODES

The simplest analogue to digital converter or quantizer is one which converts all positive analogue inputs into one positive value while all negative inputs go into one negative value, as is illustrated in Figure 4.

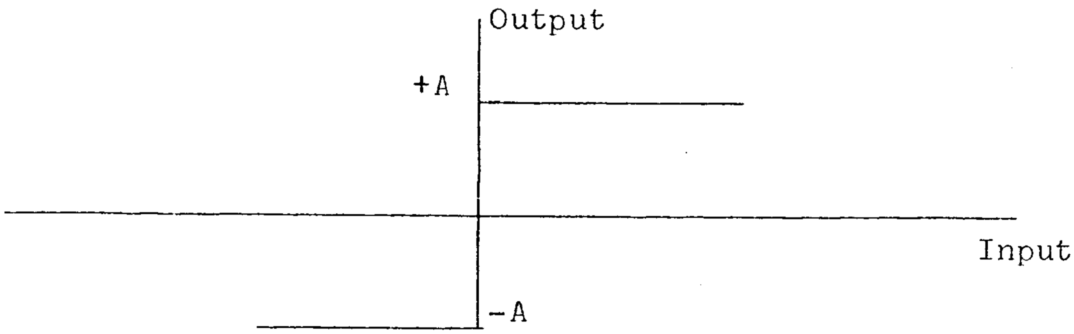

Figure 4 Two Level Quantizer

At the other extreme a quantizer which in the limit has an infinite number of equally spaced levels essentially quantizes an analogue signal into an equivalent analogue signal with no loss of information so that detection can be made optimum and a performance achieved as described by Figure 3.* In this paper we determine the performance when the A/D converter is but 2 levels and then show that with just a 4 level nonuniform quantizer a performance can be achieved which approaches within 1 db of that obtained with an infinite level quantizer.**

---

*Since the additive gaussian noise in the channel results in an analogue output from the binary digit carrier matched filter the A/D converter needs to be infinite to achieve optimum performance.

**This is demonstrated for M=4 and 8.

Performance of an Orthogonally Coded Phase Coherent System Using a 2 Level A/D Converter

In this section we assume the detector is as illustrated in Figure 5 below.

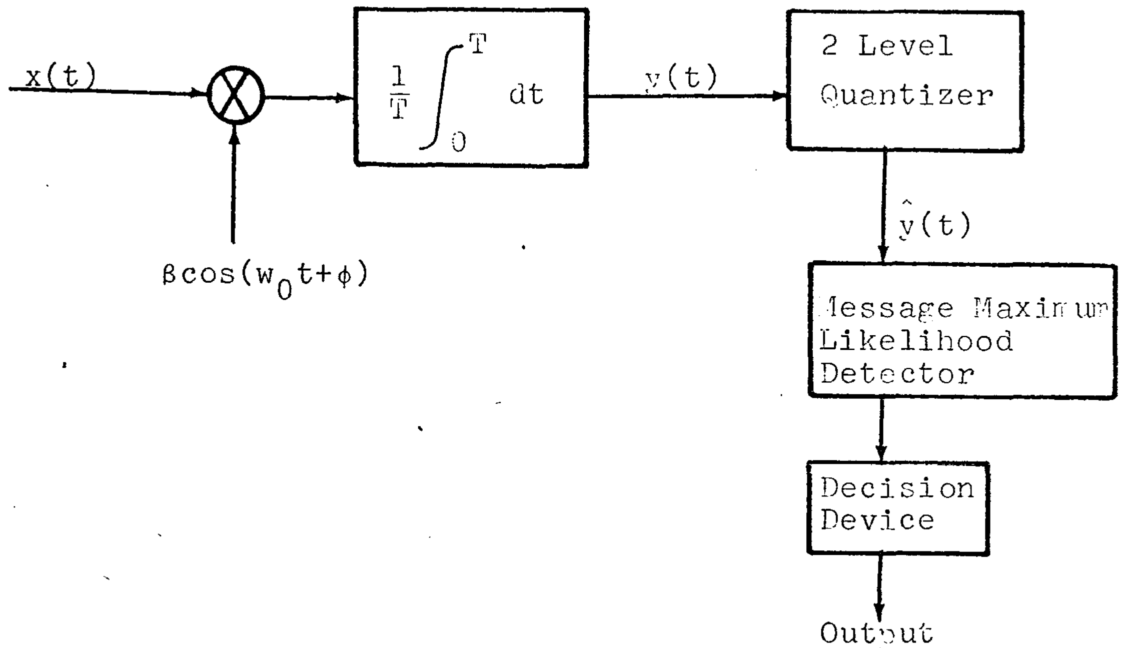

Figure 5 Orthogonally Coded Phase Coherent Detection Using A 2 Level S/D Converter

The advantage of using a 2 level A/D converter is that it reduces the storage requirements for the digital computer used as the Message detector since only one bit has to be stored for each binary digit of the M binary digit word.

Let us assume that when errors occur in the transmission data stream they occur independently. That is

P[Error will occur to the i^{th} binary digit of the u^{th} message word/error has or has not occurred to the j^{th} binary digit of the V^{th} message word, ($j\neq i$, when $u=V$)] = P[Error will occur to the i^{th} binary digit of the V^{th} message].

We next ask the question how many binary digits of a message word need be detected in error before an error in the message word is possible. From equation (1) we know that when all binary digits are received correctly for the transmission of the j^{th} word then the j^{th} correlation output is M while all the other correlation outputs are zero. Thus when e errors are made the j^{th} correlation output must be M-2e while a possibility exists that at least one of the other correlation outputs will be as high as 2e. Therefore, if

$$M-2e \leq 2e$$

or

$$e \geq \frac{M}{4}$$

there is a possibility that a word will be received in error.

It is reasonable to assume that the binary digit error rate is less than 1/2 so that the most probable event that will lead to errors occurs when e equals M/4.

This means that there is at least one correlator output i whose output value is equal to the j^{th} correlation output where S_j is the transmitted message word and $i \neq j$. To find the contribution to the probability of word error when e = M/4 we proceed in the following manner. All error vectors for a given transmitted word say S_j in which M/4 error occur are generated. There are $\frac{M!}{(\frac{M}{4})!(\frac{3}{4}M)!} = \binom{M}{\frac{M}{4}}$ such vectors. For each error vector $z_{\gamma j}$ we compute the inner product $\langle S_i z_{\gamma j} \rangle$ for all i. If we assume that when this error vector is generated the j^{th} and L other correlation outputs have the same $\frac{M}{4}$ output value than with a completely random decision rule (in case of such ties) we will make an error $\frac{L-1}{L}$ of the time. Therefore the probability that an error is made and e = M/4 given that the j^{th} message is transmitted, and $\frac{M}{4}$ binary digits are in error in such a manner as to produce error vector $z_{\gamma j}$($P(E|\ e = \frac{M}{4}, S_j, z_{\gamma j})$) is

$$P(E|\ e = \tfrac{M}{4},\ S_j,\ z_{\gamma j}) = \frac{k_{\gamma j}}{M}\ p^{\frac{M}{4}}(1-p)^{\frac{3}{4}M}$$

where

p = probability of a binary digit error.

$$\kappa_{\gamma j} = \frac{\text{Number of correlator outputs with maximum inner product minus one}}{\text{Number of correlator outputs with maximum inner product}}$$

The probability of word error when e = M/4, $P(E|\ e = \frac{M}{4})$ is thus

$$P(E|\ e = \tfrac{M}{4}) = K\ p^{\frac{M}{4}}(1-p)^{\frac{3}{4}M} \qquad (2)$$

where

$$K = \frac{1}{M} \sum_{j=1}^{M} \sum_{\gamma=1}^{\binom{M}{\frac{M}{4}}} k_{\gamma j} \qquad (3)$$

Therefore we can write a lower bound to the word error probability ($P_L(E)$) in the following manner

$$P_L(E) = Kp^{\frac{1}{4}M}(1-p)^{\frac{3}{4}M} \qquad M \geq 4 \qquad (4)^*$$

To obtain an upper bound ($P_U(E)$) we assume that for e > M/4 an error is a certainty. Thus

---

*When M = 2 equations (4) and (5) do not apply. However the same procedure used to find $P_L(E)$ for $M \geq 4$ can be used to obtain for M = 2 P(E) = p.

$$P_U(E) = P_L(E) + \sum_{i=\frac{M}{4}+1}^{M} \binom{M}{i} p^i (1-p)^{M-i} \quad M \geq 4 \qquad (5)^*$$

To illustrate the procedures used to calculate K the following example will prove helpful. The set of orthogonal S_j message words or vectors is

S_1	1	1	1	1
S_2	1	1	-1	-1
S_3	1	-1	1	-1
S_4	1	-1	-1	1

For each S_j there are $\binom{4}{1}$ error vectors with e = 1. The four z_{1j} vectors are

z_{11}	1	1	1	-1
z_{12}	1	1	-1	1
z_{13}	1	-1	1	1
z_{14}	-1	1	1	1

Therefore

$$\langle S_1 \; z_{11} \rangle = 2$$
$$\langle S_2 \; z_{11} \rangle = 2$$
$$\langle S_3 \; z_{11} \rangle = 2$$
$$\langle S_4 \; z_{11} \rangle = -2$$

---

*See footnote on the previous page.

With the result that

$$K_{11} = \frac{2}{3}$$

To compute K and the upper and lower bounds a computer program was generated. The program is described in Appendix A.* The results obtained are given in Table I. As can be seen from the table the bounds are extremely close. Unfortunately the process of finding k in the manner described for high values of M becomes unmanageable. As an example with M = 32, over ten million $\binom{32}{8}$ error vectors must be generated. Before we evaluate the results presented in Table I in detail we derive a second set of bounds which allows us to obtain information on the performance of 32'ary and 64'ary systems.

A Second Set of Bounds

Assume we transmit S_j and $\frac{M}{4}$ bit errors are made. Now we know that S_i $(i \neq j)$ is a vector that has $\frac{M}{2}$ elements in common with S_j (due to orthogonally condition) thus if $\langle S_i \; z_{\gamma j} \rangle$ $(i \neq j)$ is to equal $\frac{M}{2}$ all $\frac{M}{4}$ binary digits in error must have been made in the $\frac{M}{2}$ elements which are identical in both S_i and S_j. Thus for a given i and j the number of possible γ's for which $\langle S_i \; z_{\gamma j} \rangle = \frac{M}{2}$ is simply the combinatorial $\binom{\frac{M}{2}}{\frac{M}{4}}$. Next we note that for each $z_{\gamma j}$ there may be more than one i for which $\langle S_i \; z_{\gamma j} \rangle = \frac{M}{2}$ is true. However if we assume that for each such γ only one i can satisfy $\langle S_i \; z_{\gamma j} \rangle = \frac{M}{2}$ then at most $(M-1)\binom{\frac{M}{2}}{\frac{M}{4}}$ error patterns ($z_{\gamma j}$'s) lead to a possible error. If we apply our random decision rule in case of ties then 1/2 of the time such an error pattern leads to errors. Therefore an upper bound to the probability of error given M/4 errors were made is

---

*It was noted that for M = 4 and M = 8 $\sum_{\gamma=1}^{M} k_{\gamma j}$ was independent of j. Therefore the program has been written for j = 1 and $K = \sum_{\gamma=1}^{M} k_{\gamma 1}$.

Table I Bounds on Performance With 2 Level A/D Converter (M = 2, 4, 8, 16)

M = 2

$$P_L(E) = kp(1-p)$$

$$P_U(E) = P_L(E) + p^2$$

M = 4, 8, 16

$$P_L(E) = kp^{\frac{M}{4}}(1-p)^{(M-\frac{M}{4})}$$

$$P_U(E) = P_L(E) + \sum_{J=\frac{M}{4}+1}^{M} \binom{M}{J} p^J (1-p)^{(M-J)}$$

Probability of binary digit error p		M			
		2	4	8	16
.1 (10^{-1})	$P_L(E)$	9.0×10^{-2}	1.5×10^{-1}	7.4×10^{-2}	1.4×10^{-2}
	$P_U(E)$	1.0×10^{-1}	2.0×10^{-1}	1.1×10^{-1}	3.1×10^{-2}
.01 (10^{-2})	$P_L(E)$	1.0×10^{-2}	1.9×10^{-2}	1.3×10^{-3}	4.3×10^{-6}
	$P_U(E)$	1.0×10^{-2}	2.0×10^{-2}	1.4×10^{-3}	4.7×10^{-6}
.001 (10^{-3})	$P_L(E)$	1.0×10^{-3}	2.0×10^{-3}	1.4×10^{-5}	4.8×10^{-10}
	$P_U(E)$	1.0×10^{-3}	2.0×10^{-2}	1.4×10^{-5}	4.9×10^{-10}
.0001 (10^{-4})	$P_L(E)$	1.0×10^{-4}	2.0×10^{-4}	1.4×10^{-7}	4.9×10^{-14}
	$P_U(E)$	1.0×10^{-4}	2.0×10^{-3}	1.4×10^{-7}	4.9×10^{-14}
.00001 (10^{-5})	$P_L(E)$	1.0×10^{-5}	2.0×10^{-5}	1.4×10^{-9}	4.9×10^{-18}
	$P_U(E)$	1.0×10^{-5}	2.0×10^{-4}	1.4×10^{-9}	4.9×10^{-18}
.000001 (10^{-6})	$P_L(E)$	1.0×10^{-6}	2.0×10^{-6}	1.4×10^{-11}	4.9×10^{-22}
	$P_U(E)$	1.0×10^{-6}	2.0×10^{-5}	1.4×10^{-11}	4.9×10^{-22}
K	1	1	2	14	490

$$P_U(E|\ e=\tfrac{M}{4}) = \tfrac{1}{2}(M-1)\binom{\frac{M}{2}}{\frac{M}{4}} p^{\frac{M}{4}}(1-p)^{\frac{3}{4}M} \qquad (6)^*$$

In similar fashion we find that $P_U(E|\ e=\frac{M}{4}+1)$ is given by

$$P_U(E|\ e=\tfrac{M}{4}+1) = (M-1)\left[\binom{\frac{M}{2}}{\frac{M}{4}+1} + \tfrac{1}{2}\binom{\frac{M}{2}}{1}\binom{\frac{M}{2}}{\frac{M}{4}}\right] p^{\frac{M}{4}+1}(1-p)^{\frac{3}{4}M-1} \qquad (7)$$

while

$$P_U(E|\ e=\tfrac{M}{4}+2) = (M-1)\left[\binom{\frac{M}{2}}{\frac{M}{4}+2} + \binom{\frac{M}{2}}{\frac{M}{4}+1}\binom{\frac{M}{2}}{1} + \tfrac{1}{2}\binom{\frac{M}{2}}{\frac{M}{4}}\binom{\frac{M}{2}}{2}\right] p^{\frac{M}{4}+2}(1-p)^{\frac{3}{4}M-2} \qquad (8)$$

Thus if we add equations (6), (7) and assume that when the number of errors is $\frac{M}{4}+2$ or greater an upper bound to the symbol error probability is given by

$$P_U(E) = P\ (E|\ e=\tfrac{M}{4}) + P\ (E|\ e=\tfrac{M}{4}+1) + \sum_{\alpha=\frac{M}{4}+2}^{M} \binom{M}{\alpha} P^{\alpha}(1-p)^{M-2} \qquad (9)$$

while if we include equation (8) we have

---

*The assumption that for each γ only one i can satisfy $\langle s_i z_{\gamma j}\rangle = \frac{M}{2}$ leads to the upper bound represented in equation (6) is proven on pages 14 and 15.

$$P_U(E) = P_U(E|\ e=\tfrac{M}{4}) + P_U(E|\ e=\tfrac{M}{4}+1) + P_U(E|\ e=\tfrac{M}{4}+2)$$

$$+ \sum_{\alpha=\frac{M}{4}+3}^{M} \binom{M}{\alpha} p^{\alpha}(1-p)^{M-\alpha} \qquad (10)$$

As will be seen shortly, equation (9) is used to compute the upper bounds for M = 32 while equation (10) provides results for M = 64.

To derive the lower bound $P_L(E)$ we proceed in the following manner. Assume $\frac{M}{4}$ binary digit errors in the M'ary transmitted word S_j. Then for some $z_{\gamma j}$ and some S_i we have

$$\langle S_i\ z_{\gamma j}\rangle = \langle S_j\ z_{\gamma j}\rangle = \frac{M}{2} \qquad (11)$$

But there may be more than one i which satisfies equation (11). Let n_γ be the number of i ($i \neq j$) for which equation (11) is true. Then with our random detection rule for such cases we have that $\frac{n_\gamma}{n_\gamma+1}$ of time whenever equation (11) is true a word error is made. The number of unique γ for a given i for which (11) is true has been shown to be $\binom{\frac{M}{2}}{\frac{M}{4}}$ while the total number of γ (#) for which equation (11) is true is bounded by

$$\binom{\frac{M}{2}}{\frac{M}{4}} \frac{M-1}{n_\gamma(\text{maximum})} \leq \# \leq \binom{\frac{M}{2}}{\frac{M}{4}} \frac{M-1}{n_\gamma(\text{minimum})}$$

Therefore the probability of word error give $M/4$ binary digits were detected in error $P(E|\ e=M/4)$ is bounded by

$$\binom{\frac{M}{2}}{\frac{M}{4}} p^{\frac{M}{4}} (1-p)^{\frac{3}{4}M} \frac{M-1}{1+n_\gamma(\min)} \geq P(E|\ e=M/4) \geq \binom{\frac{M}{2}}{\frac{M}{4}} p^{\frac{M}{4}} (1-p)^{\frac{3}{4}M} \frac{M-1}{1+n_\gamma(\max)} \qquad (12)$$

As noted earlier $n_\gamma(\min)$ is equal to one. To determine $n_\gamma(\max)$ we proceed in the following manner. Let S_j be the transmitted symbol. Let the binary digit positions of a given message word $S_\alpha(\alpha \neq j)$ in which the binary digits agree be called the a positions while those in which there is no agreement are called the b positions. We now ask the following question; what is the maximum number of possible symbols (S_α $\alpha=1, 2, \ldots, M$) that can have $\frac{M}{4}$ a positions in common? To answer this and with no loss of generality we refer to Figure 6.

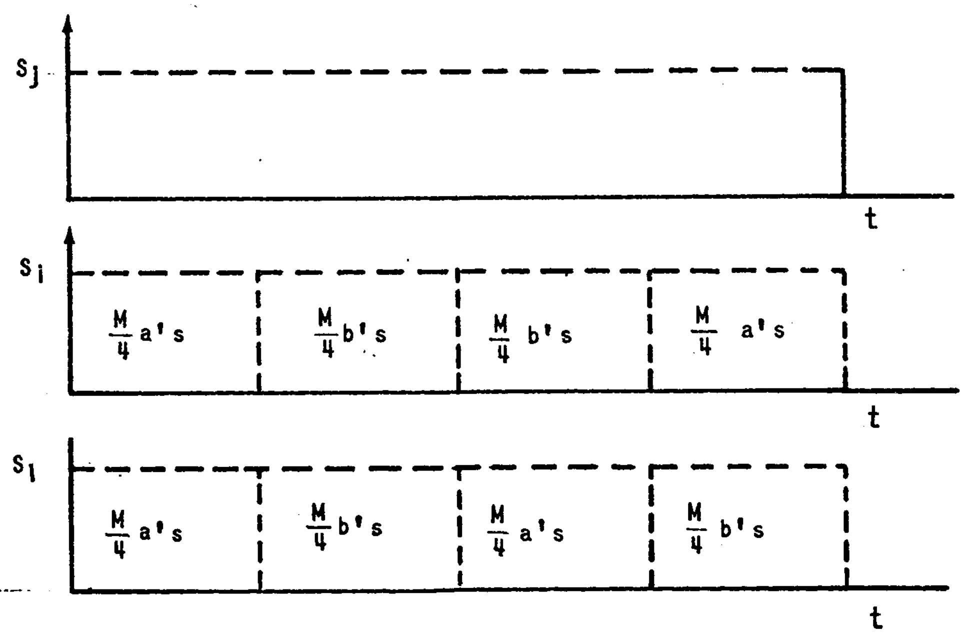

Figure 6 3 Message Words With $\frac{M}{4}$ Binary Digits In Common

We see from Figure 6 that

$$<S_j \; S_i> \; = \; <S_j \; S_\ell> \; = \; 0$$

and since each element of each message word can take on only one of two possible values

$$<S_i \; S_\ell> \; = \; 0$$

and the orthogonality condition for the three words has been met.

We next consider the structure of a fourth message word S_h given in Figure 7 which has $\frac{M}{4}$ elements in common with the previous three message words.

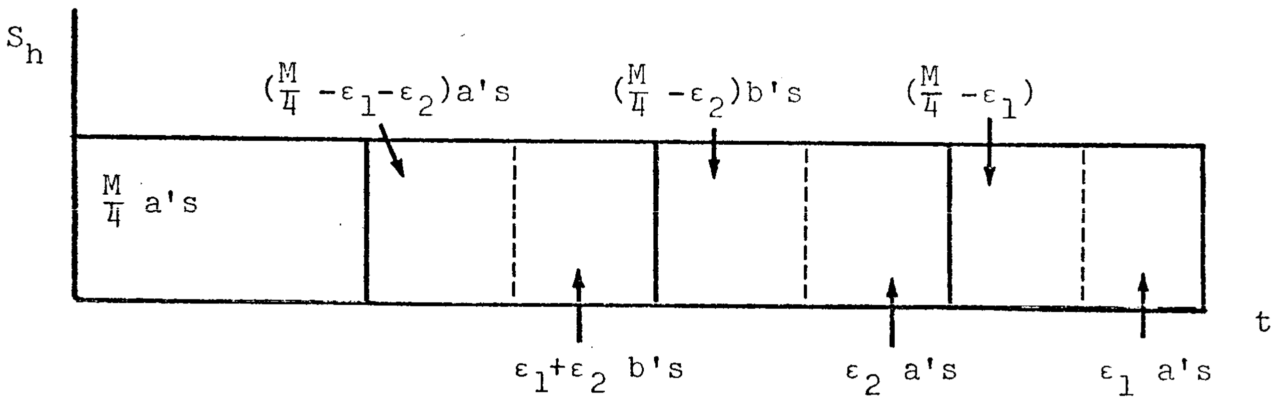

Figure 7 A Fourth Possible Message Which Has $\frac{M}{4}$ Binary Digits In Common With The Message Words S_j, S_i, and S_ℓ

We readily see that

$$<S_j \; S_h> \; = \; 0$$

but

$$\langle S_i \; S_h \rangle = \frac{M}{4} - \left(\frac{M}{4} - \varepsilon_1 - \varepsilon_2\right) + (\varepsilon_1 + \varepsilon_2) + \frac{M}{4} - \varepsilon_2 - \varepsilon_2 \quad - \left(\frac{M}{4} - \varepsilon_1\right) + \varepsilon_1$$

which reduces to

$$\langle S_i \; S_h \rangle = 4\varepsilon_1$$

or $\varepsilon_1 = 0$ for orthogonality and

$$\langle S_\ell \; S_h \rangle = 0 \quad \text{iff } \varepsilon_2 = 0$$

$\therefore$ The only message word S_h which can possibly satisfy the orthogonality condition and simultaneously the $\frac{M}{4}$ common element condition is given in Figure 8. Thus we have that at most there are 4 message words that can have $\frac{M}{4}$ of these respective binary digit elements in common ($n_{max} = 4$).

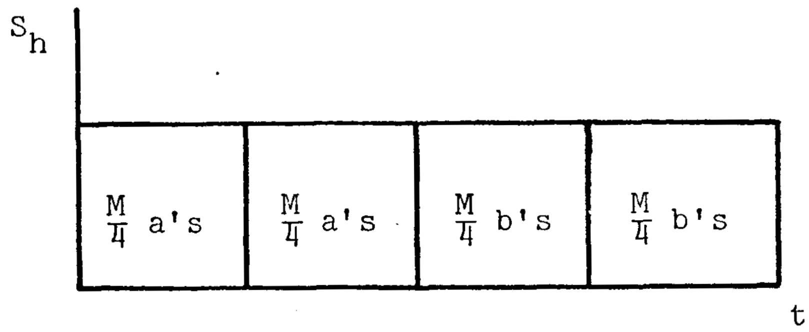

Figure 8 The Fourth Possible Message Word

The lower bound to the message error probability is then given by

$$P_L(E) = \frac{M-1}{4} p^{\frac{M}{4}} (1-p)^{\frac{3}{4}M} \tag{14}$$

Equations 9 and 14 were programmed. The program is described in Appendix B. The results of this program are presented in Table II. In comparing Table II with Table I we see that the bounds are not as close as those obtained with the program described in Appendix A. Nevertheless they allow us to obtain some useful information as to the performance of a 32'ary system. Equation 10 was also programmed (described in Appendix B) with the results presented in Table III. Comparing Table III with Table II we see that the upper bound has been significantly improved. However the upper and lower bounds for this case are still very crude.

Table III Upper Bounded Performance Of A 64'ary System Using A 2 Level A/D Converter

p	$P_U(E)$
.5	9.997×10^{-1}
.25	2.44×10^{-1}
.1	1.395×10^{-5}
.075	2.12×10^{-7}
.05	3.893×15^{10}
.025	3.525×10^{-15}
.01	5.813×10^{-22}

Table II Bounds on Performance of a PSK Coded System Using a 2 Level A/D Converter (M = 16, 32, 64)

$$P_L(E) = \frac{M-1}{4}\binom{\frac{M}{2}}{\frac{M}{4}} p^{\frac{M}{4}} (1-p)^{(M-\frac{M}{4})}$$

$$P_U(E) = \frac{M-1}{2}\binom{\frac{M}{2}}{\frac{M}{4}} p^{\frac{M}{4}} (1-p)^{(M-\frac{M}{4})} + \left[\binom{\frac{M}{2}}{\frac{M}{4}+1} + \frac{M}{4}\binom{\frac{M}{2}}{\frac{M}{4}}\right](M-1)p^{(\frac{M}{4}+1)}(1-p)^{(M-(\frac{M}{4}+1))}$$

$$+ \sum_{j=\frac{M}{4}+2}^{M} \binom{M}{J} p^J (1-p)^{M-J}$$

p	M	16	32	64
.5	$P_L(E)$	$4.0\text{x}10^{-3}$	$2.3\text{x}10^{-5}$	$5.1\text{x}10^{-10}$
	$P_U(E)$	$9.7\text{x}10^{-1}$	$9.9\text{x}10^{-1}$	$9.999\text{x}10^{-1}$
.25	$P_L(E)$	$3.2\text{x}10^{-2}$	$1.5\text{x}10^{-3}$	$2.2\text{x}10^{-6}$
	$P_U(E)$	$4.3\text{x}10^{-1}$	$2.8\text{x}10^{-1}$	$3.3\text{x}10^{-1}$
.1	$P_L(E)$	$7.4\text{x}10^{-3}$	$8.0\text{x}10^{-5}$	$6.0\text{x}10^{-1}$
	$P_U(E)$	$3.1\text{x}10^{-2}$	$1.3\text{x}10^{-3}$	$3.8\text{x}10^{-5}$
.075	$P_L(E)$	$3.3\text{x}10^{-3}$	$1.5\text{x}10^{-5}$	$2.2\text{x}10^{-10}$
	$P_U(E)$	$1.1\text{x}10^{-2}$	$1.5\text{x}10^{-5}$	$7.0\text{x}10^{-7}$
.05	$P_L(E)$	$8.9\text{x}10^{-4}$	$1.1\text{x}10^{-6}$	$1.2\text{x}10^{-12}$
	$P_U(E)$	$2.6\text{x}10^{-3}$	$6.5\text{x}10^{-6}$	$1.5\text{x}10^{-9}$
.025	$P_L(E)$	$7.6\text{x}10^{-5}$	$8.3\text{x}10^{-9}$	$6.5\text{x}10^{-17}$
	$P_U(E)$	$1.8\text{x}10^{-4}$	$2.7\text{x}10^{-8}$	$1.8\text{x}10^{-14}$
.01	$P_L(E)$	$7.6\text{x}10^{-5}$	$8.3\text{x}10^{-12}$	$5.8\text{x}10^{-23}$
	$P_U(E)$	$5.0\text{x}10^{-6}$	$1.9\text{x}10^{-11}$	$2.5\text{x}10^{-21}$

To obtain a meaningful comparison of the data presented in Tables I through III the values of p are converted to equivalent values of the ratio of signal energy per bit to noise spectral density. Since each binary digit is transmitted as a bi-phase modulation the relationship of p to the ratio of binary digit signal energy to noise spectral density is given by the performance curves of a bi-phase modulated optimal detection system and is presented in Figure 9. Since each message contains M binary digits and carries $\log_2 M$ bits of information the conversion from binary digit energy to bit energy is given by

$$\frac{ME \text{ (binary digit)}}{\log_2 M} = E \text{ (bit)}$$

where E (bit) is the equivalent received energy per information bit

E (binary digit) is the received energy per binary digit of the transmitted message word.

In Table IV the binary digit error rate is given as a function of the bit energy to noise spectral density (E/N_0).

Table IV Message Binary Digit Error Rate (p) As A Function Of The Received Information Bit Energy To Noise Spectral Density (E/N_0 bit)

p	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}
$\frac{E}{N_0}$(binary digit) (in DB)	-1	4.4	7	8.5	9.2

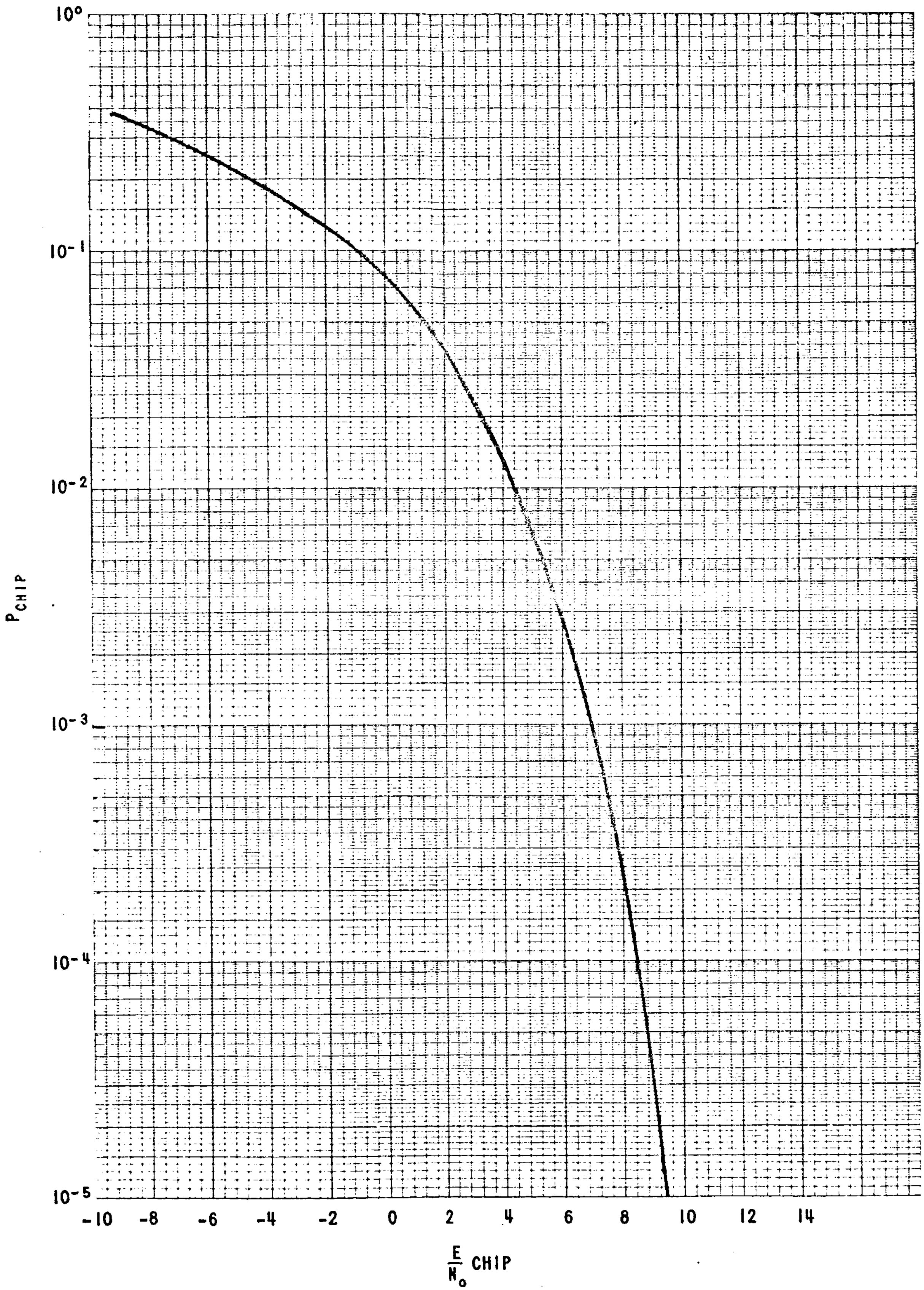

FIGURE 9 - CHIP ERROR PROBABILITY AS A FUNCTION OF THE RATIO OF THE CHIP ENERGY TO NOISE SPECTRAL DENSITY (COHERENT PSK ASSUMED)

Table IV (Continued)

M	E/N_0 (binary digit)	+	$10 \log_{10} \frac{M}{\log_2 M}$
2	"	+	3.0
4	"	+	3.0
8	"	+	4.2
16	"	+	6.0
32	"	+	8.1
64	"	+	10.3

Using Tables I through IV the upper and lower bounds of the M'ary probability of symbol error is plotted in Figure 10 as a function of the received bit energy to noise spectral density and compared with ordinary PSK. It is to be noted that for $M > 8$ and at least up to $M = 64$ performance improves as M increases. Although the bounds for $M = 32$ and $M = 64$ are very crude, they do show this phenomena occurring and in addition present results which even on a symbol error basis are an improvement over ordinary PSK.* That is if we assume that the $\log_2 M$ bits of each M'ary word were not chosen in sequence but represented quasi-random data then the following transformation from symbol to bit error rates can be used

$$P_{bit}(E) = \frac{M}{2(M-1)} P_{symbol}(E)$$

*It is obvious that if there were no improvement over ordinary PSK there would be no reason to use this more complex coding scheme which utilizes so much more bandwidth.

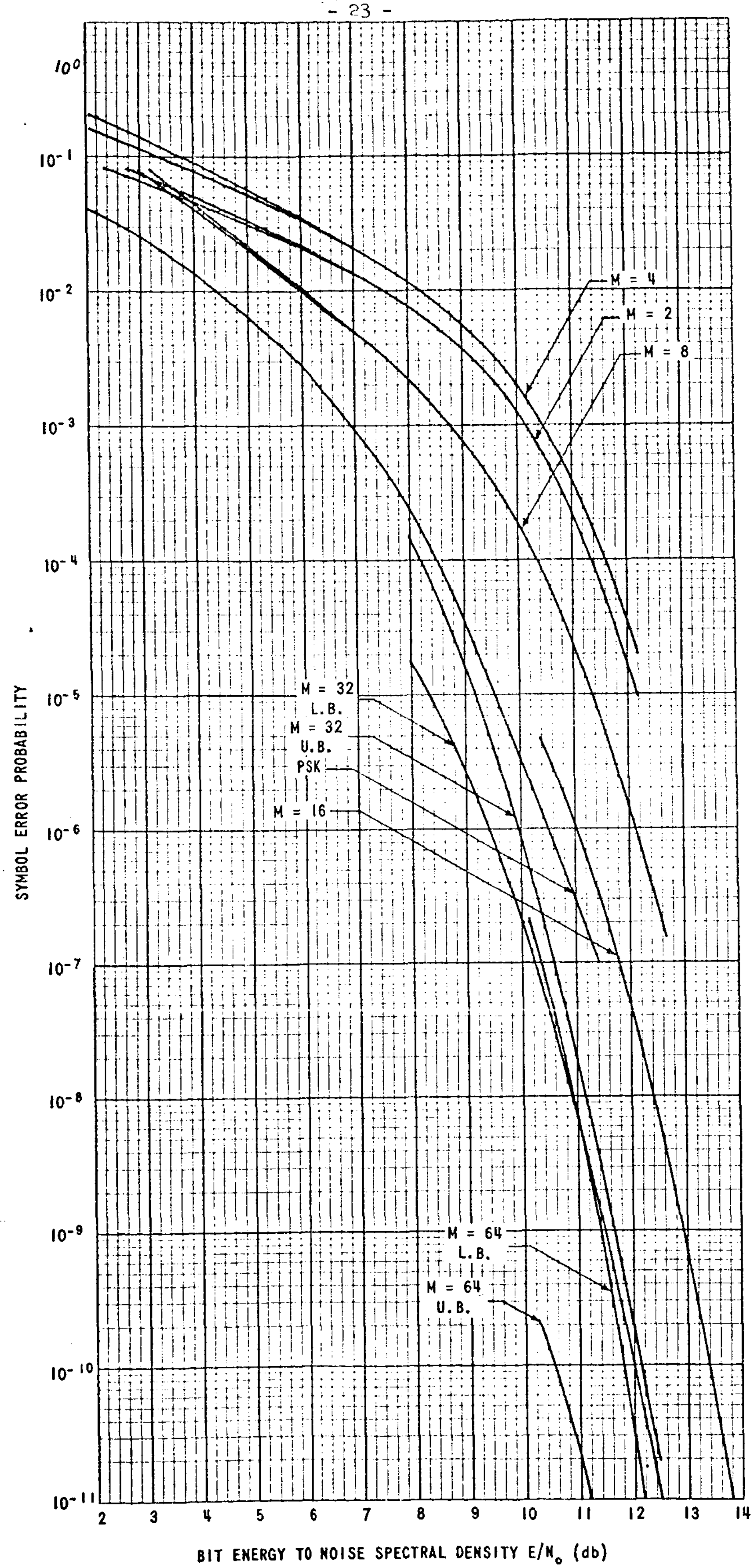

FIGURE 10 - COMPARISON OF CODED PHASE COHERENT SYSTEMS USING A 2 LEVEL A/D CONVERTER

and the improvement over PSK is seen to be even greater. Thus we have come to the interesting fact that even though the quantizer eliminated all but sign information improvements in performance is achieved as the coding is made more complex. It would be interesting to determine the performance of this receiver when M goes to infinity to see if it achieves the same performance as the optimum receiver.

Performance Of An Orthogonally Coded Phase Coherent Detector Using A 4 Level A/D Converter

In this section we assume the detector is as illustrated in Figure 5 except that the 2 level quantizer is replaced by a four level quantizer. The four level quantizer is described in Figure 11.

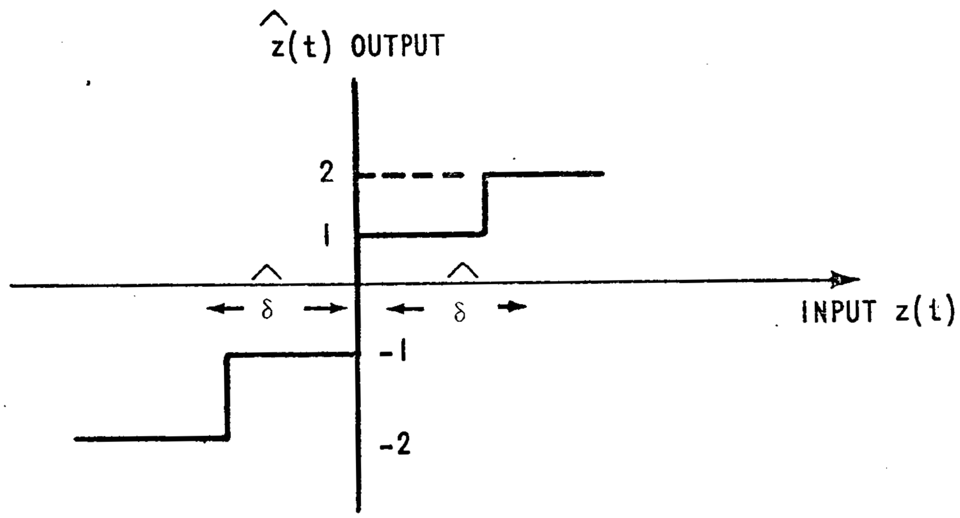

Figure 11 A 4 Level A/D Converter

As can be seen in the illustration, a positive input into the quantizer is converted to one of two possible values. The quantizer is assumed to have odd symmetry so a negative signal of equal amplitude experiences the same transformation except for the sign change. As can be seen from the diagram, all values of the input y(t) for which

$$|y(t)| \leqq \hat{\delta}$$

have an output $\hat{y}(t)$ satisfying $|\hat{y}(t)| = 1$ while $|\hat{y}(t)| = 2$

for $|y(t)| \geq \hat{\delta}$. The question arises as to what values should δ be for best performance. This is answered in the following analysis.

Let the transmitted signal ks(t) be defined as

$$ks(t) = kr(t)\ A \cos(w_0 t + \phi)*$$

where

k is the reciprocal of the channel propagation loss factor

r(t) is a random binary digit of duration T seconds with equal probability of being equal to 1 or -1

ϕ is a uniformly distirbuted random variable

$A^2 = \frac{2E}{T}$ (binary digit)

with E (binary digit) is the received signal energy per binary digit

The signal is corrupted by additive gaussian noise n(t) which is given by

$$n(t) = n_1(t) \cos w_0 t - n_2(t) \sin w_0 t$$

$n_1(t)$ and $n_2(t)$ are identically distributed zero mean gaussian random processes each of whose spectrums is white with $\frac{N_o}{2}$ the two sided noise spectral density. The received signal (x(t) = s(t) + n(t)) when multiplied by a coherent sine wave (refer to Figure 2) and integrated results in an input to the quantizer given by

---

*A more generalized representation of a bi-orthogonal PSK transmission could be used. However there is no loss in generality in using this simpler model.

$$y = \frac{\beta A r(t)}{2} + \frac{\beta r(t) A}{2T} \int_0^T \cos 2\, w_0\, t\, dt +$$

$$\frac{\beta}{T} \int_0^T n_1(t) \cos^2 w_0 t\, dt - \frac{\beta}{T} \int_0^T n_2(t) \sin w_0 t \cos w_0 t\, dt$$

Given $r(t)$, $y(t)$ is a gaussian random variable (integration is a linear operation) with the mean of y given by $\bar{y} = \frac{\beta A r(t)}{2}$ under the assumption*

$$\frac{1}{T} \int_0^T \cos 2\, w_0 t\, dt \cong 0$$

The variance of y is computed below

$$\sigma_y^2 = \frac{\beta^2}{T^2} \int_0^T \int_0^T E[n_1(t_1) n_1(t_2)] \cos^2 w_0 t_1 \cos^2 w_0 t_2\, dt_1\, dt_2 +$$

$$\frac{\beta^2}{T^2} \int_0^T \int_0^T E[n_2(t_1) n_2(t_2)] \cos w_0 t_1 \cos w_0 t_2 \sin w_0 t_1 \sin w_0 t_2\, dt_1\, dt_2$$

---

*This occurs if we design our systems such that $2w_0T = \pi \pm n\pi$ (n: 1, 2, 3, ...) or if $\int_0^T \cos 2w_0 t\, dt \ll T$.

Taking the expectation and using the assumption of white noise this reduces to

$$\sigma_y^2 = \frac{\beta^2}{T^2} \frac{N_0}{8}$$

under the assumption that

$$\int_0^T \cos \alpha w_0 t \, dt = 0$$

where α is a positive integer.

The probability of a binary digit being received in error p is then given by

$$p = P[y>0 | r(t)=-1] P(r(t)=-1) +$$

$$P[y>0 | r(t)=1] P(r(t)=1)$$

This reduces to

$$p = P[y>0] = \frac{1}{\sqrt{2\pi} \sigma_y} \int_0^\infty \exp - \left[\frac{(y+\frac{\beta A}{2})^2}{2 \sigma_y^2} \right] dy$$

and can be written in terms of the complementing error function erfc as

$$p = \frac{1}{2} \operatorname{erfc}\left(\sqrt{\frac{E(\text{binary digit})}{N_o}} \right) \tag{15}$$

Equation (15) is the error rate for a bi-phase modulated signal as is to be expected.

In the 2 level quantizer case only sign information of y was extracted. The 4 level quantizer extracts more information from y and enables us to design our system to take advantage of the "tailing off" nature of the gaussian distribution. Thus we will soon see that we can design a 4 level quantizer so that nearly all the time we do not make error the quantizer output is at its highest weighting (± 2) while nearly all the time errors are made the quantizer output is at its lowest weighting ± 1.

We return to our analysis. In figure 12 the probability density function of y is sketched assuming $\beta = \frac{4}{A}$ and $r(t) = -1$.

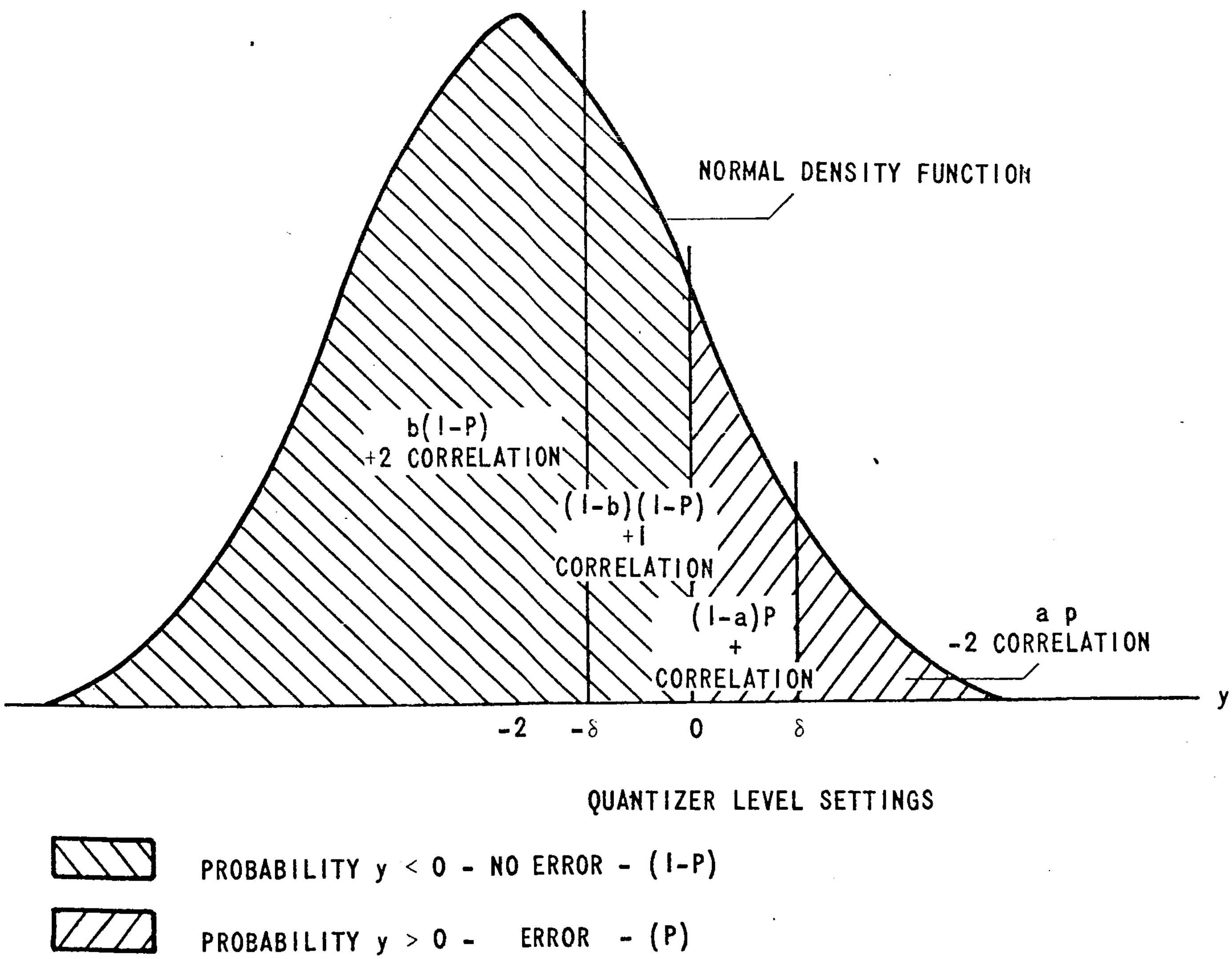

FIGURE 12 - PROBABILITY DISTRIBUTION OF POSSIBLE INPUTS AND OUTPUTS FOR A 4 LEVEL A/D CONVERTER ASSUMING r(t) = -1

The distribution has been divided into four areas representing the 4 possible quantizer output levels and their resultant effect upon the correlation operations which take place in the likelihood detector. Thus if we assume that r(t) was the i^{th} binary digit of a message word $S_j(t)$ then its contribution to U_{jj} would be V_{jji} where*

$$U_{jj} = \sum_{i=2}^{M} v_{jji} + \text{Constant } (i=1 \text{ term}) \qquad (16)^{**}$$

$$\underset{i \neq 1}{v_{jji}} = \begin{cases} 2 \text{ with probability } b(1-p) \\ 1 \text{ with probability } (1-b)(1-p) \\ -1 \text{ with probability } (1-a)(1-p) \\ -2 \text{ with probability } a\,p \end{cases} \qquad (17)$$

On the other hand $\frac{M}{2}$ of the other energy measures (U_{jk}: $k \neq j$ and k: 1, 2, ... M) would have a binary digit weighting factor V_{jki} given by

$$v_{jki} = \begin{cases} -2 \text{ with probability } b(1-p) \\ -1 \text{ with probability } (1-b)(1-p) \\ 1 \text{ with probability } (1-a)(1-p) \\ 2 \text{ with probability } a\,p \end{cases} \qquad (18)$$

---

*Maximum likelihood detector operation results in errors if $U_{jj} < \max\{U_{jk}: \; k \neq j \; i: \; 1,2,\ldots M\}$.

**C is either equal to $\pm$ 1 or $\pm$ 2 and is the same for all U_{jk} whether or not it was detected in error. The message structure is such that all first binary digits in each of the words is positive.

The parameters a and b are given by*

$$\frac{1}{p\,\sigma_y\sqrt{2\pi}}\int_b^{\delta} e^{-\frac{(y+2)^2}{2\sigma_y^2}}\,dy = 1-a \tag{19}$$

and

$$\frac{1}{(1-p)\sigma_y\sqrt{2\pi}}\int_{-\infty}^{-\delta} e^{-(y+2)^2/2\sigma_y^2}\,dy = b \tag{20}$$

To calculate the lower bound to the symbol error probability for this case we generate the $\binom{M}{\frac{M}{4}}$ $z_{\gamma j}$ error vectors as in the previous section (2 level quantizer analysis) but now for each $z_{\gamma j}$ there are 2^M different weightings since each binary digit of the message can take on one of two possible values.** Thus a total of $2^M\binom{M}{\frac{M}{4}}$ error vectors are to be calculated and for each the U_{ji} energy measures are computed and compared to see if an error has been made. Thus we may write the lower bound to the symbol error probability as

$$P_L(E) = p^{\frac{M}{4}}(1-p)^{\frac{3M}{4}}\sum{}' \sum{}' \sum k_{2+s}\left(t,\ \frac{M}{4}-t,\ s,\ \frac{3M}{4}-s\right)^{M} \cdot$$

$$a^t b^s (1-a)^{\frac{M}{4}-t}\ (1-b)^{3/4M-s} \tag{21}$$

---

*Note $\hat{\delta} = \frac{\overline{y}}{2}\,\overline{\delta}$

**The $\binom{M}{\frac{M}{4}}$ error vectors represent the possible ways in which $\frac{M}{4}$ binary digits of a transmitted word can be received in error (before quantization).

where

$$k_{\gamma ts} = \begin{cases} 0 \text{ if } U_{jj} > (U_{ji})_{max} \quad i \neq j \\ 1 \text{ if } (U_{ji})_{max} > U_{jj} \\ \frac{L-1}{L} \text{ if } L \; U_{ji} \; (i \neq j) \text{ are equal to } U_{jj} \end{cases} \tag{22}$$

In Appendix C a computer program is described which computes for a given M the $2^M \binom{M}{\frac{M}{4}}$ error vectors, $\{k_{\gamma ts}\}$ and the $P_L(E)$ and $P_U(E)$ values.

The M = 4 and 8 cases were programmed. The results are presented in Figures 13 through 16. In Figure 13 the bounds on the probability of symbol error are plotted as a function of bit energy to noise spectral density and for the optimum $\hat{\delta}$ quantizer settings. Comparing these results with those presented in Figure 3 we see that for the M = 4 and 8 cases a 4 level quantizer performs within a db of the optimum (infinite level quantizer) detector.

The optimum $\overline{\delta}$ setting is plotted as a function of the chip energy to noise spectral ratio in Figure 14 to determine if the $\overline{\delta}$ setting is independent of M. As can be seen from the graph $\overline{\delta}$ is dependent upon M.

The sensitivity to $\hat{\delta}$ is studied in Figures 15 and 16. In these figures the lower bound to the symbol error rate is plotted as a function of the normalized quantizer level setting $\overline{\delta}$ for several values of the received bit energy to noise spectral density. It can be seen that in both the M = 4 and M = 8 cases the higher E/N_0 is or the lower the $P_L(E)$ (or $P_U(E)$) value the more critical is the $\overline{\delta}$ setting. Thus we see that for $E/N_0 = 16$ and M = 4 there is more than a 2 order magnitude difference in performance between the results of a 2 level quantizer ($\overline{\delta}$=0) and the optimum 4 level quantizer. It is interesting to note that a $\overline{\delta}$ = 8 setting is nearly optimum for all values of E/N_0, in the communication range of interest, for both M = 4 and M = 8. Thus it would be reasonable to design a 4 level quantizer which was adaptive just to one measurement $\overline{y}$ (the d.c. voltage into the quantizer) since $\delta = \overline{y}$ ($\overline{\delta}$=.8).*

*For optimum 4 level quantizer performance two measurements are required, $\overline{y}$ and σ_y^2.

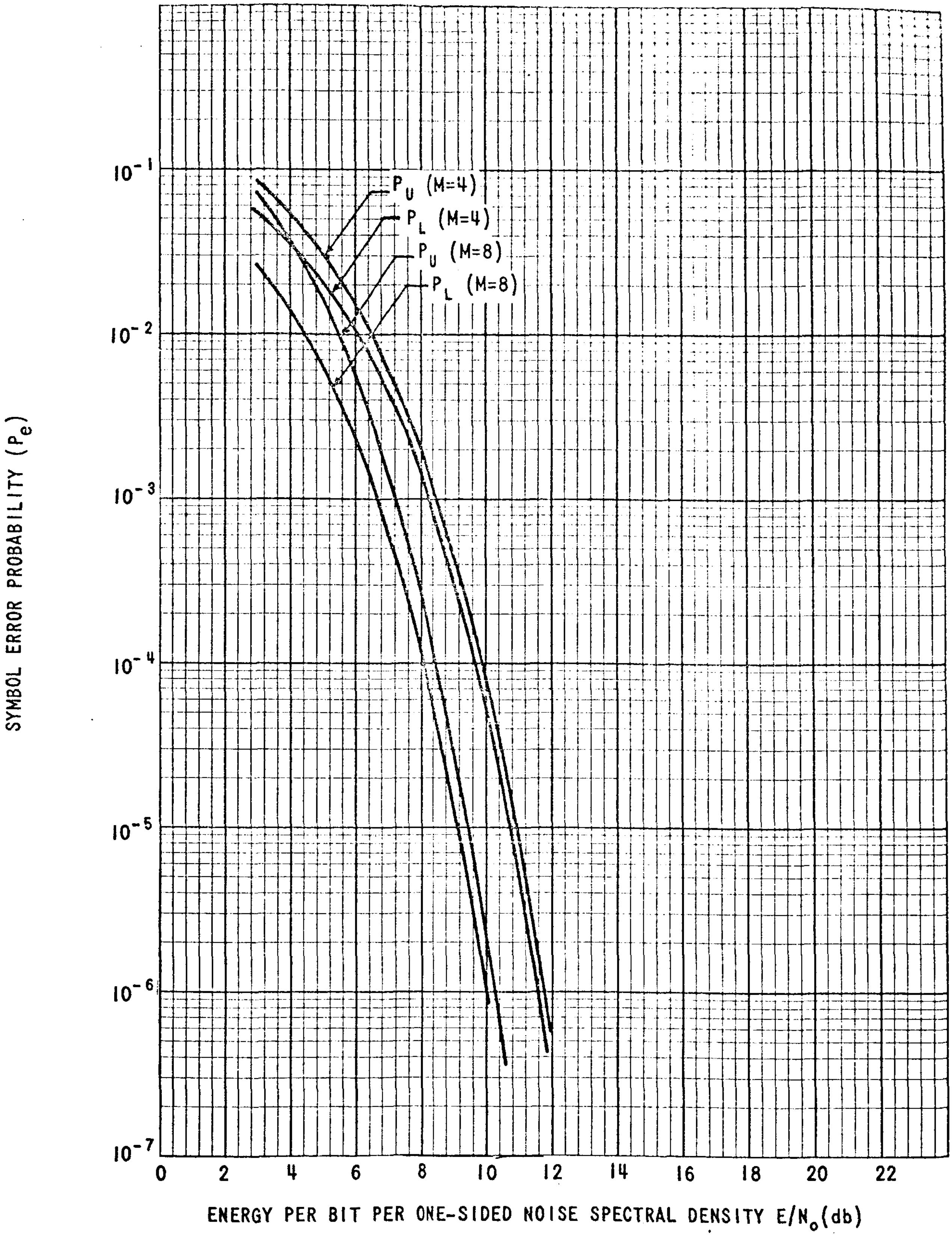

FIGURE 13 - PERFORMANCE OF CODED PHASE COHERENT SYSTEMS USING A 4 LEVEL A/D CONVERTER AND FOR AN OPTIMUM δ SETTING

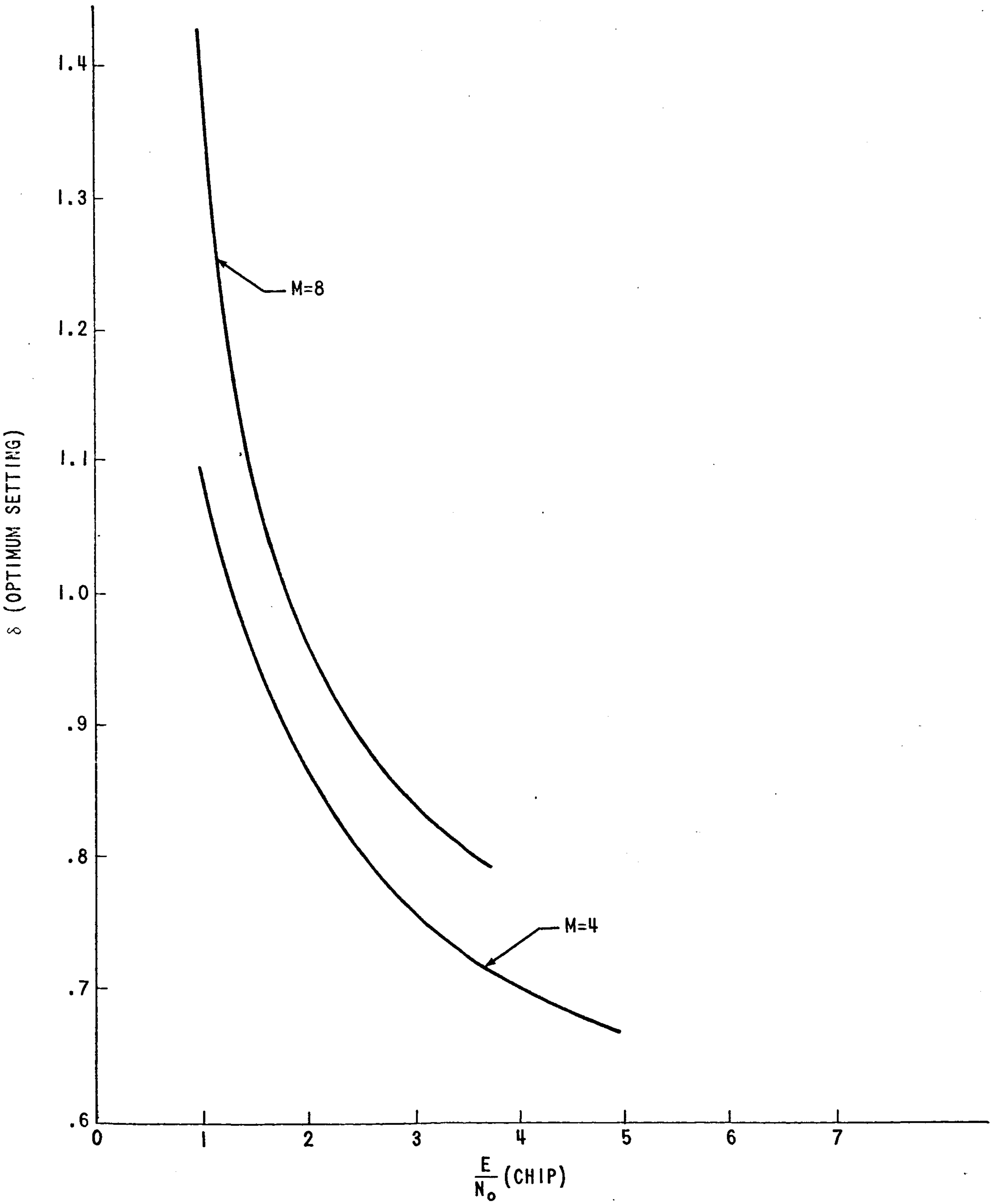

FIGURE 14 - COMPARISON OF OPTIMUM δ SETTING AS A FUNCTION OF $\frac{E}{N_o}$ (CHIP) FOR M=4 AND M=8

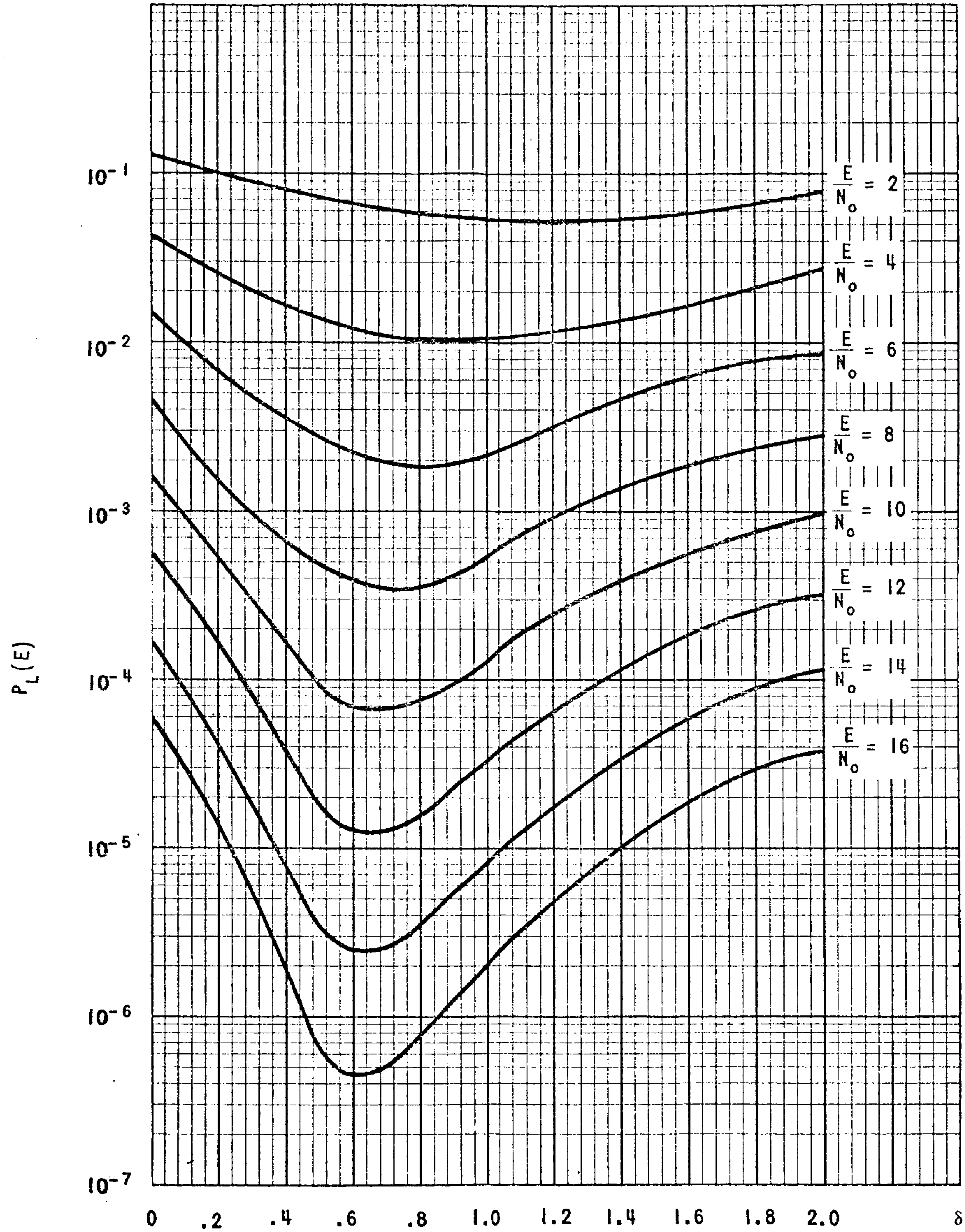

FIGURE 15 - PROBABILITY OF SYMBOL ERROR LOWER BOUND PLOTTED AS A FUNCTION OF THE NORMALIZED QUANTIZER LEVEL SETTING δ (M=4)

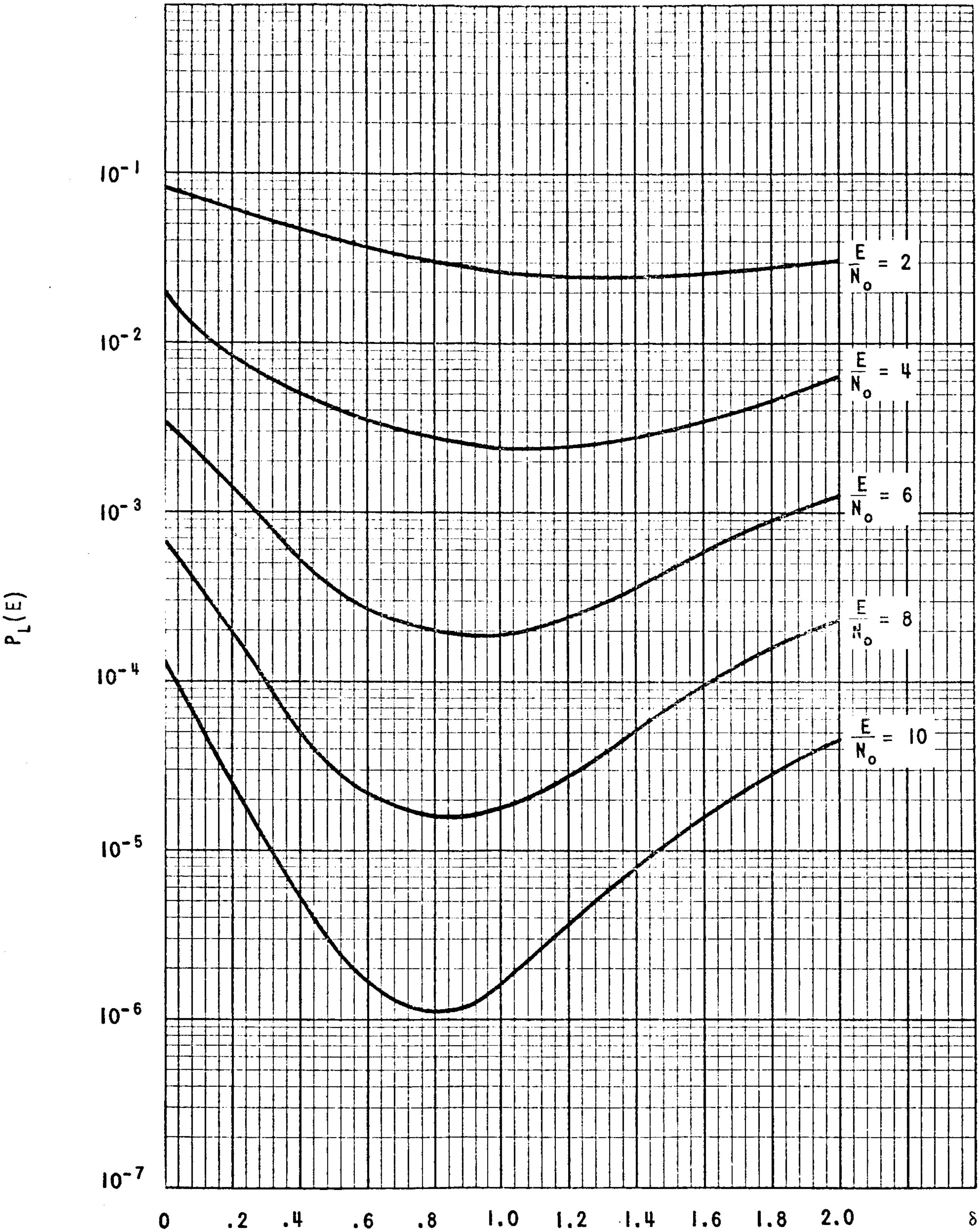

FIGURE 16 - PROABILITY OF SYMBOL ERROR LOWER BOUND PLOTTED AS A FUNCTION OF THE NORMALIZED QUANTIZER LEVEL SETTING δ (M=8)

CONCLUSIONS

When using a 2 level A/D converter, it was shown that an M'ary coding improvement could be achieved. Compared to optimum M'ary detection this receiver did not perform nearly as well. However, for M = 32 it was shown that performance was better than ordinary PSK.

A system using a four level A/D computer was shown able to perform to within a DB of optimum performance (for M = 4 and 8). To obtain this performance a receiver would have to be designed to measure the d.c. voltage $\overline{y}$ and the a.c. power σ_y^2 into the A/D converter and adapt the quantizer level setting $\hat{\delta}$ based on such measurements. If $\hat{\delta}$ is set equal to $.8\overline{y}$ it was shown that nearly optimum 4 level A/D converter performance was achieved. Thus for such a setting only one measurement need be taken.

It would be interesting to extend this work experimentally to determine if the conclusions obtained in this paper for low values of M hold for higher values of M. In addition a study of digital detection of bi-orthogonal and cyclically permutable codes are natural extensions of this paper.

ACKNOWLEDGEMENTS

The author wishes to acknowledge Carol Friend and Natalie M. Myerberg for their help in performing the computer programming as documented in Appendices A, B, and C.

2034-LS-jr L. Schuchman

Attachments
Appendices A, B, and C
References

REFERENCES

1. A. J. Viterbi, "Principles of Coherent Communications," McGraw-Hill, New York, 1966.

2. A. J. Viterbi, "On Coded Phase-Coherent Communications," Jet Propulsion Laboratory Technical Report No. 32-25, August 15, 1960.

3. A. J. Viterbi, "Phase-Coherent Communications Over The Continuous Gaussian Channel," Chapter 7 of "Digital Communications With Space Applications" edited by S. W. Golomb, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1964.

APPENDIX A

Upper and lower bounds for symbol error probability where error weight factor is computed exactly, for two level quantization.

By Natalie M. Myerberg

I. PURPOSE

To compute upper and lower bounds for symbol error probability, where error weight factor is computed exactly, for two level quantization.

II. METHOD

A set of error vectors are generated, representing all possible combinations of M elements, with M/4 errors. The inner product of the error vectors with each signal vector is compared to the maximum inner product value (M/2) and an i-th error vector as a result of these comparisons.

$$K(I) = \frac{\text{Number with maximum inner product value -1}}{\text{Number with maximum inner product value}}$$

The error weight factor (KK) is the summation of the i-th error weight factors (K(I)'s). The error weight factor is used in computing the lower (PEL) and upper (PEU) bounds for symbol error probability.

$$PEL = (KK)(P)^{M/4}\ (1-P)^{(M-M/4)}$$

$$PEU = PEL + \sum_{J=\frac{M}{4}+1}^{M} \binom{M}{J}(P)^{J}\ (1-P)^{(M-J)}$$

where P is the binary digit error probability ranging from 10^{-1} to 10^{-6}.

III. INPUT

M - Order of symbol alphabet

S - Signal vectors

M is read from the first card and S, an MxM array, is read from the remaining cards.

IV. ROUTINES USED

COMB - Finds combinations used in computing the upper bound for symbol error probability

KM16 - Computes error weight factor.

```
C TITLE          KM16
C
C AUTHOR         N. MYERBERG
C
C SPONSOR        L. SCHUCHMAN
C
C DATE           10-23-67
C
C PURPOSE        TO COMPUTE THE ERROR WEIGHT FACTOR FOR M = 4, 8, OR 16
C
C METHOD         THE INNER PRODUCT OF THE ERROR VECTORS WITH EACH S VECTOR
C                IS COMPUTED. THE MAXIMUM INNER PRODUCT VALUE IS COMPARED
C                TO EACH INNER PRODUCT VALUE AND AN ITH ERROR WEIGHT
C                FACTOR IS ASSIGNED TO THE ITH ERROR VECTOR AS A RESULT
C                OF THESE COMPARISONS. THE ERROR WEIGHT FACTOR (KK16) IS
C                THE SUMMATION OF THE ITH ERROR WEIGHT FACTORS.
C
C INPUT          THROUGH COMMON
C                MAX        MAXIMUM INNER PRODUCT VALUE
C                NCHNG      NUMBER OF ERRORS
C                N          NUMBER OF ERROR VECTORS
C                M          ORDER OF SYMBOL ALPHABET
C                S          SIGNAL VECTORS (S-VECTORS)
C
C OUTPUT         THROUGH CALL LIST
C                KK16       ERROR WEIGHT FACTOR USED IN COMPUTATION OF
C                           UPPER AND LOWER BOUNDS FOR PROBABILITY OF
C                           SYMBOL ERROR
C
C
      SUBROUTINE KM16(KK16)
      REAL KMAX,KK16
      INTEGER A,S
      DIMENSION S(16,16),A(16),IP(16)
      COMMON MAX,NCHNG,N,M,S
C
C        GENERATE ERROR VECTOR
C
      REWIND 4
      IEND=M-NCHNG+1
      KEND=M-1
      IW=0
      DO 40 ISTRT=1,IEND
      DO 5 I1=1,M
    5 A(I1)=1
      MM=M+1-ISTRT
      A(MM)=-1
      IF(NCHNG.EQ.1) GO TO 3
      IF(NCHNG.GT.2) GO TO 1
      JP1=ISTRT
      GO TO 3
    1 DO 30 I1=ISTRT,I3
      MMM1=MM-1
      DO 6 I2=1,MMM1
    6 A(I2)=1
```

```
                 A(M-II)=-1
                 IIP1=II+1
      ?          DO 20 J=IIP1,14
                 MMIIP1=M-IIP1
                 DO 7 I3=1,MMIIP1
      7          A(I3)=1
                 A(M-J)=-1
                 JP1=J+1
      ?          DO 10 K=JP1,KEND
                 MMJP1=M-JP1
                 DO 8 I4=1,MMJP1
      8          A(I4)=1
                 A(M-K)=-1
      9          WRITE(4)(A(I),I=1,M)
                 IW=IW+1
                 IF(NCHNG.EQ.1) GO TO 20
     10          CONTINUE
                 IF(NCHNG.EQ.2) GO TO 40
     20          CONTINUE
     30          CONTINUE
     40          CONTINUE
                 REWIND 4
C
C        THIS PROCEDURE DONE FOR S1 ROW BY ROW
C          READ ONE ROW OF THE ERROR VECTOR
C
                 KK16=0
                 DO 70 I=1,N
                 READ(4)(A(IA),IA=1,M)
                 NMAX=0
C
C          COMPUTE INNER PRODUCT
C
                 DO 60 L=1,M
                 IP(L)=0
                 DO 50 K=1,M
     50          IP(L)=IP(L)+S(L,K)*A(K)
C
C          COMPUTE K(I) FOR ROW I
C
                 IF(IP(L).EQ.MAX) NMAX=NMAX+1
     60          CONTINUE
                 IF(NMAX.EQ.0) NMAX=1
                 KMAX=(NMAX-1.)/NMAX
C
C          COMPUTE K BY SUMMING K(I)
C
     70          KK16=KK16+KMAX
    101          FORMAT(1H ,16I4)
                 RETURN
                 END
```

```
C TITLE          MAIN
C
C AUTHOR         N. MYERBERG
C
C
C SPONSOR        L. SCHUCHMAN
C DATE           11-15-67
C
C PURPOSE        TO COMPUTE UPPER AND LOWER BOUNDS FOR PROBABILITY OF
C                SYMBOL ERROR FOR RO[illegible] LEVEL QUANTIZATION.
C
C METHOD         THE ERFC FUNCTION IS USED TO COMPUTE P,A,AND B, WHICH ARE
C                USED IN COMPUTING THE BOUNDS FOR PROBABILITY OF ERROR
C
C INPUT          THROUGH NAMELIST
C                EO          FIRST VALUE FOR E/NO
C                EINC        INCREMENT FOR E/NO
C                EEND        LAST VALUE FOR E/NO
C                DELO        FIRST VALUE FOR DELTA
C                DELINC      INCREMENT FOR DELTA
C                DELEND      FINAL VALUE FOR DELTA
C                N          ORDER OF SYMBOL ALPHABET
C
C                THROUGH READ STATEMENT
C                S           SIGNAL VECTORS (S-VECTORS)
C
C OUTPUT         E           E/NO USED IN FOLLOWING COMPUTATIONS
C                DELTA       VALUE OF DELTA USED IN COMPUTING A AND B
C                P           P=ERFC(SQRT(E/NO))/2
C                            P REPRESENTS PROBABILITY OF BINARY DIGIT ERROR
C                F           F=(1+B)/(2+[illegible])
C                A           A=ERFC(SQRT(E/NO)*(1+DELTA/2))/(2*P)
C                B           B=1-ERFC(SQRT(E/NO)*(1-DELTA/2))/(2*(1-P))
C                            +P/(1-P)
C                PU          UPPER BOUND FOR PROBABILITY OF ERROR
C                PL          LOWER BOUND FOR PROBABILITY OF ERROR
C
C SUBR. USED     ERFC        TO COMPUTE ERFC
C                COMB        TO FIND COMBINATIONS
C                LBND        TO COMPUTE ERROR WEIGHT FACTOR
C
                 INTEGER S
                 DIMENSION S(8,8)
                 COMMON S
                 NAMELIST/INPUT/EO,EINC,EEND,DELO,DELINC,DELEND,N
C
 100             READ(5,INPUT)
                 READ(5,104)((S(I,J),J=1,N),I=1,N)
 104             FORMAT(16I3)
C
                 NCHNG=N/4
                 E=EO
                 E1=(8./3.)*E
                 INDEX=0
C
```

```
C
C     COMPUTE PEL AND PEU
C
      NCHNG1=NCHNG+1
      P=.5
4     PEL=KK*P**NCHNG*(1.-P)**(M-NCHNG)
      PEU=PEL
      DO 70 J=NCHNG1,M
70    PEU=PEU+COMB(M,J)*P**J*(1.-P)**(M-J)
      WRITE(6,205) P
205   FORMAT(1H0,50X,'P =',1PE16.8,//)
      WRITE(6,203)PEL
203   FORMAT(1H0,' PEL = ',1PE16.8)
      WRITE(6,204)PEU
204   FORMAT(1H0,'PEU = ',1PE16.8)
      IF(P.LE.5.E-2) GO TO 1
      P=P*.1
      GO TO 4
      END
```

```
C TITLE          COMB
C
C AUTHOR         C.A. FRIEND
C
C DATE           10-16-67
C
C PURPOSE        GENERALIZED ROUTINE FOR FINDING COMBINATIONS FOR N AS
C                LARGE AS 200 USING PASCAL TRIANGLE
C
C
                 FUNCTION COMB(N,R)
C
                 INTEGER R
                 DIMENSION X(200,2)
C
                 IF(R.GT.N)GO TO 20
                 IF(R.EQ.0.OR.R.EQ.N)GO TO 1
                 IF(R.EQ.1.OR.R.EQ.(N-1))GO TO 2
                 GO TO 3
  1              COMB=1.
                 RETURN
  2              COMB=N
                 RETURN
C
  3              M=N+1
                 DO 4 L=1,2
                 DO 4 K=1,M
  4              X(K,L)=0.
C
                 X(1,1)=1.
                 NN=0
  5              CONTINUE
                 J=MOD(NN,2)+1
                 JJ=MOD(NN+1,2)+1
                 NN=NN+1
                 X(1,JJ)=X(1,J)
                 DO 10 I=1,NN
 10              X(I+1,JJ)=X(I,J)+X(I+1,J)
C
                 IF(NN.LT.N)GO TO 5
                 COMB=X(R+1,JJ)
 16              RETURN
 20              WRITE(6,100)
100              FORMAT(10X,'ERROR IN COMBINATION N =',I5,'R =',I5)
                 GO TO 16
C
                 END
```

APPENDIX B

Upper and lower bounds for symbol error probability, where error weight factor is upper and lower bounded, for two level quantization.

By Natalie M. Myerberg and Carol A. Friend

I. PURPOSE

To compute upper and lower bounds for symbol error probability, without computing error weight factor exactly for two level quantization. Error weight factor is upper and lower bounded.

II. METHOD

Generalized formulas for the upper and lower bounds for symbol error probability are used:

A. Lower bound

$$PEL = \frac{M-1}{4} \binom{\frac{M}{2}}{\frac{M}{4}} \cdot P^{\left(\frac{M}{4}\right)} \cdot (1-P)^{\left(M-\frac{M}{4}\right)}$$

B. Upper bound for $M < 64$:

$$PEU_1 = \frac{M-1}{2} \binom{\frac{M}{2}}{\frac{M}{2}} \cdot P^{\left(\frac{M}{4}\right)} \cdot (1-P)^{\left(M-\frac{M}{4}\right)} + \left[\binom{\frac{M}{2}}{\frac{M}{4}+1}\right.$$

$$\left. + \frac{M}{4} \cdot \binom{\frac{M}{2}}{\frac{M}{4}}\right] \cdot (M-1) \cdot P^{\left(\frac{M}{4}+1\right)} \cdot (1-P)^{\left[M-\left(\frac{M}{4}+1\right)\right]}$$

Appendix B (Cont.)

$$PEU = PEU_1 + \sum_{J=(\frac{M}{4}+2)}^{M} \binom{M}{J} \cdot P^J \cdot (1-P)^{(M-J)}$$

for $M \geq 64$:

$$PEU = PEU_1 + \sum_{J=(\frac{M}{4}+3)}^{M} \binom{M}{J} \cdot P^J \cdot (1-P)^{(M-J)}$$

$$+ \left[\binom{\frac{M}{2}}{\frac{M}{4}+2} + \frac{M}{2} \cdot \binom{\frac{M}{2}}{\frac{M}{4}+1} + \frac{1}{2} \cdot \binom{\frac{M}{2}}{\frac{M}{4}} \cdot \binom{\frac{M}{2}}{2}\right]$$

$$\cdot (M-1) \cdot P^{(\frac{M}{4}+2)} (1-P)^{(\frac{3}{4}M-2)}$$

III. INPUT

M - Order of symbol alphabet

PP - Probability of binary digit error

M and PP are read from data cards using Namelist. PP is an array containing as many as seven elements each representing a binary digit error probability.

IV. ROUTINE USED

COMB - Finds combinations used in computing the lower and upper bounds for symbol error probability.

```
'I      FOR      BOUNDS,BOUNDS
C TITLE          BOUNDS
C
C AUTHOR         N. MYERBERG
C
C SPONSOR        L. SCHUCHMAN
C
C DATE           10-30-67
C
C PURPOSE        GENERALIZED ROUTINE FOR FINDING LOWER AND UPPER BOUNDS
C                FOR PROBABILITY OF SYMBOL ERROR FOR TWO LEVEL
C                QUANTIZATION
C
C METHOD         WEIGHT FACTOR IS UPPER AND LOWER BOUNDED
C
C INPUT          THROUGH NAMELIST
C                M          ORDER OF SYMBOL ALPHABET
C                PP          PROBABILITY OF BINARY DIGIT ERROR
C
C OUTPUT         PRINTED
C                M          ORDER OF SYMBOL ALPHABET
C                PP          PROBABILITY OF BINARY DIGIT ERROR
C                PEL         LOWER BOUND FOR PROBABILITY OF SYMBOL ERROR
C                PEU         UPPER BOUND FOR PROBABILITY OF SYMBOL ERROR
C
C ROUTINES       COMB        USED TO FIND COMBINATIONS IN COMPUTING PEL
C USED                       AND PEU
C
                 DOUBLE PRECISION P,PEU,PEL,COMB
                 DIMENSION PP(7)
                 NAMELIST/INPUT/M/PINPUT/PP
  2              READ(5,PINPUT)
  1              READ(5,INPUT)
                 WRITE(6,104) M
  104            FORMAT(1H1,57X,'M = ',I3)
                 NCHNG=M/4
                 NCHNG1=NCHNG+1
                 NCHNG2=NCHNG+2
                 NCHNG3=NCHNG+3
                 NBEG=NCHNG2
                 MAX=M/2
                 DO 20 I=1,7
                 P=PP(I)
                 PEL=((M-1.)/4.)*COMB(MAX,NCHNG)*P**NCHNG*(1.-P)**(M-NCHNG
     .)
                 PEU=((M-1.)/2.)*COMB(MAX,NCHNG)*P**NCHNG*(1.-P)**(M-NCHNG
     .)+(COMB(MAX,NCHNG1)+NCHNG*(COMB(MAX,NCHNG )))*(M-1.)*P**(NCHNG1)*(
     .1.-P)**(M-NCHNG1)
                 IF(M.GE.64) NBEG=NCHNG3
                 DO 10 J=NBEG,M
  10             PEU=PEU+COMB(M,J)*P**J*(1.-P)**(M-J)
                 IF(M.LT.64) GO TO 12
                 PEU=PEU+(COMB(MAX,NCHNG2)+COMB(MAX,NCHNG1)*(M/2)+.5*
     .           COMB(MAX,NCHNG)*COMB(MAX,2))*(M-1.)*P**(NCHNG2)
     .           *(1.-P)**(.75*M-2.)
```

```
12        WRITE(6,101) P
101       FORMAT(1H0,54X,'P = ',F8.6)
          WRITE(6,102) PEL
102       FORMAT(1H0,'PEL = ',1PD16.8)
          WRITE(6,103) PEU
103       FORMAT(1H ,'PEU = ',1PD16.8,//)
20        CONTINUE
          GO TO 1
          END
```

```
C TITLE          COMB
C
C AUTHOR         C.A. FRIEND
C
C DATE           10-16-67
C
C PURPOSE        GENERALIZED ROUTINE FOR FINDING COMBINATIONS FOR N AS
C                LARGE AS 200 USING PASCAL TRIANGLE
C
C
                 FUNCTION COMB(N,R)
C
                 INTEGER R
                 DIMENSION X(200,2)
C
                 IF(R.GT.N)GO TO 20
                 IF(R.EQ.0.OR.R.EQ.N)GO TO 1
                 IF(R.EQ.1.OR.R.EQ.(N-1))GO TO 2
                 GO TO 3
  1              COMB=1.
                 RETURN
  2              COMB=N
                 RETURN
C
  3              M=N+1
                 DO 4 L=1,2
                 DO 4 K=1,M
  4              X(K,L)=0.
C
                 X(1,1)=1.
                 NN=0
  5              CONTINUE
                 J=MOD(NN,2)+1
                 JJ=MOD(NN+1,2)+1
                 NN=NN+1
                 X(1,JJ)=X(1,J)
                 DO 10 I=1,NN
  10             X(I+1,JJ)=X(I,J)+X(I+1,J)
C
                 IF(NN.LT.N)GO TO 5
                 COMB=X(R+1,JJ)
  16             RETURN
  20             WRITE(6,100)
  100            FORMAT(10X,'ERROR IN COMBINATION N =',I5,'R =',I5)
                 GO TO 16
C
                 END
```

APPENDIX C

Upper and lower bounds for symbol error probabilty for four level quantization.

By Natalie M. Myerberg

I. PURPOSE

To compute lower and upper bounds for symbol error probability for four level quantization.

II. METHOD

Error vectors are generated representing all possible combinations of M elements with M/4 errors. For four level quantization, the elements of these error vectors are weighted, and, consequently, each element may assume one of four possible values after quantization. Two quantized element values represent error (-1, -2), while two represent non-error (+1, +2). Thus, the entire array of error vectors for four level quantization contains $\binom{M}{\frac{M}{4}} (2)^M$ error vectors of M elements each.

Corresponding to each quantized element value of an error vector, is a numerical value representing the probability that an element will be quantized with that value. Given that an error was made with probability P in the channel, then A is the probability that the error will be quantized as a -2 value; and (1-A), as a -1 value. Given that no error occurred with probability (1-P) in the channel, then B is the probability that the error will be quantized as a +2 value; and (1-B), as a +1 value.

QUANTIZED ELEMENT VALUE	NUMERICAL VALUE
+2	(B)(1-P)
+1	(1-B)(1-P)
-1	(1-A)(P)
-2	(A)(P)

Appendix C (Cont.)

where

$$A = \frac{\text{ERFC}\left[\sqrt{\frac{E}{NO}}\ \left(1+\frac{DELTA}{2}\right)\right]}{2P}$$

$$B = 1 - \frac{\text{ERFC}\left[\sqrt{\frac{E}{NO}}\ \left(1-\frac{DELTA}{2}\right)\right]}{2(1-P)} + P(1-P)$$

$$P = \frac{\text{ERFC}\left(\sqrt{\frac{E}{NO}}\right)}{2}$$, the binary digit error probability

Thus, assigned to each error vector is a weight value which is the product of the numerical values corresponding to the quantized element values of the error vector.

Another weight factor (AK) is assigned to each error vector according to values obtained by computing the inner product of the error vector with each signal vector.

(AK = $\frac{\text{Number with maximum inner product value } -1}{\text{Number with maximum inner product value}}$,

except if the first inner product value is less than any other inner product value for that error vector, in which case AK = 1).

The final weighted contribution to the lower bound for probability of symbol error caused by each error vector is the product of the two weight factors assigned to it. Since $P^{M/4}(1-P)^{(M-M/4)}$ can be factored from each assigned weight value, the weighted contributions of each error vector are summed (SUM), and then multiplied by $P^{M/4}\ (1-P)^{(M-M/4)}$ to result in the lower bound for probability of symbol error.

Appendix C (Cont.)

$$PEL = SUM\ (P)^{M/4}\ (1-P)^{(M-M/4)}$$

$$PEU = PEL + \sum_{J=\frac{M}{4}+1}^{M} \binom{M}{J} (P)^{J} (1-P)^{(M-J)}$$

III. INPUT

EO - starting value for E/NO (ratio of energy per chip to noise spectral density)

EINC - increment for E/NO

EEND - last value for E/NO

DELO - first value for DELTA (normalized quantizer lever setter)

DELINC - increment for DELTA

DELEND - last value for DELTA

N - order of symbol alphabet (M)

S - signal vectors

EO, EINC, EEND, DELO, DELINC, DELEND, and N are read from data cards using namelist. S is an N XN array, read from the remaining data cards.

V. ROUTINES USED

ERFC - Computers ERFC

COMB - Finds combinations used in computing upper bound for symbol error probability

LBND - Computes error weight factor (SUM) for lower bound for symbol error probability.

```
C TITLE          SIGNS4
C
C AUTHOR         N. MYERBERG
C
C DATE           11-16-67
C
C PURPOSE        TO GENERATE AN ARRAY OF SIGNS FOR THE ERROR VECTORS FOR
C                N=4
C
                 SUBROUTINE SIGNS4(N,NSIGN)
                 DIMENSION NSIGN(28,8)
                 DO 5 I=1,N
                 DO 5 J=1,N
  5              NSIGN(I,J)=1
                 I=1
                 DO 10 J=1,4
                 NSIGN(I,5-J)=-1
                 I=I+1
                 IF(I.EQ.5) RETURN
 10              NSIGN(I,5-J)=1
                 RETURN
                 END
```

~~11 FOR SIGNS,SIGNS~~

```
C TITLE          SIGNS
C
C AUTHOR         N. MYERBERG
C
C DATE           11-13-67
C
C PURPOSE        TO GENERATE AN ARRAY OF SIGNS FOR THE ERROR VECTORS FOR
C                N=8
C
                 SUBROUTINE SIGNS(N,NSIGN)
                 DIMENSION NSIGN(28,8)
                 DO 5 I=1,28
                 DO 5 J=1,8
  5              NSIGN(I,J)=1
                 I=1
                 DO 20 J=1,7
                 NSIGN(I,N+1-J)=-1
                 DO 10 K=J,7
                 NSIGN(I,N-K)=-1
                 I=I+1
                 IF(I.EQ.29) RETURN
                 NSIGN(I,N+1-J)=-1
 10              NSIGN(I,N-K)=1
 20              NSIGN(I,N+1-J)=1
                 RETURN
                 END
```

```
C TITLE         NONE
C
C AUTHOR        N. MYERBERG
C
C DATE          11-13-67
C
C PURPOSE       TO GENERATE ON TAPE A VECTOR OF N ELEMENTS EQUAL TO L
C
C INPUT         THROUGH CALL LIST
C               N          ORDER OF SYMBOL ALPHABET
C               L           NUMBER WHICH ELEMENTS ARE EQUAL TO
C
C OUTPUT        ON TAPE
C               NA(N)       VECTOR GENERATED
C
                SUBROUTINE NONE(N,L)
                DIMENSION NA(8)
                DO 10 I=1,N
 10             NA(I)=L
                WRITE(4)(NA(II),II=1,N)
                RETURN
                END
```

```
C TITLE          ONE
C
C AUTHOR         N. MYERBERG
C
C DATE           11-13-67
C
C PURPOSE        TO GENERATE A SET OF VECTORS REPRESENTING ALL POSSIBLE
C                COMBINATIONS OF N THINGS TAKEN ONE AT A TIME
C
C INPUT          THROUGH CALL LIST
C                N           ORDER OF SYMBOL ALPHABET
C                L           NUMBER TO BE DISTRIBUTED
C                M           NUMBER REMAINING ELEMENTS EQUAL
C
C OUTPUT         ON TAPE
C                NA          SET OF VECTORS GENERATED
C
                 SUBROUTINE ONE(N,L,M)
                 DIMENSION NA(8)
                 DO 5 I=1,N
  5              NA(I)=L
                 DO 10 J=1,N
                 NA(N+1-J)=M
                 WRITE(4)(NA(II),II=1,N)
 10              NA(N+1-J)=L
                 RETURN
                 END
```

```
C TITLE          TWO
C
C AUTHOR         M. MYERBERG
C
C DATE           11-13-67
C
C PURPOSE        TO GENERATE A SET OF VECTORS REPRESENTING ALL POSSIBLE
C                COMBINATIONS OF N THINGS TAKEN TWO AT A TIME
C
C INPUT          N         ORDER OF SYMBOL ALPHABET
C                L          NUMBER TO BE DISTRIBUTED IN EACH VECTOR
C                M          NUMBER REMAINING ELEMENTS EQUAL
C
C OUTPUT         ON TAPE
C                NA         SET OF VECTORS GENERATED
C
                 SUBROUTINE TWO(N,L,M)
                 DIMENSION NA(8)
                 DO 5 I=1,N
  5              NA(I)=L
                 NM1=N-1
                 DO 20 J=1,NM1
                 NA(N+1-J)=M
                 DO 10 K=J,NM1
                 NA(N-K)=M
                 WRITE(4)(NA(II),II=1,N)
  10             NA(N-K)=L
  20             NA(N+1-J)=L
                 RETURN
                 END
```

```
C TITLE          THREE
C
C AUTHOR         N. MYERBERG
C
C DATE           11-13-67
C
C PURPOSE        TO GENERATE A SET OF VECTORS REPRESENTING ALL POSSIBLE
C                COMBINATIONS OF N THINGS TAKEN THREE AT A TIME
C
C INPUT          N          ORDER OF SYMBOL ALPHABET
C                L           NUMBER TO BE DISTRIBUTED IN EACH VECTOR
C                M           NUMBER REMAINING ELEMENTS EQUAL
C
C OUTPUT         ON TAPE
C                NA          SET OF VECTORS GENERATED
C
                 SUBROUTINE THREE(N,L,M)
                 DIMENSION NA(8)
                 DO 5 I=1,N
 5               NA(I)=L
                 NM1=N-1
                 NM2=N-2
                 DO 30 I=1,NM2
                 NA(N+1-I)=M
                 DO 20 J=I,NM2
                 NA(N-J)=M
                 DO 10 K=J,NM2
                 NA(NM1-K)=M
                 WRITE(4)(NA(II),II=1,N)
 10              NA(NM1-K)=L
 20              NA(N-J)=L
 30              NA(N+1-I)=L
                 RETURN
                 END
```

```
C TITLE         FOUR
C
C AUTHOR        N. MYERBERG
C
C DATE          11-13-67
C
C PURPOSE       TO GENERATE A SET OF VECTORS REPRESENTING ALL POSSIBLE
C               COMBINATIONS OF N THINGS TAKEN FOUR AT A TIME
C
C INPUT         N          NUMBER OF ELEMENTS OF EACH VECTOR
C               L          NUMBER TO BE DISTRIBUTED IN EACH VECTOR
C               M          NUMBER REMAINING ELEMENTS EQUAL
C
C OUTPUT        ON TAPE
C               NA         SET OF VECTORS GENERATED
C
                SUBROUTINE FOUR(N,L,M)
                DIMENSION NA(8)
                DO 5 I=1,N
  5             NA(I)=L
                NM3=N-3
                DO 40 I=1,NM3
                NA(N+1-I)=M
                DO 30 J=I,NM3
                NA(N-J)=M
                DO 20 K=J,NM3
                NA(N-1-K)=M
                DO 10 LL=K,NM3
                NA(N-2-LL)=M
                WRITE(4)(NA(II),II=1,N)
 10             NA(N-2-LL)=L
 20             NA(N-1-K)=L
 30             NA(N-J)=L
 40             NA(N+1-I)=L
                RETURN
                END
```

4

```
C TITLE        COMB
C
C AUTHOR       C.A. FRIEND
C
C DATE         10-16-67
C
C PURPOSE      GENERALIZED ROUTINE FOR FINDING COMBINATIONS FOR N AS
C              LARGE AS 200 USING PASCAL TRIANGLE
C
C
               FUNCTION COMB(N,R)
C
               INTEGER R
               DIMENSION X(200,2)
C
               IF(R.GT.N)GO TO 20
               IF(R.EQ.0.OR.R.EQ.N)GO TO 1
               IF(R.EQ.1.OR.R.EQ.(N-1))GO TO 2
               GO TO 3
  1            COMB=1.
               RETURN
  2            COMB=N
               RETURN
C
  3            M=N+1
               DO 4 L=1,2
               DO 4 K=1,M
  4            X(K,L)=0.
C
               X(1,1)=1.
               NN=0
  5            CONTINUE
               J=MOD(NN,2)+1
               JJ=MOD(NN+1,2)+1
               NN=NN+1
               X(1,JJ)=X(1,J)
               DO 10 I=1,NN
  10           X(I+1,JJ)=X(I,J)+X(I+1,J)
C
               IF(NN.LT.N)GO TO 5
               COMB=X(R+1,JJ)
  16           RETURN
  20           WRITE(6,100)
  100          FORMAT(10X,'ERROR IN COMBINATION N =',I5,'R =',I5)
               GO TO 16
C
               END
```

```
C TITLE          ERFC
C
C AUTHOR         N. MYERBERG
C
C DATE           11-9-67
C
C PURPOSE        TO CALCULATE THE ERFC FUNCTION FOR A GIVEN X
C
C METHOD         X LESS THAN 1.51 TAYLOR'S SERIES
C                X GREATER THAN OR EQUAL TO 1.51 CONTINUED FRACTIONS
C
                 FUNCTION ERFC(X)
                 DIMENSION T(101)
                 IF(X.LT.1.51) GO TO 2
                 ERFC=EXP(-X**2)*((.5*X)/(X**2+.5-.5/(X**2+2.5-3./(X**2+4.
     .5-7.5/(X**2+6.5-10.803/(X**2+4.269))))))
                 ERFC=ERFC*(2./SQRT(3.1415927))
                 RETURN
  2              T(1)=X
                 SUM=T(1)
                 DO 10 I=1,100
                 T(I+1)=-((2.*I-1.)*X**2*T(I))/(I*(2.*I+1.))
                 SUM=SUM+T(I+1)
                 IF(ABS(T(I+1))-1.E-10) 1,1,10
  1              ERFC=1.-((2./SQRT(3.1415927))*SUM)
                 RETURN
  10             CONTINUE
                 RETURN
                 END
```

```
C TITLE          LBND
C
C AUTHOR         N. MYERBERG
C
C
C SPONSOR        L. SCHUCHMAN
C DATE           11-14-67
C
C PURPOSE        TO COMPUTE THE ERROR WEIGHT FACTOR USED IN COMPUTING
C                UPPER AND LOWER BOUNDS FOR PROBABILITY OF SYMBOL ERROR.
C
C METHOD         GENERATE ERROR VECTORS.  ASSIGN NUMERICAL VALUE (A,1-A,
C                B, OR (1-B) TO EACH QUANTIZED ELEMENT VALUE OF EACH ERROR
C                VECTOR ACCORDING TO THE VALUE OF THE ELEMENT.  ASSIGN
C                A WEIGHT FACTOR (AK) TO EACH ERROR VECTOR ACCORDING TO
C                VALUES ABTAINED BY COMPUTING THE INNER PRODUCT OF THE
C                ERROR VECTOR WITH EACH SIGNAL VECTOR.  THE FINAL WEIGHT
C                FACTOR ASSIGNED TO EACH ERROR VECTOR IS THE PRODUCT OF
C                THE NUMERICAL VALUES AND THE WEIGHT FACTOR FOR THAT ERROR
C                VECTOR.  THE ERROR WEIGHT FACTOR IS THE SUM 3F THE FINAL
C                WEIGHT FACTORS ASSIGNED TO EACH ERROR VECTOR.
C
C INPUT          THROUGH A CALL LIST
C                N         ORDER OF SYMBOL ALPHABET
C                AA         A-VALUE COMPUTED IN MAIN PROGRAM
C                B          B-VALUE COMPUTED IN MAIN PROGRAM
C                INDEX      INDICATES IF ERROR VECTORS MUST BE GENERATED
C
C                THROUGH COMMON
C                S          SIGNAL VECTORS (S-VECTORS)
C
C OUTPUT         THROUGH CALL LIST
C                SUM        ERROR WEIGHT FACTOR
C
C SUBR. USED     COMB
C                SIGNS
C                SIGNS4
C                NONE
C                ONE
C                TWO
C                THREE
C                FOUR
C
C
                 SUBROUTINE LBND(N,AA,B,SUM,INDEX)
                 INTEGER A,S
                 DOUBLE PRECISION ATERM,BTERM,TERM,DSUM
                 DIMENSION S(8,8),NSIGN(28,8),NA(8),A(8),KA(8)
                 COMMON S
C
                 NTWOS=2.**N
                 NCHNG=N/4
                 NSIGNS=COMB(N,NCHNG)
C
                 IF(N.EQ.4) GO TO 1
```

```
      CALL SIGNS(N,NSIGN)
      GO TO 2
    1 CALL SIGNS4(N,NSIGN)
C
    2 IF(INDEX.EQ.1) GO TO 3
C
C     GENERATE ERROR VECTORS
C
      REWIND 4
      CALL NONE(N,1)
      CALL ONE(N,1,2)
      CALL TWO(N,1,2)
      CALL THREE(N,1,2)
      CALL FOUR(N,1,2)
      IF(N.EQ.4) GO TO 3
      CALL THREE(8,2,1)
      CALL TWO(8,2,1)
      CALL ONE(8,2,1)
      CALL NONE(8,2)
    3 REWIND 4
C
      DSUM=0.D0
      DO 70 II=1,NTWOS
      READ(4)(NA(I),I=1,N)
      DO 70 I=1,NSIGNS
      DO 30 J=1,N
   30 A(J)=NA(J)*NSIGN(I,J)
C
C     FIND EXPONENTS FOR A,(1-A),B,(1-B), AND FIND PRODUCT OF THESE
C      TERMS
C
      NX=0
      NY=0
      MX=0
      MY=0
      DO 35 I35=1,N
      IF(A(I35).EQ.1) NX=NX+1
      IF(A(I35).EQ.2) NY=NY+1
      IF(A(I35).EQ.-1) MX=MX+1
      IF(A(I35).EQ.-2) MY=MY+1
   35 CONTINUE
      IF(NX+NY.NE.N-NCHNG) GO TO 99
      IF(MX+MY.NE.NCHNG) GO TO 99
C
      ATERM=(1.D0-DBLE(AA))**MX*DBLE(AA)**MY
      IF(AA.EQ.1..AND.MX.EQ.0) ATERM=1.D0
      BTERM=(1.D0-DBLE(B))**NX*DBLE(B)**NY
      IF(B.EQ.1..AND.NX.EQ.0) BTERM=1.D0
      TERM=ATERM*BTERM
C
C     COMPUTE INNER PRODUCT OF ERROR VECTOR WITH EACH S VECTOR AND FIND
C      CORRESPONDING WEIGHT FACTOR(AK)
C
      DO 60 J=1,N
      KA(J)=0
```

```
      DO 50 K=1,N
   50 KA(J)=KA(J)+S(J,K)*A(K)
      IF(J.EQ.1) KMAX=KA(1)
      IF(KA(J).GT.KMAX) KMAX=KA(J)
   60 CONTINUE
C
      IF(KA(1).LT.KMAX) GO TO 66
      KCNT=0
      DO 65 K65=1,N
      IF(KA(K65).EQ.KMAX) KCNT=KCNT+1
   65 CONTINUE
      IF(KCNT.EQ.0) KCNT=1
      AK=(KCNT-1.)/KCNT
      GO TO 67
   66 AK=1.
C
   67 DSUM=TERM*AK+DSUM
C
      GO TO 70
C
   99 WRITE(6,200) A
  200 FORMAT(1H0,'ERROR FOR A =',8I4)
C
   70 CONTINUE
      SUM=DSUM
      RETURN
      END
```

```
C TITLE          MAIN
C
C AUTHOR         N. MYERBERG
C
C
C SPONSOR        L. SCHUCHMAN
C DATE           11-15-67
C
C PURPOSE        TO COMPUTE UPPER AND LOWER BOUNDS FOR PROBABILITY OF
C                SYMBOL ERROR FOR FOUR LEVEL QUANTIZATION.
C
C METHOD         THE ERFC FUNCTION IS USED TO COMPUTE P,A,AND B, WHICH ARE
C                USED IN COMPUTING THE BOUNDS FOR PROBABILITY OF ERROR
C
C INPUT          THROUGH NAMELIST
C                E0          FIRST VALUE FOR E/NO
C                EINC        INCREMENT FOR E/NO
C                EEND        LAST VALUE FOR E/NO
C                DEL0        FIRST VALUE FOR DELTA
C                DELINC      INCREMENT FOR DELTA
C                DELEND      FINAL VALUE FOR DELTA
C                N          ORDER OF SYMBOL ALPHABET
C
C                THROUGH READ STATEMENT
C                S           SIGNAL VECTORS (S-VECTORS)
C
C OUTPUT         E           E/NO USED IN FOLLOWING COMPUTATIONS
C                DELTA       VALUE OF DELTA USED IN COMPUTING A AND B
C                P           P=ERFC(SQRT(E/NO))/2
C                            P REPRESENTS PROBABILITY OF BINARY DIGIT ERROR
C                F           F=(1+P)/(2+A+B)
C                A           A=ERFC(SQRT(E/NO)*(1+DELTA/2))/(2*P)
C                B           B=1-ERFC(SQRT(E/NO)*(1-DELTA/2))/(2*(1-P))
C                            +P/(1-P)
C                PU          UPPER BOUND FOR PROBABILITY OF ERROR
C                PL          LOWER BOUND FOR PROBABILITY OF ERROR
C
C SUBR. USED     ERFC        TO COMPUTE ERFC
C                COMB        TO FIND COMBINATIONS
C                LBND        TO COMPUTE ERROR WEIGHT FACTOR
C
                 INTEGER S
                 DIMENSION S(8,8)
                 COMMON S
                 NAMELIST/INPUT/E0,EINC,EEND,DEL0,DELINC,DELEND,N
C
  100            READ(5,INPUT)
                 READ(5,104)((S(I,J),J=1,N),I=1,N)
  104            FORMAT(16I3)
C
                 NCHNG=N/4
                 E=E0
                 E1=(8./3.)*E
                 INDEX=0
C
```

```
 1          WRITE(6,101) E,E1
C
            P=ERFC(SQRT(E))/2.
C
            DELTA=DELO
C
 2          A=ERFC(SQRT(E)*(1.+DELTA/2.))/(2.*P)
            B=1.-ERFC(SQRT(E)*(1.-DELTA/2.))/(2.*(1.-P))+P/(1.-P)
            F=(1.+B)/(2.+B+A)
            WRITE(6,102) DELTA,F,A,B
C
            PP=P
            CALL LBND(N,A,B,PL,INDEX)
            PL=PL*P**NCHNG*(1.-P)**(N-NCHNG)
            PU=PL
            JSTRT=1+N/4
            DO 10 J=JSTRT,N
 10         PU=PU+COMB(N,J)*PP**J*(1.-PP)**(N-J)
 20         WRITE(6,103) PP,PU,PL
C
 4          DELTA=DELTA+DELINC
            INDEX=1
            IF(DELTA.LT.DELEND) GO TO 2
C
            E=E+EINC
            E1=(8./3.)*E
            IF(E.GT.EEND) GO TO 100
            GO TO 1
C
 101        FORMAT(1H1,43X,'CHIP E/NO =',F6.2,4X,'BIT E/NO =',F3.0,//
     .)
 102        FORMAT(1H0,'DELTA = ',1PE16.8,2X,'F =',1PE16.8,2X,'A =',1
     .PE16.8,2X,'B =',1PE16.8)
 103        FORMAT(1H ,'P =',1PE16.8,2X,'PU =',1PE16.8,2X,'PL =',1PE1
     .6.8)
            END
```