New Short Constraint Length, Rate 1/N Convolutional Codes Which Minimize Required E_b/N_o for Given Bit Error Rate

P. J. Lee
Communications Systems Research Section

Instead of using the criterion of maximum free distance, d_f, or the maximum d_f with minimizing a few first distance profiles, we searched short constraint length rate 1/N convolutional codes using a new criterion of minimizing required bit energy-to-noise density ratio, E_b/N_o, for a given value of desired bit error rate (BER), for the goodness of a code. The considered channel was binary antipodal signaling over additive white Gaussian noise and no quantization at the channel output. For the BER calculations, the transfer function bounding technique was used. Partial searches were performed using some known facts and a very useful new idea that "good codes generate good codes." That is, for a given constraint length K, good rate 1/(N + 1) codes can be found by extending the code generator matrices of good rate 1/N codes. The code search results are tabulated for $3 \leqslant K \leqslant 7$ and $2 \leqslant N \leqslant 8$. For many pairs of K and N, the new codes are shown to save 0.1 to 0.4 dB in the required E_b/N_o compared to previously reported codes. Additionally, the benefits of coding bandwidth expansion are confirmed with our new codes.

I. Introduction

The use of a short constraint length convolutional code together with Viterbi decoding has been very popular for several applications. The class of time-invariant, nonsystematic, rate $1/N$ convolutional codes has been studied much more extensively than any other class, partially due to its ease of analysis. The bit error rate (BER) at the Viterbi decoder output is well bounded by the well known transfer function bound (Refs. 1 and 2):

$$\mathrm{BER} \leqslant c_o \cdot \frac{\partial}{\partial Z} T(D, Z) \bigg|_{D=D_o,\, Z=1} \tag{1}$$

where c_o and transfer function $T(D, Z)$ depend on the code and the type of channel used. D_o, which has a value between 0 and 1, is the union-Bhattacharyya distance of the coding channel (everything inside the encoder-decoder pair). The right hand side of Eq. (1) is often represented as a series expansion (Refs. 1 and 2) as

$$\mathrm{BER} \leqslant c_o \cdot \sum_{i=d_f}^{\infty} a_i \cdot D_o^i \tag{2}$$

where d_f is the free distance of the code.

A large number of good rate $1/N$ convolutional codes have been found and reported (Refs. 3, 4, 5, etc.). Up to now, all the researchers have used the maximum d_f criterion or the criterion of maximum d_f together with minimizing the first few a_i's, for determining goodness of a code in their code search procedures. These criteria are very valuable, if the values of operating D_o's are much smaller than 1, since in such cases Eq. (2) can be well approximated by the first few terms. Hence, we may consider the previously reported codes to be good when the required BER is extremely small. However when the required BER is in the moderate range of 10^{-2} to 10^{-6}, such criteria may not be good since much more than the first few times are required for a good approximation to Eq. (2).

For uses with systems which require BER in the moderate range, we searched for good rate $1/N$ convolutional codes by using a new criterion of minimizing the bit energy to noise density ratio, E_b/N_o, required for a desired value of BER. For the evaluation of BER, we directly used the transfer function bound, Eq. (1). In all cases considered we assumed the use of binary antipodal signaling over the additive white Gaussian noise (AWGN) channel with no quantization at the channel output. For the desired values of BER, we picked 10^{-3} and 10^{-6}.

In the next section, the code structures and the transfer function bounding techniques are briefly reviewed for familization with our notation. Then the partial code searching techniques used in this study are presented, where the very important idea that "good codes generate good codes" is introduced and explained. The code search results are given for $3 \leqslant K \leqslant 7$ and $2 \leqslant N \leqslant 8$ and are compared with previously reported codes. As an example, for the $K = 7$ and $N = 4$ case, our new code can save 0.4 dB in required E_b/N_o compared to any previously reported code when the desired value of BER is 10^{-6}. We also confirm the benefits of coding bandwidth expansion with our new codes.

II. Preliminaries and Notations

A typical nonsystematic, constraint length K, rate $1/N$ convolutional encoder is shown in Fig. 1. The code connection box is often represented by an $N \times K$ binary matrix $\mathbf{G}$, which is called code generator matrix. For a given pair of K and N, this code generator matrix $\mathbf{G}$ determines the performance. Let

$$G(n) = (\mathbf{G}(n,1), \ldots, \mathbf{G}(n,k), \ldots, \mathbf{G}(n,K)), \quad n = 1, 2, \ldots, N \tag{3}$$

Notice that in many reports (e.g., Refs. 4, 5), the code generator G is represented by $(G(i), \ldots, G(n), \ldots, G(N))$ and $G(n)$'s are in octal forms. We call this "a regular representation" of the code generator. For later use, we also define

$$g(k) = (\mathbf{G}(i,k), \ldots, \mathbf{G}(n,k), \ldots, \mathbf{G}(N,k)), \quad k = 1, 2, \ldots, K \tag{4}$$

The nth bit in tth output vector y_n^t (see Fig. 1) for $n = 1, 2, \ldots, N$ and $t = 1, 2, \ldots$ is given by

$$y_n^t = \sum_{k=1}^{K} \mathbf{G}(n,k) \cdot x^{t-k+1} \tag{5}$$

where Σ is a mod-2 summation, $x^t \in \{0,1\}$, $t = 1, 2, \ldots$ and $x^t = 0$ for $t < 1$ by convention. The "present state" at time t, S^t, is defined as

$$S^t = (x^{t-K+1}, \ldots, x^{t-1}) \tag{6}$$

Notice that there are 2^{K-1} (= M) distinct states for any value of N.

To find code transfer functions, one often uses state diagrams where nodes represent states and directed branches represent state transitions. The metric value on the directed branch from a state to another state (or to itself), assuming the existence of such a transition, is the product of D to the power of the Hamming weight of the output vector and Z raised to the Hamming weight of the input bit, when a binary input channel is used.

As an illustration, a $K = 3$, $N = 2$ convolutional encoder, with one of the best code connections, is depicted in Fig. 2. For this code,

$$G = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \quad G(1) = (1,1,1),\ G(2) = (1,0,1),$$

$$g(1) = (1,1), \quad g(2) = (1,0),$$

$$g(3) = (1,1)$$

and the regular representation of this code generator is (7,5). The number of states, M, is 4, and its state diagram is shown in Fig. 3.

The transfer function $T(D, Z)$ can be represented (Refs. 2 and 6) as

$$T(D, Z) = B \cdot (I - A)^{-1} \cdot C \tag{7}$$

where I is the $(M-1) \times (M-1)$ unit matrix. The $(M-1) \times (M-1)$ matrix A, the $(M-1)$-dimensional row vector B, and the $(M-1)$-dimensional column vector C can be obtained from the state diagram as

$$\left.\begin{aligned} A(i,j) &= \text{metric value on the branch from state } i \text{ to state } j \text{, if there is such a transition} \\ &= 0, \text{ otherwise} \\ B(j) &= \text{metric value on the branch from state 0 to state } j \text{, if there is such a transition} \\ &= 0, \text{ otherwise} \\ C(i) &= \text{metric value on the branch from state } i \text{ to state 0, if there is such a transition} \\ &= 0, \text{ otherwise} \end{aligned}\right\} \tag{8}$$

where neither state i nor state j is the all zero state (state 0). For Eq. (1), we need (Refs. 2 and 6)

$$\frac{\partial}{\partial Z} T(D, Z) = \frac{\partial B}{\partial Z} \cdot (I - A)^{-1} \cdot C$$

$$+ B \cdot (I - A)^{-1} \cdot \frac{\partial A}{\partial Z} \cdot (I - A)^{-1} \cdot C \tag{9}$$

For the previous example,

$$A = \begin{bmatrix} 0 & D & DZ \\ Z & 0 & 0 \\ 0 & D & DZ \end{bmatrix}, \quad B = [D^2 Z \ \ 0 \ \ 0], \quad C = \begin{bmatrix} 0 \\ D^2 \\ 0 \end{bmatrix}$$

$$(I - A)^{-1} = \frac{1}{1 - 2DZ} \begin{bmatrix} 1 - DZ & D & DZ \\ Z(1 - DZ) & 1 - DZ & DZ^2 \\ DZ & D & 1 - DZ \end{bmatrix}$$

$$T(D, Z) = \frac{D^5 Z}{(1 - 2DZ)}$$

and

$$\frac{\partial}{\partial Z} T(D, Z) = \frac{D^5}{(1 - 2DZ)^2}$$

The purpose of this report is to find good codes which minimize the required E_b/N_o for desired BER = 10^{-3} or 10^{-6}, assuming the use of binary antipodal signaling over an AWGN channel and no channel output quantization. For this special coding channel, D_o and c_o in Eq. (1) are given by (Refs. 1, 2)

$$D_o = \exp(-E_s/N_o) \tag{10}$$

and

$$c_o = Q(2 \cdot d_f \cdot E_s/N_o) \exp(d_f \cdot E_s/N_o) \tag{11}$$

where N_o is the one-sided noise power spectral density and E_s is the received signal energy per channel bit,

$$E_s = r \cdot E_b \tag{12}$$

($= E_b/N$, in our cases) and

$$Q(w) = \int_w^\infty \exp(-t^2/2) \cdot dt/\sqrt{2\pi} \tag{13}$$

In the next section the techniques used for the partial code searches are described.

III. Code Searching Techniques

For rate $1/N$ convolutional codes of constraint length K, the number of possible code generators is $2^{K \cdot N}$. For example, for $K = 6$ and $N = 4$, the number is 2^{24} which is over ten million. Furthermore, because of our new criterion, to test the goodness of code generators, we have to perform matrix inversions which require considerable amounts of computing time. These make exhaustive searches prohibitively difficult except for cases of very small K and/or very small N. For moderate sizes of K and N, partial searches are necessary. In order to find good codes with partial searches, some techniques are required for reducing the code generator space effectively.

First, note that changes in the *orders* of $G(n)$'s will not change the state diagram at all (Fact 1) since the Hamming weights of the output vectors do not depend on the orders of their elements. When we use this fact, the required search space for code generators reduces by roughly a factor of N factorial. Also observe that *reversing the order* of the $g(k)$'s gives the same performance (Fact 2) since this is equivalent to just redefining the state in reverse order, i.e., $S^t = (x^{t-1}, \ldots, x^{t-K+1})$, instead of Eq. (6). This allows one to reduce the search space by roughly another factor of two. Notice that we can use these facts for reducing the search space with no loss in the chance of finding the best code. There are some other known facts on equivalence relations between code generators. However, in our partial code searches, only Facts 1 and 2 were used since the others are much more difficult to employ.

Recently, another useful observation was made in Ref. 7. That is, every known good convolutional code satisfies the following condition (in Ref. 7, it is called "flow conservation"); for each state in the state diagram, summation of the exponents of D in all incoming branches must be equal to sum of those in all outgoing branches. In Fig. 3, we can easily check that the code satisfies this condition. We noticed that this condition is automatically satisfied in our cases of rate $1/N$ if

$$g(i) = g(K) = (1, 1, \ldots, 1) \tag{14}$$

Notice that every reported good code in this class also satisfies Eq. (14). Hence we restricted our code searches to only those codes which satisfy Eq. (14). With this restriction, there may be some possibility of losing the optimum code. However, the use of Eq. (14) further reduces the required size of code search space by roughly a factor of 2^{2N}.

For a given pair of K and N (at least one of them is small), we established the code space to be searched using Eq. (14) and Facts 1 and 2. Then catastrophic codes were deleted, as were codes for which d_f was too small. We considered the value of d_f to be too small if

$$d_f < d_{fm} - \lceil (K \cdot N)/10 \rceil \tag{15}$$

where d_{fm} is the maximum free distance of $(K, 1/N)$ convolutional codes and $\lceil x \rceil$ is the smallest integer that is greater than or equal to x. For an example, consider the $K = 4$ and $N = 3$ case. The size of the reduced code space is 13 by using Eq. (14) and Facts 1 and 2, while the size of the original space is 4096 ($= 2^{4 \cdot 3} = 2^{12}$). After deleting 5 catastrophic codes, the following are the code generators to be considered further, in regular representations with a natural ordering of the largest element to the left:

(17,17,15), (17,15,15), (17,15,13), (17,15,11),

(15,15,13), (15,15,11), (15,13,11), and (15,11,11)

In this case, d_{fm} is 10. Hence (15,11,11) is deleted since its d_f is 7.

For the pair of K and N, we estimated two values of E_b/N_o at which we believed the best code(s) could achieve the desired values of BER. Then for each of the remaining codes, we calculated the values of BER using Eq. (1) at those two values of E_b/N_o. Then we listed the codes with an ordering of the best one to the top. We consider a code to be better than another if the sum of the two common logarithmic values of calculated BER's is the smaller. Such a listing is illustrated in Table 1 for the above example of the $K = 4$ and $N = 3$ case.

By examining these listings for several values of N with a given (small) K, we made a very useful observation that *good codes generate good codes*. That is, for a given K, we can find good rate $1/(N + 1)$ codes by extending the code generator of good rate $1/N$ codes. For example, from the best (4,1/2) code, (17,15), we can obtain (17,17,15), (17,15,15), (17,15,13) and (17,15,11). Notice that all of these (4,1/3) codes are shown in Table 1. For more insight on this idea, still for $K = 4$, some of the upper parts of the listings for $N = 2$ to 5 are shown in Fig. 4, with lines indicating that the left codes (with smaller N) generate the right codes. We limited the number of listings in Fig. 4 so as not to complicate the figure. Since good codes are generated by good codes, we do not have to use all the rate $1/N$ codes for the generation of good rate $1/(N + 1)$ codes. For the example of $K = 4$, the 5 best rate 1/3 codes in Table 1 are generated by the 2 best rate 1/2 codes. Also, the 8 best rate 1/4 codes are obtained from the 4 best rate 1/3 codes, and so on. Hence when we use this idea, we can reduce the size of the code space very effectively.

This idea that "good codes generate good codes" can be supported by the following fact: If a code B is generated by a noncatastrophic code A with free distance d_{fA}, then code B is also noncatastrophic and its free distance, d_{fB}, is greater than d_{fA} by at least 2. The reason for this fact is as follows: The value of the exponent of D on a branch in the state diagram for code B is always greater than or equal to the value of the exponent of D on the same branch in the state diagram for code A. Hence the code B cannot be catastrophic, since the code A is noncatastrophic. Additionally, the values of the exponents of D on the departing branch from state 0 and on the incoming branch to state 0 for code B are greater than those for code A by one [see Eq. (14) and recall that the rate of code B is $1/(N + 1)$ while that of code A is $1/N$]. Therefore $d_{fB} - d_{fA}$ is always greater than or equal to 2.

Before presenting our code search results, we mention that we used the following approximation for the matrix inversion in Eq. (9);

$$(I-A)^{-1} = \sum_{m=0}^{\infty} A^m \approx \sum_{m=0}^{\nu} A^m \qquad (16)$$

and we picked the stopping number ν such that

$$\sum_i \sum_j A^m(i,j) < 10^{-5} \text{ for any } m \geqslant \nu \qquad (17)$$

Notice that the left hand side of Eq. (17) approaches 0 monotonically as m increases. With the approximation of Eqs. (16) and (17), we have 5 to 7 digits of accuracy for most values of E_b/N_o of interest.

IV. Results, Discussion, and Conclusions

Table 2 summarizes our code search results which give the best codes in the sense of minimizing the required E_b/N_o for desired BER = 10^{-6} and 10^{-3} (among the searched codes). Also, previously reported codes (in Refs. 3, 4 and 5) are compared. Note that some of the previously reported codes are shown to be the best with our criterion as well. In particular, the codes reported in Ref. 3 are very good since only $N = 2$ and 3 cases were considered and the criterion of maximum d_f with minimizing the first few a_i's in Eq. (2) was used. Note also that, as the code rate gets smaller (or N gets larger), the importance of maximum d_f diminishes, since the value of D_o gets larger for a given value of E_b/N_o [see Eqs. (10) and (12)]. An interesting point is that the (7,1/3) code reported in Ref. 3 happens to be the best for the desired BER = 10^{-6}, despite the fact that the larger d_f codes were accidently overlooked (as noted in Ref. 4). In Fig. 5, BER versus E_b/N_o plots are given for the $K = 7$ cases.

Using our new codes it is possible to save up to 0.4 dB in required E_b/N_o compared to previously reported codes [see the (7,1/4) case]. To see the gains visually, we plot the required E_b/N_o for BER = 10^{-3} and 10^{-6} versus N curves in Fig. 6. Note that for our new codes, the required E_b/N_o becomes smaller as the code rate gets smaller (Fact A) for a given (large) value of K. This fact A is known as the gain due to coding bandwidth expansion. That is, we can save in required signal energy at the expanse of using the larger bandwidth associated with a lower code rate (with similarly complex codes[1]). Note that for the previously found codes, this fact cannot be observed, since the codes are not necessarily optimum in the sense of minimizing the required E_b/N_o. Note also that for small values of K (3 and 4) this fact cannot be seen even with our codes. Therefore, we conjectured that the benefit of coding bandwidth expansion can be obtained only when enough complexity is allowed.

Now consider the capacity and cutoff rate for further insight into Fact A. The capacity, C, in the dimension of code rate [information bits/channel bit] of this coding channel (binary antipodal signaling over AWGN, no channel output quantization) is given by (Ref. 2)

$$C = [-(1 + \ln 2\pi)/2 - \int p(y) \cdot \ln p(y) \cdot dy]/\ln 2 \qquad (18)$$

where

$$p(y) = [\exp(-(y-b)^2/2) + \exp(-(y+b)^2/2)]/\sqrt{8\pi} \qquad (19)$$

and

$$b^2/2 = E_s/N_o \qquad (20)$$

The channel coding theorem (Ref. 2) says, with coding, arbitrary small error rate can be achieved provided that the data rate R_b [information bits/sec] is less than the channel capacity $\mathscr{C}$ [information bits/sec] of the coding channel. With dimensions of code rate, the above condition can be restated as

$$r < C \qquad (21)$$

That is, the choice of code rate must be smaller than C, in theory. Note that C is a function of E_s/N_o, and $C(E_s/N_o)$ is a monotonically increasing function of its argument. Hence its inverse function exists. Therefore Eq. (21) can be rewritten as

$$E_s/N_o > C^{-1}(r) \qquad (22)$$

With Eq. (12),

$$E_b/N_o > C^{-1}(r)/r \qquad (23)$$

We call the right hand side of Eq. (23) "the required E_b/N_o to achieve capacity." These values were found and plotted in Fig. 7 as a function of coding bandwidth expansion factor (or 1/code rate r). Also the cutoff rate of the coding channel R_o [information bits/channel bit] is given by (Ref. 2)

$$R_o = 1 - \log_2(1 + \exp(-E_s/N_o)) \qquad (24)$$

[1]The complexity of (K, $1/N$) convolutional code increases as K increases and also as N increases. But since K is a much more important factor for complexity than N, we will consider the codes with the same K to have similar complexities for any N.

It is often said (e.g., Ref. 8) that a small enough error rate can be achieved, *with practical coding*, provided that R_b is less than the cutoff rate $\mathscr{R}_o$ in [information bits/sec]. Or, equivalently,

$$r < R_o\,(E_s/N_o) \tag{25}$$

This can likewise be inverted to yield

$$E_b/N_o > R_o^{-1}\,(r)/r = -\frac{1}{r}\cdot \ln\,[2^{(1-r)} - 1] \tag{26}$$

We call the right hand side of Eq. (26) "the required E_b/N_o to achieve cutoff rate." These values are also plotted in Fig. 7. These theoretical curves tell us that to be in the region below the capacity curve is impossible in theory; to be in the region between the capacity and the cutoff rate curves is possible in theory but difficult in practice; to be in the region above the cutoff rate curve is practically possible. Note that from both the capacity and the cutoff rate curves, we see that lowering the code rate gives the benefit of reducing the required E_b/N_o, which is the theoretical view of Fact A.

In the same figure, we have plotted the performances of our new codes for $K = 5$, 6 and 7 (from Fig. 6). From comparisons of the slopes of the curves for BER = 10^{-3} with those for BER = 10^{-6} cases, we conclude that the benefit of coding bandwidth expansion is greater when the desired BER is larger. Also, increasing the system complexity (i.e., increasing K) provides more gain when the desired BER is smaller, as seen by the larger spacing between the curves for the BER = 10^{-6} cases compared to the BER = 10^{-3} cases. Note also the similarity between curves for our new codes with those derived from theory (especially the cutoff rate curve).

In conclusion, we have found good (K, $1/N$) codes which minimize the required E_b/N_o for the desired BER = 10^{-3} and 10^{-6} for $3 \leqslant K \leqslant 7$ and $2 \leqslant N \leqslant 8$. For the partial searches of codes, we used some known facts together with another very useful idea that "good codes generate good codes." For many pairs of K and N, our new codes are shown to save 0.1 to 0.4 dB in required E_b/N_o for moderate values of required BER, compares to the previously reported codes. Also, we confirmed the benefits of coding bandwidth expansion with our new codes, whereas the previously reported codes did not uniformly confirm this property.

References

1. A. J. Viterbi, "Convolutional Codes and Their Performance in Communication Systems," *IEEE Trans. Comm. Tech.*, Vol. COM-19, pp. 751-772, October 1971.
2. A. J. Viterbi and J. K. Omura, *Principles of Digital Communications and Coding*, McGraw-Hill Book Co., New York, NY, 1979.
3. J. P. Odenwalder, *Optimal Decoding of Convolutional Codes*, Ph.D. Thesis, Dept. Sys. Sci., Sch Eng. Appl. Sci., Univ. Calif., L.A., Los Angeles, CA, 1970.
4. K. L. Larsen, "Short Convolutional Codes with Maximal Free Distance for Rates 1/2, 1/3, and 1/4," *IEEE Trans. Info. Theory*, Vol. IT-19, pp. 371-372, May 1973.
5. D. G. Daut, J. W. Modestino, and L. D. Wismer, "New Short Constraint Length Convolutional Code Constructions for Selected Rational Rates," *IEEE Trans. Info. Theory*, Vol. IT-28, pp. 794-800, Sept. 1982.
6. J. K. Omura and M. K. Simon, *Modulation/Demodulation Techniques for Satellite Communications, Part IV: Appendices*, JPL Publication 81-73, Jet Propulsion Lab., Pasadena, CA, Nov. 1981.
7. R. Polazzo, *Analysis of Periodic Nonlinear Trellis Codes*, Ph.D. Thesis, Dept. Sys. Sci., Sch. Eng. Appl. Sci., Univ. Calif., L.A., Los Angeles, CA, 1984.
8. J. L. Massey, "Coding and Modulation in Digital Communications," *Proc. Intern. Zurich Seminar on Digital Comm.*, Zurich, Switzerland, March 1974.

Table 1. Examples of listing of code search results for the $K = 4$ and $N = 3$ case

G in octal	d_f	$\log_{10}$ (BER) at E_b/N_o = 6.0 dB	$\log_{10}$ (BER) at E_b/N_o = 3.5 dB	Sum of the Two Values
(17, 15, 13)	10	-6.059	-3.070	-9.129
(17, 15, 11)	9	-6.008	-3.082	-9.090
(15, 13, 11)	8	-5.609	-3.014	-8.623
(17, 15, 15)	10	-5.702	-2.567	-8.289
(15, 15, 13)	9	-5.516	-2.510	-8.026
(15, 15, 11)	8	-5.422	-2.490	-7.912
(17, 17, 15)	8	-5.279	-2.488	-7.767

Table 2. Code search results

K	N	Code Generator G in Regular Representation, octal	d_f	Required E_b/N_o, dB for desired BER= 1.E-6	1.E-3	Notes
3	2	7, 5	5	6.706	4.017	1, a, b
3	3	7, 7, 5	8	6.736	4.140	1, a, b
3	4	7, 7, 7, 5	10	6.938	4.354	2
3	4	7, 7, 5, 5	10	6.706	4.017	4, a, b
3	5	7, 7, 7, 5, 5	13	6.669	4.055	3, a, b
3	6	7, 7, 7, 7, 5, 5	16	6.736	4.140	3
3	6	7, 7, 7, 5, 5, 5	15	6.706	4.017	4, a, b
3	7	7, 7, 7, 7, 5, 5, 5	18	6.664	4.035	3, a, b
3	8	7, 7, 7, 7, 7, 5, 5, 5	21	6.685	4.080	3, a
3	8	7, 7, 7, 7, 5, 5, 5, 5	20	6.706	4.017	4, b
4	2	17, 15	6	6.180	3.735	1, a, b
4	3	17, 15, 13	10	5.958	3.437	1, a
4	3	17, 15, 11	9	5.994	3.427	4, b
4	4	17, 15, 15, 13	13	6.004	3.511	2
4	4	17, 15, 13, 11	12	5.906	3.286	4, a, b
4	5	17, 17, 15, 15, 13	16	5.991	3.483	3
4	5	17, 15, 15, 13, 11	15	5.909	3.332	4, a, b
4	6	17, 17, 15, 15, 13, 13	20	5.958	3.437	3
4	6	17, 17, 15, 15, 13, 11	19	5.865	3.325	4, a, b
4	7	17, 17, 15, 15, 15, 13, 13	23	5.974	3.462	3
4	7	17, 17, 15, 15, 13, 13, 11	22	5.849	3.306	4, a, b
4	8	17, 17, 17, 15, 15, 15, 13, 13	26	5.972	3.456	3
4	8	17, 17, 15, 15, 15, 13, 13, 11	25	5.860	3.332	4, a
4	8	17, 17, 15, 15, 13, 13, 11, 11	24	5.906	3.286	4, b
5	2	35, 23	7	5.745	3.495	1, a
5	2	31, 23	6	5.845	3.430	4, b
5	3	37, 33, 25	12	5.395	3.115	1, a, b
5	4	37, 35, 33, 25	16	5.303	2.999	2
5	4	37, 35, 25, 23	15	5.298	3.000	4, a
5	4	35, 31, 27, 23	14	5.317	2.965	4, b
5	5	37, 35, 33, 27, 25	20	5.270	2.923	3
5	5	37, 35, 31, 27, 25	19	5.243	2.924	4, a
5	5	37, 35, 33, 25, 21	18	5.297	2.912	4, b
5	6	37, 35, 35, 33, 27, 25	24	5.291	2.957	3
5	6	37, 35, 33, 27, 25, 23	23	5.211	2.880	4, a
5	6	37, 35, 33, 27, 25, 21	22	5.294	2.867	4, b
5	7	37, 35, 35, 33, 27, 27, 25	28	5.286	2.955	3
5	7	37, 35, 33, 31, 27, 25, 23	26	5.211	2.845	4, a
5	7	37, 35, 33, 27, 25, 23, 21	25	5.256	2.839	4, b
5	8	37, 37, 35, 33, 33, 27, 25, 25	32	5.284	2.949	3
5	8	37, 35, 35, 33, 31, 27, 25, 23	30	5.211	2.860	4, a
5	8	37, 35, 33, 31, 27, 25, 23, 21	28	5.280	2.819	4, b

Table 2 (contd)

K	N	Code Generator **G** in Regular Representation, octal	d_f	Required E_b/N_o, dB for desired BER= 1.E–6	1.E–3	Notes
6	2	75, 53	8	5.310	3.289	1
6	2	77, 45	8	5.236	3.242	4, a
6	2	75, 57	8	5.293	3.211	4, b
6	3	75, 53, 47	13	4.918	2.900	1, a
6	3	75, 67, 41	12	5.034	2.854	4, b
6	4	75, 71, 67, 53	18	4.836	2.747	2
6	4	77, 73, 55, 45	18	4.779	2.729	4, a
6	4	77, 73, 51, 45	17	4.807	2.719	4, b
6	5	75, 73, 71, 65, 57	22	4.826	2.680	3
6	5	77, 73, 71, 55, 45	22	4.742	2.660	4, a
6	5	75, 71, 65, 57, 53	22	4.753	2.645	4, b
6	6	75, 73, 65, 57, 55, 47	27	4.764	2.616	3
6	6	77, 73, 67, 55, 51, 45	26	4.694	2.591	4, a
6	6	75, 71, 65, 57, 53, 47	26	4.704	2.590	4, b
6	7	75, 75, 67, 65, 57, 53, 47	32	4.762	2.630	3
6	7	77, 73, 67, 63, 55, 51, 45	30	4.696	2.565	4, a
6	7	75, 71, 65, 57, 53, 47, 43	29	4.717	2.564	4, b
6	8	75, 73, 67, 65, 57, 57, 51, 47	36	4.728	2.599	3
6	8	77, 73, 67, 63, 57, 55, 51, 45	35	4.693	2.552	4, a
6	8	77, 73, 67, 63, 55, 51, 45, 41	32	4.772	2.541	4, b
7	2	171, 133	10	4.802	3.036	1, a
7	2	161, 133	9	4.818	3.035	4, b
7	3	175, 145, 133	15	4.599	2.706	2
7	3	171, 145, 133	14	4.489	2.672	1, a
7	3	161, 135, 107	13	4.600	2.666	4, b
7	4	163, 147, 135, 135	20	4.761	2.814	2
7	4	175, 151, 133, 117	20	4.372	2.520	4, a
7	4	173, 167, 135, 111	20	4.433	2.511	4, b
7	5	175, 147, 135, 135, 131	25	4.562	2.653	3
7	5	175, 151, 133, 127, 117	25	4.310	2.444	4, a
7	5	175, 165, 151, 133, 117	25	4.350	2.438	4, b
7	6	173, 163, 151, 137, 135, 135	30	4.394	2.499	3
7	6	175, 171, 151, 133, 127, 117	30	4.286	2.387	4, a
7	6	175, 165, 151, 137, 133, 117	30	4.307	2.385	4, b
7	7	173, 165, 147, 145, 137, 135, 135	36	4.381	2.456	3
7	7	175, 171, 155, 127, 123, 117, 113	34	4.266	2.356	4, a
7	7	175, 171, 165, 151, 133, 127, 117	35	4.282	2.353	4, b
7	8	173, 165, 153, 147, 137, 135, 135, 111	40	4.312	2.387	3
7	8	175, 171, 165, 151, 133, 127, 117, 113	39	4.251	2.327	4, a, b

Notes:
1: Found by Odenwalder (Ref. 3)
2: Found by Larsen (Ref. 4)
3: Found by Daut, et al. (Ref 5)
4: Found by the author
a: Minimizes required E_b/N_o for desired BER = 1.E–6
b: Minimizes required E_b/N_o for desired BER = 1.E–3

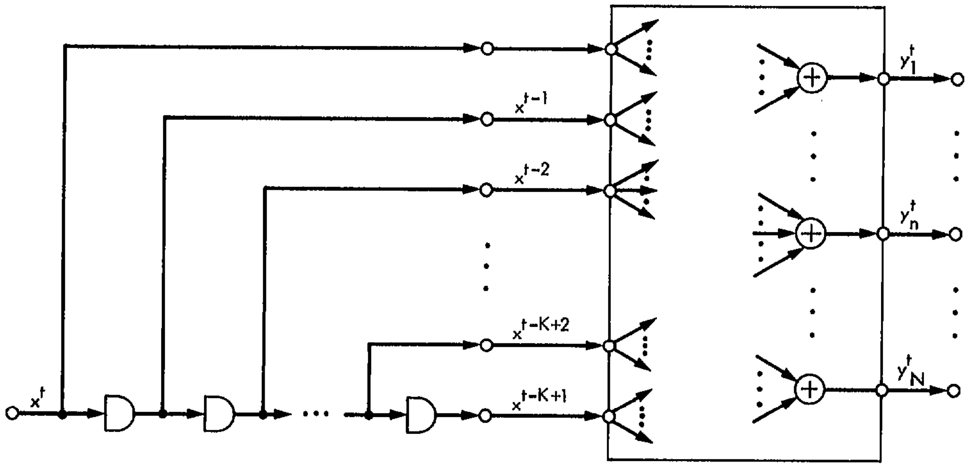

Fig. 1. A nonsystematic, constraint length K, rate $1/N$ convolutional encoder structure

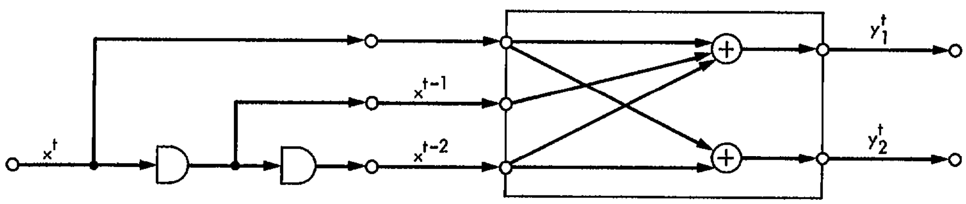

Fig. 2. A best (3,1/2) convolutional encoder

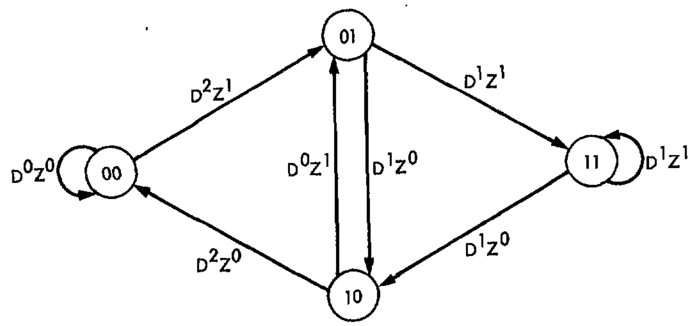

Fig. 3. The state diagram of the encoder in Fig. 2

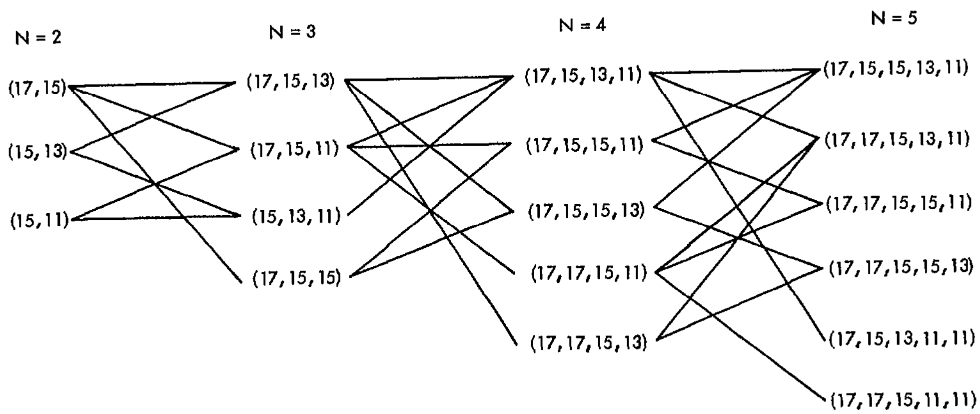

Fig. 4. An example for the idea of "good codes generate good codes," $K = 4$

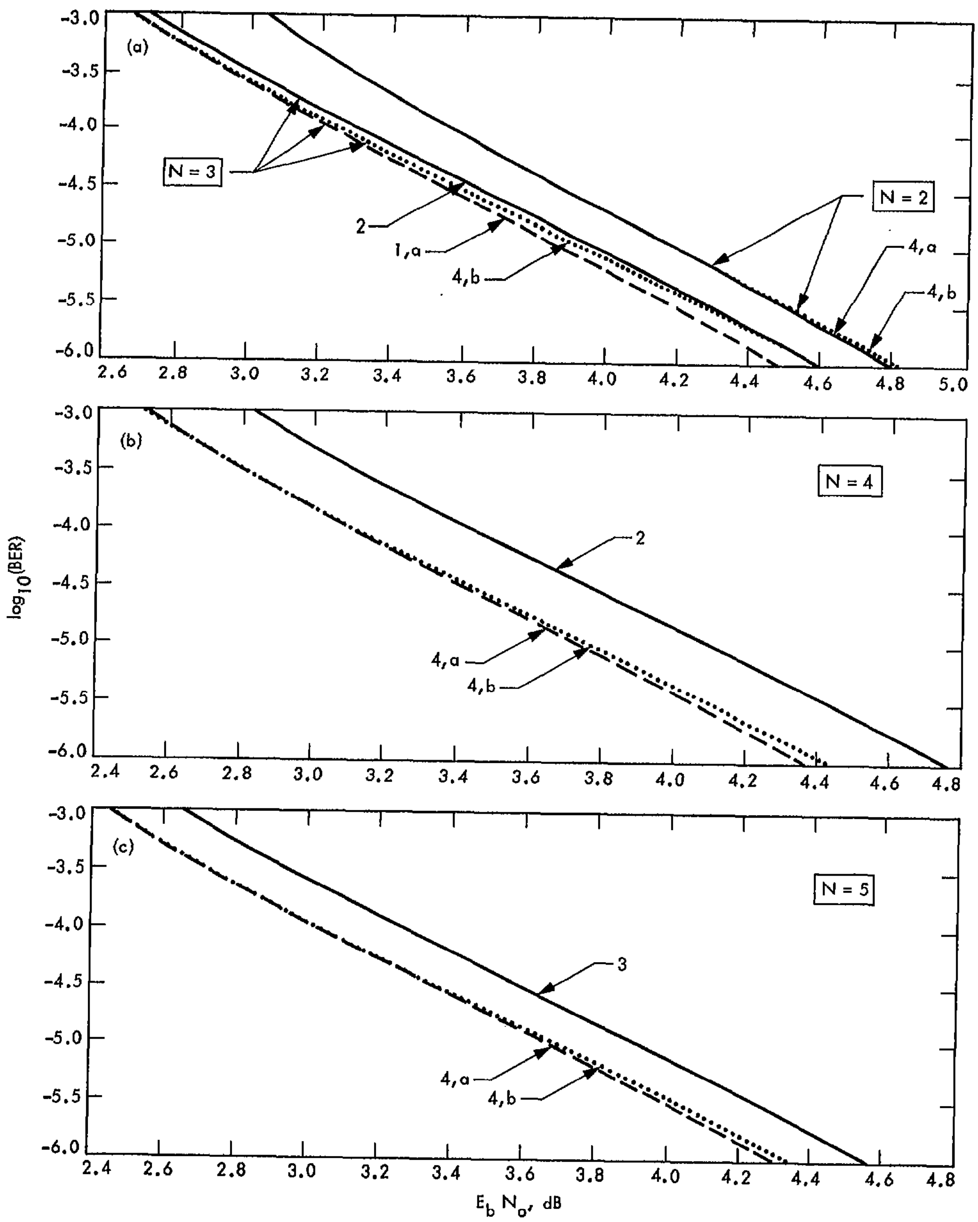

Fig. 5. BER versus E_b/N_o curves for $K = 7$ codes (see Table 2 for the notes)

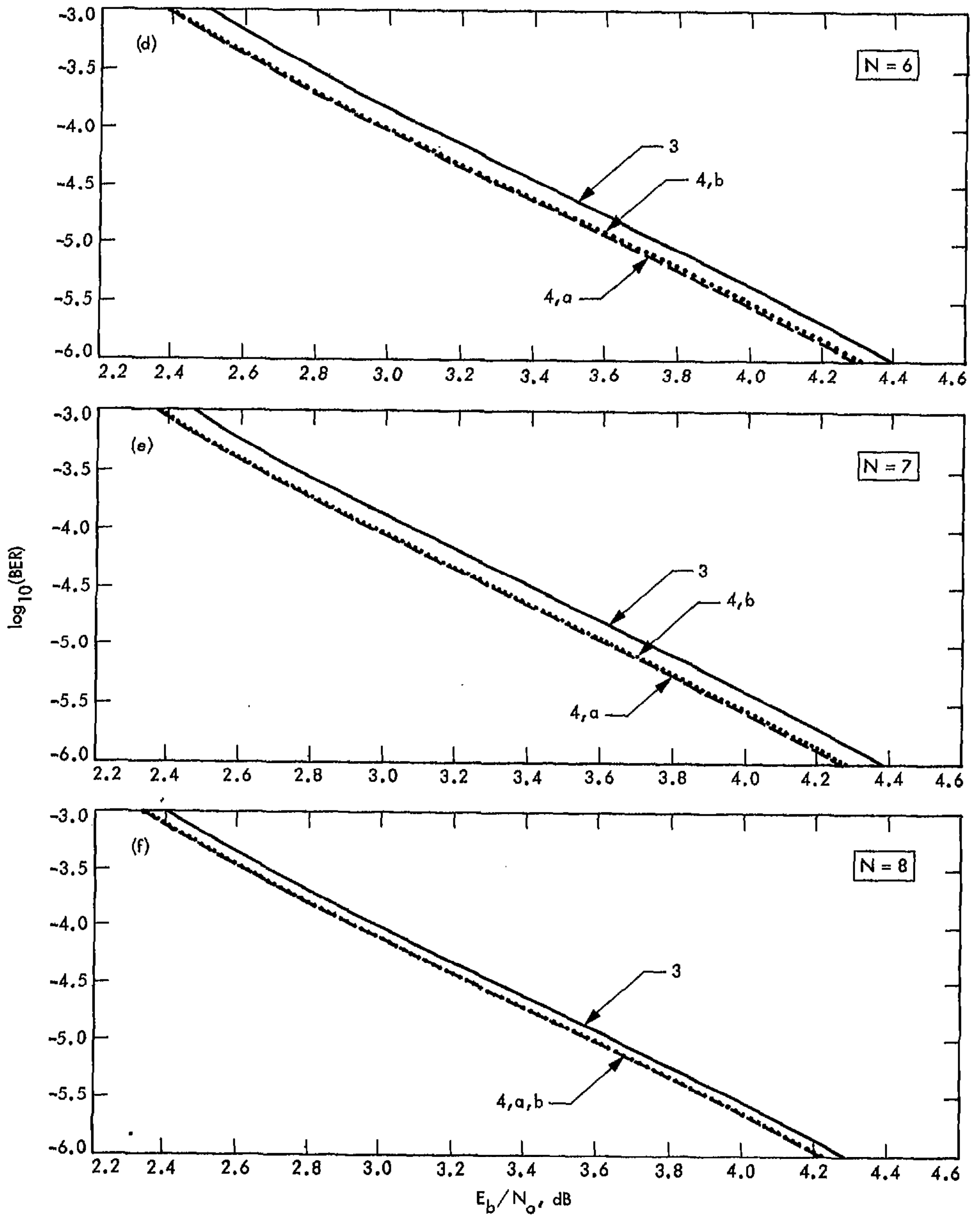
(d)
N = 6
3
4,b
4,a
(e)
N = 7
3
4,b
4,a
(f)
N = 8
3
4,a,b
$\log_{10}$(BER)
-3.0
-3.5
-4.0
-4.5
-5.0
-5.5
-6.0
2.2 2.4 2.6 2.8 3.0 3.2 3.4 3.6 3.8 4.0 4.2 4.4 4.6
E_b/N_o, dB

Fig. 5 (contd)

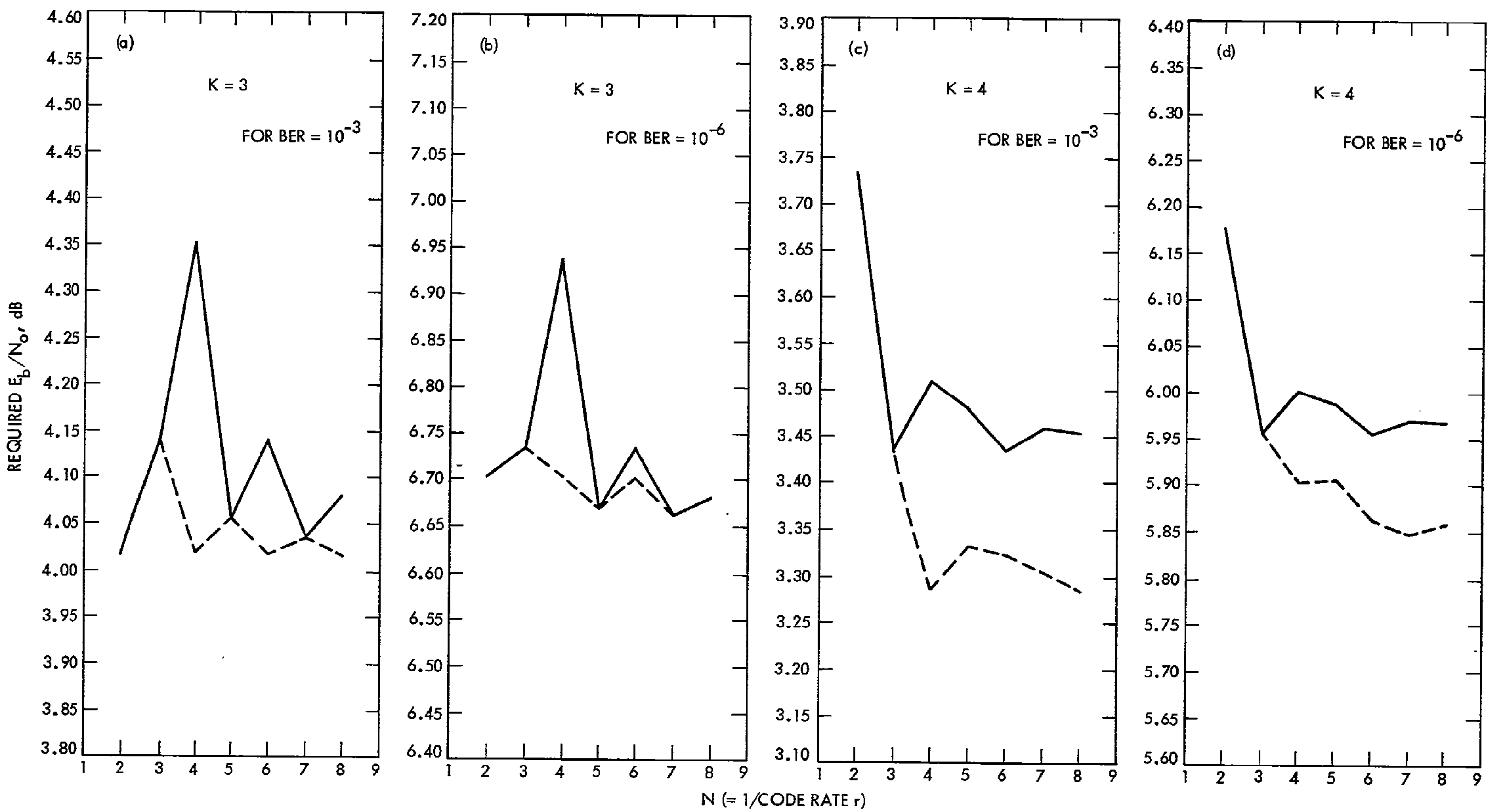

Fig. 6. Required E_b/N_o versus N (= 1/code rate r). Solid line indicates codes found previously; dashed line indicates codes found here.

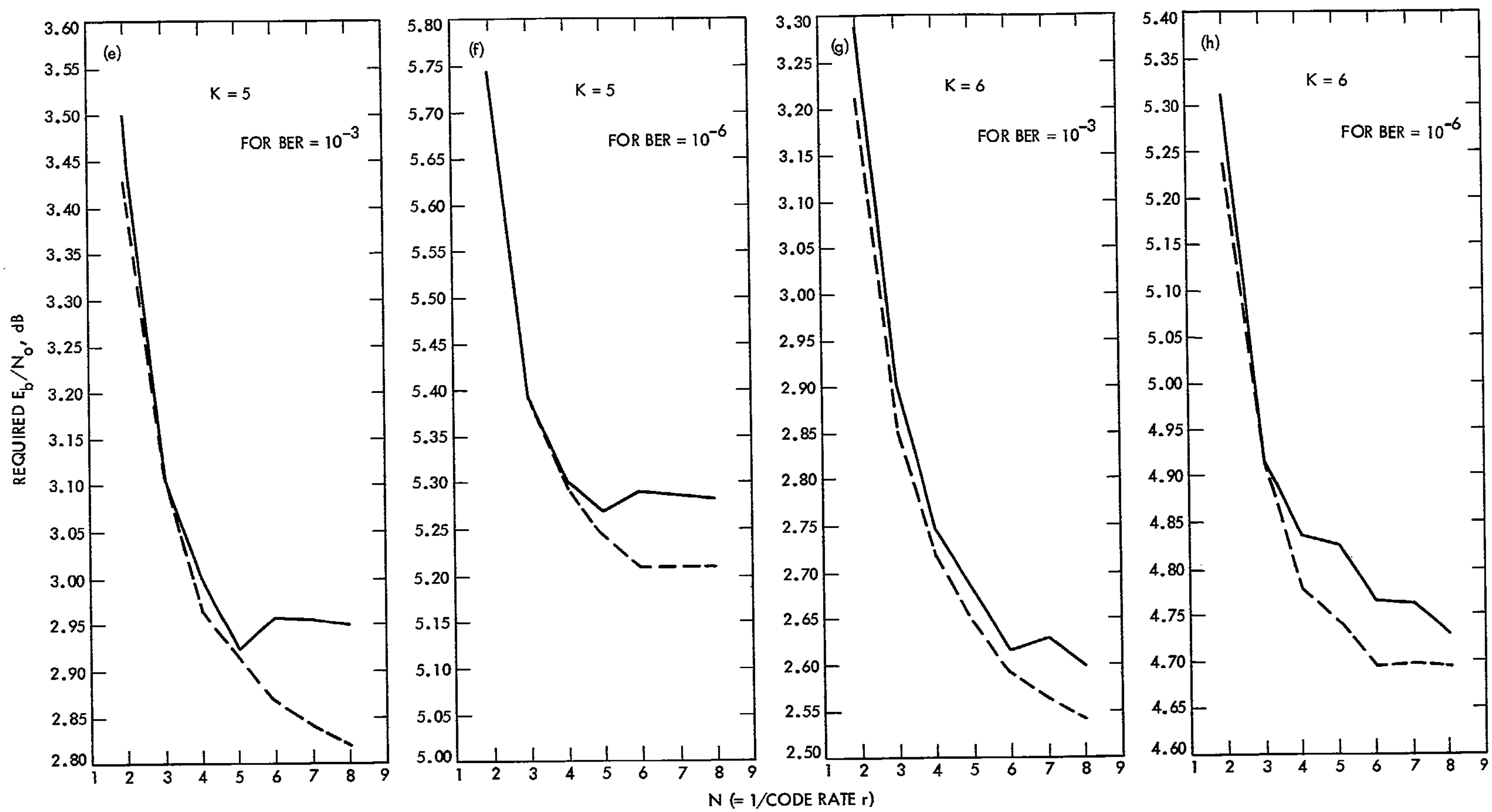
(e)
K = 5
FOR BER = 10^-3
(f)
K = 5
FOR BER = 10^-6
(g)
K = 6
FOR BER = 10^-3
(h)
K = 6
FOR BER = 10^-6
REQUIRED E_b/N_o, dB
N (= 1/CODE RATE r)

Fig. 6 (contd)

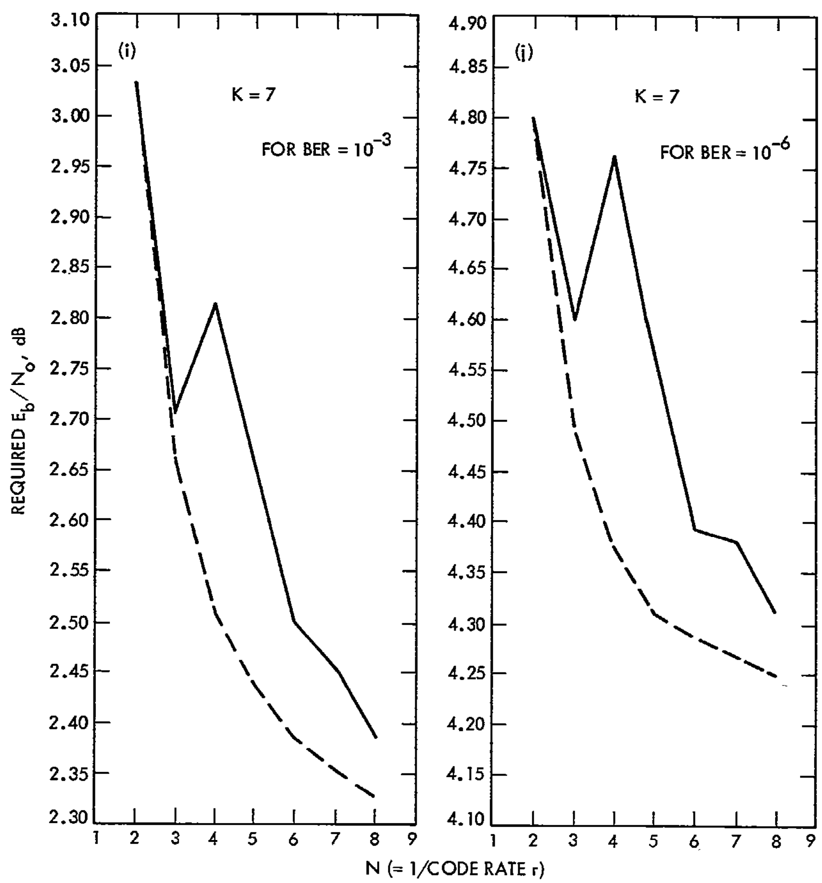

Fig. 6 (contd)

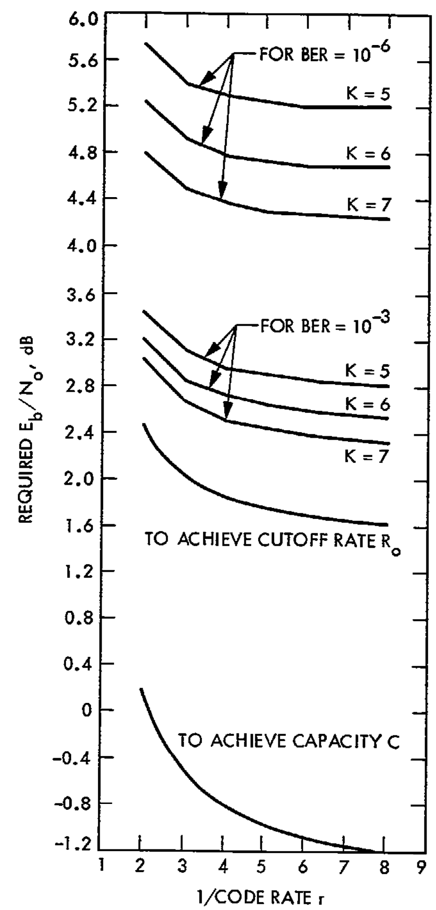

Fig. 7. Required E_b/N_o versus 1/code rate r