A Very Efficient Transfer Function Bounding Technique on Bit Error Rate for Viterbi Decoded, Rate 1/N Convolutional Codes

P. J. Lee
Communications Systems Research Section

For rate 1/N convolutional codes, a recursive algorithm for finding the transfer function bound on bit error rate (BER) at the output of a Viterbi decoder is described. This technique is very fast and requires very little storage since all the unnecessary operations are eliminated. Using this technique, we find and plot bounds on the BER performance of known codes of rate 1/2 with $K \leqslant 18$, rate 1/3 with $K \leqslant 16$, and rate 1/4 with $K \leqslant 14$. When more than one reported code with the same parameters is known, we select the code that minimizes the required signal-to-noise ratio for a desired bit error rate of 10^{-6}. This criterion of determining goodness of a code had previously been found to be more useful than the maximum free distance criterion and was used in the code search procedures of very short constraint length codes. This very efficient technique can also be used for searches of longer constraint length codes.

I. Introduction

The best decoding method for convolutional codes is maximum-likelihood (ML) decoding (often called Viterbi decoding) (Refs. 1 and 2), which is considered to be practical only for "short" constraint length codes. For longer constraint length codes sequential decoding is often employed (Ref. 2). However, due to rapidly developing hardware technologies, the length which is considered to be "short" has been increasing. Also, the bit error rate (BER) with sequential decoding may be lower bounded by the BER with ML decoding. Hence, finding the performance of Viterbi-decoded long constraint length codes is useful, even if construction of Viterbi decoders for such length codes is impractical with today's technologies.

The BER at the Viterbi decoder output is well bounded by the well-known transfer function bound (Refs. 1 and 2). For this bound, matrix inversion is required (Refs. 2 and 3). Accordingly, a large amount of computer storage and a substantial amount of computing time have been required. In this report, for rate $1/N$ convolutional codes, we present a recursive algorithm for finding the transfer function bound. By eliminating all the unnecessary computations (e.g., multiplication-by-zero, etc.), we can perform the matrix inversion with vector operations only. Hence, this technique is very fast computationally, and it furthermore requires only a very small amount of computer memory storage. After a brief review of preliminaries in the next section, the algorithm is described in Section III.

A large number of good rate $1/N$ convolutional codes have been found and reported (Refs. 4 to 7). In those code search procedures, maximum free distance (d_f) or the maximum d_f together with minimizing the first few weight spectral components (number of adversaries) were used for determining the goodness of a code. More recently, in Ref. 8, we introduced a better criterion of "minimum required signal-to-noise ratio (SNR) for a given desired BER" and used this criterion for code searches of rate $1/N$ codes with very short constraint length ($K \leqslant 7$). For searches of longer constraint length codes, this technique will be used for finding BER performance bounds.

In the last section, we give the BER performance of some rate $1/N$ codes which require minimum SNR for desired BER of 10^{-6} among codes reported in Refs. 4 to 8. (For $K \geqslant 8$, we expect that there may be better codes with our criterion.) For the calculations, we assume binary antipodal signaling over the additive white Gaussian noise (AWGN) channel with no channel output quantization. Such codes are tabulated in Table 1. With a discussion on the accuracy of the bound, we conclude that, as compared to the series expansion approximation to the transfer function bound, our method is shown to be preferable in two aspects: (1) it gives uniform accuracy and (2) it is more suitable for comparison of the codes.

II. Preliminaries

In this section a general background for the transfer function bound is briefly reviewed, mainly to define necessary notation. A typical nonsystematic, constraint length K, rate $1/N$ convolutional encoder is shown in Fig. 1. The connection box with mod-2 adders is often represented by an $N \times K$ binary matrix **G**, which is called the code generator matrix. The n-th bit in t-th output vector y_n^t (see Fig. 1) for $n = 1, 2, \ldots, N$ and $t = 1, 2, \ldots$ is:

$$y_n^t = mod\left\{\sum_{k=0}^{K-1} \mathbf{G}(n,k) \cdot x^{t-k}, 2\right\} \tag{1}$$

where $mod\{a, b\}$ is the remainder when a is divided by b, $x^t \in \{0, 1\}$ is the encoder input sequence for $t = 1,2,\ldots,$ and $x^t = 0$ for $t \leqslant 0$ by convention. The code rate r [information bits/channel bit] is $1/N$. The "present" state at time t, S^t, is defined as $S^t = (x^{t-K+1}, \ldots, x^{t-1})$, and we denote S^t by i, if

$$\sum_{k=1}^{K-1} x^{t-k}\, 2^{k-1} = i \tag{2}$$

Hence the state space is $\{0,1,2,\ldots,M-1\}$, where $M = 2^{K-1}$ is the size of the state space. Suppose S^t is i. Then by the definition of state, the next state j is given by:

$$j = x^t + mod\{2i, M/2\} \tag{3}$$

To find a transfer function, one often uses a state diagram where nodes represent states and directed branches represent state transitions. Let $\mathbf{W}(i,j)$, $i,j = 0,1,\ldots,M-1$, be the matrix representation of the branch metric on the directed branch from state i to state j. When such a transition exists, $\mathbf{W}(i,j)$ is given by the product of D to the power of the Hamming weight of the corresponding output vector and Z raised to the Hamming weight of the input bit, when a binary input channel is used. (D and Z are dummy variables. See Eq. (5).) While, $\mathbf{W}(i,j) = 0$ when there is no such transition. As an illustration, a $K = 4$, $r = 1/5$ convolutional encoder is shown in Fig. 2. For this code, **G** and **W** are given below and the state diagram is shown in Fig. 3.

$$\mathbf{G} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{W} = \left[\begin{array}{c|ccccccc} 1 & D^5Z & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & D^3 & D^2Z & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & D^2 & D^3Z & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & D^3 & D^2Z \\ D^5 & Z & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & D^2 & D^3Z & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & D^3 & D^2Z & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & D^2 & D^3Z \end{array}\right]$$

The transfer function $T(D,Z)$ can be represented by (Refs. 2 and 3)

$$T(D,Z) = \mathbf{b} \cdot (\mathbf{I} - \mathbf{A})^{-1} \cdot \mathbf{c} \tag{4}$$

where $(M-1)$-dimensional row and column vectors **b** and **c** and $(M-1) \times (M-1)$ matrix **A** are portions of the $M \times M$ matrix **W**, such that $\mathbf{b}(j) = \mathbf{W}(0,j)$, $\mathbf{c}(i) = \mathbf{W}(i,0)$, and $\mathbf{A}(i,j) = \mathbf{W}(i,j)$, where $i,j = 1,2,\ldots,M-1$. Here, **I** is the $(M-1) \times (M-1)$ identity matrix.

The BER at the output of an ML decoder is well upper-bounded by the following expression, called the "transfer function bound" (Refs. 1 to 3)

$$BER \leqslant C_o \cdot \frac{\partial}{\partial Z} T(D,Z) \bigg|_{D=D_o, Z=1} \tag{5}$$

where the coefficient C_o depends on the channel and code used, while the Bhattacharyya bound D_o depends only on the coding channel (everything between the encoder output and the decoder input, including decision rule or channel output quantization). For the AWGN channel with antipodal signaling and with no channel output quantization, C_o and D_o are given by (Refs. 1 and 2)

$$D_o = \exp(-E_s/N_o) \tag{6}$$

and

$$C_o = Q(\sqrt{2d_f E_s/N_o}) \cdot \exp(d_f E_s/N_o) \tag{7}$$

where E_s is the received signal energy per channel symbol which is related to the received signal energy per information bit E_b by, $E_s = r\,E_b$ ($r = 1/N$ in our case). N_o is the one-sided noise power spectral density,

$$Q(z) = \int_z^\infty \exp(-t^2/2) \cdot dt/\sqrt{2\pi}$$

and d_f is the free distance of the code.

III. Transfer Function Bound

When the matrix $(\mathbf{I} - \mathbf{A})$ has an inverse, the following holds:

$$(\mathbf{I} - \mathbf{A})^{-1} = \sum_{\ell=0}^{\infty} \mathbf{A}^{\ell} \tag{8}$$

or

$$T(D,Z) = \sum_{\ell=0}^{\infty} \mathbf{b} \cdot \mathbf{A}^{\ell} \cdot \mathbf{c} \tag{9}$$

Let

$$\mathbf{f}_\ell = \mathbf{b} \cdot \mathbf{A}^{\ell} \tag{10}$$

which can be found recursively as

$$\mathbf{f}_{\ell+1} = \mathbf{f}_\ell \cdot \mathbf{A},\ \ell = 0,1,2,\ldots;\ \text{with } \mathbf{f}_o \triangleq \mathbf{b} \tag{11}$$

Hence

$$T(D,Z) = \sum_{\ell=0}^{\infty} \mathbf{f}_\ell \cdot \mathbf{c} \tag{12}$$

Let

$$\mathbf{g}_\ell = \partial \mathbf{f}_\ell / \partial Z \tag{13}$$

This also can be found recursively

$$\mathbf{g}_{\ell+1} = \mathbf{g}_\ell \cdot \mathbf{A} + \mathbf{f}_\ell \cdot \mathbf{A}',\ \ell = 0,1,\ldots;\ \text{with } \mathbf{g}_o = \mathbf{b}' \tag{14}$$

where $\mathbf{A}' = \partial \mathbf{A}/\partial Z$ and $\mathbf{b}' = \partial \mathbf{b}/\partial Z$.

Since $\partial \mathbf{c}/\partial Z = \mathbf{0}$, we have

$$\frac{\partial}{\partial Z} T(D,Z)\bigg|_{D=D_o,Z=1} = \left\{\sum_{\ell=0}^{\infty} \mathbf{g}_\ell \cdot \mathbf{c}\right\}\bigg|_{D=D_o,Z=1} \tag{15}$$

Equations (4) through (15) hold for any convolutional code. That is, for any rate b/N convolutional code, we can find the transfer function bound recursively, with a proper truncation of Eq. (15), by performing vector-by-matrix multiplications.

For rate $1/N$ codes, we can further reduce the required computer storage and computational effort. Notice that the matrix $\mathbf{W}$ has only $2M$ nonzero elements in special positions (see Eq. (3) and the example). That is, for the iterations of Eqs. (11) and (14), one often performs lots of multiplication-by-zero operations. By eliminating these unnecessary operations, we can reduce the computational burden substantially. Also we do not have to waste storage for the zero entries in the matrix $\mathbf{W}$. Instead of the $M \times M$ matrix $\mathbf{W}$, we may use the following two M-dimensional vectors;

$$\mathbf{u}(j) = \mathbf{W}([j/2], j)\big|_{D=D_o,Z=1}$$

and $\qquad , j = 0,1,\ldots,M-1$

$$\mathbf{v}(j) = \mathbf{W}([j/2] + M/2, j)\big|_{D=D_o,Z=1} \tag{16}$$

where $[a]$ denotes the integer part of a. Let

$$\tilde{\mathbf{f}}_\ell = \mathbf{f}_\ell\big|_{D=D_o,Z=1} \quad \text{and} \quad \tilde{\mathbf{g}}_\ell = \mathbf{g}_\ell\big|_{D=D_o,Z=1} \tag{17}$$

Then we can iterate $\tilde{\mathbf{f}}_\ell$ and $\tilde{\mathbf{g}}_\ell$ for $\ell = 1,2,\ldots$, with $\tilde{\mathbf{f}}_o = \tilde{\mathbf{g}}_o \triangleq (\mathbf{u}(1),0,0,\ldots,0)$, as;

$$\begin{aligned} \tilde{\mathbf{f}}_{\ell+1}(1) &= \tilde{\mathbf{f}}_\ell(M/2) \cdot \mathbf{v}(1) \\ \tilde{\mathbf{g}}_{\ell+1}(1) &= \tilde{\mathbf{g}}_\ell(M/2) \cdot \mathbf{v}(1) + \tilde{\mathbf{f}}_{\ell+1}(1) \end{aligned} \tag{18}$$

and for $j = 2,3, \ldots, M-1$,

$$\widetilde{\mathbf{f}}_{\ell+1}(j) = \widetilde{\mathbf{f}}_{\ell}([j/2]) \cdot \mathbf{u}(j) + \widetilde{\mathbf{f}}_{\ell}([j/2] + M/2) \cdot \mathbf{v}(j)$$

$$\widetilde{\mathbf{g}}_{\ell+1}(j) = \widetilde{\mathbf{g}}_{\ell}([j/2]) \cdot \mathbf{u}(j) + \widetilde{\mathbf{g}}_{\ell}([j/2] + M/2) \cdot \mathbf{v}(j) + \widetilde{\mathbf{f}}_{\ell+1}(j) \cdot \mathrm{mod}\,(j,2) \tag{19}$$

Finally, for the transfer function bound, we have a recursive solution which requires only vector operations;

$$\frac{\partial}{\partial Z} T(D,Z)\bigg|_{D=D_o, Z=1} = \left\{\sum_{\ell=0}^{\infty} \widetilde{\mathbf{g}}_{\ell}(M/2)\right\} \cdot \mathbf{v}(0) \tag{20}$$

Note that we need to truncate (20) at some depth. One may choose the stopping number L such that

$$\widetilde{\mathbf{g}}_L(M/2) < 10^{-6} \cdot \left\{\sum_{\ell=0}^{L} \widetilde{\mathbf{g}}_{\ell}(M/2)\right\} \tag{21}$$

This gives 4 or more digits of accuracy for most cases of interest. Also notice that if $\mathbf{G}(n,K) = 1$ for all $n = 1,2, \ldots, N$, then we need only one of $\mathbf{u}$ or $\mathbf{v}$, since $\mathbf{u}(j) \cdot \mathbf{v}(j) = D^N$ for all j. (The same is true if $\mathbf{G}(n,1)$ are all n, by redefining the states in reverse order.)

IV. Applications, Discussion, and Conclusions

In the previous section we presented a technique for finding the transfer function bound using only vectors. Hence we can apply this technique for rather long constraint length codes very efficiently. This technique was used to compute the performance of reported codes in Refs. 4 through 8 for $r = 1/2$ codes with $K \leqslant 18$, $r = 1/3$ codes with $K \leqslant 16$, and $r = 1/4$ codes with $K \leqslant 14$. With a given K and r, we picked the best code, using a criterion of minimizing required E_b/N_o for a given desired BER of 10^{-6}. Such codes are tabulated in Table 1 and their performances are shown in Figs. 4 through 6. The vertical lines in those figures are the computational cutoff rate limit for the corresponding code rate ($E_b/N_o = -1/r \cdot \ln\{2^{1-r} - 1\}$). Note that the transfer function bound becomes loose as the operating SNR approaches the cutoff rate limit.

Notice that the code which is the best by our criterion may not have the maximum free distance. The transfer function bound itself is often represented by a series expansion as (Refs. 1 and 2)

$$\frac{\partial}{\partial Z} T(D,Z)\bigg|_{D=D_o, Z=1} = \sum_{i=d_f}^{\infty} a_i D_o^i \tag{22}$$

The first term of Eq. (22) may be used to determine the code performance if the value of operating D_o is extremely small. However, the value of operating D_o may not be so small in practice. Hence, rather than a single term approximation, a several-term truncated version of Eq. (22) is often used as an approximation to the transfer function bound. The coefficients a_i's (often called "weight spectra" or "number of bit errors in the adversaries") are needed for such approximations. For example, in Ref. 4, Odenwalder found the first 8 a_i's for his own codes, while in Ref. 9 Conan found the first 18 a_i's for the codes in Refs. 4 and 5. The technique of reducing required storage and computational effort described in the previous section can be applied for finding the a_i's also. However, note that the number of required terms for a good approximation varies with the value of operating D_o and the a_i's themselves. That is, a larger number of terms is required when the operating signal-to-noise ratio is small, when the code rate is low, and/or when the values of a_i's are large. As an illustration, in Fig. 7, 8-term (dotted lines) and 18-term (dashed lines) approximations are compared with our results (solid lines) for six codes considered in Ref. 9 with $r = 1/2$, $1/3$, and $1/4$, and $K = 7$ and 11. Notice that our method of finding the transfer function bound with the truncation rule (21) provides "uniform accuracy" for all cases considered.

Finding a_i's and using them for the performance evaluation of a code is very useful, if enough terms are provided for an accurate approximation. However, comparing two codes using those a_i's may not be practical, since vector comparison is not trivial. For example, consider the (7,1/3) case. Although the criterion for a good code in the code search in Ref. 4 was the maximum d_f, $d_f = 15$ codes were overlooked and a $d_f = 14$ code was found. Its transfer function bound is approximated as

$$\begin{aligned}\frac{\partial}{\partial Z} T(D,Z)\bigg|_{Z=1} &= D^{14} + 20D^{16} + 53D^{18} + 184D^{20} + 555D^{22} \\ &\quad + 1961D^{24} + 6384D^{26} + 20655D^{28} \\ &\quad + 64598D^{30} + 203027D^{32} + 631873D^{34} \\ &\quad + 1958874D^{36} + 6028601D^{38} \\ &\quad + 18460857D^{40} + \ldots\end{aligned}$$

Later, a $d_f = 15$ code was found in Ref. 5 whose transfer function bound is approximated as

$$\begin{aligned}\frac{\partial}{\partial Z} T(D,Z)\bigg|_{Z=1} &= 11D^{15} + 16D^{16} + 19D^{17} + 28D^{18} + 55D^{19} \\ &\quad + 96D^{20} + 169D^{21} + 338D^{22} + 636D^{23}\end{aligned}$$

$$+ 1276D^{24} + 2172D^{25} + 3628D^{26}$$
$$+ 6580D^{27} + 12048D^{28} + 20820D^{29}$$
$$+ 36358D^{30} + 65009D^{31} + 115368D^{32}$$
$$+ 204997D^{33} + 356650D^{34} + 622913D^{35}$$
$$+ 1097466D^{36} + 1924564D^{37}$$
$$+ 3356610D^{38} + 5848017D^{39}$$
$$+ 10215732D^{40} + 17821463D^{41} + \ldots$$

With the values of d_f and a_i's only, we cannot compare the two codes. The proper way of comparing them is substituting the actual value of D_o for D and comparing the summations, which is nothing but the evaluation of the transfer function bound.

In conclusion, we have described a technique for calculating the transfer function bound on the BER at the Viterbi decoder output which requires only vector operations for the matrix inversion. Using this technique on previously reported codes of some selected code rates and constraint lengths, we determined codes which require minimum SNR for a given desired BER of 10^{-6}, and provided their BER performance curves. As compared to the series expansion approximation method, our technique is shown to be preferable, since it gives a better approximation of the actual transfer function bound and it can be used directly for the comparison of codes.

References

1. A. J. Viterbi, "Convolutional Codes and Their Performance in Communication Systems," *IEEE Trans. Commun. Tech.*, Vol. COM-19, pp. 751-772, October 1971.
2. A. J. Viterbi and J. K. Omura, *Principles of Digital Communication and Coding*, McGraw-Hill Book Co., New York, NY, 1979.
3. J. K. Omura and M. K. Simon, *Modulation/Demodulation Techniques for Satellite Communications, Part IV. Appendices*, JPL Publication 81-73, Jet Propulsion Laboratory, Pasadena, CA, Nov. 1981.
4. J. P. Odenwalder, *Optimal Decoding of Convolutional Codes*, Ph. D. Thesis, Dept. Sys. Sci., Univ. Calif., L.A., Los Angeles, CA, 1970.
5. K. L. Larsen, "Short Convolutional Codes with Maximal Free Distance for Rate 1/2, 1/3, and 1/4," *IEEE Trans. Infor. Theory*, Vol. IT-19, pp. 371-372, May 1973.
6. S. J. Cury, *Selection of Convolutional Codes Having Large Free Distance*, Ph. D. Thesis, Dept. Sys. Sci., Univ. Calif., L.A., Los Angeles, CA, 1971.
7. R. Johannesson and E. Paaske, "Further Results on Binary Convolutional Codes with an Optimal Distance Profile," *IEEE Trans. Infor. Theory*, Vol. IT-24, pp. 264-268, March 1978.
8. P. J. Lee, "New Short Constraint Length, Rate $1/N$ Convolutional Codes Which Minimize Required E_b/N_o for Given Bit Error Rate," *JPL TDA Progress Report* 42-77, pp. 41-56, May 1984.
9. J. Conan, "The Weight Spectra of Some Short Low-Rate Convolutional Codes," *IEEE Trans. Commun.*, Vol. COM-32, pp 1050-1053, Sept. 1984.

Table 1. Some of the best known convolutional codes, which require minimum E_b/N_o for desired BER of 10^{-6}

(K,r)	d_f	Found in Ref.	Code Generator G in octal			
(3, 1/2)	5	4	7	5		
(4, 1/2)	6	4	17	15		
(5, 1/2)	7	4	35	23		
(6, 1/2)	8	8	77	45		
(7, 1/2)	10	4	171	133		
(8, 1/2)	10	4	247	371		
(9, 1/2)	12	4	561	753		
(10, 1/2)	12	7	1755	1363		
(11, 1/2)	14	7	3645	2671		
(12, 1/2)	15	7	7173	5261		
(13, 1/2)	16	7	12767	16461		
(14, 1/2)	16	7	22555	37457		
(15, 1/2)	18	7	63121	55367		
(16, 1/2)	19	7	111653	145665		
(17, 1/2)	20	7	347241	246277		
(18, 1/2)	20	7	506477	673711		
(3, 1/3)	8	4	7	7	5	
(4, 1/3)	10	4	17	15	13	
(5, 1/3)	12	4	37	33	25	
(6, 1/3)	13	4	75	53	47	
(7, 1/3)	14	4	171	145	133	
(8, 1/3)	16	4	225	331	367	
(9, 1/3)	18	5	557	663	711	
(10, 1/3)	19	6	1765	1631	1327	
(11, 1/3)	22	5	2353	2671	3175	
(12, 1/3)	24	5	4767	5723	6265	
(13, 1/3)	24	5	10533	10675	17661	
(14, 1/3)	25	6	37515	33457	20553	
(15, 1/3)	26	6	77233	67175	41327	
(16, 1/3)	28	6	172465	156371	102657	
(3, 1/4)	10	8	7	7	5	5
(4, 1/4)	12	8	17	15	13	11
(5, 1/4)	15	8	37	35	25	23
(6, 1/4)	18	8	77	73	55	45
(7, 1/4)	20	8	175	151	133	117
(8, 1/4)	22	5	235	275	313	357
(9, 1/4)	23	5	463	535	733	745
(10, 1/4)	27	5	1117	1365	1633	1653
(11, 1/4)	29	5	2327	2353	2671	3175
(12, 1/4)	32	5	4767	5723	6265	7455
(13, 1/4)	33	5	11145	12477	15573	16727
(14, 1/4)	36	5	21113	23175	35527	35537

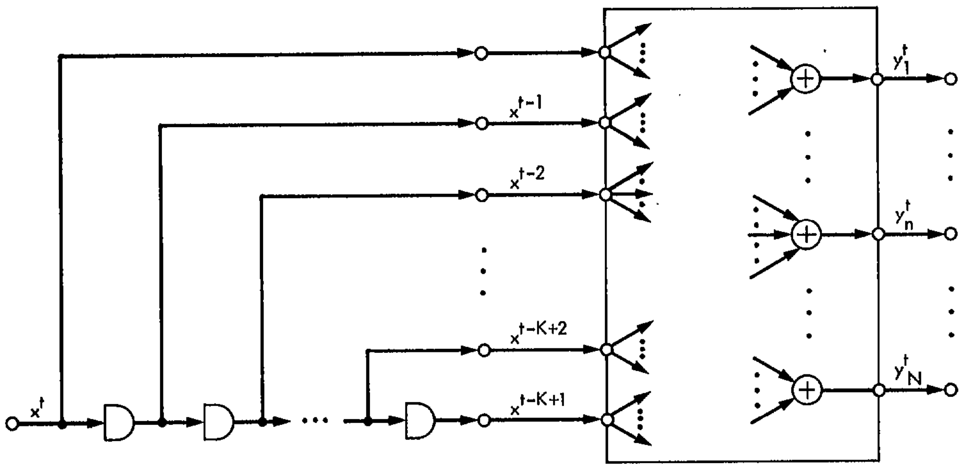

Fig. 1. A nonsystematic, constraint length *K*, rate 1/*N* convolutional encoder structure

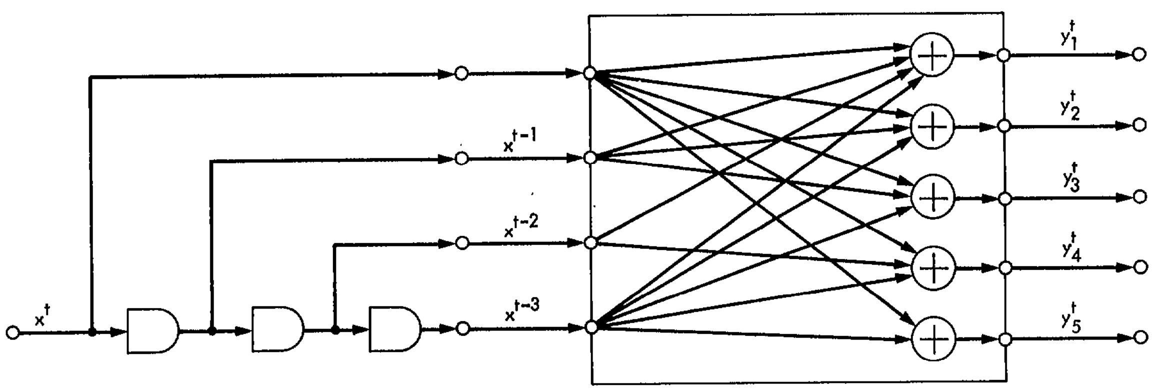

Fig. 2. A (4,1/5) convolutional encoder structure

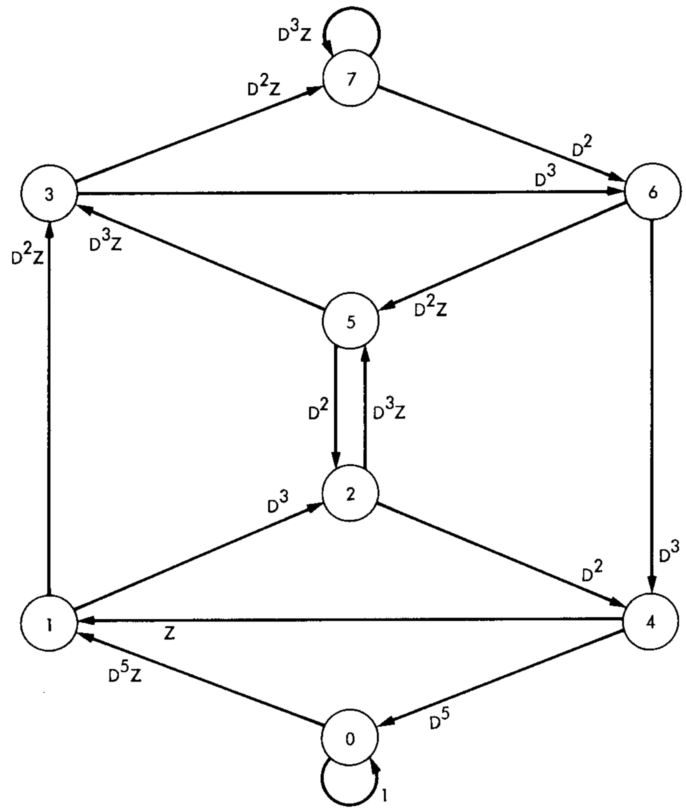

Fig. 3. The state diagram of the encoder in Fig. 2

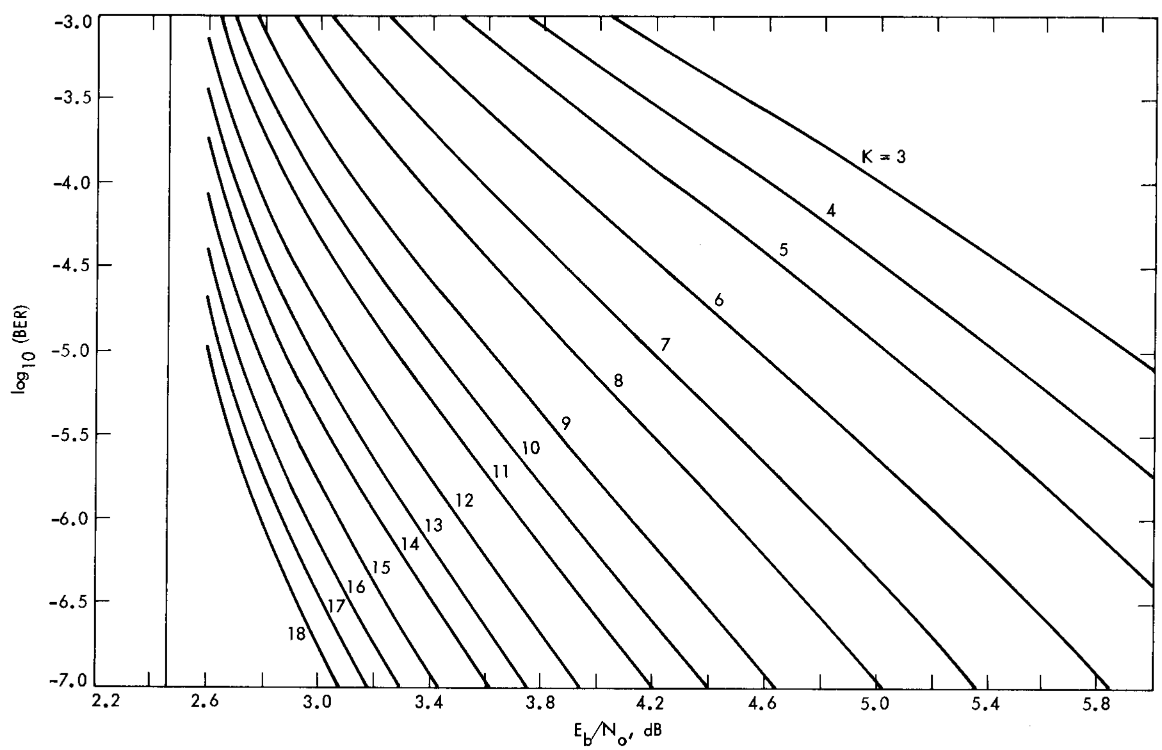

Fig. 4. Transfer function bounds on BER at the output of Viterbi decoder with best known rate 1/2 codes

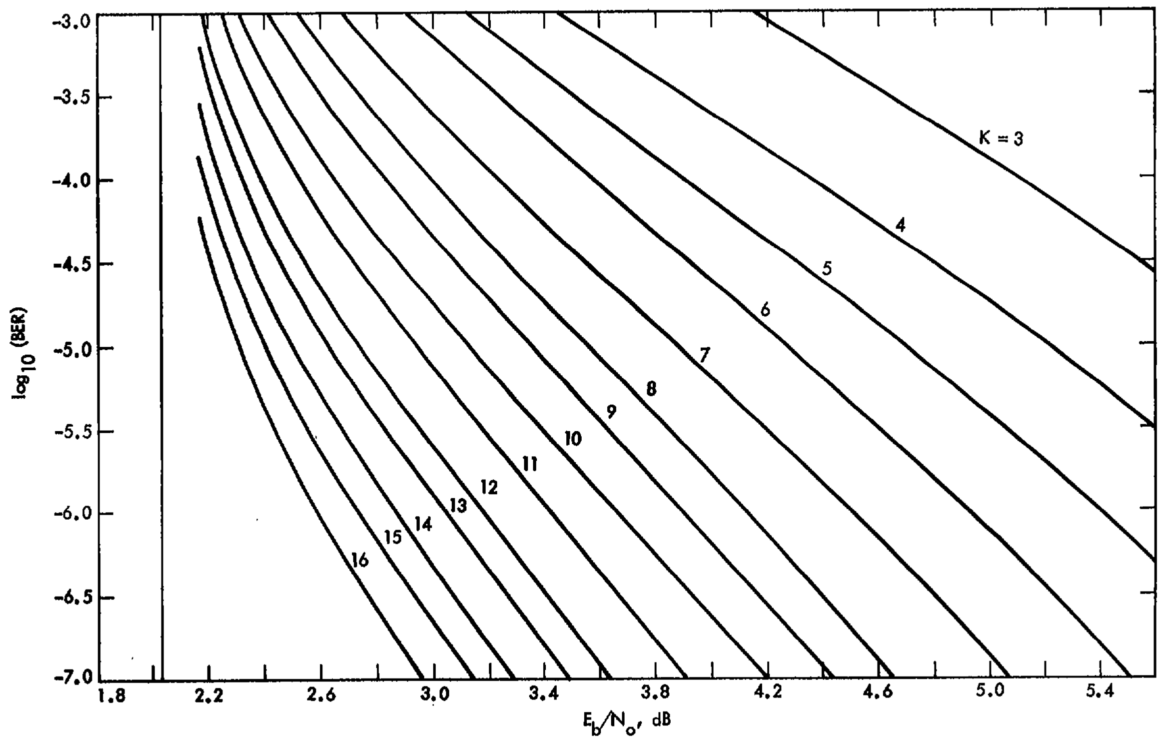

Fig. 5. Transfer function bounds on BER at the output of Viterbi decoder with best known rate 1/3 codes

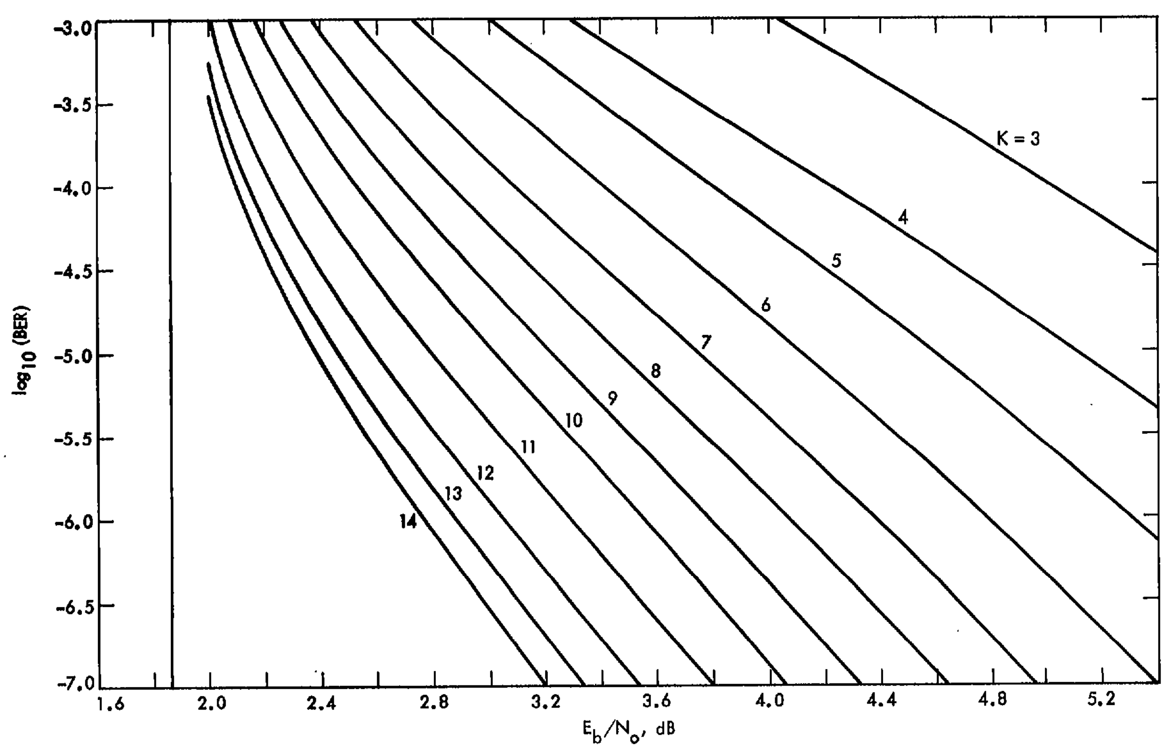

Fig. 6. Transfer function bounds on BER at the output of Viterbi decoder with best known rate 1/4 codes

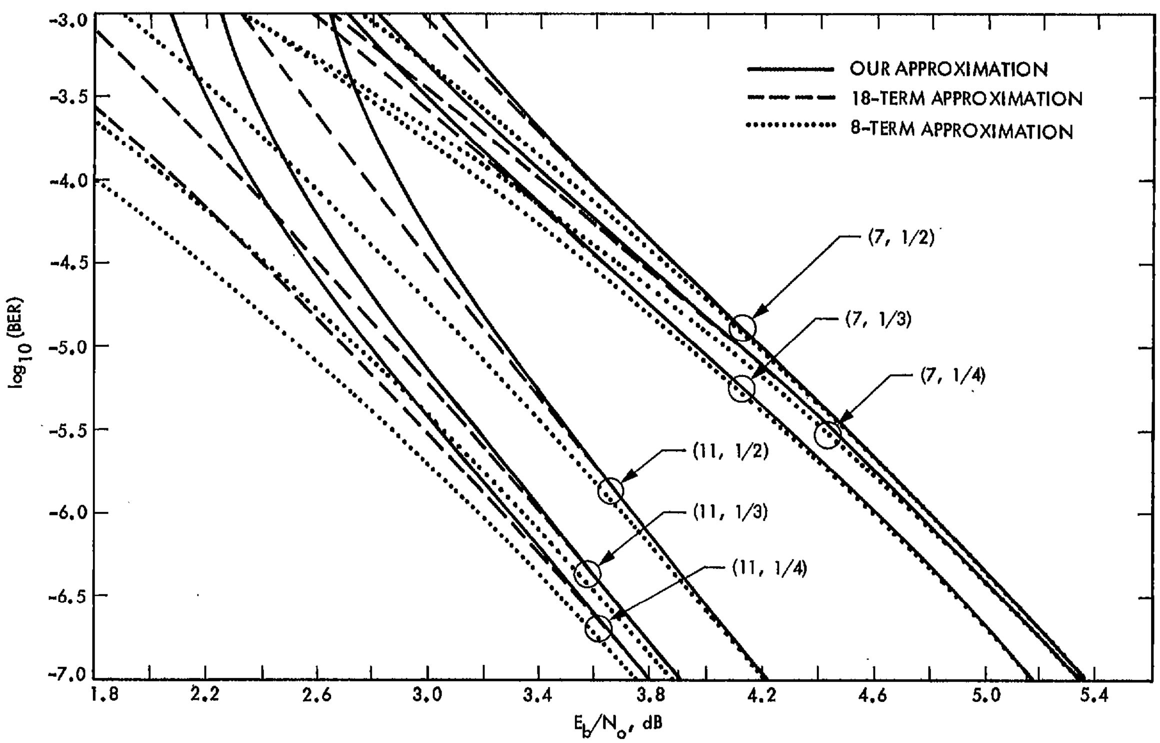

Fig. 7. Comparisons of approximations on transfer function bounds