Transactions Letters

On the Weight Distribution of Linear Block Codes Formed From Convolutional Codes

Jack Keil Wolf and Andrew J. Viterbi

Abstract— **In this note we explain how to obtain the weight enumerator and the performance of linear block codes formed in several distinct ways from a convolutional code.**

ASSUME a rate 1/2 binary convolutional code of constraint length $(m+1)$. Define $\boldsymbol{A}$ to be a 2^m by 2^m matrix with i-jth element A_{ij} given by $A_{ij} = D^h$, if there is an input (either zero or one) that takes the encoder from state i to state j and produces an output pair of Hamming weight h ($h = 0, 1$, or 2). Otherwise, assume that $A_{ij} = 0$. For example, for the simple four-state rate 1/2 binary convolutional encoder shown in Fig. 1, the matrix $\boldsymbol{A}$ is given as

$$\boldsymbol{A} = \begin{matrix} 1 & D^2 & 0 & 0 \\ 0 & 0 & D & D \\ D^2 & 1 & 0 & 0 \\ 0 & 0 & D & D. \end{matrix}$$

The ordering of the rows and columns is easy to deduce for this example. For later examples, we will always assume that the first row and column corresponds to the all-zero state and the last row and column corresponds to the all-one state. Note that for this ordering, the only nonzero terms on the diagonal of the matrix $\boldsymbol{A}$ are the first and last, where the first (i.e., the upper left most term) is always equal to one. For a noncatastrophic encoder, the last (the lower right-most term) can be equal to D or D^2 (but not one).

Here we consider five different methods of constructing binary linear block codes from a rate 1/2 binary convolutional code of constraint length $(m+1)$. All methods except the first two result in block codes with rate equal to 1/2. Of these five methods, the first [the zero tail (ZT) method] and the fourth [the Tailbiting (TB) method] are of most practical interest, the others being included for purposes of generality. As an aside, all of these codes are described by a 2^m state trellis, where m can be much smaller than the number of parity digits or information digits in the code word. In what follows assume that m', m, and k are positive integers such that $0 \leq m' \leq m < k$.

1) *ZT*: The code words of the binary linear code formed by this method are all of the output sequences of the encoder when the encoder is initialized to the all-zero state and $(k-m)$ arbitrary data bits are input into the encoder followed by m zeros. The rate of the resultant block code is $(k-m)/2k$.
2) *Generalized ZT*: The code words of the binary linear code formed by this method are all of the output sequences of the encoder when the encoder is initialized to the all-zero state and $(k-m')$ arbitrary data bits are input into the encoder followed by m' zeros. The rate of the resultant block code is $(k-m')/2k$. For $m' = m$, method 2 reduces to method 1 while if $m' = 0$, method 2 reduces to method 3 (to be described next).
3) *Direct Truncation*: The code words of the binary linear code formed by this method are all of the output sequences of the encoder when the encoder is initialized to the all-zero state and k arbitrary data bits are input into the encoder.
4) *TB* [1]: The code words of the binary linear code formed by this method are all of the output sequences of the encoder when the encoder is initialized to the state corresponding to the last m binary digits of an arbitrary string of k data digits and then those k data digits are input into the encoder.
5) *Generalized TB* [1]: The code words of the binary linear code formed by this method are all of the output sequences of the encoder when the encoder is initialized to the state corresponding to the last m' binary digits of an arbitrary string of k data digits (with the last $(m-m')$ digits of the state set to zero) and then those k data digits are input into the encoder.

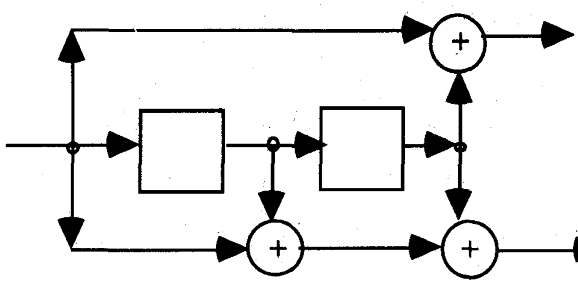

Fig. 1. Four-state, binary, rate 1/2 convolutional encoder.

The $i-j$th element of $[A^k]$ is the weight enumerator for all output sequences when the encoder starts in state i and ends in state j and k binary digits are input into the encoder. The weight enumerators for the binary linear codes formed by the five different methods described above are obtained

Paper approved by T. M. Aulin, the Editor for Coding and Communications Theory of the IEEE Communications Society. Manuscript received July 20, 1995; revised January 20, 1996.

The authors are with the Qualcomm, Inc., San Diego, CA 92121-2779 USA.

Publisher Item Identifier S 0090-6778(96)07089-4.

$$\begin{matrix} X & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - \end{matrix}$$

(a)

$$\begin{matrix} X & - & - & - & X & - & - & - \\ - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - \end{matrix}$$

(b)

$$\begin{matrix} X & X & X & X & X & X & X & X \\ - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - \end{matrix}$$

(c)

$$\begin{matrix} X & - & - & - & - & - & - & - \\ - & X & - & - & - & - & - & - \\ - & - & X & - & - & - & - & - \\ - & - & - & X & - & - & - & - \\ - & - & - & - & X & - & - & - \\ - & - & - & - & - & X & - & - \\ - & - & - & - & - & - & X & - \\ - & - & - & - & - & - & - & X \end{matrix}$$

(d)

$$\begin{matrix} X & - & - & - & X & - & - & - \\ - & X & - & - & - & X & - & - \\ - & - & X & - & - & - & X & - \\ - & - & - & X & - & - & - & X \\ - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - \end{matrix}$$

(e)

Fig. 2. Terms of $[A^k]$ which contribute to the weight enumerator (a) ZT, (b) generalized ZT with $m' = 2$, (c) direct truncation, (d) TB, and (e) generalized TB with $m' = 2$.

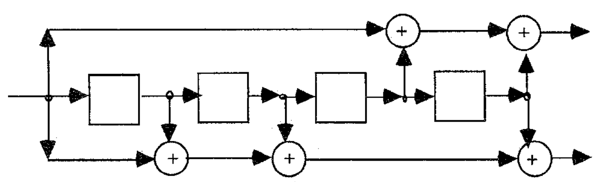

Fig. 3. Sixteen-state, binary, rate 1/2 convolutional encoder.

by summing specific terms in $[A^k]$. To illustrate this we consider a slightly more complicated example of an $m = 3$, (eight-state) rate 1/2 convolutional code, where the rows and columns of the matrix $\boldsymbol{A}$ correspond to encoder states in the following order: $\{000, 100, 010, 110, 001, 101, 011, 111\}$. The X's in Fig. 2(a)–(e) indicate the terms of $[A^k]$ that must be added to give the weight enumerator for the five different types of block codes. For example, for ZT codes, only the term in the upper left most corner of the matrix $[A^k]$ is used while for TB codes we add the terms on the diagonal of $[A^k]$.

The determination of the weight enumerator for any of the block codes requires raising the matrix $\boldsymbol{A}$ (with symbolic coefficients) to the power k. This can be done easily using one of several readily available computer programs. The results obtained using one of them for ZT and TB block codes produced from the 16-state convolutional encoder shown in Fig. 3 are given in Table I. All TB codes in the above table have rate 1/2, while the ZT codes for $k = 12, 15$, and 18 have rates $(12-4)/24 = 1/3, (15-4)/30 = 11/30$, and $(18-4)/36 = 7/18$, respectively. It should be noted that the block length 24 TB code has a smaller minimum Hamming distance ($d_{\min} = 5$) than the ZT code ($d_{\min} = 7$) but for the two larger block lengths, the two methods both yield block codes with the same minimum Hamming distance ($d_{\min} = 7$).

By examination of the individual elements of $[A^k]$, one can find other interesting block codes derived from the convolutional code. For example, if one examines the entries of $[A^{12}]$, one finds several entries with minimum Hamming weight seven. One can construct a linear block code of minimum Hamming weight seven by choosing the union of the code words in the ZT code and the code words corresponding to one of the entries with Hamming weight seven. The resultant code is linear since it consists of a group code and one of its cosets. It has rate $9/24 = 3/8$ (as compared to the ZT code with rate $8/24 = 1/3$) and yet has the same minimum distance as the ZT code.

Sometimes, one does not need the weight enumerator itself but rather the weight enumerator evaluated at numerical values of D. For example, consider that one wishes to calculate a union bound to the block error rate for a maximum likelihood decoder for bipolar signaling over an additive white Gaussian noise (AWGN) channel. If $T(D)$ is the weight enumerator for the block code in question, one knows that the probability of block error is upper-bounded as

$$\begin{aligned} P_{\text{error}} \leq\, & Q(\text{sqr}(2d_{\min}E_s/N_o)) \\ & \cdot \exp(d_{\min}E_s/N_o)\,(T(D)-1)|_{D=\exp(-E_s/N_o)} \end{aligned}$$

where E_s/N_o is the energy per symbol divided by the noise power density and $d_{\min}$ is the minimum distance of the block code. The reason that a one is subtracted from $T(D)$ is to remove the effect of the all-zero code word ($D^o = 1$). This upper-bound can be obtained by simple matrix manipulations on $\boldsymbol{A}$ with numerical values substituted for its nonzero coefficients. The results of using this expression for the ZT and TB codes with $k = 12$ given in Table I are shown in Fig. 4. Here, the probability of block error for each code is plotted versus the energy per bit divided by the noise power density. Note, that even though the TB code has a smaller minimum distance than the ZT code, it has a better performance because of its higher code rate.

For the case of TB codes, the upper-bound can be expressed in terms of the eigenvalues of the matrix $\boldsymbol{A}$. This is the case since the sum of the diagonal elements of the matrix $[A^k]$ is equal to the sum of the eigenvalues of $[A^k]$ and the eigenvalues of $[A^k]$ are equal to the eigenvalues of $\boldsymbol{A}$ raised to the power k. Note that this formulation is convenient if one is interested

TABLE I
WEIGHT DISTRIBUTION OF BLOCK CODES FORMED FROM THE Sixteen-STATE CONVOLUTIONAL CODE

Hamming Weight	ZT k=12	TB k=12	ZT k=15	TB k=15	ZT k=18	TB k=18
0	1	1	1	1	1	1
5	-	12	-	-	-	-
6	-	30	-	-	-	-
7	13	84	19	75	25	54
8	12	174	21	195	30	126
9	12	316	24	440	36	258
10	36	522	80	990	128	972
11	37	612	108	1620	208	2376
12	30	608	121	2510	282	4677
13	38	612	210	3720	570	9144
14	34	498	276	4470	939	14706
15	21	316	268	4674	1174	20580
16	13	177	276	4485	1581	27837
17	6	84	252	3720	2006	32940
18	2	38	176	2650	2066	34288
19	1	12	112	1620	2016	33192
20	-	-	59	858	1824	28170
21	-	-	26	440	1398	21240
22	-	-	12	210	956	14508
23	-	-	5	75	577	8406
24	-	-	2	15	308	4644
25	-	-	-	-	150	2466
26	-	-	-	-	70	1044
27	-	-	-	-	32	416
28	-	-	-	-	6	81
29	-	-	-	-	-	-
30	-	-	-	-	1	18

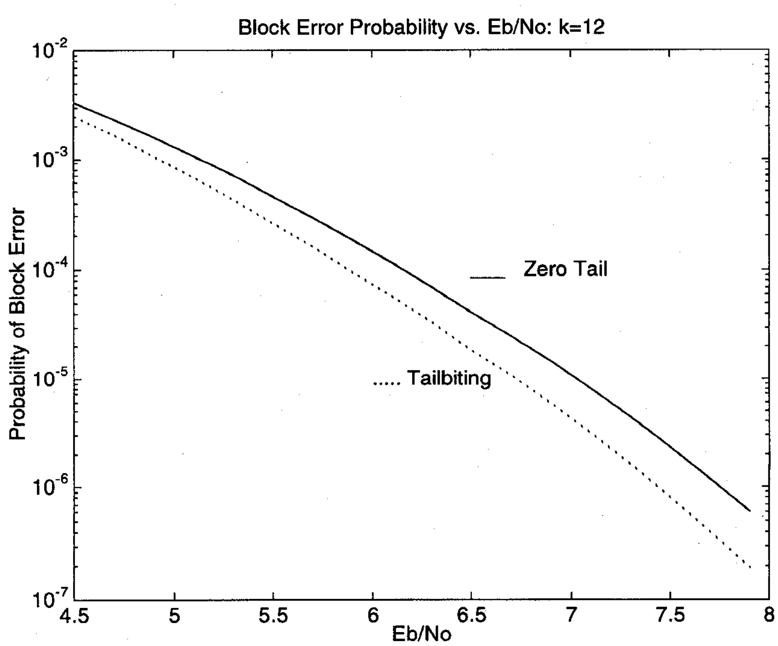

Fig. 4. Block error probability versus the energy per bit divided by the noise power density for the TB and ZT Codes of Table I with $k = 12$.

in the upper-bound for a number of different values of k. A more complicated method for obtaining the performance of these codes was given in a previous paper [1].

Although for simplicity we have restricted this discussion to binary block codes derivable from a rate 1/2 binary convolutional code, the techniques discussed above apply equally well to binary block codes derivable from convolutional codes of all rates, and also to nonbinary block codes derivable from nonbinary convolutional codes.

REFERENCES

[1] H. A. Ma and J. K. Wolf, "On tailbiting convolutional codes," *IEEE Trans. Commun.*, vol. COM-34, no. 2, pp. 104–111, Feb. 1986.