Weight Distribution of a Class of Binary Linear Block Codes Formed from RCPC Codes

Yushi Shen Pamela C. Cosman, *Senior Member, IEEE* Laurence B. Milstein, *Fellow, IEEE*

***Abstract*— In this paper, we study the weight enumerator and the numerical performance of a class of binary linear block codes formed from a family of rate-compatible punctured convolutional (RCPC) codes. Also, we present useful numerical results for a well-known family of RCPC codes.**

***Index Terms*— Block codes, punctured convolutional codes, weight distribution, soft-decision decoding.**

I. Introduction

Rate-compatible punctured convolutional (RCPC) codes, first introduced by Hagenauer [1], are a powerful form of punctured convolutional codes, having flexible rates and requiring an adaptive decoder. Any binary (punctured) convolutional code can be transmitted as a fixed length binary block code, and the knowledge of the weight distribution of linear codes is crucial in its error performance analysis. Methods to obtain the weight distribution of linear block codes formed from a convolutional code were presented in [2], [3]. In this paper, we extend the previous results to compute the weight enumerator of a family of RCPC codes.

II. RCPC Codes: Encoding and Decoding

RCPC codes are a special case of punctured convolutional codes, obtained by adding a rate-compatibility restriction which implies that a high rate code is embedded in the lower rate codes [1]. Mathematically, a family of RCPC codes is described by a mother code and a sequence of puncturing matrices. Assume the generator matrix is $G = (g_{i,j})_{S\times(M+1)}$, with rate $R = 1/S$ and memory order M. Also assume the puncturing matrices are $a(l) = (a_{i,j}(l))_{S\times P}$ for $l = 1, \cdots, (S-1)P$, with the puncturing period P, and $a_{i,j}(l)\epsilon(0,1)$ where 0 implies puncturing. The rate-compatibility restriction implies

$$\text{if } a_{i,j}(l_0) = 1, \quad \text{then } a_{i,j}(l) = 1 \ \text{ for all } 1 \le l_0 \le l.$$

Note the rate of a RCPC code is $R(l) = P/(P+l)$, so a code with a larger value of l has more powerful error correction capability.

A simple example of a family of RCPC codes is given in [1], where a rate $1/2$ convolutional code with $M = 2$ is punctured periodically with $P = 4$. The generator polynomial of the mother code is $G(D) = \{D^2+D+1, D^2+1\}$, and a sequence of puncturing tables is

$$a(1) = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}, \quad a(2) = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix},$$
$$a(3) = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix}, \quad a(4) = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix},$$

with code rates $4/5$, $4/6$, $4/7$ and $4/8$, respectively.

On the receiving side, the decoder can use the Viterbi algorithm (VA) with a trellis modified by the current puncturing matrix $a(l)$. Suppose x is sent and y is received. For binary transmission over an an additive white Guassian noise (AWGN) channel, the VA will find the path $\hat{x}^m$ which satisfies

$$\max_m \Big(\sum_{j=1}^{J}\sum_{i=1}^{S} a_{i,j}\ \hat{x}^m_{i,j}\ y_{i,j}\Big) \tag{1}$$

where $a_{i,(j+P)} = a_{i,j}$ is the (i,j)-th entry of $a(l)$, and J is the trellis length.

III. Transition Matrix Sequence

The transition matrix of a convolutional code is used to describe the state transition possibilities and corresponding output weight of the code [2]. For a convolutional code with rate $R = 1/S$ and memory M, the transition matrix is a 2^M by 2^M matrix. Assuming $i_n\epsilon(0,1)$ is the weight of the n-th output, and $H = \sum_{n=1}^{S} i_n$ is the Hamming weight of the entire output. Denote by $A_{i,j}$ the (i,j)-th entry of the transition matrix, $A_{i,j} = D^H$ if there is an input (either zero or one) that takes the encoder from state i to state j; otherwise, $A_{i,j} = 0$. For example, the transition matrix of the convolutional code given in Section II is

$$A = \begin{pmatrix} D^0\cdot D^0 & D^1\cdot D^1 & 0 & 0 \\ 0 & 0 & D^1\cdot D^0 & D^0\cdot D^1 \\ D^1\cdot D^1 & D^0\cdot D^0 & 0 & 0 \\ 0 & 0 & D^0\cdot D^1 & D^1\cdot D^0 \end{pmatrix}$$
$$= \begin{pmatrix} 1 & D^2 & 0 & 0 \\ 0 & 0 & D & D \\ D^2 & 1 & 0 & 0 \\ 0 & 0 & D & D \end{pmatrix}. \tag{2}$$

As stated in [2], the (i,j)-th element of the K-th power of A, $(A^K)_{i,j}$, gives the output weight enumerator, given that the encoder starts in state i, ends in state j, and K binary digits are fed into the encoder.

Manuscript received January 26, 2005. The associate editor coordinating the review of this letter and approving it for publication was Prof. Marc Fossorier. This research was supported by the California Institute for Telecommunications and Information Technology, by Ericsson, Inc., by the State of California under the CoRe program, and by the Office of Naval Research.

Y. Shen, P. C. Cosman and L. B. Milstein are with the Department of Electrical and Computer Engineering, University of California, San Diego, La Jolla, CA 92093-0407 USA (E-mail: {yushen,pcosman,lmilstein}@ucsd.edu).

EDICS: CL1.3, CL1.5.0

For RCPC codes, the output information changes periodically due to the periodic puncturing. Therefore, we need a transition matrix sequence to describe the state transition possibilities and the weights of the outputs. We denote this sequence by $A_1, A_2, \cdots, A_x, \cdots$, where $A_{x+P} = A_x$. Each matrix A_x is obtained from the structure of the mother code and the λ-th column of the puncturing matrix $a(l)$, where $\lambda \equiv x \pmod P$ and $\lambda \epsilon (1, \cdots, P)$. Specifically, the (i,j)-th entry of A_x is equal to D^h, if there is an input that takes the encoder from state i to state j, and h is the Hamming weight of the punctured output using the λ-th column of $a(l)$; otherwise $(A_x)_{i,j} = 0$.

For example, for the family of RCPC codes described by Section II, define matrix A as in Equation (2), and define matrices B and C as follows, which correspond to the first and second output, respectively:

$$B = \begin{pmatrix} 1 & D & 0 & 0 \\ 0 & 0 & D & 1 \\ D & 1 & 0 & 0 \\ 0 & 0 & 1 & D \end{pmatrix}, \quad C = \begin{pmatrix} 1 & D & 0 & 0 \\ 0 & 0 & 1 & D \\ D & 1 & 0 & 0 \\ 0 & 0 & D & 1 \end{pmatrix}.$$

The matrix sequences of the four RCPC codes are

$$\begin{aligned} a(1): &\ A_1 = A,\ A_2 = B,\ A_3 = B,\ A_4 = C,\ \cdots \\ a(2): &\ A_1 = A,\ A_2 = A,\ A_3 = B,\ A_4 = C,\ \cdots \\ a(3): &\ A_1 = A,\ A_2 = A,\ A_3 = B,\ A_4 = A,\ \cdots \\ a(4): &\ A_1 = A_2 = A_3 = A_4 = \cdots = A. \end{aligned}$$

We define matrix Φ^K by $\Pi_{x=1}^{K} A_x$, which yields the output information for K continuous steps of the RCPC code. In particular, the (i,j)-th element of the matrix Φ^K, $(\Phi^K)_{i,j}$, gives the output weight enumerator, given that the encoder starts in state i, ends in state j, and K binary digits are sent into the RCPC encoder.

IV. Weight Enumerator

Several different methods for constructing binary linear block codes from a convolutional code were presented in [2], along with a way to find the corresponding weight enumerator $T(D) = \sum_d A_d D^d$, where A_d is the number of codewords of weight d. These methods can be applied to the block codes formed from a RCPC code. Denote by R_p and R_{block} the rate of an unterminated punctured convolutional code and of the resultant block code, respectively, and denote by K and N the fixed block length of the input and output of the block RCPC encoder, respectively. As an example from [2], for the zero tail (ZT) method, $T(D)$ is given by $(\Phi^k)_{1,1}$, $R_{block} = \frac{(K-M)}{K} R_p$, and $N = K/R_p = (K-M)/R_{block}$.

Having the weight enumerator $T(D)$ of a linear block code, we may use it to evaluate its performance. Denote by d_{min} the minimum distance of the block code, and by E_s/N_0 the energy-per-symbol divided by the noise power density. Note that $E_s/N_0 = R_{block} \cdot E_b/N_0$ where E_b/N_0 is the energy-per-bit divided by the noise power density. The union bound on the block error rate of a RCPC code for binary transmission over an AWGN channel is given by

$$P_{block} \leq \sum_{d=d_{min}}^{N} A_d Q(\sqrt{2d\frac{E_s}{N_0}}). \quad (3)$$

Furthermore, a good approximation to the union bound of the bit error probability P_{bit} is obtained by scaling each term in the sum of Equation (3) by (d/N) [4].

V. Numerical Examples

In this section, we show the results for the block codes formed from the "Good" RCPC codes with $M = 6$ and $P = 8$ [1], whose encoder is shown in Fig. 1. In particular, we examine the block codes formed from the rate $8/9$, $2/3$ and $1/3$ codes in this family [5].

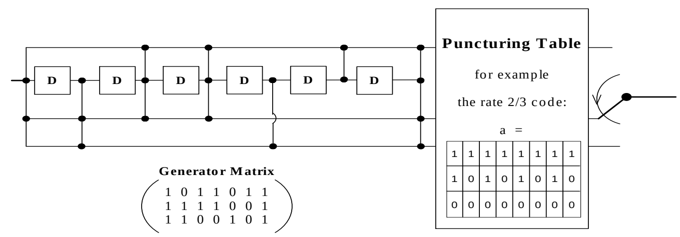


Fig. 1. The 64-state RCPC codes with puncturing period $P = 8$.

With the ZT method, using the method illustrated in this paper, the weight distributions of the block codes with different input block lengths K are given in Table I. The block error rates and bit error rates of these block codes with $K = 400$ for an AWGN channel are shown in Fig. 2.

VI. Conclusions

We illustrated how to compute the weight enumerator and evaluate the performance of binary linear block codes formed from a family of RCPC codes. The concept of the transition matrix sequence was introduced and explained. Numerical results for a well-known family of RCPC codes were also presented.

References

[1] J. Hagenauer, "Rate-compatible punctured convolutional codes (RCPC codes) and their applications," *IEEE Transactions on Communications*, vol. 36, pp. 389-400, Apr. 1988.
[2] J. Wolf and A. Viterbi, "On the weight distribution of linear block codes formed from convolutional codes," *IEEE Transactions on Communications*, vol. 44, pp. 1049-1051, Sept. 1996.
[3] M. P. C. Fossorier, S. Lin and D.J. Costello, Jr., "On the weight distribution of terminated convolutional codes," *IEEE Transactions on Information Theory*, vol. 45, no. 5, pp. 1646-1648, July 1999.
[4] M. P. C. Fossorier, S. Lin and D. Rhee, "Bit error probability for maximumLikelihood decoding of linear block codes and related soft-decision decoding methods," *IEEE Transactions on Information Theory*, vol. 44, no. 7, pp. 3083-3090, Nov. 1998.
[5] P. G. Sherwood and K. Zeger, "Progressive image coding on noisy channels," *IEEE Signal Processing Letters*, vol. 4, no. 7, pp. 189-191, July 1997.

TABLE I

WEIGHT DISTRIBUTION OF BLOCK CODES FORMED FROM THE 64-STATE RCPC CODES, WITH PERIOD $P = 8$.

Weight Distance	Rate 8/9 RCPC Code			
	K=200	K=400	K=600	K=800
0	1	1	1	1
3	94	194	294	394
4	1390	2965	4540	6115
5	17247	37822	58397	78972
6	195637	455037	724437	1003837
7	2254907	5634757	9329607	13339457
8	25932510	70104784	120872684	178236209

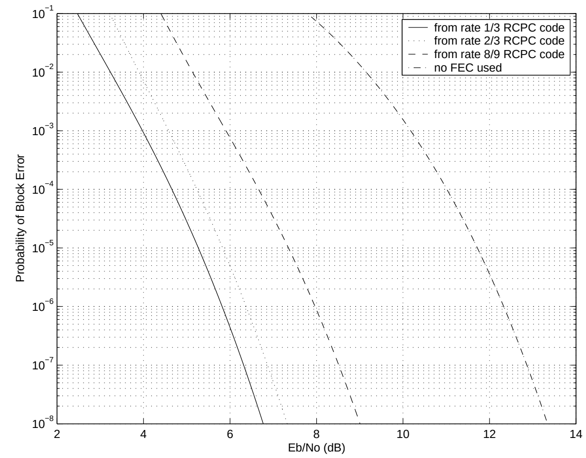
(a) $d_{min} = 3$

Weight Distance	Rate 2/3 RCPC Code			
	K=200	K=400	K=600	K=800
0	1	1	1	1
6	96	196	296	396
7	1509	3109	4709	6309
8	4447	9247	14047	18847
9	14350	30150	45950	61750
10	57369	121569	185769	249969
11	213677	457177	700677	944177
12	794911	1726461	2668011	3619561

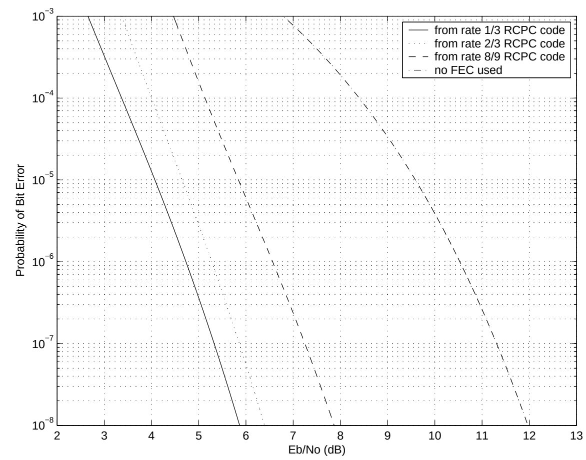
(b) $d_{min} = 6$

Weight Distance	Rate 1/3 RCPC Code			
	K=200	K=400	K=600	K=800
0	1	1	1	1
14	194	394	594	794
16	1338	2738	4138	5538
18	2072	4272	6472	8672
20	6546	13546	20546	27546
22	16698	34698	52698	70698
24	51209	107009	162809	218609
26	147582	309782	471982	634182

(c) $d_{min} = 14$



(a)



(b)

Fig. 2. Performance evaluation for the ZT codes formed from the RCPC codes, as shown in Fig. 1, with fixed length $K = 400$. The corresponding weight distributions are given in Table I. (a) Block error prob. vs. the energy-per-bit divided by the noise power density. (b) Bit error prob. vs. the energy-per-bit divided by the noise power density.