On Design of Interleavers with Practical Size for Turbo Codes

Duanyi Wang and Hisashi Kobayashi

Department of Electrical Engineering, Princeton University
Princeton, New Jersey 08544, USA

Tel: (609) 258-4643 Fax: (609) 258-2158
Email: {duanyi, hisashi}@ee.princeton.edu

***Abstract* – In this paper, we propose a new method to design an interleaver with practical size for turbo codes. In order to construct an optimal interleaver, we first set up the following design rules: (i) to suppress the interleaver correlation; (ii) to break up self-terminating weight-2 input sequences; and (iii) to avoid edge effects. We then synthesize a new type of interleaver structure, starting with the conventional block interleaver and applying simple transformation steps. The resulting interleaver can create turbo codes with relatively small block size that yield better performance than previously known interleavers. Advantages of the new interleaver are demonstrated by conducting comprehensive comparisons based on both theoretical analyses and computer simulations.**

I. Introduction

Since the introduction of turbo codes [1], a considerable effort has been made in search for good interleavers. One major issue of great practical interest is concerned with design of an interleaver with small size. Generally speaking, the structure of an interleaver in terms of its type and depth plays a key role in determining the performance of a given turbo code. It has been reported that turbo codes can achieve a remarkable performance gain by using a sufficiently large interleaver (more than 1000 bits long). However, the long delay caused by such a large interleaver is often too prohibitive to be practical in real-time applications, such as voice or video transmission. There have been some successful attempts in designing turbo-code interleavers of practical size suitable for short frame transmissions. For example, 192-bit and 256-bit interleavers, corresponding to 9.6 kbps and 13 kbps with roughly 20ms frames have been proposed for IS-95 and GSM [2], [3]. However, the gains achievable by these small-size interleavers are not large compared with large frame situations, hence a further study and development of even better small-size interleavers are called for.

In this paper, we propose a new design method for small-size interleavers and show that it can provide better performance than previously known interleavers. Advantages of the new interleaver over the others are demonstrated by conducting comprehensive comparisons based on both theoretical analyses and computer simulations.

II. Role of Interleaver in Turbo Codes and Optimum Interleaving Criteria

We may treat a turbo code, acting on blocks of data, as an equivalent linear block code [7]. Consider a rate 1/3 turbo code as shown in Fig. 1, which is constructed by parallel concatenation of two identical RSC (recursive systematic convolutional) constituent encoders with a turbo internal interleaver in-between, where each RSC encoder has a rate 1/2 with constraint length $K = 3$ (i.e., memory $\nu = 2$) and the parity polynomial $g_1(D) = 1+D^2$ and the feedback polynomial $g_0(D) = 1+D+D^2$. We often represents this turbo encoder by $(1, g_1(D)/g_0(D))_{oct}$. The turbo encoder of Fig. 1 is therefore denoted (1, 5/7). For each input block $\mathbf{u} = (u_0, u_1, ..., u_{N-1})$ of length N, the first component encoder RSC1 operates directly on it and produces the first parity sequence $\mathbf{y}_1 = (y_{1,0}, y_{1,1}, ..., y_{1,N-1})$; the second component encoder RSC2 operates on the interleaved version of $\mathbf{u}$ and produces the second parity sequence $\mathbf{y}_2 = (y_{2,0}, y_{2,1}, ..., y_{2,N-1})$. The resulting overall coded sequence of the turbo code is the combination of the three components $\mathbf{u}$, $\mathbf{y}_1$ and $\mathbf{y}_2$.

For a linear code, its error correcting capability is closely related to the distribution of Hamming weights of its codewords, i.e. the weight distribution of the code. The role of the interleaver in a turbo code is essentially to improve the weight distribution of the code. The weight distribution of a turbo-encoded sequence depends on how coded sequences from the component encoders are composed together, and a good interleaver is the one that reduces the number of relatively low-weight overall turbo-encoded output sequences. Since the class of turbo codes we study here satisfy the group properties, we assume, without loss of generality, that the true input sequence transmitted is an all-zero sequence. Therefore, we only need to consider the Hamming weights of other nonzero sequences. In the turbo coding scheme depicted in Fig. 1, the Hamming weights

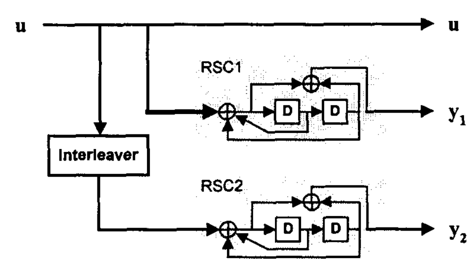


Fig. 1. Example of turbo encoder

0-7803-6283-7/00/$10.00

(originating from non-zero input sequences) of the overall output sequences are composed of three parts: the weight of the input sequence $w(\mathbf{u})$ and the weights of the parity sequences $w(\mathbf{y}_1)$ and $w(\mathbf{y}_2)$. A good interleaver should associate a low-weight sequence from one component encoder with a large-weight coded sequence from the other component encoder in order to make the overall weights as large as possible.

Because of the IIR (infinite impulse response) property, associated with the recursive structure of the RSC component encoders, the input sequences can be classified into two types with respect to the "true" input sequence (i.e., the all-zero sequence) that we are interested in correctly decoding: (1) self-terminating sequences, which return the encoder to the all-zero state after encoding N information bits, without any need for trellis termination technique; and (2) non-self-terminating sequences, where additional tail bits are required in order to force the encoder back to the all-zero state at the end of the sequence block. According to some probabilistic argument based on random interleaving [8], [9], non-self-terminating sequences have little effect on the turbo decoder performance because the coded sequences resulted from them can usually accumulate large weights unless the first bit "1" in these non-self-terminating sequences starts near the end of the block (in this situation, the position of the first 1 in the information sequence is also near the end of the block, the resulting overall coded sequence will still have a low weight, and this situation is called "edge effect"). The coded sequence resulting from a non-self-terminating input sequences should have some periodic structure because the RSC's IIR response is periodic. This period is called the "intrinsic period" of a given turbo code, and is denoted p. If the memory of the RSC encoder is v, then $p \leq 2^v-1$. The intrinsic period has a significant effect on the performance of the turbo code. To construct a good code, we usually choose a primitive polynomial (of degree v) as the feedback polynomial $g_0(D)$ of the RSC component encoders so as to make $p = 2^v-1$.

Among the self-terminating input sequences, it is primarily low-weight input sequences (e.g. w = 2, 3, 4, ...) that result in low-weight overall coded sequences. In particular, self-terminating weight-2 input sequences of the polynomial form $\mathbf{u}(D) = D^k+D^l$, where $|l-k| \bmod p = 0$ are most likely to produce fairly low weights at each component encoders. For example, the RSC encoders of Fig. 1 have the intrinsic period p = 3, and problematical input sequences are the weight-2 sequences with the 1's separated by $2+3t_1$ zeros (t_1 = 0, 1, 2, ...), and the corresponding RSC encoded output weights are $4+2\,t_1$. Such a sequence will, after being interleaved, become another weight-2 sequence with its 1's separated by $2+3t_2$, where t_2 is multiple of 1/3. If the t_2 is not an integer, the corresponding coded output will have a large weight because the RSC output is non-terminating (until the end of the block). But if t_2 is an integer, the resulting weight will be only $10+2(t_1+t_2)$ if the integer duplex (t_1, t_2) are both small [8], [9].

As for other possible offending self-terminating input sequences of weight n = 3, 4, 5, ... , it has been shown that these higher-weight input sequences, compared with self-terminating weight-2 input sequences, are much more likely to be broken up after proper interleaving. Therefore, they will have much less important effects on the turbo code performance than weight-2 sequences.

Based on the above analysis of the weight distribution characteristics of turbo codes, we conclude that the interleaver design should be focussed on the following rules:

1) Define the "duo-distance" between position i and position j for a given interleaver:

$$d_{\text{duo}}(i,j) = |i-j| + |\pi(i) - \pi(j)|,$$

where $\pi(i)$, $\pi(j)$ are interleaved positions of i, j, where i, j = 0, 1, 2, ... , $N-1$ (N is the length of interleaver block), and $i \neq j$.
The d_{duo} should be made as large as possible in order to lower the correlation between the interleaver output sequence and its input sequence.
2) The distances between any two input information bits before and after the interleaver, denoted $d(i,j) = |i-j|$ and $d(\pi(i), \pi(j))$, i, j = 0, 1, 2, ... , $N-1$ should not be both integer multiple of the intrinsic period p so that the chance of feeding self-terminating weight-2 sequences may be avoided or reduced.
3) The positions of any input information bit before and after the interleaver, i.e., i and $\pi(i)$ $(0 \leq i \leq N-1)$, should not be both near the end of the interleaver block in order to avoid edge effects.
In other words, if i is nearly N, then both $\pi(i)$ and $\pi^{-1}(i)$ should be much smaller than N.

III. New Interleaver Design Method

Using the design criteria discussed in Section 2, we construct a new type of interleaver structure, with intrinsic period p and interleaving block length $m \times m$ bits, by the following procedure:

1) Start with the conventional block interleaver of size $m \times m$. Data are written into column by column and read out row by row.
2) Apply the "modulo-p" operation to the order in which the data are read out from above the block interleaver, and then partition the data into p groups as follows:

$$\text{Group}_0, \text{Group}_1, \text{Group}_2, \dots, \text{Group}_{p-1}$$

where data in Group$_i$ consisting of those which are in position k of the block interleaver output, where $k = i \pmod p$, i = 0, 1, 2, ... , $p-1$;
3) Divide the data sequence in each group, according to the order in which the data are read out from the block interleaver, into blocks of length p (Note: the number of data in the last block in a group will be less than p unless

the length of data in the group is an integer multiple of p):

$$\begin{array}{lll} \text{Group}_0 & \Rightarrow & \text{Block}_{0,1}\,,\ \ \text{Block}_{0,2}\,,\ \ldots\ldots \\ \text{Group}_1 & \Rightarrow & \text{Block}_{1,1}\,,\ \ \text{Block}_{1,2}\,,\ \ldots\ldots \\ \vdots & & \vdots \qquad\qquad \vdots \\ \text{Group}_{p-1} & \Rightarrow & \text{Block}_{p-1,1}\,,\ \ \text{Block}_{p-1,2}\,,\ \ldots\ldots \end{array}$$

4) Permute the above-arranged data in the following sequence:

$$\begin{array}{l} \text{Block}_{0,1}\,,\ \text{Block}_{2,1}\,,\ \ldots\,,\ \text{Block}_{p-1,1}\,; \\ \text{Block}_{0,2}\,,\ \text{Block}_{2,2}\,,\ \ldots\,,\ \text{Block}_{p-1,2}\,; \\ \vdots \end{array}$$

5) Finally output the data sequence obtained in Step 4 in the reverse order.
The purpose to output the sequence backward here is to avoid the edge effect mentioned earlier.

Fig. 2 gives the design process for this new interleaver of length 8×8 bits that can be used in the turbo encoder of Fig. 1.

0	8	16	24	32	40	48	56
1	9	17	25	33	41	49	57
2	10	18	26	34	42	50	58
3	11	19	27	35	43	51	59
4	12	20	28	36	44	52	60
5	13	21	29	37	45	53	61
6	14	22	30	38	46	54	62
7	15	23	31	39	47	55	63

(a) Data put in the conventional block interleaving mode (8×8 bits)

0	8	16	24	32	40	48	56
1	9	17	25	33	41	49	57
2	10	18	26	34	42	50	58
3	11	19	27	35	43	51	59
4	12	20	28	36	44	52	60
5	13	21	29	37	45	53	61
6	14	22	30	38	46	54	62
7	15	23	31	39	47	55	63

(b) Data partitioned into p groups (black, white and grey corresponding to p = 3)

0	24	48	8	32	56	16	40
1	9	33	57	17	41	2	25
49	10	18	42	3	26	50	11
34	58	19	27	51	12	35	59
20	43	4	28	36	60	21	44
5	29	52	13	37	45	6	30
53	14	38	61	22	46	54	15
39	62	23	47	7	31	55	63

(c) After permutation in Step 4

63	55	31	7	47	23	62	39
15	54	46	22	61	38	14	53
30	6	45	37	13	52	29	5
44	21	60	36	28	4	43	20
59	35	12	51	27	19	58	34
11	50	26	3	42	18	10	49
25	2	41	17	57	33	9	1
40	16	56	32	8	48	24	0

(d) After reversing in Step 5 (The output sequence: 63, 55, 31, ...)

Fig. 2. The new interleaver with p = 3, length 8×8 bits

IV. Analytical Results

Using the new interleaver of length 8×8 bits designed in Section 3 as an example, comprehensive comparisons are made between some known interleavers and the interleaver we propose. As criteria for comparison, we evaluate the distribution of the edge distance d_{edge} ($d_{\text{edge}} = 2(N-1) - i - \pi(i)$, where $\pi(i)$ is the interleaved position of position i and N the interleaver block length); the output weight distribution d_{st} where the inputs to both RSC encoders are all self-terminating weight-2 input sequences, i.e., both $|i-j|$ and $|\pi(i)-\pi(j)|$ $(0 \le i \le N-1)$ are all integer multiple of the intrinsic period p; the duo-distance $d_{\text{duo}}(i, j)$ and the general weight distribution of turbo code listed in Figs. 3 through 6.

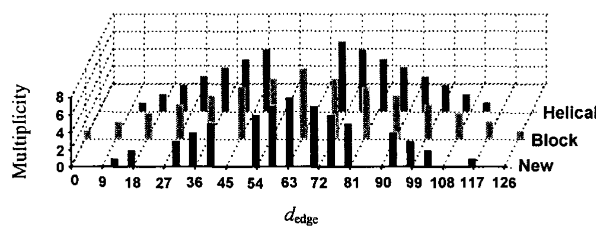


Fig. 3. Distributions of d_{edge} in the proposed interleavers (New), the conventional block interleaver (Block) and helical block interleaver (Helical) (N=64 bits)

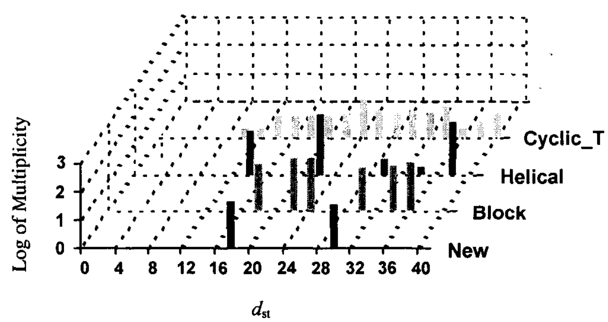


Fig. 4. Distributions of d_{st} (≤40) the output weight for self-terminating weight-2 input sequences in some interleavers (N=64 bits, turbo code (1, 5/7))

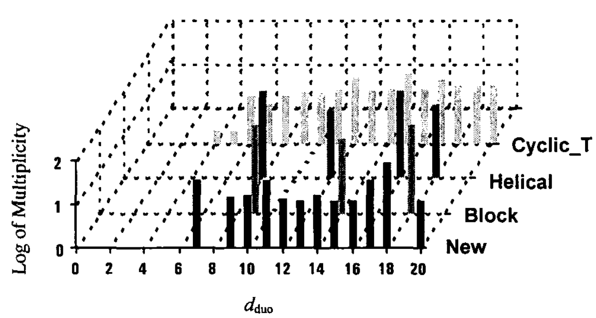


Fig. 5. Duo-distance d_{duo} (≤ 20) of some interleavers (N=64 bits)

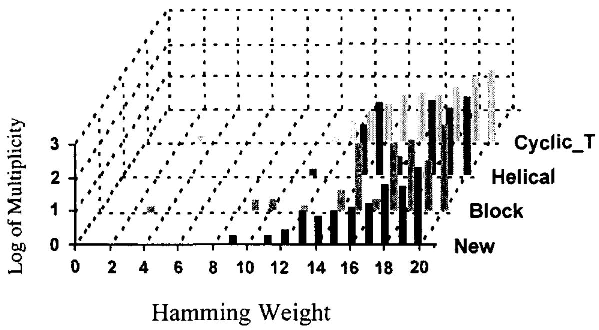


Fig. 6. Weight distribution ($w \leq 20$) of turbo code (1, 5/7) using some interleavers (Input sequence of weight-n (n=1,2,3,4), N=64 bits)

By referring to Fig. 3, we find that the new interleaver and the helical block interleaver [10] have the same minimum d_{edge} (=12). The minimum d_{edge} for the new interleaver corresponds, in referring to Fig. 2(a) and (d), to the input position 56 and output position $\pi(56)$=58 (as noted by circles). This value is much larger than the minimum d_{edge} of the conventional block interleaver which is zero. The distribution of d_{edge} for Cyclic_T interleaver [10] is not shown in Fig. 3 since the d_{edge} for this kind of interleaver depends on some address translation matrix that the interleaver adopts, but it is easy to see that its minimum d_{edge} is also zero. And the multiplicity of the shorter d_{edge} for the new interleaver is smaller than that for the other interleavers.

Fig. 4 shows the output weight distribution for self-terminating weight-2 input sequences to both RSC encoders. Suppose that this kind of input sequence has the polynomial form D^k+D^l, where $|l-k| \bmod p = 0$, as above mentioned, according to [11], the d_{st} for the p=3 turbo code (1, 5/7) of Fig. 1 is computed by

$$d_{\text{st}} = 6+2\left(\frac{|l-k|+|\pi(l)-\pi(k)|}{3}\right),$$

and the minimum d_{st} is defined as the free distance (d_{free}) of the turbo code, i.e.,

$$d_{\text{free}} = \min\{d_{\text{st}}\} = 6+2\min_{l,k}\left(\frac{|l-k|+|\pi(l)-\pi(k)|}{3}\right).$$

We see that the new interleaver and conventional block interleaver have the same d_{free} (=18), which is larger than the d_{free} of the other interleavers. But the multiplicity of the new interleaver for small d_{st}, such as the region d_{st}<40 as shown in this figure, is smaller than that of the conventional block interleaver, hence this supports the superiority of the new interleaver.

Fig. 5 gives the distribution of the duo-distance for the various interleavers. Only the situation for the d_{duo} less than 20 is shown. Although the conventional block interleaver and the helical block interleaver have larger minimum d_{duo}s (= 9, 8 respectively) than the new interleaver (minimum d_{duo} = 7), the multiplicities of the former two interleavers for smaller d_{duo}, such as the region d_{duo} <10, are much larger than that of the new interleaver, so it is difficult to claim which has advantage over the other.

Fig. 6 shows the weight distribution of these interleavers, only for the Hamming weights less than 20. From the points of both the values of Hamming weight and the multiplicity, it is clear that the new interleaver is, in general, better than the others. But for the region of the Hamming weights less than 13, the helical block interleaver is the best.

To sum up, the new interleaver has been shown to be better in terms of above various criteria than the previously known interleavers, i.e., it can outperform the others in avoiding the edge effects, breaking up self-terminatiing weight-2 input sequences, and suppressing the interleaver correlation.

V. Simulation Results

To augment the above theoretical analyses and verify the superiority of the new interleaver, we evaluated the BER performance of the new and other interleavers on computer simulation. We tested two block size, N = 64 and N = 256, and apply both to the turbo code (1, 5/7) of Fig. 1, using MAP decoding algorithm with eight iterations. The simulation results are shown in Figs. 7(a) and 7(b), which support the theoretical results presented in Section 4. In these results, the curves marked "Pseudo-Random" correspond to the case where the interleaver is a random permutation obtained by a pseudo-random sequence. Although random interleavers is most favourable for turbo codes design when their size is very large, they usually cannot give the performance satisfactory enough to be the best choice when their size is small. This is demonstrated in these Fig.s. For the shorter block length (N = 64 bits), the new interleaver shows a better performance than the others for SNR > 1.5dB, and is not much inferior to the others even in the lower SNR region. The performance improvement of the new interleaver is much clearer for the longer block length (N=256), especially in the region SNR > 1.0 dB.

VI. Conclusion and Discussion

A new interleaver design, which can provide turbo codes with better performance than previously known interleavers for relatively small block size, has been presented and its advantages have been confirmed by both theoretical analyses and computer simulations for the turbo code shown in Fig. 1. What we presented in this paper are preliminary results, and it will require a further analysis and experiments to draw a conclusion for more general class of turbo codes. In the design method presented here, we mainly concentrated on the weight-2 sequences since they largely determine the weight distribution of turbo codes. However, as mentioned in Section 2, input sequences with weight-n, for $n \geq 3$, also produce self-terminating output, possibly with low encoded weights. In fact, there is some possibility that such design would unnaturally amplify the effects of these higher-weight sequences [8]. An interesting extension of our current results will be to consider several sequences of low-weight simultaneously in order to overcome the disadvantages from the self-terminating input sequences as much as possible. We are currently investigating how to select an appropriate set of error patterns for these low-weight sequences that need to be broken up.

ACKNOWLEDGMENT

This work has been supported, in part, by the National Science Foundation (NCR-9706045), the New Jersey Center for Wireless Technology (NJCWT), Asahi Chemical Industry Co., the Ogasawara Foundation on Science and Engineering, and Mitsubishi Electric Information Technology Center.

REFERENCES

[1] C. Berrou, A. Glavieux, and P. Thitimasjshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes(1)," Proc. of 1993 IEEE Int. Conf. on Commun.(ICC'93), pp.1064-70, May 1993.

[2] P. Jung and M. NaBhan, "Performance evaluation of turbo codes for short frame transmission systems," Electron. Letters, vol. 30, pp.111-3, Jan. 20th 1994.

[3] P. Jung, "Comparison of turbo-code decoders applied to short frame transmission system," IEEE J. Select. Areas Commun., vol. 14, pp.530-7, Apr. 1996.

[4] J. Hokfelt and T. Maseng, "Methodical Interleaver Design for Turbo Codes," International Symposium on Turbo Codes & Related Topics, pp.212-215, Brest, France, September 1997.

[5] A.K. Khandani, "Optimization of the interleaver structure for turbo codes," Proc. of the 6th Canadian Workshop on Information Theory (CWIT'99), pp.25-28, Kingston, Ontario, June 15-18, 1999.

[6] A.S. Barbulescu and S.S. Pietrobon, "Interleaver design for turbo codes," Electron. Lett., vol.30, No.25, pp.2107-2108, Dec. 1994.

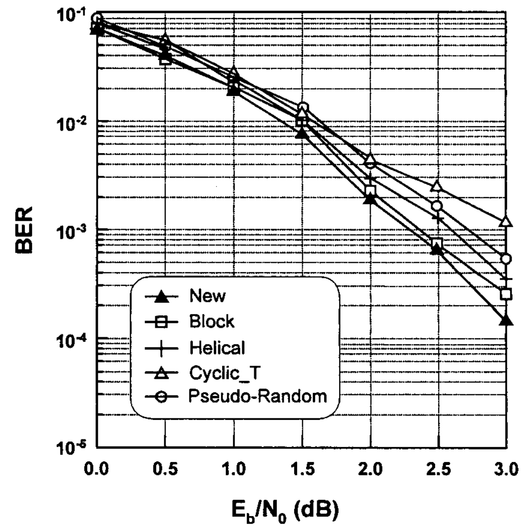
[7] D. Divsalar and F. Pollara, "Turbo codes for PCS applications," Proc. of 1995 IEEE Int. Conf. on Commun.(ICC'95), pp.54-9, May 1995.

[8] S. Dolinar and D. Divsalar, "Weight distributions for turbo codes using random and nonrandom permutations," TDA PR 42-122, pp. 56-65, August 15, 1995.

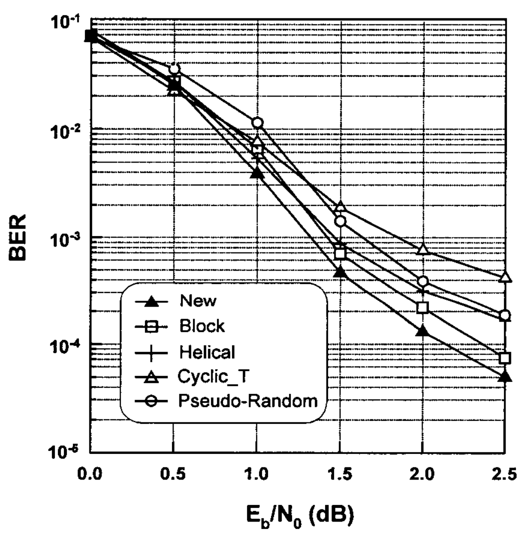
[9] A.S. Barbulescu and S.S. Pietrobon, "Terminating the trellis of turbo-codes in the same state," Electron. Lett., vol. 31, pp. 22-3, Jan. 5th 1995.

[10] J.Y. Couleaud, "High gain coding scheme for space communications," Final Year Project, Univ. of South Australia, 1995

[11] W.J. Blackert, E.K. Hall, and S.G. Wilson, "An upper bound on turbo code free distance," Proc. of 1996 IEEE Int. Conf. on Commun.(ICC'96), pp. 957-61, Dallas, TX, USA, May 1996.



(a) $N = 64$ bits



(b) $N = 256$ bits

Fig. 7 The BER performance of the new and other interleavers