This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the WCNC 2007 proceedings.

Binary Turbo Coding with Interblock Memory

Chia-Jung Yeh*, Yeong-Luh Ueng⊤, Mao-Chao Lin‡, and Ming-Che Lu§

*Graduate Institute of Communication Engineering, National Taiwan Univ., Taipei, Taiwan, R.O.C.
⊤Dept. of Electrical Engineering, National Tsing Hua Univ., Hsinchu, Taiwan, R.O.C.
‡Dept. of Electrical Engineering, National Taiwan Univ., Taipei, Taiwan, R.O.C.
§Novatek Microelectronics Corp., Hsinchu, Taiwan, R.O.C.

***Abstract*— We investigate the performance of binary codes T constructed from turbo coding with interblock memory. The encoding of T is implemented by serially concatenating a multiplexer, a multilevel delay processor, and a signal mapper to the encoder of a conventional binary turbo code C. With such a construction, in T, there is some irregularity for the code bits in C. To provide more variety of irregularity, we can construct T_C which is obtained by passing only a fraction of C through a multilevel delay processor and a signal mapper. We propose iterative decoding between adjacent codewords (IDAC), which provides error performance much better than the iterative decoding within a single codeword (IDSC). Simulation shows that T can have a lower error floor than C for either short or long code length. In some cases, T_C can provide better error floors and waterfall regions than C.**

Index Terms : turbo codes, concatenated codes.

I. Introduction

In the last decade, random-like codes such as binary turbo codes [1], regular low-density parity-check (LDPC) codes [2], and turbo trellis coded modulation (TTCM) [4], have pushed the error performances of channel coding close to the Shannon limit. There is a gap between capacity and the iterative decoding threshold (or pinch-off SNR limit) in these capacity-approaching codes. This gap can be narrowed by introducing irregularity to the code bits. Irregular LDPC codes [3] and irregular turbo codes [12] are such examples. Although these codes can approach the Shannon limits for very long codes, there is still some room for improvement if the code lengths and coding complexity are taken into consideration. For example, in [5], a multiple turbo code (MTC) which employs multiple interleavers and constituent codes was proposed. MTC can provide the flexibility of designing codes with low state complexity and good error performance in the waterfall and error-floor region.

In [7], a coded modulation scheme which is implemented by serially concatenating a multiplexer, a multilevel delay processor, and an 8PSK signal mapper to the encoder of a conventional binary turbo code C was proposed, where the delay processor is a rate one convolutional code with a transfer function matrix which is diagonal. With the introduction of the delay processor and the signal mapper, memory is introduced between two adjacent turbo code words. Hence the resultant coding scheme is a turbo coded 8PSK scheme with interblock memory.

In this paper, we investigate the case of binary codes T constructed from turbo coding with interblock memory, in which we replace the 8PSK signal mapper in [7] by a signal mapper with binary output. With such a construction, the output of a conventional binary turbo code C is split into multiple streams, and then each stream undergoes a different delay. The different delayed streams are linearly combined to form T. This design allows streams with different delays to have different levels of protection. Hence, in T, there is some irregularity for the code bits in C. To provide more variety of irregularity, in this paper, we can construct T_C which employs various degrees of interblock memory. T_C is a mixed design of T and C for which various fractions in the mixture can be employed. In this way, we can obtain binary codes with various degrees of complexity, decoding delay, and the error performance in error-floor and waterfall regions. In particular, there are two specific constructions of special interest. One is the pure binary turbo coding with interblock memory, i.e., T, and the other is T_C which is constructed by passing only a half of C through a multilevel delay processor and a signal mapper. The error floors of these two constructions can be roughly estimated by the associated effective minimum distances [9]. The pinch-off signal-to-noise ratios (SNRs) for both constructions can also be derived by the EXIT (extrinsic information transfer) charts [10] under some assumption.

We can use a two-stage decoding which employs iterative decoding within a single codeword (IDSC) of C used in [7] to decode T. Since the interblock memory is introduced between adjacent codewords of C, in this paper, we propose an improved decoding which is referred to as iterative decoding between adjacent codewords (IDAC) of C. Using IDAC, we can achieve much better error performance for short-to-moderate code lengths and slightly better error performance for long code lengths with some additional complexity as compared to using IDSC. Simulation for both the additive white Gaussian noise (AWGN) channel and the independent Rayleigh fading channel shows that T can provide much lower error floors than that of C for short-to-long code lengths. In some cases, T_C can provide better error floors and waterfall regions than C.

II. Binary Turbo Coding with Inter-Block Memory

A. Encoding

Fig. 1 shows the encoding structure of binary turbo coding with interblock memory which is constructed from a rate-1/2 binary turbo code C with a K-bit interleaver and

identical constituent codes RSC1 and RSC2. Denote the t-th code word of C as $\bar{a}(t)$= $[\bar{u}(t), \bar{p}(t)]$, where $\bar{u}(t)$=[$u(t,0)$, $u(t,1)$, $\cdots$, $u(t,K-1)$] and $\bar{p}(t)$=[$p_1(t,0)$, $p_2(t,1)$, $p_1(t,2)$, $p_2(t,3)$, $\cdots$, $p_1(t,K-2)$, $p_2(t,K-1)$] are the K-bit message and parity vectors, respectively. Noticeably, $p_1(t,i)$ and $p_2(t,i)$, $i = 0,1,\cdots,K-1$, are parity bits from RSC1 and RSC2, respectively. The turbo coded sequence $\overrightarrow{a}$=$\{\cdots,\bar{a}(t),\bar{a}(t+1),\cdots\}$ is sequentially processed by the multiplexer and the delay processor to produce the associated output sequences $\overrightarrow{v} = \{\cdots, \bar{v}(t), \bar{v}(t+1), \cdots\}$ and $\overrightarrow{s} = \{\cdots, \bar{s}(t), \bar{s}(t+1), \cdots\}$, respectively, where $\bar{v}(t)$=[$\hat{v}(t,0)$, $\hat{v}(t,1)$, $\cdots$, $\hat{v}(t,\lambda-1)$], $\bar{s}(t)$=[$\hat{s}(t,0)$, $\hat{s}(t,1)$,$\cdots$, $\hat{s}(t,\lambda-1)$], and each of $\hat{v}(t,k)$=[$v_1(t,k),\cdots,v_m(t,k)$] and $\hat{s}(t,k)$=[$s_1(t,k),\cdots,s_m(t,k)$] is a binary m-tuple. Note that $m\lambda$=$\frac{K}{R}$. The sequence $\overrightarrow{s}$ is then processed by a memoryless signal mapper to produce the output sequence $\overrightarrow{z}$ = $\{\cdots,\bar{z}(t)$, $\bar{z}(t+1)$, $\cdots\}$, where $\bar{z}(t)$=[$\hat{z}(t,0)$, $\hat{z}(t,1)$,$\cdots$, $\hat{z}(t,\lambda-1)$] and $\hat{z}(t,k)\in\ \Omega$, Ω=$\{0,1\}^m$, and $\hat{z}(t,k)$=[$z_1(t,k)$, $\cdots$, $z_m(t,k)$] is an m-bit symbol. The resultant code T takes $\overrightarrow{z}$ as its code sequence. For an m-level delay processor, the relationship between $v_j(t,k)$ and $s_j(t,k)$ is $s_j(t,k) = v_j(t-(m-j),k)$, $1 \le j \le m, k = 0,1,\cdots,\lambda-1$. The relationship between the input and output of the signal mapper is $\hat{z}(t,k) = \hat{s}(t,k)M$, where M is an $m\times m$ nonsingular matrix. For T, the insertion of delay processor and signal mapper introduces interblock memory since $\bar{z}(t)$ and $\bar{z}(t+1)$ are correlated.

B. Two specific constructions

Let C be a rate 1/2 binary turbo code. Let $m = 2$, $\Omega = \{0,1\}^2$ and $M = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$, we can have the following two constructions.

Construction I: Use the structure shown in Fig. 1. The relation among $\overrightarrow{s}$, $\overrightarrow{v}$ and $\overrightarrow{a}$ is given in Fig. 2, where $v_1(t,k) = u(t,k)$, $v_2(t,k) = p_1(t,k)$ for even k, and $v_2(t,k) = p_2(t,k)$ for odd k. The resultant code T is a rate-1/2 binary code. □

We can modify T to a construction, T_C, which is obtained by the combination of two parts. The first part is obtained by passing only a fraction, P_{IB}, $0 \le P_{IB} \le 1$, of the code bits of C through the delay processor and signal mapper and the second part is the other fraction, i.e., $1 - P_{IB}$ of the code bits of C. Clearly, T is a special case of T_C with P_{IB} = 1. Another case of special interest is the following construction.

Construction II: This construction is T_C with P_{IB} = 0.5, which is the same as Construction I except that $u(t,1)$, $u(t,3)$,$\cdots$, and $p_2(t,1)$, $p_2(t,3)$,$\cdots$ are not processed by the delay processor and the signal mapper. $u(t,0)$, $u(t,2)$,$\cdots$, and $p_1(t,0)$, $p_1(t,2)$,$\cdots$ are still processed by these two processing units. □

C. Iterative decoding within a single codeword (IDSC) [7]

We illustrate IDSC for Construction I. Let $\bar{y}(t)$ be the received word as $\bar{z}(t)$ is transmitted. For $j = 1,2$, write $\bar{v}_j(t)$=[$v_j(t,0)$, $v_j(t,1)$, $\cdots$, $v_j(t,\lambda-1)$]. Now consider the decoding of $\bar{a}(t)$. The decoder of IDSC consists of a MAP demapper and a turbo decoder of C. Suppose the extrinsic L-values (or log-likelihood ratios) $L_{D,e}(\bar{a}(t-1)) = L_{D,p}(\bar{a}(t-1)) - L_{D,c}(\bar{a}(t-1))$ for bits in $\bar{a}(t-1)$ have been obtained from the decoding of $\bar{a}(t-1)$, where $L_{D,p}(\bar{a}(t-1))$ and $L_{D,c}(\bar{a}(t-1))$ are the associated $a\ posteriori$ and channel values, respectively.

- Step 1 The demapper computes the $a\ posteriori$ L-values $L_{M,p}(\bar{v}_2(t))$ of bits in $\bar{v}_2(t)$ with the received word $\bar{y}(t)$ and the $a\ priori$ L-values $L_{M,a}(\bar{v}_1(t-1))$ which can be obtained from $L_{D,e}(\bar{a}(t-1))$. Similarly, the demapper computes $L_{M,p}(\bar{v}_1(t))$ with $\bar{y}(t+1)$ and $L_{M,a}(\bar{v}_2(t+1)) = 0$. In the calculation of $L_{M,p}(\bar{v}(t))$, $L_{M,a}(\bar{v}(t))$ is zero.
- Step 2 The turbo decoder of C uses $L_{M,p}(\bar{v}(t))$ as channel value $L_{D,c}(\bar{a}(t))$ for bits in $\bar{a}(t)$ to recover bits of $\bar{a}(t)$ and obtain $L_{D,e}(\bar{a}(t))$ which is stored for the calculation of $L_{M,p}(\bar{v}(t+1))$.

D. Iterative decoding between adjacent codewords (IDAC)

For turbo coded 8PSK scheme with interblock memory, IDSC results in large error coefficients [7]. Using IDSC to decode T_C results in the same problem. The error coefficients can be reduced by using iterative decoding between adjacent codewords (IDAC) which iteratively decodes two adjacent turbo code words, $\bar{a}(t)$ and $\bar{a}(t+1)$. We illustrate the decoding of $\bar{a}(t)$ for Construction I. Suppose that $L_{D,e}(\bar{a}(t-1))$ has been obtained.

- Step 1 With $L_{D,e}(\bar{a}(t)) = 0$ (or equivalently $L_{M,a}(\bar{v}(t)) = 0$), we use IDSC to decode $\bar{a}(t+1)$ and obtain $L_{D,e}(\bar{a}(t+1))$. Noticeably, in the demapper, the calculation of $L_{M,p}(\bar{v}(t+1))$ is based on $L_{M,a}(\bar{v}(t)) = 0$.
- Step 2 With $L_{D,e}(\bar{a}(t+1))$ obtained in Step 1, $L_{D,e}(\bar{a}(t-1))$, and $L_{M,a}(\bar{v}(t))$=0, the demapper calculates $L_{M,p}(\bar{v}(t))$. The decoder of C use $L_{M,p}(\bar{v}(t))$ as $L_{D,c}(\bar{a}(t))$ to decode $\bar{a}(t)$ and obtain $L_{D,e}(\bar{a}(t))$.
- Step 3 We use IDSC to re-decode $\bar{a}(t+1)$ and update $L_{D,e}(\bar{a}(t+1))$ with $L_{D,e}(\bar{a}(t))$ obtained in Step 2.
- Step 4 With updated $L_{D,e}(\bar{a}(t+1))$ obtained in Step 3, $L_{D,e}(\bar{a}(t-1))$, and $L_{M,a}(\bar{v}(t))$=0, the demapper calculates $L_{M,p}(\bar{v}(t))$. The decoder of C uses updated $L_{M,p}(\bar{v}(t))$ as new channel values $L_{D,c}(\bar{a}(t))$ to decode $\bar{a}(t)$ and obtain $L_{D,e}(\bar{a}(t))$.
- Step 5 With a certain number of repeating Steps 3 and 4, we can decode $\bar{a}(t)$ and obtain $L_{D,e}(\bar{a}(t))$.

Throughout this paper, the Max-Log-MAP algorithm with correction factors [11] is employed and N_I iterations are used in the turbo decoding of C. In addition, one iteration of IDAC consists of Steps 1 and 2 (or Steps 3 and 4). For IDAC, the number of iterations between adjacent codewords is denoted by N_{IDAC}.

III. Performance Analysis

For T_C, we can use the effective minimum distances [9] to estimate the error performance in the error-floor region and use EXIT charts to analyze the pinch-off limits in the

waterfall region. In the following, analysis based on 4-state RSC1 and RSC2 with generator matrix $(1, 5/7)_8$ in additive white Gaussian noise (AWGN) channels is given.

A. Effective minimum distances for asymptotic performances

In [9], the effective minimum distance $d_{2,min}$ which equals $2+z_{min}$ of C is defined to be the minimum Hamming weight generated by weight-2 message sequences, where z_{min} is the associated minimum parity-check weight.

For Construction I, it can be shown that the Hamming weight of $\overrightarrow{z}$ generated by a weight-$d_{2,min}$ code sequence $\overrightarrow{a}$ of C is $2 + 2z_{min}$ which is larger than $d_{2,min}$ = $2 + z_{min}$ if $z_{min} > 0$. Hence Construction I is expected to achieve lower BER at high SNRs as compared to C. Consider the case of Construction II. Suppose that the weight of $\overrightarrow{p}_1$ for a weight-$d_{2,min}$ turbo coded sequence $\overrightarrow{a}$ is z_ℓ and hence the weight of $\overrightarrow{p}_2$ is $z_{min} - z_\ell$. Regardless of the positions of nonzero message bits, the output sequence $\overrightarrow{z}$ will be with weight $2 + z_\ell + z_{min}$. Hence the asymptotic performance of Construction II is expected to be worse than that of Construction I and better than that of C if $z_{min} > z_\ell > 0$.

It can also be shown that using IDSC for Construction I and II, the associated multiplicity will be increased by a factor of 2^2 and a factor of 2^w respectively, where w is the weight of $\bar{v}_1(t)$. If IDAC is used, the increased factor can be reduced and can be close to 1 if the feedback information $L_{D,e}(\bar{a}(t+1))$, in the final iteration of step 4 is significant. Hence, using IDAC for either Construction I or II, the waterfall region can be closer to the Shannon limit as compared to using IDSC.

Estimated asymptotic performances based on the calculated distance spectra which are generated by weight-2 and weight-3 message sequences and the equation of BER given in [8] are shown in Fig. 3. From Fig. 3, the asymptotic performances of Constructions I and II are dominated by low weight message sequences like the case of conventional turbo codes C.

B. Analysis of EXIT charts for pinch-off limits

The performance of Constructions I and II in the water-fall region can be analyzed by using EXIT charts. Let $I_a(\bar{c})$ denote the mutual information, $I_a[c, L_a(c)]$, between bit c in $\bar{c}$ and its $a\ priori$ L-value $L_a(c)$. Similarly, $I_e(\bar{c})$ denotes the mutual information, $I_e[c, L_e(c)]$, between bit c in $\bar{c}$ and its extrinsic L-value $L_e(c)$. These EXIT charts of $\bar{a}(t)$ are calculated based on the conditions that $K = 2097152$ and $I_a(\bar{a}(t-1)) = 1$. In addition, $I_a(\bar{a}(t+1)) = 0$ and $I_a(\bar{a}(t+2)) = 0$ are assumed for IDSC and IDAC, respectively. For T_C with IDSC, we consider the information exchange between RSC1 and RSC2 in a way similar to the conventional turbo code except that the channel values $I_{D,c}(\bar{a}(t))$ for code bits of RSC1 and RSC2 are obtained from the demapper instead of obtaining from channel directly. The pinch-off SNRs for Constructions I and II using IDSC and C are 0.64, 0.54, and 0.69 dB, respectively. In Fig. 4, we consider the information exchange between $\bar{a}(t)$ and $\bar{a}(t+1)$ for Constructions I and II. We run simulation on $\bar{a}(t)$ with $N_I = 60$ to obtain $I_{e1}(\bar{v}_1(t))$ for various $I_{a1}(\bar{v}_2(t+1))$. Similarly, we run simulation on $\bar{a}(t+1)$ with $N_I = 60$ to obtain $I_{e2}(\bar{v}_2(t+1))$ for various $I_{a2}(\bar{v}_1(t))$. The pinch-off SNRs for Constructions I and II using IDAC are 0.69 and 0.50 dB, respectively. Compared to the pinch-off SNRs obtained using IDSC, we observe that the advantage of information exchange between $\bar{a}(t)$ and $\bar{a}(t+1)$ will diminish for long code lengths and will sometimes provides negative effect. This trend can also be observed from the BER obtained from simulation for various code lengths.

It is known that introducing the irregular structure allows us to construct LDPC codes with improved pinch-off performance. The usage of the delay processor and the signal mapper in T enables different delayed bit streams multiplexed from C to obtain different levels of protection, which is a kind of irregularity. For T_C, the mixture of T and C allows more irregularity. ¿From this point of view, the introduction of memory between turbo codewords not only is a method to artificially increase the interleaver size and but also is a method of providing irregularity. Hence, improved pinch-off SNRs as compared to C can be achieved.

IV. Performance Evaluation

BER results in AWGN channels are given in Section IV.A to IV.E. The results of independent Rayleigh fading channels are given in Section IV.F. The constituent codes used in these examples are 4-state and 16-state codes with generator matrices given by $(1, 5/7)_8$ and $(1, 21/37)_8$, respectively.

A. BER results for short-to-moderate code lengths

Simulation results for 4-state T_C with K=1024 using either IDSC or IDAC and conventional 4-state turbo code C are given in Fig. 5. Significant gain can be obtained by using IDAC as compared to IDSC for either Construction I or II. For IDAC, N_{IDAC}=2 is sufficient to achieve good error performances. T_C has a longer decoding delay (DL) as compared to C based on the same K. Decoding delay (decoding latency) is calculated by assuming zero processing delay. Now consider DL for IDSC. For Construction I with a K-bit interleaver, we need $\bar{y}(t)$ and $\bar{y}(t+1)$ to calculate $L_{M,p}(\bar{v}(t))$. Hence, DL is $2K$ message bits. Similarly, DL for Construction II is $\frac{3K}{2}$ message bits. DL for Constructions I and II with IDAC are $3K$ and $\frac{5K}{2}$, respectively.

In general, T_C has an increased decoding complexity as compared to C based on the same K. For IDSC, the computational complexity in the calculation of $L_{M,p}(\bar{v}(t))$ is negligible as compared to that of the iterative decoding of C for either Construction I or II. Hence, the computational complexity of IDSC of T_C is only slightly higher than that of C. For IDAC, the computational complexity of T_C is about $2N_{IDAC}$ if the computational complexity of C is normalized to be 1. Hence, the computational complexities of 4-state Construction I (or II) using IDAC with $N_{IDAC} = 2$ are roughly the same as those of conventional 16-state turbo codes. From Fig. 5, we find that the error performances of 4-state Constructions I and II using either IDSC or IDAC are better than those of the conventional binary turbo code based on the conditions of similar decoding

complexities and delays, and moderate-to-high SNR for the 4-state and 16-state constituent codes used in this paper.

B. BER results for various degrees of interblock memory and various constituent codes

Fig. 6 shows the simulation results for 4-state T_C with different P_{IB}. We use the same two-level delay processor but with different λ, and the same signal mapper as those used in Construction I. Simulation results indicate that codes with larger P_{IB} can achieve better performance at high SNR and worse error performance at low SNR. Similar behavior can be observed for 16-state T_C. Due to space limitation, the simulation results are not shown here.

C. BER results for long code lengths

Simulation results for Constructions I and II with long interleavers (K=32768) using either IDSC or IDAC and conventional turbo codes C are given in Fig. 7. Slight gain can be obtained by using IDAC as compared to IDSC for either Construction I or II. Based on similar decoding delay and the same state complexity, either Construction I or II can achieve better error performance as compared to C. In addition, using a 4-state Construction II with $K = 32768$ and $N_{IDAC} = 1$, we can have BER $\approx 10^{-6}$ at $E_b/N_o = 0.75$ dB. This performance is similar to that of the 16-state turbo code [1] with K=65536. The computational complexities of Construction II using IDAC with $N_{IDAC} = 1$ are roughly the same as those of conventional 8-state turbo codes. Moreover, from curve B3 in Fig. 7, a 4-state conventional turbo code with $K = 98304$ can achieve a BER of 5×10^{-6} at $E_b/N_o = 0.90$ dB. However, from Fig. 6, 4-state Construction II ($P_{IB} = \frac{4}{8}$) with $K = 8192$ can achieve a BER of 5×10^{-6} at $E_b/N_o = 0.85$ dB. Our construction can achieve better error performance as compared to conventional turbo code with much shorter decoding delay.

D. Performance of turbo cliff

To see the pinch-off SNRs (or turbo cliff) for Constructions I and II, we perform simulations with very long interleavers ($K = 262144$). The results are also given in Fig. 7. We see that 4-state Construction II can achieve a BER of 10^{-6} at $E_b/N_o = 0.6$ dB. The pinch-off SNR obtained from the EXIT charts for conventional 4-state turbo code C is 0.69 dB. Hence, T_C constructed from turbo coding with interblock memory has better pinch-off SNR as compared to C. For EXIT charts given in Section III.B, $I_a(\bar{a}(t-1)) = 1$ is assumed. In the BER simulation, we do not use such assumption and N_I, and N_{IDAC} are limited. Hence, the pinch-off SNRs for Constructions I and II obtained from the analysis of EXIT charts are better than those obtained from the BER simulation with $K = 262144$.

E. Comparisons with multiple turbo codes (MTC) and stream-oriented turbo code (SOTC)

MTC can achieve good error performance with low state complexity [5]. In [5], a rate 1/2 MTC using four 2-state constituent encoders was investigated and comparisons with 3GPP (Third-Generation Partnership Project) 8-state turbo codes were given. MTC, in general, needs more iterations to achieve the same BER performance of the 3GPP code and hence the 2-state MTC can achieve the same BER with a slightly lower computational complexity as compared to the 8-state 3GPP code [5]. The simulation results of this 2-state MTC with $K = 1024$ obtained from Fig. 6 of [5] are also included in Fig. 5. We find that 4-state Construction I and Construction II with K=1024 and $N_{IDAC} = 2$ can achieve BER=10^{-6} and BER=10^{-5} at $E_b/N_o = 1.6$ dB and $E_b/N_o = 1.4$ dB, respectively, while this MTC with K=1024 can achieve BER=10^{-6} at $E_b/N_o = 1.8$ dB and BER=10^{-5} at $E_b/N_o = 1.68$ dB.

In [6], a binary stream-oriented turbo code (SOTC) which is implemented by using a convolutional interleaver in a conventional turbo encoder was proposed. Although convolutional interleavers are employed in both SOTC and T_C and the outputs of both T_C and SOTC are continuous, the encoding of T_C is totally different from SOTC. T_C has explicit boundary and hence is still suitable for package transmission while the goal of SOTC is to avoid data framing and is suitable for stream-oriented applications.

F. BER results in the independent Rayleigh fading channels

Fig. 8 shows the simulation results for 4-state T_C with short interleaver length using either IDSC or IDAC and conventional 4-state turbo code C. Significant gain can be obtained by using IDAC as compared to IDSC for either Construction I or II. From Fig. 8, we find that the error performances of 4-state Constructions I and II using either IDSC or IDAC are better than those of the conventional binary turbo code based on the conditions of similar decoding complexities and delays, and moderate-to-high SNR for the 4-state and 16-state constituent codes used in this paper. We also simulate the case of long interlevaer length. T_C has similar advantage over C as in the case of AWGN channels. For example, 4-state Construction II with K=262144 and $N_{IDAC} = 3$ can achieve a BER of 2×10^{-7} at $E_b/N_o = 2.37$ dB while 4-state C with K=262144 can achieve a BER of 3×10^{-6} at $E_b/N_o = 2.6$ dB.

V. CONCLUSION

In this paper, the performance of binary turbo coding with interblock memory is investigated. Binary codes, T_C, can be constructed based on various degrees of interblock memory and decoded by using either IDSC or IDAC. Simulation shows that T_C can have a lower error floor than the conventional turbo code C for either short or long code length considering the associated complexity, and decoding delay. In the waterfall region, in some cases, T_C can have a lower pinch-off signal-to-noise ratio (SNR) limit than C. The concept of interblock memory can be applied to LDPC codes. Similar advantage can be obtained.

REFERENCES

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding : turbo-codes(1)," in *Proc. IEEE ICC* (Geneva, Switzerland, May 1993) pp. 1064-1070.

[2] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inform. Theory,* vol. 45, no. 2, pp. 399–431, Mar. 1997.

[3] T.J. Richardson, M.A. Shokrollahi, and R.L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inform. Theory,* vol. 47, no. 2, pp. 619–637, Feb. 2001.

[4] Li Ping, B. Bai, and X. Wang, "Low-complexity concatenated two-state TCM schemes with near-capacity performance," *IEEE Trans. Inform. Theory,* vol. 49, no. 12, pp. 3225–3234, Dec. 2003.

[5] D. J. Costello, Jr., A. Banerjee, C. He, and P. C. Massey, "A comparision of low complexity turbo-like codes", in *Proceedings of the 36th Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, California, Nov. 2002*

[6] E. K. Hall, and S. G. Wilson, "Stream-oriented turbo codes," *IEEE Trans. Inform. Theory,* vol. 47, no. 5, pp. 1813–1831, Sept. 2001.

[7] Y.L. Ueng, C.J. Yeh and M.C. Lin, "A turbo coded modulation scheme using a delay processor," 2000 International Symposium on Information Theory and Its Applications, Honolulu, Hawaii, USA, Nov. 5-8, 2000.

[8] L. C. Perez, J. Segher, and D. J. Costello, "A distance spectrum interpretation of turbo codes," *IEEE Trans. Inform. Theory,* vol. 42, no. 6, pp. 409–428, Nov. 1996.

[9] S. Benedetto and G. Montorsi, "Design of parallel concatenated convolutional codes," *IEEE Trans. Commun.,* vol. 44, no. 5, pp. 591–600, May 1996.

[10] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Trans. Commun.,* vol. 49, no. 10, pp. 1727–1737, Oct 2001.

[11] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and suboptimal MAP decoding algorithms operating in the log domain," in *Proc. ICC '95,* pp. 1009–1013.

[12] B. J. Frey, and D. J. C. Mackay, "Irregular turbocodes," in *Proc. ISIT 2000,* pp. 121, 25-30 June 2000.

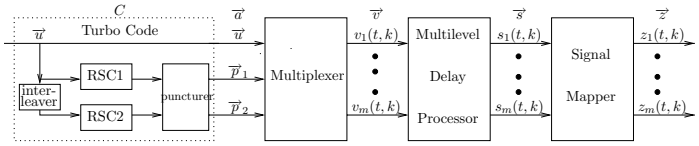

Fig. 1. Encoding structure of binary turbo coding with interblock memory.

	$\hat{s}(t,0)$	$\hat{s}(t,1)$	$\cdots$	$\hat{s}(t+1,0)$	$\hat{s}(t+1,1)$	$\cdots$	$\hat{s}(t+2,0)$	$\hat{s}(t+2,1)$
s_1	$u(t-1,0)$ = $v_1(t-1,0)$	$u(t-1,1)$ = $v_1(t-1,1)$	$\cdots$	$u(t,0)$ = $v_1(t,0)$	$u(t,1)$ = $v_1(t,1)$	$\cdots$	$u(t+1,0)$ = $v_1(t+1,0)$	$u(t+1,1)$ = $v_1(t+1,1)$
s_2	$p_1(t,0)$ = $v_2(t,0)$	$p_2(t,1)$ = $v_2(t,1)$	$\cdots$	$p_1(t+1,0)$ = $v_2(t+1,0)$	$p_2(t+1,1)$ = $v_2(t+1,1)$	$\cdots$	$p_1(t+2,0)$ = $v_2(t+2,0)$	$p_2(t+2,1)$ = $v_2(t+2,1)$

Fig. 2. Relation among sequences $\overrightarrow{s}$, $\overrightarrow{v}$ and $\overrightarrow{a}$ for Construction I.

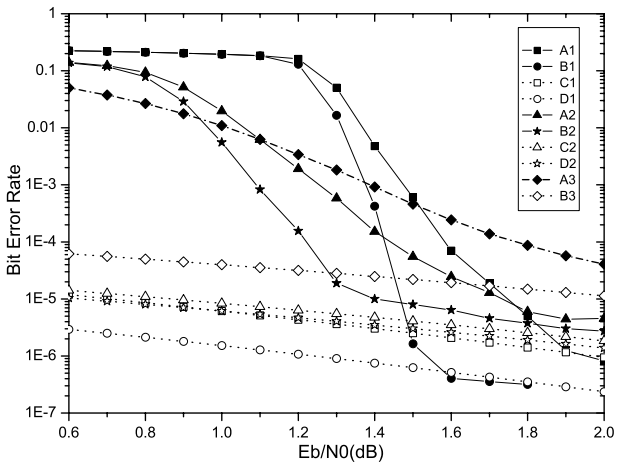

Fig. 3. Simulation and estimation results in AWGN channels for 4-state Construction I (Constr. I), Construction II (Constr. II), and conventional turbo codes with K=1024 and N_I=10. (A1): Simulation, Constr. I, IDSC. (B1): Simulation, Constr. I, IDAC, N_{IDAC}=3. (C1): Estimation, Constr. I, IDSC. (D1): Estimation, Const. I, IDAC. (A2): Simulation, Constr. II, IDSC. (B2): Simulation, Constr. II, IDAC, N_{IDAC}=3. (C2): Estimation, Constr. II, IDSC. (D2): Estimation, Constr. II, IDAC. (A3): Simulation, turbo code. (B3): Estimation, turbo code.

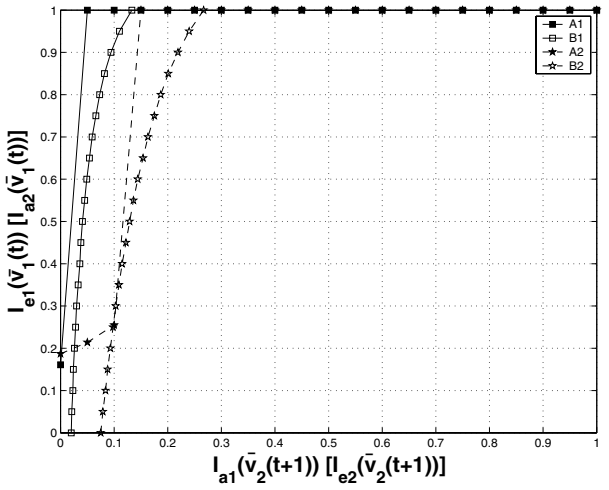

Fig. 4. EXIT charts of $\bar{a}(t)$ and $\bar{a}(t+1)$ for 4-state Constructions I and II based on IDAC and AWGN channels. (A1): Constr. I, $\bar{a}(t)$, $\frac{E_b}{N_0} = 0.69$ dB. (B1): Constr. I, $\bar{a}(t+1)$, $\frac{E_b}{N_0} = 0.69$ dB. (A2): Constr. II, $\bar{a}(t)$, $\frac{E_b}{N_0} = 0.50$ dB. (B2): Constr. II, $\bar{a}(t+1)$, $\frac{E_b}{N_0} = 0.50$ dB.

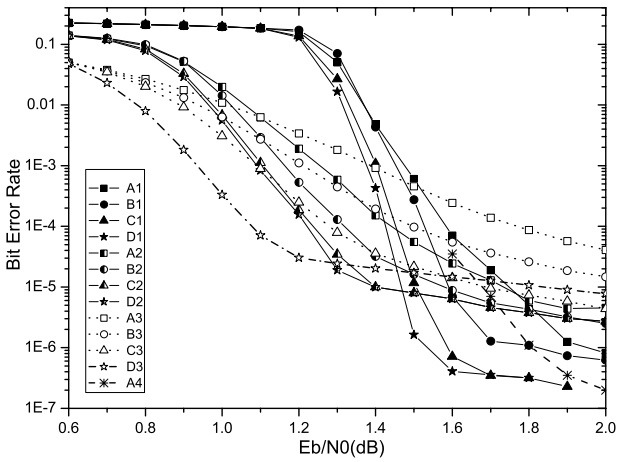

Fig. 5. Simulation results in AWGN channels for 4-state Construction I (Constr. I) with K=1024 (DL=2048 for IDSC and DL=3072 for IDAC), 4-state Construction II (Constr. II) with K=1024 (DL=1536 for IDSC and DL=2560 for IDAC), and conventional turbo codes. All the simulation results are based on with N_I=10 except (A4). (A1): Constr. I, IDSC. (B1): Constr. I, IDAC , N_{IDAC}=1. (C1): Constr. I, IDAC, N_{IDAC}=2. (D1): Constr. I, IDAC, N_{IDAC}=3. (A2): Constr. II, IDSC. (B2): Constr. II, IDAC, N_{IDAC}=1. (C2): Constr. II, IDAC, N_{IDAC}=2. (D2): Constr. II, IDAC, N_{IDAC}=3. (A3): 4-state turbo code, K=1024, DL=1024. (B3): 4-state turbo code, K=2048, DL=2048. (C3): 4-state turbo code, K=4096, DL=4096. (D3): 16-state turbo code, K=3072, DL=3072. (A4): MTC [5], K=1024, N_I=25.

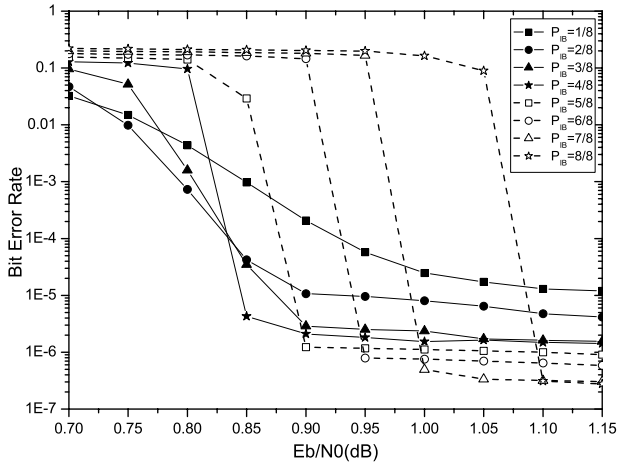

Fig. 6. Simulation results for 4-state T_C with various P_{IB} in the AWGN channels(N_I = 10, K=8192, and IDAC with N_{IDAC}=3).

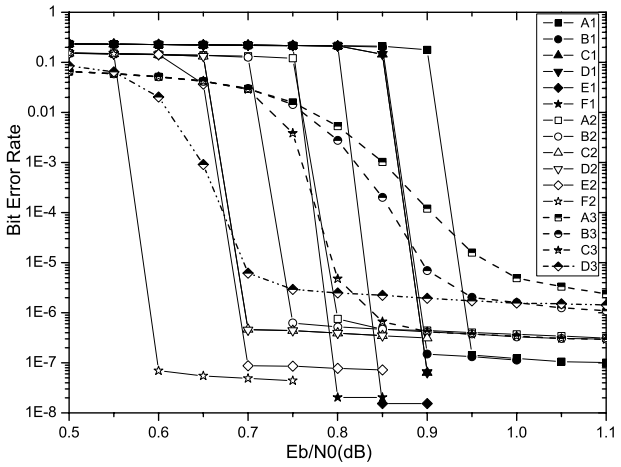

Fig. 7. Simulation in the AWGN channels for 4-state Construction I (Constr. I), 4-state Construction II (Constr. II), and conventional turbo codes with long interleavers. (A1): Constr. I, IDSC, N_I=18, K=32768. (B1): Constr. I, IDAC, N_{IDAC}=1, N_I=18, K=32768. (C1): Constr. I, IDAC, N_{IDAC}=2, N_I=18, K=32768. (D1): Constr. I, IDAC, N_{IDAC}=3, N_I=18, K=32768. (E1): Constr. I, IDSC, N_I=30, K=262144. (F1): Constr. I, IDAC, N_{IDAC}=3, N_I=30, K=262144. (A2): Constr. II, IDSC, N_I=18, K=32768. (B2): Constr. II, IDAC, N_{IDAC}=1, N_I=18, K=32768. (C2): Constr. II, IDAC, N_{IDAC}=2, N_I=18, K=32768. (D2): Constr. II, IDAC, N_{IDAC}=3, N_I=18, K=32768. (E2): Constr. II, IDSC, N_I=30, K=262144. (F2): Constr. II, IDAC, N_{IDAC}=3, N_I=30, K=262144. (A3): 4-state turbo code, N_I=18, K=32768. (B3): 4-state turbo code, N_I=18, K=98304. (C3): 4-state turbo code, N_I=30, K=262144. (D3): 16-state turbo code, N_I=18, K=65536.

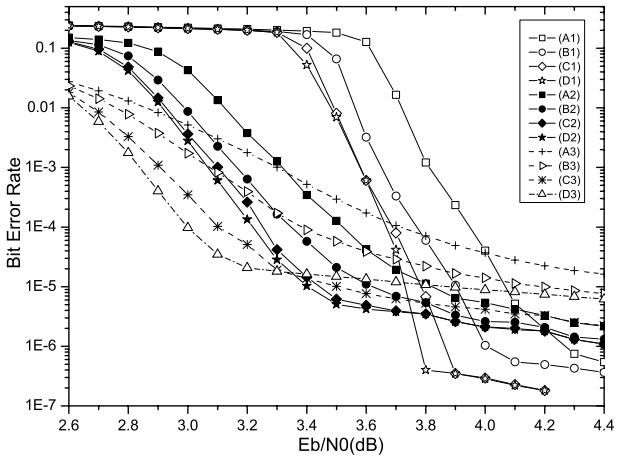

Fig. 8. Simulation results in independent Rayleigh fading channels for 4-state Construction I (Constr. I) with K=1024 (DL=2048 for IDSC and DL=3072 for IDAC), 4-state Construction II (Constr. II) with K=1024 (DL=1536 for IDSC and DL=2560 for IDAC), and conventional turbo codes. All the simulation results are based on with N_I=10. (A1): Constr. I, IDSC. (B1): Constr. I, IDAC , N_{IDAC}=1. (C1): Constr. I, IDAC, N_{IDAC}=2. (D1): Constr. I, IDAC, N_{IDAC}=3. (A2): Constr. II, IDSC. (B2): Constr. II, IDAC, N_{IDAC}=1. (C2): Constr. II, IDAC, N_{IDAC}=2. (D2): Constr. II, IDAC, N_{IDAC}=3. (A3): 4-state turbo code, K=1024, DL=1024. (B3): 4-state turbo code, K=2048, DL=2048. (C3): 4-state turbo code, K=4096, DL=4096. (D3): 16-state turbo code, K=3072, DL=3072.