Improvements and Extensions of Low-Rate Turbo-Hadamard Codes

Noriyuki Shimanuki[1], Brian M. Kurkoski[2], Kazuhiko Yamaguchi[2], Kingo Kobayashi[2]

[1]Graduate School of Information Systems, [2]Dept. of Information and Communication Engineering
University of Electro-Communications, 1-5-1 Chofugaoka, Chofu, Tokyo, Japan
{sima,kurkoski,yama,kingo}@ice.uec.ac.jp

Abstract

In this paper, we consider a number of system improvements that have been proposed for convolutional turbo codes, and apply them to turbo-Hadamard codes (THC). Our findings are as follows. 1. Asymmetrical THC, employing differing constituent codes, improves the waterfall region performance compared to symmetrical THC. 2. Punctured THC have improved performance in both the waterfall and error floor regions, relative to non-punctured codes of the same rate. 3. An LDPC code, when serially concatenated with an inner THC code, can correct low weight errors to improve the error floor performance. 4. Parallel-schedule decoding of THC can converge significantly faster than conventional serial-mode decoding, leading to a reduction in decoder complexity.

1. INTRODUCTION

Information hiding and wideband data communications are applications which can benefit from powerful low-rate error-correcting codes, such as recently proposed turbo-Hadamard codes (THC) [1]. In information hiding applications, low-rate codes can be used to reliably embed information into digitized audio or video. Wideband data communications can use low-rate error-correcting codes to replace the spreading code in traditional direct-sequence spread spectrum systems.

Hadamard codes are low-rate block codes with good distance properties, and have been used in communication systems for synchronization and bandwidth spreading. THC codes adopt Hadamard codes so that they may be used in turbo codes, and are briefly reviewed in Section 2. It is reported that symmetric THC codes can potentially achieve successful decoding at $E_b/N_0 \approx -1.3$ dB, which is 0.29 dB away from the low-rate Shannon limit [2]. However, these codes have a high error floor, and are complex to decode.

This research was partially supported by the Ministry of Education, Science, Sports and Culture; Grant-in-Aid for JSPS Fellows, number 00004908, 2004; and Grant-in-Aid for Scientific Research (C) number 16560324, 2004.

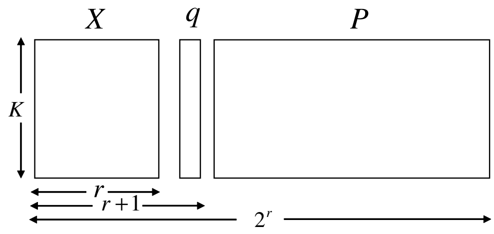

Figure 1: Structure of a convolutional-Hadamard codeword.

To address these problems, we take a number of system improvements studied for convolutional turbo codes, and apply them to THC codes. In Section 3.1, we show that asymmetric THC codes have better performance in the waterfall region than symmetric codes. Takeshita, et al. showed that asymmetric convolutional turbo codes have better performance in both the waterfall and error floor regions [3]. Various asymmetric designs have been presented in [9] [10] [11] [12].

For convolutional turbo codes, punctured multiple turbo codes outperform unpunctured turbo code with only two component codes [4] [9]. This is possible because increasing the number of component codes increases the interleaving gain, and leads to a lower error floor even though the codes are punctured. In Section 3.2, we show that this applies to punctured THC codes as well. Favorable performance-complexity tradeoffs are obtained by using these punctured codes, which obtain good performance on trellises with fewer states.

We consider the serial concatenation of an outer LDPC code and an inner THC code in order to reduce the observed error floors. Parallel-concatenated turbo codes have a relatively high error floor due to a small multiplicity of low-weight codewords. For improving this error floor, various methods have been presented. In [5], a scheme to improve the error floor of convolutional turbo codes by using high rate BCH codes was

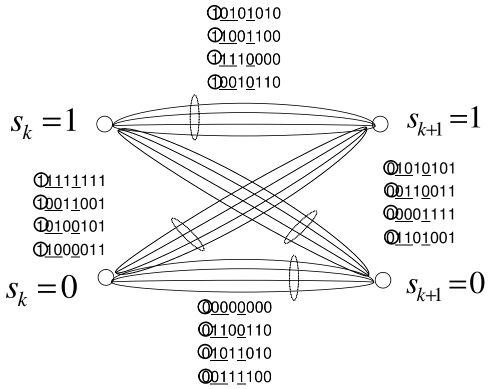

Figure 2: Trellis section for a convolutional Hadamard code with $\widehat{C}(1, 1/3)_8$ and $r = 3$.

proposed. In [8] and [7], the serial concatenation of an outer LDPC code and inner turbo code was proposed. However, it was shown that the concatenation of an LDPC and turbo code gave worse performance than a Reed-Solomon code at moderate block length [7]. In Section 3.3, we show that for low-rate THC codes, the concatenation with an outer LDPC code is effective at correcting low-weight errors. In particular, we observe error floors for THC systems, but these error floors disappear when a high-rate LDPC code is added. As a side benefit, the LDPC code acts as a soft-input stopping condition on the THC turbo iterations when the LDPC decoder detects a valid codeword.

In Section 3.4, we show that for decoding THC codes, a parallel decoding schedule has better convergence performance than the usual serial decoding schedules. This is consistent with results for multiple convolutional turbo codes, where parallel-mode turbo decoding converges faster than serial-mode decoding [6].

Section 4 is the conclusion.

2. TURBO-HADAMARD CODES

The component code in THC codes, convolutional Hadamard codes (CHC) are low-rate codes with good distance properties, and have high output Hamming weight for low input weight sequences. The vector $\mathbf{X}$ of N information bits is split into $K = N/r$ blocks of r bits $\mathbf{x}_k, k = 1, 2, \ldots, K$, with $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_K\}$. A single parity bit is q'_k, is computed for each block. The bits q'_k are inputs to a rate 1, S-state convolutional code $\widehat{C}$ with a specified generator polynomial, to produce q_k. An order-r Hadamard code encodes $\{\mathbf{x}_k, q_k\}$ to produce the systematic codeword $\{\mathbf{x}_k, q_k, \mathbf{p}_k\}$, with parity $\mathbf{p}_k$.

Table 1: Asymmetric Configurations for $M = 4$

Component Encoder	Generator Polynomial	
	Code A	Code B
ENC1	$(1, 1/3)_8$	$(1, 1/3)_8$
ENC2	$(1, 3)_8$	$(1, 3)_8$
ENC3	$(1, 1/3)_8$	$(1, 3)_8$
ENC4	$(1, 3)_8$	$(1, 1/3)_8$

Fig. 1 shows the structure of the CHC codeword. The CHC code has a trellis representation, with the same number of states as the convolutional code $\widehat{C}$, and parallel transitions which correspond Hadamard codewords. Fig. 2 shows the trellis for a CHC code with convolutional code $\widehat{C}$ $(1, 1/3)_8$ and an order $r = 3$ Hadamard code. The circled, underscored and remaining bits represent q_k, $\mathbf{x}_k$ and $\mathbf{p}_k$, respectively. The CHC systematic bits $\mathbf{x}_k$ are not transmitted, as these are already sent as the systematic bits of the THC code.

THC codes are a parallel concatenation of M CHC codes, separated with $M - 1$ interleavers with length N. The decoding of THC codes is performed as in multiple parallel turbo codes, where extrinsic information is shared between the constituent codes' decoders. and is decoded using the BCJR algorithm. The trellis has a large number of parallel transitions, which are computed using the structural properties of the Hadamard code. Turbo decoding proceeds for I_t iterations. Refer to [1] for details on THC and CHC codes.

3. PROPOSED IMPROVEMENTS

3.1. Asymmetric THC

To compare the performance of symmetric and asymmetric codes, we consider $r = 5, M = 4, S = 2, N = 1000$ THC codes. We use a symmetric code with four $(1, 1/3)_8$ component convolutional codes, and two asymmetric designs, Code A and Code B which use mixtures of $(1, 3)_8$ and $(1, 1/3)_8$ component convolutional codes, as indicated in Table 1.

Simulation results are shown in Fig. 3. As can be seen, both asymmetric Codes A and B achieve a gain of nearly 0.1 dB in the waterfall region, making asymmetric codes a promising candidate to improve the noise thresholds found in [2].

3.2. Punctured THC

In this section, we compare punctured and unpunctured THC of the same rate. A pseudo-random puncturing pattern is used, where one parity bit is randomly

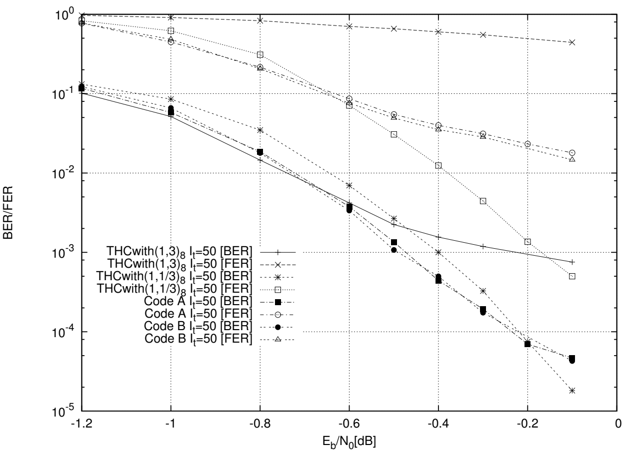

Figure 3: BER and FER of $R = 5/113$ asymmetric and symmetric THC codes, with $M = 4$, $r = 5$, $N = 1000$, $I_t = 50$.

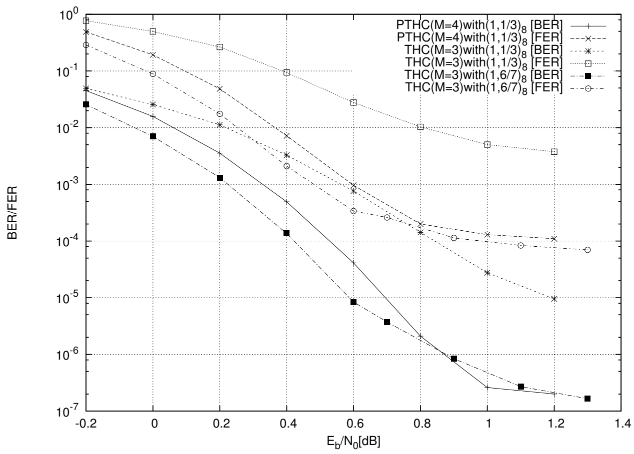

Figure 4: BER and FER performance of $R \approx 3/18$ punctured and unpunctured codes with $N = 999$, $r = 3$, $I_t = 50$. For punctured, $M = 4$, generator polynomial $(1, 1/3)_8$ and for unpunctured, $M = 3$, generator polynomial $(1, 1/3)_8$ and $(1, 6/7)_8$.

punctured from the vector $\mathbf{p}_k$ of each component decoder. The parity-check bits q_k were not punctured, as this disproportionately reduced the strength of the strength of the BCJR transition metrics. This pattern gives the BER performance averaged over all possible puncturing patterns. Further improvements may be possible by choosing specific puncturing patterns. If P is the number parity bits punctured per trellis section, then the code rate R is:

$$R = \frac{r}{r + M(2^r - P - r)}, \qquad (1)$$

where $P = 0$ gives the code rate for an unpunctured code.

We evaluate $M = 4$ THC codes punctured with $P = 1$, and unpunctured $M = 3$ THC codes, both with rate $R \approx 3/18$. It has been observed that gains in the waterfall region can be had for increasing M, for $M \leq 4$, while gains in the error floor region continue for increasing M [1].

For the $M = 4$ punctured code, an $S = 2$ generator polynomial $(1, 1/3)_8$ is used. For the punctured code, one parity bit is randomly punctured from the vector $\mathbf{p}_k$ of each component encoder. For the $M = 3$ unpunctured code, $S = 2, 4$ generator polynomials $(1, 1/3)_8$ and $(1, 6/7)_8$ are considered. For both codes, $r = 3$ and the number of decoding iterations is $I_t = 50$. The BER and FER performance is shown in Fig. 4.

For a fixed number of states $S = 2$, the punctured code significantly outperforms the unpunctured code in both the waterfall and error floor region. The punctured code achieves BER$= 10^{-5}$ at $E_b/N_0 \approx 0.7$ dB, an improvement of about 0.5 dB relative to the unpunctured code. The much lower error floor of the punctured code may be attributed to its higher interleaving gain.

We also compare the punctured code with an unpunctured code using a trellis with $S = 4$ generator polynomial $(1, 6/7)_8$. The punctured code has better performance in the waterfall region, despite having a simpler trellis.

3.3. LDPC-THC Concatenation

In this section, we consider the serial concatenation of an outer LDPC code with a THC code, so that the input to the THC code is an LDPC codeword. Fig. 5 shows the block diagram of the encoder, where L denotes the overall number of information bits.

The decoder for the hybrid LDPC-THC system is:

1. Perform one iteration of THC turbo decoding (serial mode).

2. Perform syndrome check; if the LDPC syndrome is 0, then stop decoding.

3. Perform I_s iterations of LDPC decoding using the sum-product algorithm. If after any iteration the syndrome is 0, then stop decoding.

4. If the number of turbo iterations is less than I_t, go to step 1, otherwise stop. (Extrinsic information from the sum-product algorithm is not used by the turbo decoder.)

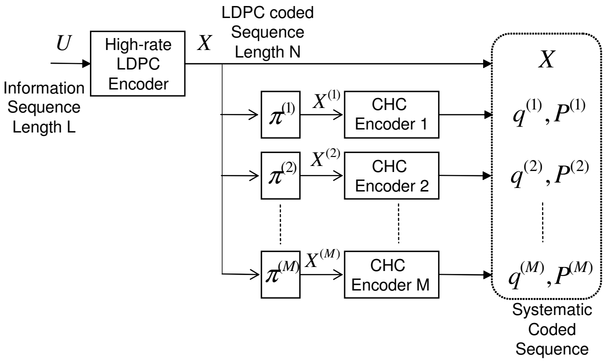

Figure 5: Encoder structure for the LDPC-THC concatenation.

This algorithm is shown as a flowchart in Fig. 6.

In simulations, we compare our proposed LDPC-THC system with a standard THC code. The LDPC-THC system uses a (3,27)-regular LDPC code with block length 999, and a THC code with $N = 999$, $M = 3$, $r = 3$, and $S = 2$ generator polynomial $(1,3)_8$. The overall rate of the hybrid concatenated code is $R = 4/27 = 0.15$. The maximum number of decoding iterations for the outer sum-product decoder is $I_s = 10$. The parameters of the standard THC code are $N = 999$, $M = 3$, $r = 3$, and $S = 2$ generator polynomial $(1,1/3)_8$, with rate $R = 3/18 = 0.17$. We consider $I_t = 5, 10, 50$ iterations.

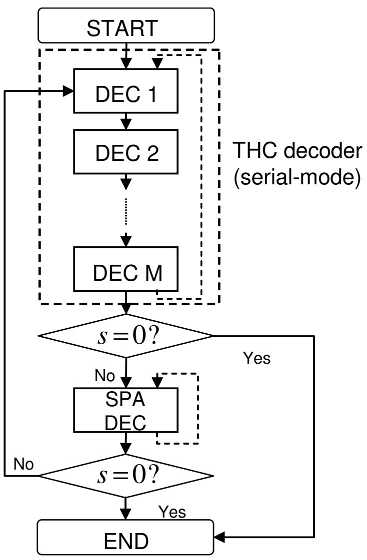

Figure 6: Flowchart for iterative decoding of LDPC-THC codes.

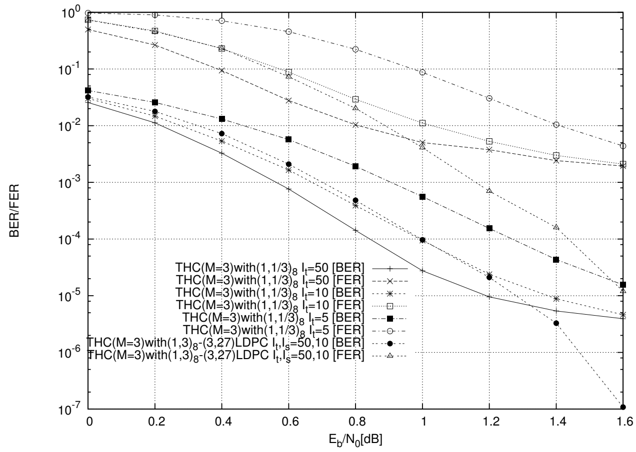

Figure 7: BER and FER performance of LDPC-THC concatenated code with overall rate $R = 0.15$ and generator polynomial $(1,3)_8$, and standard THC code with rate $R = 0.17$ and generator polynomial $(1,1/3)_8$. For both codes $N = 999$, $r = 3$, $M = 3$.

The BER and FER performance of these codes are compared in Fig. 7. In the waterfall region, the standard code has an better performance. However, in the SNR range of 1.3-1.6 dB, an error floor for the standard THC appears, and the BER of the LDPC-THC code continues to improve. This general trend was consistant as we varied the THC code, by using differing values of M, S, as well as using the asymmetric and punctured THC codes. We believe this favorable code performance is attributable to the good distance spectrum of LDPC codes coupled with the low rate of THC codes.

Significantly, with the LDPC code detecting valid codewords, superior performance can be obtained in a small average number of iterations. Fig. 8 shows the average number of iterations I_t required for convergence versus E_b/N_0. For example, at $E_b/N_0 = 1.4$, convergence was obtained in 1.9 iterations, considerably less than the 50 iterations required by the standard code.

3.4. Decoder Scheduling

In standard serial-mode decoding, each of M decoders operate sequentially, with the extrinsic output of decoder i passed to decoder $i+1$. In parallel-mode decoding, all decoders operate simultaneously, where each decoder's *a priori* information is obtained from the $M-1$ other decoders.

Fig. 9 shows the convergence behavior at $E_b/N_0 = -0.3$ dB for symmetric and asymmetric THC codes,

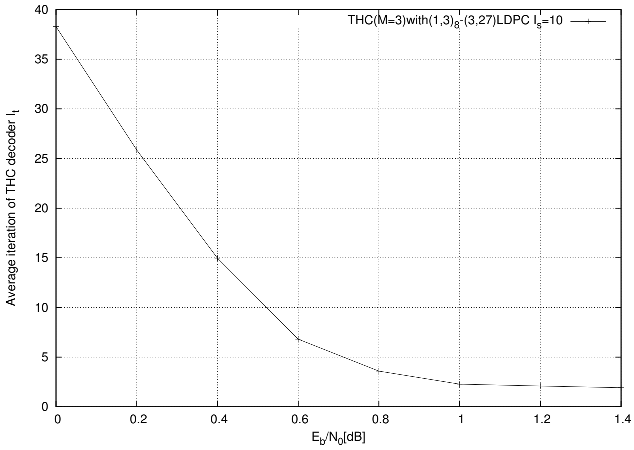

Figure 8: Average number of iterations I_t, for a THC code with $N = 999, r = 3, M = 3$ and a (3,27)-regular LDPC code with $I_s = 10$. Rate $R \approx= 0.17$.

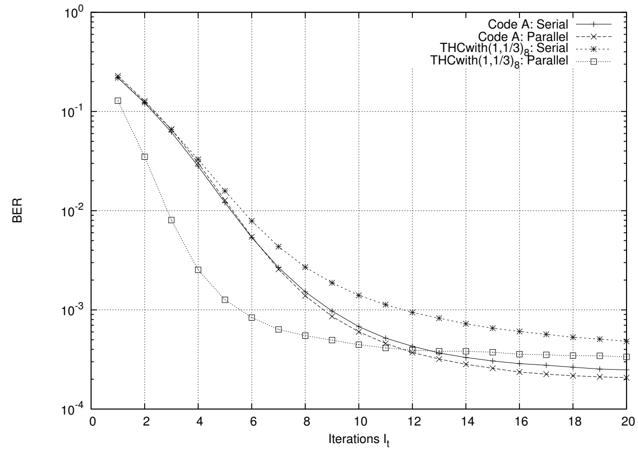

Figure 9: BER performance under serial and parallel turbo decoding as a function of the number of iterations, I_t ($E_b/N_0 = -0.3$ dB).

as a function of the number of iterations. Serial and parallel iterations have been normalized to have the same complexity. It can be seen that the parallel-mode decoding of the symmetric code converges much more quickly than the serial-mode decoding of the same code, conferring a significant complexity advantage on the parallel schedule. This is consistent with previous results for multiple convolutional turbo codes [6]. Also shown in Fig. 9 is the convergence for serial and parallel mode decoding of the asymmetric codes of the previous section.

4. CONCLUSIONS

Turbo-Hadamard codes have previously been shown to be effective low-rate codes, but have problems with high error floors and high decoding complexity. We have shown that the waterfall region of these codes can be improved by using asymmetric component codes. The error floor can be reduced by using punctured codes and by using an outer LDPC code. Decoder complexity can be reduced by using a parallel schedule, or by using the outer LDPC code as a stopping condition.

References

[1] L. Ping, W. K. Leung, and K. Y. Wu, “Low-rate turbo-Hadamard codes,” *IEEE Transactions Information Theory*, vol. 49, pp. 3213-3224, Dec. 2003.

[2] Y. J. Wu and L. Ping, “On the limiting performance of turbo-Hadamard codes,” *IEEE Communications Letters*, vol. 8, no. 7, pp. 449-451, Jul. 2004.

[3] O. Y. Takeshita, O. M. Collins, P. C. Massey, and D. J. Costello,Jr., “A note on asymmetric turbo-codes,” *IEEE Communications Letters*, vol. 3, no. 3, pp. 2082-2084, Mar. 1999.

[4] P. C. Massey, and D. J. Costello, Jr., “New low-complexity turbo-like codes,” in *Proceedings IEEE Information Theory Workshop(ITW)*, Cairns, Australia, Sept. 2001, pp. 70-72.

[5] O. Y. Takeshita, O. M. Collins, P. C. Massey, and D. J. Costello, Jr., “On the frame-error rate of concatenated turbo codes,” *IEEE Transactions Communications*, vol. 49, no. 4, pp. 602-608, Apr. 2001.

[6] J. Han and O. Y. Takeshita, “On the decoding structure for multiple turbo codes,” in *Proceedings IEEE Int. Symp. Information Theory(ISIT)*, Washington D.C., USA, Jun. 2001, p. 98.

[7] C. Y. Liu, H. Tang, S. Lin, and M. P. C. Fossorier, “An interactive concatenated turbo coding system,” *IEEE Transactions Vehicular Technology*, vol. 51, no. 5, pp. 998-1010, Sept. 2002.

[8] Y. Kou, S. Lin, and M. P. C. Fossorier, “Low-density parity-check codes based on finite geometries: A rediscovery and new results,” *IEEE Transactions Information Theory*, vol. 47, pp. 2711-2736, Nov. 2001.

[9] D. J. Costello, A. Banerjee, C. He, and P. C. Massey, "A comparison of low complexity turbo-like codes," in *36th Asilomar Conference on Signals, Systems, and Computers*, Nov. 2002.

[10] C. He, A. Banerjee, D. J. Costello, Jr., and P. C.Massey, "On the performance of low complexity multiple turbo codes," in *Proceedings 40th Annual Allerton Conference on Communications, Control, and Computing*, Oct. 2002.

[11] P. C. Massey and D. J. Costello, Jr., "New developments in asymmetric turbo codes," in *Proceedings 2nd Int. Symp. Turbo Codes and Related Topics*, Brest, France, Sept. 2000, pp. 93-99.

[12] D. J. Costello, Jr., A. Banerjee, T. E. Fuja, and P. C. Massey, "Some reflections on the design of bandwidth efficient turbo codes," in *Proceedings 4th Int. ITG Conf. Source and Channel Coding*, Berlin, Germany, Jan. 2002, pp. 357-363.