

Concatenated Zigzag Hadamard Codes

W. K. Raymond Leung, *Member, IEEE*, Guosen Yue, *Member, IEEE*,
Li Ping, *Member, IEEE*, and Xiaodong Wang, *Member, IEEE*

Abstract—In this correspondence, we introduce a new class of low-rate error correction codes called *zigzag Hadamard (ZH)* codes and their concatenation schemes. Each member of this class of codes is specified by a highly structured zigzag graph with each segment being a Hadamard codeword. The ZH codes enjoy extremely simple encoding and very low-complexity soft-input–soft-output (SISO) decoding based on a *posteriori* probability (APP) fast Hadamard transform (FHT) technique. We present an asymptotic performance analysis of the proposed concatenated ZH codes using the extrinsic mutual information transfer (EXIT) chart for infinite-length codes. We also provide a union bound analysis of the error performance for finite-length codes. Furthermore, the concatenated ZH codes are shown to be a good class of codes in the low-rate region. Specifically, a rate-0.0107 concatenated code with three ZH components and an interleaver size of 65536 can achieve the bit error rate (BER) performance of 10^{-5} at -1.15 dB, which is only 0.44 dB away from the ultimate Shannon limit. The proposed concatenated ZH codes offer similar performance as another class of low-rate codes—the turbo-Hadamard codes, and better performance than superorthogonal turbo codes, with much lower encoding and decoding complexities.

Index Terms—Extrinsic mutual information transfer (EXIT) chart, Hadamard codes, low-complexity decoding, low rate, parallel concatenation, union bound, zigzag codes.

I. INTRODUCTION

In the presence of additive white Gaussian noise (AWGN), the ultimate Shannon capacity in terms of E_b/N_0 is about $\log 2 = -1.592$ dB for codes with rates approaching zero [1], where E_b/N_0 is the signal-to-noise ratio (SNR) per information bit. Among existing channel coding schemes, Hadamard codes [2], [3] and superorthogonal convolutional codes [4] are traditional low-rate codes. However, these two classes of codes offer relatively low coding gain, and their performance is far away from the ultimate Shannon limit.

Recently, along with the discovery of turbo codes [5], low-rate concatenated codes have been studied in [6], [7] which offer relatively higher performance but they also incur higher complexity due to the complex trellis that specify the codes. The repeat-accumulate (RA) codes proposed in [8] have been proved to achieve the Shannon limit in AWGN channels when the code rate approaches zero. Another class of advanced codes, the low-density parity-check (LDPC) codes [9], [10], also exhibit capacity-approaching performance for various code rates when the ensemble profiles are optimized [11], [12]. However, in the low-rate region, both RA codes and LDPC codes suffer from performance loss and extremely slow convergence using iterative decoding. Other low-rate codes that have been devised to approach the Shannon limit include [13]–[15]. In particular, it is reported in [15]

Manuscript received October 10, 2004; revised December 7, 2005. The work of X. Wang was supported in part by the National Science Foundation under Grants DMS 0244583 and CCR-0207550, and in part by the U.S. Office of Naval Research under Grant N00014-03-1-0039.

W. K. R. Leung is with the Research Department of PSMT, Huawei Technologies Co., LTD, Shenzhen 518129, China (e-mail: wkleung@huawei.com).

G. Yue is with the NEC Laboratories America, Inc., Princeton, NJ 08540 USA (e-mail: yueg@nec-labs.com).

L. Ping is with the Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong (e-mail: eeliping@cityu.edu.hk).

X. Wang is with the Department of Electrical Engineering, Columbia University, New York, NY 10027 USA (e-mail: wangx@ee.columbia.edu).

Communicated by Ø. Ytrehus, Associate Editor for Coding Techniques.

Digital Object Identifier 10.1109/TIT.2006.871613

that the turbo-Hadamard codes constructed from Hadamard code arrays offer the bit error rate (BER) of 10^{-5} at $E_b/N_0 = -1.2$ dB, only around 0.39 dB away from the Shannon limit.

In this correspondence, we propose a new class of low-rate codes called *zigzag Hadamard (ZH)* codes and their concatenation schemes. This new class of codes are constructed by a highly structured zigzag graph [16] with each segment being a Hadamard codeword. The soft decoding algorithm for the turbo-Hadamard codes can be applied to decode the concatenated ZH codes, which employs an efficient *a posteriori* probability (APP) Hadamard decoding method based on the fast Hadamard transform (FHT). The performance of the concatenated ZH codes is close to that of the turbo-Hadamard codes, but the former enjoy much simpler encoding and decoding structures. We will derive an efficient decoding algorithm for the ZH codes using a simple two-way decoding technique. We will also conduct EXIT chart analysis to predict the performance of very long ZH codes and derive a bounding technique for the performance of short ZH codes. The proposed codes work well in the low-rate region. Simulation results demonstrate that the proposed concatenated ZH codes with long block length achieve 10^{-5} BER at $E_b/N_0 = -1.15$ dB, about 0.44 dB from the ultimate Shannon limit. The principles presented in this correspondence may also be generalized to design high performance concatenated codes using other block codes.

The remainder of this correspondence is organized as follows. In Section II, we introduce the ZH codes and the concatenation schemes. Both systematic and nonsystematic codes are considered. We also present the decoding algorithm. In Section III, we present the performance analysis of concatenated ZH codes based on the EXIT chart analysis for infinite-length codes, and based on the union bound analysis for finite-length codes. Numerical results and discussions are provided in Section IV. Section V contains the conclusions.

II. ZH CODES AND CONCATENATION SCHEMES

A. Hadamard Codes

A Hadamard codeword is obtained from a Hadamard matrix. Starting from $\mathbf{H}_1 = [+1]$, an $n \times n$ ($n = 2^r$) Hadamard matrix \mathbf{H}_n over $\{+1, -1\}$ can be constructed recursively as

$$\mathbf{H}_n = \begin{bmatrix} +\mathbf{H}_{\frac{n}{2}} & +\mathbf{H}_{\frac{n}{2}} \\ +\mathbf{H}_{\frac{n}{2}} & -\mathbf{H}_{\frac{n}{2}} \end{bmatrix}. \quad (1)$$

A length- 2^r Hadamard codeword set is formed by the columns of the biorthogonal Hadamard matrix $\pm \mathbf{H}_n$, denoted by $\{\pm \mathbf{h}^j : j = 0, 1, \dots, 2^r - 1\}$, in a binary $\{0, 1\}$ form, i.e., $+1 \rightarrow 0$ and $-1 \rightarrow 1$. We call r the order of the Hadamard code. Each Hadamard codeword carries $(r + 1)$ bits of information. In systematic encoding, the bit indexes $\{0, 1, 2, 4, \dots, 2^{r-1}\}$ are used as information bit positions of the length- 2^r Hadamard code. The other bits are parity bits. Denote \mathcal{H} as a Hadamard encoder. Then, the Hadamard encoding can be represented by

$$\mathbf{c} = \mathcal{H}(b_0, b_1, \dots, b_r) \quad (2)$$

where \mathbf{c} is the resulting codeword. Clearly, \mathbf{c} is a column of either $+\mathbf{H}_n$ or $-\mathbf{H}_n$. Hereafter, for simplicity we will drop the subscript n in \mathbf{H}_n .

B. Zigzag Hadamard Codes

Similar to building the zigzag code in [16], we can describe a zigzag Hadamard (ZH) code graphically as shown in Fig. 1. Each *segment* is a length- 2^r Hadamard code. The information bit sequence \mathbf{D} is first segmented into blocks $\{\mathbf{d}_k\}$, where $\mathbf{d}_k = [d_k(1), d_k(2) \dots d_k(r)]$,

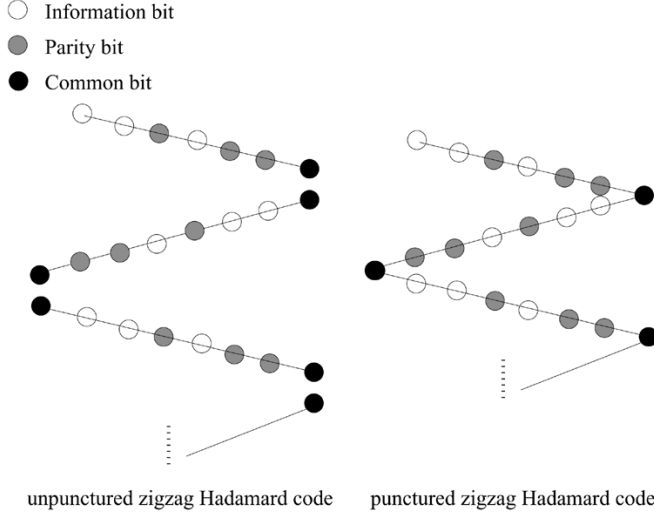


Fig. 1. Graphical representation of a systematic ZH code: $r = 3$.

$k = 1, 2, \dots, K$, represented by the white nodes. With the last parity bit of the previous segment being the first input bit to the Hadamard encoder for the current segment, the coded bits of the k th segment are obtained by

$$\mathbf{c}_k = \mathcal{H}(c_{k-1}(2^r - 1), d_k(1), d_k(2), \dots, d_k(r)) \quad (3)$$

where the codeword vector $\mathbf{c}_k = [c_k(0), c_k(1), \dots, c_k(2^r - 1)]$ with $c_k(0) = c_{k-1}(2^r - 1)$ and $c_k(2^{j-1}) = d_k(j)$, $j = 1, \dots, r$. Note that the first segment contains one extra information bit since its first bit can be freely assigned. In determining the rate (see below), we have to consider the first segment differently from other segments. For convenience, we will ignore the first bit by setting it to zero. It does not need to be transmitted and so it is not counted in the transmit energy. Since the first coded bit of each segment is the last bit of the previous segment, we call it the *common bit*, represented by a black node in Fig. 1. All other nodes are parity nodes, represented by the gray nodes. Denote $q_k = c_k(0)$ and let \mathbf{p}_k represent all the parity bits, i.e., $\mathbf{p}_k = \{c_k(i), i \neq 0, i \neq 2^{j-1}, j = 1, \dots, r\}$. The coded bit sequence is then denoted by a triplet, i.e., $\mathbf{c}_k = \{\mathbf{d}_k, q_k, \mathbf{p}_k\}$, $k = 1, \dots, K$. Clearly, the above ZH code is a *systematic code*.

We can also construct a *nonsystematic* ZH code as follows. (The purpose of considering nonsystematic ZH codes is to guarantee that the codes contain the recursive nature, which will be elaborated in detail in Section III. We will show the difference of the systematic and nonsystematic zigzag Hadamard codes regarding their recursive properties.) First, let

$$\tilde{d}_k(j) = c_{k-1}(2^r - 1) \oplus d_k(j), \quad j = 1, 2, \dots, r \quad (4)$$

where \oplus denotes the binary addition. Then, we perform the Hadamard encoding for $\{\tilde{d}_k(j)\}$, i.e.,

$$\mathbf{c}_k = \mathcal{H}(c_{k-1}(2^r - 1), \tilde{d}_k(1), \tilde{d}_k(2), \dots, \tilde{d}_k(r)) \quad (5)$$

where $c_k(2^{j-1}) = \tilde{d}_k(j)$, $j = 1, \dots, r$. Denote $\tilde{\mathbf{d}}_k = [\tilde{d}_k(1), \dots, \tilde{d}_k(r)]$. The codeword sequence of a nonsystematic zigzag Hadamard code is then given by $\mathbf{c}_k = \{\tilde{\mathbf{d}}_k, q_k, \mathbf{p}_k\}$,

$k = 1, \dots, K$. The code rate for both the systematic and the nonsystematic ZH codes is given by

$$R = \frac{r}{2^r}. \quad (6)$$

Finally, we can puncture the common bits $\{q_k\}$ to obtain the punctured ZH code with the code graph shown on the right side of Fig. 1. The code rate of this punctured ZH code is then

$$R = \frac{r}{2^r - 1}. \quad (7)$$

C. Concatenated Zigzag Hadamard Codes

It is easily seen from the code graph in Fig. 1 that the whole code structure does not contain any loop. Therefore, a ZH code can be viewed as an alternative *tree code* described in [17] with the strong Hadamard constraints replacing the single parity check constraints. A concatenated ZH code can be formed based on the coding structure of the concatenated tree (CT) codes [17].

Concatenation of systematic codes: A concatenated systematic code is constructed by concatenating M systematic ZH codes in a parallel manner, as shown in Fig. 2. Let $\{\mathbf{D}, \mathbf{q}, \mathbf{P}\}$ denote the information data block, the common bits, and the parity bits. Denote $\mathbf{D}^{(m)}$ as the m th interleaved information data block and $(\mathbf{q}^{(m)}, \mathbf{P}^{(m)})$ as the corresponding common bits and parity bits. The output from each systematic encoder defined in Section II-B contains the triplet sequence $\{\mathbf{D}^{(m)}, \mathbf{q}^{(m)}, \mathbf{P}^{(m)}\}$, $m = 1, 2, \dots, M$. To avoid repeating the information sequence, we only transmit the information bits once, as shown in the upper part of Fig. 2, and only collect $\{\mathbf{q}^{(m)}, \mathbf{P}^{(m)}\}$ from the m th component encoder. In this way, the final transmitted sequence of a systematic codeword becomes $\{\mathbf{D}, \mathbf{q}^{(1)}, \mathbf{P}^{(1)}, \dots, \mathbf{q}^{(M)}, \mathbf{P}^{(M)}\}$. The overall code rate is given by

$$R = \frac{r}{r + M(2^r - r)}. \quad (8)$$

Concatenation of nonsystematic codes: The encoder of a nonsystematic concatenated code is illustrated in the lower part of Fig. 2. The encoder output is $\{\tilde{\mathbf{D}}^{(m)}, \mathbf{q}^{(m)}, \mathbf{P}^{(m)}\}$, $m = 1, \dots, M$. All coded bits from the M ZH component codes are transmitted (which is necessary since there are no repeated systematic bits in this case). The final transmitted sequence of a nonsystematic codeword is $\{\tilde{\mathbf{D}}^{(1)}, \mathbf{q}^{(1)}, \mathbf{P}^{(1)}, \dots, \tilde{\mathbf{D}}^{(M)}, \mathbf{q}^{(M)}, \mathbf{P}^{(M)}\}$. The code rate of this nonsystematic concatenated code is then given by

$$R = \frac{r}{M2^r}. \quad (9)$$

Concatenation of punctured codes: We can also construct a concatenated code with the punctured ZH codes where the common bits $\mathbf{q}^{(m)}$, $\forall m$, (the first bit in each segment) are not transmitted. The rates for concatenated punctured systematic and nonsystematic codes are given, respectively, by

$$\text{systematic: } R = \frac{r}{r + M(2^r - r - 1)} \quad (10)$$

$$\text{nonsystematic: } R = \frac{r}{M(2^r - 1)}. \quad (11)$$

D. Decoding Algorithms

1) *Low-Complexity APP Hadamard Decoding:* Denote $\mathbf{x} = [x(0), x(1), \dots, x(2^r - 1)]$ as the received sequence of channel corrupted Hadamard codeword $\mathbf{c} = [c(0), c(1), \dots, c(2^r - 1)]$. The

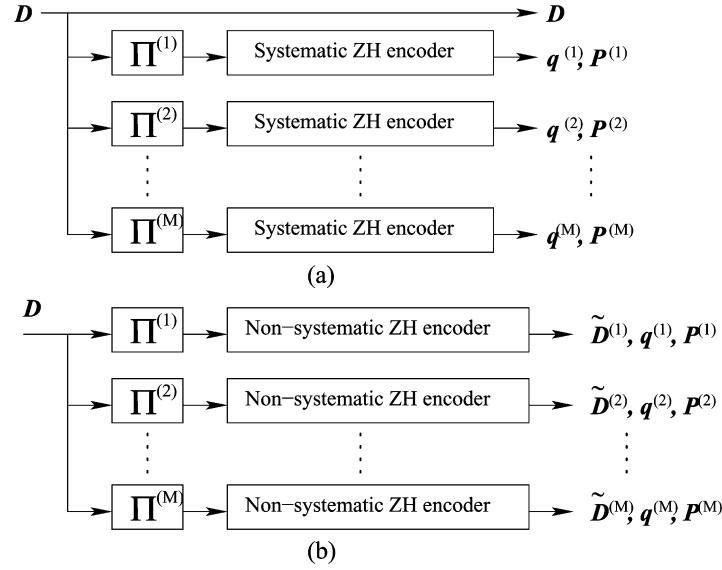


Fig. 2. Concatenated ZH code: (a) systematic code and (b) nonsystematic code.

log-likelihood-ratios (LLRs) of the coded bits from the APP decoding over $\{+1, -1\}$ are obtained by

$$\begin{aligned}
 L(i) &= \log \frac{P(c(i) = +1 | \mathbf{x})}{P(c(i) = -1 | \mathbf{x})} \\
 &= \log \frac{P(\mathbf{x} | c(i) = +1) P(c(i) = +1)}{P(\mathbf{x} | c(i) = -1) P(c(i) = -1)} \\
 &= \log \frac{\sum_{\mathbf{c} \in \{\pm \mathbf{h}^j\}; c(i) = +1} P(\mathbf{x} | \mathbf{c}) P(\mathbf{c})}{\sum_{\mathbf{c} \in \{\pm \mathbf{h}^j\}; c(i) = -1} P(\mathbf{x} | \mathbf{c}) P(\mathbf{c})}, \\
 & \quad i = d0, 1, 2, \dots, 2^r - 1. \quad (12)
 \end{aligned}$$

Denote $H(i, j)$ as the (i, j) th entry of the Hadamard matrix \mathbf{H} . Thus, for $c(i) = +1$, we have $c = \mathbf{h}^j$ with $H(i, j) = +1$ or $c = -\mathbf{h}^j$ with $H(i, j) = -1$; similarly for $c(i) = -1$. Assume the coded bits are transmitted through an AWGN channel with noise variance $\sigma^2 = \frac{N_0}{2}$. Given the *a priori* LLR $L_{\text{apr}}(i) = \log \frac{P(c(i) = +1)}{P(c(i) = -1)}$, we obtain (13) shown at the bottom of the page. Then, we have (14), also shown at the bottom of the page, where

$$\gamma(\pm \mathbf{h}^j) \triangleq \exp \left(\frac{1}{2} \left\langle \pm \mathbf{h}^j, \frac{2\mathbf{x}}{\sigma^2} + \mathbf{L}_{\text{apr}} \right\rangle \right) \quad (15)$$

and $\langle \cdot, \cdot \rangle$ denotes the inner product. Direct calculation of the LLR in (14) has a high computational complexity, i.e., $\mathcal{O}(2^{2r})$. The fast

$$L(i) = \log \frac{\sum_{\mathbf{h}^j: H(i, j) = +1} \exp \left(-\frac{\|\mathbf{h}^j - \mathbf{x}\|^2}{2\sigma^2} \right) P(\mathbf{c} = +\mathbf{h}^j) + \sum_{\mathbf{h}^j: H(i, j) = -1} \exp \left(-\frac{\|-\mathbf{h}^j - \mathbf{x}\|^2}{2\sigma^2} \right) P(\mathbf{c} = -\mathbf{h}^j)}{\sum_{\mathbf{h}^j: H(i, j) = -1} \exp \left(-\frac{\|\mathbf{h}^j - \mathbf{x}\|^2}{2\sigma^2} \right) P(\mathbf{c} = +\mathbf{h}^j) + \sum_{\mathbf{h}^j: H(i, j) = +1} \exp \left(-\frac{\|-\mathbf{h}^j - \mathbf{x}\|^2}{2\sigma^2} \right) P(\mathbf{c} = -\mathbf{h}^j)}. \quad (13)$$

$$\begin{aligned}
 L(i) &= \log \frac{\sum_{\mathbf{h}^j: H(i, j) = \pm 1} \exp \left(-\frac{\|\pm \mathbf{h}^j - \mathbf{x}\|^2}{2\sigma^2} \right) P(\mathbf{c} = \pm \mathbf{h}^j)}{\sum_{\mathbf{h}^j: H(i, j) = \mp 1} \exp \left(-\frac{\|\pm \mathbf{h}^j - \mathbf{x}\|^2}{2\sigma^2} \right) P(\mathbf{c} = \pm \mathbf{h}^j)} \\
 &= \log \frac{\sum_{\mathbf{h}^j: H(i, j) = \pm 1} \exp \left(\frac{1}{2} \left\langle \pm \mathbf{h}^j, \frac{2\mathbf{x}}{\sigma^2} \pm \sum_{i'} L_{\text{apr}}(i') H(i', j) / 2 \right\rangle \right)}{\sum_{\mathbf{h}^j: H(i, j) = \mp 1} \exp \left(\frac{1}{2} \left\langle \pm \mathbf{h}^j, \frac{2\mathbf{x}}{\sigma^2} \pm \sum_{i'} L_{\text{apr}}(i') H(i', j) / 2 \right\rangle \right)} \\
 &= \log \frac{\sum_{\mathbf{h}^j: H(i, j) = \pm 1} \exp \left(\frac{1}{2} \left\langle \pm \mathbf{h}^j, \frac{2\mathbf{x}}{\sigma^2} \right\rangle + \frac{1}{2} \left\langle \pm \mathbf{h}^j, \mathbf{L}_{\text{apr}} \right\rangle \right)}{\sum_{\mathbf{h}^j: H(i, j) = \mp 1} \exp \left(\frac{1}{2} \left\langle \pm \mathbf{h}^j, \frac{2\mathbf{x}}{\sigma^2} \right\rangle + \frac{1}{2} \left\langle \pm \mathbf{h}^j, \mathbf{L}_{\text{apr}} \right\rangle \right)} \\
 &= \log \frac{\sum_{\mathbf{h}^j: H(i, j) = \pm 1} \gamma(\pm \mathbf{h}^j)}{\sum_{\mathbf{h}^j: H(i, j) = \mp 1} \gamma(\pm \mathbf{h}^j)} \quad (14)
 \end{aligned}$$

Hadamard transform (FHT) can be employed to reduce the complexity of the inner product in (14) based on the butterfly graph of the Hadamard matrix. To compute $y(j) = \langle \mathbf{h}^j, \mathbf{x} \rangle$ for $j = 0, 1, \dots, 2^r - 1$, we have

$$\mathbf{y} = \mathbf{H}_n \mathbf{x} = \begin{bmatrix} +\mathbf{I}_{\frac{n}{2}} & +\mathbf{I}_{\frac{n}{2}} \\ +\mathbf{I}_{\frac{n}{2}} & -\mathbf{I}_{\frac{n}{2}} \end{bmatrix} \begin{bmatrix} \mathbf{H}_{\frac{n}{2}} \mathbf{x}_1 \\ \mathbf{H}_{\frac{n}{2}} \mathbf{x}_2 \end{bmatrix} \quad (16)$$

where $\mathbf{y} = [y(0), \dots, y(2^r - 1)]^T$, $\mathbf{I}_{\frac{n}{2}}$ is an identity matrix, and \mathbf{x}_1 , \mathbf{x}_2 are the first and the second halves of \mathbf{x} . It is seen that we can recursively factorize $\mathbf{H}_{\frac{n}{2}} \mathbf{x}_1$ and $\mathbf{H}_{\frac{n}{2}} \mathbf{x}_2$ in the same way and stop at $\mathbf{H}_1 x(j)$, $j = 0, 1, \dots, 2^r - 1$. The process is similar to the fast Fourier transform (FFT) computation, resulting in a butterfly-type flow graph. With such a FHT method, the complexity for calculating (14) is reduced to $\mathcal{O}(r2^r)$.

The summations in the numerator and denominator in (14) for computing $L(i)$, $i = 0, 1, \dots, 2^r - 1$, can be efficiently computed by the APP-FHT algorithm [15]. First consider \mathbf{H}_2 . We have

$$\begin{bmatrix} \sum_{H(0,j)=\pm 1} \gamma(\pm \mathbf{h}^j) \\ \sum_{H(0,j)=\mp 1} \gamma(\pm \mathbf{h}^j) \\ \sum_{H(1,j)=\pm 1} \gamma(\pm \mathbf{h}^j) \\ \sum_{H(1,j)=\mp 1} \gamma(\pm \mathbf{h}^j) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} \gamma(+\mathbf{h}^0) \\ \gamma(-\mathbf{h}^0) \\ \gamma(+\mathbf{h}^1) \\ \gamma(-\mathbf{h}^1) \end{bmatrix}. \quad (17)$$

Denote

$$\begin{aligned} \underline{a}(j) &= [\gamma(+\mathbf{h}^j), \gamma(-\mathbf{h}^j)] \\ \underline{b}(i) &= \begin{bmatrix} \sum_{H(i,j)=\pm 1} \gamma(\pm \mathbf{h}^j), & \sum_{H(i,j)=\mp 1} \gamma(\pm \mathbf{h}^j) \end{bmatrix} \\ \mathbf{a}_n &= [\underline{a}(0), \dots, \underline{a}(n-1)]^T \\ \mathbf{b}_n &= [\underline{b}(0), \dots, \underline{b}(n-1)]^T \end{aligned}$$

where $n = 2^r$. Define

$$\mathbf{J} \triangleq \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \bar{\mathbf{J}} \triangleq \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \text{and } \mathbf{Q}_2 \triangleq \begin{bmatrix} \mathbf{J} & \bar{\mathbf{J}} \\ \bar{\mathbf{J}} & \mathbf{J} \end{bmatrix}. \quad (18)$$

Obviously, we have $\mathbf{b}_1 = \mathbf{J} \mathbf{a}_1$. Then (17) can be written as

$$\mathbf{b}_2 = \mathbf{Q}_2 \mathbf{a}_2 \quad (19)$$

where \mathbf{a}_2 and \mathbf{b}_2 are 2×1 compound vectors with every entry being a pair of real numbers, $\underline{a}(j)$ and $\underline{b}(j)$, $j = 0, 1$, respectively. Similarly to the Hadamard matrix \mathbf{H}_n , we can recursively construct \mathbf{Q}_n as

$$\mathbf{Q}_n = \begin{bmatrix} \mathbf{Q}_{\frac{n}{2}} & \mathbf{Q}_{\frac{n}{2}} \\ \mathbf{Q}_{\frac{n}{2}} & \bar{\mathbf{Q}}_{\frac{n}{2}} \end{bmatrix} \quad (20)$$

where $\bar{\mathbf{Q}}$ denotes the complementary matrix of \mathbf{Q} . Define $\hat{\mathbf{I}} \triangleq \text{diag}\{\mathbf{J}, \dots, \bar{\mathbf{J}}\}$. We then have $\bar{\mathbf{Q}} = \hat{\mathbf{I}} \mathbf{Q}$. Hence, for \mathbf{H}_n , we can obtain following factorization:

$$\mathbf{b}_n = \mathbf{Q}_n \mathbf{a}_n = \begin{bmatrix} \mathbf{I} & \hat{\mathbf{I}} \\ \hat{\mathbf{I}} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{\frac{n}{2}} \mathbf{a}_{n,1} \\ \mathbf{Q}_{\frac{n}{2}} \mathbf{a}_{n,2} \end{bmatrix} \quad (21)$$

where $\mathbf{a}_{n,1}$ and $\mathbf{a}_{n,2}$ are the first and the second halves of \mathbf{a}_n ; \mathbf{a}_n , \mathbf{b}_n are $n \times 1$ compound vectors with every entry being a pair of real numbers, $\underline{a}(j)$ and $\underline{b}(j)$, $j = 0, \dots, n-1$, respectively. Similar to the Hadamard matrix \mathbf{H}_n , we can continue the above factorization until $\mathbf{Q}_1 \equiv \mathbf{J}$ appears. Thus, similarly to FHT, we can compute \mathbf{b}_n efficiently by this so-called n -point APP-FHT.

2) *Low-Complexity APP Decoding of a ZH Code*: The encoding of a ZH code defined in Section II-B is a Markov process. Let $[\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_K]$ be the K segments in a ZH codeword. Any two halves of this codeword, $[\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_{k-1}]$ and $[\mathbf{c}_k, \mathbf{c}_{k+1}, \dots, \mathbf{c}_K]$,

are related to each other only through two common bits (i.e., the last bit in \mathbf{c}_{k-1} and the first bit in \mathbf{c}_k). Consequently, the APP decoding of a ZH code can be accomplished using the following two-way algorithm. We will not provide a detailed proof of this algorithm since it can be easily verified using the well known backward-forward estimation method of a Markov process [18].

The Two-Way Decoding Algorithm of a ZH Code With K Segments

[1] Preparation

- Let $x_k(i)$ be the noisy observation of $c_k(i)$ (the i th bit c_k). Generate the *a priori* LLRs $\{\tilde{L}_k(i)\}$ for $\{c_k(i)\}$ defined by

$$\tilde{L}_k(i) = \frac{2x_k(i)}{\sigma^2}, \quad i = 0, 1, \dots, 2^r - 1, \quad k = 1, \dots, K.$$

[2] Forward Recursion

- For $k = 1, \dots, K$, apply APP decoding (14) to the last bit in the k th segment.¹ Then update the *a priori* LLR of the first bit in the $(k+1)$ th segment by adding the *a posteriori* LLR of the last bit in the k th segment.

[3] Backward Recursion

- Keep the updating for the first bit in every segment as in the Forward Recursion.
- For $k = K, \dots, 1$, apply APP decoding employing FHT and APP-FHT to the k th segment to obtain the output LLRs. Then update the *a priori* LLR of the last bit in the $(k-1)$ th segment by adding the extrinsic LLR of the first bit in the k th segment.²

The following is an estimation of the complexity per segment per iteration involved in the above procedure. Let $r = \log_2 n$. The Forward Recursion requires $n \times r$ additions for an FHT and $2n$ additions for the summations in (14). The Backward Recursion involves $n \times r$ additions for an FHT and $6n$ additions for a reduced-output APP-FHT [15]. Cost can be reduced by modifying the FHT outputs of the Forward Recursion to obtain those for the Backward Recursion, since the inputs to the two FHT operations only differ in the first bit of each segment and the modification only costs n additions. The overall complexity is then about $r \times n + 9n$ additions per segment per iteration. (We have ignored operations such as updating the first and last bits.) Since each segment carries r bits of information, the about complexity is equivalent to $n + 9n/r$ additions per information bit per iteration, which is quite moderate. The reduced-input FHT technique discussed in [15] can also be used for further cost reduction.

3) *Decoding of Concatenated ZH Codes*: The iterative decoder for a concatenated zigzag Hadamard code with M component codes is illustrated in Fig. 3. The decoding principle is similar to that of the concatenated tree (CT) decoder discussed in [17]. The decoder consists of M iterative local decoders. The LLRs of the information bits are exchanged between two adjacent local decoders during decoding.

Denote $\tilde{\mathbf{L}}_{ch}$ as the initial *a priori* LLR vector for information bits from the channel. For nonsystematic codes, $\tilde{\mathbf{L}}_{ch} = \mathbf{0}$. Denote $\tilde{\mathbf{L}}^{(m,q)}$, $\mathbf{L}_{\text{ext}}^{(m,q)}$, and $\mathbf{L}_{\text{ext}}^{(m,q)}$ as respectively the *a priori*, *a posteriori*, and extrinsic LLR vectors for information bits in the m th component code and the q th decoding iteration. The extrinsic information produced by the m th component decoder during the q th iteration is given by [5]

$$\mathbf{L}_{\text{ext}}^{(m,q)} = \mathbf{L}^{(m,q)} - \left(\tilde{\mathbf{L}}^{(m,q)} - \mathbf{L}_{\text{ext}}^{(m,q-1)} \right). \quad (22)$$

¹In the Forward Recursion, we only need to compute the LLR for the last bit in each segment, which can be accomplished by direct summations in (14). The use of APP-FHT is used in the Backward Recursion where the LLRs for all information bits are required.

²The extrinsic LLR here is the difference between the *a posteriori* LLRs produced in the Backward and the Forward Recursions.

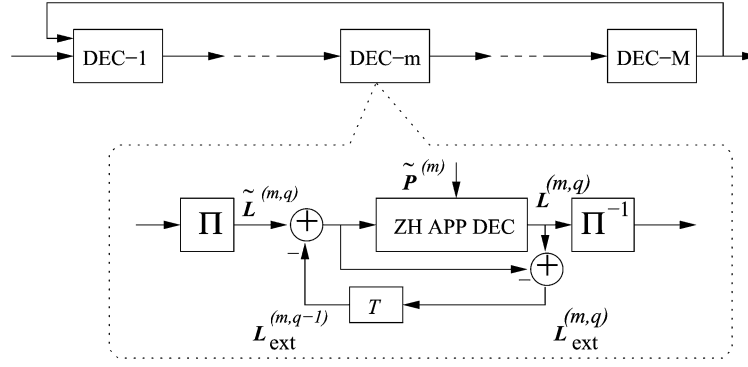
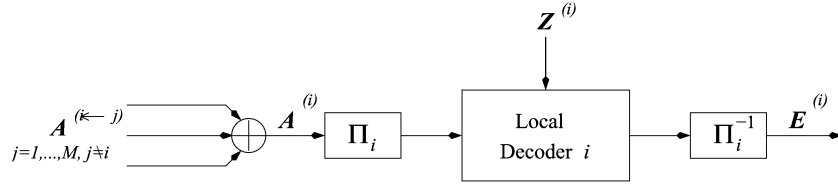


Fig. 3. Iterative decoder of a concatenated ZH code.


 Fig. 4. Decoder of one component code in a parallel concatenated ZH code with M constituent codes.

It is easy to verify by deduction that the output *a posteriori* LLR vector of the m th component decoder can be expressed as

$$\begin{aligned} \mathbf{L}^{(m,q)} = & \tilde{\mathbf{L}}_{ch} + \mathbf{L}_{ext}^{(m,q-1)} + \mathbf{L}_{ext}^{(m+1,q-1)} + \dots \\ & + \mathbf{L}_{ext}^{(M,q-1)} + \mathbf{L}_{ext}^{(1,q)} + \mathbf{L}_{ext}^{(2,q)} + \dots + \mathbf{L}_{ext}^{(m-1,q)} \end{aligned} \quad (23)$$

for instance, the channel information plus the most recent extrinsic LLRs produced by the M component decoders. The input *a priori* LLR vector fed into the m th component decoder is given by $\mathbf{L}^{(m-1,q)} - \mathbf{L}_{ext}^{(m,q-1)}$ as shown in Fig. 3. The subtraction here prevents the output of the m th component decoder from circulating back to itself again.

III. PERFORMANCE ANALYSIS

In the following, we will discuss an EXIT chart analysis technique for very long codes and a union bound technique for relatively short codes.

A. EXIT Chart Analysis

The extrinsic information transfer (EXIT) chart [19], [20] is an efficient tool for assessing the asymptotic performance (infinite code length) and the convergence behavior of trellis codes. We next employ this technique to analyze the performance of the concatenated ZH codes.

1) *Threshold Analysis*: A concatenated code with M component codes usually requires an M -dimensional EXIT plot. In order to characterize it in a two-dimensional EXIT chart, we first outline an equivalent decoder of one component code illustrated in Fig. 4. Denote $\mathbf{Z}^{(m)}$ as the channel observations corresponding to the m th component code. Denote $\mathbf{A}^{(m)}$ as the input *a priori* LLRs to the m th component code for the information bits \mathbf{D} . Initially, we set $\mathbf{A}^{(m)}$ to zero. During each iteration, the extrinsic LLRs of \mathbf{D} , denoted by $\mathbf{E}^{(m)}$, are obtained from the APP decoding based on $\mathbf{Z}^{(m)}$ and $\mathbf{A}^{(m)}$. $\mathbf{A}^{(m)}$ is updated as $\mathbf{A}^{(m)} = \sum_{m' \neq m} \mathbf{A}^{(m-m')}$, where $\mathbf{A}^{(m-m')}$ is obtained from $\mathbf{E}^{(m')}$ after interleaving.³

³Here we assume that all component decoders operate in parallel. This is slightly different from the serial scheduling in Fig. 3. The serial scheduling has faster convergence speed but is more difficult to analyze.

Denote $I(\mathbf{D}; \mathbf{A}^{(m-m')})$ as the mutual information between the information sequence \mathbf{D} and $\mathbf{A}^{(m-m')}$ and $I(\mathbf{D}; \mathbf{E}^{(m)})$ the mutual information between \mathbf{D} and $\mathbf{E}^{(m)}$. Because all component codes are identical and all $\mathbf{Z}^{(m)}$, $m = 1, 2, \dots, M$, are statistically equivalent, for all m, m' , $I(\mathbf{D}; \mathbf{A}^{(m-m')})$ are equal during a given iteration, so are $I(\mathbf{D}; \mathbf{E}^{(m)})$. Hence, we can define

$$I(\mathbf{D}; \mathbf{A}^{(m-m')}) = I_A \quad \text{and} \quad I(\mathbf{D}; \mathbf{E}^{(m)}) = I_E. \quad (24)$$

We now consider an example of a concatenated systematic ZH code with common bits punctured. In this example, $M = 3$ and $r = 8$. The code rate is 0.0107. The EXIT chart of I_E versus I_A on a two-dimensional plane is plotted in Fig. 5. We can clearly see the evolution of I_E . When the extrinsic mutual information evolves to one, it indicates the density converges to infinity and the error probability approaches zero. The EXIT curves vary with E_b/N_0 . When E_b/N_0 is below a certain value, the EXIT curves intersect with the line $y = x$ before reaching the point (1,1). That means the iterative decoding converges to the intersection point and the extrinsic information will not grow beyond this point. In this case, the decoding error occurs. The E_b/N_0 value where the EXIT curve is just tangent to the diagonal line $y = x$ is called *threshold*. In this example, we find that the E_b/N_0 threshold is -1.25 dB, only 0.34 dB away from the ultimate Shannon limit of -1.59 dB for low-rate codes. The EXIT analysis can be validated through the simulations of long codes. The simulation results will be given in Section IV.

We now consider punctured concatenated ZH codes. We obtain the E_b/N_0 thresholds using EXIT chart technique for both systematic and nonsystematic codes with different Hadamard order r . The threshold results are listed in Table I. It is seen that with the Hadamard order r increasing and code rate decreasing, the E_b/N_0 threshold is improved to approach Shannon limit until $r = 8$ for systematic codes and $r = 7$ for nonsystematic codes. The corresponding best low thresholds are -1.25 and -1.20 dB, respectively.

Interestingly, some systematic concatenated codes do not have the threshold. In these cases, the EXIT curves cannot approach to one. Therefore, they must have an intersection with the line $y = x$. An example for $r = 5$ and $M = 3$ is shown in Fig. 6. We find that this happens for all systematic codes with the odd Hadamard orders, i.e.,

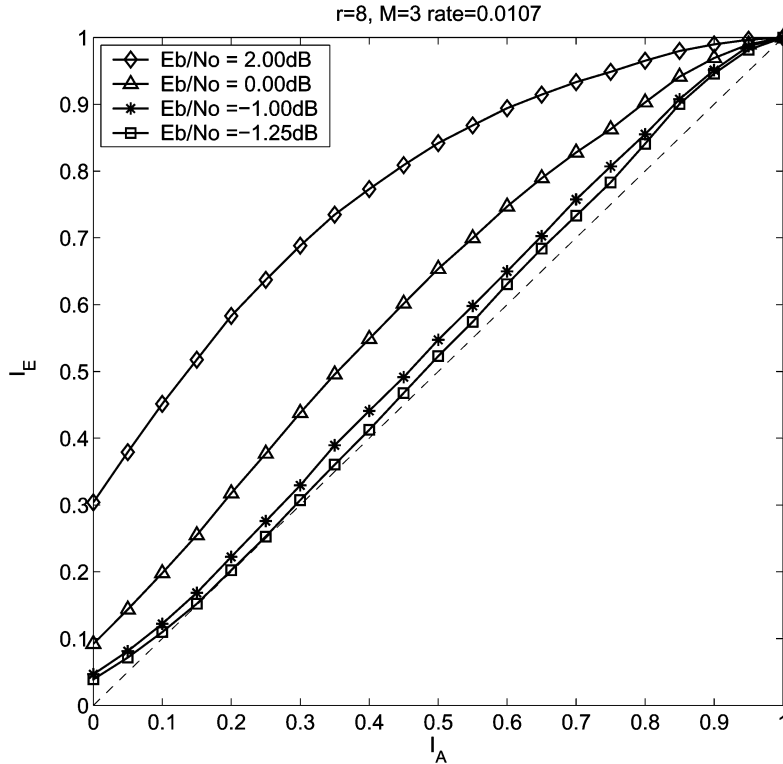


Fig. 5. EXIT chart for a punctured systematic concatenated ZH code with $r = 8$ and $M = 3$.

TABLE I
 E_b/N_0 THRESHOLD PERFORMANCE OF CONCATENATED ZH CODES AND TURBO-HADAMARD CODES WITH $M = 3$ FOR AWGN CHANNELS

	Systematic ZH Codes						
order and rate R	$r = 2$	$r = 3$	$r = 4$	$r = 5$	$r = 6$	$r = 7$	$r = 8$
Capacity $\frac{E_b}{N_0}$ (dB)	-0.24	-0.95	-1.25	-1.41	-1.48	-1.52	-1.54
$\frac{E_b}{N_0}$ EXIT (dB)	0.90	N/A	-0.60	N/A	-1.10	N/A	-1.25
$\frac{E_b}{N_0}$ Simu. (dB)	0.97	N/A	-0.44	N/A	-0.98	N/A	-1.15
	Non-systematic ZH Codes						
order and rate R	$r = 2$	$r = 3$	$r = 4$	$r = 5$	$r = 6$	$r = 7$	$r = 8$
Capacity $\frac{E_b}{N_0}$ (dB)	-0.89	-1.14	-1.31	-1.43	-1.49	-1.53	-1.54
$\frac{E_b}{N_0}$ EXIT (dB)	0.20	-0.20	-0.45	-0.75	-1.00	-1.20	-1.20
$\frac{E_b}{N_0}$ Simu. (dB)	0.26	-0.10	-0.41	-0.69	-0.95	-1.12	-1.11
	Turbo-Hadamard Codes						
order and rate R	$r = 2$	$r = 3$	$r = 4$	$r = 5$	$r = 6$	$r = 7$	$r = 8$
Capacity $\frac{E_b}{N_0}$ (dB)	-0.25	-0.167	-0.100	-0.058	-0.033	-0.019	-0.0106
$\frac{E_b}{N_0}$ Simu. (dB)	0.95	0.07	-0.38	-0.70	-0.95	-1.18	-1.13

$r = 3, 5, 7, \dots$. In the follows, we will provide an explanation for this phenomenon.

2) *Recursiveness of Systematic and Nonsystematic ZH Codes:* We first introduce some definitions. By “weight” we always mean Hamming weight. We will say that a ZH code is convolutional-like. A convolutional-like code is a code that can be extended indefinitely with a regular pattern. The rate of a convolutional-like code is independent of its length, which is true for the ZH code defined in Fig. 1 (with the first bit in the first segment ignored).

Definition 3.1: A binary convolutional-like code is said to be weight-recursive if any information sequence with Hamming distance

one will produce a codeword with Hamming weight $\rightarrow \infty$ when its length $\rightarrow \infty$.

As an example, the recursive convolutional code used in [5] is weight recursive. The weight-recursive property is the necessary condition for a code to achieve error free performance in an AWGN channel (assuming finite transmission energy) since otherwise, there are always codewords with finite weight even when the length goes to infinity. It is easy to verify that if the minimum Hamming weight of a linear code is finite then it cannot achieve error-free performance. Note that here error-free performance does not imply channel capacity, but it implies the existence of a threshold.

Note that the weight-recursive property is not a sufficient condition for error-free performance, since there may be codewords with finite weight caused by information sequences with weight larger than one. However, if several weight-recursive codes are concatenated using random interleavers, then the resultant code may achieve error-free performance. On the other hand, the concatenation of several nonweight-recursive codes will still produce a nonweight-recursive code and so it cannot achieve error-free performance. We will not discuss in detail this issue. The interested readers are referred to [21].

Next, we will show that the nonsystematic component codes and systematic component codes with even order are weight recursive. First, we examine the systematic component codes with even order.

Theorem 3.1: A systematic ZH code with even Hadamard order r is weight-recursive.

Proof: We first prove the following relationship for a systematic ZH code with even r :

$$c_{k+1}(0) = c_k(0) \oplus d_k(1) \oplus d_k(2) \oplus \cdots \oplus d_k(r). \quad (25)$$

Based on (25), it is easy to verify that the weight of an infinitely long code will approach infinite when there is only one nonzero input bit. To prove (25), return to the basic definitions in (1) and (3). We know $c_{k+1}(0) = c_k(2^r - 1)$ and also for systematic code, we have $c_k(2^{j-1}) = d_k(j)$ for $j = 0, 1, \dots, r - 1$. Then, to prove (25), it suffices to show

$$c_k(2^r - 1) = c_k(0) \oplus c_k(1) \oplus c_k(2) \oplus c_k(4) \oplus \cdots \oplus c_k(2^{r-1}). \quad (26)$$

This can be proved by induction. First, we examine $r = 2$ case. Denote \mathbf{M} as the Hadamard matrix in binary $\{0, 1\}$ form. The Hadamard codeword matrix is given by

$$\mathbf{M}_4 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \quad \text{and} \quad \overline{\mathbf{M}}_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \quad (27)$$

where $\overline{\mathbf{M}}$ is denoted as binary complementary matrix of \mathbf{M} . It is easily verified that (26) is satisfied by \mathbf{M}_4 and $\overline{\mathbf{M}}_4$. Now assume (26) is also true for the Hadamard matrix with order $r = m$, i.e., \mathbf{M}_n , where $n = 2^m$, given by

$$c_k(2^m - 1) = c_k(0) \oplus c_k(1) \oplus \cdots \oplus c_k(2^{m-1}). \quad (28)$$

Then, the Hadamard matrix with order $r = m + 2$ is given by

$$\mathbf{M}_{4n} = \begin{bmatrix} \mathbf{M}_n & \mathbf{M}_n & \mathbf{M}_n & \mathbf{M}_n \\ \mathbf{M}_n & \overline{\mathbf{M}}_n & \mathbf{M}_n & \overline{\mathbf{M}}_n \\ \mathbf{M}_n & \mathbf{M}_n & \overline{\mathbf{M}}_n & \mathbf{M}_n \\ \mathbf{M}_n & \overline{\mathbf{M}}_n & \overline{\mathbf{M}}_n & \mathbf{M}_n \end{bmatrix}. \quad (29)$$

From the above matrix, we can find the following relationship for all the codewords with length 2^{m+2} , given by

$$c_k(2^{m+2} - 1) \oplus c_k(2^m - 1) = c_k(2^m) \oplus c_k(2^{m+1}). \quad (30)$$

Therefore, we have

$$\begin{aligned} c_k(2^{m+2} - 1) &= c_k(2^m - 1) \oplus c_k(2^m) \oplus c_k(2^{m+1}) \\ &= c_k(0) \oplus c_k(1) \oplus \cdots \oplus c_k(2^{m-1}) \\ &\quad \oplus c_k(2^m) \oplus c_k(2^{m+1}). \end{aligned} \quad (31)$$

Similarly we can prove this for $\overline{\mathbf{M}}_{4n}$. \square

Theorem 3.2: A systematic ZH code with odd Hadamard order r is not weight-recursive.

Proof: We need to show that the recursive constraint in (25) does not exist for systematic codes with odd orders. Consider the odd order case with $r = 3$. The codeword set is given by

$$\mathbf{M}_8 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

and

$$\overline{\mathbf{M}}_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}. \quad (32)$$

It is seen that (26) holds for the codewords from \mathbf{M}_8 , but does not hold for the complementary set $\overline{\mathbf{M}}_8$. With the same induction procedure, we can show that (26) is not satisfied for all the systematic ZH codes with odd order. Assuming that $r = m$ and m is odd, for $\overline{\mathbf{M}}_n$, $n = 2^m$, we have

$$c_k(2^m - 1) \neq c_k(0) \oplus c_k(1) \oplus \cdots \oplus c_k(2^{m-1}). \quad (33)$$

When $r = m + 2$, for all the codewords in $\overline{\mathbf{M}}_{4n}$, we have the same relationship in (30). Therefore, we have

$$\begin{aligned} c_k(2^{m+2} - 1) &= c_k(2^m - 1) \oplus c_k(2^m) \oplus c_k(2^{m+1}) \\ &\neq c_k(0) \oplus c_k(1) \oplus \cdots \oplus c_k(2^{m-1}) \\ &\quad \oplus c_k(2^m) \oplus c_k(2^{m+1}). \end{aligned} \quad (34)$$

\square

Theorem 3.3: A nonsystematic ZH code with any order $r > 1$ is weight-recursive.

Proof: Starting from (25), for a nonsystematic ZH code, we have

$$\begin{aligned} c_k(2^r - 1) &= c_k(0) \oplus \underbrace{c_k(0) \oplus \tilde{d}_k(1)}_{d_k(1)} \oplus \underbrace{c_k(0) \oplus \tilde{d}_k(2)}_{d_k(2)} \\ &\quad \oplus \cdots \oplus \underbrace{c_k(0) \oplus \tilde{d}_k(r)}_{d_k(r)} \\ &= \underbrace{c_k(0) \oplus c_k(0) \oplus \cdots \oplus c_k(0)}_{r+1} \oplus c_k(1) \oplus c_k(2) \\ &\quad \oplus \cdots \oplus c_k(2^{r-1}). \end{aligned} \quad (35)$$

We can also prove (35) by induction. When $r = 2$, it is proved above. Assume (35) is true when $r = m$, i.e.,

$$\begin{aligned} c_k(2^m - 1) &= \underbrace{c_k(0) \oplus c_k(0) \oplus \cdots \oplus c_k(0)}_{m+1} \\ &\quad \oplus c_k(1) \oplus c_k(2) \oplus \cdots \oplus c_k(2^{m-1}). \end{aligned} \quad (36)$$

Now consider $r = m + 1$. We have

$$\mathbf{M}_{2n} = \begin{bmatrix} \mathbf{M}_n & \mathbf{M}_n \\ \mathbf{M}_n & \overline{\mathbf{M}}_n \end{bmatrix}, \quad \overline{\mathbf{M}}_{2n} = \begin{bmatrix} \overline{\mathbf{M}}_n & \overline{\mathbf{M}}_n \\ \overline{\mathbf{M}}_n & \mathbf{M}_n \end{bmatrix}. \quad (37)$$

We find that the following equation holds for every column in \mathbf{M}_{2n} , $\overline{\mathbf{M}}_{2n}$ in (37):

$$c_k(2^{m+1} - 1) \oplus c_k(2^m - 1) = c_k(2^m) \oplus c_k(0). \quad (38)$$

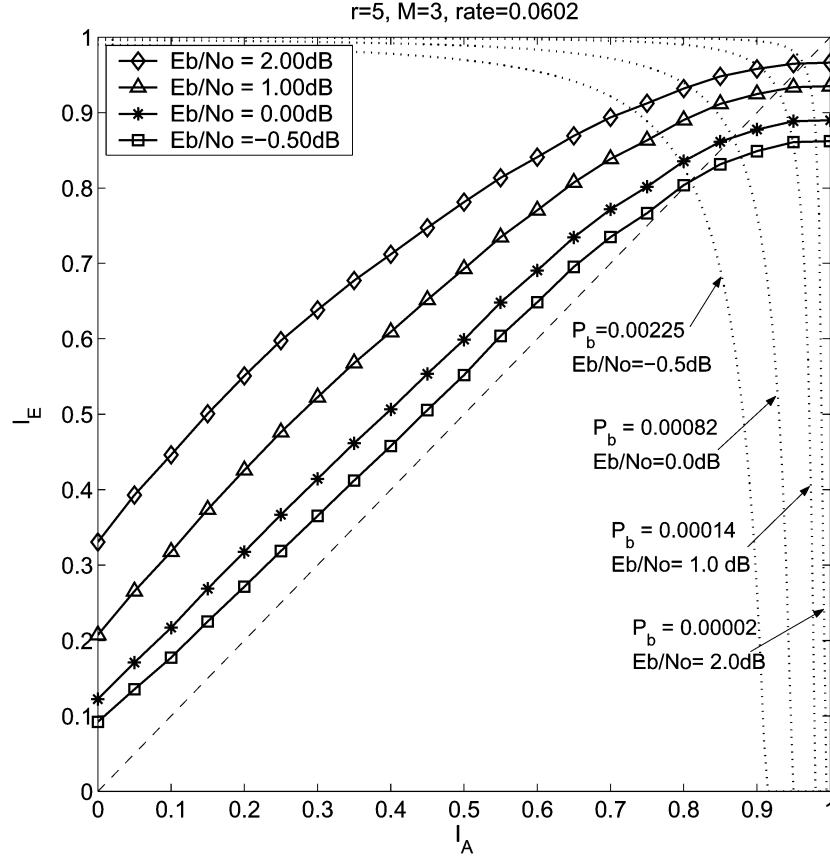


Fig. 6. EXIT chart for a punctured systematic concatenated ZH code with $r = 5$ and $M = 3$.

We then have

$$\begin{aligned}
 c_k(2^{m+1} - 1) &= c_k(2^m - 1) \oplus c_k(2^m) \oplus c_k(0) \\
 &= \underbrace{c_k(0) \oplus c_k(0) \oplus \cdots \oplus c_k(0)}_{m+2} \oplus c_k(1) \\
 &\quad \oplus c_k(2) \oplus \cdots \oplus c_k(2^{m-1}) \oplus c_k(2^m).
 \end{aligned} \tag{39}$$

We can now see the advantages of the nonsystematic ZH code over the systematic one, as the former is more flexible regarding order (and the related rate) selection.

3) *Error Probability Estimation:* Although there is no threshold for the systematic codes with odd orders, the converged bit error probability P_b can be estimated from the intersection points in the EXIT charts [20], given by

$$P_b \approx Q\left(\frac{1}{2}\sqrt{8R\frac{E_b}{N_0} + J^{-1}(I_A)^2 + J^{-1}(I_E)^2}\right) \tag{40}$$

where $Q(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-t^2/2} dt$; and the function $J(\sigma)$ is defined by

$$J(\sigma) \triangleq 1 - \int_{-\infty}^{\infty} \frac{e^{-\frac{y-\sigma^2/2}{2\sigma^2}}}{2\sqrt{\pi x}} \log_2(1 + e^{-y}) dy. \tag{41}$$

The decoding of concatenated ZH code converges at the intersection point of the EXIT curve and the diagonal line $y = x$, where $I_E = I_A$. Recall that we have M concatenated component codes and the EXIT

chart we considered is mapped from M -dimension to 2-D because of the identical component codes. Therefore, we have

$$\begin{aligned}
 P_b &\approx Q\left(\frac{1}{2}\sqrt{8R\frac{E_b}{N_0} + (M-1) \cdot J^{-1}(I_A)^2 + J^{-1}(I_E)^2}\right) \\
 &= Q\left(\frac{1}{2}\sqrt{8R\frac{E_b}{N_0} + M \cdot J^{-1}(I_x)^2}\right)
 \end{aligned} \tag{42}$$

where I_x denotes the extrinsic mutual information at the crossing point, $I_x = I_A = I_E$. Some contours of error probability for different E_b/N_0 values are illustrated in Fig. 6. In Section IV, we will compare the estimated bit error probabilities from the EXIT charts with simulation results.

B. Distance Spectrum and Union Bound

The union bound is an upper bound for the code error probability. Although the union bound is loose at low SNRs, it has been shown to be very useful for determining the error floor at high SNRs. The union bound can be obtained through the distance spectrum analysis. Define the input-output weight enumerating function (IOWEF) of a code C [22] as

$$A^C(W, Y) = \sum_{w,y} A_{w,y} W^w Y^y \tag{43}$$

where $A_{w,y}$ denotes the number of weight- y codewords generated by weight- w information words. Notice that for a conventional code, $A_{w,y}$ is usually an integer. For a concatenated code, $A_{w,y}$ is defined in the form of an ensemble weight distribution, which is a probability measure averaged over all possible interleavers (for a detailed discussion, see [21], [22]).

Here we consider only the unpunctured codes. If the common bit of a Hadamard codeword is not punctured, there are only three possible weights: 0, 2^{r-1} , and 2^r associated with each segment (i.e., the weight of a length- 2^r Hadamard codeword). Therefore, we only need to consider the terms associated with those y that are multiples of 2^{r-1} . Consequently, the IOWEF approach is computationally more efficient than the method based on the input-redundancy weight enumerating function (IRWEF) [21]. The conditional output weight enumerating function (COWEF) of a code C is given by

$$A_w^C(Y) = \sum_y A_{w,y} Y^y \quad (44)$$

which represents the weight distribution of the codewords generated by weight- w information input. Assume that $C^{(1)}, C^{(2)}, \dots, C^{(M)}$, each containing N information bits, are concatenated in parallel using ‘‘uniform’’ interleavers [21] to form an overall code C . Given the COWEF of the constituent code, $A_w^{C^{(m)}}(Y)$, $m = 1, \dots, M$, we then obtain the COWEF of the concatenated code C as follows.

Nonsystematic Codes: The COWEF of the parallel concatenated code C using uniform interleaver conditioned on the information weight w , $A_w^C(Y)$, is given by [21], [22]

$$A_w^C(Y) = \sum_y A_{w,y} Y^y = \frac{\prod_{m=1}^M A_w^{C^{(m)}}(Y)}{\binom{N}{w}^{M-1}}. \quad (45)$$

Systematic Codes: Notice that in systematic codes, the information weight has been counted for M times above. If every information bit occurs only once in C , the required COWEF of the concatenated code in terms of the COWEF of the component codes is given by

$$\begin{aligned} A_w^C(Y) &= \sum_y A_{w,y} Y^y = \frac{Y^w \prod_{m=1}^M A_w^{C^{(m)}}(Y) Y^{-w}}{\binom{N}{w}^{M-1}} \\ &= \frac{Y^{-(M-1)w} \prod_{m=1}^M A_w^{C^{(m)}}(Y)}{\binom{N}{w}^{M-1}}. \end{aligned} \quad (46)$$

Finally, the IOWEF of parallel concatenated code C is given by

$$A^C(W, Y) = \sum_w \sum_y A_{w,y} W^w Y^y = \sum_{w=0}^N W^w A_w^C(Y). \quad (47)$$

The bit error probability of C is upper-bounded by [21]

$$P_b(e) \leq \sum_w \sum_y \frac{w}{N} A_{w,y} Q \left(\sqrt{2yR \frac{E_b}{N_0}} \right) \quad (48)$$

where R is the code rate of C .

The key issue for the method outlined above is to derive the COWEF in (45) and (46) for the constituent codes, which can be obtained from the IOWEF of the constituent codes. In the following, we derive a simple recursive technique to generate the IOWEF for an unpunctured nonsystematic ZH code. The IOWEF for an unpunctured systematic ZH code with an even r can be constructed similarly. Denote $A^{C_K}(W, Y)$ as the IOWEF of an unpunctured nonsystematic ZH code with K segments. Again we assume that the first bit in the first segment is always zero. Therefore, the first segment can be considered similarly as all other segments. We then decompose $A^{C_K}(W, Y)$ into even and odd parts as

$$A^{C_K}(W, Y) = A_{\text{even}}^{C_K}(W, Y) + A_{\text{odd}}^{C_K}(W, Y) \quad (49)$$

where $A_{\text{even}}^{C_K}(W, Y)$ (respectively, $A_{\text{odd}}^{C_K}(W, Y)$) represents all the terms with even (respectively, odd) weight input of W . First, let us consider $K = 1$. In this case, the weight of output codeword is zero

for the all-zero input and the weight of output codeword is always $n/2$, where $n = 2^r$, for nonzero inputs. Thus, we have

$$A^{C_1}(W, Y) = 1 + \sum_{j=1}^r \binom{r}{j} W^j Y^{n/2}$$

with

$$\begin{aligned} A_{\text{even}}^{C_1}(W, Y) &= 1 + \sum_{j=2,4,6,\dots}^r \binom{r}{j} W^j Y^{n/2} \\ &= \frac{1}{2} ((1+W)^r + (1-W)^r) Y^{n/2} + (1 - Y^{n/2}) \end{aligned} \quad (50)$$

$$\begin{aligned} A_{\text{odd}}^{C_1}(W, Y) &= \sum_{j=1,3,5,\dots}^r \binom{r}{j} W^j Y^{n/2} \\ &= \frac{1}{2} ((1+W)^r - (1-W)^r) Y^{n/2}. \end{aligned} \quad (51)$$

Now suppose that $A^{C_{k-1}}(W, Y)$ is known and based on the zigzag structure, we have

$$\begin{aligned} A_{\text{even}}^{C_k}(W, Y) &= A_{\text{even}}^{C_{k-1}}(W, Y) A_{\text{even}}^{C_1}(W, Y) \\ &\quad + A_{\text{odd}}^{C_{k-1}}(W, Y) A_{\text{odd}}^{C_1}(W, Y) \end{aligned} \quad (52)$$

$$\begin{aligned} A_{\text{odd}}^{C_k}(W, Y) &= A_{\text{even}}^{C_{k-1}}(W, Y) A_{\text{odd}}^{C_1}(W, Y) \\ &\quad + A_{\text{odd}}^{C_{k-1}}(W, Y) (A_{\text{even}}^{C_1}(W, Y) - 1 + Y^n). \end{aligned} \quad (53)$$

The last term in (53) corresponds to the situation that $c_{k-1}(n-1) = 1$ with $n = 2^r$ and the input weights of the k th segment are even. For nonzero inputs to the k th segment, the corresponding term in the IOWEF is $A_{\text{odd}}^{C_{k-1}}(W, Y) (A_{\text{even}}^{C_1}(W, Y) - 1)$. For the zero input to the k th segment, the corresponding term in the IOWEF is $A_{\text{odd}}^{C_{k-1}}(W, Y) Y^n$, since $\bar{d}_k(1) = \bar{d}_k(2) = \dots = 1$. (Recall that in (4), $\bar{d}_k(j) = c_{k-1}(n-1) \oplus d_k(j)$.) This will produce an all-1 Hadamard codeword with weight n , i.e., Y^n . Adding these two terms, we obtain the last term in (53). Other terms in (52) and (53) are easy to verify. Therefore, starting with $k = 1$, we can obtain the IOWEF $A^{C_K}(W, Y)$ for C_K by recursively applying (52) and (53) until $k = K$.

The above discussion is for unpunctured codes. Puncturing the common bits usually causes marginal performance difference. The discussion is more complex for punctured codes. So we will not pursue this issue further.

IV. NUMERICAL RESULTS

In this section, we provide the performance results of concatenated ZH codes from computer simulations. We first present results of long codes to demonstrate their capacity approaching performance. Then, we present the performance of short codes.

A. Performance of Long Concatenated ZH Codes

In Fig. 7, we illustrate the BER performance of the systematic concatenated ZH codes. The interleaver length, or the number of information bits N , is set around 65536 (N is set to be divisible by the Hadamard order r). The concatenated code consists of three component codes, i.e., $M = 3$. The maximum number of decoding iterations is 50. Here we only consider the punctured codes. It is seen from Fig. 7 that all codes with even Hadamard orders have waterfalls in the BER curves. The E_b/N_0 thresholds at $\text{BER} = 10^{-5}$ are listed in Table I and are compared with the threshold results from the EXIT analysis and the capacity bound. It is seen that the simulation results matched very well

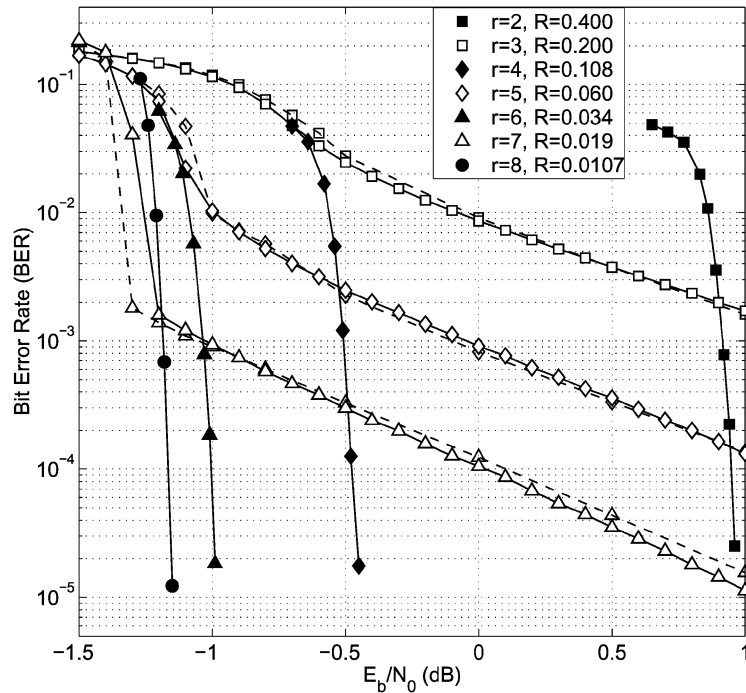


Fig. 7. BER performance of concatenated punctured systematic ZH codes with $M = 3$ and various orders, the interleaver length $N \approx 65\,536$. Dashed lines represent the approximated bit error probability from EXIT charts. The solid lines represent the BER performance from computer simulations.

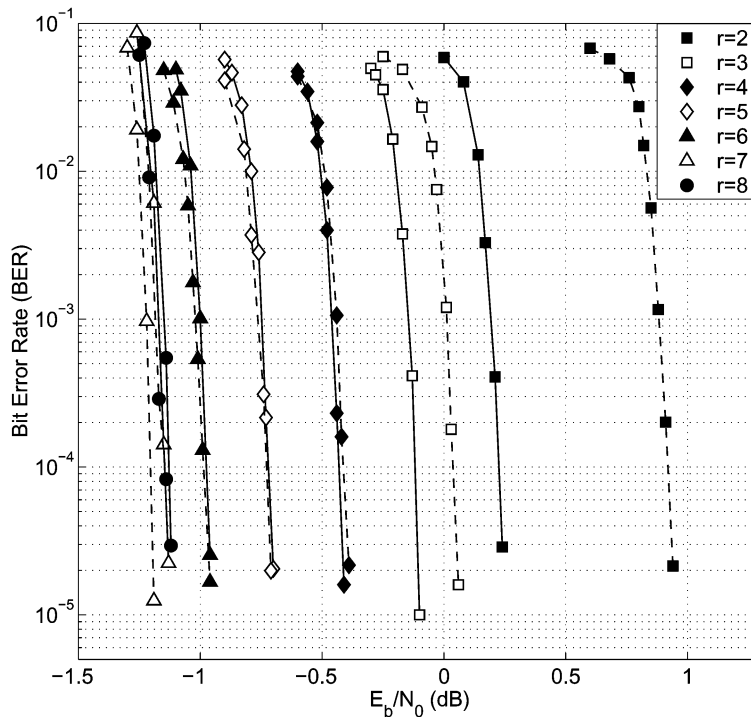


Fig. 8. BER performance of concatenated punctured nonsystematic ZH codes (solid lines) and turbo-Hadamard codes (dashed lines) with $M = 3$ and various orders, the interleaver length $N \approx 65\,536$.

with the EXIT analysis. The difference is within 0.2 dB. Except for the case of $r = 2$, all other codes with even Hadamard orders perform only several tenth decibel away from the Shannon capacity bounds. The smaller the code rate, the closer is its performance to the capacity. This improvement continues until $r = 8$ where the code rate $R = 0.0107$.

In this case, the threshold from the EXIT chart is -1.25 dB, only 0.29 dB away from the capacity. The E_b/N_0 threshold from the simu-

lation is -1.15 dB, with a 0.39-dB gap from the capacity and 0.44 dB from the ultimate Shannon limit.

It is also seen from Fig. 7 that the systematic codes with odd r do not have thresholds. The reason is explained in the previous section. The converged error probabilities estimated from the EXIT chart are illustrated by the dashed curves in the same figure. It is seen that the estimated BER performance matches very well with the simulations.

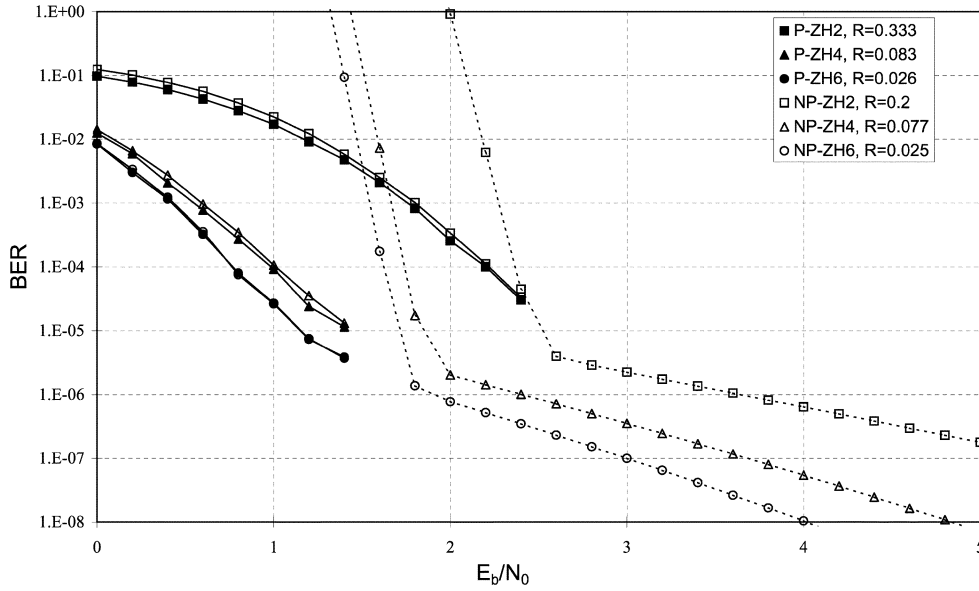


Fig. 9. Performance of short *systematic* concatenated ZH codes: $N \approx 200$, $M = 4$. The dashed lines represent upper bounds. The solid lines represent the BER performance from computer simulations.

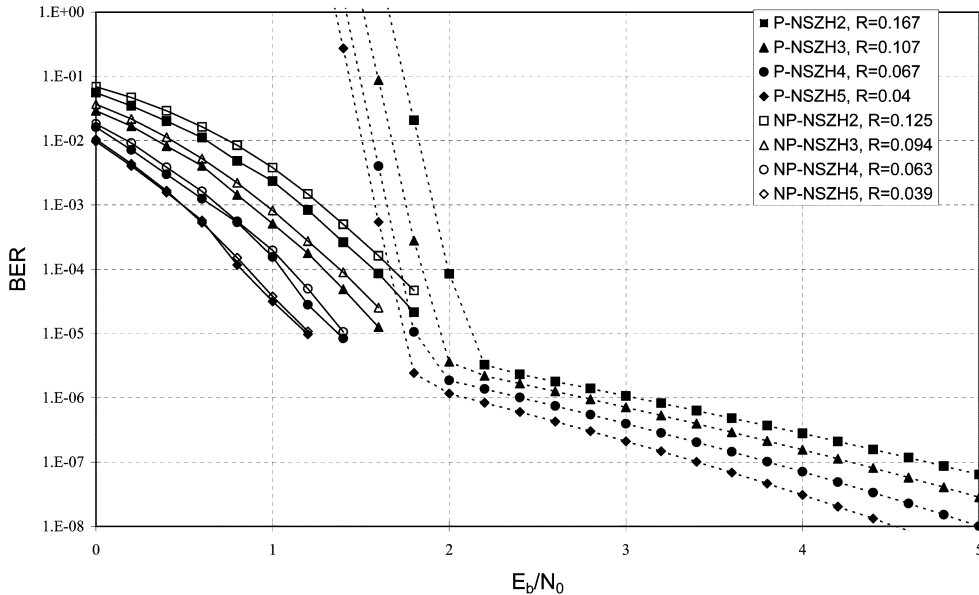


Fig. 10. Performance of short *nonsystematic* concatenated ZH codes: $N \approx 200$, $M = 4$. The dashed lines represent upper bounds. The solid lines represent the BER performance from computer simulations.

These demonstrate that the EXIT chart is a useful tool to analyze the asymptotic performance of the concatenated ZH codes with large code length.

The BER performance of nonsystematic ZH codes and the turbo-Hadamard codes is illustrated in Fig. 8. The code rate of turbo-Hadamard codes is given by

$$R = \frac{r}{r + M(2^r - r)}. \quad (54)$$

All nonsystematic codes with different order r , $r = 2, 3, \dots, 8$, exhibit a sharp waterfall. The E_b/N_0 thresholds are also summarized in Table I. Similarly to the systematic ZH codes, we find that the nonsystematic ZH codes perform well in the low-rate region. The best performance is from the code with $r = 7$ and $R = 0.018$. The E_b/N_0 threshold from the simulation is -1.10 dB, 0.43 dB from the Shannon

capacity and only 0.49 dB from ultimate low-rate bound. The concatenated ZH code with $r = 8$ has a lower code rate but performs almost the same as the one with $r = 7$. Compared with best turbo-Hadamard codes, the concatenated nonsystematic ZH codes with $r = 7$ performs only 0.06 dB worse.

B. Performance of Short Concatenated ZH Codes

We now present simulation results for the concatenated ZH codes with short code length. For short codes, we set $N \approx 200$ and $M = 4$. The maximum iteration number is 30. For all the short codes, we use random interleavers. No attempt has been made to optimize the interleavers.

Fig. 9 illustrates the performance results of systematic codes. The BER performance of both punctured (denoted as “P” in the legend) and unpunctured (denoted as “NP” in the legend) concatenated ZH codes

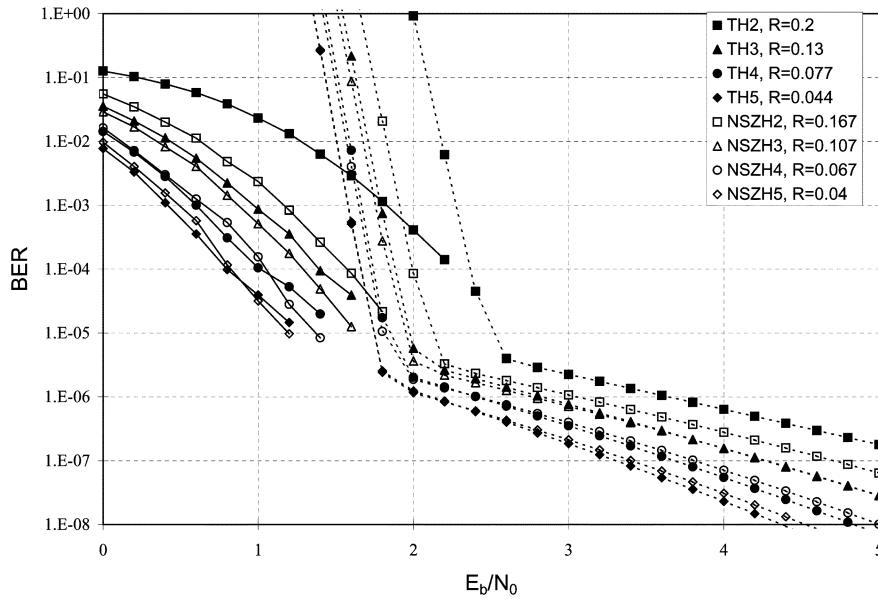


Fig. 11. Performance comparisons of punctured nonsystematic concatenated ZH codes and turbo-Hadamard (TH) codes: $N \approx 200, M = 4$. The dashed lines represent upper bounds. The solid lines represent the BER performance from computer simulations.

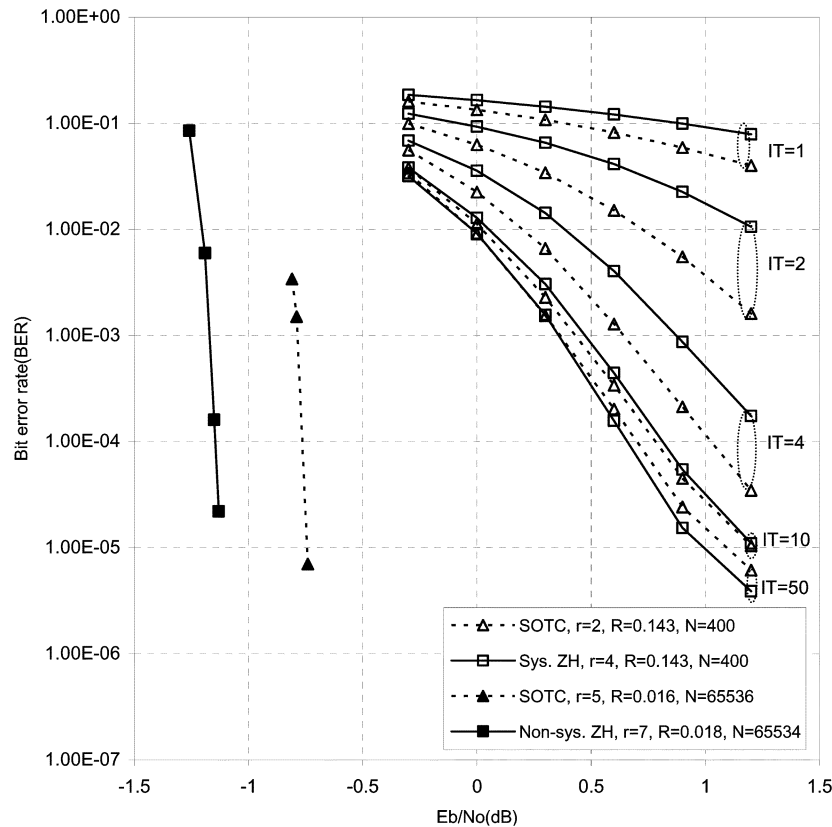


Fig. 12. Performance comparison between concatenated ZH codes and superorthogonal turbo (SOT) code. The generator polynomial of SOT codes is $1/(1 + x + x^3)$ for short codes ($N = 400$) and $1/(1 + x^2 + x^3 + x^5 + x^6)$ for long codes ($N = 65\ 536$), respectively.

and the union bound for unpunctured codes are presented. The number after “ZH” in the legend represents the Hadamard order r . From the simulations, we find that punctured and unpunctured codes perform almost the same as long as their code rates are close. The punctured codes perform slightly better. It is seen that the E_b/N_0 values where the BER reaches 10^{-5} are 2.6, 1.4, and 1.2 dB for the codes with $r = 2, 4$, and 6, respectively. Similar results can be found for non-systematic codes, illustrated in Fig. 10.

The performance of the concatenated punctured nonsystematic ZH codes and that of the turbo-Hadamard codes is compared in Fig. 11 for orders $r = 2, 3, 4$, and 5. The generator polynomial of the turbo-Hadamard code is $1/(1 + D)$. In Figs. 9 and 10, it has been demonstrated that the simulated performance of unpunctured concatenated ZH code is very close to that of the punctured one. Therefore, for the nonsystematic ZH code, we only present the simulation results of punctured codes. We find that from the union bound results, if the two dif-

ferent codes have the same order r and their code rates are close, the turbo-Hadamard codes perform slightly better in the error floor region than the nonsystematic ZH codes. However, from the simulations it is seen that the concatenated ZH codes have lower error floor when $E_b/N_0 > 0.9$ dB for $r = 5$, and when $E_b/N_0 > 1.05$ dB for $r = 4$. For $r \leq 3$, the concatenated ZH codes outperform turbo-Hadamard codes in the whole E_b/N_0 region.

Performance Comparison With Other Low-Rate Codes: Super-orthogonal turbo (SOT) codes are another class of low-rate capacity approaching codes [6]. Fig. 12 presents a performance comparison between the concatenated ZH codes and SOT codes. The maximum number of decoding iterations is 50 for both concatenated ZH codes and SOT codes. Both long and short codes are compared. In Fig. 12, the BER performance after a certain number of iterations is denoted by “IT = {number of iterations}.”

The generator polynomial of the short SOT code (information length = 400) is $1/(1+x+x^3)$ with the associated Hadamard order $r = 2$ and $R = 1/7 = 0.143$. The short concatenated ZH code (information length = 400) is systematic with punctured common bits and $r = 4$. Some extra puncturing is applied to this code to reduce its rate to $R = 1/7 = 0.143$, matching it with the SOT one. We can see that the performances of the two codes are quite similar, with the SOT code converges faster in the first few iterations. After about ten iterations, the concatenated ZH code becomes slightly better. Based on the discussion in Section II-D, the decoding complexity of the concatenated ZH code with $r = 4$ is about $4 \times (16 + 9 \times 16/4) = 208$ additions per information bit per iteration. The short SOT code has 8 states and its decoding involves approximately 240 operations (including additions and multiplications) per information bit per iteration. (We have considered the fact that the SOT code has two component codes and the concatenated ZH one has four. We have also considered the necessary normalization operations during the decoding processes for both codes.)

Next we compare the long codes. The generator polynomial of the long SOT code ($N = 65\ 536$) is $1/(1+x^2+x^3+x^5+x^6)$ with the associated Hadamard order $r = 5$ and $R = 0.016$. The long ZH code ($N = 65\ 534$) is nonsystematic with punctured common bits, $r = 7$ and $R = 0.018$. It is seen that the long concatenated ZH code performs 0.38 dB better than the long SOT one and the concatenated ZH also has lower complexity.

V. CONCLUSION

In this correspondence, we have presented a new class of low-rate error correction codes, the concatenated *zigzag Hadamard* (ZH) codes. This type of codes enjoy extremely simple encoding and very low-complexity decoding. We have presented the performance analysis of such codes including the EXIT chart analysis for infinite-length codes and the union bound analysis for finite-length codes. The results show that a rate-0.0107 concatenated ZH code performs only 0.44 dB away from the ultimate low-rate Shannon limit. Compared with other low-rate codes, the proposed concatenated ZH codes offer similar performance as the turbo-Hadamard codes, and better performance than superorthogonal turbo codes, but with much lower encoding and decoding complexities.

REFERENCES

- [1] C. Shannon, “A mathematical theory of communication,” *Bell Syst. Tech. J.*, vol. 27, pp. 379–423, Jul./Oct. 1948.
- [2] F. MacWilliams and N. Sloane, *The Theory of Error-Correcting Codes*. Amsterdam, The Netherlands: North-Holland, 1997.
- [3] R. Yarlagadda and J. Hershey, *Hadamard Matrix Analysis and Synthesis: With Applications to Communications and Signal/Image Processing*. Boston, MA: Kluwer, 1997.

- [4] A. Viterbi, “Orthogonal tree codes for communication in the presence of white Gaussian noise,” *IEEE Trans. Commun. Technol.*, vol. COM-15, no. 4, pp. 238–242, Apr. 1967.
- [5] C. Berrou, A. Glavieux, and P. Thitimajshima, “Near-Shannon limit error-correcting and decoding: Turbo-codes,” in *Proc. Int. Conf. Communications*, Geneva, Switzerland, May 1993, pp. 1064–1070.
- [6] P. Komulainen and K. Pehkonen, “Performance evaluation of super-orthogonal turbo codes in AWGN and flat Rayleigh fading channels,” *IEEE J. Sel. Areas Commun.*, vol. 16, no. 2, pp. 196–205, Feb. 1998.
- [7] M. Bingeman and A. Khandani, “Symbol based turbo codes,” *IEEE Commun. Lett.*, vol. 3, no. 10, pp. 285–287, Oct. 1999.
- [8] H. Jin and R. McEliece, “RA codes achieve AWGN channel capacity,” in *Proc. Int. Symp. Applied Algebra and Error-Correcting Codes*, Honolulu, HI, Nov. 1999, pp. 14–19.
- [9] R. Gallager, “Low-density parity-check codes,” *IRE Trans. Inf. Theory*, vol. IT-8, no. 1, pp. 37–45, Jan. 1962.
- [10] D. MacKay and R. Neal, “Near Shannon limit performance of low density parity check codes,” *Electron. Lett.*, vol. 33, no. 6, pp. 457–458, Mar. 1997.
- [11] S. Chung, G. Forney, T. Richardson, and R. Urbanke, “On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit,” *IEEE Commun. Lett.*, vol. 5, no. 2, pp. 58–60, Feb. 2001.
- [12] T. Richardson, M. Shokrohalli, and R. Urbanke, “Design of capacity approaching irregular low density parity check codes,” *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.
- [13] O. Wintzell, M. Lentmaier, and K. Zigangirov, “Asymptotic analysis of superorthogonal turbo codes,” *IEEE Trans. Inf. Theory*, vol. 49, no. 1, pp. 253–258, Jan. 2003.
- [14] J. Hamkins and D. Divsalar, “Coupled receiver-decoders for low rate turbo codes,” in *Proc. IEEE Int. Symp. Information Theory*, Yokohama, Japan, Jun./Jul. 2003, p. 381.
- [15] L. Ping, W. Leung, and K. Wu, “Low-rate turbo-Hadamard codes,” *IEEE Trans. Inf. Theory*, vol. 49, no. 12, pp. 3213–3224, Dec. 2003.
- [16] L. Ping, X. Huang, and N. Phamdo, “Zigzag codes and concatenated zigzag codes,” *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 800–807, Feb. 2001.
- [17] L. Ping and K. Wu, “Concatenated tree codes: A low-complexity high-performance approach,” *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 791–799, Feb. 2001.
- [18] R. Elliott, L. Aggoun, and J. Moore, *Hidden Markov Models: Estimation and Control*. New York: Springer-Verlag, 1995.
- [19] S. ten Brink, “Convergence of iterative decoding,” *Electron. Lett.*, vol. 35, no. 13, pp. 1117–1118, Jun. 1999.
- [20] —, “Convergence behavior of iterative decoded parallel concatenated codes,” *IEEE Trans. Commun.*, vol. 49, no. 10, pp. 1727–1737, Oct. 2001.
- [21] S. Benedetto and G. Montorsi, “Unveiling turbo codes: Some results on parallel concatenated coding schemes,” *IEEE Trans. Inf. Theory*, vol. 42, no. 2, pp. 409–428, Mar. 1996.
- [22] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, “Analysis, design, and iterative decoding of double serially concatenated codes with interleavers,” *IEEE J. Sel. Areas Commun.*, vol. 16, no. 2, pp. 231–244, Feb. 1998.