

Low-Rate Generalized Low-Density Parity-Check Codes with Hadamard Constraints

Guosen Yue

NEC Laboratories America, Inc.
4 Independence Way
Princeton, NJ 08540

Li Ping

Dept. of Electronic Engineering
City University of Hong Kong
Kowloon, Hong Kong

Xiaodong Wang

Dept. of Electrical Engineering
Columbia University
New York, NY 10027.

Abstract—We consider the design and analysis of generalized low-density parity-check (GLDPC) codes specified by a bipartite Tanner graph, as with standard LDPC codes, but with the single parity-check constraints replaced by general coding constraints. In particular, we consider imposing Hadamard code constraints at the check nodes for a low-rate approach, termed LDPC-Hadamard codes. The achievable capacity with the GLDPC codes is then discussed. A modified LDPC-Hadamard code graph is also proposed. We then optimize the LDPC-Hadamard code ensemble using a low-complexity optimization method based on approximating the density evolution by a one-dimensional dynamic system represented by an extrinsic mutual information transfer (EXIT) chart. Simulation results show that a rate-0.003 LDPC-Hadamard code with large block length can achieve a bit-error-rate (BER) performance of 10^{-5} at -1.44 dB, only 0.15 dB away from the ultimate Shannon limit (-1.592 dB).

I. INTRODUCTION

Near Shannon-capacity achieving codes have attracted increasing interest during the last decade. One of the important milestones has been the re-discovery of low-density parity-check (LDPC) codes [1], which were shown to asymptotically achieve the capacity of a binary erasure channel under iterative message-passing decoding [2], [3]. A standard LDPC code is characterized by the random connection of variable nodes and check nodes. This principle can be generalized, for example, by replacing the the single parity check (SPC) constraints at the check nodes in the standard LDPC code by constraints based on other block codes, such as Hamming codes and BCH codes [4], [5]. With more powerful (compared with simple SPC codes) block codes involved, GLDPC codes have many potential advantages, including improved performance in very noisy channels (i.e., low rate applications), fast convergence speed and low error floor.

However, the design of a GLDPC code remains a quite difficult issue. This may be attributed to two factors. First, since more complex codes are used at the check nodes, more complex *a posteriori* probability (APP) decoders are necessary. Thus decoding complexity may become a serious concern. Second, the code optimization techniques based on density evolution [3] or extrinsic mutual information transfer (EXIT) charts [6] for the standard LDPC codes may not be effective for GLDPC codes. In this paper, we study a family of GLDPC codes that use Hadamard codes at the check nodes. The complexity issue mentioned above is solved by employing the fast APP decoding technique for the Hadamard codes [7].

With regard to the code optimization issue, we analyze the difficulties in directly applying density evolution and EXIT chart based design techniques to GLDPC codes. Difficulties include mismatching of EXIT functions and the halting of mutual information (or density) evolution caused by degree-1 variables nodes. We propose a solution to both difficulties by introducing the so-called LDPC-Hadamard code structure. We demonstrate the optimization procedure for LDPC-Hadamard codes using the low-cost EXIT chart technique. We will focus on low-rate applications and demonstrate an LDPC-Hadamard code with large block length that achieves 10^{-5} BER at $E_b/N_0 = -1.44$ dB. This represents an improvement towards the Shannon limit, the ultimate Shannon capacity being $E_b/N_0 = -1.592$ dB for the codes with rates approaching zero [7], with a remaining gap of only 0.15 dB.

II. CODE STRUCTURE

A. Hadamard Codes

An $n \times n$ ($n = 2^r$) Hadamard matrix \mathbf{H}_n over $\{+1, -1\}$ can be constructed recursively as

$$\mathbf{H}_n = \begin{bmatrix} +\mathbf{H}_{\frac{n}{2}} & +\mathbf{H}_{\frac{n}{2}} \\ +\mathbf{H}_{\frac{n}{2}} & -\mathbf{H}_{\frac{n}{2}} \end{bmatrix} \text{ with } \mathbf{H}_1 = [+1]. \quad (1)$$

A length- 2^r Hadamard codeword set is then formed by the columns of the bi-orthogonal 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 with $+1 \rightarrow 0$ 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 of a length- 2^r Hadamard code, the information bit positions are indicated by indexes $\{0, 1, 2, 4, \dots, 2^{r-1}\}$. The other bits are parity bits. Denote \mathcal{H} as a Hadamard encoder. Then, given the input information sequence $b(0), b(1), \dots, b(r)$, the Hadamard encoding can be represented by $\mathbf{c} = \mathcal{H}(b(0), b(1), \dots, b(r))$, where $\mathbf{c} = [c(0), c(1), \dots, c(2^r - 1)]$. Clearly, \mathbf{c} is a column of either $+\mathbf{H}_n$ or $-\mathbf{H}_n$. For *systematic* Hadamard codes, we have $c(0) = b(0)$ and $c(2^{j-1}) = b(j)$, $j = 1, \dots, r$.

We can also construct a *non-systematic* Hadamard code as follows. First, let

$$\tilde{b}(j) = b(0) \oplus b(j), \quad j = 1, 2, \dots, r, \quad (2)$$

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

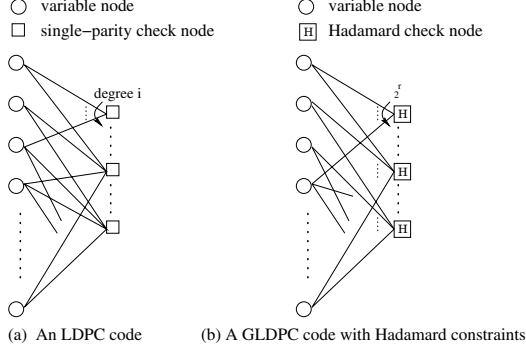


Fig. 1. Bipartite Tanner graph representations of a standard LDPC code and a GLDPC code with Hadamard constraints.

$\mathcal{H}(b(0), \tilde{b}(1), \tilde{b}(2), \dots, \tilde{b}(r))$, where $c(0) = b(0)$, $c(2^{j-1}) = \tilde{b}(j)$, $j = 1, \dots, r$.

B. LDPC and GLDPC Codes

Illustrated on the left part of Fig. 1, an LDPC code can be represented by a bipartite Tanner graph with the edges connecting the left and right nodes that satisfy both repetition code constraints at the variable nodes and SPC constraints at the check nodes. Based on the bipartite graph in Fig. 1, a GLDPC code can be defined by replacing the right SPC codes in an LDPC code with any other block codes, for example, Hamming codes or BCH codes [5], [8], [9]. In this paper, we consider GLDPC's using Hadamard codes at the right check nodes. Such codes have good performance in very noisy channels, as demonstrated below. We call the right nodes in Fig. 1(b) *Hadamard check nodes*. The Hadamard check nodes can either be systematic Hadamard codes or non-systematic Hadamard codes.

III. THE DECODER STRUCTURE

A. Generalized Message-Passing Algorithm

Similarly to the LDPC decoding algorithm, we can decode a GLDPC code iteratively by applying generalized message-passing based on the Tanner graph representation. The extrinsic messages are iteratively exchanged between the variable nodes and the Hadamard check nodes as the input and output of the decoders. After a certain number of iterations, the decoder outputs the decisions.

B. Fast APP Decoding of Hadamard Codes

Denote $\mathbf{x} = [x(0), x(1), \dots, x(2^r - 1)]^T$ as the received sequence of Hadamard code bits c corrupted by AWGN with variance σ^2 . Given the *a priori* LLRs of the Hadamard coded bits, $\mathbf{L}_{apr} = [L_{apr}(0), L_{apr}(1), \dots, L_{apr}(2^r - 1)]^T$, the LLRs from the APP decoding over $\{+1, -1\}$ are obtained by

$$L_{app}(i) = \log \frac{\sum_{\mathbf{h}^j \in \{\mathbf{H} : \mathbf{H}(i,j) = \pm 1\}} \gamma(\pm \mathbf{h}^j)}{\sum_{\mathbf{h}^j \in \{\mathbf{H} : \mathbf{H}(i,j) = \mp 1\}} \gamma(\pm \mathbf{h}^j)},$$

where $\gamma(\pm \mathbf{h}^j) \triangleq \exp\left(\frac{1}{2}(\pm \mathbf{h}^j, \frac{2\mathbf{x}}{\sigma^2} + \mathbf{L}_{apr})\right)$ and $\langle \cdot, \cdot \rangle$ denotes the inner product. The extrinsic LLR is then given

by $L_{ext}(i) = L_{app}(i) - L_{apr}(i)$. Direct calculation of $L_{app}(i)$ has a high computational complexity, i.e., $\mathcal{O}(2^{2r})$. The fast Hadamard transform (FHT) [7] can be employed to reduce the complexity of the inner product based on the butterfly graph of the Hadamard matrix. The summations in the numerator and denominator in $L_{app}(i)$ for computing $L_{app}(i)$, $i = 0, 1, \dots, 2^r - 1$, can be efficiently computed by the APP-FHT algorithm [7]. With the APP-FHT algorithm, the complexity for calculating $L_{app}(i)$ is reduced to $\mathcal{O}(r2^r)$.

C. APP Decoding of Non-systematic Hadamard Codes

Let us compare the non-systematic Hadamard codewords with the corresponding systematic Hadamard codes of the same order r . When $b_0 = 0$, the codeword set of the non-systematic Hadamard codes is exactly the same as that of the systematic codes. When $b_0 = 1$, the non-systematic codeword with the input $b_1 b_2 \dots b_r$ is equal to the systematic codeword $\bar{b}_1, \bar{b}_2, \dots, \bar{b}_r$, i.e., the binary complementary sequence of input $b_1 b_2 \dots b_r$. Therefore, the APP decoding of non-systematic Hadamard code can be obtained by computing $L_{app}(i)$ with $\gamma(\pm \mathbf{h}^j)$ replaced by pre-processed $\gamma'(\pm \mathbf{h}^j)$ following the rules below:

- $\gamma'(+\mathbf{h}^j) = \gamma(+\mathbf{h}^j)$, $j = 0, \dots, 2^r - 1$.
- $\gamma'(-\mathbf{h}^j) = \gamma(-\mathbf{h}^k)$, where j is the binary complementary of k .

IV. GLDPC CODE DESIGN AND ANALYSIS

A. The EXIT Function Mismatching Problem

The iterative decoding processes can be represented by an EXIT chart with two EXIT curves [6], [10]. One is for the decoding of repetition codes at the variable nodes, the other is for the Hadamard decoding at the check node. Define $I_{A,V}$ and $I_{E,V}$ as the input and output extrinsic mutual information of the decoding process for variable nodes, respectively. Similarly define $I_{A,H}$ and $I_{E,H}$ for the Hadamard check nodes. The EXIT functions for the variable nodes of degree- j and Hadamard check nodes of order r are denoted by $\mathcal{G}_{V,j}(I_{A,V}, j, \frac{E_s}{N_0})$ and $\mathcal{G}_{H,r}(I_{A,H}, r)$, respectively. The value of $\mathcal{G}_{V,j}(I_{A,V}, j, \frac{E_s}{N_0})$ and $\mathcal{G}_{H,r}(I_{A,H}, r)$ can be computed from the output extrinsic conditional pdfs [10]. For binary codes, the EXIT curve of a mixture of codes is an average of the EXIT curves of its component codes [11], [12]. We specify the LDPC-Hadamard code ensemble by the profiles of $\{\lambda_j\}_{j=1}^{d_{v,\max}}$ and $\{\rho_r\}_{r=1}^{d_{r,\max}}$, where r denotes the order of the Hadamard check nodes. The average output extrinsic mutual information is then given by

$$I_{E,V} \triangleq \mathcal{G}_V(I_{A,V}, \frac{E_s}{N_0}) = \sum_j \lambda_j \mathcal{G}_{V,j}(I_{A,V}, j, \frac{E_s}{N_0}),$$

$$I_{E,H} \triangleq \mathcal{G}_H(I_{A,H}) = \sum_r \rho_r \mathcal{G}_{H,r}(I_{A,H}, r).$$

Figure 2 shows the EXIT curves for GLDPC codes in AWGN channels. In [11], it is shown that to approach capacity on erasure channels for an LDPC code, the EXIT curve of the variable nodes must match with the EXIT curve of the check nodes. We conjecture that this is also true for GLDPC codes in AWGN channel. Based on this conjecture, we now examine the EXIT curves in Fig. 2. The set of curves for

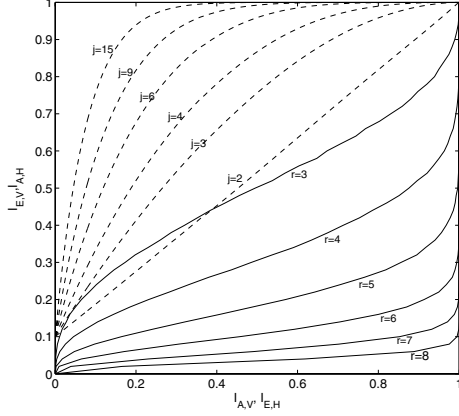


Fig. 2. EXIT curves for GLDPC codes. Dashed lines: variable nodes of degree- j , solid lines: Hadamard check nodes of order- r .

repetition codes are concave and do not have saddle points. The EXIT curves for SPC codes are also concave and so they may be optimized for matching. However, the EXIT curves for the Hadamard codes are neither concave or convex. They have saddle points (except for $r = 2$, which is equivalent to a length-4 SPC code). Also, it is easy to verify that any linear combinations of the curves for Hadamard codes of $r \geq 3$ have saddle points. Thus we cannot use these codes for matching. Consequently, the EXIT curves of the variable and Hadamard check nodes cannot be optimized to match with each other.

B. The Degree-1 Nodes Problem

The code rate of GLDPC is given by

$$R = 1 - \frac{\sum_{j=1}^{d_c^{\max}} \rho_j (1 - R_{c,j})}{\sum_{j=1}^{d_v^{\max}} \lambda_j / j}.$$

Since $\sum_{j=1}^{d_v^{\max}} \lambda_j = 1$, we have

$$\sum_{j=1}^{d_v^{\max}} \lambda_j / j \leq \lambda_1 + \frac{1 - \lambda_1}{2} = \frac{1 + \lambda_1}{2}.$$

Then,

$$\lambda_1 \geq \frac{2 \sum_{j=1}^{d_c^{\max}} \rho_j (1 - R_{c,j})}{1 - R} - 1 = \frac{1 - 2R_c + R}{1 - R},$$

where R_c is the average code rate at check nodes. Assuming $0 < R < 1$, we then have the following theorem from (IV-B).

Theorem 4.1: For any GLDPC codes with the left nodes being the repetition codes, if the average code rate at check nodes, R_c , is smaller than $\frac{1}{2}$, then the code contains a non-negligible portion of degree-1 nodes, i.e., λ_1 is bounded away from 0.

Since $R_c = \sum_j \rho_j R_{c,j} < \max_j R_{c,j}$ and $R_c < \max_i r_{c,i}$, we have the following corollary.

Corollary 4.1: For any GLDPC codes with the left nodes being the repetition codes, if $\max_j R_{c,j} < \frac{1}{2}$ or $\max_i r_{c,i} < \frac{1}{2}$, λ_1 is bounded away from 0.

For a GLDPC code with Hadamard constraints, when $r \geq 3$, we have $R_c = \sum_r \rho_r \frac{r+1}{2r} \leq \frac{1}{2}$. Then, we have λ_1 bounded by

$$\lambda_1 > 1 - 2R_c = 1 - \frac{r+1}{2r-1}.$$

For $r = 3, 4, 5, 6, 7, 8, \dots$, we have $\lambda_1 > 0, \frac{3}{8}, \frac{5}{8}, \frac{25}{32}, \frac{7}{8}, \frac{119}{128}, \dots$. As r increases, the proportion of degree-1 nodes increases.

From the standard EXIT chart analysis of LDPC codes, we know that the outbound extrinsic messages of degree-1 nodes

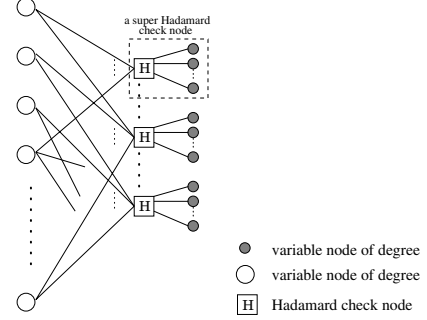


Fig. 3. Modified bipartite Tanner graph representation of an LDPC-Hadamard code.

will not be updated during the iterative decoding process. This indicates that when λ_1 is not bounded away from zero, we cannot use the EXIT chart technique to obtain a threshold where the EXIT information converges to 1 since the outbound extrinsic messages from degree-1 nodes remain unchanged. It should be noted that the discussion above does not mean that a threshold does not exist by iterative decoding. It only states that we cannot get the threshold, if it exists, by the EXIT chart technique. The reason is that the mutual information used in the EXIT chart method is averaged over all edges. Global convergence in actual decoding can be achieved even if only some of messages converge.

C. LDPC-Hadamard Code

We now propose a new code structure for LDPC-Hadamard codes to solve the two problems mentioned above. The code graph of a LDPC-Hadamard codes is illustrated in Fig. 3. A LDPC-Hadamard code C is generated using a two-step approach.

- Step 1: Construct a LDPC code C^L . We will assume that there are no degree-1 variable nodes in C^L .
- Step 2: Use the variable bits connected to every check node in C^L to encode a Hadamard code.

The LDPC-Hadamard code C is then formed by the union of C^L and the Hadamard codes generated in Step 2. The efficient approach to generate the Hadamard codes in Step 2 is to use the m variable bits to encode an order- r Hadamard code with $r = m - 2$ based on the theorem below.

Theorem 4.2: In a systematic Hadamard code with even order r , and a non-systematic Hadamard code with any order $r > 1$, we have the following SPC constraint

$$c(2^r - 1) \oplus \sum_{i=0, \dots, r} \oplus b(i) = 0, \quad (3)$$

where \oplus denotes binary addition and $\sum_{i=0, \dots, r} \oplus$ denotes binary summation. Moreover, the SPC constraint in (3) is not satisfied by systematic Hadamard codes with odd r .

The proof is omitted due to space limitation.

With Theorem 4.2, we can build the check nodes for an LDPC-Hadamard code using a super check node as shown in Fig. 3. We first design a standard LDPC code C^L . We then replace each check node by a Hadamard check node consisting

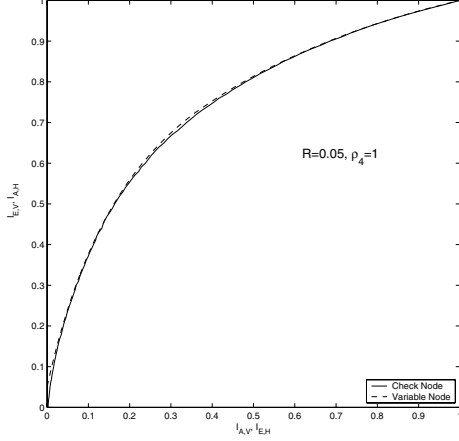


Fig. 4. EXIT charts for optimized LDPC-Hadamard codes.

of a Hadamard check node together with the attached degree-1 nodes. The edges connected to a super check node are the same as those connected to the corresponding check node in C^L , and thus the variable nodes on the left side remain unchanged. The degree-1 nodes inside a super check node represent the extra bits generated during the Hadamard encoding. These degree-1 nodes include all the parity bits of H except $c(2^r - 1)$. (See (3).) As a brief summary, an LDPC-Hadamard code described above can be simply viewed as a standard LDPC encoder cascaded with a special form of Hadamard encoder. Graphically, the extra parity bits generated by the Hadamard encoding form the degree-1 variable nodes as each of them is connected only to one super check node. Empirically, we observed that with the modified structure, the EXIT curves for the output from the check nodes to the non-degree-1 nodes can be concave, which makes the EXIT function matching possible.

We now redefine the profiles of the LDPC-Hadamard codes built on the modified code graph as follows. We only consider the proportions of the edges connected to the non-degree-1 nodes. Define λ_j as the fraction of the edges connected to the nodes of degree- j over all the edges connected to the non-degree-1 nodes and ρ_r as the fraction of edges that are connected to the check nodes. The rates of LDPC-Hadamard codes built with systematic or non-systematic Hadamard check nodes are then, respectively, given by

$$R_{\text{sys}} = 1 - \frac{\sum_{r=2,4,\dots} \frac{\rho_r(2^r - (r+1))}{(r+2)}}{\sum_{j=2}^{d_{v\text{max}}} \lambda_j/j + \sum_{r=2,4,\dots} \frac{\rho_r(2^r - (r+2))}{(r+2)}},$$

$$R_{\text{NS}} = 1 - \frac{\sum_{r=2,3,\dots} \frac{\rho_r(2^r - 1)}{(r+2)}}{\sum_{j=2}^{d_{v\text{max}}} \lambda_j/j + \sum_r \frac{\rho_r(2^r - 2)}{(r+2)}}.$$

D. Code Optimization

Using the *convergence property* [6] and the *area property* [11] of the EXIT chart, we can form design rules for LDPC-Hadamard codes. Given E_s/N_0 , the LDPC-Hadamard code ensemble can be designed by enforcing the EXIT transfer function of the decoder to satisfy $\mathcal{G}_H^{-1}(x, \frac{E_s}{N_0}) < \mathcal{G}_V(x, \frac{E_s}{N_0})$ and maximizing the code rate. The optimization problem can be summarized as follows:

$$\text{maximize } \sum_{j=2}^{d_{v\text{max}}} \lambda_j/j,$$

$$\text{s.t. } \mathcal{G}_V(x, \frac{E_s}{N_0}) > \mathcal{G}_H^{-1}(x, \frac{E_s}{N_0}), \text{ with } \sum_{j=2}^{d_{v\text{max}}} \lambda_j = 1.$$

Given the capacity limit E_s/N_0 of a certain code rate, we can obtain $\mathcal{G}_H^{-1}(x, m_{\text{ch}})$ using Monte Carlo simulations. We then obtain $\{\lambda_j\}$ by solving the above optimization problem, which can be easily solved using linear programming [13]. One optimization result is shown in Fig. 4 for the systematic codes at the check nodes with $\rho_4 = 1$. The designed rate is $R = 0.05$. It can be seen that the optimized EXIT curve of function $\mathcal{G}_V(x, \frac{E_s}{N_0})$ matches very well with that of function $\mathcal{G}_H^{-1}(x, \frac{E_s}{N_0})$.

V. SIMULATION RESULTS

A. Approaching Low-Rate Limit

We consider several examples to demonstrate that the optimized LDPC-Hadamard codes exhibit performance approaching the low-rate Shannon limit.

Example of an LDPC-Hadamard code with $r = 4$: In the first example, we consider the LDPC-Hadamard code with systematic Hadamard check nodes of order $r = 4$, i.e., $\rho_4 = 1$. The designed LDPC-Hadamard code has the rate $R = 0.050$. The BER performance is illustrated in Fig. 5. The error bits are counted from all the variable nodes with degree $j > 1$. The performance of low-rate turbo Hadamard codes with various orders is also illustrated in the same figure for comparison. The number of concatenated component codes in the turbo Hadamard codes is $M = 3$. The rates of turbo Hadamard codes with $r = 5, 6$, and 7 are $0.058, 0.033$, and 0.019 , respectively. For a fair comparison, we set the information length for all these codes to be approximately equal, i.e., $K \approx 65536$. The results show that the LDPC-Hadamard code has a 10^{-5} BER performance at $E_b/N_0 = -1.18$ dB, outperforming the turbo Hadamard code with the closest rate ($r = 5, R = 0.058$) by 0.48 dB. It is seen that the designed LDPC-Hadamard code even performs 0.23 dB better than the turbo Hadamard code with lower rate ($r = 6, R = 0.033$), and performs very close to the best turbo Hadamard code with $r = 7, R = 0.019$. It is only 0.26 dB shy of the Shannon capacity for $R = 0.05$ with binary input (-1.44 dB) and 0.41 dB away from the ultimate low-rate limit.

Examples of LDPC-Hadamard codes with high orders:

We now consider the design of LDPC-Hadamard codes with higher orders and lower rates to push the performance nearer to the ultimate Shannon limit. Fig. 6 illustrates the BER performance of designed LDPC-Hadamard codes with $r = 8, R = 0.0080$, and $r = 10, R = 0.0030$. It is seen that the E_b/N_0 threshold of the ($r = 8, R = 0.008$) LDPC-Hadamard code with $K = 71,000$ is -1.36 dB, improving the best threshold of the turbo Hadamard, $E_b/N_0 = -1.18$ dB, by 0.18 dB. The gap between performance of this LDPC-Hadamard code and the ultimate low-rate limit is now only 0.23 dB. By increasing the code length to $K = 238,000$, the threshold is improved to -1.38 dB, and the gap to the Shannon limit is reduced to 0.21 dB. With an extremely low-rate approach, i.e. $r = 10, R = 0.0030$, and extremely

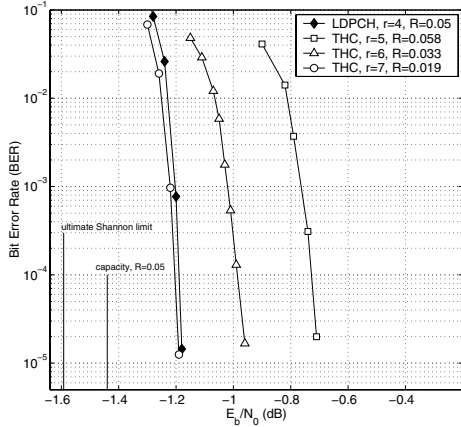


Fig. 5. BER performance of designed LDPC-Hadamard code with $r = 4$ and turbo Hadamard codes with $M = 3$, $r = 5, 6, 7$. $K \approx 65536$.

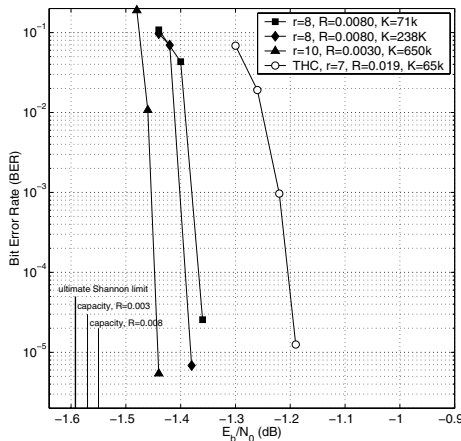


Fig. 6. BER performance of designed LDPC-Hadamard codes and turbo Hadamard codes with $M = 3$, $r = 7$.

long code, i.e. $K = 650,000$, the E_b/N_0 threshold from the simulation is -1.44 dB which is only -0.15 dB from the theoretical Shannon limit for the code rate approaching zero.

B. Fast Convergence

We consider the LDPC-Hadamard code in Section V-A. Figure 7 illustrates the evolution of the bit-error rate as a function of the iteration number for $E_b/N_0 = -1.18, -1.15, -1.00$, and 0 dB. We can see that it takes 217 iterations for LDPC-Hadamard decoding to converge for $E_b/N_0 = -1.18$ dB, the achievable threshold of this LDPC-Hadamard code from the simulation. With E_b/N_0 slightly increased to -1.15 dB, the number of iterations to converge is reduced to 183, a 15% drop. As E_b/N_0 is increased to -1.00 dB, only 0.18 dB away from the achievable threshold, the average number of iterations to converge becomes 124. When $E_b/N_0 = 0$ dB, it takes only 38 iterations to converge for decoding this LDPC-Hadamard code with length $N \approx 1.31 \times 10^6$.

VI. CONCLUSION

In this paper, we have studied GLDPC codes with Hadamard constraints, referred to as LDPC-Hadamard codes. The perfor-

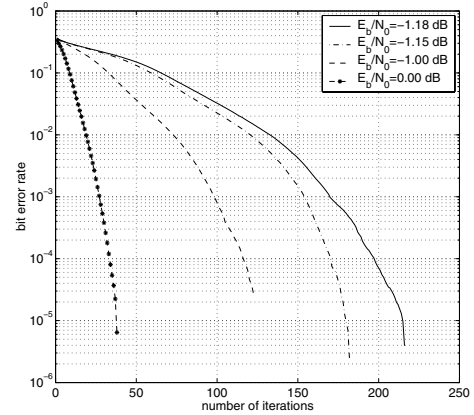


Fig. 7. Decoding convergence of LDPC-Hadamard illustrated by BER evolution as a function of the iteration number.

mance of the GLDPC codes are discussed, based on which a modified LDPC-Hadamard code graph is proposed in order to obtain performance approaching the Shannon limit. Simulation results show that optimized LDPC-Hadamard codes perform better than low-rate turbo Hadamard codes while also having a fast convergence rate. A rate-0.003 LDPC-Hadamard code with long block length has performance only 0.15 dB away from the ultimate Shannon limit and 0.24 dB better than the best low-rate turbo-Hadamard codes.

REFERENCES

- [1] R. Gallager, "Low-density parity check codes," *IRE Trans. Inform. Theory*, vol. 39, no. 1, pp. 37–45, Jan. 1962.
- [2] M. Luby, M. Mitzenmacher, A. Shokrollahi, and D. Spielman, "Analysis of low-density codes and improved designs using irregular graphs," in *Proc. ACM Symp. Theory Comp.*, Dallas, Texas, 1998, pp. 249–258.
- [3] T. Richardson, M. Shokrollahi, and R. 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] L. Lentmaier and K. Zigangirov, "Iterative decoding of generalized low-density parity-check codes," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, Cambridge, MA, Aug. 1999, p. 149.
- [5] —, "On generalized low-density parity-check codes based on hamming component codes," *IEEE Commun. Lett.*, vol. 3, no. 8, pp. 248–250, Aug. 1999.
- [6] S. ten Brink, "Convergence behavior of iterative decoded parallel concatenated codes," *IEEE Trans. Commun.*, vol. 49, no. 10, pp. 1727–1737, Oct. 2001.
- [7] L. Ping, W. Leung, and K. Wu, "Low-rate turbo-Hadamard codes," *IEEE Trans. Inform. Theory*, vol. 49, no. 12, pp. 3213–3224, Dec. 2003.
- [8] J. Boutros, O. Pothier, and G. Zemor, "Generalized low density (Tanner) codes," in *Proc. of Int. Conf. Commun. (ICC)*, Vancouver, Canada, June 1999, pp. 441–445.
- [9] A. Ashikhmin, G. Kramer, and S. ten Brink, "Extrinsic information transfer functions: Model and erasure channel properties," *IEEE Trans. Inform. Theory*, vol. 50, no. 11, pp. 2657–2673, Nov. 2004.
- [10] S. ten Brink, G. Kramer, and A. Ashikhmin, "Design of low-density parity-check codes for modulation and detection," *IEEE Trans. Commun.*, vol. 52, no. 4, pp. 670–678, April 2004.
- [11] A. Ashikhmin, G. Kramer, and S. ten Brink, "Extrinsic information transfer functions: A model and two properties," in *Proc. Conf. Info. Sci. Syst. (CISS)*, Princeton, NJ, Mar. 2002.
- [12] M. Tüchler and J. Hagenauer, "EXIT charts and irregular codes," in *Proc. Conf. Info. Sci. Syst. (CISS)*, Princeton, NJ, Mar. 2002.
- [13] G. Yue and X. Wang, "IRA code design for mimo systems with iterative receivers," in *Proc. IEEE Int. Symp. Control, Commun., & Signal Processing*, Hammamet, Tunisia, Mar. 2004.