

Protograph-based LDPC-Hadamard Codes

P. W. Zhang and F. C. M. Lau

*Department of Electronic and Information Engineering
The Hong Kong Polytechnic University
Hong Kong*

pengwei.zhang@connect.polyu.hk, encmlau@polyu.edu.hk

C.-W. Sham

*School of Computer Science
The University of Auckland
Auckland, New Zealand*

b.sham@auckland.ac.nz

Abstract—We propose a novel type of ultimate-Shannon-limit-approaching codes, namely protograph-based low-density parity-check Hadamard (PLDPC-Hadamard) codes in this paper. We also propose a systematic way of analyzing such codes using Protograph EXtrinsic Information Transfer (PEXIT) charts. Using the analytical technique we have found a code of rate about 0.05 having a theoretical threshold of -1.42 dB. At a BER of 10^{-5} , the gaps of our code to the Shannon capacity for $R = 0.05$ and to the ultimate Shannon limit are 0.25 dB and 0.40 dB, respectively.

Index Terms—PLDPC-Hadamard code, PEXIT chart, ultimate Shannon limit

I. INTRODUCTION

In 1943, Claude Shannon derived the channel capacity theorem. Based on the theorem, the maximum rate that information can be sent through a channel without errors can be evaluated. In the 50 years that followed, huge efforts were spent in designing error correction codes that would allow communication systems to work close to the channel capacity. Despite the efforts made, the gaps to the capacity remained large. It was not until 1993 when a major breakthrough took place.

In 1993, Berrou *et al.* invented the turbo codes and the turbo decoder which allowed two or more “sub-decoders” to update and exchange decoded information. They demonstrated that with a code rate of 0.5, the proposed turbo code and decoder could work within 0.7 dB from the capacity limit at a bit error rate of 10^{-5} [1], [2]. Subsequently, a lot of research effort has been spent on investigating coding schemes that are also capacity-approaching. The most well-known ones are low-density parity-check codes (proposed by Gallager in 1960s [3] and rediscovered by MacKay and Neal in 1990s [4]) and polar codes (proposed by Arikan in 2009 [5]). These capacity-approaching codes have since been used in many wireless communication systems (e.g., 3G/4G/5G, Wifi, satellite communications) and optical communication systems.

In practice, different channels possess different capacities, depending on factors such as modulation scheme, signal-to-noise ratio and code rate. However, the “ultimate Shannon limit” over an additive white Gaussian channel equals -1.59 dB, i.e., $E_b/N_0 = -1.59$ dB. It defines the bit-energy-to-noise-power-density ratio below which no digital

communications can be error-free. This ultimate Shannon limit has, in the past, been only of academic interest and only a few attempts have been made to achieve this limit. With the proliferation of Internet-of-Things (IoT) systems, billions of IoT devices will be connected to the Internet in the near future and multiple access techniques need to be advanced accordingly to accommodate all such connections. As a consequence, IoT systems may have to work closer and closer to the ultimate Shannon limit.

The most notable channel codes with performance close to the limit are turbo-Hadamard codes [6]–[9], concatenated zigzag Hadamard codes [10], and low-density parity-check (LDPC) Hadamard codes [11], [12]. However, both turbo-Hadamard codes and concatenated zigzag Hadamard codes suffer from long decoding latency due to the serial processing of the Bahl-Cocke-Jelinek-Raviv (BCJR) decoder [6], [10]. The LDPC-Hadamard codes allow parallel processing and hence the decoding latency can be made shorter [12]. However, in optimizing the threshold of LDPC-Hadamard codes, only the degree distribution of the variable nodes is found for a given order of the Hadamard code used. Therefore, the optimizing method used in [12] suffers from the following drawbacks.

- 1) For the same variable-node degree distribution, many different code realizations with very diverse bit-error-rate performances can be obtained. As a consequence, the degree distribution should be further refined to provide more information for code construction.
- 2) The degree distribution analysis in [12] requires a minimum variable-node degree of 2 because the EXtrinsic Information Transfer (EXIT) curve cannot be produced for degree-1 variable nodes. Moreover, LDPC-Hadamard codes with punctured variable nodes cannot be analyzed. In summary, LDPC-Hadamard code with degree-1 and/or punctured variable nodes cannot be analyzed by the method in [12]. Yet in many application scenarios, it has been shown that LDPC codes with degree-1 and/or punctured variable nodes can result better theoretical thresholds and hence error performances. New techniques should therefore be proposed to analyze ultimate-Shannon-limit-approaching codes with degree-1 and/or punctured variable nodes.

The concept in [12] has been applied in designing other low-

The work described in this paper was supported by a grant from the RGC of the Hong Kong SAR, China (Project No. PolyU 152170/18E).

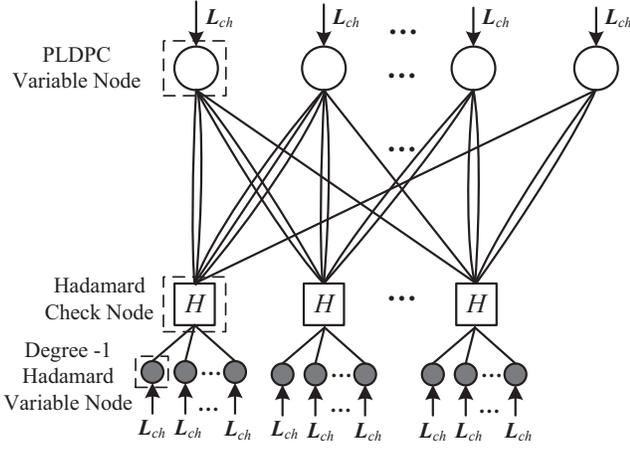


Fig. 1. The protograph of a PLDPC-Hadamard code. Each blank circle denotes a protograph variable node (P-VN), each square with an “H” inside denotes a Hadamard check node (H-CN), and each filled circle denotes a degree-1 Hadamard variable node (D1H-VN). n P-VNs and m H-CNs are assumed in the protograph.

rate generalized LDPC codes [13]. However, the main criterion of those codes is to provide low latency communications and hence their performance is relatively far from the ultimate Shannon limit.

In this paper, we propose a new type of ultimate-Shannon-limit-approaching codes, namely protograph-based low-density parity-check Hadamard (PLDPC-Hadamard) codes. We also propose a systematic way of analyzing such codes. Using the analytical technique we have found a code of rate about 0.05 having a theoretical threshold of -1.42 dB. At a BER of 10^{-5} , the gaps of our code to the Shannon capacity for $R = 0.05$ and to the ultimate Shannon limit are 0.25 dB and 0.40 dB, respectively.

II. PROTOGRAPH-BASED LDPC HADAMARD CODE

We propose a new type of ultimate-Shannon-limit-approaching channel code called *protograph-based LDPC Hadamard (PLDPC-Hadamard) code*, the base structure of which is shown in Fig. 1. Referring to the figure, each blank circle denotes a protograph variable node (P-VN), each square with an “H” inside denotes a Hadamard check node (H-CN), and each filled circle denotes a degree-1 Hadamard variable node (D1H-VN). We assume that there are n P-VNs and m H-CNs. The base matrix of the proposed PLDPC-Hadamard code is then denoted by $\mathbf{B}_{m \times n} = \{b_{i,j}\}$, where $b_{i,j}$ represents the number of edges connecting the i -th H-CN ($i = 0, 1, \dots, m-1$) and the j -th P-VN ($j = 0, 1, \dots, n-1$). Moreover, we denote the weight of the i -th row by $d_{c_i} = \sum_{j=0}^{n-1} b_{i,j}$, which represents the total number of edges connecting the i -th H-CN to all P-VNs. For example in Fig. 1, the number of edges connecting each of the three displayed H-CNs to all P-VNs is equal to 6. These d_{c_i} edges are considered as (input) information bits to the i -th Hadamard code while the connected D1H-VNs represent the corresponding (output) parity bits in the Hadamard code. Below we briefly introduce

the Hadamard code.

A Hadamard code with an order r is a class of linear block codes. We consider a $q \times q$ Hadamard matrix $\mathbf{H}_q = \{\mathbf{h}_j, j = 0, 1, \dots, q-1\}$, which can be constructed recursively using

$$\mathbf{H}_q = \begin{bmatrix} +\mathbf{H}_{q/2} & +\mathbf{H}_{q/2} \\ +\mathbf{H}_{q/2} & -\mathbf{H}_{q/2} \end{bmatrix}, \quad (1)$$

with $q = 2^r$ and $\mathbf{H}_1 = [+1]$. Each row $\pm \mathbf{h}_j$ is a Hadamard codeword and thus $\pm \mathbf{H}_q$ contains $2q$ codewords. We denote a Hadamard codeword by $\mathbf{c}^H = [c_0^H \ c_1^H \ \dots \ c_{2^r-1}^H]$. Assuming r is an even number, it has been shown that [12] $c_0^H \oplus c_1^H \oplus c_2^H \oplus c_4^H \oplus \dots \oplus c_{2^{r-1}}^H \oplus c_{2^r-1}^H = 0$. Viewing from another perspective, if there is a length- $(r+2)$ single-parity-check (SPC) codeword denoted by $\mathbf{c}_\mu = [c_{\mu_0} \ c_{\mu_1} \ \dots \ c_{\mu_r} \ c_{\mu_{r+1}}]$, these bits can be used as inputs to a systematic Hadamard encoder and form a Hadamard codeword $\mathbf{c}^H = [c_0^H \ c_1^H \ \dots \ c_{2^r-1}^H]$ where $c_0^H = c_{\mu_0}$; $c_1^H = c_{\mu_1}$; \dots ; $c_{2^{r-1}}^H = c_{\mu_r}$ and $c_{2^r-1}^H = c_{\mu_{r+1}}$. Fig. 2 shows an example in which a $(6, 5)$ SPC codeword is encoded into a length-16 ($r = 4$) Hadamard codeword. In this case, $c_0^H = c_{\mu_0}$, $c_1^H = c_{\mu_1}$, $c_2^H = c_{\mu_2}$, $c_4^H = c_{\mu_3}$, $c_8^H = c_{\mu_4}$, $c_{15}^H = c_{\mu_5}$ and the other 10 ($= 2^r - (r+2)$) code bits are Hadamard parity bits. We call this systematic Hadamard encoding when the information bits can be exactly located in the codeword.

Referring to Fig. 1, the links connecting the P-VNs to the i -th H-CN always form a SPC. These links can make use of the above mechanism to derive the parity bits of the Hadamard code (denoted as D1H-VNs of the Hadamard check node in Fig. 1) if d_{c_i} is even. In this case, the Hadamard code length equals $2^{d_{c_i}-2}$, and the number of D1H-VNs equals $2^{d_{c_i}-2} - d_{c_i}$. Assuming d_{c_i} is even for all $i = 0, 1, \dots, m-1$, the total number of D1H-VNs is given by

$$\sum_{i=0}^{m-1} (2^{d_{c_i}-2} - d_{c_i}). \quad (2)$$

When all variable nodes are sent to the channel, the code rate of protograph given in Fig. 1 equals

$$R = \frac{n - m}{\sum_{i=0}^{m-1} (2^{d_{c_i}-2} - d_{c_i}) + n}. \quad (3)$$

If we further assume that all rows in $\mathbf{B}_{m \times n}$ have the same weight which is equal to d , i.e., $d_{c_i} = d$ for all i , the code rate is simplified to

$$R_{d_{c_i}=d} = \frac{n - m}{m(2^{d-2} - d) + n}. \quad (4)$$

If $n_p (< n)$ P-VNs are punctured, the code rate further becomes

$$R_{\text{punctured}} = \frac{n - m}{m(2^{d-2} - d) + n - n_p}. \quad (5)$$

Once a protograph is formed, we can apply the “copy-and-permute” process [14] to form a PLDPC-Hadamard code.

We assume that binary phase-shift-keying (BPSK) modulation is used and the channel is an additive white Gaussian noise

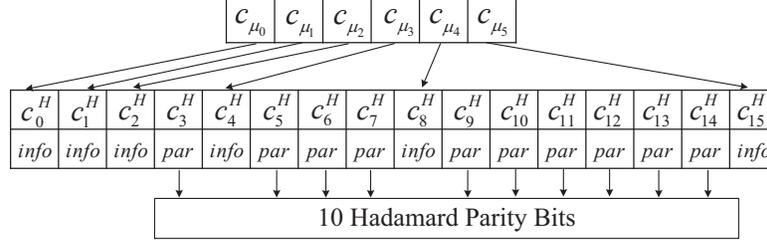


Fig. 2. Example of encoding a length-6 SPC codeword into a length-16 ($r = 4$) Hadamard codeword.

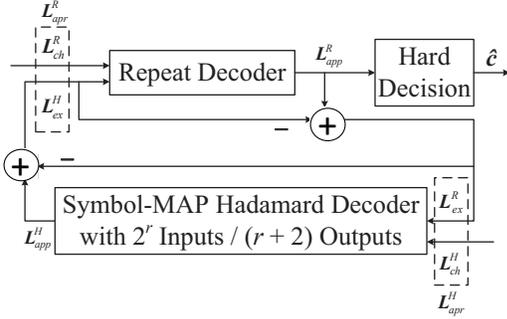


Fig. 3. Block diagram of a PLDPC-Hadamard decoder. The repeat decoder is the same as the variable-node processor used in LDPC decoder.

(AWGN) channel with mean 0 and variance σ_{ch}^2 . The iterative decoder shown in Fig. 3 is used to perform the decoding process. It consists of a repeat decoder and a symbol-by-symbol maximum *a posteriori probability* (symbol-MAP) Hadamard decoder. The repeat decoder is the same as the variable-node processor used in LDPC decoder and is therefore not described here. The symbol-MAP Hadamard decoder of order- r has a total of 2^r inputs, among which $r + 2$ come from the repeat decoder and are updated in each iteration; and the remaining $2^r - (r + 2)$ inputs come from the channel log-likelihood-ratio (LLR) information which do not change during the iterative process. Moreover, the symbol-MAP Hadamard decoder will produce $r + 2$ extrinsic LLR outputs which are fed back to the repeat decoder.

III. PROPOSED PROTOGRAPH EXTRINSIC INFORMATION TRANSFER (PEXIT) ALGORITHM

We assume that r is an even number. A H-CN has $r + 2$ links to P-VNs and is connected to $2^r - (r + 2)$ D1H-VNs. The bits corresponding to these VNs form a systematic Hadamard codeword. The LLR values coming from these VNs therefore form the 2^r inputs to the symbol-MAP Hadamard decoder. Specifically, we denote

- $\mathbf{L}_{apr}^H = [L_{apr}^H(0)L_{apr}^H(1) \cdots L_{apr}^H(2^r - 1)]$ as the *a priori* LLR inputs to the Hadamard decoder,
- $\mathbf{L}_{ex}^R = [L_{ex}^R(0)L_{ex}^R(1) \cdots L_{ex}^R(r + 1)]$ as the extrinsic LLR information output from the repeat decoder (P-VNs), and
- $\mathbf{L}_{ch}^H = [L_{ch}^H(0)L_{ch}^H(1) \cdots L_{ch}^H(2^r - (r + 2) - 1)]$ as the channel LLR observations from the D1H-VNs.

Since the Hadamard codeword is encoded using systematic encoding, we assign $L_{apr}^H(0) = L_{ex}^R(0)$, $L_{apr}^H(2^i) = L_{ex}^R(i + 1)$ for $i = 0, 1, \dots, r - 1$, $L_{apr}^H(2^r - 1) = L_{ex}^R(r + 1)$; and the remaining inputs $L_{apr}^H(j)$ are assigned the entries in \mathbf{L}_{ch}^H . The symbol-MAP Hadamard decoder then computes the *a posteriori* LLR (L_{app}^H) of the code bits using

$$L_{app}^H(i) = \log \frac{\sum_{H[i,j]=\pm 1} \gamma(\pm \mathbf{h}_j)}{\sum_{H[i,j]=\mp 1} \gamma(\pm \mathbf{h}_j)} \quad i = 0, 1, \dots, 2^r - 1, \quad (6)$$

where $\gamma(\pm \mathbf{h}_j) = \exp\left(\frac{1}{2} \langle \pm \mathbf{h}_j, \mathbf{L}_{apr}^H \rangle\right)$, $j = 0, 1, \dots, 2^r - 1$; $\langle \cdot \rangle$ denotes the inner product; and $H[i, j]$ denotes the (i, j) -th element of the Hadamard matrix \mathbf{H} . Based on the butterfly-like structure of the Hadamard matrix, \mathbf{L}_{app}^H can be computed using the fast Hadamard transform (FHT) and the dual FHT (DFHT) [6], [8], [9]. By subtracting the *a priori* LLRs from the *a posteriori* LLRs, the extrinsic LLRs (\mathbf{L}_{ex}^H) can be obtained. The $r + 2$ extrinsic LLRs corresponding to the information bits (i.e., connected P-VNs) will be fed back to the repeat decoder as the updated input *a priori* LLRs while those corresponding to D1H-VNs will be ignored. Fig. 4 illustrates the computation of \mathbf{L}_{app}^H and \mathbf{L}_{ex}^H when $r = 4$, corresponding to $r + 2 = 6$ information bits and $2^r - (r + 4) = 10$ Hadamard parity bits. The iterative process between the repeat decoder and symbol-MAP Hadamard decoder continues until the information bits corresponding to all Hadamard codes (after hard decision) become valid SPCs or the maximum number of iterations has been reached.

We define the following symbols.

- $I_{av}(i, j)$: a priori mutual information (MI) from the i -th H-CN to the j -th P-VN
- $I_{ev}(i, j)$: extrinsic MI from the j -th P-VN to the i -th H-CN
- $I_{ah}(i, k)$: a priori MI of the k -th information bit in the i -th H-CN
- $I_{eh}(i, k)$: extrinsic MI of the k -th information bit in the i -th H-CN
- $I_{app}(j)$ a posteriori MI of the j -th P-VN

We propose a low-complexity PEXIT algorithm for analyzing PLDPC-Hadamard designs. This technique can analyze base-matrices containing degree-1 and/or punctured VNs in the protograph. It will also produce different analytical results for base matrices with the same weight distributions but different structures.

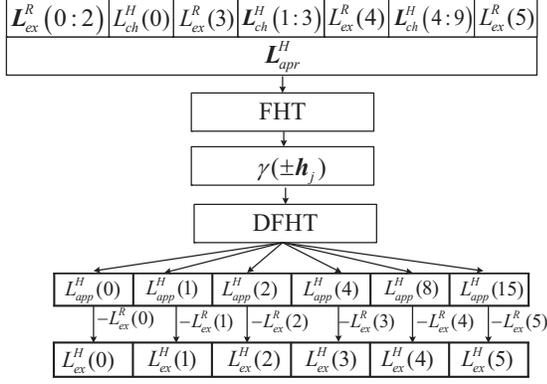


Fig. 4. The symbol-MAP Hadamard decoder with $r = 4$, i.e., 16 LLR inputs and 6 output LLRs for the information bits.

Referring to Fig. 1, the channel LLR value L_{ch} follows a normal distribution $\mathcal{N}(\sigma_{ch}^2/2, \sigma_{ch}^2)$ where $\sigma_{ch}^2 = 8R \cdot E_b/N_0$, R is the code rate, and E_b/N_0 denotes the bit-energy-to-noise-power-spectral-density ratio. Also, the MI between a code bit and L_{ch} is computed by the $J(\sigma_{ch})$ function [14], [15], i.e.,

$$J(\sigma_{ch}) = 1 - \int_{-\infty}^{\infty} \frac{e^{-(\xi - \sigma_{ch}^2/2)^2 / 2\sigma_{ch}^2}}{\sqrt{2\pi\sigma_{ch}^2}} \log_2 [1 - e^{-\xi}] d\xi. \quad (7)$$

When the output MI of a decoder is I , the corresponding LLR values of the extrinsic information obeys a Gaussian distribution of $(\pm\sigma^2/2, \sigma^2)$, where $\sigma = J^{-1}(I)$ is computed using

$$J^{-1}(I) = \begin{cases} a_1 I^2 + b_1 I + c_1 \sqrt{I}, & 0 \leq I \leq 0.3646 \\ -a_2 \ln[b_2(1 - I)] - c_2 I, & 0.3646 < I < 1 \end{cases}$$

with $a_1, a_2, b_1, b_2, c_1, c_2$ being constants [14], [15].

A. PEXIT Algorithm

To generate the PEXIT curves for the repeat decoder and symbol-MAP Hadamard decoder, we apply the following steps for a given σ_{ch}^2 and a given base matrix $\mathbf{B}_{m \times n}$.

- 1) For $i = 0, 1, \dots, m-1$ and $j = 0, 1, \dots, n-1$, set $I_{ch}(j) = J(\sigma_{ch})$ and $I_{av}(i, j) = 0$. Set $I_{ch}^H = J(\sigma_{ch})$.
- 2) For $i = 0, 1, \dots, m-1$ and $j = 0, 1, \dots, n-1$, compute (8). if $b_{i,j} > 0$; else set $I_{ev}(i, j) = 0$. Taking the following 3×4 base matrix as an example,

$$\begin{bmatrix} 2 & 0 & 2 & 2 \\ 0 & 2 & 2 & 2 \\ 3 & 2 & 0 & 1 \end{bmatrix},$$

the degree of each row is $d = 6$ and hence $r + 2 = 6 \Rightarrow r = 4$. After analyzing the MI of the P-VNs, the corresponding 3×4 $\{I_{ev}(i, j)\}$ MI matrix can be represented by (9).

- 3) Convert the $m \times n$ $\{I_{ev}(i, j)\}$ MI matrix into an $m \times d$ $\{I_{ah}(i, k)\}$ MI matrix by (i) eliminating the 0 entries and (ii) repeating $\{I_{ev}(i, j)\}$ $b_{i,j} (\geq 1)$ times in the same row. Using the previous example, the 3×4 $\{I_{ev}(i, j)\}$ MI matrix is converted into the 3×6 $\{I_{ah}(i, k)\}$ MI matrix shown in (11).

- 4) For $i = 0, 1, \dots, m-1$, using the d entries in the i -th row of I_{ah} and σ_{ch}^2 , generate a large number of sets of LLR values as inputs to the symbol-MAP Hadamard decoder and record the output extrinsic LLR values of the k -th information bit ($k = 0, 1, \dots, d-1$). Compute the extrinsic MI of the information bit using (10), where $p_e(\xi|X=x)$ is the probability density function (PDF) of the LLR values for the corresponding bit being 0 or 1. Form the extrinsic MI matrix $\{I_{eh}(i, k)\}$ of size $m \times d$. Using the previous example, the matrix is represented by (12).

Remark: Our Monte Carlo method in obtaining the extrinsic MI values of the symbol-MAP Hadamard decoder not only has a low complexity, but also is generic and applicable to analyzing both systematic and non-systematic Hadamard codes. Our technique makes use of multiple a priori MI values ($\{I_{ah}(i, k)\}$) and produces multiple extrinsic MI values ($\{I_{eh}(i, k)\}$). (MI from channel messages can be regarded as a constant.) In [16], an EXIT function of symbol-MAP Hadamard decoder under the AWGN channel is obtained. However, the function involves very high computational complexity, which increases rapidly with an increase of the Hadamard order r . The function also cannot be used for analyzing non-systematic Hadamard codes. In [12], simulation is used to characterize the symbol-MAP Hadamard decoder but the method is based on a single a priori MI value and produces only one output extrinsic MI.

- 5) Convert the $m \times d$ $\{I_{eh}(i, k)\}$ MI matrix into an $m \times n$ $\{I_{av}(i, j)\}$ MI matrix. For $i = 0, 1, \dots, m-1$ and $j = 0, 1, \dots, n-1$; if $b_{i,j} > 0$, set the value of $I_{av}(i, j)$ as the average of the corresponding $b_{i,j}$ MI values in the i th-row of $\{I_{eh}(i, k)\}$; else set $I_{av}(i, j) = 0$. In the above example, $\{I_{av}(i, j)\}$ becomes (13).
- 6) Repeat Steps 2 to 5 until the maximum number of iterations is reached; or when $I_{app}(j) = 1$ for all $j = 0, 1, \dots, n-1$ where $I_{app}(j) = J\left(\sqrt{\sum_{i=0}^{m-1} b_{i,j} (J^{-1}(I_{av}(i, j)))^2 + \sigma_{ch}^2}\right)$.

Note that our PEXIT algorithm can be used to analyze PLDPC-Hadamard designs with degree-1 and/or punctured VNs. In case of puncturing, the corresponding channel LLR values in the analysis will be set to zero.

B. Optimization Criterion

For a given code rate, our objective is to find a proto-graph of the PLDPC-Hadamard code such that it achieves $I_{app}(j) = 1 \forall j$ within a fixed number of iterations and with the largest σ^2 (i.e., lowest E_b/N_0 or threshold). To reduce the search space, we impose the following constraints. First, the weights of all rows in the base matrix are fixed at d . Second, the maximum and minimum column weights are 9 and 1, respectively. Third, the maximum value of each entry in base matrix (i.e., maximum number of connections between

$$I_{ev}(i, j) = J \left(\sqrt{\sum_{s \neq i} b_{s,j} (J^{-1}(I_{av}(s, j)))^2 + (b_{i,j} - 1) \cdot (J^{-1}(I_{av}(i, j)))^2 + \sigma_{ch}^2} \right) \quad (8)$$

$$I_{ev} = \begin{bmatrix} I_{ev}(0,0) & 0 & I_{ev}(0,2) & I_{ev}(0,3) \\ 0 & I_{ev}(1,1) & I_{ev}(1,2) & I_{ev}(1,3) \\ I_{ev}(2,0) & I_{ev}(2,1) & 0 & I_{ev}(2,3) \end{bmatrix} \quad (9)$$

$$I_E = \frac{1}{2} \sum_{x \in \{\pm 1\}} \int_{-\infty}^{\infty} p_e(\xi|X=x) \log_2 \frac{2 \cdot p_e(\xi|X=x)}{p_e(\xi|X=-1) + p_e(\xi|X=1)} d\xi \quad (10)$$

$$I_{ah} = \begin{bmatrix} I_{ah}(0,0) & I_{ah}(0,1) & I_{ah}(0,2) & I_{ah}(0,3) & I_{ah}(0,4) & I_{ah}(0,5) \\ I_{ah}(1,0) & I_{ah}(1,1) & I_{ah}(1,2) & I_{ah}(1,3) & I_{ah}(1,4) & I_{ah}(1,5) \\ I_{ah}(2,0) & I_{ah}(2,1) & I_{ah}(2,2) & I_{ah}(2,3) & I_{ah}(2,4) & I_{ah}(2,5) \\ I_{ev}(0,0) & I_{ev}(0,0) & I_{ev}(0,2) & I_{ev}(0,2) & I_{ev}(0,3) & I_{ev}(0,3) \\ I_{ev}(1,1) & I_{ev}(1,1) & I_{ev}(1,2) & I_{ev}(1,2) & I_{ev}(1,3) & I_{ev}(1,3) \\ I_{ev}(2,0) & I_{ev}(2,0) & I_{ev}(2,0) & I_{ev}(2,1) & I_{ev}(2,1) & I_{ev}(2,3) \end{bmatrix} \quad (11)$$

$$I_{eh} = \begin{bmatrix} I_{eh}(0,0) & I_{eh}(0,1) & I_{eh}(0,2) & I_{eh}(0,3) & I_{eh}(0,4) & I_{eh}(0,5) \\ I_{eh}(1,0) & I_{eh}(1,1) & I_{eh}(1,2) & I_{eh}(1,3) & I_{eh}(1,4) & I_{eh}(1,5) \\ I_{eh}(2,0) & I_{eh}(2,1) & I_{eh}(2,2) & I_{eh}(2,3) & I_{eh}(2,4) & I_{eh}(2,5) \end{bmatrix} \quad (12)$$

$$I_{av} = \begin{bmatrix} I_{av}(0,0) & 0 & I_{av}(0,2) & I_{av}(0,3) \\ 0 & I_{av}(1,1) & I_{av}(1,2) & I_{av}(1,3) \\ I_{av}(2,0) & I_{av}(2,1) & 0 & I_{av}(2,3) \end{bmatrix} \\ = \begin{bmatrix} \frac{1}{2} \sum_{k=0}^1 I_{eh}(0, k) & 0 & \frac{1}{2} \sum_{k=2}^3 I_{eh}(0, k) & \frac{1}{2} \sum_{k=4}^5 I_{eh}(0, k) \\ 0 & \frac{1}{2} \sum_{k=0}^1 I_{eh}(1, k) & \frac{1}{2} \sum_{k=2}^3 I_{eh}(1, k) & \frac{1}{2} \sum_{k=4}^5 I_{eh}(1, k) \\ \frac{1}{3} \sum_{k=0}^2 I_{eh}(2, k) & \frac{1}{2} \sum_{k=3}^4 I_{eh}(2, k) & 0 & I_{eh}(2, 5) \end{bmatrix} \quad (13)$$

a P-VN and a H-CN) is 3. Fourth, the maximum number of iterations used in the PEXIT algorithm is set to 300. Fifth, we set a target threshold below -1.40 dB¹. Thus, we start our analysis with a E_b/N_0 of -1.40 dB. If a base matrix is found satisfying $I_{app}(j) = 1 \forall j$, we reduce E_b/N_0 by 0.01 dB and attempt to search a suitable base matrix again. The process continues until a suitable base matrix cannot be found.

IV. RESULTS

We have attempted to find a PLDPC-Hadamard code with a code rate of approximately 0.05. We also assume $d = 6$ and hence $r = d - 2 = 4$. We then substitute the above values into (4) and obtain $\frac{m}{n} \approx 0.63$. We select a base matrix $\mathbf{B}_{7 \times 11}$ of size 7×11 , i.e., $m = 7$ and $n = 11$, and hence the code rate equals $R = 0.0494$. We have used our proposed PEXIT algorithm in analyzing some base matrices generated according to the above constraints and our results are as follows.

We have found the following base matrix having a theoret-

ical threshold of -1.42 dB.

$$\mathbf{B}_{7 \times 11} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 3 & 0 & 1 \\ 0 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 \\ 2 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 3 & 0 \\ 3 & 0 & 0 & 2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 2 & 0 \end{bmatrix}$$

Fig. 5 plots the PEXIT curves of the repeat decoder and the symbol-MAP Hadamard decoder under $E_b/N_0 = -1.42$ dB. It can be observed that the two curves are matched. By lifting the base matrix with factors of $z_1 = 32$ and $z_2 = 512$ using the progressive edge-growth (PEG) method [17], we obtain a PLDPC-Hadamard code with information length $k = z_1 z_2 (n - m) = 65, 536$ and code length $N_{total} = z_1 z_2 [m(2^{d-2} - d) + n] = 1, 327, 104$.

We simulate the PLDPC-Hadamard code. At a particular E_b/N_0 , the simulation continues until 100 frame errors have been reached. Then we record corresponding the bit error rate (BER) and frame error rate (FER). The BER and FER results of the PLDPC-Hadamard code found are shown in Fig. 6. Our code achieves a BER of 10^{-5} at $E_b/N_0 = -1.19$ dB, which is 0.23 dB from the threshold. At a BER of 10^{-5} , the gaps of our code to the Shannon capacity for $R = 0.05$ and to the ultimate Shannon limit are 0.25 dB and 0.40 dB,

¹We only consider codes with code rates that have capacities below -1.40 dB.

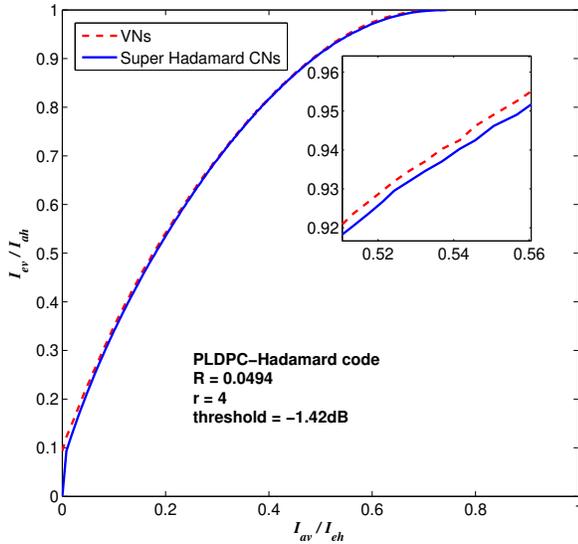


Fig. 5. The PEXIT chart of the PLDPC-Hadamard code with $R = 0.0494$ and $r = 4$.

respectively. Compared with the LDPC-Hadamard code in [12] which uses $R = 0.05$ and $r = 4$, our proposed PLDPC-Hadamard code has a slight performance improvement. The relatively advantage of our proposed PLDPC-Hadamard code over the LDPC-Hadamard code is probably due to degree-1 VNs in the protograph. Such degree-1 VNs are regarded as a kind of precoding structure. They can not only increase the linear minimum distance but also reduce the threshold [14], [18].

V. CONCLUSION

In this paper, we have proposed the protograph-based LDPC-Hadamard code which is ultimate-Shannon-limit approaching. We have also presented a PEXIT algorithm to analyze the threshold of the code. We have shown that the protograph-based LDPC-Hadamard code found using the analytical technique can outperform existing ultimate-Shannon-limit LDPC-Hadamard code. In future, we plan to conduct a further investigation on PLDPC-Hadamard codes with one or a combination of the following characteristics: (i) r being odd, (ii) r being larger, (iii) protograph with punctured VNs, (iv) enlarged searching space for the base matrix. We will also design hardware architectures for a complete PLDPC-Hadamard encoder/decoder system, and implement it onto an FPGA.

REFERENCES

- [1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding," in *Proc. of IEEE ICC*, vol. 2, pp. 1064–1070, 1993.
- [2] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: turbo-codes," *IEEE Trans. Commun.*, vol. 44, pp. 1261–1271, Oct. 1996.
- [3] R. G. Gallager, *Low-Density Parity-Check Codes*. PhD thesis, Cambridge, USA, 1963.

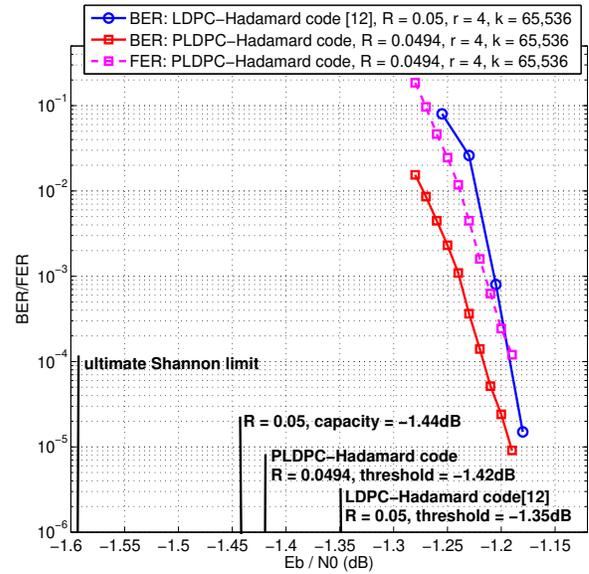


Fig. 6. BER and FER performance of our proposed PLDPC-Hadamard code compared with the BER of the LDPC-Hadamard code in [12], $r = 4$.

- [4] D. J. MacKay and R. M. Neal, "Good Codes Based on Very Sparse Matrices," in *Proc. Cryptography Coding. 5th IMA Conf., Number 1025 Lecture Notes Comput. Sci.*, pp. 100–111, 1995.
- [5] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. Inf. Theory*, vol. 55, pp. 3051–3073, July 2009.
- [6] P. Li, W. K. Leung, and K. Y. Wu, "Low-Rate Turbo-Hadamard Codes," *IEEE Trans. Inf. Theory*, vol. 49, no. 12, pp. 3213–3224, Dec. 2003.
- [7] H. H. Xu and Z. S. Bie, "CRC-aided Iterative Decoding of Turbo-Hadamard Codes," in *Proc. of IEEE ICC*, 2018.
- [8] S. Jiang, P. W. Zhang, F. C. M. Lau, and C.-W. Sham, "An Ultimate-Shannon-Limit-Approaching Gbps Throughput Encoder/Decoder System," *IEEE Trans. Circ. Sys. II*, to appear, DOI 10.1109/TC-SII.2019.2960613.
- [9] S. Jiang, P. W. Zhang, F. C. M. Lau, C.-W. Sham, and K. Huang, "A Turbo-Hadamard Encoder/Decoder System with Hundreds of Mbps Throughput," in *Proc. of ISTC*, 2018.
- [10] W. K. R. Leung, G. Yue, L. Ping, and X. Wang, "Concatenated zigzag Hadamard codes," *IEEE Trans. Inf. Theory*, vol. 52, pp. 1711–1723, Apr. 2006.
- [11] Guosen Yue, Ping Li, and Xiaodong Wang, "Low-rate generalized low-density parity-check codes with Hadamard constraints," in *Proc. of ISIT*, pp. 1377–1381, 2005.
- [12] G. Yue, L. Ping, and X. Wang, "Generalized Low-Density Parity-Check Codes Based on Hadamard Constraints," *IEEE Trans. on Information Theory*, vol. 53, pp. 1058–1079, Mar. 2007.
- [13] Y. Liu, P. M. Olmos and D. G. M. Mitchell, "On Generalized LDPC Codes for 5G Ultra Reliable Communication," in *Prof. of ITW*, 2018.
- [14] Y. Fang, G. A. Bi, Y. L. Guan, and F. C. M. Lau, "A Survey on Protograph LDPC Codes and Their Applications," *IEEE Commun. Surv. Tut.*, vol. 17, no. 4, pp. 1989–2016, 2015.
- [15] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Trans. Commun.*, vol. 49, pp. 1727–1737, Oct 2001.
- [16] K. Li, X. D. Wang, and A. Ashikhmin, "EXIT Functions of Hadamard Components in Repeat-Zigzag-Hadamard (RZH) Codes with Parallel Decoding," *IEEE Trans. Inf. Theory*, vol. 54, no. 4, pp. 1773–1785, Apr. 2008.
- [17] X. Y. Hu, E. Eleftheriou, and D. M. Arnold, "Regular and Irregular Progressive Edge-Growth Tanner Graphs," *IEEE Trans. Inf. Theory*, vol. 51, no. 1, pp. 386–398, Jan. 2005.
- [18] A. Abbasfar, D. Divsalar, and K. Yao, "Accumulate-Repeat-Accumulate Codes," *IEEE Trans. Commun.*, vol. 55, no. 4, pp. 692–702, Apr. 2007.