Capacity-Approaching Bandwidth-Efficient Coded Modulation Schemes Based on Low-Density Parity-Check Codes

Jilei Hou, *Student Member, IEEE*, Paul H. Siegel, *Fellow, IEEE*, Laurence B. Milstein, *Fellow, IEEE*, and Henry D. Pfister, *Student Member, IEEE*

Abstract—We design multilevel coding (MLC) and bit-interleaved coded modulation (BICM) schemes based on low-density parity-check (LDPC) codes. The analysis and optimization of the LDPC component codes for the MLC and BICM schemes are complicated because, in general, the equivalent binary-input component channels are not necessarily symmetric. To overcome this obstacle, we deploy two different approaches: one based on independent and identically distributed (i.i.d.) channel adapters and the other based on coset codes. By incorporating i.i.d. channel adapters, we can force the symmetry of each binary-input component channel. By considering coset codes, we extend the concentration theorem based on previous work by Richardson *et al.* and Kavčić *et al.* We also discuss the relation between the systems based on the two approaches and show that they indeed have the same expected decoder behavior. Next, we jointly optimize the code rates and degree distribution pairs of the LDPC component codes for the MLC scheme. The optimized irregular LDPC codes at each level of MLC with multistage decoding (MSD) are able to perform well at signal-to-noise ratios (SNR) very close to the capacity of the additive white Gaussian noise (AWGN) channel. We also show that the optimized BICM scheme can approach the parallel independent decoding (PID) capacity as closely as does the MLC/PID scheme. Simulations with very large codeword length verify the accuracy of the analytical results. Finally, we compare the simulated performance of these coded modulation schemes at finite codeword lengths, and consider the results from the perspective of a random coding exponent analysis.

Index Terms—Bit-interleaved coded modulation (BICM), coding exponent analysis, coset codes, density evolution, independent and identically distributed (i.i.d.) channel adapters, irregular low-density parity-check (LDPC) codes, LDPC codes, multilevel coding (MLC).

I. INTRODUCTION

MULTILEVEL coding (MLC) [3], [4] and bit-interleaved coded modulation (BICM) [5], [6] are two well-known coded modulation schemes proposed to achieve both power and bandwidth efficiency. However, past research has primarily focused on the maximization of minimum Euclidean distance and asymptotic gains [3], [7]. Recently, the application of methods from information theory has helped to overcome the shortcomings of this traditional coding philosophy. It is proved in [4] that MLC together with multistage decoding (MSD) suffices to approach the channel capacity if the component code rates are properly chosen. Reference [4] also concludes that if we use Gray mapping and employ parallel independent decoding (PID) at each level separately, the information loss relative to the channel capacity is negligible if optimal component codes are used. Furthermore, it is recognized that Gray-mapped BICM provides mutual information very close to the channel capacity [6] and is actually a derivative of the MLC/PID scheme using a single binary code [4], [8]. Since the invention and refinement of turbo codes [9], the research community also realized the change in the paradigm of coding optimality, i.e., not to pay attention to only minimum distances. These discoveries allow us to draw one important conclusion: Using powerful component codes with properly designed rates for MLC or BICM enables us to get very close to channel capacity at a desired bandwidth efficiency.

On the other hand, low-density parity-check (LDPC) codes [10] have been shown to achieve low bit-error rates (BERs) at signal-to-noise ratios (SNR) very close to the Shannon limits on many interesting binary-input channels [11]–[13], and they outperform turbo codes when the block length of the code is large, even though the decoding complexity is less than that of turbo codes. Therefore, LDPC codes are considered to be among the most power-efficient binary codes for digital transmission.

In this paper, we explore the use of LDPC codes [10], [11] as the component codes of both MLC and BICM schemes designed to approach the channel capacity (also cf. [14], [15]). The BICM scheme we study refers to the scheme which does not iterate between the demodulator and the decoder.[1] It is shown in [4], [8] that the concept of an equivalent binary-input component channel for each individual bit level is an effective tool for the analysis and design of these coded modulation schemes. Using this observation, we transform the design of LDPC codes for these coded modulation schemes into the design of LDPC codes for the equivalent binary-input component channels. Density evolution [1] has been proven to be a powerful tool for the anal-

[1]Some authors have studied iterative demodulation and decoding for the BICM and MLC schemes, e.g., [16]–[18].

ysis and design of LDPC codes for various binary-input symmetric channels. However, the equivalent binary-input component channels of these coded modulation schemes are not necessarily symmetric. To address this problem, we use the idea of an independent and identically distributed (i.i.d.) channel adapter, introduced in [15]. We show that the i.i.d. channel adapter and the equivalent binary-input channel can be considered together as a new "augmented channel" which is output symmetric, satisfying the symmetry condition in [1]. Therefore, the analysis and design of LDPC codes is greatly simplified. In [2], by considering LDPC coset codes instead of linear LDPC codes, concentration theorems are proven for LDPC coset code ensembles on channels with binary inputs and intersymbol interference (ISI) due to channel memory. In this paper, we apply the concept of LDPC coset codes to both MLC and BICM cases and provide a similar concentration theorem over almost all graphs, almost all input sequences (time-multiplex of coset code codewords), and almost all channel noise realizations. We also discuss the relation between the systems based on i.i.d. channel adapters and coset codes and show that these two systems have the same expected decoder behavior.

The outline of this paper is as follows. In Section II, we introduce the system model of MLC and BICM, including their encoder structures, decoding strategies, and related capacity results. In Section III, we first discuss the concept of an i.i.d. channel adapter and prove the corresponding properties. Next, we introduce the coset code scheme and present the coset code concentration theorem. In Section IV, we extend density evolution to evaluate the asymptotic performance of the LDPC component codes for the MLC and BICM schemes incorporating the i.i.d. channel adapter. We describe the optimization technique for both MLC (joint optimization of component code rates and code parameters) and BICM (only component code parameters) in this section, as well. In Section V, we present the optimization results for both Gray-mapped MLC and BICM schemes based on 4-PAM and 8-PSK modulation.[2] We show that the optimized thresholds are very close to their associated capacities and we verify the validity of the code designs by very large block-size simulation results. Finally, we simulate the performance of these MLC and BICM schemes based on optimized LDPC codes at moderate block sizes and consider the results from the perspective of a random coding exponent analysis. Section VI concludes the paper.

II. SYSTEM MODEL

A. Multilevel Coding (MLC)

The encoder structure of the LDPC coded MLC scheme is shown in Fig. 1. Each bit c_i, $i = 0, 1, \ldots, m - 1$, is protected by a different binary LDPC code C^i of length n and rate $R_i = k_i/n$, where k_i is the information word length in bits. The mapping device maps a binary vector $\boldsymbol{c} = (c_0, ..., c_{m-1})$ to a signal point $x \in A$, where A is the signal set and $|A| = 2^m$. We consider a discrete equivalent additive white Gaussian noise
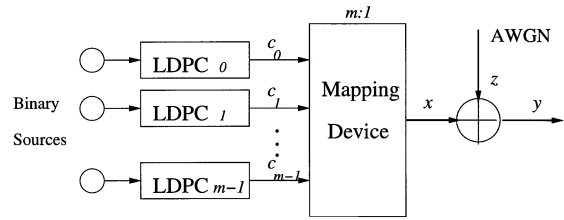
Fig. 1.   Encoder structure of the MLC scheme with LDPC component codes.

(AWGN) channel model, and we denote by z and y the channel noise and the channel output, respectively. The spectral efficiency R_s (bits per symbol) of the scheme is equal to the sum of the component code rates, i.e., $R_s = \sum_{i=0}^{m-1} R_i$.

Under the constraint of i.i.d equiprobable inputs, the capacity of such a channel with the channel input x and output y is given by[3] [19]

$$C = I(C_0, C_1, \ldots, C_{m-1}; Y) = I(X; Y)$$
$$= m - \boldsymbol{E}_{x,y} \left[\log_2 \frac{\sum_{a \in A} p(y|a)}{p(y|x)} \right]. \tag{1}$$

B. Multistage Decoding (MSD)

Applying the mutual information chain rule to (1) yields

$$I(C_0, \ldots, C_{m-1}; Y) = I(C_0; Y) + I(C_1; Y|C_0) + \cdots$$
$$+ I(C_{m-1}; Y|C_0, \ldots, C_{m-2}).$$

This equation implies that the transmission of vector $\boldsymbol{c} = (c_0, ..., c_{m-1})$ can be separated into the parallel transmission of c_i over m equivalent binary input channels, provided that $c_0, \ldots, c_{i-1}$ are known [4]. Accordingly, the component codes C^i are successively decoded based on the channel output and the decisions from lower levels. This is the well-known multistage decoding (MSD). The probability density function (pdf) $p(Y = y|c_i, c_0, \ldots, c_{i-1})$ for the equivalent channel i is given by

$$p(Y = y|c_i, c_0, \ldots, c_{i-1}) = \boldsymbol{E}_{a \in A(c_0, ..., c_i)}\{p(Y = y|X = a)\} \tag{2}$$

where $A(c_0, \ldots, c_i)$ denotes the subset of all the symbols of A whose labels have the value c_j in position j, $j = 0, \ldots, i$. The equivalent channel i is then specified by a set of pdfs [4]

$$\boldsymbol{p}(Y = y|C_i = c_i) = \{p(Y = y|c_i, c_0, \ldots, c_{i-1}),$$
$$\forall (c_0, \ldots, c_i) \in \{0, 1\}^i\}. \tag{3}$$

At the receiver side, for each equivalent binary input channel i, an *a posteriori* probability (APP) module uses y and the decisions from lower levels to compute the log-APP-ratio (LAPPR) for the coded bits, c_i, $i = 0, 1, \ldots, m - 1$. Applying Bayes' rule, it can be shown that the LAPPR of c_i is given by

$$v_i = \log \frac{\boldsymbol{Pr}(C_i = 0|Y = y, c_0, \ldots, c_{i-1})}{\boldsymbol{Pr}(C_i = 1|Y = y, c_0, \ldots, c_{i-1})}$$
$$= \log \frac{\boldsymbol{E}_{a \in A(c_0, ..., c_{i-1}, 0)}\{p(Y = y|X = a)\}}{\boldsymbol{E}_{a \in A(c_0, ..., c_{i-1}, 1)}\{p(Y = y|X = a)\}} \tag{4}$$
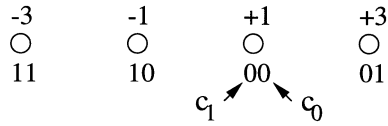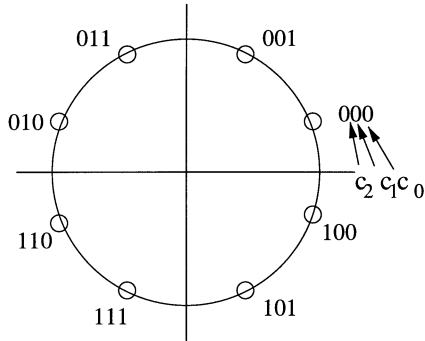
Fig. 2.   Gray-mapped 4-PAM modulation.

Fig. 3.   Gray-mapped 8-PSK modulation.

and v_i is used as the decoder input of the component code at level i.

C. Parallel Independent Decoding (PID)

Since the c_i, $i = 0, 1, \ldots, m - 1$, are independent of each other, it can be shown that

$$\sum_{i=0}^{m-1} I(C_i; Y) \le I(C_0, \ldots, C_{m-1}; Y).$$

The gap between $\sum_{i=0}^{m-1} I(C_i; Y)$ and $I(C_0, \ldots, C_{m-1}; Y)$ strongly depends on the mapping rule for the signal points. In particular, [4], [6] showed that this gap is surprisingly small if Gray mapping is employed. This result leads to a suboptimal but quite effective decoding strategy, namely, the decoding of the binary code C^i at each level i without using the decisions at any other level. With this PID strategy, the system can also be decomposed into an equivalent set of m parallel binary-input channels. Each equivalent binary-input channel i is characterized by the pdf

$$p(Y = y|C_i = b) = \boldsymbol{E}_{a \in A_b^i}\{p(Y = y|X = a)\} \quad (5)$$

where A_b^i denotes the subset of all the symbols of A whose labels have the value $b \in \{0, 1\}$ in position i.

At the receiver side, the LAPPR of c_i is calculated as

$$\begin{aligned} v_i &= \log \frac{\boldsymbol{Pr}(C_i = 0|Y = y)}{\boldsymbol{Pr}(C_i = 1|Y = y)} \\ &= \log \frac{\boldsymbol{E}_{a \in A_0^i} p(Y = y|X = a)}{\boldsymbol{E}_{a \in A_1^i} p(Y = y|X = a)}. \end{aligned} \quad (6)$$

With the i.i.d equiprobable inputs constraint, we define the "PID capacity" [4]

$$\hat{C} = \sum_{i=0}^{m-1} I(C_i; Y). \quad (7)$$

We consider both Gray-mapped 4-PAM (Fig. 2) and 8-PSK (Fig. 3) modulations. In Fig. 4, capacity results are plotted for a Gray-mapped 4-PAM modulation on an AWGN channel. The
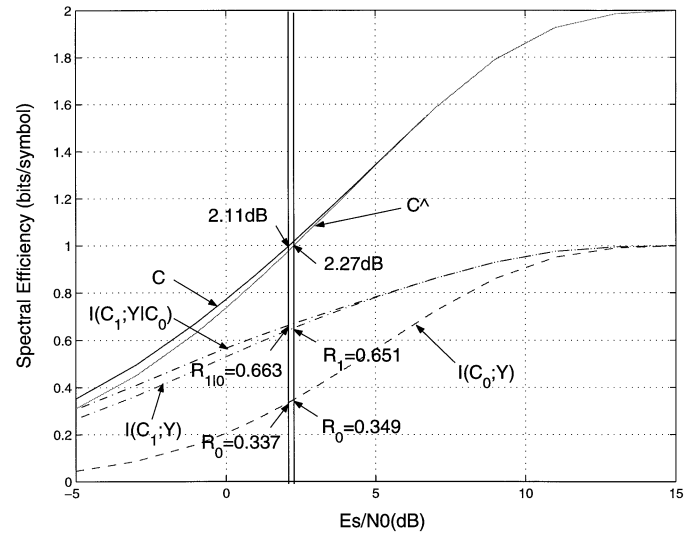
Fig. 4.   Capacity comparison for a Gray-mapped 4-PAM modulation on an AWGN channel.
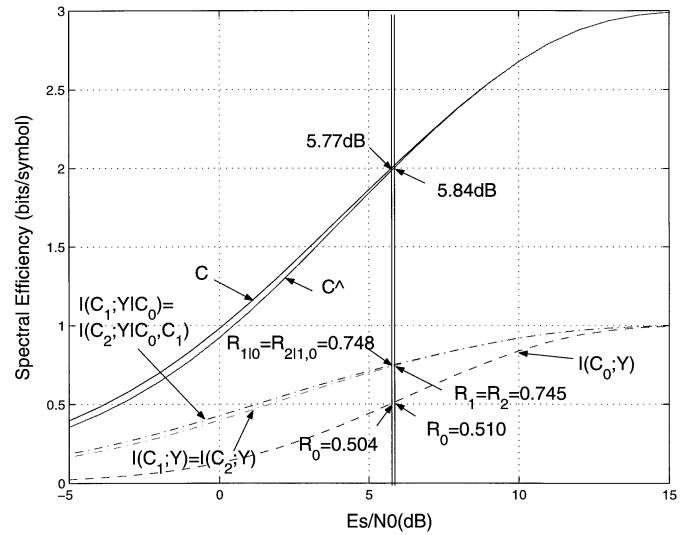
Fig. 5.   Capacity comparison for a Gray-mapped 8-PSK modulation on an AWGN channel.

plot shows that the PID capacity $\hat{C}$ suffers almost no degradation compared to the channel capacity C. For example, at a spectral efficiency of 1 bit/symbol, the reliable transmission SNRs corresponding to the channel capacity C and the PID capacity $\hat{C}$ are 2.11 and 2.27 dB, respectively. The curves also suggest the optimal individual code rates at each level: at an R_s of 1 bit/symbol, if MSD is used, the optimal code rates for each level are $R_0/R_1 = 0.337/0.663$; however, if PID is employed, $R_0/R_1 = 0.349/0.651$. Fig. 5 shows similar results for a Gray-mapped 8-PSK modulation on an AWGN channel. At an R_s of 2 bits/symbol, the performance loss of the PID capacity (5.84 dB) compared to the channel capacity (5.77 dB) is only 0.07 dB. Note that $I(C_1; Y) = I(C_2; Y)$ since the Gray labeling for c_1 and c_2 differs only by a rotation of 90° [4]. Furthermore, if c_0 is known, the optimal decision region of c_1 is independent of c_2. It can be shown that the equivalent transmission model of c_1 when c_0 is known is the same as the equivalent transmission model of c_2 when c_0 and c_1 are known. Therefore,

we have[4] $I(C_1; Y|C_0) = I(C_2; Y|C_1, C_0)$. According to the capacity results, the component code rate distribution at $R_s = 2$ bits/symbol is $R_0/R_1/R_2 = 0.504/0.748/0.748$ for MSD and $R_0/R_1/R_2 = 0.510/0.745/0.745$ for PID.

D. Bit-Interleaved Coded Modulation (BICM)

A pragmatic but quite effective approach for bandwidth-efficient transmission is to use BICM which requires only one encoder. The coded bits are interleaved bit-wise and grouped into blocks of m address bits $c_0, \ldots, c_{m-1}$. The signal point addressed by $c_0, \ldots, c_{m-1}$ is transmitted through the channel. The decoding metric for each $c_i, i = 0, \ldots, m-1$, is computed the same way as for the MLC/PID scheme. Finally, the decoder processes the deinterleaved metrics and outputs the decisions. As pointed out in [4], the equivalent channel models for bits $c_i, i = 0, \ldots, m - 1$, are identical for BICM and MLC/PID. The independence of the different bits c_i for the BICM scheme, which is inherent in the MLC/PID scheme, is based on the assumption of an ideal bit interleaver. Note that the equivalent channels for c_i in the BICM scheme are used serially rather than in parallel. Therefore, the MLC/PID capacity is the same as the performance limit that can be achieved by the BICM scheme, which is called the "BICM capacity" in [6].

III. ANALYSIS OF LDPC COMPONENT CODES FOR THE MLC AND BICM SCHEMES

A. LDPC Codes

An LDPC code is a linear block code which is specified by either its parity-check matrix $\boldsymbol{H}_{(n-k)\times n}$ or its corresponding generator matrix $\boldsymbol{G}_{k\times n}$ satisfying $\boldsymbol{H} \cdot \boldsymbol{G}^T = 0$. An LDPC code can be associated with a bipartite graph [1] which consists of n bit nodes, $n - k$ check nodes, and a certain number of edges. Each bit node represents a bit of the codeword. Each check node denotes one parity check of the code. An edge exists between the ith check node and the jth bit node if the entry $\boldsymbol{H}(i, j)$ is 1. An irregular LDPC code can be specified by either a degree distribution pair (λ, ρ) [1], [11] or, equivalently, its corresponding generating functions

$$\lambda(x) = \sum_{j=2}^{d_{v\max}} \lambda_j x^{j-1} \quad \text{and} \quad \rho(x) = \sum_{j=2}^{d_{c\max}} \rho_j x^{j-1}$$

where λ_j (resp., ρ_j) is the fraction of edges with bit (resp., check) degree j and $d_{v\max}$ (resp., $d_{c\max}$) is the maximal bit (resp., check) degree of any edge. A regular (d_v, d_c) LDPC code has $\lambda_{d_v} = 1$, $\rho_{d_c} = 1$. We define an LDPC code ensemble $C^n(\lambda, \rho)$ as the set of all LDPC codes of length n whose corresponding bipartite graphs satisfy the degree distribution pair (λ, ρ).

In [1], a numerical technique called density evolution is used to analyze the performance of message-passing decoders on a binary-input symmetric AWGN channel, enabling the accurate

determination of the noise thresholds [1] of LDPC code ensembles. The interpretation of the thresholds as predictors of actual decoder behavior and bounds on achievable performance relies upon a general "concentration theorem"—stating that, asymptotically in the block size, the decoder behavior for individual instances (of the code and the channel noise) concentrates around the average behavior of a cycle-free graph, which can be computed using the density evolution algorithm. The application of the concentration theorem and density evolution to the determination of the noise threshold of LDPC code ensembles is simplified by the symmetry of the channel /and decoding algorithm [1]. Specifically, under appropriate symmetry conditions, it suffices to consider the performance of the all-zeros codeword.

B. I.I.D. Channel Adapters

Our objective is to develop a similar algorithmic approach for the analysis of LDPC component codes for the MLC and BICM schemes. The application of density evolution and concentration theorem for the MLC and BICM schemes is complicated because, in general, the equivalent binary-input component channels are not necessarily symmetric where a binary-input channel is symmetric if

$$p(Y = y|C = 0) = p(Y = -y|C = 1) \qquad (8)$$

with c and y as the input and output of the binary-input channel, respectively [1]. Therefore, the decoding analysis of the all-zeros codeword alone may not suffice to predict the average decoder behavior; in fact, for the specific Gray-labeled constellation we considered in Section II, it is easy to see that this is the case.

In the following, we introduce a new analytical tool: i.i.d. channel adapters. We show that we can force the symmetry of the equivalent binary-input component channels with the use of i.i.d. channel adapters. Thus, the analysis and design of binary LDPC codes are greatly simplified.

We use the MLC/PID scheme as an example, and the extensions to the MLC/MSD and BICM schemes are straightforward. Fig. 6 shows an MLC/PID scheme with an i.i.d. channel adapter on each equivalent binary-input component channel. Each i.i.d. channel adapter has three modules. The first one is an i.i.d. source which generates binary symbol $t_i \in \{0, 1\}$, $i = 0, 1, \ldots, m - 1$, according to an i.i.d. equiprobable distribution. The second one is a mod-2 adder and performs the following operation: $d_i = c_i \oplus t_i$, where c_i is the LDPC-coded bit. The last module is a sign adjuster and functions as follows: $v_i = u_i \cdot (1 - 2 \cdot t_i)$, which means $v_i = u_i$ if $t_i = 0$ and $v_i = -u_i$ if $t_i = 1$, where u_i is the APP module output and v_i is the LDPC decoder input. We can see that the last module undo-es the effect of the second module. Therefore, each equivalent binary-input channel $i, i = 0, 1, \ldots, m-1$, is transformed into a new binary-input channel with input c_i and output v_i. We have the following theorem.

Theorem 1: All of the new augmented binary-input channels as previously defined satisfy the symmetry condition. That is, $\forall i = 0, 1, \ldots, m - 1$

$$p(V_i = v_i|C_i = 0) = p(V_i = -v_i|C_i = 1).$$

---

[4]Actually, it can be proven that both $I(C_1; Y|C_0)$ and $I(C_2; Y|C_1, C_0)$ are equal to the uniform average of the capacities of two equivalent binary phase-shift keying (BPSK) modulations.
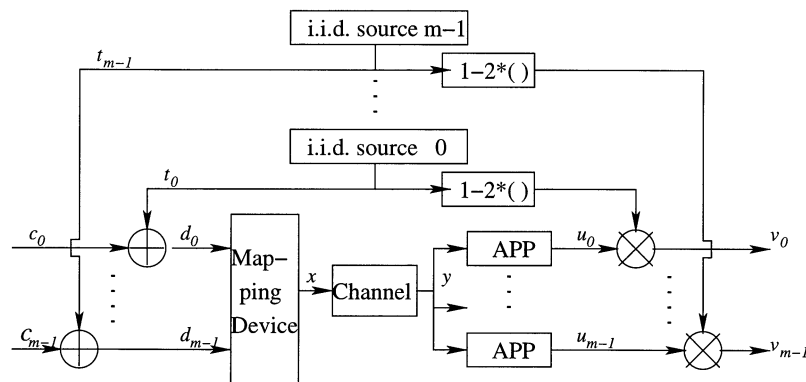
Fig. 6.   The MLC/PID scheme with an i.i.d. channel adapter on each equivalent binary-input channel.
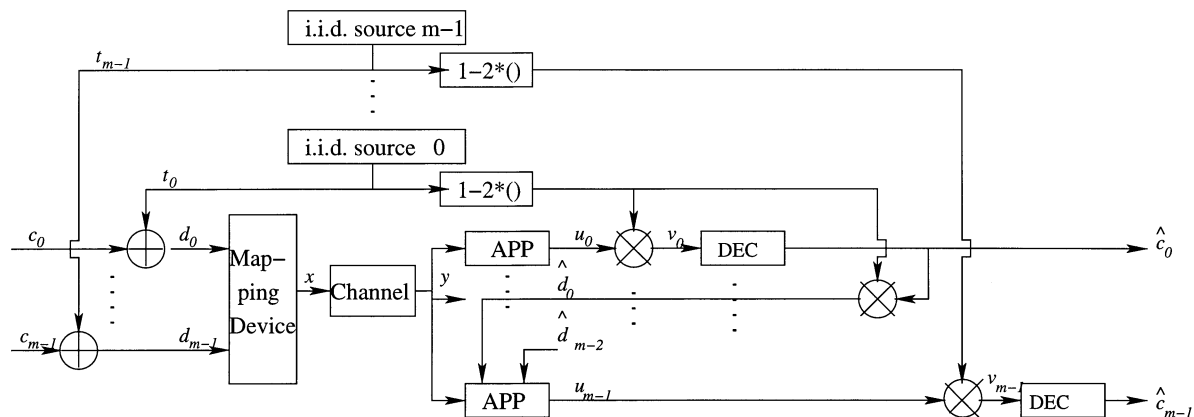

Fig. 7.   The MLC/MSD scheme with an i.i.d. channel adapter on each equivalent binary-input channel.
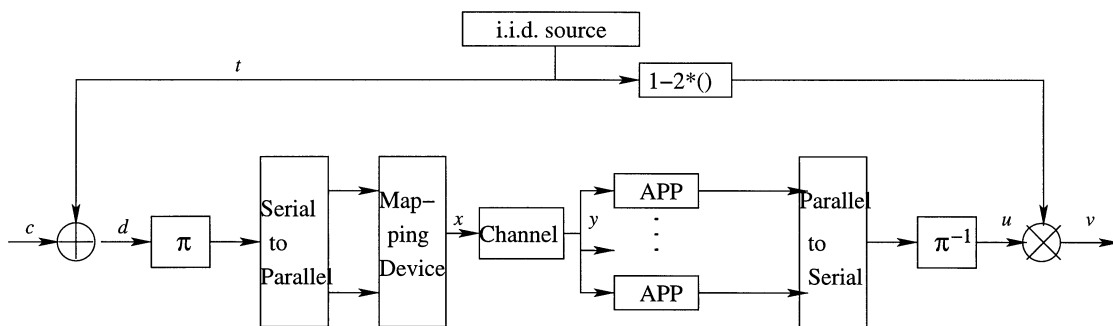
Fig. 8.   The BICM scheme with an i.i.d. channel adapter.

Proof: It is easy to see that, $\forall\, i = 0, 1, \ldots, m-1$, both T_i and D_i are i.i.d. equiprobable random variables. Noticing that T_i and C_i are independent, we have

$$p(V_i = v_i | C_i = 0)$$
$$= p(V_i = v_i | T_i = 0, C_i = 0)\mathbf{Pr}(T_i = 0 | C_i = 0)$$
$$\quad + p(V_i = v_i | T_i = 1, C_i = 0)\mathbf{Pr}(T_i = 1 | C_i = 0)$$
$$= p(U_i = v_i | T_i = 0, C_i = 0)\mathbf{Pr}(T_i = 0)$$
$$\quad + p(U_i = -v_i | T_i = 1, C_i = 0)\mathbf{Pr}(T_i = 1)$$
$$= \tfrac{1}{2}p(U_i = v_i | D_i = 0) + \tfrac{1}{2}p(U_i = -v_i | D_i = 1).$$

Similarly, we have

$$p(V_i = -v_i | C_i = 1)$$
$$= \tfrac{1}{2}p(U_i = -v_i | D_i = 1) + \tfrac{1}{2}p(U_i = v_i | D_i = 0).$$

Then Theorem 1 follows directly. $\square$

We show the MLC/MSD and BICM block diagrams with i.i.d. channel adapters in Figs. 7 and 8, respectively. By similar arguments as in Theorem 1, it can be proven that each of the new augmented binary-input channels shown in Figs. 7 and 8 is symmetric, as well.

Therefore, for any of the new augmented binary-input output-symmetric component channels, if we use an LDPC code for transmission through this channel, by [1, Lemma 1], the decoding (bit or block) error probability is independent of any particular codeword. Thus, the threshold analysis and code design of LDPC codes on these kinds of channels are greatly simplified as we need only consider the all-zeros codeword. Furthermore, by [1, Theorem 2], the average behavior of individual instances (of the code and of the noise) concentrate around the expected behavior when the codeword length gets sufficiently large.

Theorem 2: The capacity of the new augmented binary-input channel formed by adding an i.i.d. channel adapter to the original equivalent binary-input component channel is equal to the mutual information of the original binary-input channel with i.i.d. equiprobable input distribution.

Proof: By Theorem 1, the new augmented channel with input c and output v is output symmetric, and, therefore, the capacity of this new channel can be achieved by an i.i.d. equiprobable input distribution [20]. That is, the capacity of the new channel is equal to the average of the mutual information between channel input $c = b$ and output v, where $b \in \{0, 1\}$ [20]. However, no matter which value (0 or 1) the channel input c takes, the original binary-input channel sees i.i.d. equiprobable inputs because of the i.i.d. channel adapter, therefore limiting the mutual information between $c = b$ and v, where $b \in \{0, 1\}$, to the i.i.d. mutual information of the original binary-input channel. Therefore, the capacity of the new augmented binary-input channel equals the i.i.d. mutual information of the original binary-input channel.   □

By Theorem 2, if we can approach the capacity of the new augmented binary-input channel with the i.i.d. channel adapters, we are able to approach the i.i.d. channel capacity (see (1)) by the MLC/MSD scheme and approach the PID capacity (see (7)) by the MLC/PID scheme or the BICM scheme.

In a system with i.i.d. channel adapters, on each new augmented binary-input channel, the expected decoder behavior (BER) is effectively averaged over all possible LDPC graphs and all possible channel realizations. In particular, each channel realization includes one randomly chosen binary vector—length-mn for MLC and length-n for BICM—chosen according to an i.i.d. equiprobable distribution and one channel noise realization. Therefore, the average of the expected behavior over the channel is an average with respect to both the i.i.d. binary vector and the channel noise.

C. LDPC Coset Codes and Concentration Theorem

In Section III-B, by incorporating i.i.d. channel adapters, we showed that the new binary-input augmented channels are symmetric. Therefore, the analysis of LDPC codes is simplified. In this section, we keep the channel unchanged. Instead, by considering a slightly broader class of codes—coset codes—we are able to show that if the input sequence s has an i.i.d. equiprobable distribution, for almost all graphs and almost all input sequences, the decoder performance over each equivalent binary-input channel of the MLC and BICM schemes concentrates around its expected behavior. The outline of the proof is very similar to [1], [2], which is to form a martingale process by revealing information one step at a time (of the graph ensemble, the input sequences, and the channel noise realizations). If the impact of the information revealing at one step is restricted to a finite value independent of n, a tight concentration bound results from the Azuma inequality [21].

1) LDPC Coset Codes and Decoding: Following the definition of [20], we specify a coset code by G (or H) and a fixed but arbitrary coset leader $o_{n \times 1} = [o_1, o_2, \ldots, o_n]^T$. If $u_{k \times 1}$ is an

information vector, the codeword $w_{n \times 1} = [w_1, w_2, \ldots, w_n]^T$ is generated as

$$w = G^T \cdot u \oplus o = c \oplus o \qquad (9)$$

where $\oplus$ represents $\mathrm{mod}\text{-}2$ addition and c is the codeword of the associated linear code. The codeword w of the coset code satisfies

$$H \cdot w = r^s = H \cdot o \qquad (10)$$

where

$$r^s_{(n-k) \times 1} = [r^s_1, r^s_2, \ldots, r^s_{(n-k)}]^T$$

is the syndrome of the corresponding coset leader o and $r^s_i = \oplus_{j:\, H(i,\, j)=1} o_j$. The coset code is linear if and only if $r^s = 0$. Since an LDPC code is a linear block code, an LDPC coset code obeys this definition.

By decoder symmetry [1], it can be proven that the following two ways of decoding an LDPC coset code are identical. One way is to include the syndrome r of the coset leader o into the iterative message-passing decoding algorithm as described in [2], i.e., the message $q_{v,\, ij}$ passed from check node i to bit node j is determined by

$$\tanh\left(\frac{q_{v,\, ij}}{2}\right) = (-1)^{r^s_i} \prod_{l \neq j} \tanh\left(\frac{q_{c,\, il}}{2}\right)$$

where $q_{c,\, il}$ is the message passed from bit node l to check node i. When the decoder makes a final decision on w, after subtracting o, the decoder recovers the information sequence u according to (9). The other way is similar to the decoding algorithm described in [20] (the example discussed there is for a binary-symmetric channel (BSC)), where we remove the coset leader first and then perform the message-passing decoding on its associated linear LDPC code. That is, for each bit position j of a codeword, $j = 1, \ldots, n$, assuming that v_j is the LAPPR value of the corresponding bit position, we calculate $q_j = (-1)^{o_j} v_j$, where o_j is the jth component of coset leader o. Then we use q_j as the input to the associated linear LDPC decoder, and the expression

$$\tanh\left(\frac{q_{v,\, ij}}{2}\right) = \prod_{l \neq j} \tanh\left(\frac{q_{c,\, il}}{2}\right)$$

is used as the basis for the message passing from check nodes to bit nodes. Finally, we recover u from the final decision of the decoder.

For the MLC scheme, we consider one LDPC coset code at each level. If $w^i = [w^i_1, w^i_2, \ldots, w^i_n]^T$, $i = 0, 1, \ldots, m-1$, is a codeword of a coset code at level i, we define an input sequence of the MLC scheme as the time multiplex of m codewords w^i, $i = 0, 1, \ldots, m-1$, i.e.,

$$s_{mn \times 1} = [w^0_1, w^1_1, \ldots, w^{m-1}_1, \ldots, w^0_n, w^1_n, \ldots, w^{m-1}_n]^T.$$

Then, the mapper output is defined as

$$x_{n \times 1} = [x_1, x_2, \ldots, x_n]^T$$

where $[w^0_j, w^1_j, \ldots, w^{m-1}_j]$ is assigned to x_j, $j = 1, \ldots, n$, according to a given mapping rule. We call x the symbol sequence, which is the input to the noisy channel.

For the BICM scheme, if w is a codeword of an LDPC coset code, we define $w' = [w'_1, w'_2, \ldots, w'_n]^T$ to be the interleaver output. Then, the symbol sequence is given by
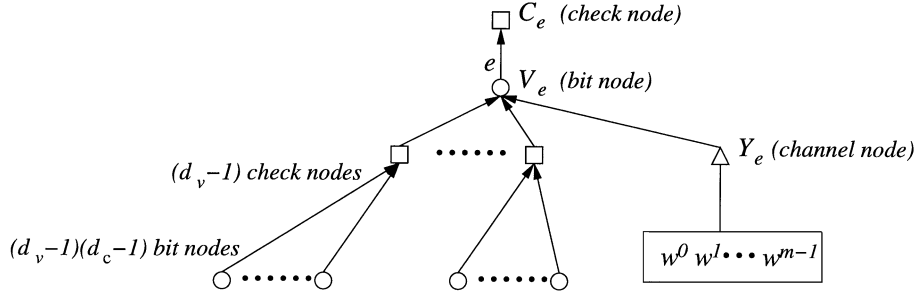
Fig. 9.   The modified directed neighborhood of depth 2 of the directed edge $\vec{e} = (V_e, C_e)$.

$\boldsymbol{x}_{\frac{n}{m} \times 1} = [x_1, x_2, \ldots, x_{\frac{n}{m}}]^T$ (we assume that m divides n), where

$$[w'_{(j-1)m+1}, w'_{(j-1)m+2}, \ldots, w'_{jm}]$$

is mapped to x_j, $j = 1, \ldots, \frac{n}{m}$.

2) Concentration Theorem: First, we consider a given graph g (representing the LDPC code at level i) and a given input sequence $\boldsymbol{s}$ ($\boldsymbol{s}$ is the time multiplex of coset codewords $\boldsymbol{w}^0$, $\boldsymbol{w}^1, \ldots, \boldsymbol{w}^{m-1}$). At level i, let $\boldsymbol{c}^i$ be an arbitrary codeword of g. The coset-defining vector $\boldsymbol{o}^i$ then is given by $\boldsymbol{o}^i = \boldsymbol{w}^i \oplus \boldsymbol{c}^i$. This means that different (actually, any) codewords of g can result in the same coset codeword by using different coset-defining vectors. In the following lemma, we will show that for a given g, under the same $\boldsymbol{s}$ (thereby, the same $\boldsymbol{w}^i$), the number of errors committed by the decoder is independent of $\boldsymbol{c}^i$.

Lemma 1:   Let g be the bipartite graph representing a given binary linear LDPC code at level i of the MLC scheme. Consider the belief propagation decoding algorithm described in [10], [1] which satisfies the decoder symmetry (including both bit node symmetry and check node symmetry) defined in [1]. Let $\boldsymbol{c}^i_1$ and $\boldsymbol{c}^i_2$ be two arbitrary linear codewords of g. Assume that $\boldsymbol{s}$, the input sequence of the MLC scheme, is the time multiplex of coset codewords $\boldsymbol{w}^0, \boldsymbol{w}^1, \ldots, \boldsymbol{w}^{m-1}$. If $\boldsymbol{c}^i_1 \oplus \boldsymbol{o}^i_1 = \boldsymbol{c}^i_2 \oplus \boldsymbol{o}^i_2 = \boldsymbol{w}^i$, the number of decoding errors is exactly the same, irrespective of whether $\boldsymbol{c}^i_1$ or $\boldsymbol{c}^i_2$ is the transmitted codeword.

The proof of Lemma 1 is given in Appendix A. From Lemma 1, we conclude that the number of errors committed by the decoder is a function of only graphs, input sequences, and channel noise realizations.

The proof of the concentration theorem is for the MLC/PID scheme, but it can be extended to the MLC/MSD and BICM schemes in a very similar manner. For the message-passing decoder of each LDPC component code, we consider the first decoding approach described in Section III-C1. Similar to [1], to simplify the subsequent notation, we assume that the number of iterations that the decoder performs is denoted as l. All subsequent notations refer to iteration l, and we frequently omit the index l.

Here, we limit our consideration to regular LDPC codes.[5] Reference [1] introduced the idea of a directed neighborhood $N^{2l}_{\vec{e}}$ of depth $d = 2l$ which consists of an edge $\vec{e} = (V_e, C_e)$ that connects a bit node V_e and a check node C_e, and all nodes and edges traversed by paths of length at most $2l$ ending at V_e,

where l is the message-passing iteration number. Similar to [2], we modify the directed neighborhood for the MLC/PID scheme. The modified directed neighborhood of depth-2 $(l = 1)$ $N^2_{\vec{e}}$, shown in Fig. 9, consists of the two nodes V_e and C_e, the edge $\vec{e}$, all nodes and edges traversed by paths of length at most 2 ending at V_e, one channel node Y_e, and its associated m binary symbols $[w^0, w^1, \ldots, w^{m-1}]$. In the MLC/PID scheme, the binary symbols $[w^0, w^1, \ldots, w^{m-1}]$ are part of the input sequence $\boldsymbol{s}$ with w^i corresponding to the coded bit from the component code at level i. The vector $[w^0, w^1, \ldots, w^{m-1}]$ maps to a channel symbol x associated with the channel node Y_e that contributes to the message passing along edge $\vec{e}$. A modified directed neighborhood of depth $2l$ can be obtained by branching out the neighborhood of depth 2.

Since the transmitted binary symbols associated with the channel nodes influence the statistics of the messages passed in the modified directed neighborhood, we must distinguish between different neighborhoods by different types. We specify each type of neighborhood by the transmitted binary symbols on its associated channel nodes. For a (d_v, d_c) regular LDPC code, the total number of the channel nodes in a modified directed neighborhood of depth $2l$ is given by

$$g_c(2l) = \frac{(d_v - 1)^l (d_c - 1)^l - 1}{(d_v - 1)(d_c - 1) - 1}. \tag{11}$$

We can arrange the binary symbols of the channel nodes in a modified directed neighborhood into a binary vector

$$\boldsymbol{t} = [w^{1, 0}, w^{1, 1}, \ldots, w^{1, m-1}, \ldots,$$
$$w^{g_c(2l), 0}, w^{g_c(2l), 1}, \ldots, w^{g_c(2l), m-1}] \tag{12}$$

and the length of $\boldsymbol{t}$ is defined as $h(2l) \triangleq m \cdot g_c(2l)$. Since each type of neighborhood is specified by a vector $\boldsymbol{t}$, the number of possible modified directed neighborhood types of depth $2l$ is $2^{h(2l)}$. As in [1], we say that the modified directed neighborhood is tree-like if all nodes in the neighborhood are distinct; otherwise, we say that it is not tree-like. The inequality proved in [1] applies directly here

$$\mathbf{Pr}(N^{2l}_{\vec{e}} \text{ is not tree-like}) \leq \frac{\gamma}{n} \tag{13}$$

where γ is a constant that may depend on (d_v, d_c) and l, but is independent of n.

Let Z_j be the number of incorrect messages passed along edge $\vec{e_j}$. Let T_r be the random variable which has a value of 0 if $N^{2l}_{\vec{e_j}}$ is tree-like and 1 otherwise. Given a modified directed neighborhood $N^{2l}_{\vec{e_j}}$, we define

$$\tau_{\boldsymbol{t}} = \boldsymbol{E}(Z_j | \boldsymbol{t}, T_r = 0) \tag{14}$$

---

[5]The extension of the proof to irregular LDPC codes is straightforward.

as the expected number of incorrect messages passed along $\vec{e_j}$ with a tree-like neighborhood $N_{\vec{e_j}}^{2l}$ of depth $2l$ at the lth iteration when t is the neighborhood type.

We define the probability $\overline{\Pr}(t|s, T_r = 0)$ as the probability that a modified directed neighborhood $N_{\vec{e_j}}^{2l}$ is of type t when the input sequence of the MLC scheme is s, $N_{\vec{e_j}}^{2l}$ is tree-like, and the associated bipartite graph is chosen randomly from all possible graphs of the code ensemble $C^n(d_v, d_c)$.

Therefore, we can define the expected number of incorrect messages passed along edge $\vec{e_j}$ with a tree-like neighborhood of depth $2l$ after the lth iteration when an input sequence s is transmitted as

$$
\begin{aligned}
p(s) &= \boldsymbol{E}(Z_j|s, T_r = 0) \\
&= \sum_{\kappa=1}^{2^{h(2l)}} \boldsymbol{E}[Z_j|t_\kappa, s, T_r = 0]\Pr(t_\kappa|s, T_r = 0) \\
&= \sum_{\kappa=1}^{2^{h(2l)}} \tau_{t_\kappa}\Pr(t_\kappa|s, T_r = 0)
\end{aligned}
\tag{15}
$$

where

$$
\boldsymbol{E}[Z_j|t_\kappa, s, T_r = 0] = \boldsymbol{E}[Z_j|t_\kappa, T_r = 0] = \tau_{t_\kappa}
$$

since Z_j is independent of s if the type t is known.

If all possible $2^{h(2l)}$ modified directed neighborhood types t_κ, $\kappa = 1, \ldots, 2^{h(2l)}$, are equally likely, we call this equiprobable expected value of error p_{eq}, i.e.,

$$
p_{\mathrm{eq}} = 2^{-h(2l)} \sum_{\kappa=1}^{2^{h(2l)}} \tau_{t_\kappa}.
\tag{16}
$$

In the following, we present the concentration theorem based on coset codes and the proof is shown in Appendix B.

Theorem 3: Define an $mn \times 1$ binary random vector

$$
\boldsymbol{S} = [S_1^0, S_1^1, \ldots, S_1^{m-1}, \ldots, S_n^0, S_n^1, \ldots, S_n^{m-1}]
$$

where $\{S_j^i, j = 1, 2, \ldots, n, i = 0, 1, \ldots, m - 1\}$ are i.i.d. equiprobable binary random variables. We assume that $\boldsymbol{S}$ is the input sequence of the MLC scheme. At level i of the MLC/PID scheme, over the probability space of all graphs $C^n(d_v^i, d_c^i)$, all realizations of $\boldsymbol{S}$, and all channel noise realizations, let Z^i be the random variable that denotes the number of incorrect messages among all nd_v^i bit-to-check node messages passed at iteration l. Then, for any $\epsilon > 0$, there exist positive numbers β and γ (they may depend on d_v^i, d_c^i, and l, but not on n), such that if $n > \frac{2\gamma}{\epsilon}$, we have

$$
\Pr\left(\left| \frac{Z^i}{nd_v^i} - p_{\mathrm{eq}}^i \right| \geq \epsilon \right) \leq 2e^{-\beta\epsilon^2 n}
$$

where p_{eq}^i is given by (16) and i denotes level i in the MLC scheme.

Corollary: Let $\boldsymbol{U}^i$, $i = 0, 1, \ldots, m - 1$, be the information vector with each binary symbol i.i.d. equiprobable. Let $C(G^i, \boldsymbol{O}^i)$ be an LDPC coset code for which G^i is a code graph in the LDPC code ensemble $C^n(d_v^i, d_c^i)$ and $\boldsymbol{O}^i$ is a coset-defining binary vector with each binary symbol satisfying an i.i.d. equiprobable distribution. Define Z^i to be the random variable that denotes the number of incorrect messages among all nd_v^i bit-to-check node messages passed at iteration l, assuming $C(G^i, \boldsymbol{O}^i)$ is the component code used at level i of

the MLC/PID scheme. Then, for the same ϵ, β, γ and n as in Theorem 3, we have

$$
\Pr\left(\left| \frac{Z^i}{nd_v^i} - p_{\mathrm{eq}}^i \right| \geq \epsilon \right) \leq 2e^{-\beta\epsilon^2 n}.
$$

Proof: If $\boldsymbol{U}^i$ and $\boldsymbol{O}^i$ satisfy an i.i.d. equiprobable distribution, the resulting input sequence of the MLC/PID scheme $\boldsymbol{S}$ is a sequence of i.i.d. equiprobable random binary symbols. Therefore, Theorem 3 applies directly. $\square$

It follows from the corollary that (for sufficiently large n) the decoding behavior of almost all input sequences converges to the expected value p_{eq}^i. Therefore, if we can find the maximum channel noise standard deviation, namely, the threshold such that p_{eq}^i goes to zero, almost all input sequences can transmit reliably up to the threshold value, but they have an error probability bounded away from zero above the threshold value.

D. Relation Between These Two Systems

We can see that the LDPC linear code system with i.i.d. channel adapters and the LDPC coset code system with the second decoding approach are virtually the same system. The critical difference between these two systems is that in the first system, we take the i.i.d. binary vector as a channel-adapting vector and, therefore, as part of the new augmented channel; however, in the second system, we consider the i.i.d. binary vector to be a time multiplex of coset-defining vectors and, therefore, as part of the LDPC coset codes.

In the first system, each codeword belongs to a linear code, and the averaged (over the channel) decoder behavior conditioned on a particular codeword is the same for each possible codeword since the new augmented binary-input channels are symmetric.

In the second system, one codeword is a codeword of a coset code, which is the $\mathrm{mod}\text{-}2$ sum of a codeword of its associated linear code and a coset-defining vector, and the averaged (over the channel) decoder behavior conditioned on a particular codeword may be different from one codeword to another, since the equivalent binary-input channels are not necessarily symmetric. Therefore, a coset code concentration theorem is given for almost all possible input sequences (or its related coset codewords).

Nevertheless, the expected decoder behavior of the two systems is the same, since they have the same configuration (encoder and decoder structure) and the expectations are taken over the same probability space, i.e., over all possible LDPC graphs, over all possible i.i.d. binary vectors (referring to channel-adapting vectors in the first system and referring to coset-defining vectors in the second system), and over all possible noise realizations.

Next, we will determine the expected decoder behavior for the system with i.i.d. channel adapters by considering only the all-zeros codeword.

IV. DENSITY EVOLUTION AND CODE OPTIMIZATION

A. Density Evolution

Here, we briefly describe the manner in which we extend density evolution to the MLC and BICM schemes based on the

i.i.d. channel adapters. We consider transmitting the all-zeros codeword on each augmented binary-input component channel. First, we consider the Gray-mapped MLC/MSD scheme on an AWGN channel. The conditional pdf $p(Y = y|X = a)$ is the Gaussian density function

$$p(Y = y|X = a)$$
$$= \begin{cases} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y-a)^2}{2\sigma^2}\right), & \text{if } y \text{ is real-valued} \\ \frac{1}{2\pi\sigma^2} \exp\left(-\frac{|y-a|^2}{2\sigma^2}\right), & \text{if } y \text{ is complex-valued} \end{cases} \quad (17)$$

where σ^2 is the variance of the channel noise and $a \in A$.

Note that $d_0, \ldots, d_{m-1}$ are the inputs to the mapping device. At level i, assuming we know $d_0, \ldots, d_{i-1}$, for any given $a \in A(d_0, \ldots, d_{i-1})$, from the relation between y and u_i (u_i is LAPPR value of d_i, given y and $d_0, \ldots, d_{i-1}$), we can calculate the conditional pdf $p(U_i = u_i|X = a)$. Since $\{d_i, i = 0, \ldots, m-1\}$ are variables satisfying i.i.d. equiprobable distributions, all the signal points in the constellation are equally likely to be transmitted. Therefore,

$$f_0^i = p(V_i = v_i)$$
$$= \frac{1}{2^m} \left[\sum_{d_0, \ldots, d_{i-1}} \left(\sum_{a \in A(d_0, \ldots, d_{i-1}, 0)} p(U_i = v_i|X = a) \right. \right.$$
$$\left. \left. + \sum_{a \in A(d_0, \ldots, d_{i-1}, 1)} p(U_i = -v_i|X = a) \right) \right]. \quad (18)$$

Then, f_0^i is used as the initial density of the observed LAPPRs of the augmented binary-input component channels in the density evolution program. Following [11], for a specified noise standard deviation σ, at each level i of the MLC/MSD scheme, $i = 0, 1, \ldots, m-1$, we use density evolution to track the fraction of incorrect messages $p_e^i(l)$ after l decoding iterations on a cycle-free graph corresponding to a specified degree distribution pair (λ^i, ρ^i). We let f_l^i denote the density of the messages passed from the bit nodes to the check nodes after l iterations. The density evolution can be described by

$$f_l^i = f_0^i \otimes \lambda^i(\Gamma^{-1}(\rho^i(\Gamma(f_{l-1}^i))))$$

where

$$\lambda^i(f) = \sum \lambda_j^i f^{\otimes(j-1)}$$
$$\rho^i(f) = \sum \rho_j^i f^{\otimes(j-1)}$$

$\otimes$ denotes convolution, and Γ and Γ^{-1} are operators defined in [11, eqs. (5) and (6)], respectively. For each level i, we define the corresponding noise threshold $(\sigma^*)^i$ to be the supremum of the σ for which $\lim_{l\to\infty} p_e^i(l) = 0$.

Similarly, for the MLC/PID scheme, we have

$$f_0^i = p(V_i = v_i) = \frac{1}{2^m} \left(\sum_{a \in A_0^i} p(U_i = v_i|X = a) \right.$$
$$\left. + \sum_{a \in A_1^i} p(U_i = -v_i|X = a) \right). \quad (19)$$

Recall that the MLC/PID scheme and the BICM scheme have the same equivalent channel models for each level. Therefore, for the BICM scheme, the initial density function of the ob-

served LAPPRs is the average of f_0^i obtained in the MLC/PID scheme, i.e., $f_0 = (\sum_i f_0^i)/m$.

B. Code Optimization

In general, we need to optimize the LDPC component codes so that the MLC scheme can approach the channel capacity. If we know that the LDPC codes can achieve capacity for each equivalent component channel, we can fix the component code rates to the required rates computed based on mutual information chain rule and simply optimize the thresholds of the component code rates. However, for a fixed maximal bit (or check) degree, no one had proved that LDPC codes can get arbitrarily close to the capacity of these equivalent channels. Therefore, in this work, to design an optimal MLC coding scheme with LDPC codes as component codes, we perform joint optimization of both the code rates and the degree distributions of the LDPC component codes for all the levels. If the target system spectral efficiency is R_s and the code rate of the LDPC code C^i is R_i, $i = 0, 1, \ldots, m-1$, we have $R_s = \sum_{i=0}^{m-1} R_i$. Since the optimal design of MLC schemes requires that the component codes at each level have equal performance [4], under the constraint imposed by R_s, we should optimize both the code rates and the degree distributions of the LDPC component codes in such a way that all the LDPC component codes have the same noise thresholds.

For a 4-PAM modulation, the joint optimization is as follows. Under the constraint of $R_s = 1$ bit/symbol, we randomly pick a combination of R_i, $i = 0, 1$. For this combination of R_i, $i = 0, 1$, we use a nonlinear optimization technique, called differential evolution [22], to search for the optimal degree distribution pair (λ^i, ρ^i) and its corresponding noise threshold $(\sigma^*)^i$ for the LDPC code C^i at each level. If the optimized LDPC codes at both levels have different noise thresholds, we make the code rate adjustments continuously to minimize the difference of the noise thresholds of the LDPC codes at both levels. If the LDPC codes at both levels have the same threshold value, we stop the search algorithm and claim that we have found both the optimal code rates and the degree distribution pairs of the LDPC codes at both levels for the MLC scheme. This optimization method applies to both the MLC/MSD scheme and the MLC/PID scheme.

For the BICM scheme, the rate of the LDPC code is predetermined by the system spectral efficiency. In the case of 4-PAM modulation, if R_s is 1 bit/symbol, the LDPC code rate $r_c = R_s/m = 1/2$. Then, we just need to combine the differential evolution and density evolution to find a degree distribution pair which has the best threshold.

V. NUMERICAL RESULTS

A. Thresholds and Very Large Block-Size Simulation Results

By applying the optimization technique discussed above, we optimize both the code rates and degree distributions of the LDPC component codes for the MLC and BICM schemes. In the following, we primarily focus on the discussion of the optimization results for the 4-PAM modulation case, but show some optimization results for the 8-PSK modulation as well.

1) Gray-Mapped 4-PAM: In the case of 4-PAM modulation, the target R_s is 1 bit/symbol. For the MLC scheme,

TABLE I
OPTIMIZED RESULTS FOR THE EQUIVALENT CHANNEL $i = 0, 1$ OF THE
GRAY-MAPPED MLC/MSD SCHEME (4-PAM MODULATION), $R_s = 1$
BIT/SYMBOL, AND THE CHANNEL CAPACITY IS 2.11 dB

$d_{v_{max}}$	10	20	50
$\left(\frac{E_s}{N_0}\right)^*$ (dB), i=0	2.353	2.267	2.181
$\left(\frac{E_s}{N_0}\right)^*$ (dB), i=1	2.353	2.267	2.181

TABLE II
OPTIMIZED RESULTS FOR THE EQUIVALENT CHANNEL $i = 0, 1$ OF THE
GRAY-MAPPED MLC/PID SCHEME (4-PAM MODULATION), $R_s = 1$
BIT/SYMBOL, AND THE PID CAPACITY IS 2.27 dB

$d_{v_{max}}$	10	15	30	50
$\left(\frac{E_s}{N_0}\right)^*$ (dB), i=0	2.534	2.454	2.388	2.352
$\left(\frac{E_s}{N_0}\right)^*$ (dB), i=1	2.542	2.454	2.396	2.324

the optimized code rates of the two component LDPC codes are very close to the capacity results shown in Section II. For the MLC/MSD scheme, the joint optimization produces slightly different code rate distributions for different $d_{v_{max}}$: for $d_{v_{max}} = 10$, the code rates of C^0 and C^1 are 0.339 and 0.661 and for $d_{v_{max}} = 20$, and 50, the code rates are 0.338 and 0.662. Similarly, for the MLC/PID scheme, the optimized code rate combinations of C^0 and C^1 are 0.350 and 0.650, respectively. For the BICM scheme, we simply set the LDPC component code rate to $1/2$.

Table I lists the optimized results for the MLC/MSD scheme with constraints of $d_{v_{max}} = 10, 20,$ and 50. For each $d_{v_{max}}$, the threshold $(E_s/N_0)^*$ (decibels) of the optimized degree distribution pair is given. Note that the channel capacity is 2.11 dB. The degree distribution pairs of code C^0 and C^1 with $d_{v_{max}} = 50$ both have threshold of 2.18 dB, which is only 0.07 dB away from the channel capacity. By comparison, the quasi-regular [13] rate-0.338 $(3, 4/5)$ LDPC code (C^0) and rate-0.662 $(3, 8/9)$ LDPC code (C^1) have much worse thresholds of 3.29 and 3.32 dB, which are 1.18 and 1.21 dB away from the channel capacity, respectively.

For the MLC/PID scheme, we list in Table II the optimized results for C^0 and C^1 with $d_{v_{max}} = 10, 15, 30,$ and 50. Note that the PID capacity is 2.27 dB. The thresholds of the optimized degree distribution pairs are very close to the PID capacity. The degree distribution pairs of code C^0 and C^1 with $d_{v_{max}} = 50$ have thresholds of 2.35 and 2.32 dB,[6] which are only 0.082 and 0.054 dB away from the PID capacity, respectively. By comparison, the quasi-regular rate-0.35 $(3, 4/5)$ LDPC code (C^0) and rate-0.65 $(3, 8/9)$ LDPC code (C^1) have thresholds of 3.44 and 3.35 dB, respectively.

Fig. 10 compares the simulation results for the Gray-mapped 4-PAM MLC/MSD scheme and the MLC/PID scheme on an AWGN channel. The simulation refers to the i.i.d. channel adapter system. In the implementation of the i.i.d. channel adapters, for each binary-input component channel, we used two identical random number generators (RNG) yielding an i.i.d. equiprobable distribution at both the transmitter and receiver sides. We set the same initial seed to each pair of the RNGs. Therefore, each pair of RNGs generate the same random sequence.[7] The codeword length of each component code is 10^6. For the MLC/MSD scheme, we use the irregular LDPC codes optimized for the MLC/MSD scheme with $d_{v_{max}} = 50$ on C^0 and C^1 in one case, and the quasi-regular LDPC codes in another case. For the MLC/PID scheme, we use the irregular LDPC codes optimized for the
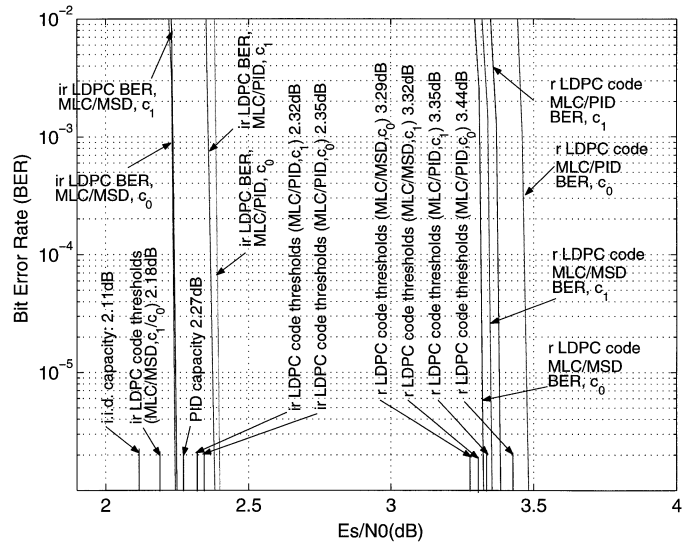

Fig. 10.   Simulation of Gray-mapped MLC/MSD and MLC/PID schemes with 4-PAM modulation on an AWGN channel. The codeword length is 10^6.

MLC/PID scheme with $d_{v_{max}} = 50$ on C^0 and C^1 in one case, and the quasi-regular LDPC codes in another case. However, the component code rates of the MLC/MSD scheme $(R_0/R_1 = 0.338/0.662)$ are slightly different from the MLC/PID scheme $(R_0/R_1 = 0.350/0.650)$. Each BER curve represents one component code. The calculated thresholds for all the component codes are shown, as well as the channel capacity and the PID capacity. We observe that the calculated thresholds accurately predict the performance of both the MLC/MSD and the MLC/PID schemes with long LDPC component codes: for the quasi-regular (resp., optimized irregular) LDPC codes, the E_s/N_0 values at which the BERs are below 10^{-6} are within 0.04 dB (resp., 0.06 dB) of their respective thresholds. The simulation curves of the optimized irregular LDPC codes for the MLC/MSD scheme are better than the PID capacity and very close to the channel capacity as predicted by the threshold results: both codes achieve BERs of 10^{-6} less than 0.14 dB away from the channel capacity. The optimized irregular LDPC codes have a substantial gain ($\sim$1 dB) over the quasi-regular LDPC codes in both the threshold and simulation results. Also, the MLC/MSD scheme performs slightly better than the MLC/PID scheme for both regular and irregular codes, which is consistent with the threshold results as well.

The optimized results for the BICM scheme are shown in Table III. The thresholds of these degree distribution pairs are very close to the PID capacity (about 0.07 dB gap at $d_{v_{max}} = 50$). The threshold of the regular rate-$1/2$ $(3, 6)$ LDPC code is 3.41 dB, more than 1 dB worse than the PID capacity. In Fig. 11, we compare the simulation results for the Gray-mapped BICM

---

[6]The slight difference between the thresholds is due to the step size used for the code rate adjustment.

[7]This method makes the scheme realizable in practical systems.

TABLE III
OPTIMIZED RESULTS FOR THE GRAY-MAPPED BICM SCHEME
(4-PAM MODULATION), $R_s = 1$ BIT/SYMBOL, $r_c = 1/2$, AND THE
PID CAPACITY IS 2.27 dB

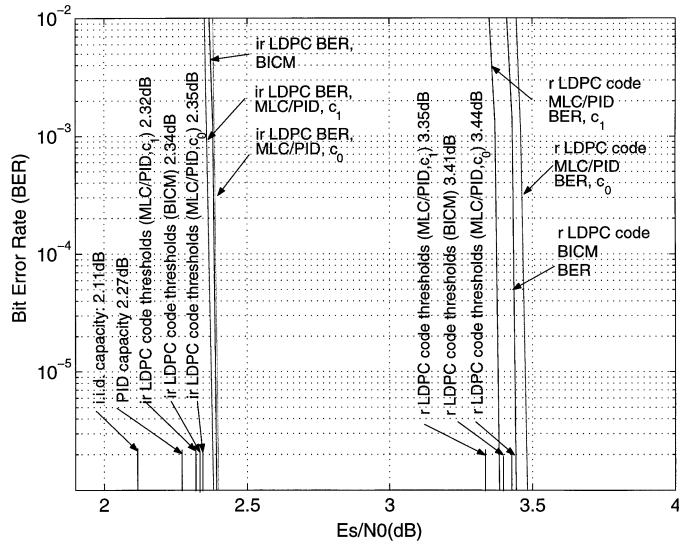| $d_{v_{max}}$	10	15	30	50
$\left(\frac{E_s}{N_0}\right)^* (dB)$	2.527	2.469	2.381	2.345

Fig. 11. Simulation of Gray-mapped MLC/PID and BICM schemes with 4-PAM modulation on an AWGN channel. The codeword length is 10^6.

Fig. 12. Random coding exponent analysis for coded Gray-mapped 4-PAM transmission of 1 bit/symbol.

scheme and the MLC/PID scheme with i.i.d. channel adapters on an AWGN channel. The codeword length of each component code is 10^6 as well. For the BICM scheme, we use both the irregular LDPC code from Table III with $d_{v_{\max}} = 50$ and the regular $(3, 6)$ LDPC code. For the MLC/PID scheme, we use the codes from Fig. 10. Similar to the case of the MLC/MSD and the MLC/PID schemes, for the BICM scheme, the simulated BER curves are very close to the threshold results. It is interesting that in both the regular and irregular cases, the BICM scheme can perform as well as the MLC/PID scheme; however, the decoding complexity and delay are only roughly half that of the MLC/PID scheme.

2) Gray-Mapped 8-PSK: For the Gray-mapped 8-PSK modulation, the target R_s is 2 bits/symbol. First, we consider the MLC/MSD scheme. As we mentioned in Section II, the equivalent transmission model of c_1 if c_0 is known is the same as the equivalent transmission model of c_2 if c_0 and c_1 are known, therefore, $f_0^1 = f_0^2$, where f_0^i is given by (18). Hence, we only need to optimize the degree distribution pairs for the equivalent binary-input channels 0 and 1. The optimized component code rates are $R_0/R_1/R_2 = 0.504/0.748/0.748$ for $d_{v_{\max}} = 50$, in agreement with the results predicted by the capacity calculation. The gap between the thresholds of the optimized degree distribution pairs for level 0 and 1 and the minimum SNR for reliable transmission corresponding to the PID capacity $\hat{C}$ is only about 0.07 dB.

Similarly, the optimized degree distribution pairs for the MLC/PID scheme and the BICM scheme have thresholds very close to the PID capacity, as well.
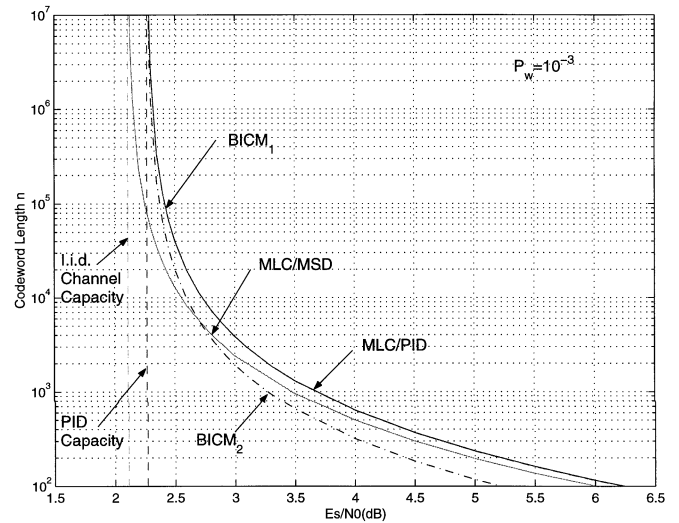
B. Coding Exponent Analysis and Moderate Blocksize Simulations

As shown in Figs. 10 and 11, the thresholds predict the asymptotic performance as the block length of the component LDPC codes approaches infinity. We would also like to analytically compare these power- and bandwidth-efficient schemes based on LDPC codes with a finite block size. However, there are very few accurate analytical tools for analyzing LDPC code performance at finite length. Therefore, we first compare these schemes by the well-known random coding bound technique [4], [20] which could provide a relation between the codeword length n and the required SNR (decibels) for a given word error probability p_w. For the MLC scheme, the analysis can even give the relation between n and component code rate distributions $\{R_i, \ i = 0, 1, \ldots, m - 1\}$. Even though the comparison is based on the average performance of a random block code ensemble, the analysis is still a basis for us to interpret the simulation results based on specific LDPC codes. By the method described in [4], we carry out the coding exponent analysis for the Gray-mapped 4-PAM transmission of $R_s = 1$ bit/symbol on an AWGN channel. The allowable word error probability is $p_w = 10^{-3}$. For all these schemes, the required E_s/N_0 (decibels) versus codeword length n is calculated, and the results are shown in Fig. 12. Note that for both the MLC/MSD and the MLC/PID schemes, the codeword lengths n of binary component codes and of the Euclidean space signal points are equal. For the BICM scheme, BICM$_1$ refers to the case where the codeword length n means the length of binary component code, and BICM$_2$ refers to the case where n means the length of Euclidean space signal points. Note that for the same n, it is fair to compare the BICM$_2$ and the MLC schemes since they have the same (number of information bits) delay. BICM$_1$ has only half of the delay of MLC and we also show its curve for reference purposes. The plot shows that as codeword length goes to infinity, BICM$_1$, BICM$_2$, and MLC/PID all approach the PID capacity, and the MLC/MSD gets very close to the channel capacity. For the MLC/MSD and MLC/PID
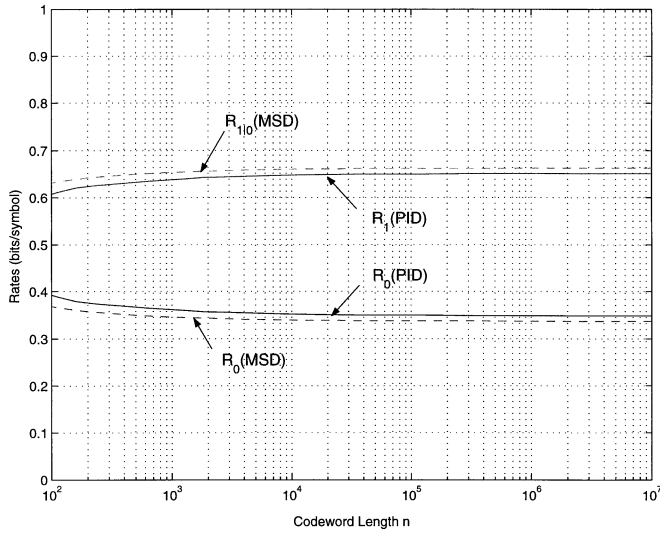
Fig. 13.   Random coding exponent analysis for the MLC/MSD and MLC/PID schemes: component code rate distribution versus codeword length n.

Fig. 14.   Simulation of the Gray-mapped 4-PAM modulated MLC/MSD and MLC/PID schemes on an AWGN channel. The codeword length is 10^4.

schemes, we plot in Fig. 13 the relation between n and $\{R_i\}$. The coding exponent analysis shows that the component code rate distributions for very large block size are virtually the same as those derived by the capacity calculation and those optimized using LDPC component codes. However, the analysis also suggests that for small to moderate block size, the component-code rate distributions are slightly different. Next, we will use Figs. 12 and 13 to explain some finite block-size simulation results based on LDPC component codes.

We construct optimized irregular LDPC codes with $d_{v_{\max}} = 15$ for both the MLC/MSD and MLC/PID schemes. The component codeword length is 10^4. Fig. 14 compares the simulation performance of these two schemes. For comparison, we also plot the simulation results based on quasi-regular LDPC component codes. The results show that the optimized LDPC codes have excellent performance. In general, for each equivalent component channel, the irregular LDPC codes outperform the quasi-regular LDPC codes by about 0.7 at a BER of 10^{-5}. As shown in the threshold results, the simulation performance of the MLC/MSD scheme is better than that of the MLC/PID scheme. For the MLC/MSD scheme, the component code on level $i = 1$ performs slightly worse than the code on level $i = 0$, which suggests that the code rates optimized by density evolution need to be adjusted for a more balanced performance at finite block size and is consistent with the conclusion drawn from the coding exponent analysis. The component code rates of the MLC/MSD scheme used in the simulation are $R_0/R_1 = 0.338/0.662$, as derived from the joint optimization results based on LDPC component codes. However, the coding exponent analysis (Fig. 13) shows that at a block size of 10^4, a better choice of code rates is $R_0/R_1 = 0.340/0.660$.

In Fig. 15, we compare the MLC/PID scheme and the BICM scheme based on optimized irregular LDPC codes with $d_{v_{\max}} = 15$. For the MLC/PID scheme, the component codeword length is 10^4. For the BICM scheme, the component codeword lengths are 10^4 and 2×10^4. The simulated curves based on regular
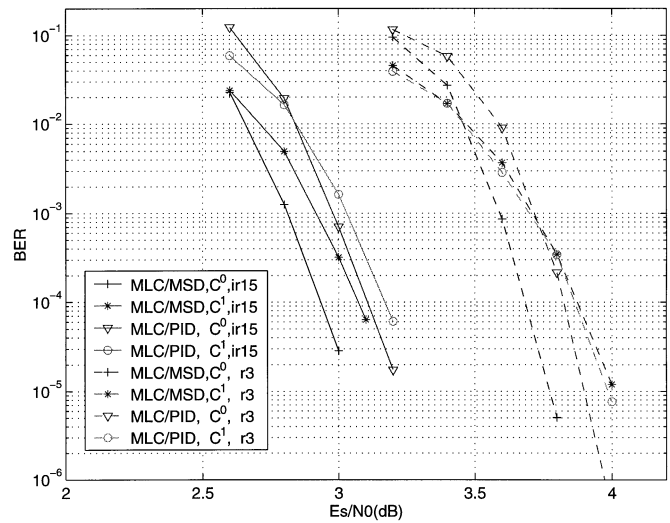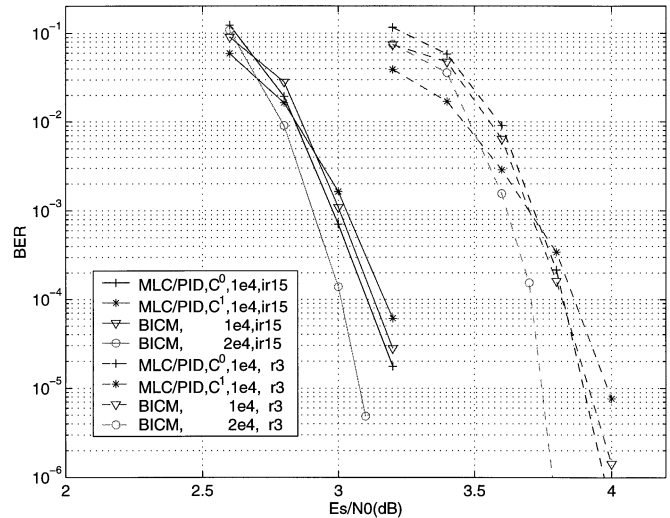
Fig. 15.   Simulation of the Gray-mapped 4-PAM modulated MLC/PID and BICM schemes on an AWGN channel.

LDPC codes are shown as well. By the coding exponent analysis, the performance of $BICM_1$ and MLC/PID is essentially the same. The simulation results shows a very similar trend: the length-10^4 BICM scheme has virtually the same performance as the MLC/PID scheme while having only about half the delay and decoding complexity. On the other hand, the length-2×10^4 BICM scheme, which has roughly the same delay and decoding complexity as the MLC/PID scheme, performs better than the MLC/PID scheme (about 0.1-dB gain at a BER of 10^{-5}). A similar conclusion can also be reached from the coding exponent analysis plot (see Fig. 12).

Another interesting phenomenon reflected in Fig. 12 is that the $BICM_2$ curve and the MLC/MSD curve cross around codeword length $n^* = 6 \times 10^3$, which suggests that for codeword length larger than n^*, MLC/MSD should be better than $BICM_2$; while, for smaller codeword length, $BICM_2$ should be more favorable. At $n = 10^4$, these two schemes should be comparable

as was found in the simulation of MLC/MSD and BICM$_2$ performance, shown in Figs. 14 and 15.

In the moderate block size simulation discussed here, we simulate the MLC and BICM schemes in two ways: only using linear LDPC codes and using linear LDPC codes with i.i.d. channel adapters. In the first approach, we construct the generator matrix and encode the information bits with an i.i.d. equiprobable distribution. In the second approach, it suffices to just transmit the all-zeros codeword. The results show that within the precision of the Monte Carlo simulation, these two approaches have virtually the same performance. For a very large block size simulation, it is difficult to construct the generator matrix, therefore, we do not have simulation results for the first scheme in this situation. However, we conjecture that as the block size goes to infinity, these two approaches should have the same asymptotic performance as well. Therefore, at least for large block size, we can consider the thresholds of the LDPC codes in the MLC and BICM schemes with i.i.d. channel adapters as good approximations to the thresholds of the linear LDPC codes in the MLC and BICM schemes without i.i.d. channel adapters.

VI. CONCLUSION

We designed power- and bandwidth-efficient MLC and BICM schemes for the AWGN channel with binary LDPC component codes. For the decoding of the MLC scheme, we considered both MSD and PID strategies. The design of LDPC codes for these coded modulation schemes essentially was transformed into the design of LDPC codes for the equivalent binary-input component channels. One difficulty encountered in applying the concentration theorem and density evolution technique in the performance analysis of LDPC codes over these equivalent binary-input channels is that these channels are not necessarily output symmetric. To solve this problem, we took two different approaches: one based on i.i.d. channel adapters and the other based on coset codes. We proved that the augmented binary-input channel obtained by incorporating an i.i.d. channel adapter is output symmetric. Thus, the analysis and design of LDPC codes is greatly simplified in this case since it suffices to carry out the analysis for only the all-zeros codeword. Alternatively, by considering LDPC coset codes, we proved that the decoder behavior concentrates around its expected behavior, which corresponds to the average performance of a cycle-free graph with an i.i.d. equiprobable channel input distribution. After discussing the relation between these two approaches, we showed that these two systems indeed have the same expected behavior. Furthermore, based on simulation results, we conjecture that the original system, without either coset codes or i.i.d. channel adapters, has the same expected decoder behavior as the block size goes to infinity as does either of the alternative approaches.

We extended the density evolution technique to determine the noise thresholds of LDPC component codes for both MLC and BICM schemes incorporating i.i.d. channel adapters. It was shown that the computed thresholds accurately predicted the simulation performance of the associated LDPC codes for large block sizes. By joint optimization of the component code rate

distributions and the degree distribution pairs of the MLC/MSD scheme, we showed that both the thresholds and simulation results of the optimized LDPC coded system are very close to the i.i.d. channel capacity. We also showed that the optimized BICM scheme can perform as well as the optimized MLC/PID scheme and that both approach the PID capacity. We compared the simulated performance of these schemes at moderate block sizes, and some conclusions drawn from the simulation were supported by a random coding exponent analysis. At small to moderate block size, BICM seems to be the more attractive alternative, achieving a better balance among complexity, delay, and performance.

APPENDIX A
PROOF OF LEMMA 1

Let $\boldsymbol{y} = [y_1, \ldots, y_n]$ be a noisy channel output realization and $\boldsymbol{u}^i = [u_1^i, \ldots, u_n^i]$ be its associated LAPPR vector, where u_j^i is the LAPPR value computed based on y_j. Consider the second decoding approach described in Section III-A1. If we assume that $\boldsymbol{o}_1^i$ is the coset-defining vector, we have[8] $\boldsymbol{q}_1^i = \boldsymbol{u}^i \cdot \boldsymbol{o}_1^{B,i}$. Similarly, we have $\boldsymbol{q}_2^i = \boldsymbol{u}^i \cdot \boldsymbol{o}_2^{B,i}$. We define

$$\boldsymbol{o}_\triangle^i = \boldsymbol{c}_1^i \oplus \boldsymbol{c}_2^i = \boldsymbol{o}_1^i \oplus \boldsymbol{o}_2^i.$$

Therefore, we have

$$\boldsymbol{q}_1^i = \boldsymbol{q}_2^i \cdot \boldsymbol{o}_1^{B,i} \cdot \boldsymbol{o}_2^{B,i} = \boldsymbol{q}_2^i \cdot \boldsymbol{o}_\triangle^{B,i}.$$

By definition, because $\boldsymbol{c}_1^i$ and $\boldsymbol{c}_2^i$ are two codewords of g, $\boldsymbol{o}_\triangle^i$ is a codeword of g as well. Thus, $\prod_{j:\,\exists e=(v_j, c_\kappa)} o_{\triangle, j}^{B,i} = 1$, where v_j denotes a bit node and c_κ denotes a check node. Let $q_{c,\kappa j}^{i,(l)}[\boldsymbol{q}^i]$ denote the message sent from v_j to c_κ at iteration l, assuming $\boldsymbol{q}^i$ is the decoder input, and let $q_{v,\kappa j}^{i,(l)}[\boldsymbol{q}^i]$ denote the message sent from c_κ to v_j at iteration l, assuming $\boldsymbol{q}^i$ is the decoder input. From the bit node symmetry, at $l = 0$

$$q_{c,\kappa j}^{i,(0)}[\boldsymbol{q}_1^i] = o_{\triangle, j}^{B,i} \cdot q_{c,\kappa j}^{i,(0)}[\boldsymbol{q}_2^i].$$

Assume at iteration l that we have

$$q_{c,\kappa j}^{i,(l)}[\boldsymbol{q}_1^i] = o_{\triangle, j}^{B,i} \cdot q_{c,\kappa j}^{i,(l)}[\boldsymbol{q}_2^i]. \tag{20}$$

Since $\prod_{j:\,\exists e=(v_j, c_\kappa)} o_{\triangle, j}^{B,i} = 1$, from the check node symmetry, we have

$$q_{v,\kappa j}^{i,(l)}[\boldsymbol{q}_1^i] = o_{\triangle, j}^{B,i} \cdot q_{v,\kappa j}^{i,(l)}[\boldsymbol{q}_2^i]. \tag{21}$$

Furthermore, from the bit node symmetry, it follows that at iteration $l + 1$

$$q_{c,\kappa j}^{i,(l+1)}[\boldsymbol{q}_1^i] = o_{\triangle, j}^{B,i} \cdot q_{c,\kappa j}^{i,(l+1)}[\boldsymbol{q}_2^i].$$

Thus, by induction, at any iteration l, (20) and (21) hold. Therefore, $\widehat{\boldsymbol{c}_1^i} = \boldsymbol{o}_\triangle^i \oplus \widehat{\boldsymbol{c}_2^i}$, where $\widehat{\boldsymbol{c}_1^i}$ and $\widehat{\boldsymbol{c}_2^i}$ are decoder decisions on $\boldsymbol{c}_1^i$ and $\boldsymbol{c}_2^i$, respectively. That is, both decoders commit exactly the same number of errors. $\square$

---

[8]$\boldsymbol{o}_1^{B,i} = (-1)^{\boldsymbol{o}_1^i}$. In the sequel of the proof, we will use the same notation for the other variables.

APPENDIX B
PROOF OF THEOREM 3

We can write

$$\mathbf{Pr}\left(\left|\frac{Z^i}{nd_v^i} - p_{\text{eq}}^i\right| \geq \epsilon\right) \leq \mathbf{Pr}\left(\left|\frac{Z^i}{nd_v^i} - \frac{\mathbf{E}[Z^i]}{nd_v^i}\right| \geq \epsilon/2\right)$$
$$+ \mathbf{Pr}\left(\left|\frac{\mathbf{E}[Z^i]}{nd_v^i} - p_{\text{eq}}^i\right| \geq \epsilon/2\right) \quad (22)$$

where $\bar{\mathbf{E}}[Z^i]$ is averaged over all graphs, all input sequences, and all channel realizations. This inequality means that we can separate the proof into two parts. First, we show that the random variable Z^i/nd_v^i is concentrated around its expected value $\mathbf{E}[Z^i]/nd_v^i$, and, second, we prove the convergence of $\mathbf{E}[Z^i]/nd_v^i$ to p_{eq}^i.

We start with the second term on the right-hand side of (22). Let $\mathbf{E}[Z_j^i]$, $j \in [1, nd_v^i]$, denote the expected number of incorrect messages passed along edge $\vec{e}_j$; then by linearity of expectation

$$\mathbf{E}[Z^i] = \sum_{j=1}^{nd_v^i} \mathbf{E}[Z_j^i].$$

We have

$$\bar{\mathbf{E}}[Z_j^i] = \bar{\mathbf{E}}[Z_j^i | T_r = 0]\mathbf{Pr}(T_r = 0) + \bar{\mathbf{E}}[Z_j^i | T_r = 1]\mathbf{Pr}(T_r = 1)$$

where[9]

$$\bar{\mathbf{E}}[Z_j^i | T_r = 0]$$
$$= \sum_{\boldsymbol{s}} \mathbf{E}[Z_j^i | \boldsymbol{s}, T_r = 0]\mathbf{Pr}(\boldsymbol{s} | T_r = 0)$$
$$= \sum_{\boldsymbol{s}} \sum_{\boldsymbol{t}_\kappa^i} \mathbf{E}[Z_j^i | \boldsymbol{t}_\kappa^i, T_r = 0]\mathbf{Pr}(\boldsymbol{t}_\kappa^i | \boldsymbol{s}, T_r = 0)\mathbf{Pr}(\boldsymbol{s})$$
$$= \sum_{\boldsymbol{t}_\kappa^i} \left(\sum_{\boldsymbol{s}} \mathbf{Pr}(\boldsymbol{s})\mathbf{Pr}(\boldsymbol{t}_\kappa^i | \boldsymbol{s}, T_r = 0)\right)\mathbf{E}[Z_j^i | \boldsymbol{t}_\kappa^i, T_r = 0]$$
$$= \sum_{\boldsymbol{t}_\kappa^i} \bar{\mathbf{E}}_{\boldsymbol{S}}[\mathbf{Pr}(\boldsymbol{t}_\kappa^i | \boldsymbol{S}, T_r = 0)]\mathbf{E}[Z_j^i | \boldsymbol{t}_\kappa^i, T_r = 0]$$

where $\mathbf{E}_{\boldsymbol{S}}$ is the expectation over all input sequences. Note that if $\boldsymbol{S}$ is a sequence of i.i.d. equiprobable binary random variables, each binary symbol in the type-defining vector $\boldsymbol{t}$ specified by (12) is equally likely to be zero or one. Thus, all tree-like modified directed neighborhood types are equally probable, i.e., $\forall \boldsymbol{t}_\kappa^i$, $\kappa = 1, \ldots, 2^{h^i(2l)}$, $i = 0, 1, \ldots, m-1$, $\mathbf{E}_{\boldsymbol{S}}[\mathbf{Pr}(\boldsymbol{t}_\kappa^i | \boldsymbol{S}, T_r = 0)] = 2^{-h^i(2l)}$. Next, $\mathbf{E}[Z_j^i | \boldsymbol{t}_\kappa^i, T_r = 0]$ is just $\tau_{\boldsymbol{t}_\kappa^i}$ as defined by (14), therefore,

$$\mathbf{E}[Z_j^i | T_r = 0] = \sum_{\kappa=1}^{2^{h^i(2l)}} 2^{-h^i(2l)}\tau_{\boldsymbol{t}_\kappa^i} = p_{\text{eq}}^i$$

where the last equality comes from (16). Since $\mathbf{Pr}[T_r = 1] \leq \frac{\gamma}{n}$ for some positive number γ, and $\bar{\mathbf{E}}[Z_j^i | T_r = 1] \leq 1$, we have

$$\left(1 - \frac{\gamma}{n}\right)p_{\text{eq}}^i \leq \bar{\mathbf{E}}[Z_j^i] \leq p_{\text{eq}}^i + \frac{\gamma}{n}.$$

[9]Here $\mathbf{Pr}(\boldsymbol{s} | T_r = 0) = \mathbf{Pr}(\boldsymbol{s})$ since the input sequence is independent of whether or not the neighborhood is tree-like. Also,

$$\mathbf{E}[Z_j^i | \boldsymbol{t}_\kappa^i, \boldsymbol{s}, T_r = 0] = \mathbf{E}[Z_j^i | \boldsymbol{t}_\kappa^i, T_r = 0]$$

since Z_j^i is independent of input sequence $\boldsymbol{s}$ if the type $\boldsymbol{t}$ is known.

Since $\mathbf{E}[Z^i] = \sum_{j=1}^{nd_v^i} \mathbf{E}[Z_j^i]$, we have

$$\left(1 - \frac{\gamma}{n}\right)p_{\text{eq}}^i \leq \frac{\mathbf{E}[Z^i]}{nd_v^i} \leq p_{\text{eq}}^i + \frac{\gamma}{n}. \quad (23)$$

Therefore, we get $|\frac{\mathbf{E}[Z^i]}{nd_v^i} - p_{\text{eq}}^i| < \frac{\gamma}{n}$. If $n > \frac{2\gamma}{\epsilon}$, we have

$$\mathbf{Pr}\left(\left|\frac{\mathbf{E}[Z^i]}{nd_v^i} - p_{\text{eq}}^i\right| \geq \epsilon/2\right) = 0. \quad (24)$$

Now we consider the first term on the right-hand side of (22). Recall that Z^i denotes the number of incorrect messages among all nd_v^i bit-to-check node messages passed in the lth iteration for a $(G, \boldsymbol{S}, \boldsymbol{R})$, where G is a graph in the ensemble $C^n(d_v^i, d_c^i)$, $\boldsymbol{S}$ is an input sequence, and $\boldsymbol{R}$ is a channel noise vector. Similar to [1], we define partial equalities on $(G, \boldsymbol{S}, \boldsymbol{R})$ based on information revealing. Define $M = d_v^i n + mn + n$. Let $=_t$, $0 \leq t \leq M$ be a sequence of equivalence relations, where the first $d_v^i n$ steps of partial equivalence correspond to edge revealing on the graph as specified in [1], the next mn steps correspond to symbol revealing of the input sequence, and the last n steps correspond to the revealing of the channel realizations which is also described in [1]. More specifically, we have $(G', \boldsymbol{S}', \boldsymbol{R}') =_t (G'', \boldsymbol{S}'', \boldsymbol{R}'')$ if and only if the information revealed in the first t steps for both $(G', \boldsymbol{S}', \boldsymbol{R}')$ and $(G'', \boldsymbol{S}'', \boldsymbol{R}'')$ is the same. By the definition, $(G', \boldsymbol{S}', \boldsymbol{R}') =_t (G'', \boldsymbol{S}'', \boldsymbol{R}'')$ means $(G', \boldsymbol{S}', \boldsymbol{R}') =_{t-1} (G'', \boldsymbol{S}'', \boldsymbol{R}'')$.

Next, we define a Doob's martingale process $P_0, P_1, \ldots, P_M$ by

$$P_t = \mathbf{E}\big[Z^i(G', \boldsymbol{S}', \boldsymbol{R}')\big|(G', \boldsymbol{S}', \boldsymbol{R}') =_t (G, \boldsymbol{S}, \boldsymbol{R})\big],$$
$$t = 0, 1, \ldots, M \quad (25)$$

where P_t represents the expected number of incorrect messages conditioned on the information revealed in the first t steps, assuming that the information revealed is consistent with $(G, \boldsymbol{S}, \boldsymbol{R})$, which implies[10] $P_0 = \mathbf{E}[Z^i]$ and $P_M = Z^i$. Applying Azuma's inequality as in [1], [21] we have

$$\mathbf{Pr}\left(\left|\frac{Z^i}{nd_v^i} - \frac{\mathbf{E}[Z_i]}{nd_v^i}\right| \geq \epsilon/2\right)$$
$$= \mathbf{Pr}(|P_M - P_0| \geq nd_v^i\epsilon/2) \leq 2e^{-\frac{(nd_v^i\epsilon/2)^2}{2\sum_{t=0}^{M-1}\delta_t^2}} \quad (26)$$

if we can prove that

$$|P_{t+1} - P_t| \leq \delta_t, \qquad t = 0, 1, \ldots, M-1 \quad (27)$$

where δ_t may depend on d_v^i, d_c^i, and l, but not on n.

The first $d_v^i n$ steps are edge revealing steps. Therefore, as in [1], for $t \in [0, nd_v^i)$, we conclude that (27) holds with $\delta_t = 8(d_v^i d_c^i)^l$. The last n steps correspond to the revealing of the channel realizations, so, again referring to [1], for $t \in [nd_v^i + mn, M)$, we can use $\delta_t = 2(d_v^i d_c^i)^l$.

Now, we consider the steps of symbol revealing on the i.i.d. input sequence, i.e., $t \in [nd_v^i, nd_v^i + mn)$. Let $t_\triangle = t - nd_v^i$. For a given length-mn binary sequence $\boldsymbol{s}_1$, we define sequence $\bar{\boldsymbol{s}}_1$ as follows: $(\bar{\boldsymbol{s}}_1)_1^{t_\triangle} = (\boldsymbol{s}_1)_1^{t_\triangle}$, $(\bar{\boldsymbol{s}}_1)_{t_\triangle+2}^{mn} = (\boldsymbol{s}_1)_{t_\triangle+2}^{mn}$, and $(\bar{\boldsymbol{s}}_1)_{t_\triangle+1} \oplus (\boldsymbol{s}_1)_{t_\triangle+1} = 1$. In other words, $\bar{\boldsymbol{s}}_1$ differs from $\boldsymbol{s}_1$ only at position $t_\triangle+1$. From this definition, we have $\boldsymbol{s}_1 =_{t_\triangle} \bar{\boldsymbol{s}}_1$ but $\boldsymbol{s}_1 \neq_{t_\triangle+1} \bar{\boldsymbol{s}}_1$. Referring to the modified directed neighborhood definition, we recall that each binary symbol belongs to

[10]$\mathbf{E}[Z^i]$ is averaged over all graphs, all input sequences, and all noise realizations as we described before.

a channel node. Therefore, if $\overline{\boldsymbol{s}}_1$ differs from $\boldsymbol{s}_1$ only at position $t_\triangle + 1$, this binary symbol affects only two kinds of message flow: the first kind is the message flow within the modified directed neighborhoods to which this binary symbol's corresponding channel node belongs. As to the second kind of message flow, note that this binary symbol is involved in d_c^i checks of the parity-check matrix if this binary symbol belongs to the component code at level i. Therefore, complementing this binary symbol also affects the message flow within the modified directed neighborhoods to which those involved checks connect. By (11), the number of channel nodes in a modified directed neighborhood is upper-bounded by $(d_v^i d_c^i)^l$. By the node symmetry property [1], the number of neighborhoods to which a channel node belongs is upper-bounded by $(d_v^i d_c^i)^l$. By similar reasoning, the number of neighborhoods to which those involved checks connect is upper-bounded by $d_c^i (d_v^i d_c^i)^l$. Therefore, in total, complementing one binary symbol in a sequence $\boldsymbol{s}_1$, the number of neighborhoods within which message flow is affected is upper-bounded by $(1 + d_c^i)(d_v^i d_c^i)^l$, i.e., for any given g, $\boldsymbol{s}_1$, and $\boldsymbol{r}$

$$|Z^i(g, \boldsymbol{s}_1, \boldsymbol{r}) - Z^i(\boldsymbol{g}, \overline{\boldsymbol{s}_1}, \boldsymbol{r})| \le (1 + d_c^i)(d_v^i d_c^i)^l. \quad (28)$$

From the definition of P_t, for $t \in [nd_v^i, nd_v^i + mn)$, we have

$$|P_{t+1} - P_t|$$
$$\le \max_{b_1 \ne b_2, \, b_1, b_2 \in \{0, 1\}}$$
$$|\boldsymbol{E}[Z^i(G, \boldsymbol{S}_1, \boldsymbol{R}')|G, \boldsymbol{S}_1 =_{t_\triangle} \boldsymbol{S}, (\boldsymbol{S}_1)_{t_\triangle + 1} = b_1]$$
$$- \boldsymbol{E}[Z^i(G, \boldsymbol{S}_2, \boldsymbol{R}')|G, \boldsymbol{S}_2 =_{t_\triangle} \boldsymbol{S}, (\boldsymbol{S}_2)_{t_\triangle + 1} = b_2]|.$$

Using (28), along with the fact that for any random variable

$$|\boldsymbol{E}[W]| \le \boldsymbol{E}[|W|]$$

we can conclude that for any $b_1 \ne b_2$, $b_1, b_2 \in \{0, 1\}$

$$|\boldsymbol{E}[Z^i(G, \boldsymbol{S}_1, \boldsymbol{R}')|G, \boldsymbol{S}_1 =_{t_\triangle} \boldsymbol{S}, (\boldsymbol{S}_1)_{t_\triangle + 1} = b_1]$$
$$- \boldsymbol{E}[Z^i(G, \boldsymbol{S}_2, \boldsymbol{R}')|G, \boldsymbol{S}_2 =_{t_\triangle} \boldsymbol{S}, (\boldsymbol{S}_2)_{t_\triangle + 1} = b_2]|$$
$$= |\boldsymbol{E}[Z^i(G, \boldsymbol{S}_1, \boldsymbol{R}')|G, \boldsymbol{S}_1 =_{t_\triangle} \boldsymbol{S}, (\boldsymbol{S}_1)_{t_\triangle + 1} = b_1]$$
$$- \boldsymbol{E}[Z^i(G, \overline{\boldsymbol{S}_2}, \boldsymbol{R}')|G, \boldsymbol{S}_2 =_{t_\triangle} \boldsymbol{S}, (\boldsymbol{S}_2)_{t_\triangle + 1} = b_2]$$
$$+ \boldsymbol{E}[Z^i(G, \overline{\boldsymbol{S}_2}, \boldsymbol{R}')|G, \boldsymbol{S}_2 =_{t_\triangle} \boldsymbol{S}, (\boldsymbol{S}_2)_{t_\triangle + 1} = b_2]$$
$$- \boldsymbol{E}[Z^i(G, \boldsymbol{S}_2, \boldsymbol{R}')|G, \boldsymbol{S}_2 =_{t_\triangle} \boldsymbol{S}, (\boldsymbol{S}_2)_{t_\triangle + 1} = b_2]|$$
$$= |\boldsymbol{E}[Z^i(G, \overline{\boldsymbol{S}_2}, \boldsymbol{R}')|G, \boldsymbol{S}_2 =_{t_\triangle} \boldsymbol{S}, (\boldsymbol{S}_2)_{t_\triangle + 1} = b_2]$$
$$- \boldsymbol{E}[Z^i(G, \boldsymbol{S}_2, \boldsymbol{R}')|G, \boldsymbol{S}_2 =_{t_\triangle} \boldsymbol{S}, (\boldsymbol{S}_2)_{t_\triangle + 1} = b_2]|$$
$$\le \boldsymbol{E}[|Z^i(G, \overline{\boldsymbol{S}}_2, \boldsymbol{R}') - Z^i(G, \boldsymbol{S}_2, \boldsymbol{R}')||G,$$
$$\boldsymbol{S}_2 =_{t_\triangle} \boldsymbol{S}, (\boldsymbol{S}_2)_{t_\triangle + 1} = b_2] \le (1 + d_c^i)(d_v^i d_c^i)^l.$$

That is,

$$\delta_t = (1 + d_c^i)(d_v^i d_c^i)^l, \qquad \text{for } t \in [nd_v^i, nd_v^i + mn).$$

Therefore, for any $t \in [0, M - 1)$, if we set

$$\delta_t = (\max(8, 1 + d_c^i)) \cdot (d_v^i d_c^i)^l$$

and choose

$$\beta = \frac{(d_v^i)^2}{8(d_v^i + m + 1)\delta_t^2}$$

then Theorem 3 follows from (22) and (26). $\qquad \square$

REFERENCES

[1] T. J. Richardson and R. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, pp. 599–618, Feb. 2001.

[2] A. Kavčić, X. Ma, and M. Mitzenamcher, "Binary intersymbol interference channels: Gallager codes, density evolution, and code performance bounds," *IEEE Trans. Inform. Theory*, vol. 49, pp. 1636–1652, July 2003.

[3] H. Imai and S. Hirakawa, "A new multilevel coding method using error correcting codes," *IEEE Trans. Inform. Theory*, vol. IT-23, pp. 371–377, May 1977.

[4] U. Wachsmann, R. F. H. Fischer, and J. B. Huber, "Multilevel codes: Theoretical concepts and practical design rules," *IEEE Trans. Inform. Theory*, vol. 45, pp. 1361–1391, July 1999.

[5] E. Zehavi, "8-PSK trellis codes for a Rayleigh channel," *IEEE Trans. Commun.*, vol. 40, pp. 873–884, May 1992.

[6] G. Caire, G. Taricco, and E. Biglieri, "Bit-interleaved coded modulation," *IEEE Trans. Inform. Theory*, vol. 44, pp. 927–946, May 1998.

[7] A. R. Calderbank, "Multilevel codes and multistage decoding," *IEEE Trans. Commun.*, vol. 37, pp. 222–229, Mar. 1989.

[8] U. Wachsmann, "Coded modulation: Theoretical concepts and practical design Rules," Ph.D. dissertation, Univ. Erlangen-Nurnberg, Erlangen, Germany, May 1998.

[9] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo codes," in *Proc. IEEE Int. Conf. Communications '93*, Geneva, Switzerland, May 1993, pp. 1064–1070.

[10] R. G. Gallager, *Low-Density Parity-Check Codes*. Cambridge, MA: MIT Press, 1963.

[11] T. J. Richardson, A. Shokrollahi, and R. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 47, pp. 619–637, Feb, 2001.

[12] S. Y. Chung, G. D. 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, pp. 58–60, Feb. 2001.

[13] J. Hou, P. Siegel, and L. Milstein, "Performance analysis and code optimization of low density parity-check codes on Rayleigh fading channels," in *Proc. 38th Annu. Allerton Conf. Communication, Control, and Computing*, Monticello, IL, Nov. 2000, pp. 266–275.

[14] J. Hou, P. Siegel, L. Milstein, and H. D. Pfister, "Design of low-density parity-check codes for bandwidth efficient modulation," in *Proc. IEEE Information Theory Workshop (ITW'01)*, Cairns, Australia, Sept. 2001, pp. 24–26.

[15] ——, "Multilevel coding with low-density parity-check component codes," in *Proc. IEEE Global Telecommunications Conf. (GLOBECOM'01)*, San Antonio, TX, Nov. 2001, pp. 1016–1020.

[16] X. Li and J. A. Ritcey, "Bit-interleaved coded modulation with iterative decoding," *IEEE Commun. Lett.*, vol. 1, pp. 169–171, Nov. 1997.

[17] K. R. Narayanan and J. Li, "Bandwidth efficient low density parity check coding using multi level coding and iterative multi stage decoding," in *Proc. 2nd Turbo Symp. Turbo Codes and Related Topics*, Brest, France, July 2000, pp. 165–168.

[18] R. Narayanaswami, "Coded modulation with low density parity check codes," Master thesis, Texas A&M Univ., College Station, June 2001.

[19] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 55–67, Jan. 1982.

[20] R. G. Gallager, *Information Theory and Reliable Communication*. New York: Wiley, 1968.

[21] R. Motwani and P. Raghavan, *Randomized Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 1995.

[22] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic adaptive scheme for global optimization over continuous spaces," *J. Global Optimiz.*, vol. 11, pp. 341–359, 1997.