Design of LDPC Codes: A Survey and New Results

Gianluigi Liva, Shumei Song, Lan Lan, Yifei Zhang, Shu Lin, and William E. Ryan

Abstract— **This survey paper provides fundamentals in the design of LDPC codes. To provide a target for the code designer, we first summarize the EXIT chart technique for determining (near-)optimal degree distributions for LDPC code ensembles. We also demonstrate the simplicity of representing codes by protographs and how this naturally leads to quasi-cyclic LDPC codes. The EXIT chart technique is then extended to the special case of protograph-based LDPC codes. Next, we present several design approaches for LDPC codes which incorporate one or more accumulators, including quasi-cyclic accumulator-based codes. The second half the paper then surveys several algebraic LDPC code design techniques. First, codes based on finite geometries are discussed and then codes whose designs are based on Reed-Solomon codes are covered. The algebraic designs lead to cyclic, quasi-cyclic, and structured codes. The masking technique for converting regular quasi-cyclic LDPC codes to irregular codes is also presented. Some of these results and codes have not been presented elsewhere. The paper focuses on the binary-input AWGN channel (BI-AWGNC). However, as discussed in the paper, good BI-AWGNC codes tend to be universally good across many channels. Alternatively, the reader may treat this paper as a starting point for extensions to more advanced channels. The paper concludes with a brief discussion of open problems.**

I. Introduction

The class of low-density parity-check (LDPC) codes represents the leading edge in modern channel coding. They have held the attention of coding theorists and practitioners in the past decade because of their near-capacity performance on a large variety of data transmission and storage channels and because their decoders can be implemented with manageable complexity. They were invented by Gallager in his 1960 doctoral dissertation [1] and were scarcely considered in the 35 years that followed. One notable exception is Tanner, who wrote an important paper in 1981 [2] which generalized LDPC codes and introduced a graphical representation of LDPC codes, now called Tanner graphs. Apparently independent of Gallager's work, LDPC codes were re-invented in the mid-1990's by MacKay, Luby, and others [3][4][5][6] who noticed the advantages of linear block codes which possess sparse (low-density) parity-check matrices.

This papers surveys the state-of-the-art in LDPC code design for binary-input channels while including a few new results as well. While it is tutorial in some aspects, it is not entirely a tutorial paper, and the reader is expected to be fairly versed on the topic of LDPC codes. Tutorial coverages of LDPC codes can be found in [7][8]. The purpose of this paper is to give the reader a detailed overview of various LDPC code design approaches and also to point the reader to the literature. While our emphasis is on code design for the binary-input AWGN channel (BI-AWGNC), the results in [9][10][11][12] demonstrate that a LDPC code that is good on the BI-AWGNC tends to be universally good and can be expected to be good on most wireless, optical, and storage channels.

We favor code designs which are most appropriate for applications, by which we mean codes which have low-complexity encoding, good waterfall regions, and low error floors. Thus, we discuss quasi-cyclic (QC) codes because their encoders may be implemented by shift-register circuits [13]. We also discuss accumulator-based codes because low-complexity encoding is possible from their parity-check matrices, whether they are quasi-cyclic or not. The code classes discussed tend to be the ones (or related to the ones) used in applications or adopted for standards. Due to time and space limitations, we cannot provide a complete survey. The present survey is biased toward the expertise and interests of the authors.

Before a code can be designed, the code designer needs to know the design target. For this reason, Section II first briefly reviews the belief propagation decoder for LDPC codes and then presents the so-called extrinsic information transfer (EXIT) chart technique for this decoder. The EXIT chart technique allows one to obtain near-optimal parameters for LDPC code ensembles which guide the code designer. The EXIT technique is extended in Section III to the case of codes based on protographs. Section IV considers LDPC codes based on accumulators. The code types treated in that section are: repeat-accumulate, irregular repeat-accumulate, irregular repeat-accumulate-accumulate, generalized irregular repeat-accumulate, and accumulate-repeat-accumulate. That section also gives examples of quasi-cyclic code design using protograph (or base matrix) representations. Section V surveys the literature on cyclic and quasi-cyclic LDPC code design based on finite geometries. Section VI presents several LDPC code design techniques based on Reed-Solomon codes. Section VII presents the masking technique for converting regular QC codes to irregular QC codes to conform to prescribed code parameters. Section VIII contains some concluding remarks and some open problems.

II. Design via EXIT Charts

We start with an $m \times n$ low-density parity-check matrix $\mathbf{H}$, which corresponds to a code with design rate $(n-m)/n$,

Manuscript received July 04, 2006; revised August 25, 2006. This work was supported by the University of Bologna, NASA-Goddard, and NSF.

This paper has been approved by F. Chiaraluce.

Gianluigi Liva is with the University of Bologna (email: gliva@deis.unibo.it).

Shumei Song, Lan Lan, and Shu Lin are with the University of California at Davis (e-mail: ssmsong@ece.ucdavis.edu, squashlan@gmail.com, shulin@ece.ucdavis.edu).

Yifei Zhang and William E. Ryan are with the University of Arizona, U.S.A. (e-mail: {yifeiz, ryan}@ece.arizona.edu).

1845-6421/06/6101 © 2006 CCIS

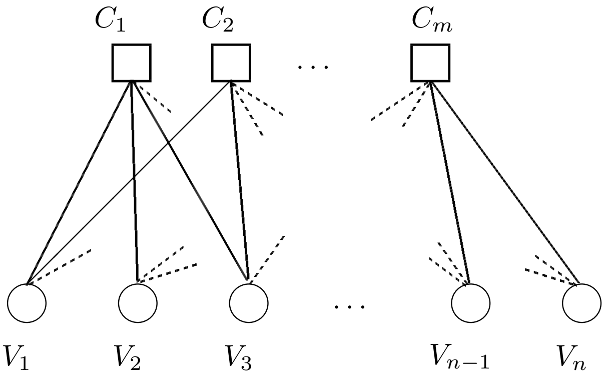


Fig. 1. Tanner graph representation of LDPC codes.

which could be less than the actual rate, $R = k/n$, where k is the number of information bits per codeword. $\mathbf{H}$ gives rise to a Tanner graph which has m check nodes, one for each row of $\mathbf{H}$, and n variable nodes, one for each column of $\mathbf{H}$. Considering the general case in which $\mathbf{H}$ has non-uniform row and column weight, the Tanner graph can be characterized by degree assignments $\{d_v(i)\}_{i=1}^{n}$ and $\{d_c(j)\}_{j=1}^{m}$, where $d_v(i)$ is the degree of the i-th variable node and $d_c(j)$ is the degree of the j-th check node. Such a graph, depicted in Fig. 1, is representative of the iterative decoder, with each node representing a soft-in/soft-out processor (or node decoder).

We shall assume the BI-AWGNC in our description of the LDPC iterative decoder. In this model, a received channel sample y is given by $y = x + w$, where $x = (-1)^c \in \{\pm 1\}$ is the bipolar representation of the transmitted code bit $c \in \{0, 1\}$ and w is a white Gaussian noise sample distributed as $\eta\left(0, \sigma_w^2\right)$, where $\sigma_w^2 = N_0/2$, following convention. The channel bit log-likelihood ratios (LLRs) are computed as

$$L_{ch} = \log\left(\frac{p\left(x = +1 \mid y\right)}{p\left(x = -1 \mid y\right)}\right) = \frac{2y}{\sigma_w^2}. \qquad (1)$$

In one iteration of the conventional, flooding-schedule iterative decoder, the variable node decoders (VNDs) first process their input LLRs and send the computed outputs (messages) to each of their neighboring check node decoders (CNDs); then the CNDs process their input LLRs and send the computed outputs (messages) to each of their neighboring VNDs. More specifically, the message from the i-th VND to the j-th CND is

$$L_{i\to j} = L_{ch,i} + \sum_{j'\neq j} L_{j'\to i} \qquad (2)$$

where $L_{j'\to i}$ is the incoming message from CND j' to VND i and where the summation is over the $d_v(i) - 1$ check node neighbors of variable node i, excluding check node j. The message from CND j to VND i is given by

$$L_{j\to i} = 2\tanh^{-1}\left(\prod_{i'\neq i} \tanh\left(L_{i'\to j}\right)\right) \qquad (3)$$

where $L_{i'\to j}$ is the incoming message from VND i' to CND j and where the product is over the $d_c(j) - 1$ variable node neighbors of check node j, excluding variable node i . This decoding algorithm is called the sum-product algorithm (SPA).

We now discuss the EXIT chart technique [14][15][11] for this decoder and channel model. The idea is that the VNDs and the CNDs work cooperatively and iteratively to make bit decisions, with the metric of interest generally improving with each half-iteration. A transfer curve which plots the input metric versus the output metric can be obtained for both the VNDs and the CNDs, where the transfer curve for the VNDs depends on the channel SNR. Further, since the output metric for one processor is the input metric for its companion processor, one can plot both transfer curves on the same axes, but with the abscissa and ordinate reversed for one processor. Such a chart aids in the prediction of the *decoding threshold* of the ensemble of codes characterized by given VN and CN degree distributions: the decoding threshold is the SNR at which the two transfer curves just touch, precluding convergence of the two processors. EXIT chart computations are thus integral to the optimization of Tanner graph node degree distributions for LDPC codes and are the main computation in the optimization process. We emphasize that decoding threshold prediction techniques such as EXIT charts or density evolution [16] assume a graph with no cycles, an infinite codeword length, and an infinite number of decoding iterations.

An EXIT chart example is depicted in Fig. 2 for the ensemble of regular LDPC codes on the BI-AWGNC with $d_v(i) = d_v = 3$ for $i = 1, ..., n$, and $d_c(j) = d_c = 6$ for $j = 1, ..., m$. In the figure, the metric used for the transfer curves is extrinsic mutual information, giving rise to the name extrinsic information transfer (EXIT) chart. (The notation used in the figure is explained below.) Also shown in the figure is the decoding trajectory corresponding to these EXIT curves. As the SNR increases, the top curve shifts upwards, increasing the "tunnel" between the two curves and thus the decoder convergence rate. The SNR for this figure is just above the decoding threshold for codes with $(d_v, d_c) = (3, 6)$, $(E_b/N_0)_{thres} = 1.1$ dB. Other metrics, such as SNR and mean [17][18] and error probability [19] are possible, but mutual information generally gives the most accurate prediction of the decoding threshold [14][20] and is a universally good metric across many channels [9][10][11][12].

To facilitate EXIT chart computations, the following Gaussian assumption is made. First, we note that the LLR L_{ch} in (1) corresponding to the BI-AWGNC is Gaussian with mean $\mu_{ch} = 2x/\sigma_w^2$ and variance $\sigma_{ch}^2 = 4/\sigma_w^2$. From this and the usual assumption that the all-zeros codeword was transmitted (thus, $x_i = +1$ for $i = 1, ..., n$), $\sigma_{ch}^2 = 2\mu_{ch}$. This is equivalent to the *symmetric condition* of [16] which states that the conditional pdf of an LLR value L must satisfy $p_L\left(l \mid x\right) = p_L\left(-l \mid x\right) e^{xl}$. Now, it has been observed that under normal operating conditions and after a few iterations, the LLRs $L_{i\to j}$ and $L_{j\to i}$ are approximately Gaussian and, further, if they are assumed to be symmetric-Gaussian, as is the case for L_{ch}, the decoding threshold predictions are very accurate (e.g., when compared to the more accurate, but more computationally intensive density evolution results [16]). Moreover, the symmetric-Gaussian assumption vastly simplifies EXIT chart analyses.

We now consider the computation of EXIT transfer curves

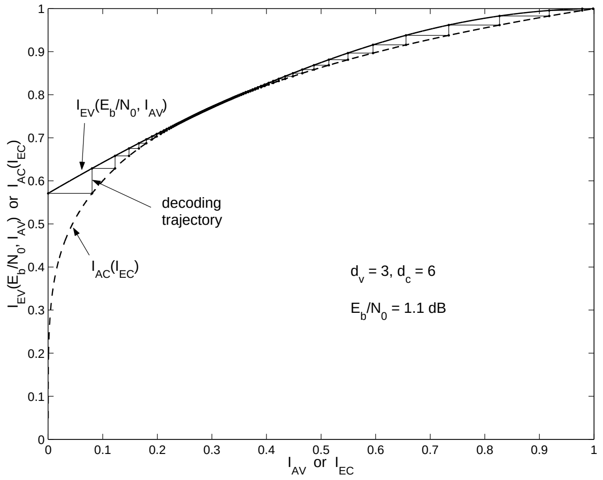


Fig. 2. EXIT chart example for $(d_v, d_c) = (3, 6)$ regular LDPC code.

for both VNDs and the CNDs, first for regular LDPC codes and then for irregular codes. Following [14][15], excluding the inputs from the channel, we consider VND and CND inputs to be *a priori* information, designated by ‘A’, and their outputs to be extrinsic information, designated by ‘E’. Thus, an extrinsic information transfer curve for the VNDs plots the extrinsic information I_E as a function of its input *a priori* information, I_A, and similarly for the CNDs.

The VND EXIT curve, $I_{E,V}$ versus $I_{A,V}$, under the symmetric-Gaussian assumption for VND inputs, $L_{ch,i}$ and $\{L_{j'\to i}\}$, and outputs, $L_{i\to j}$, can be obtained as follows. From (2) and an independent-message assumption, $L_{i\to j}$ is Gaussian with variance $\sigma^2 = \sigma_{ch}^2 + (d_v - 1)\,\sigma_A^2$ (hence, mean $\sigma^2/2$). The mutual information between the random variable X (corresponding to the realization x_i) and the extrinsic LLR $L_{i\to j}$ is therefore (for simplicity, we write L for $L_{i\to j}$, x for x_i, and $p_L(l \mid \pm)$ for $p_L(l \mid x = \pm 1)$)

$$\begin{aligned} I_{E,V} &= H(X) - H(X \mid L) \\ &= 1 - E\left[\log_2\left(1/p_{X\mid L}(x \mid l)\right)\right] \\ &= 1 - \sum_{x=\pm 1} \frac{1}{2}\int_{-\infty}^{\infty} p_L(l \mid x) \\ &\qquad \cdot \log_2\left(\frac{p_L(l \mid +) + p_L(l \mid -)}{p_L(l \mid x)}\right) dl \\ &= 1 - \int_{-\infty}^{\infty} p_L(l \mid +) \log\left(1 + \frac{p_L(l \mid -)}{p_L(l \mid +)}\right) dl \\ &= 1 - \int_{-\infty}^{\infty} p_L(l \mid +) \log\left(1 + e^{-l}\right) dl \end{aligned}$$

where the last line follows from the symmetry condition and because $p_L(l \mid x = -1) = p_L(-l \mid x = +1)$ for Gaussian densities.

Since $L_{i\to j} \sim \eta\left(\sigma^2/2, \sigma^2\right)$ (when conditioned on $x_i = +1$), we have

$$I_{E,V} = 1 - \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-\left(l-\sigma^2/2\right)^2/2\sigma^2} \log\left(1 + e^{-l}\right) dl\ . \tag{4}$$

For convenience we write this as

$$I_{E,V} = J(\sigma) = J\left(\sqrt{(d_v - 1)\,\sigma_A^2 + \sigma_{ch}^2}\right)\ , \tag{5}$$

following [15]. To plot $I_{E,V}$ versus $I_{A,V}$, where $I_{A,V}$ is the mutual information between the VND inputs $L_{j\to i}$ and the channel bits x_i, we apply the symmetric-Gaussian assumption to these inputs so that

$$I_{A,V} = J(\sigma_A) \tag{6}$$

and

$$I_{E,V} = J(\sigma) = J\left(\sqrt{(d_v - 1)\left[J^{-1}(I_{A,V})\right]^2 + \sigma_{ch}^2}\right)\ . \tag{7}$$

The inverse function $J^{-1}(\cdot)$ exists since $J(\sigma_A)$ is monotonic in σ_A. Lastly, $I_{E,V}$ can be parameterized by E_b/N_0 for a given code rate R since $\sigma_{ch}^2 = 4/\sigma_w^2 = 8R\,(E_b/N_0)$. Approximations of the functions $J(\cdot)$ and $J^{-1}(\cdot)$ are given in [15].

To obtain the CND EXIT curve, $I_{E,C}$ versus $I_{A,C}$, we can proceed as we did in the VND case, e.g., begin with the symmetric-Gaussian assumption. However, this assumption is not sufficient because determining the mean and variance for a CND output $L_{j\to i}$ is not straightforward, as is evident from the computation for CNDs in (3). Closed-form expressions have been derived for the check node EXIT curves [21][22]. Computer-based numerical techniques can also be used to obtain these curves. However, the simplest technique exploits the following duality relationship (proven to be exact for the binary erasure channel [11]): the EXIT curve for a degree-d_c check node (i.e., rate-$(d_c - 1)/d_c$ single-parity check (SPC) code) and that of a degree-d_c variable node (i.e., rate-$1/d_c$ repetition code) are related as

$$I_{E,SPC}(d_c, I_A) = 1 - I_{E,REP}(d_c, 1 - I_A)\,.$$

This relationship was shown to be very accurate for the BI-AWGNC in [21][22]. Thus,

$$\begin{aligned} I_{E,C} &= 1 - I_{E,V}\left(\sigma_{ch} = 0, d_v \leftarrow d_c, I_{A,V} \leftarrow 1 - I_{A,C}\right) \\ &= 1 - J\left(\sqrt{(d_c - 1)\left[J^{-1}(1 - I_{A,C})\right]^2}\right). \end{aligned} \tag{8}$$

For irregular LDPC codes, $I_{E,V}$ and $I_{E,C}$ are computed as weighted averages. The weighting is given by the coefficients of the ”edge perspective” degree distribution polynomials $\lambda(z) = \sum_{d=1}^{d_v} \lambda_d z^{d-1}$ and $\rho(z) = \sum_{d=1}^{d_c} \rho_d z^{d-1}$, where λ_d is the fraction of edges in the Tanner graph connected to degree-d variable nodes, ρ_d is the fraction of edges connected to degree-d check nodes, and $\lambda(1) = \rho(1) = 1$. Then, for irregular LDPC codes,

$$I_{E,V} = \sum_{d=1}^{d_v} \lambda_d I_{E,V}(d, I_{A,V}) \tag{9}$$

where $I_{E,V}(d)$ is given by (7) with d_v replaced by d, and

$$I_{E,C} = \sum_{d=1}^{d_c} \rho_d I_{E,C}(d, I_{A,C}) \tag{10}$$

where $I_{E,C}(d)$ is given by (8) with d_c replaced by d.

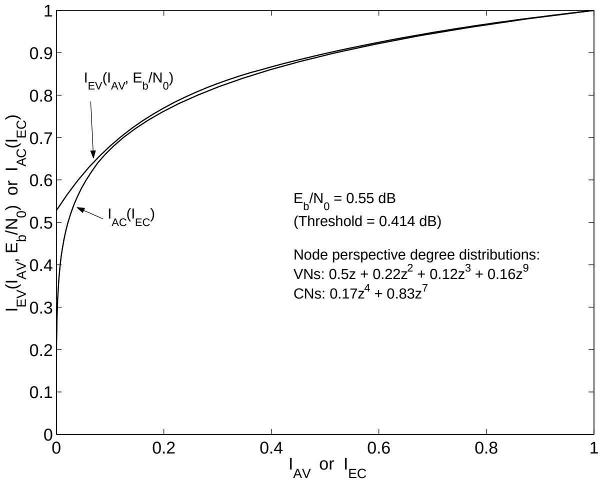


Fig. 3. EXIT chart for rate-1/2 irregular LDPC code. (Ack: S. AbuSurra)

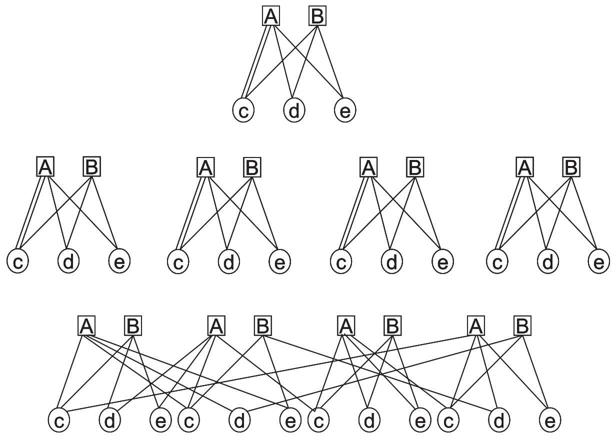


Fig. 4. Illustration of the protograph copy and permute procedure with $q = 4$ copies.

It has been shown [11] that to optimize the decoding threshold on the binary erasure channel, the shapes of the VND and CND transfer curves must be well matched in the sense that the CND curve fits inside the VND curve (an example will follow). This situation has also been observed on the BI-AWGNC [15]. Further, to achieve a good match, the number of different VN degrees need only be about 3 or 4 and the number of different CN degrees need only be 1 or 2.

Example 1: We consider the design of a rate-1/2 irregular LDPC code with four possible VN degrees and two possible CN degrees. Given than $\lambda(1) = \rho(1) = 1$ and $R = 1 - \int_0^1 \rho(z)dz / \int_0^1 \lambda(z)dz$ [16],[4], only two of the four coefficients for $\lambda(z)$ need be specified and only one of the two for $\rho(z)$ need be specified. A non-exhaustive search yielded $\lambda(z) = 0.267z + 0.176z^2 + 0.127z^3 + 0.430z^9$ and $\rho(z) = 0.113z^4 + 0.887z^7$ with a decoding threshold of $(E_b/N_0)_{thres} = 0.414$ dB. The EXIT chart for $E_b/N_0 = 0.55$ dB is presented in Fig. 3. The figure also gives the "node perspective" degree distribution information. □

The references contain additional information on EXIT charts, including the so-called area property, EXIT charts for the Rayleigh channel, for higher-order modulation, and for multi-input/multi-output channels [14][15][11][23].

III. Design of Protograph-Based Codes

A. Definition and Problem Statement

A protograph [24][25][26][27] is a relatively small bipartite graph from which a larger graph can be obtained by a copy-and-permute procedure: the protograph is copied Q times, and then the edges of the individual replicas are permuted among the replicas (under restrictions described below) to obtain a single, large graph. An example is presented in Fig. 4. The permuted edge connections are specified by the parity-check matrix $\mathbf{H}$. Note that the edge permutations cannot be arbitrary. In particular, the nodes of the protograph are labeled so that if variable node V is connected to check node C in the protograph, then variable node V in a replica can only connect to one of the Q replicated C check nodes. Doing so preserves the decoding threshold properties of the protograph. A protograph can possess parallel edges, i.e., two nodes can be connected by more than one edge. For LDPC codes, the copy-and-permute procedure must eliminate such parallel connections in order to obtain a derived graph appropriate for a parity-check matrix.

It is convenient to choose the parity-check matrix $\mathbf{H}$ as an $M \times N$ array of $Q \times Q$ (weight-one) circulant permutation matrices (some of which may be the $Q \times Q$ zero matrix). When $\mathbf{H}$ is an array of circulants, the LDPC code will be quasi-cyclic. Such a structure has a favorable impact on both the encoder and the decoder. The encoder for QC codes can be implemented with shift-register circuits with complexity linearly proportional to m for serial encoding and to n for parallel encoding [13]. By contrast, encoders for unstructured LDPC codes require much more work. The decoder for QC LDPC codes can be implemented in a modular fashion by exploiting the circulant-array structure of $\mathbf{H}$ [28][29].

Below we present an extension of the EXIT approach to codes defined by protographs. This extension is a multi-dimensional numerical technique and as such does not have a two-dimensional EXIT chart representation of the iterative decoding procedure. Still, the technique yields decoding thresholds for LDPC code ensembles specified by protographs. This multi-dimensional technique is facilitated by the relatively small size of protographs and permits the analysis of protograph code ensembles characterized by the presence of *critical node types*, i.e., node types which can lead to failed EXIT-based convergence of code ensembles. Examples of critical node types are degree-1 variable nodes and punctured variable nodes.

A code ensemble specified by a protograph is a refinement (sub-ensemble) of a code ensemble specified simply by the protograph's (hence, LDPC code's) degree distributions. To demonstrate this, we introduce the adjacency matrix $\mathbf{B} = [b_{ji}]$ for a protograph, also called a base matrix [25], where b_{ji} is the number of edges between CN j and VN i. As an example, for the protograph at the top of Fig. 4,

$$\mathbf{B} = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} .$$

Consider also an alternative protograph and base matrix specified by

$$\mathbf{B}' = \begin{pmatrix} 2 & 0 & 2 \\ 1 & 2 & 0 \end{pmatrix} .$$

The degree distributions of both of these protographs are identical and are easily seen to be

$$\begin{aligned} \lambda(z) &= \frac{4}{7}z + \frac{3}{7}z^2 \\ \rho(z) &= \frac{3}{7}z^2 + \frac{4}{7}z^3. \end{aligned}$$

However, the ensemble corresponding to $\mathbf{B}$ has a threshold of $E_b/N_0 = 0.78$ dB and that corresponding to $\mathbf{B}'$ has a threshold at 0.83 dB. (For reference, density evolution [16] applied to the above degree distributions gives 0.817 dB.)

As another example, let

$$\mathbf{B} = \begin{pmatrix} 1 & 2 & 1 & 1 & 0 \\ 2 & 1 & 1 & 1 & 0 \\ 1 & 2 & 0 & 0 & 1 \end{pmatrix}$$

and

$$\mathbf{B}' = \begin{pmatrix} 1 & 3 & 1 & 0 & 0 \\ 2 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 \end{pmatrix} ,$$

noting that they have identical degree distributions. We also puncture the bits corresponding to the second column in each base matrix. Using the multidimensional EXIT algorithm described below, the thresholds for $\mathbf{B}$ and $\mathbf{B}'$ in this case were computed to be 0.48 dB and $+\infty$, respectively.

Thus, standard EXIT analysis based on degree distributions is inadequate for protograph-based LDPC code design. In fact, the presence of degree-1 variable nodes as in our second example implies that there is a term in the summation in (9) of the form

$$\lambda_1 I_{E,V}(1, I_{A,V}) = J\left(\sigma_{ch}\right) .$$

Since $J\left(\sigma_{ch}\right)$ is always less than one for $0 < \sigma_{ch} < \infty$ and since $\sum_{d=1}^{d_v} \lambda_d = 1$, the summation in (9), that is, $I_{E,V}$, will be strictly less than one. Again, standard EXIT analysis implies failed convergence for codes with the same degree distributions as $\mathbf{B}$ and $\mathbf{B}'$. This is in contrast with the fact that codes in the $\mathbf{B}$ ensemble do converge when the SNR exceeds the threshold of 0.48 dB.

In the following, a multidimensional EXIT technique [30][31] will be presented which overcomes this issue and allows the determination of the decoding threshold for codes based on protographs (possibly with punctured nodes).

B. Multidimensional EXIT Analysis

The algorithm presented in [30][31] eliminates the average in (9) and considers the propagation of the messages on a decoding tree which is specified by the protograph of the ensemble. Let $\mathbf{B} = [b_{ji}]$ be the $M \times N$ base matrix for the protograph under analysis. Let $I_{E,V}^{i \to j}$ be the extrinsic mutual information between code bits associated with "type i" VNs and the LLRs $L_{i \to j}$ sent from these VNs to "type j" CNs. Similarly, let $I_{E,C}^{j \to i}$ be the extrinsic mutual information between code bits associated with "type i" VNs and the LLRs $L_{j \to i}$ sent from "type j" CNs to these VNs. Then, because $I_{E,C}^{j \to i}$ acts as *a priori* mutual information in the calculation of $I_{E,V}^{i \to j}$, following (7) we have (given an edge exists between CN j and VN i, i.e., given $b_{ji} \neq 0$)

$$I_{E,V}^{i \to j} = J\left(\sqrt{\sum_{c=1}^{M} (b_{ci} - \delta_{cj}) \left(J^{-1}(I_{E,C}^{c \to i})\right)^2 + \sigma_{ch,i}^2}\right), \tag{11}$$

where $\delta_{cj} = 1$ when $c = j$ and $\delta_{cj} = 0$ when $c \neq j$. $\sigma_{ch,i}^2$ is set to zero if code bit i is punctured. Similarly, because $I_{E,V}^{i \to j}$ acts as *a priori* mutual information in the calculation of $I_{E,C}^{j \to i}$, following (8) we have (when $b_{ji} \neq 0$)

$$I_{E,C}^{j \to i} = 1 - J\left(\sqrt{\sum_{v=1}^{N} (b_{jv} - \delta_{ci}) \left(J^{-1}(1 - I_{E,V}^{v \to j})\right)^2}\right). \tag{12}$$

The multidimensional EXIT algorithm can now be presented as follows.

1) *Initialization*. Select E_b/N_0. Initialize a vector $\boldsymbol{\sigma}_{ch} = (\sigma_{ch,0}, \ldots, \sigma_{ch,N-1})$ such that

$$\sigma_{ch,i} = 8R\left(\frac{E_b}{N_0}\right)_i$$

where $(E_b/N_0)_i$ equals zero when x_i is punctured and equals the selected E_b/N_0 otherwise.

2) *VN to CN*. For $i = 0, \ldots, N-1$ and $j = 0, \ldots, M-1$, compute (11).

3) *CN to VN*. For $i = 0, \ldots, N-1$ and $j = 0, \ldots, M-1$, compute (12).

4) *Cumulative mutual information*. For $i = 0, \ldots, N-1$, compute

$$I_{CMI}^{i} = J\left(\sqrt{\sum_{c=1}^{M} \left(J^{-1}(I_{E,C}^{c \to i})\right)^2 + \sigma_{ch,i}^2}\right).$$

5) If $I_{CMI}^{i} = 1$ (up to desired precision) for all i, then stop; otherwise, go to step 2.

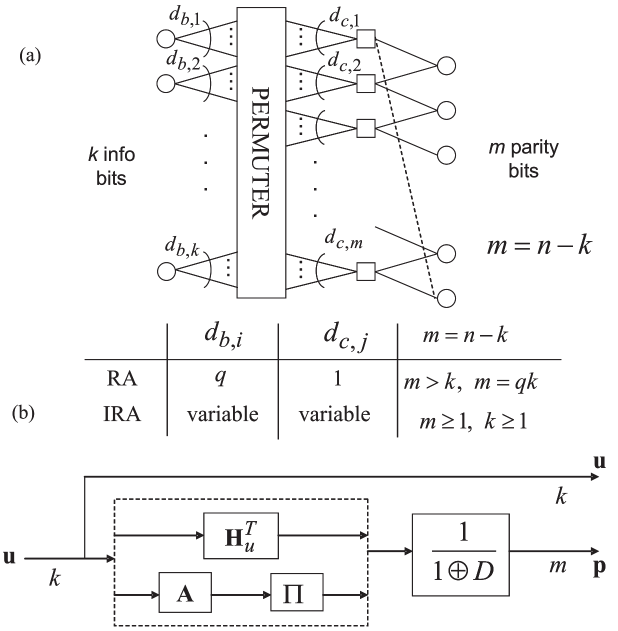
This algorithm converges only when the selected E_b/N_0 is above the threshold. Thus, the threshold is the lowest value of E_b/N_0 for which all I_{CMI}^{i} converge to 1. As shown in [30][31], the thresholds computed by this algorithm are typically within 0.05 dB of those computed by density evolution. Recalling that many classes of multi-edge type (MET) [26] LDPC codes rely on simple protographs, the above algorithm provides an accurate threshold estimation for MET ensembles, with a remarkable reduction in computational complexity relative to the density evolution analysis proposed in [26].

IV. Accumulator-Based Code Designs

A. Repeat-Accumulate Codes

This section provides an overview of the design of LDPC codes that can be considered to be a concatenation of a set of repetition codes with one or more accumulators, through an interleaver. The first example of accumulator-based codes were the so-called repeat-accumulate (RA) codes [32]. Despite their simple structure, they were shown to provide good performance and, more importantly, they paved a path toward the design of efficiently encodable LDPC codes. RA codes and other accumulator-based codes are LDPC codes that can be decoded as serial turbo codes or as LDPC codes.

An RA code consists of a serial concatenation of a single rate-$1/q$ repetition code through an interleaver with an accumulator having transfer function $1/(1 \oplus D)$. RA codes can be either non-systematic or systematic. In the first case, the accumulator output, $\mathbf{p}$, is the codeword and the code rate is $1/q$. For systematic RA codes, the information word, $\mathbf{u}$, is combined with $\mathbf{p}$ to yield the codeword $\mathbf{c} = [\mathbf{u}\ \mathbf{p}]$ and so that the code rate is $1/(1+q)$. RA codes perform reasonably well on the AWGN channel, and they tend to approach the channel capacity as their rate $R \rightarrow 0$ and their block length $n \rightarrow \infty$. Their main limitations are the code rate, which cannot be higher than $1/2$, and the performance of short and medium-length RA codes. The following subsections will present a brief overview of the major enhancements to RA codes which permit operation closer to capacity for both high and low rates.



	$d_{b,i}$	$d_{c,j}$	$m = n - k$
RA	q	1	$m > k,\ m = qk$
IRA	variable	variable	$m \geq 1,\ k \geq 1$

Fig. 5. Tanner graph (a) and encoder (b) for irregular repeat-accumulate codes.

B. Irregular Repeat-Accumulate codes

The systematic irregular repeat-accumulate (IRA) codes generalize the systematic RA codes in that the repetition rate may differ across the k information bits and that a variable number of bits in the repeated word are combined (modulo 2) prior to sending them through the accumulator. Irregular repeat-accumulate [33] codes provide several advantages over RA codes. They allowing both flexibility in the choice of the repetition rate for each information bit so that high rate codes may be designed and capacity is more easily approached. 57.8317in;original-height 7.0188in;cropleft ”0”;croptop ”1”;cropright

The Tanner graph for IRA codes is presented in Fig. 5(a) and the encoder structure (to be discussed further later) is depicted in Fig. 5(b). The variable repetition rate is accounted for in the graph by letting $d_{b,i}$ vary with i. The accumulator is represented by the right-most part of the graph, where the dashed edge is added to include the possibility of a tail-biting trellis. Also, we see that $d_{c,j}$ interleaver output bits are added (modulo 2) to produce the j-th accumulator input. Fig. 5 also includes the representation for RA codes. As indicated in the table in the figure, for an RA code, each information bit node connects to exactly q check nodes ($d_{b,i} = q$) and each check node connects to exactly one information bit node ($d_{c,j} = 1$). We remark that $\{d_{b,i}\}$ and $\{d_{c,j}\}$ can be related to our earlier notation, $\{d_v(i)\}$ and $\{d_c(j)\}$, as follows: $d_v(i) = d_{b,i}$ for $i = 1, ..., k$, $d_v(i) = 2$ for $i = k+1, ..., n$, and $d_c(j) = d_{c,j} + 2$ for $j = 1, ..., m$.

To determine the code rate for an IRA code, define $\overline{q}$ to be the average repetition rate of the information bits

$$\overline{q} = \frac{1}{k} \sum_{i=1}^{k} d_{b,i},$$

and $\bar{a}$ as the average of the degrees $\{d_{c,j}\}$,

$$\bar{a} = \frac{1}{m} \sum_{j=1}^{m} d_{c,j} \ .$$

Then the code rate for systematic IRA codes is

$$R = \frac{1}{1 + \overline{q}/\bar{a}}.$$

For non-systematic IRA codes, $R = \bar{a}/\overline{q}$.

The parity-check matrix for systematic RA and IRA codes has the form

$$\mathbf{H} = [\mathbf{H}_u\ \mathbf{H}_p], \tag{13}$$

where $\mathbf{H}_p$ is an $m \times m$ ”dual-diagonal” square matrix,

$$\mathbf{H}_p = \begin{bmatrix} 1 & & & & (1) \\ 1 & 1 & & & \\ & \ddots & \ddots & & \\ & & 1 & 1 & \\ & & & 1 & 1 \end{bmatrix}, \tag{14}$$

where the upper-right 1 is included for tailing-biting accumulators. For RA codes, $\mathbf{H}_u$ is a regular matrix having column weight q and row weight 1. For IRA codes, $\mathbf{H}_u$ has column weights $\{d_{b,i}\}$ and row weights $\{d_{c,j}\}$. The encoder of Fig. 5(b) is obtained by noting that the generator matrix corresponding to $\mathbf{H}$ in (13) is $\mathbf{G} = \begin{bmatrix}\mathbf{I} & \mathbf{H}_u^T \mathbf{H}_p^{-T}\end{bmatrix}$ and writing $\mathbf{H}_u$ as $\mathbf{\Pi}^T \mathbf{A}^T$, where $\mathbf{\Pi}$ is a permutation matrix. Note also that $\mathbf{H}_p^{-T}$ performs the same computation as $1/(1 \oplus D)$ (and $\mathbf{H}_p^{-T}$ exists only when the ”tail-biting 1” is absent). Two encoding alternatives exist: (1) When the accumulator is not tail-biting, one may use $\mathbf{H}$ to encode since one may solve for the parity bits sequentially from the equation $\mathbf{c}\mathbf{H}^T = \mathbf{0}$ starting with the top row of $\mathbf{H}$ and moving on downward. (2) As discussed in the next section, quasi-cyclic IRA code designs are possible, in which case the techniques of [13] may be used.

We remark that the choice of the degree distributions of the variable nodes for an IRA code are constrained by the presence of (at least) $n-k-1$ degree-2 variable nodes. Although such a constraint ostensibly limits the code designer, for rates $R \geq 1/2$, EXIT analysis leads to optimized degree distributions having approximately $n-k-1$ degree-2 variable nodes. Moreover, when the number of degree-2 variable nodes is exactly $n-k-1$, the edge connections involving the degree-2 variable nodes induced by the IRA structure are optimal in the sense of avoiding low weight codewords [34][35].

IRA codes and a generalization will be discussed in the next two sections. Additional information may be found in the following references: [33][35][36][24][40][41] [42][43].

C. Structured IRA and IRAA Codes

Given the code rate, length, and degree distributions, an IRA code is defined entirely by the matrix $\mathbf{H}_u$ (equivalently, by $\mathbf{A}$ and $\mathbf{\Pi}$). While a random-like $\mathbf{H}_u$ would generally give good performance, it is problematic for both encoder and decoder implementations. For, in this case, a substantial amount of memory would be required to store the connection information implicit in $\mathbf{H}_u$. In addition, although standard message-passing decoding algorithms for LDPC codes are inherently parallel, the physical interconnections required to realize a code's bipartite graph becomes an implementation bottleneck and prohibits a fully parallel decoder [29]. Using a structured $\mathbf{H}_u$ matrix mitigates these problems.

Tanner [24] was the first to consider structured RA codes, more specifically, quasi-cyclic RA codes, which require tailbiting in the accumulator. Simulation results in [24] demonstrate that the QC-RA codes compete well with random-like RA codes and surpass their performance at high SNR values. Similar ideas were applied to IRA codes in [29][44][36]. In [36], IRA codes with quasi-cyclic structure are called structured IRA (S-IRA) codes.

Toward the goal of attaining structure in $\mathbf{H}$, one cannot simply choose $\mathbf{H}_u$ to be an array of circulant permutation matrices. For, it is easy to show that doing so will produce a poor LDPC code in the sense of minimum distance (consider weight-2 encoder inputs with adjacent ones). Instead, the following strategy is proposed in [36]. Let $\mathbf{P}$ be an $L \times J$ array of $Q \times Q$ circulant permutation matrices (for some convenient Q). (Conditions for designing $\mathbf{P}$ to avoid 4-cycles, etc., are described in [36].) Then set $\mathbf{A^T} = \mathbf{P}$ so that $\mathbf{H}_u = \mathbf{\Pi}^T\mathbf{P}$ and

$$\mathbf{H}_a = \begin{bmatrix}\mathbf{\Pi}^T\mathbf{P} & \mathbf{H}_p\end{bmatrix}, \tag{15}$$

where $\mathbf{H}_p$ represents the tailbiting accumulator. Note that $m = L \times Q$ and $k = J \times Q$.

We now choose $\mathbf{\Pi}$ to be a standard deterministic "row-column" interleaver so that row $lQ+q$ in $\mathbf{P}$ becomes row $qL+l$ in $\mathbf{\Pi}^T\mathbf{P}$, for all $0 \leq l < L$ and $0 \leq q < Q$. Next, we permute the rows of $\mathbf{H}_a$ by $\mathbf{\Pi}^{-T}$ to obtain

$$\mathbf{H}_b = \mathbf{\Pi}^{-T}\mathbf{H} = [\mathbf{P} \quad \mathbf{\Pi}\mathbf{H}_p], \tag{16}$$

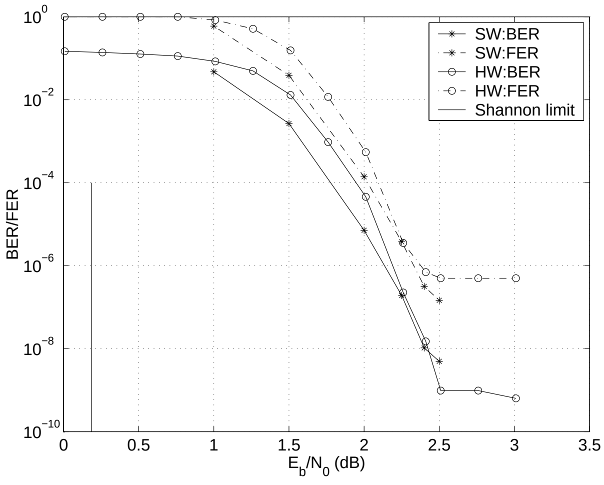


Fig. 6. Performance of a (2044,1024) S-IRA code on the BI-AWGNC. HW=hardware simulator. SW=software simulator.

where we have used the fact that $\mathbf{\Pi}^{-T} = \mathbf{\Pi}$. Finally, we permute only the columns corresponding to the parity part of $\mathbf{H}_b$, which gives

$$\mathbf{H}_{\text{S-IRA}} = [\mathbf{P} \quad \mathbf{\Pi}\mathbf{H}_p\mathbf{\Pi}^T]. \tag{17}$$

It is easily shown that the parity part of $\mathbf{H}_{\text{S-IRA}}$, that is, $\mathbf{\Pi}\mathbf{H}_p\mathbf{\Pi}^T$, is exactly in QC form,

$$\begin{bmatrix} I_0 & & & & I_1 \\ I_0 & I_0 & & & \\ & \ddots & \ddots & & \\ & & I_0 & I_0 & \\ & & & I_0 & I_0 \end{bmatrix}, \tag{18}$$

where I_0 is the $Q \times Q$ identity matrix and I_1 is obtained from I_0 by cyclically shifting all of its rows leftward. Therefore, $\mathbf{H}_{\text{S-IRA}}$ corresponds to a quasi-cyclic IRA code since $\mathbf{P}$ is also an array of $Q \times Q$ circulant permutation matrices. Observe that, except for a re-ordering of the parity bits, $\mathbf{H}_{\text{S-IRA}}$ describes the same code as $\mathbf{H}_a$ and $\mathbf{H}_b$.

As described in [36], in addition to simplifying encoder and decoder implementations, the QC structure simplifies the code design process. Simulation results for the example codes, which are produced by the design algorithms proposed in [36][37][38][39], show that the S-IRA codes perform as well as IRA codes in the waterfall region and possess very low error floors. As an example, Fig. 6 depicts the performance of a rate-1/2 $(2044, 1024)$ S-IRA code simulated in software and hardware.[1] It is seen that the floors, both bit error rate (BER) and frame error rate (FER), are quite low (it can be lower or higher depending on the decoder implementation). Lastly, S-IRA codes are suitable for rate-compatible code family design [36].

We now consider irregular repeat-accumulate-accumulate (IRAA) codes which are obtained by concatenating the parity arm of the IRA encoder of Fig. 5(b) with another accumulator, through a permuter, as shown in Fig. 7. (ARAA codes were

[1] Acknowledgment to C. Jones of JPL for simulating this code for us on an FPGA decoder.

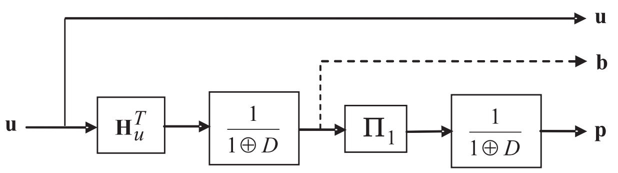


Fig. 7. IRAA encoder.

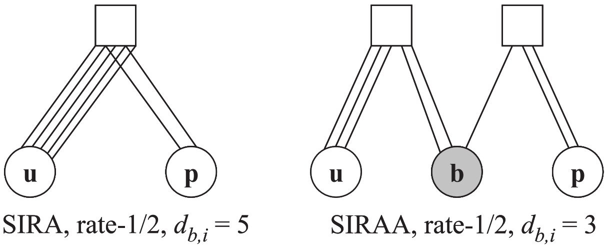


Fig. 8. Rate-1/2 SIRA and SIRAA protographs for the codes in Fig. 9. The shaded node in the SIRAA protograph represents punctured bits. SIRA: $(E_b/N_0)_{thres} = 0.97$ dB. SIRAA: $(E_b/N_0)_{thres} = 1.1$ dB.

considered in [49].) The IRAA codeword can be either $\mathbf{c} = [\mathbf{u}\ \mathbf{p}]$ or $\mathbf{c} = [\mathbf{u}\ \mathbf{b}\ \mathbf{p}]$, depending on whether the intermediate parity bits $\mathbf{b}$ are punctured or not. The parity-check matrix of the general IRAA code corresponding to Fig. 7 is

$$\mathbf{H}_{\text{IRAA}} = \begin{bmatrix} \mathbf{H}_u & \mathbf{H}_p & 0 \\ 0 & \mathbf{\Pi}_1^T & \mathbf{H}_p \end{bmatrix}, \tag{19}$$

where $\mathbf{\Pi}_1$ is the interleaver between the two accumulators. When the parity bits $\mathbf{b}$ are not transmitted, they are considered to be punctured, that is, the log-likelihood ratios for these bits are initialized by zeros before decoding. When an IRAA code is structured, we use the notation S-IRAA.

Example 2: We compare the performance of rate-1/2 (2048, 1024) S-IRA and S-IRAA codes in this example. For the S-IRA code, $d_{b,i} = 5$ for all i and for the S-IRAA code, $d_{b,i} = 3$ for all i, and the intermediate parity vector $\mathbf{b}$ is not transmitted to maintain the code rate at $1/2$. The protographs for these codes are given in Fig. 8. Because decoder complexity is proportional to the number of edges in a code's parity-check matrix, the complexity of the S-IRAA decoder is slightly greater than the complexity of the S-IRA decoder, even though the column weight in $\mathbf{H}_u$ is 3 for the former versus 5 for the latter. We observe in Fig. 9 that, for both codes, there are no error floors in the BER curves down to BER $= 5 \times 10^{-8}$ and in the FER curves down to FER $= 10^{-6}$. While the S-IRAA code is 0.2 dB inferior to the S-IRA code in the waterfall region, we conjecture that it has a lower floor (which is difficult to measure), which would be due to the second accumulator whose function is to increase minimum distance. □

Example 3: This second example is a comparison of rate-1/3 (3072,1024) S-IRA and S-IRAA codes, with $d_{b,i} = 4$ for the S-IRA code and $d_{b,i} = 3$ for the S-IRAA code. The protographs for these codes are given in Fig. 10. In this case, $\mathbf{b}$ is part of the transmitted S-IRAA codeword and the decoder complexities are the same. We see in Fig. 11 that, in the low SNR region, the performance of the S-IRA code is 0.4 dB better than the S-IRAA code. However, for high SNRs, the S-IRAA code will outperform the S-IRA code due to its lower error floor. □

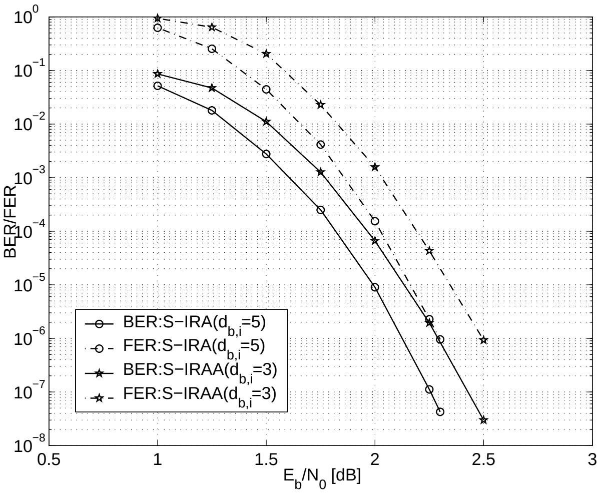


Fig. 9. Performance comparison between rate-1/2 S-IRA and S-IRAA codes on the BI-AWGNC, $n = 2048$ and $k = 1024$.

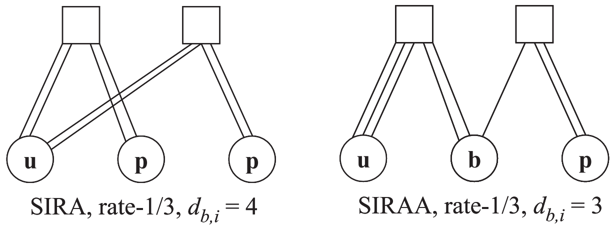


Fig. 10. Rate-1/3 SIRA and SIRAA protographs for the codes in Fig. 11. SIR: $(E_b/N_0)_{thres} = 0.40$ dB. SIRAA: $(E_b/N_0)_{thres} = 0.83$ dB.

D. Generalized IRA codes

Generalized IRA (G-IRA) codes [40][41] increase the flexibility in choosing degree distributions relative to IRA codes, allowing, for example, the design of near-regular efficiently encodable codes. The encoding algorithms for G-IRA codes are similar to those of IRA codes. For G-IRA codes, the accumulator $1/(1 \oplus D)$ in Fig. 5(b) is replaced by a generalized accumulator with transfer function $1/g(D)$ where $g(D) = \sum_{j=0}^{t} g_j D^j$ and $g_j \in \{0, 1\}$, except $g_0 = 1$. The systematic encoder therefore has the same generator matrix format, $\mathbf{G} = \begin{bmatrix} \mathbf{I} & \mathbf{H}_u^T \mathbf{H}_p^{-T} \end{bmatrix}$, but now

$$\mathbf{H}_p = \begin{bmatrix} 1 & & & & & & \\ g_1 & 1 & & & & & \\ g_2 & g_1 & \ddots & & & & \\ \vdots & g_2 & \ddots & \ddots & & & \\ g_t & \vdots & \ddots & \ddots & \ddots & & \\ & g_t & \ddots & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots \\ & & g_t & \cdots & g_2 & g_1 & 1 \end{bmatrix}.$$

Further, the parity-check matrix format is unchanged, $\mathbf{H} = [\mathbf{H}_u\ \mathbf{H}_p]$.

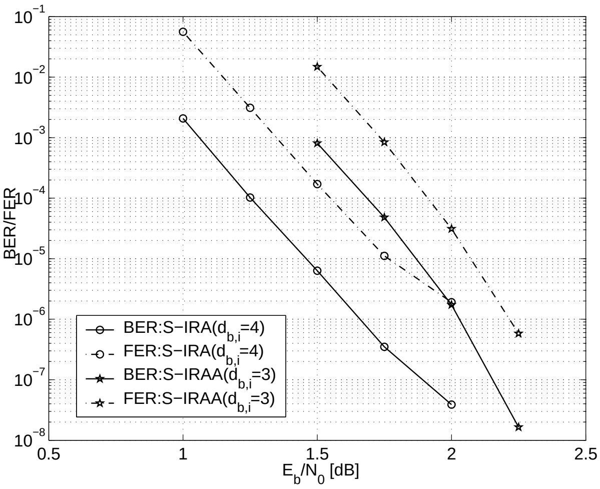


Fig. 11. Performance comparison between rate-1/3 S-IRA and S-IRAA codes on the BI-AWGNC, $n = 3072$ and $k = 1024$.

To design a G-IRA code, one must choose $g(D)$ so that the bipartite graph for $\mathbf{H}_p$ contains no length-4 cycles [40]. Once $g(D)$ has been chosen, $\mathbf{H}$ can be completed by constructing the sub-matrix $\mathbf{H}_u$, according to some prescribed degree distribution, again avoiding short cycles, this time in all of $\mathbf{H}$.

G-IRA codes are highly reconfigurable in the sense that an encoder and decoder can be designed for a set of different polynomials $g(D)$. This could be useful when faced with different channels conditions.

E. Accumulate-Repeat-Accumulate Codes

For accumulate-repeat-accumulate (ARA) codes, introduced in [45], an accumulator is added to precode a subset of the information bits of an IRA code. The primary role of this second accumulator is to improved the decoding threshold of a code, that is, to shift the BER waterfall region leftward. ARA codes are a subclass of LDPC codes and Fig. 12 presents a generic ARA Tanner graph in which punctured variable nodes are highlighted. The sparseness of the ARA graph is achieved at the price of these punctured variable nodes which act as auxiliary nodes that enlarge the $\mathbf{H}$ used by the decoder. The iterative graph-based ARA decoder thus has to deal with a redundant representation of the code, implying a larger $\mathbf{H}$ matrix than the nominal $(n-k) \times n$. This issue, together with the presence of a large number of degree-1 and degree-2 variable nodes, results in slow decoding convergence.

The ARA codes presented in [45] relies on very simple protographs. Several modified ARA protographs have been introduced in [46][47], leading to ARA and ARA-like code families with excellent performance in both the waterfall and floor regions of the codes' performance curves. The protograph of a rate-$1/2$ ARA code ensemble with repetition rate 4, denoted AR4A, is depicted in Fig. 13(a). The dark circle corresponds to a state-variable node, and it is associated with the precoded fraction of the information bits. As emphasized in the figure, such a protograph is the serial concatenation of an accumulator protograph and an IRA protograph. Half (node 2) of the information bits are sent directly to the IRA encoder, while the other half (node 1) is first precoded by the outer accumulator. This encoding procedure corresponds to a systematic code.

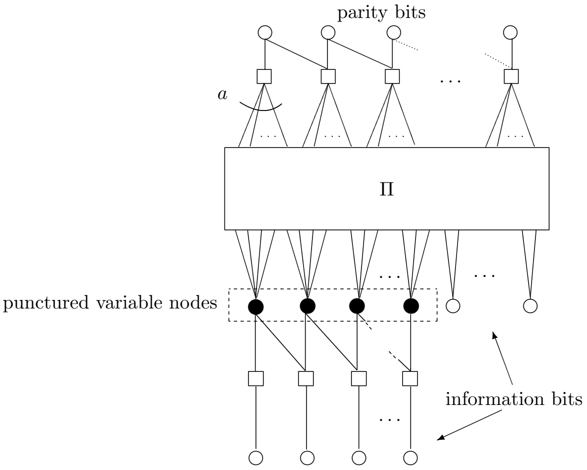


Fig. 12. Generic bipartite graph for ARA codes.

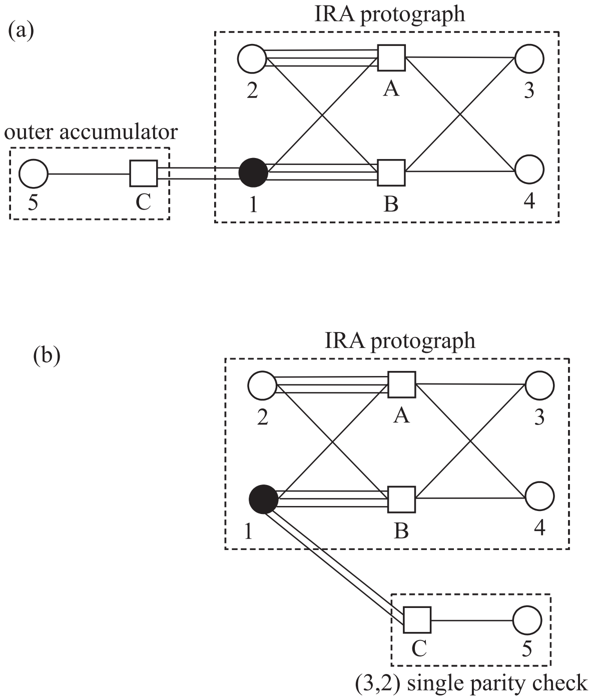


Fig. 13. AR4A protographs in (a) serial-concatenated form and (b) parallel-concatenated form. $(E_b/N_0)_{thres} = 0.55$ dB.

A different code structure is represented by the protograph in Fig. 13(b), which has a parallel-concatenated form. In this case, half (node 2) of the information bits are encoded by the IRA encoder and the other half (node 3) are encoded by both the IRA encoder and a $(3,2)$ single-parity-check encoder. The node-3 information bits (corresponding to the dark circle in the protograph) are punctured and so codes corresponding to this protograph are non-systematic. While the codes (actually, code ensembles) specified by the protographs in Fig. 13(a) are the same in the sense that the same set of codewords are

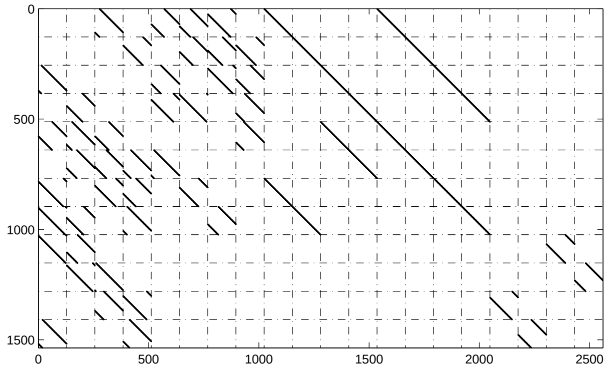


Fig. 14. **H** matrix for the (2048,1024) AR4A code.

implied, the $\mathbf{u} \to \mathbf{c}$ mappings are different. The advantage of the non-systematic protograph is that, although the node-3 information bits in Fig. 13(b) are punctured, the node degree is 6, in contrast with the node-1 information bits in Fig. 13(a), in which the node degree is only 1. Given that ARA code decoders converge so slowly, the faster-converging degree-6 node is to be preferred over the slowly converging degree-1 node.

To demonstrate this, we designed a (2048,1024) QC AR4A code whose $\mathbf{H}$ matrix is depicted in Fig. 14. The first group of 512 columns (of weight 6) correspond to variable node type 1 (Fig. 13) whose bits are punctured, and the subsequent four groups of 512 columns correspond, respectively, to node types 2, 3, 4, and 5. The first group of 512 rows correspond to check node type A, and the two subsequent groups of rows correspond to node types B and C, respectively. The performance of the code, with a maximum of $I_{max} = 50$ iterations is shown in Fig. 15. We note that the (2048,1024) AR4A code reported in [47] achieves BER $= 10^{-7}$ at $E_b/N_0 = 2$ dB with 200 iterations, whereas in the simulation here, BER $= 10^{-7}$ is achieved at $E_b/N_0 = 2.2$ dB with 50 iterations. In Fig. 16, we present the BER performance at $E_b/N_0 = 2.25$ dB for the five node types that appear in Fig. 13 for I_{max} ranging from 5 to 20. With 20 iterations, we collected 400 error events, while with fewer iterations, the numbers of collected error events were larger. From the figure, we see that the high-degree variable nodes (node types 2 and 3) converge the fastest. We note also that, while type 3 nodes have degree 6 and type 2 nodes have degree 4, type 3 nodes initially converge slower because the bits corresponding to those nodes are punctured so that the decoder receives no channel LLRs for those bits. However, by 20 iterations, the type 3 bits become more reliable than the type 2 bits.

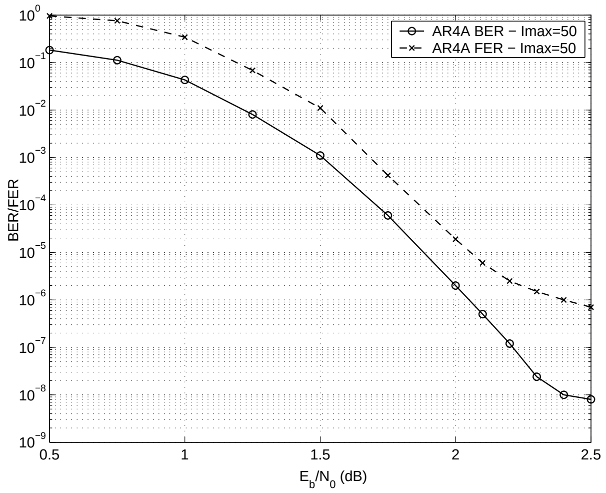


Fig. 15. BER and FER performance for an AR4A code.

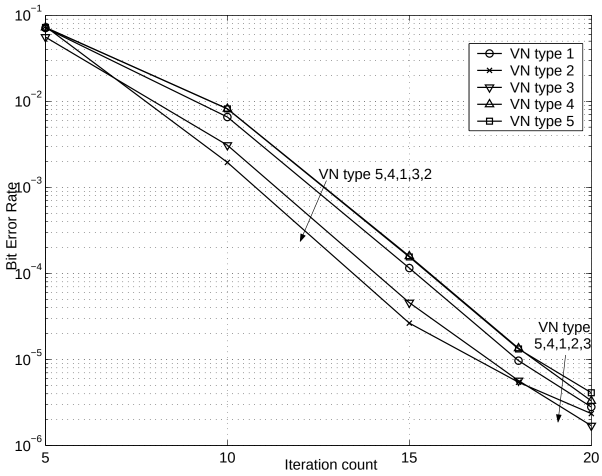


Fig. 16. Node convergence analysis for a (2048,1024) AR4A code at $E_b/N_0 = 2.25$ dB.

F. Accumulator-Based Codes in Standards

IRA codes and IRA-influenced codes are being considered for several communication standards. The ETSI DVB S2 [48] standard for digital video broadcast specifies two IRA code families with block lengths 64800 and 16200. The code rates supported by this standard range from $1/4$ to $9/10$, and a wide range of spectral efficiencies is achieved by coupling these LDPC codes with QPSK, 8-PSK, 16-APSK, and 32-APSK modulation formats. A further level of protection is afforded by an outer BCH code.

The IEEE standards bodies are also considering IRA-influenced QC LDPC codes for 802.11n (wireless local-area networks) and 802.16e (wireless metropolitan-area networks). Rather than employing a tailing-biting accumulator (which avoids weight-one columns), these standards have replaced the last block-column in (18) with a weight-three block-column and moved it to the first column, as displayed below. Encoding is facilitated by this matrix since the sum of all block-rows gives the block-row $\begin{pmatrix} I_0 & 0 & \cdots & 0 \end{pmatrix}$, so that encoding is initialized by summing all of the block-rows of $\mathbf{H}$ and solving for first Q parity bits using the resulting block-row.

$$\begin{bmatrix} I_0 & I_0 & & & & & \\ & I_0 & I_0 & & & & \\ & & I_0 & \ddots & & & \\ I_0 & & & \ddots & \ddots & & \\ & & & & \ddots & I_0 & \\ & & & & & I_0 & I_0 \\ I_0 & & & & & & I_0 \end{bmatrix}$$

ARA codes are being considered by the Consultative Committee for Space Data Systems (CCSDS) for high data-rate bandwidth-efficient space links. Very low floors are required for this applications because the scientific data (e.g., images) being transmitted from space to the ground are typically in a compressed format.

V. LDPC Codes Based on Finite Geometries

In [50], it is shown that structured LDPC codes can be constructed based on the lines and points of geometries over finite fields, namely Euclidean and projective geometries. These codes are known as finite-geometry (FG) LDPC codes. Among the FG-LDPC codes, an important subclass is the subclass of cyclic FG-LDPC codes. A cyclic LDPC code is completely characterized by its generator polynomial and its encoding can be implemented with a shift-register with feedback connections based on its generator polynomial [7]. The systematic-form generator matrix of a cyclic LDPC code can be constructed easily based on its generator polynomial [7]. Another important subclass of FG-LDPC codes is the subclass of quasi-cyclic FG-LDPC codes. As pointed out earlier, QC-LDPC codes can also be encoded easily with simple shift-registers. In this section, we give a brief survey of constructions of cyclic and quasi-cyclic FG-LDPC codes.

A. *Cyclic Euclidean Geometry LDPC Codes*

The m-dimensional Euclidean geometry over the finite field GF(q) [7][51][52], denoted by EG(m, q), consists of q^m points, and each point is represented by an m-tuple over GF(q). The point represented by the all-zero m-tuple $\mathbf{0} = (0, 0, \dots, 0)$, is called the origin of the geometry. A line in EG(m, q) is either a one-dimensional subspace of the vector space of all the m-tuples over GF(q), or a coset of it. There are $q^{m-1}(q^m - 1)/(q-1)$ lines in total. Each line consists of q points. Two points are connected by one and only one line. If $\mathbf{a}$ is a point on the line $\mathcal{L}$, we say that the line $\mathcal{L}$ passes through the point $\mathbf{a}$. Two lines either do not have any point in common or they have one and only one point in common. If two lines have a common point $\mathbf{a}$, we say that they intersect at $\mathbf{a}$. For any point $\mathbf{a}$ in EG(m, q), there are exactly $(q^m - 1)/(q-1)$ lines passing through (or intersecting at) $\mathbf{a}$. In particular, if $\mathbf{a}$ is not the origin, then it lies on $q(q^{m-1}-1)/(q-1)$ lines not passing through the origin. Furthermore, there are in total $(q^{m-1} - 1)(q^m - 1)/(q-1)$ lines not passing through the origin.

The extension field GF(q^m) of GF(q) is a realization of EG(m, q) [7][51]. Let α be a primitive element of GF(q^m). Then, the elements $0, 1, \alpha, \alpha^2, \dots, \alpha^{q^m-2}$ of GF(q^m) represent the q^m points of EG(m, q), and 0 represents the origin of the geometry. A line is a set of points of the form $\{\mathbf{a} + \beta\mathbf{a}' : \beta \in \text{GF}(q)\}$, where $\mathbf{a}$ and $\mathbf{a}'$ are linearly independent over GF(q).

Let $n_{\text{EG}} = q^m - 1$ be the number of non-origin points in the geometry. Let $\mathcal{L}$ be a line not passing through the origin. Define the n_{EG}-tuple over GF(2),

$$\mathbf{v}_{\mathcal{L}} = (v_0, v_1, \dots, v_{n_{\text{EG}}-2}),$$

whose components correspond to the $q^m - 1$ non-origin points, $\alpha^0, \alpha, \cdots, \alpha^{q^m-2}$, of EG($m, q$), where $v_i = 1$ if the point α^i lies on $\mathcal{L}$, otherwise $v_i = 0$. The vector $\mathbf{v}_{\mathcal{L}}$ is called the incidence vector of $\mathcal{L}$. Clearly, $\alpha\mathcal{L}$ is also a line in the geometry whose incidence vector $\mathbf{v}_{\alpha\mathcal{L}}$ is the right cyclic-shift of $\mathbf{v}_{\mathcal{L}}$. The lines $\mathcal{L}, \alpha\mathcal{L}, \cdots, \alpha^{n_{\text{EG}}-1}\mathcal{L}$ are all different [7] and they do not pass through the origin. Since $\alpha^{q^m-1} = 1$, $\alpha^{n_{EG}}\mathcal{L} = \mathcal{L}$. These n_{EG} lines form a cyclic class. The $(q^{m-1}-1)(q^m-1)/(q-1)$ lines in EG(m, q) not passing through the origin can be partitioned into $K = (q^{m-1} - 1)/(q-1)$ cyclic classes, denoted $\mathcal{Q}_1, \mathcal{Q}_2, \cdots, \mathcal{Q}_K$ where $\mathcal{Q}_i = \{\mathcal{L}_i, \alpha\mathcal{L}_i, \cdots, \alpha^{n_{\text{EG}}-1}\mathcal{L}_i\}$ with $1 \le i \le K$. For each cyclic class $\mathcal{Q}_i$, we form an $n_{\text{EG}} \times n_{\text{EG}}$ matrix $\mathbf{H}_{\text{EG},i}$ over GF(2) with the incidence vectors $\mathcal{L}_i, \alpha\mathcal{L}_i, \cdots, \alpha^{n_{\text{EG}}-1}\mathcal{L}_i$ as rows. $\mathbf{H}_{\text{EG},i}$ is a circulant matrix with column and row weights equal to q. For $1 \le k \le K$, let

$$\mathbf{H}_{\text{EG}(m,q),k} = \begin{bmatrix} \mathbf{H}_{\text{EG},1} \\ \mathbf{H}_{\text{EG},2} \\ \vdots \\ \mathbf{H}_{\text{EG},k} \end{bmatrix}. \tag{20}$$

Then $\mathbf{H}_{\text{EG}(m,q),k}$ consists of a column of k circulants of the same size $n_{\text{EG}} \times n_{\text{EG}}$, and it has column and row weights, kq and q, respectively. Since no two lines in EG(m, q) have more than one point in common, it follows that no two rows or two columns in $\mathbf{H}_{\text{EG}(m,q),k}$ have more than a single 1-element in common. We say that $\mathbf{H}_{\text{EG}(m,q),k}$ satisfies the *RC-constraint*. The null space of $\mathbf{H}_{\text{EG}(m,q),k}$ gives a cyclic EG-LDPC code of length $n_{\text{EG}} = q^m - 1$ and minimum distance at least $kq + 1$ [50][7], whose Tanner graph has a girth of at least 6.

Of particular interest is the two-dimensional Euclidean geometry, EG($2, q$), which is also called an affine plane over GF(q) [52]. This geometry has q^2 points and $q(q+1)$ lines, and $q^2 - 1$ of them do not pass through the origin. Each line has q points and each point lies on $q+1$ lines. Each nonorigin point lies on q lines that do not pass through the origin. If $\mathcal{L}$ is a line in EG($2, q$) not passing through the origin, then $\mathcal{L}, \alpha\mathcal{L}, \dots, \alpha^{q^2-2}\mathcal{L}$, where α is a primitive element in GF(q^2), are *all* the lines in the geometry not passing through the origin. Hence, all the lines in EG($2, q$) not passing through the origin form a single cyclic class $\mathcal{Q}$ (i.e., $K = 1$). Let $\mathbf{H}_{\text{EG}(2,q)}$ denote the $(q^2-1) \times (q^2-1)$ circulant formed by the incidence vectors of lines in $\mathcal{Q}$. It is a $(q^2-1) \times (q^2-1)$ matrix over GF(2) with both column and row weights equal to q. The null space of $\mathbf{H}_{\text{EG}(2,q)}$ gives a cyclic EG-LDPC code of length $q^2 - 1$ and minimum distance at least $q + 1$. For $q = 2^s$, the parameters

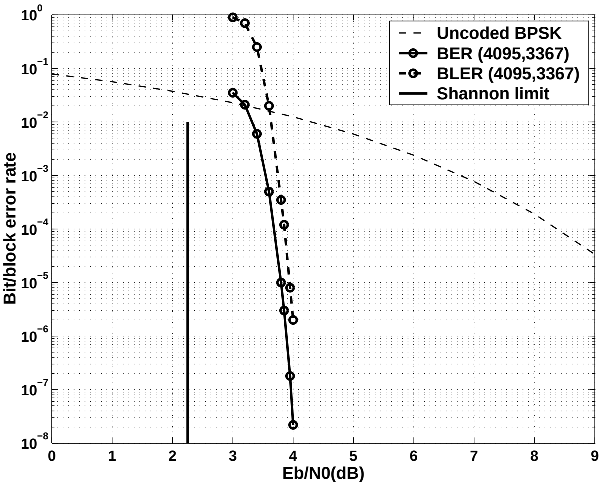


Fig. 17. Performance of the binary (4095,3367) cyclic EG-LDPC code given in Example 4 over the BI-AWGNC.

of the code with parity-check matrix $\mathbf{H}_{\mathrm{EG}(2,q)}$ are as follows [7]:

Length	$n = 2^{2s} - 1$,
Number of parity bits	$n - k = 3^s - 1$,
Dimension	$k = 2^{2s} - 3^s$,
Minimum distance	$d_{\min} \geq 2^s + 1$,
Size of the LDPC matrix	$(2^{2s} - 1) \times (2^{2s} - 1)$,
Row weight	2^s,
Column weight	2^s.

Generators polynomials for these codes can be readily obtained from [7].

Example 4: The cyclic LDPC code constructed based on the two-dimensional Euclidean geometry EG$(2, 2^6)$ over GF(2^6) is a $(4095, 3367)$ LDPC code with rate 0.822 and minimum distance 65. The performance of this code with iterative decoding using the SPA is shown in Fig. 17. At a BER of 10^{-6}, it performs 1.65 dB from the Shannon limit. Since it has a very large minimum distance, it has a very low error-floor. □

B. Cyclic Projective Geometry LDPC Codes

The m-dimensional projective geometry over GF(q), denoted by PG(m, q), consists of $n_{\mathrm{PG}} = (q^{m+1} - 1)/(q - 1)$ points. Each point is represented by a non-zero $(m+1)$-tuple $\mathbf{a}$ over GF(q) such that all $q-1$ non-zero multiples $\beta\mathbf{a}$, where β is a non-zero element in GF(q), represent the same point. A line in PG(m, q) consists of all points of the form $\beta_1\mathbf{a}_1 + \beta_2\mathbf{a}_2$, where $\mathbf{a}_1$ and $\mathbf{a}_2$ are two $(m + 1)$-tuples that are linearly independent over GF(q) and β_1 and β_2 are elements in GF(q), with β_1 and β_2 not simultaneously equal to zero. There are $(q^{m+1} - 1)(q^m - 1)/(q^2 - 1)(q - 1)$ lines in PG(m, q) and each line consists of $q+1$ points. Two points are connected by one and only one line and each point lies on $(q^m - 1)/(q - 1)$ lines.

The extension field GF(q^{m+1}) of GF(q) is a realization of PG(m, q) [7]. Let α be a primitive element of GF(q^{m+1}). A point in PG(m, q) is represented by a non-zero element α^i. Every nonzero element in the base field GF(q) can be written as α^l for some l which is divisible by $(q^{m+1} - 1)/(q - 1)$. Hence, the elements α^i and α^j represent the same point in PG(m, q) if and only if $i \equiv j \pmod{(q^{m+1} - 1)/(q - 1)}$. Therefore, we can take the elements $1, \alpha, \ldots, \alpha^{n_{\mathrm{PG}}-1}$ to represent all the points in PG(m, q).

Let $\mathcal{L}$ be a line in PG(m, q). Define the n_{PG}-tuple over GF(2) $\mathbf{v}_{\mathcal{L}} = (v_0, v_1, \ldots, v_{n_{\mathrm{PG}}-1})$ whose components correspond to the $n_{\mathrm{PG}} = (q^{m+1} - 1)/(q - 1)$ points of PG(m, q), where $v_i = 1$ if the point represented by α^i lies on $\mathcal{L}$, otherwise $v_i = 0$. The vector $\mathbf{v}_{\mathcal{L}}$ is called the incidence vector of $\mathcal{L}$. Clearly, $\alpha\mathcal{L}$ is also a line in the geometry whose incidence vector $\mathbf{v}_{\alpha\mathcal{L}}$ is the cyclic-shift of $\mathbf{v}_{\mathcal{L}}$.

For even m, the lines in PG(m, q) can be partitioned into $K_1 = (q^m - 1)/(q^2 - 1)$ cyclic classes $\mathcal{Q}_1, \mathcal{Q}_2, \cdots, \mathcal{Q}_{K_1}$, each class consisting of n_{PG} lines. For each cyclic class $\mathcal{Q}_i$, we can form an $n_{\mathrm{PG}} \times n_{\mathrm{PG}}$ circulant $\mathbf{H}_{\mathrm{PG},i}$ with both column and row weights equal to $q + 1$. For $1 \leq k \leq K_1$, form the following matrix:

$$\mathbf{H}^{(1)}_{\mathrm{PG}(m,q),k} = \begin{bmatrix} \mathbf{H}_{\mathrm{PG},1} \\ \mathbf{H}_{\mathrm{PG},2} \\ \vdots \\ \mathbf{H}_{\mathrm{PG},k} \end{bmatrix}, \qquad (21)$$

which has column and row weights $k(q + 1)$ and $q + 1$, respectively. The null space of $\mathbf{H}^{(1)}_{\mathrm{PG}(m,q),k}$ gives a cyclic PG-LDPC code of length $n_{\mathrm{PG}} = (q^{m+1}-1)/(q-1)$ and minimum distance at least $k(q + 1) + 1$ whose Tanner graph has a girth of at least 6. For odd m, the lines in PG(m, q) can be partitioned into $K_2 + 1$ cyclic classes, $\mathcal{Q}_0, \mathcal{Q}_1, \mathcal{Q}_2, \cdots, \mathcal{Q}_{K_2}$, where $K_2 = q(q^{m-1}-1)/(q^2-1)$. Except for $\mathcal{Q}_0$, each cyclic class consists of n_{PG} lines. The cyclic class $\mathcal{Q}_0$ consists of only $\lambda = (q^{m+1} - 1)/(q^2 - 1)$ lines. For each cyclic class $\mathcal{Q}_i$ with $i \neq 0$, we can form a $n_{\mathrm{PG}} \times n_{\mathrm{PG}}$ circulant $\mathbf{H}_{\mathrm{PG},i}$ with the incidence vectors of the lines in $\mathcal{Q}_i$ as rows. For $1 \leq k \leq K_2$, we can form a matrix $\mathbf{H}^{(2)}_{\mathrm{PG}(m,q),k}$ of the form given by (21). The null space of $\mathbf{H}^{(2)}_{\mathrm{PG}(m,q),k}$ gives a cyclic PG-LDPC code of length n_{PG} and minimum distance at least $k(q + 1) + 1$ whose Tanner graph has a girth of at least 6.

As in the case of Euclidean geometries, the two-dimensional projective geometry, PG$(2, q)$, which is also called a projective plane over GF(q) [52], is of particular interest. This geometry has $q^2 + q + 1$ points and $q^2 + q + 1$ lines. Each line has $q + 1$ points and each point lies on $q + 1$ lines. If $\mathcal{L}$ is a line in PG$(2, q)$, then $\mathcal{L}, \alpha\mathcal{L}, \ldots, \alpha^{q^2+q}\mathcal{L}$, where α is a primitive element in GF(q^2), are *all* the lines in the geometry. Hence, all the lines in PG$(2, q)$ form a single cyclic class $\mathcal{Q}$ (i.e., $K_1 = 1$). Let $\mathbf{H}_{\mathrm{PG}(2,q)}$ denote the $n_{\mathrm{PG}} \times n_{\mathrm{PG}}$ circulant formed by the incidence vectors of the lines in $\mathcal{Q}$. It is a $(q^2 + q + 1) \times (q^2 + q + 1)$ matrix over GF(2) with both column and row weights equal to $q + 1$. The null space of $\mathbf{H}_{\mathrm{PG}(2,q)}$ gives a cyclic PG-LDPC code of length $q^2 + q + 1$ and minimum distance at least $q + 2$. For $q = 2^s$, the parameters of the cyclic PG-LDPC code given by the null space of $\mathbf{H}_{\mathrm{PG}(2,q)}$ are as follows [7]:

Length	$n = 2^{2s} + 2^s + 1,$
Number of parity bits	$n - k = 3^s + 1,$
Dimension	$k = 2^{2s} + 2^s - 3^s,$
Minimum distance	$d_{\min} \geq 2^s + 2,$
Size of the LDPC matrix	$(2^{2s} + 2^s + 1) \times (2^{2s} + 2^s + 1),$
Row weight	$2^s + 1,$
Column weight	$2^s + 1.$

Generators polynomials for these codes can also be readily obtained from [7].

C. Quasi-Cyclic Finite Geometry LDPC Codes

Let $\mathbf{R}_{\mathrm{EG}(m,q),k}$ be the transpose of the parity-check matrix $\mathbf{H}_{\mathrm{EG}(m,q),k}$ of a cyclic EG-LDPC code given by (20), i.e.,

$$\mathbf{R}_{\mathrm{EG}(m,q),k} \triangleq \mathbf{H}^T_{\mathrm{EG}(m,q),k} = [\mathbf{H}^T_1 \mathbf{H}^T_2 \cdots \mathbf{H}^T_k], \qquad (22)$$

which consists of a row of k circulants of size $n_{\mathrm{EG}} \times n_{\mathrm{EG}}$. It is a $(q^m - 1) \times k(q^m - 1)$ matrix with column and row weights q and kq, respectively. The null space of $\mathbf{R}_{\mathrm{EG}(m,q),k}$ gives a quasi-cyclic EG-LDPC code of length $k(q^m - 1)$ and minimum distance at least $q + 1$ whose Tanner graph has a girth of at least 6.

Similarly, let $\mathbf{R}^{(e)}_{\mathrm{PG}(m,q),k}$ be the transpose of $\mathbf{H}^{(e)}_{\mathrm{PG}(m,q),k}$ with $e = 1$ or 2. Then the null space of $\mathbf{H}^{(e)}_{\mathrm{PG}(m,q),k}$ gives a quasi-cyclic PG-LDPC code of length $k(q^{m+1} - 1)/(q - 1)$ and minimum distance at least $q + 2$.

Example 5: Consider the 3-dimensional projective geometries PG$(3, 2^3)$ over GF(2^3). This geometry consists of 585 points and 4745 lines, each line consists of 9 points. The lines in this geometry can be partitioned into 9 cyclic classes, $\mathcal{Q}_0, \mathcal{Q}_1, \cdots, \mathcal{Q}_8$, where $\mathcal{Q}_0$ consists of 65 lines and each of the other 8 cyclic classes consists of 585 lines. For each $\mathcal{Q}_i$ with $1 \leq i \leq 8$, we can form a 585×585 circulant $\mathbf{H}_{\mathrm{PG},i}$ over GF(2) with the incidence vectors in $\mathcal{Q}_i$ as the rows. Set $k = 6$. Form the following 585×3510 matrix: $\mathbf{R}^{(2)}_{\mathrm{PG}(3,2^3),6} = [\mathbf{H}^T_{\mathrm{PG},1}\, \mathbf{H}^T_{\mathrm{PG},2} \cdots \mathbf{H}^T_{\mathrm{PG},6}]$, which has column and row weights 9 and 54, respectively. The null space of this matrix gives a $(3510, 3109)$ quasi-cyclic PG-LDPC code with rate 0.8858 and minimum distance at least 10. The performance of this code decoded with iterative decoding using the SPA is shown in Fig. 18. At a BER of 10^{-6}, it performs 1.3 dB from the Shannon limit. □

Other LDPC codes constructed based on finite geometries can be found in [53][54][55][56][57]. Finite geometry LDPC codes can also be effectively decoded with one-step majority-logic decoding [7], hard-decision bit-flipping (BF) decoding [1][50][7] and weighted BF decoding [50][58][59][60]. These decoding methods together with the soft-input and soft-output (SISO) iterative decoding based on belief propagation offer various trade-offs between performance and decoding complexity. The one-step majority-logic decoding requires the least decoding complexity while the (SISO) iterative decoding based on belief propagation requires the most decoding complexity and the other two decoding methods are in between. Fig. 19 shows the performances of the (4095,3367) cyclic EG-LDPC code given in Example 4 with various decoding methods.

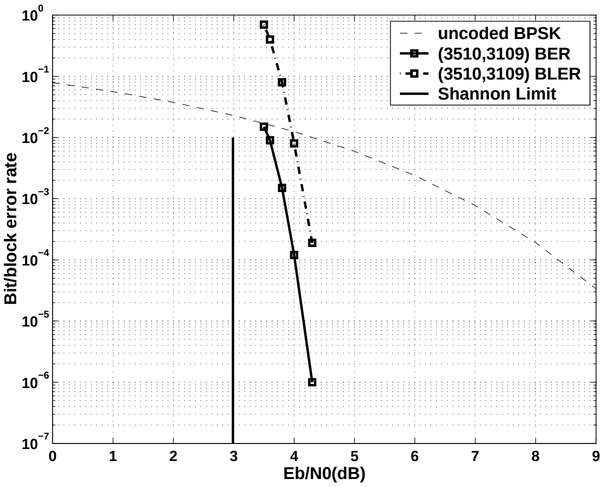


Fig. 18. Performance of the binary (3510,3109) quasi-cyclic PG-LDPC code given in Example 5 over the BI-AWGNC.

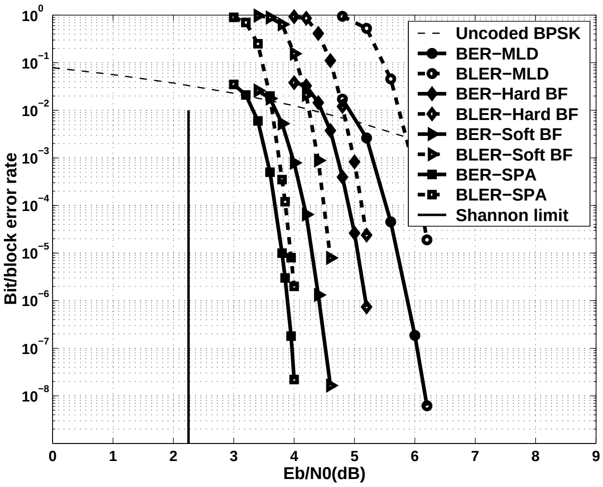


Fig. 19. Performance of the binary (4095,3367) EG-LDPC code given in Example 4 with various decoding techniques over the BI-AWGNC. MLD = majority-logic decoding. BF = bit-flipping. SPA = sum-product algorithm.

VI. Regular RS-Based LDPC Codes

This section first gives a brief survey of a class of structured LDPC codes that are constructed from the codewords of Reed-Solomon (RS) codes with two information symbols. Then two new classes of Reed-Solomon-based quasi-cyclic LDPC codes are presented. Experimental results show that constructed codes perform very well over the AWGN channel with iterative decoding.

In [61], a class of structured regular LDPC codes was presented which were constructed from the codewords of RS codes with two information symbols. These codes are referred to as RS-based LDPC codes and their parity-check matrices are arrays of permutation matrices. RS-based LDPC codes perform well with iterative decoding over the AWGN channel. Most importantly, they have low error-floors and their decoding converges very fast. These features are important in high-speed communication systems where very low error rates are required, such as the 10G Base-T Ethernet. In this section, we first give a more general form of the RS-based LDPC codes presented in [61] and then we present two classes of RS-based QC LDPC codes.

Let α be a primitive element of the finite field GF(q). Then the following powers of α, $\alpha^{-\infty} \triangleq 0, \alpha^0 = 1, \alpha, \ldots, \alpha^{q-2}$, form the q elements of GF(q) and $\alpha^{q-1} = 1$. For $i = -\infty, 0, 1, \cdots, q-2$, represent each element α^i of GF(q) by a q-tuple over GF(2),

$$\mathbf{z}(\alpha^i) = (z_{-\infty}, z_0, z_1, z_2, \ldots, z_{q-2}), \tag{23}$$

with components corresponding to the q elements, $\alpha^{-\infty}, \alpha^0, \cdots, \alpha^{q-2}$, of GF($q$), where the i-th component $z_i = 1$ and all the other components equal to zero. This binary q-tuple $\mathbf{z}(\alpha^i)$ is an unit-vector with one and only one 1-component and is called the *location vector* of α^i. It is clear that the location vectors of two different elements in GF(q) have their 1-components at two different locations. Suppose we form a $q \times q$ matrix $\mathbf{A}$ over GF(2) with the location vectors of the q elements of GF(q) as rows arranged in any order. Then $\mathbf{A}$ is a $q \times q$ permutation matrix.

Consider an extended $(q, 2, q-1)$ RS code $\mathcal{C}_b$ over GF(q) [7] of length q with two information symbols and minimum distance $q-1$. The nonzero codewords of $\mathcal{C}_b$ have two different weights, $q-1$ and q. Because the minimum distance of $\mathcal{C}_b$ is $q-1$, two codewords in $\mathcal{C}_b$ differ in at least $q-1$ places, i.e., they have at most one place where they have the same code symbols. Let $\mathbf{v}$ be a nonzero codeword in $\mathcal{C}_b$ with weight q. Then, the set $\mathcal{C}_b^{(0)} = \{c\mathbf{v} : c \in GF(q)\}$ of q codewords in $\mathcal{C}_b$ of weight q forms a one-dimensional subcode of $\mathcal{C}_b$ with minimum distance q and is a $(q, 1, q)$ extended RS code over GF(q). Any two codewords in $\mathcal{C}_b^{(0)}$ differ at every location. Partition $\mathcal{C}_b$ into q cosets, $\mathcal{C}_b^{(0)}, \mathcal{C}_b^{(1)}, \cdots, \mathcal{C}_b^{(q-1)}$, based on the subcode $\mathcal{C}_b^{(0)}$. Then two codewords in any coset $\mathcal{C}_b^{(i)}$ differ at every location and two codewords from two different cosets $\mathcal{C}_b^{(i)}$ and $\mathcal{C}_b^{(j)}$ with $i \neq j$ differ in at least $q-1$ locations. For $0 \leq i < q$, form a $q \times q$ matrix $\mathbf{G}_i$ over GF(q) with the codewords in $\mathcal{C}_b^{(i)}$ as rows. Then all the q entries in a column of $\mathbf{G}_i$ are different and they form all the q elements of GF(q). It follows from the structural properties of the cosets of $\mathcal{C}_b^{(0)}$ that any two rows from any matrix $\mathbf{G}_i$ differ at every position and any two rows from two different matrices $\mathbf{G}_i$ and $\mathbf{G}_j$ with $i \neq j$ can have at most one location where they have identical symbols.

For $0 \leq i < q$, replacing each entry in $\mathbf{G}_i$ by its location vector, we obtain a $q \times q^2$ matrix $\mathbf{B}_i$ over GF(2) which consists of a row of q permutation matrices of size $q \times q$,

$$\mathbf{B}_i = [\mathbf{A}_{i,0}\ \mathbf{A}_{i,1} \cdots\ \mathbf{A}_{i,q}], \tag{24}$$

where $\mathbf{A}_{i,j}$ has the location vectors of the q entries of the j-th column of $\mathbf{G}_i$ as rows. Next, we form the following $q \times q$ array of $q \times q$ permutation matrices with $\mathbf{B}_0, \mathbf{B}_1, \cdots, \mathbf{B}_{q-1}$ as submatrices arranged in a column:

$$\begin{aligned}\mathbf{H}_{rs,1} &= \begin{bmatrix} \mathbf{B}_0 \\ \mathbf{B}_1 \\ \vdots \\ \mathbf{B}_{q-1} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{A}_{0,0} & \mathbf{A}_{0,1} & \cdots & \mathbf{A}_{0,q-1} \\ \mathbf{A}_{1,0} & \mathbf{A}_{1,1} & \cdots & \mathbf{A}_{1,q-1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_{q-1,0} & \mathbf{A}_{q-1,1} & \cdots & \mathbf{A}_{q-1,q-1} \end{bmatrix}.\end{aligned} \tag{25}$$

$\mathbf{H}_{rs,1}$ is a $q^2 \times q^2$ matrix over GF(2) with both column and row weights q. For $q > 7$, each permutation matrix $\mathbf{A}_{i,j}$ is a sparse matrix and hence $\mathbf{H}_{rs,1}$ is also a sparse matrix. It follows from the structural properties of the matrices $\mathbf{G}_i$'s that no two rows (or two columns) of $\mathbf{H}_{rs,1}$ can have more than one 1-component in common. This implies that there are no four 1-components at the four corners of a rectangle in $\mathbf{H}_{rs,1}$, that is, $\mathbf{H}_{rs,1}$ satisfies the RC-constraint and, hence, has a girth of at least 6 [50][7].

For any pair of integers, (d_v, d_c), with $1 \leq d_v, d_c \leq q$, let $\mathbf{H}_{rs,1}(d_v, d_c)$ be a $d_v \times d_c$ subarray of $\mathbf{H}_{rs,1}$. Then $\mathbf{H}_{rs,1}(d_v, d_c)$ is a $d_v q \times d_c q$ matrix over GF(2) with column and row weights d_v and d_c, respectively. It is a (d_v, d_c)-regular matrix which also satisfies the RC-constraint. The null space of $\mathbf{H}_{rs,1}(d_v, d_c)$ gives a (d_v, d_c)-regular RS-based LDPC code $\mathcal{C}_{rs,1}$ of length $d_c q$ with rate at least $(d_c - d_v)/d_c$ and minimum distance at least $d_v + 1$ [50], [7], whose Tanner graph has a girth of at least 6. Since $\mathbf{H}_{rs,1}$ consists of an array of permutation matrices, no odd number of columns of $\mathbf{H}_{rs,1}$ can be added to zero. This implies that the RS-based regular LDPC code $\mathcal{C}_{rs,1}$ has only even-weight codewords. Consequently, its minimum distance is even, at least $d_v + 2$ for even d_v and $d_v + 1$ for odd d_v. The above construction gives a class of regular LDPC codes whose Tanner graphs have girth at least 6. For each $(q, 2, q-1)$ extended RS code $\mathcal{C}_b$ over GF(q), we can construct a family of regular RS-based LDPC codes with various lengths, rates and minimum distances. $\mathcal{C}_b$ is referred to as the base code.

Example 6: Consider the $(64, 2, 63)$ extended RS code $\mathcal{C}_b$ over GF(2^6). Based on the codewords of this RS code $\mathcal{C}_b$, we can construct a 64×64 array $\mathbf{H}_{rs,1}$ of 64×64 permutation matrices. Suppose we choose $d_v = 6$ and $d_c = 32$. Take a 6×32 subarray $\mathbf{H}_{rs,1}(6, 32)$ from $\mathbf{H}_{rs,1}$, say the 6×32 subarray at the upper left corner of $\mathbf{H}_{rs,1}$. $\mathbf{H}_{rs,1}(6, 32)$ is a 384×2048 matrix over GF(2) with column and row weights 6 and 32, respectively. The null space of this matrix gives a $(2048, 1723)$ regular RS-based LDPC code with rate 0.841 and minimum distance at least 8. Assume transmission over the AWGN channel with BPSK signaling. The performance of this code with iterative decoding using the SPA (50 iterations) is shown in Fig. 20. At a BER of 10^{-6}, the code performs 1.55 dB from the Shannon limit. The standard code for the IEEE 802.2 10G Base-T Ethernet is a $(2048, 1723)$ regular RS-based LDPC code given by the null space of a 6×32 subarray of the array $\mathbf{H}_{rs,1}$ constructed above. □

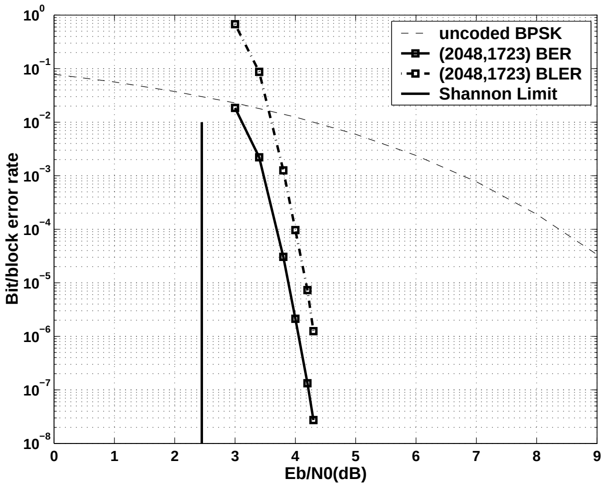


Fig. 20. Performance of the binary (2048,1723) regular RS-based LDPC code given in Example 6 over the BI-AWGNC.

A. Class-I RS-Based QC-LDPC Codes

RS codes were originally defined in polynomial form in frequency domain [63]. Using the polynomial form, arrays of circulant permutation matrices that satisfy the RC-constraint can be constructed from all the codewords of an RS code over a prime field GF(p) with two information symbols. Based on these arrays of circulant permutation matrices, a class of QC-LDPC codes can be constructed.

Let p be a prime. Consider the prime field GF(p) = $\{0, 1, \cdots, p-1\}$ under modulo-p addition and multiplication. Let $\mathcal{P} = \{\mathbf{a}(X) = a_1X + a_0 : a_1, a_0 \in GF(p)\}$ be the set of p^2 polynomials of degree one or less with coefficients from GF(p). For each polynomial $\mathbf{a}(X)$ in $\mathcal{P}$, define the following p-tuple over GF(p): $\mathbf{v} = (\mathbf{a}(0), \mathbf{a}(1), \cdots, \mathbf{a}(p-1))$, where $\mathbf{a}(j) = a_1 \cdot j + a_0$ with $j \in GF(p)$. Then the set of p^2 p-tuples,

$$\mathcal{C}_b = \{\mathbf{v} = (\mathbf{a}(0), \mathbf{a}(1), \cdots, \mathbf{a}(p-1)) : \mathbf{a}(X) \in \mathcal{P}\}, \quad (26)$$

gives a $(p, 2, p-1)$ RS code over GF(p) with two information symbols. The RS code $\mathcal{C}_b$ given by (26) is not cyclic.

Consider the subset $\mathcal{P}_0 = \{\mathbf{a}(X) = a_0 : a_0 \in GF(p)\}$ of zero-degree polynomials in $\mathcal{P}$. Then the set of p-tuples,

$$\begin{array}{rl}\mathcal{C}_b^{(0)} & = \{(\mathbf{a}(0), \mathbf{a}(1), \cdots, \mathbf{a}(p-1)) : \mathbf{a}(X) \in \mathcal{P}_0\} \\ & = \{(a_0, a_0, \cdots, a_0) : a_0 \in GF(p)\}\end{array}, \quad (27)$$

constructed from the zero-degree polynomials in $\mathcal{P}_0$ forms a one-dimensional subcode of $\mathcal{C}_b$ and is a $(p, 1, p-1)$ RS code over $GF(p)$ with minimum distance p. Partition $\mathcal{C}_b$ with respect to $\mathcal{C}_b^{(0)}$ into p cosets, $\mathcal{C}_b^{(0)}, \mathcal{C}_b^{(1)}, \cdots, \mathcal{C}_b^{(p-1)}$, where

$$\mathcal{C}_b^{(i)} = \{(\mathbf{a}(0), \cdots, \mathbf{a}(p-1)) : \mathbf{a}(X) = iX + a_0, a_0 \in GF(p)\}. \quad (28)$$

For $0 \leq i < p$, $\mathcal{C}_b^{(i)}$ contains p codewords in $\mathcal{C}_b$ of the following form:

$$(i \cdot 0 + a_0, i \cdot 1 + a_0, \cdots, i \cdot (p-1) + a_0). \quad (29)$$

The codeword $(i \cdot 0, i \cdot 1, ..., i \cdot (p-1))$ in $\mathcal{C}_b^{(i)}$ is the coset leader.

For $0 \leq i < p$, form a $p \times p$ matrix $\mathbf{G}_i$ over GF(p) with the codewords in the i-th coset $\mathcal{C}_b^{(i)}$ as rows. For $0 \leq j < p$, the j-th column of $\mathbf{G}_i$ consists of the following entries: $i \cdot j + 0, i \cdot j + 1, \cdots, i \cdot j + (p-1)$, which form all the p elements of GF(p). From (27) to (29), we readily see that any two rows in $\mathbf{G}_i$ differ in all p places. Replacing each entry in $\mathbf{G}_i$ by its location vector, we obtain a row of p permutation matrices of size $p \times p$,

$$\mathbf{B}_i = [\ \mathbf{A}_{i,0} \quad \mathbf{A}_{i,1} \quad \cdots \quad \mathbf{A}_{i,p-1}\ ],$$

where $\mathbf{A}_{i,j}$ has the location vectors of $i \cdot j + 0, i \cdot j + 1, \cdots, i \cdot j + (p-1)$ as the rows,

$$\mathbf{A}_{i,j} = \begin{bmatrix} \mathbf{z}(i \cdot j + 0) \\ \mathbf{z}(i \cdot j + 1) \\ \vdots \\ \mathbf{z}(i \cdot j + (p-1)) \end{bmatrix}. \quad (30)$$

Under modulo-p addition and multiplication, the location vector $\mathbf{z}(i \cdot j + (k+1))$ of the field element $i \cdot j + (k+1)$ is the right cyclic-shift (one place to the right) of the location vector $\mathbf{z}(i \cdot j + k)$ of the field element of $i \cdot j + k$ and $\mathbf{z}(i \cdot j + 0)$ is the right cyclic-shift of $\mathbf{z}(i \cdot j + (p-1))$. Therefore $\mathbf{A}_{i,j}$ is not just a permutation matrix but also a circulant, called a circulant permutation matrix. For $0 \leq i < p$, $\mathbf{B}_i$ is hence a row of p circulant permutation matrices of size $p \times p$.

Form the following $p \times p$ array of $p \times p$ circulant permutation matrices:

$$\begin{aligned}\mathbf{H}_{rs,2} &= \begin{bmatrix} \mathbf{B}_0 \\ \mathbf{B}_1 \\ \vdots \\ \mathbf{B}_{p-1} \end{bmatrix} \quad (31) \\ &= \begin{bmatrix} \mathbf{A}_{0,0} & \mathbf{A}_{0,1} & \cdots & \mathbf{A}_{0,p-1} \\ \mathbf{A}_{1,0} & \mathbf{A}_{1,1} & \cdots & \mathbf{A}_{1,p-1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_{p-1,0} & \mathbf{A}_{p-1,1} & \cdots & \mathbf{A}_{p-1,p-1} \end{bmatrix}.\end{aligned}$$

$\mathbf{H}_{rs,2}$ is a $p^2 \times p^2$ matrix over GF(2) with both column and row weights p. Since the rows of $\mathbf{H}_{rs,2}$ correspond to the codewords in the $(p, 2, p-1)$ RS code $\mathcal{C}_b$ over GF(p) given by (26) and two codewords in $\mathcal{C}_b$ can have at most one place with the same code symbol, no two rows (or two columns) in $\mathbf{H}_{rs,2}$ can have more than one 1-component in common. Hence $\mathbf{H}_{rs,2}$ satisfies the RC-constraint and its associated Tanner graph has a girth of at least 6.

For any pair of integers, (d_v, d_c), with $1 \leq d_v, d_c \leq p$, let $\mathbf{H}_{rs,2}(d_v, d_c)$ be a $d_v \times d_c$ subarray of $\mathbf{H}_{rs,2}$. $\mathbf{H}_{rs,2}(d_v, d_c)$ is a $d_vp \times d_cp$ matrix over GF(2) with column and row weights d_v and d_c, respectively, and it also satisfies the RC-constraint. The null space of $\mathbf{H}_{rs,2}(d_v, d_c)$ gives a regular RS-based QC-LDPC code of length d_cp with rate at least $(d_c - d_v)/d_c$ and minimum distance at least $d_v + 2$ for even d_v, and $d_v + 1$ for odd d_v, whose Tanner graph has a girth of at least 6. The above construction gives a class of QC-LDPC codes with various lengths, rates and minimum distances.

Example 7: Consider the $(73, 2, 72)$ RS code $\mathcal{C}_b$ over the prime field GF(73) that is constructed based on the set of

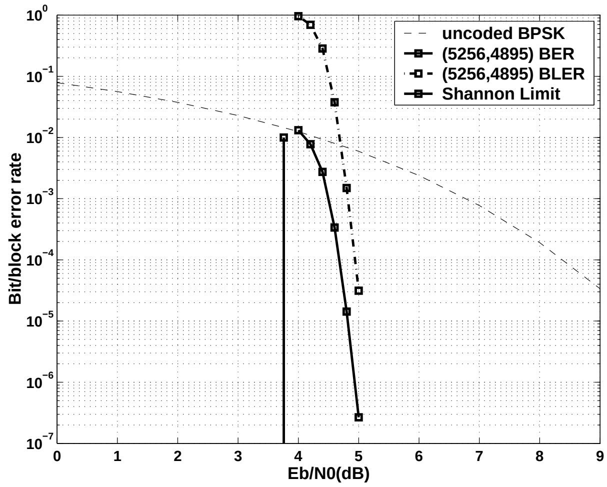


Fig. 21. Performance of the binary (5256,4895) regular RS-based QC-LPDC code given in Example 7 over the BI-AWGNC.

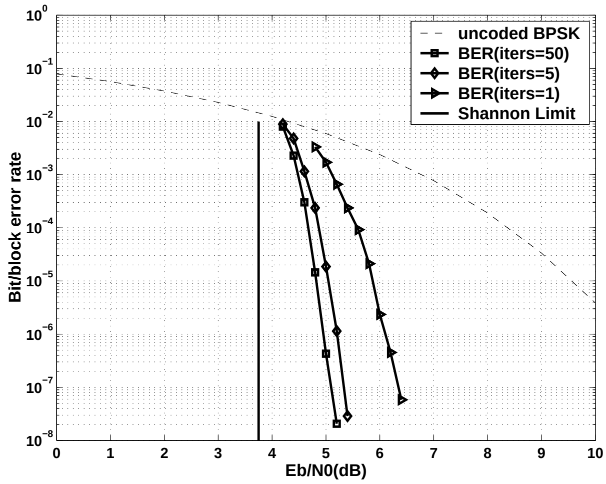


Fig. 22. The decoding convergence rate of the (5256,4895) QC-LDPC code given in Example 7.

polynomials of degree 1 or less over GF(73). Using this base RS code, we can construct a 73×73 array $\mathbf{H}_{rs,2}$ of 73×73 circulant permutation matrices. Choose $d_v = 5$ and $d_c = 72$. Take a 5×72 subarray $\mathbf{H}_{rs,2}(5,72)$ from $\mathbf{H}_{rs,2}$. $\mathbf{H}_{rs,2}(5,72)$ is a 365×5256 matrix over GF(2) with column and row weights 5 and 72, respectively. The null space of $\mathbf{H}_{rs,2}(5,72)$ gives a $(5256,4895)$ regular RS-based QC-LDPC code with rate 0.9313. The minimum distance of this code is estimated to be 12 which is twice as large as its lower bound $d_v + 1 = 6$. The performance of this code with iterative decoding using the SPA with 50 iterations is shown in Fig. 21. At a BER of 10^{-6}, it perform 1.15 dB from the Shannon limit. The rate of decoding convergence of this code is shown in Fig. 22. We see decoding of this code converges very fast. At a BER of 10^{-6}, the gap between 5 and 50 iterations is about 0.2 dB. □

B. Class-II RS-Based QC-LDPC Codes

So far in this section, we have presented two classes of RS-based LDPC codes. A code in either class is constructed based on partitioning all the codewords of an extended RS code with two information symbols into cosets with respect to a one-dimensional RS subcode. In this subsection, we present another class of RS-based LDPC codes. The construction of this class of LDPC codes is based on only the minimum weight (m-w) codewords of extended RS codes with two information symbols. In the construction, the m-w codewords of an extended RS code with two information symbols are first partitioned into q uniform classes (defined below), each with $q-1$ m-w codewords. Then based on these uniform classes, a $q \times q$ array of $(q-1) \times (q-1)$ circulant permutation matrices is formed. The null space of any subarray of this array of circulant permutation matrices gives a QC-LDPC code.

Earlier we defined the location vector of an element in the Galois field GF(q) as a q-tuple with exactly one 1-component. In our new construction of RS-based LDPC codes, we introduce a new type of location vector for the elements of GF(q). Let α be a primitive element in GF(q). For each *nonzero* element α^i in GF(q) with $0 \leq i < q-1$, its location vector $\mathbf{z}(\alpha^i)$ is defined as a $(q-1)$-tuple,

$$\mathbf{z}(\alpha^i) = (z_0, z_1, \cdots, z_{q-1}), \tag{32}$$

with components corresponding to the $q-1$ nonzero elements, $\alpha^0, \alpha, \cdots, \alpha^{q-2}$, of GF($q$), where the i-th component $z_i = 1$ and all the other $q-2$ components are zeros. Note that the 0-element of GF(q) is not included in formation of this location vector of a nonzero element in GF(q). The location vector of the 0-element of GF(q) is defined as the all-zero $(q-1)$-tuple, $(0, 0, ..., 0)$.

Again consider the $(q, 2, q-1)$ extended RS code $\mathcal{C}_b$ with two information symbols. It contains $q(q-1)$ codewords of weight $q-1$. Each of these m-w codewords contains one and only one 0-component. For $i = -\infty, 0, 1, \cdots, q-2$, let $\mathbf{v}_i = (v_{-\infty}, v_0, v_1, \cdots, v_{q-2})$ be a m-w codeword in $\mathcal{C}_b$ with i-th component $v_{i,i} = 0$. Let $U_i = \{\mathbf{v}_i, \alpha\mathbf{v}_i, \cdots, \alpha^{q-2}\mathbf{v}_i\}$ be the set of $q-1$ m-w codewords with the i-th components equal to zero. Then the $q(q-1)$ m-w codewords can be partitioned into q subsets, $U_{-\infty}, U_0, U_1, \cdots, U_{q-2}$, each consisting of $q-1$ m-w codewords. These sets are called uniform classes of m-w codewords in $\mathcal{C}_b$. Two m-w codewords in the same uniform class U_i differ in all the $q-1$ nonzero positions and they both have zeros at the i-th position. Two m-w codewords from two different classes differ in at least $q-1$ positions.

For the i-th uniform class U_i of m-w codewords, we form a $(q-1) \times q$ matrix $\mathbf{G}_i$ over GF(q) with the $q-1$ m-w codewords in U_i as rows,

$$\begin{aligned}\mathbf{G}_i &= \begin{bmatrix} \mathbf{v}_i \\ \alpha\mathbf{v}_i \\ \vdots \\ \alpha^{q-2}\mathbf{v}_i \end{bmatrix} \\ &= \begin{bmatrix} v_{i,-\infty} & v_{i,0} & \cdots & v_{i,q-2} \\ \alpha v_{i,-\infty} & \alpha v_{i,0} & \cdots & \alpha v_{i,q-2} \\ \cdots & \cdots & \ddots & \cdots \\ \alpha^{q-2}v_{i,-\infty} & \alpha^{q-2}v_{i,0} & \cdots & \alpha^{q-2}v_{i,q-2} \end{bmatrix}.\end{aligned} \tag{33}$$

The i-th column of $\mathbf{G}_i$ is a column of $q-1$ zeros and any other column consists of $q-1$ distinct nonzero entries which are the $q-1$ nonzero elements of GF(q). It follows from the structural properties of the uniform classes of m-w codewords of the $(q,2,q-2)$ extended RS code $\mathcal{C}_b$ that any two rows in the same matrix $\mathbf{G}_i$ differ in exactly $q-1$ places and any two rows from two different matrices $\mathbf{G}_i$ and $\mathbf{G}_j$ differ in at least $q-1$ places.

Replacing each entry in $\mathbf{G}_i$ by its location vector defined by (32), we obtain a row of q submatrices of size $(q-1)\times(q-1)$,

$$\mathbf{B}_i = [\ \mathbf{A}_{i,-\infty} \quad \mathbf{A}_{i,0} \quad \cdots \quad \mathbf{A}_{i,q-2}\ ], \tag{34}$$

where $\mathbf{A}_{i,i}$ is a $(q-1)\times(q-1)$ zero matrix and all the other $q-1$ submatrices $\mathbf{A}_{i,j}$'s are $(q-1)\times(q-1)$ circulant permutation matrices. Form the following $q\times q$ array of $(q-1)\times(q-1)$ circulant permutation and zero matrices:

$$\begin{aligned}\mathbf{H}_{rs,3} &= \begin{bmatrix}\mathbf{B}_{-\infty}\\ \mathbf{B}_0\\ \vdots\\ \mathbf{B}_{q-2}\end{bmatrix} \qquad (35)\\ &= \begin{bmatrix}\mathbf{A}_{-\infty,-\infty} & \mathbf{A}_{-\infty,0} & \cdots & \mathbf{A}_{-\infty,q-2}\\ \mathbf{A}_{0,-\infty} & \mathbf{A}_{0,0} & \cdots & \mathbf{A}_{0,q-2}\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{A}_{q-2,-\infty} & \mathbf{A}_{q-2,0} & \cdots & \mathbf{A}_{q-2,q-2}\end{bmatrix},\end{aligned}$$

where the submatrices, $\mathbf{A}_{-\infty,-\infty}, \mathbf{A}_{0,0}, \cdots, \mathbf{A}_{q-2,q-2}$, on the main diagonal of $\mathbf{H}_{rs,3}$ are zero matrices and the other submatrices are $(q-1)\times(q-1)$ circulant permutation matrices. $\mathbf{H}_{rs,3}$ is a $q(q-1)\times q(q-1)$ matrix over GF(2) with both column and row weights $q-1$. It follows from the structural properties of matrices $\mathbf{G}_i$'s that no two rows (or two columns) of $\mathbf{H}_{rs,3}$ have more than one 1-component in common and hence it satisfies the RC-constraint. The associated Tanner graph of $\mathbf{H}_{rs,3}$ is free of cycles of length 4 and hence has a girth of at least 6.

For $1 \le d_v, d_c \le q$, let $\mathbf{H}_{rs,3}(d_v,d_c)$ be a $d_v \times d_c$ subarray of $\mathbf{H}_{rs,3}$. It is a $d_v(q-1)\times d_c(q-1)$ matrix over GF(2). If $\mathbf{H}_{rs,3}(d_v,d_c)$ does not contain zero matrices on the main diagonal of $\mathbf{H}_{rs,3}$, it is a regular matrix with column and row weights d_v and d_c, respectively. The null space of $\mathbf{H}_{rs,3}(d_v,d_c)$ gives a (d_v,d_c)-regular RS-based QC-LDPC code of length $d_c(q-1)$ with minimum distance at least d_v+2 for even d_v and d_v+1 for odd d_v, whose Tanner graph has a girth of at least 6. If $\mathbf{H}_{rs,3}(d_v,d_c)$ contains some zero matrices of $\mathbf{H}_{rs,3}$, then it has two column weights d_v-1 and d_v and may have two row weights d_c-1 and d_c. In this case, the null space of $\mathbf{H}_{rs,3}(d_v,d_c)$ gives a near regular QC-LDPC code. The above construction gives another class of RS-based QC-LDPC codes.

Example 8: Suppose the $(64,2,63)$ extended RS code over GF(2^6) is used as the base code $\mathcal{C}_b$ for constructing QC-LDPC codes. Based on the m-w codewords of this base code, we can construct a 64×64 array $\mathbf{H}_{rs,3}$ of 63×63 circulant permutation and zero matrices. Set $d_v=6$ and $d_c=32$. Take a 6×32 subarray $\mathbf{H}_{rs,3}(6,32)$ from $\mathbf{H}_{rs,3}$, avoiding the zero matrices. Then $\mathbf{H}_{rs,3}(6,32)$ is a 378×2016 matrix over GF(2) with column and row weights 6 and 32, respectively. The null space of $\mathbf{H}_{rs,3}(6,32)$ gives a $(2016,1692)$ regular RS-based QC-LDPC code with rate 0.8392. The performance of this code with iterative decoding using the SPA with 50 iterations is shown in Fig. 23. At a BER of 10^{-6}, it performs 1.55 dB from the Shannon limit. This code is the quasi-cyclic counterpart of the $(2048,1723)$ regular RS-based LDPC code given in Example 6 (or the standard code for the IEEE 802.3 10G Base-T Ethernet). Its encoding can be implemented with 6 shift-register-adder-accumulator (SRAA) units [13], each consisting of 126 flip-flops, 64 two-input XOR gates and 64 two-input AND gates. The performance of this code is almost the same as that of the standard code for the IEEE 802.3 10G Base-T Ethernet. □

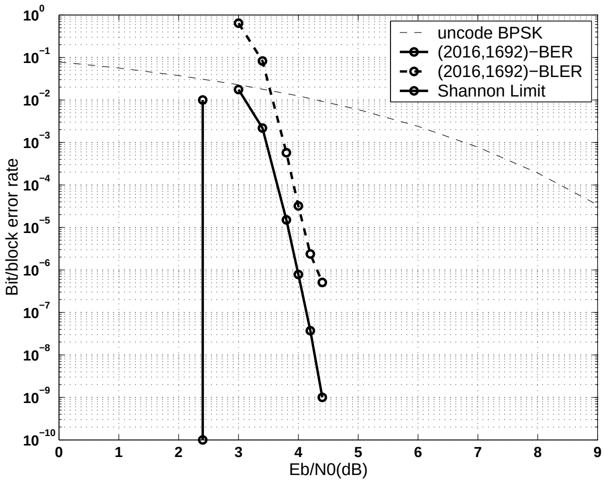


Fig. 23. Performance of the binary (2016,1692) QC-LDPC code given in Example 8 over the BI-AWGNC.

VII. Masking

Given a $d_v \times d_c$ array of permutation matrices, $\mathbf{H}_{rs,e} = [\mathbf{A}_{i,j}]$ with $e = 1, 2$ or 3, a set of permutation matrices can be masked (i.e., replaced by zero matrices) to generate a new LDPC code. The masking operation can be modeled mathematically as a special matrix product [7][57]. Let $\mathbf{W}(d_v,d_c) = [w_{i,j}]$ be a $d_v \times d_c$ matrix over GF(2). Define the following matrix product:

$$\mathbf{M}_{rs,e}(d_v,d_c) = \mathbf{W}(d_v,d_c) \circledast \mathbf{H}_{rs,e}(d_v,d_c) = [w_{i,j}\mathbf{A}_{i,j}], \tag{36}$$

where $w_{i,j}\mathbf{A}_{i,j} = \mathbf{A}_{i,j}$ for $w_{i,j}=1$ and $w_{i,j}\mathbf{A}_{i,j} = \mathbf{0}$ (a zero matrix) for $w_{i,j}=0$. With this operation, a set of permutation matrices in $\mathbf{H}_{rs,e}(d_v,d_c)$ is masked by the 0-entries of $\mathbf{W}(d_v,d_c)$. We call $\mathbf{W}(d_v,d_c)$ the masking matrix, $\mathbf{H}_{rs,e}(d_v,d_c)$ the base array (or base matrix), and $\mathbf{M}_{rs,e}(d_v,d_c)$ the masked array (or matrix). The masked matrix $\mathbf{M}_{rs,e}(d_v,d_c)$ is an array of permutation and zero matrices. The distribution of permutation matrices in $\mathbf{M}_{rs,3}(d_v,d_c)$ is identical to the distribution of 1-entries in the masking matrix $\mathbf{W}(d_v,d_c)$.

It is clear that masking operation preserves the RC-constraint on the rows and columns of the base array $\mathbf{H}_{rs,e}(d_v,d_c)$ and hence the masked matrix $\mathbf{M}_{rs,e}(d_v,d_c)$ also satisfies the RC-constraint. Furthermore, masking reduces the density of 1-entries in the base matrix and therefore the masked matrix is a sparser matrix. Consequently, the associated Tanner graph of $\mathbf{M}_{rs,e}(d_v, d_c)$ has either a larger girth or a smaller number of short cycles than that of the base matrix. If the girth of the masking matrix is $g > 6$, then the girth of the Tanner graph of the masked matrix is at least g. Since the size of a masking matrix is in general not very large, it is quite easy to construct masking matrices with relatively large girth, say 8, 10 and 12, either by computer search or by the techniques given in [64][65].

The null space of the masked matrix $\mathbf{M}_{rs,e}(d_v, d_c)$ gives an LDPC code $\mathcal{C}_{rs,e}^{(m)}$ with girth at least 6. For $e = 2$ or 3, $\mathcal{C}_{rs,e}^{(m)}$ is a QC-LDPC code. If the masking matrix is a regular matrix with constant column and row weights, then $\mathcal{C}_{rs,e}^{(m)}$ is a regular LDPC code. If the masking matrix has varying column and row weights, then $\mathcal{C}_{rs,e}^{(m)}$ is an irregular LDPC code. Masking is an effective technique for constructing long structured regular and irregular LDPC codes. The performance of an LDPC code constructed by masking depends on the choice of the masking matrix. Regular masking matrices can be constructed using algebraic or combinatorial methods. An irregular masking matrix can be constructed by computer search based on the variable- and check-node degree distributions of a code's Tanner graph derived by the evolution of the probability densities of the messages passed between the two types of nodes in a belief propagation decoder as proposed in [66].

Example 9: In this example, we choose the $(257, 2, 256)$ extended RS code over GF(257) as the base code $\mathcal{C}_b$ for code construction. Using the method given in Section VI-B, a 257×257 array $\mathbf{H}_{rs,3}$ of 256×256 circulant permutation matrices can be constructed based on the minimum weight codewords of $\mathcal{C}_b$. Choose $d_v = 8$ and $d_c = 64$. Take a 8×64 subarray $\mathbf{H}_{rs,3}(8, 64)$ from $\mathbf{H}_{rs,3}$ (avoiding zero matrices) as the base array for masking. Construct an 8×64 masking matrix $\mathbf{W}(8, 64)$ that consists of a row of eight 8×8 circulant matrices whose generators (top rows) are given in Table 1. $\mathbf{W}(8, 64)$ has column and row weights 4 and 32, respectively. Masking the base array $\mathbf{H}_{rs,3}(8, 16)$ with $\mathbf{W}(8, 64)$, we obtain a 2048×16384 regular masked matrix $\mathbf{M}_{rs,3}(8, 64)$ with column and row weights 4 and 32, respectively. The null space of $\mathbf{M}_{rs,3}(8, 64)$ gives a $(16384, 14337)$ regular RS-based QC-LDPC code with rate 0.875. The performance of this code with iterative decoding using the SPA is shown in Fig. 24. At a BER of 10^{-6}, it performs 0.85 dB from the Shannon limit. □

TABLE I. GENERATORS OF CIRCULANTS IN THE MASKING MATRIX OF EXAMPLE 9.

$\mathbf{g}_1 = (10011010)$	$\mathbf{g}_2 = (11011000)$
$\mathbf{g}_3 = (00111010)$	$\mathbf{g}_4 = (01100110)$
$\mathbf{g}_5 = (01111000)$	$\mathbf{g}_6 = (11100010)$
$\mathbf{g}_7 = (11010010)$	$\mathbf{g}_8 = (01010110)$

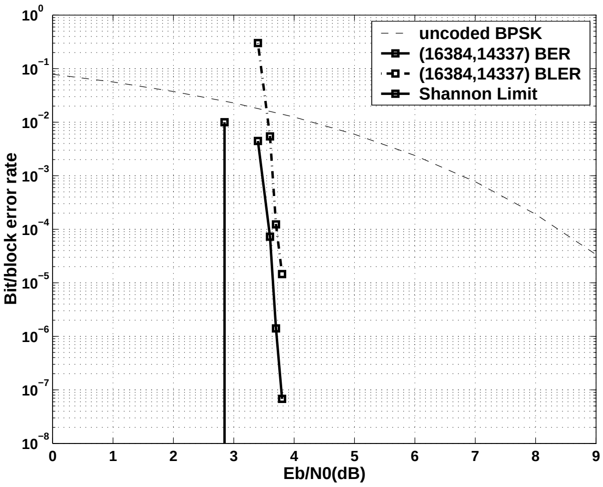


Fig. 24. Performance of the binary (16384,14337) QC-LDPC code given in Example 9 over the BI-AWGNC.

An irregular LDPC code is given by the null space of a sparse matrix $\mathbf{H}$ with varying column weights and/or varying row weights so that the code's Tanner graph has varying nodal degrees. The nodal degree distributions (hence, row/column weight distributions) from the node perspective (see Example 1) are expressed in terms of two polynomials [66], $v(X) = \sum_{i=1}^{d'_v} v_i X^{i-1}$ and $c(X) = \sum_{i=1}^{d'_c} c_i X^{i-1}$, where v_i and c_i denote the fractions of variable- and check-node with degree i, respectively, d'_v and d'_c denote the maximum variable- and check-node degrees, respectively. Irregular LDPC codes can be constructed based on the degree distributions of a code graph and masking an array of permutation matrices. First we design the degree distributions, $v(X)$ and $c(X)$, of the variable- and check-nodes of the graph of a code of rate R based on EXIT charts (or density evolution [16]). Then choose proper parameters, d_v, d_c and q (or p) that will give us the desired code length and rate R,where $d_v \geq d'_v$ and $d_c \geq d'_c$. By computer search, we construct a masking matrix $\mathbf{W}(d_v, d_c)$ that has column and row weight distributions identical (or close) to $v(X)$ and $c(X)$. Construct a base array $\mathbf{H}_{rs,e}(d_v, d_c)$ with $e = 1, 2$ or 3 using a method described above. Masking the base matrix $\mathbf{H}_{rs,e}(d_v, d_c)$ by $\mathbf{W}(d_v, d_c)$, we obtain a masked matrix $\mathbf{M}_{rs,e}(d_v, d_c)$ which has column and row weight distributions identical (or close) to $v(X)$ and $c(X)$. This masking not only gives a structured irregular LDPC code but also simplifies the code construction. Since the Tanner graph of the base matrix $\mathbf{H}_{rs,e}(d_v, d_c)$ is already free of cycles of length 4, the Tanner graph of the resultant irregular LDPC code is also free of cycles of length 4 and hence has a girth of at least 6. By contrast, in random construction, a large random bipartite graph based on the degree distributions must first constructed. In the process of constructing a code graph by computer, effort must be made to avoid cycles of length 4, which may not be easy.

Since optimal degree distributions for a given code rate are derived based on the assumptions of infinite code length, cycle-free code graph, and an infinite number of decoding iterations. When applied to construct short codes, the optimal degree distributions are no longer optimal any more and they usually result in an irregular code with a high error-floor. Therefore, proper adjustment of the degree distributions must be made to achieve good performance.

Example 10: The following degree distributions of variable- and check-nodes of a bipartite graph are designed for a code with rate 1/2 and length between 4000 and 5000: $v(X) = 0.25X + 0.625X^2 + 0.125X^8$ and $c(X) = X^6$.

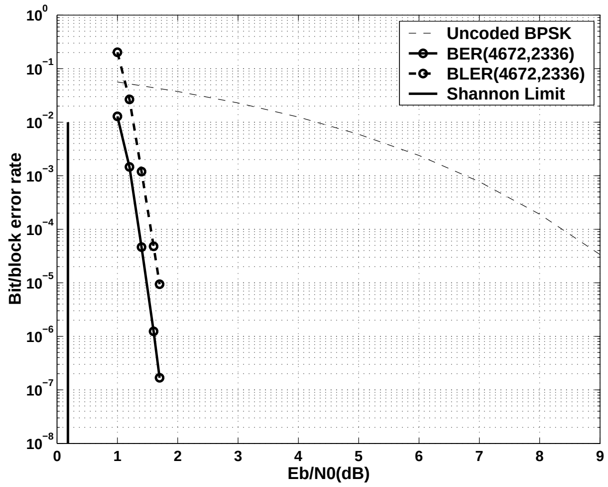


Fig. 25. Performance of the binary (4672,2336) QC-LDPC code given in Example 10 over the BI-AWGNC.

The average variable-node and check-node degrees are 3.5 and 7, respectively. Suppose we want to construct a code of length about 4600. To construct such a code, we choose the $(73, 2, 72)$ RS code $\mathcal{C}_b$ over GF(73) as the base code and construct a 73×73 array $\mathbf{H}_{rs,2}$ of 73×73 circulant permutation matrices based on the method presented in Section VI-A. Choose $d_v = 32$ and $d_c = 64$. Take a 32×64 subarray $\mathbf{H}_{rs,2}(32, 64)$ from $\mathbf{H}_{rs,2}$ as the base array for masking. It is a 2336×4672 matrix over GF(2) with column and row weights 32 and 64, respectively. Construct a masking matrix $\mathbf{W}(\gamma, \rho)$ by computer search with column and row weight distributions close to the degree distributions $v(X)$ and $c(X)$ given above. Masking the base array $\mathbf{H}_{rs,2}(32, 64)$ with $\mathbf{W}(32, 64)$, we obtain a masked 32×64 array $\mathbf{M}_{rs,2}(32, 64)$ of circulant permutation and zero matrices. The column and row weight distributions of $\mathbf{M}_{rs,2}(32, 64)$ are identical to $v(X)$ and $c(X)$. The null space of $\mathbf{M}_{rs,2}(32, 64)$ gives a $(4672, 2336)$ irregular RS-based QC-LDPC code. The performance of this code with iterative decoding using the SPA (50 iterations) is shown in Fig. 25. The code performs very well: at a BER of 10^{-6}, it is 1.6 dB from the Shannon limit. □

VIII. Conclusion and Open Problems

This paper provided fundamentals in the design of LDPC codes. The EXIT chart technique for determining near-optimal degree distributions for LDPC code ensembles was first discussed to provide a target for the code designer. The utility of representing codes by protographs and how this naturally leads to quasi-cyclic LDPC codes was also discussed, after which the EXIT chart technique was extended to the special case of protograph-based LDPC codes. Discussed next was several design approaches for LDPC codes which incorporate one or more accumulators, including quasi-cyclic accumulator-based codes. The second half the paper then switched to several algebraic LDPC code design techniques including codes based on finite geometries and codes whose designs are based on Reed-Solomon codes. The algebraic designs lead to cyclic, quasi-cyclic, and structured codes. Finally, the masking technique for converting regular quasi-cyclic LDPC codes to irregular codes was presented. While the paper focuses on the BI-AWGNC, as discussed in the paper, good BI-AWGNC codes tend to be universally good across many channels.

The ultimate goal in the LDPC code field is a situation that is analog of BCH or RS codes, that is, a straightforward design technique and a straightforward performance analysis. While this may be possible someday, in the short term, some of the open problems that are undergoing studies by researchers are as follows. It is well known that error-floors can be due to a small minimum distance or it can be the fault of the iterative decoder. Thus, there is a tremendous amount of research being undertaken to understand the floor phenomenon. Another issue is the design of short codes. As mentioned in Section II, decoding threshold prediction techniques assume an infinite codeword length and an infinite number of decoding iterations. This leads one to ask about threshold prediction for short codes with a finite number of iterations. Another problem being studied is generalized LDPC codes in which the single parity-check nodes and repetition nodes of Tanner graphs were replaced by more complex constraints. This was first considered by Tanner [2]. Other problems include lower bounding the minimum distance of an LDPC code and understanding the impact of cycle structure and distribution on an iterative decoder.

References

[1] R. G. Gallager, *Low-Density Parity-Check Codes*, M.I.T. Press, Cambridge, MA, 1963. (Also, R. G. Gallager, "Low density parity-check codes," *IRE Trans. Inform. Theory*, IT-8, pp. 21-28, Jan. 1962.)

[2] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inform. Theory*, vol. 27, pp. 533-547, Sept. 1981.

[3] D. MacKay and R. Neal, "Good codes based on very sparse matrices," *Cryptography and Coding, 5th IMA Conf., C. Boyd, Ed., Lecture Notes in Computer Science*, Oct. 1995.

[4] N. Alon and M. Luby, "A linear time erasure-resilient code with nearly optimal recovery," *IEEE Trans. Inform. Theory*, vol. 42, pp. 1732-1736, Nov. 1996.

[5] J. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A digital fountain approach to reliable distribution of bulk data," *Proc. ACM SIGCOMM '98*, Vancouver, BC, Canada, Jan. 1998, pp. 56-67.

[6] D. J. C. MacKay, "Good Error-Correcting Codes Based on Very Sparse Matrices," *IEEE Trans. Inform. Theory*, vol. 45, no. 2, pp. 399-431, 1999.

[7] S. Lin and D. J. Costello, Jr., *Error Control Coding: Fundamentals and Applications*, 2nd editon, Prentice-Hall, Upper Saddle River, NJ., 2004.

[8] W. E. Ryan, "An Introduction to LDPC Codes," *CRC Handbook for Coding and Signal Processing for Recording Systems*, Ed., B. Vasic and E. Kurtas, CRC Press, 2004.

[9] C. Jones, A. Matache, T. Tian, J. Villasenor, R. Wesel, "The universality of LDPC codes on wireless channels," in *Proc. Military Comm. Conf. (MILCOM)*, Oct. 2003.

[10] M. Franceschini, G. Ferrari, and R. Raheli, "Does the performance of LDPC codes depend on the channel?" in *Proc. Int. Symp. Inf. Theory and its Applns*, 2004.

[11] A. Ashikhmin, G. Kramer, and S. ten Brink, "Extrinsic information transfer functions: Model and erasure channel properties," *IEEE Trans. Inform. Theory*, vol. 50, pp. 2657-2673, Nov. 2004.

[12] F. Peng and W. E. Ryan and R. D. Wesel, "Surrogate channel design of universal LDPC codes," *IEEE Commun. Letters*, vol. 10, pp. 480-482, Jun. 2006.

[13] Z. Li, L. Chen, L. Zeng, S. Lin, and W. Fong, "Efficient encoding of low-density parity-check codes," *IEEE Trans. Commun.*, vol. 54, pp. 71-81, Jan. 2006.

[14] S. ten Brink, "Convergence Behavior of Iteratively Decoded Parallel Concatenated Codes," *IEEE Trans. Commun.*, vol. 49, pp. 1727-1737, Oct. 2001.

[15] S. ten Brink, G. Kramer, and A. Ashikhmin, "Design of low-density parity-check codes for modulation and detection," *IEEE Trans. Commun.*, vol. 52, pp. 670-678, Apr. 2004.

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

[17] D. Divsalar, S. Dolinar, and F. Pollara, "Iterative Turbo Decoder Analysis Based on Density Evolution," *IEEE Journal on Selected Areas in Communications*, vol. 19, pp. 891-907, May, 2001.

[18] H. El Gamal and A. R. Hammons, "Analyzing the Turbo Decoder Using the Gaussian Approximation," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 671-686, Feb. 2001.

[19] M. Ardakani and F. R. Kschischang, "A More Accurate One-Dimensional Analysis and Design of LDPC codes," *IEEE Trans. on Comm.*, Dec. 2004, pp 2106-2114.

[20] M. Tüchler, S. ten Brink, and J. Hagenauer, "Measures for tracing convergence of iterative decoding algorithms," *Proc. 4th IEEE/ITG Conf.on Source and Channel Coding*, Berlin, Germany, Jan. 2002.

[21] E. Sharon, A. Ashikhmin, and S. Litsyn, "EXIT functions for the Gaussian channel," *Proc. 40th Annu. Allerton Conf. Communication, Control, Computers*, Allerton, IL, Oct. 2003, pp. 972-981.

[22] E. Sharon, A. Ashikhmin, and S. Litsyn, "EXIT functions for continuous channels - Part I: Constituent codes," submitted, *IEEE Trans. Commun.*

[23] S. ten Brink and G. Kramer, "Design of repeat-accumulate codes for iterative detection and decoding," *IEEE Trans. Sig. Proc.*, vol. 51, pp. 2764-2772, Nov. 2003.

[24] R. Michael Tanner, "On quasi-cyclic repeat-accumulate codes," in *Proc. 37th Allerton Conf. on Communication, Control, and Computing*, Sept. 1999.

[25] Jun Xu, Lei Chen, Lingqi Zeng, Lan Lan, and Shu Lin, "Construction of low-density parity-check codes by superposition," *IEEE Trans. Commun.*, vol. 53, pp. 243-251, Feb. 2005.

[26] T. J. Richardson and R. L. Urbanke, "Multi-edge type ldpc codes," to appear, *IEEE Trans. Inf. Theory*. [Online]. Available: http://lthcwww.epfl.ch/

[27] J. Thorpe, "Low-Density Parity-Check (LDPC) Codes Constructed from Protographs," JPL INP, Tech. Rep., Aug. 2003, 42-154.

[28] H. Zhong and T. Zhang "Design of VLSI implementation-oriented LDPC codes," in *Proc. 58th Vehicular Technology Conf.*, Oct. 2003, pp. 670-673.

[29] M. M. Mansour, "High-performance decoders for regular and irregular repeat-accumulate codes," in *Proc. IEEE GLOBECOM*, Nov. 29-Dec. 3, 2004, pp. 2583-2588.

[30] G. Liva, *Block Codes Based on Sparse Graphs for Wireless Communication Systems*, Ph.D. thesis, Università degli Studi di Bologna, Italy, 2006.

[31] G. Liva and M. Chiani, "Extrinsic information transfer analysis for protograph-based LDPC codes", submitted, *IEEE Trans. Comm.*, 2006.

[32] D. Divsalar, H. Jin, and R. J. McEliece, "Coding theorems for Turbo-like codes," in *Proc. 36th Allerton Conf. on Communication, Control, and Computing*, Allerton, Illinois, Sept. 1998, pp. 201-210.

[33] H. Jin, A. Khandekar, and R. McEliece, "Irregular repeat-accumulate codes," in *Proc. International Symposium on Turbo codes and Related Topics*, Sept. 2000, pp. 1-8.

[34] M. Chiani and A. Ventura, "Design and performance evaluation of some high-rate irregular low-density parity-check codes, *Proc. IEEE Globecom*, Nov. 2001.

[35] M. Yang, Y. Li, and W. E. Ryan, "Design of efficiently encodable moderate-length high-rate irregular LDPC codes," *IEEE Trans. Commun.*, vol. 52, pp. 564-571, Apr. 2004.

[36] Y. Zhang and W. E. Ryan, "Structured IRA Codes: Performance Analysis and Construction," *IEEE Trans. Commun.*, 2006, to appear.

[37] Xiao Yu Hu and Evangelos Eleftheriou and Dieter Michael Arnold, "Progressive edge-growth Tanner graphs," Proc. 2001 GlobeCom Conf., San Antonio, Texas, Nov. 2001, pp. 995-1001.

[38] T. Tian and C. Jones and J. Villasenor and R. D. Wesel, "Characterization and selective avoidance of cycles in irregular LDPC codes," in *Proc. ICC'03*, May, 2003.

[39] M. Fossorier, "Quasi-Cyclic Low-Density Parity-Check Codes From Circulant Permutation Matrices," IEEE Trans. Inf. Theory, vol. 50., Aug. 2004, pp. 1788-1793.

[40] G. Liva, E. Paolini, and M. Chiani, "Simple Reconfigurable Low-Density Parity-Check Codes," *IEEE Commun. Letters*, vol. 9, pp. 258-260, March, 2005

[41] S. J. Johnson and S. R. Weller, "Constructions for irregular repeat-accumulate codes," in *Proc. IEEE Int. Sym. Inform. Theory*, Adelaide, Sept. 2005.

[42] L. Dinoi, F. Sottile, and S. Benedetto, "Design of variable-rate irregular LDPC codes with low error floor," *2005 IEEE Int. Conf. Comm.*, May 2005.

[43] A. Roumy, S. Guemghar, G. Caire, and S. Verdu, "Design methods for irregular repeat-accumulate codes," *IEEE Trans. Inform. Theory,* vol. 50, pp. 1711-1727, Aug. 2004.

[44] Y. Zhang, W. E. Ryan, and Y. Li, "Structured eIRA codes," in *Proc. 38th IEEE Asilomar Conf. on Signals, Systems, and Computers*, Pacific Grove, CA, Nov. 2004, pp. 7-10.

[45] A. Abbasfar, K. Yao, and D. Disvalar, "Accumulate Repeat Accumulate Codes," in *Proc. IEEE GLOBECOM*, Dallas, Texas, Nov. 2004.

[46] D. Divsalar, S. Dolinar, J. Thorpe, and C. Jones, "Constructing LDPC codes from simple loop-free encoding modules," in *Proc. IEEE International Conference on Communications*, May 2005.

[47] D. Divsalar, C. Jones, S. Dolinar, and J. Thorpe, "Protograph based LDPC codes with minimum distance linearly growing with block size," in *Proc. IEEE GLOBECOM*, Nov. 2005.

[48] *Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications*, ETSI, EN 302 307, 2004.

[49] D. Divsalar, S. Dolinar, and J. Thorpe, "Accumulate-repeat-accumulate-accumulate-codes", *Proc. 60th Vehicular Technology Conf.*, Sept. 2004, pp. 2292-2296.

[50] Y. Kou, S. Lin, and M. Fossorier, "Low density parity check codes based on finite geometries: a rediscovery and new results," *IEEE Trans. Inform. Theory,* vol. 47, no.11, pp. 2711-2736, Nov. 2001.

[51] H. B. Mann, *Analysis and Design of Experiments*. New York: Dover, 1949

[52] A. P. Street and D. J. Street, *Combinatorics of Experimental Design*. Oxford, UK: Clarendon Press, 1987.

[53] H. Tang, J. Xu, Y. Kou, S. Lin, and K. Abdel-Ghaffar, " On algebraic construction of Gallager and circulant low density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 50, no.6 , pp. 1269-1279, June 2004.

[54] L. Chen, J. Xu, I. Djurdjevic, and S. Lin, " Near Shannon limit quasi-cyclic low-density parity-check codes," *IEEE Trans. Commun.*, vol. 52, no. 7, pp. 1038-1042, July 2004.

[55] H. Tang, J. Xu, S. Lin, and K. A. S. Abdel-Ghaffar, " Codes on finite geometries," *IEEE Trans. Inform. Theory*, vol. 51, no. 2, pp. 572–596, Feb. 2005.

[56] Y. Y. Tai, L. Lan, L. Zeng, S. Lin, and K. Abdel-Ghaffar, "Algebraic construction of quasi-cyclic LDPC codes for the AWGN and erasure channels," *IEEE Trans. Commun.*, to appear.

[57] J. Xu, L. Chen, I. Djurdjevic, S. Lin, and K. Abdel-Ghaffar, "Construction of regular and irregular LDPC codes: geometry decomposition and masking," submitted to *IEEE Trans. Inform. Theory* in 2004 (in revision).

[58] Juntan Zhang and Marc P. C. Fossorier, "A modified weighted bit-flipping decoding of low-density parity-check codes," *IEEE Commun. Letters*, vol. 8, no. 3, March 2004

[59] Zhenyu Liu and Dimitris A. Pados, " A decoding algorithm for finite-geometry LDPC codes," *IEEE Trans. on Commun.* , vol. 53, no. 3, March 2005

[60] Ming Jiang, Chunming Zhao, Zhihua Shi, and Yu Chen, "An improvement on the modified weighted bit flipping decoding algorithm for LDPC codes," *IEEE Commun. Letters*, vol.9, no. 9, Sept. 2005

[61] I. Djurdjevic, J. Xu, K. Abdel-Ghaffar, and S. Lin, "Construction of low-density parity-check codes based on Reed-Solomon codes with two information symbols," *IEEE Commun. Lett.*, vol. 8, no. 7, pp. 317-319, July 2004.

[62] Z. -W. Li, L. Chen, L. -Q. Zeng, S. Lin, and W. Fong, "Efficient encoding of quasi-cyclic low-density parity-check codes," *IEEE Trans. Commun.*, vol.54, no. 1, pp. 71-81, Jan. 2006.

[63] I. S. Reed and G. Solomon,"Polynomial codes over certain fields," *J. Soc. Ind. Appl. Math.*, 8: 300-304, Jun. 1960.

[64] L. Lan, L. -Q Zeng, Y. Y. Tai, L. Chen, S. Lin, and K. Abdel-Ghaffar,"Construction of quasi-cyclic LDPC codes for AWGN and binary erasure channels: a finite field approach," *IEEE Trans. Inform. Theory,* in revision, 2006.

[65] L. Lan, Y. Y. Tai, L. Chen, S. Lin, and K. Abdel-Ghaffar,"A trellis-based method for removing cycles for bipartite graphs and construction of low density parity check codes," *IEEE Commun. Lett.*, vol. 8, no. 7, pp. 443-445, Jul. 2004.

[66] X. -Y. Hu, E. Eleftheriou, and D. M. Arnold,"Regular and irregular progressive edge-growth Tanner graphs," *IEEE Trans. Inform. Theory*, vol 51, no.1, pp. 386-398, Jan. 2005.

Gianluigi Liva was born in Spilimbergo, Italy. He received the M.S. degree in Electrical Engineering, in 2002, and the Ph.D. degree, in 2006, from the University of Bologna, Bologna, Italy. He is currently working at the Institute of Communications and Navigation of the German Aerospace Center (DLR) in Munich. His research interests include the design and the analysis of error correcting codes based on sparse graphs for space communication systems.

Shumei Song was born in Henan, China. She received the B.S.E.E. degree from Tsinghua University, Beijing, China, in 2000, and the M.S.E.E degree from Peking University, Beijing, China, in 2003. She is currently working toward the Ph.D degree in communication and coding theory at the University of California, Davis.

Lan Lan received the B.E. and M.E. degrees from the University of Electronic Science and Technology of China, Chengdu, China, in 1998 and 2001, and the Ph.D. degree in electrical engineering from the University of California, Davis, in 2006. She is currently working for Keyeye communications company as a DSP design engineer. Her research interests include error-control coding techniques and their application in digital communications and digital storage systems.

Yifei Zhang received the B.E. and M.E. degrees in electrical and communication systems from Beijing University of Posts and Telecommunications, Beijing, China, in 1998 and 2001, respectively. She is currently working toward the Ph.D. degree in the Department of Electrical and Computer Engineering, University of Arizona, Tucson. Her research interests include error-control coding and its implementation for digital communications and data storage systems.

Shu Lin (S'62-M'65-SM'78-F'80-LF'00) received the B.S.E.E. degree from the National Taiwan University, Taipei, Taiwan, in 1959, and the M.S. and Ph.D. degrees in electrical engineering from Rice University, Houston, TX, in 1964 and 1965, respectively.

In 1965, he joined the Faculty of the University of Hawaii, Honolulu, as an Assistant Professor of Electrical Engineering. He became an Associate Professor in 1969 and a Professor in 1973. In 1986, he joined Texas A&M University, College Station, as the Irma Runyon Chair Professor of Electrical Engineering. In 1987, he returned to the University of Hawaii. From 1978 to 1979, he was a Visiting Scientist at the IBM Thomas J. Watson Research Center, Yorktown Heights, NY, where he worked on error control protocols for data communication systems. He spent the academic year of 1996-1997 as a Visiting Professor at the Technical University of Munich, Munich, Germany. He retired from University of Hawaii in 1999 and he is currently an Adjunct Professor at University of California, Davis. He has published numerous technical papers in IEEE TRANSACTIONS and other refereed journals. He is the author of the book, *An Introduction to Error-Correcting Codes* (Englewood Cliff, NJ: Prentice-Hall, 1970). He also co-authored (with D. J. Costello) the book, *Error Control Coding: Fundamentals and Applications* (Upper Saddle River, NJ: Prentice-Hall, 1st edition, 1982, 2nd edition, 2004), and (with T. Kasami, T. Fujiwara, and M. Fossorier) the book, *Trellises and Trellis-Based Decoding Algorithms,* (Boston, MA: Kluwer Academic, 1998). His current research areas include algebraic coding theory, coded modulation, error control systems, and satellite communications. He has served as the Principle Investigator on 32 research grants.

Dr. Lin is a Member of the IEEE Information Theory Society and the Communication Society. He served as the Associate Editor for Algebraic Coding Theory for the IEEE TRANSACTIONS ON INFORMATION THEORY from 1976 to 1978, and as the Program Co-Chairman of the IEEE International Symposium of Information Theory held in Kobe, Japan, in June 1988. He was the President of the IEEE Information Theory Society in 1991. In 1996, he was a recipient of the Alexander von Humboldt Research Prize for U.S. Senior Scientists and a recipient of the IEEE Third-Millennium Medal, 2000.

William E. Ryan received the Ph.D. degree in electrical engineering from the University of Virginia (Charlottesville, VA) in 1988 after receiving the B.S. and M.S. degrees from Case Western Reserve University and the University of Virginia, respectively, in 1981 and 1984.

After receiving the Ph.D. degree Prof. Ryan held positions in industry for five years, first at The Analytic Sciences Corporation, then at Ampex Corporation, and finally at Applied Signal Technology. From 1993 to 1998, he was an assistant professor and then associate professor in the Department of Electrical and Computer Engineering at New Mexico State University, Las Cruces, NM. From 1998 to present, he has been on the faculty in the Department of Electrical and Computer Engineering at the University of Arizona, Tucson, AZ, first as an associate professor and then as full professor.

Prof. Ryan has over 75 publications in the leading conferences and journals in the area of communication theory and channel coding. He is also preparing the textbook *Iteratively Decodable Codes: Construction and Decoding* (Shu Lin, co-author) to be published by Cambridge University Press. His research interests are in coding and signal processing with applications to magnetic data storage and wireless data communications. He was an associate editor for the IEEE Transactions on Communications from 1998 through 2005. He is a Senior Member of the IEEE.