# Quasi-Cyclic Generalized LDPC Codes with Low Error Floors

Gianluigi Liva, *Member, IEEE*, William E. Ryan, *Senior Member, IEEE*, and Marco Chiani, *Senior Member, IEEE* 

Abstract—In this paper, a novel methodology for designing structured generalized LDPC (G-LDPC) codes is presented. The proposed design results in quasi-cyclic G-LDPC codes for which efficient encoding is feasible through shift-register-based circuits. The structure imposed on the bipartite graphs, together with the choice of simple component codes, leads to a class of codes suitable for fast iterative decoding. A pragmatic approach to the construction of G-LDPC codes is proposed. The approach is based on the substitution of check nodes in the protograph of a low-density parity-check code with stronger nodes based, for instance, on Hamming codes. Such a design approach, which we call LDPC code doping, leads to low-rate quasi-cyclic G-LDPC codes with excellent performance in both the error floor and waterfall regions on the additive white Gaussian noise channel.

Index Terms—Channel coding, error control codes, generalized LDPC codes, graph-based codes, LDPC codes.

### I. Introduction

HEREAS it is now easy to design *long* low-density parity-check (LDPC) codes with good decoding thresholds and low floors [1] [2], it is particularly difficult to design short, low-rate LDPC codes (i.e., R < 1/2) which possess both low error-rate floors and good decoding thresholds [3]–[5]. Such codes have applications in wireless and deep-space communications. In this paper we consider an alternative code structure which facilitates the design of good short (or moderate-length), low-rate iteratively decodable codes. The adjective "good" here refers to both good thresholds and low floors. Specifically, we focus on generalized LDPC (G-LDPC) codes [6] with a quasi-cyclic structure [7]. G-LDPC codes represent the broadest known class of iteratively decodable codes, comprising many code classes, including low-density parity-check codes (LDPCC) and turbo codes [8].

In the Tanner graph of a G-LDPC code, depicted in Fig. 1, constraint nodes (CNs) are more general than single parity-check (SPC) constraints. In fact, the variable nodes (VNs) may

Paper approved by F. Fekri, the Editor for Coding Theory of the IEEE Communications Society. Manuscript received December 13, 2005; revised October 17, 2006. This research was supported, in part, by the Ministero dell'Istruzione, Università e della Ricerca Scientifica (MIUR), by the University of Bologna, and by NASA-JPL grant 1264726. This paper was presented in part at the International Symposium on Turbo Codes and Related Topics, Munich, 2006.

G. Liva was with WiLAB/DEIS, University of Bologna. Currently he is with the Institute for Communications and Navigation, German Aerospace Centre (DLR), Oberpfaffenhofen, 82234 Weßling, Germany (e-mail: gianluigi.liva@dlr.de).

W. E. Ryan is with the Department of Electrical and Computer Engineering, University of Arizona, Tucson, AZ 85721 (e-mail: ryan@ece.arizona.edu).

M. Chiani is with WiLAB/DEIS, University of Bologna, Via Venezia 52, 47023 Cesena, Italy (e-mail: mchiani@deis.unibo.it).

Digital Object Identifier 10.1109/TCOMM.2008.050600.

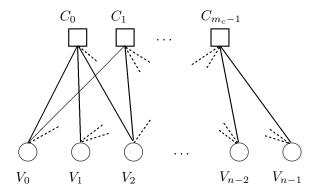


Fig. 1. Bipartite graph for a generalized LDPC code.

represent more complex codes than the repetition codes they represent in standard LDPC codes [9], [10], but we shall not consider such G-LDPC codes here. In the figure, the CNs are the square nodes marked by  $C_0, C_1, \ldots, C_{m_c-1}$  to signify the  $m_c$  code constraints placed on the code bits associated with the CNs. Note that, while a bipartite graph gives a complete description of an LDPC code, for a generic G-LDPC code the specifications of the component (linear) block codes are also required. Let us define m as the summation of all the redundancies introduced by the  $m_c$  CNs, namely,

$$m = \sum_{i=0}^{m_c - 1} (n_i - k_i),$$

where  $(n_i, k_i)$  represent the parameters of the component code for constraint node  $C_i$  whose parity-check matrix is denoted by  $\mathbf{H}_i$ . Thus, the rate of a length-n G-LDPC code satisfies

$$R \ge 1 - \frac{m}{n}$$

with equality only if all of the check equations derived from the CNs are linearly independent.

There are several compelling reasons for studying G-LDPC codes, including:

- Design of low-rate codes. Good low-rate (R ≤ 1/2) codes are usually obtained by a proper design of convolutional turbo codes. The construction of good low-rate LDPC codes is still difficult because good waterfall performance usually leads to high error floors. A wider search in the G-LDPC code ensemble permits the design of good low-rate codes. In fact, extremely low rate G-LDPC codes with remarkable performance have been proposed in [11] and [12].
- Low-error-floor codes. While recent advances in the design of LDPC codes have led to codes with particularly

low error floors [13]–[15], the requirement of low-rate codes in some applications poses several challenges to the channel coding community. G-LDPC codes take advantage of strong component codes to achieve very low error floors. Further, the G-LDPC Tanner graph adjacency matrix is in general simpler than that for an LDPC code, making it easier to avoid cycles and other impediments to effective iterative decoding.

 Decoder design. It is not yet completely clear whether turbo codes or LDPC codes offer the lower decoder complexity. These two code classes represent two extreme cases of G-LDPC codes: turbo codes (typically) have two complex CNs, whereas LDPC codes have many simple CNs. A G-LDPC code can be designed which exploits the decoding advantages of these two extreme cases. An example of a G-LDPC code that strikes such a balance is a turbo product code (also called block turbo code [16]), which permits the use of low-complexity Chasetype decoding.

Excluding the LDPC codes and turbo codes mentioned above, only a few classes of G-LDPC codes have been studied to date [6], [12], [17]–[20]. Our work differs from these earlier works in that we focus on low floors, low rates, a quasi-cyclic (QC) structure, and irregularity. The QC characteristic facilitates encoder and decoder implementations and irregularity improves the decoding threshold (the waterfall characteristic).

The paper is organized as follows. Section II introduces some of the concepts and notation that underlie the QC G-LDPC codes that we study. Section III gives a description of proposed G-LDPC code design techniques and provides performance results of some low-rate codes. Concluding remarks are contained in Section IV.

## II. QUASI-CYCLIC G-LDPC CODES

Let  $V = \{V_j\}_{j=0}^{n-1}$  be the set of n VNs and  $C = \{C_i\}_{i=0}^{m_c-1}$  be the set of  $m_c$  CNs in the bipartite graph of a G-LDPC code (Fig. 1). The connection between the nodes in V and C can be summarized in an  $m_c \times n$  adjacency matrix  $\Gamma$ . In this section, we first introduce the relationship between the adjacency matrix  $\Gamma$  and the parity-check matrix  $\Gamma$  for a G-LDPC code. We then show how  $\Gamma$  can be structured in order for the code to be quasi-cyclic.

While  $\Gamma$  for an LDPC code serves as its parity-check matrix, for a G-LDPC code, to obtain  $\mathbf{H}$ , one requires also knowledge of the parity-check matrices  $\mathbf{H}_i$  of the CN component codes. The  $n_i$  1's in the i-th binary row of  $\Gamma$  indicate which of the n G-LDPC code bits are constrained by constraint node  $C_i$ . Because the parity checks corresponding to  $C_i$  are represented by the rows of  $\mathbf{H}_i$ ,  $\mathbf{H}$  is easily obtained by replacing the 1's in row i of  $\Gamma$  by the columns of  $\mathbf{H}_i$ , for all i. Note that this procedure allows for the case when  $C_i$  is a SPC code, in which case  $\mathbf{H}_i$  has only one row. Note also that  $\mathbf{H}$  will be of size  $m \times n$ , where  $m = \sum_{i=0}^{m_c-1} m_i$ , with  $m_i = n_i - k_i$ .

To design a quasi-cyclic G-LDPC code, that is, to obtain a matrix **H** that is an array of circulant permutation matrices, we first review the concept of protograph [21]–[24]. A protograph is a relatively small bipartite graph from which a larger graph can be obtained by a copy-and-permute procedure:

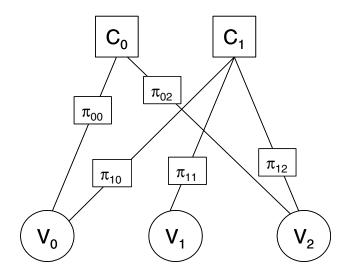


Fig. 2. Protograph with implied expansion (copy and permute) along edges. Each permutation matrix is  $q \times q$  and each node is interpreted to be a q-vector of nodes of the same type.

the protograph is copied q times, and then the edges of the individual replicas are permuted among the q replicas (under restrictions described below) to obtain a single, large bipartite graph. Suppose the protograph possesses  $n_p$  variable nodes and  $m_p$  constraint nodes. Then the derived graph will consist of  $n=n_pq$  variable nodes and  $m_c=m_pq$  constraint nodes.

Note that the edge permutations cannot be arbitrary. In particular, the nodes of the protograph are labelled so that if VN  $V_j$  is connected to CN  $C_i$  in the protograph, then VN  $V_j$  in a replica can only connect to one of the q replicated CNs  $C_i$ . Doing so preserves the decoding threshold properties of the protograph while permitting the design of quasi-cyclic codes. In particular, if the edge permutations are organized in a cyclic manner such that the final adjacency matrix is an array of circulant permutation matrices, the code will be quasi-cyclic (demonstrated below). In this case,  $\Gamma$  is an  $m_p \times n_p$  array of circulant permutation matrices of the form

$$\mathbf{\Gamma} = \begin{bmatrix} \pi_{0,0} & \pi_{0,1} & \cdots & \pi_{0,n_p-1} \\ \pi_{1,0} & \pi_{1,1} & \cdots & \pi_{1,n_p-1} \\ & & \ddots & \\ \pi_{m_p-1,0} & \pi_{m_p-1,1} & \cdots & \pi_{m_p-1,n_p-1} \end{bmatrix},$$

where each  $\pi_{\mu,\nu}$  is either a  $q \times q$  circulant permutation matrix or a  $q \times q$  zero matrix. Note that  $\Gamma$  is a  $m_p q \times n_p q = m_c \times n$  binary matrix.  $\Gamma$  is constructed by substituting circulant permutation matrices for the 1's in the protograph adjacency matrix,  $\Gamma_p$ , in such a manner that short cycles are avoided. The 0's in  $\Gamma_p$  are replaced by all-zero column vectors of appropriate lengths. As demonstrated in Fig. 2, the substitution of circulant permutation matrices for 1's in  $\Gamma_p$  effectively applies the copy-and-permute procedure to the protograph corresponding to  $\Gamma_p$ .

Example 1. Consider a protograph with the  $m_p \times n_p = 2 \times 3$  adjacency matrix

$$\mathbf{\Gamma}_p = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix} .$$

Let q=2 and expand  $\Gamma_p$  by substituting  $2\times 2$  circulant permutation matrices for each 1 in  $\Gamma_p$ . The result is

$$oldsymbol{\Gamma} = \left( egin{array}{cccccc} 1 & 0 & & 0 & 0 & & 1 & 0 \\ 0 & 1 & & 0 & 0 & & 0 & 1 \\ 1 & 0 & & 0 & 1 & & 0 & 1 \\ 0 & 1 & & 1 & 0 & & 1 & 0 \end{array} 
ight) \; ,$$

so that  $m_c = m_p q = 4$  and  $n = n_p q = 6$ . Suppose now that

$$\mathbf{H}_1 = \begin{bmatrix} 1 & 1 \end{bmatrix}$$

and

$$\mathbf{H}_2 = \left[ \begin{array}{ccc} 1 & 1 & 0 \\ 1 & 0 & 1 \end{array} \right].$$

Then, upon replacing the 1's in  $\Gamma$  by the corresponding columns of  $\mathbf{H}_1$  and  $\mathbf{H}_2$ , the  $\mathbf{H}$  matrix for the G-LPDC code is found to be

$$\mathbf{H} = \begin{bmatrix} \mathbf{1} & 0 & & 0 & 0 & & \mathbf{1} & 0 \\ 0 & \mathbf{1} & & 0 & 0 & & 0 & \mathbf{1} \\ \mathbf{1} & 0 & & 0 & \mathbf{1} & & 0 & \mathbf{0} \\ \mathbf{1} & 0 & & 0 & \mathbf{0} & & 0 & \mathbf{1} \\ 0 & \mathbf{1} & & \mathbf{1} & 0 & & \mathbf{0} & 0 \\ 0 & \mathbf{1} & & \mathbf{0} & 0 & & \mathbf{1} & 0 \end{bmatrix},$$

where the inserted columns of  $\mathbf{H}_1$  and  $\mathbf{H}_2$  are highlighted in bold. Note that because the  $\mu$ -th row of  $\mathbf{\Gamma}_p$ ,  $\mu=0,1,\ldots,m_p-1$ , corresponds to constraint  $C_\mu$  (matrix  $\mathbf{H}_\mu$ ) the  $\mu$ -th block-row (the  $\mu$ -th group of q binary rows) within  $\mathbf{\Gamma}$  corresponds to constraint  $C_\mu$  (matrix  $\mathbf{H}_\mu$ ). In the form given above, it is not obvious that  $\mathbf{H}$  corresponds to a quasi-cyclic code. However, if we permute the last four rows of  $\mathbf{H}$ , we obtain

$$\mathbf{H}' = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

The permutation used on the last four rows of  $\mathbf{H}$  can be considered to be a mod- $m_2$  de-interleave of the rows, where  $m_2=2$  is the number of rows in  $\mathbf{H}_2$ . No de-interleaving was necessary for the first two rows of  $\mathbf{H}$  because  $m_1=1.\blacksquare$ 

The fact that in general a code's  $\mathbf{H}$  matrix will be an array of permutation matrices (after appropriate row permutation) whenever  $\Gamma$  is an array of permutation matrices should be obvious from the construction of  $\mathbf{H}$  from  $\Gamma$  in the above example, so we omit the proof.

A protograph can possess parallel edges, i.e., two nodes can be connected by more than one edge. The copy-and-permute procedure (equivalently, substitution of permutation matrices in  $\Gamma_p$ ) must eliminate such parallel connections in order to obtain a derived graph that meaningfully corresponds to a parity-check matrix.

Example 2. Change the upper-left element of  $\Gamma_p$  of the previous example to "2" so that

$$\mathbf{\Gamma}_p = \begin{pmatrix} 2 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix} .$$

Thus, there are two edges connecting VN  $V_0$  and CN  $C_0$ . A possible q-fold expansion of  $\Gamma_p$  is

$$oldsymbol{\Gamma} = \left[ egin{array}{cccccc} \pi_0 & \pi_1 & & 0 & 0 & & 0 & \pi_2 \\ \pi_3 & \pi_4 & & 0 & 0 & & \pi_5 & 0 \\ & & & & & & & & & & & \\ \pi_6 & 0 & & 0 & \pi_8 & & & \pi_{10} & 0 \\ 0 & \pi_7 & & \pi_9 & 0 & & 0 & \pi_{11} \end{array} 
ight],$$

where the permutation matrices are  $\frac{q}{2} \times \frac{q}{2}$  and are selected to avoid short cycles. **H** is then obtained by replacing the 1's in the rows of  $\Gamma$  by the columns of  $\mathbf{H}_1$  and  $\mathbf{H}_2$ ; the first q rows of  $\Gamma$  correspond to  $\mathbf{H}_1$  and the second q rows correspond to  $\mathbf{H}_2$ .

Note that a G-LDPC code has a parity-check matrix which is 4-cycle free if its adjacency matrix is 4-cycle free and the component codes possess parity-check matrices which are 4-cycle free. We remark that this is a sufficient condition, not a necessary one.

# III. PROTOGRAPH-BASED G-LDPC CODE DESIGNS

In this section, we consider the design of low-rate G-LDPC codes with low floors. The design of G-LDPC codes can be based on the same principles as the design of LDPC codes. For a given code rate, a density evolution (DE) analysis [2] can be exploited to derive the decoding thresholds of bipartite graphs whose nodal distributions are optimized with the aim of minimizing the signal-to-noise ratio (SNR) threshold. However, G-LDPC codes deal with a larger set of possible node types than do LDPC codes. Thus, the search for a good mixture of component codes can be impractical. To facilitate the code design, one can place restrictions on the number of component codes that can be used or on the regularity of the Tanner graph. A case in point is the first code we present in this section, a rate-1/2 code with ostensibly only two constraint types. In fact, only one constraint type is used: the two constraint nodes differ only in the component code bit orders.

#### A. A Rate-1/2 G-LDPC Code

To demonstrate the performance advantage of G-LDPC codes, we start with a very simple protograph: 2 CNs, 15 VNs, with both CNs connected to each of the 15 VNs. Thus, the CNs have degree 15 and the VNs have degree 2. Both CNs correspond to the (15,11) Hamming code constraint, but with different code bit orders. Specifically, protograph CN  $C_0$  is described by the parity-check matrix

and protograph CN  $C_1$  is described by

$$\mathbf{H}_1 = [\mathbf{M}_2 \ \mathbf{M}_1]. \tag{2}$$

Next, the protograph is replicated q=146 times, yielding a derived graph with 2190 VNs and 292 Hamming CNs, 146 of which are described by (1) and 146 of which are described by (2). Because the number of parity bits is  $m=292 \cdot (15-11)=1168$ , the resulting code has as parameters

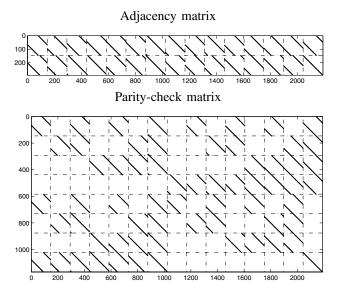


Fig. 3. Adjacency matrix of the (2190,1022) G-LDPC code and its block-circulant parity-check matrix **H**.

(2190, 1022). The connections between the VNs and the CNs are given by the adjacency matrix  $\Gamma$  at the top of Fig. 3, which was chosen simply to avoid 4-cycles and 6-cycles in the Tanner graph corresponding to that matrix. Therefore, the girth of the Tanner graph corresponding to  $\Gamma$  is 8. The H matrix (with appropriately re-ordered rows) is given in the bottom of that figure. Observe that an alternative approach for obtaining the re-ordered matrix H is to replace each 1 in the rows of the matrix in (1) (the matrix in (2)) by the corresponding permutation matrices of the first (second) block row of the adjacency matrix in Fig. 3 and to then stack the first resulting matrix on the second.

We may obtain a quasi-cyclic rate-1/2 (2044, 1022) G-LDPC code by puncturing the first 146 bits of each (2190, 1022) codeword. Observe that this corresponds to puncturing a single VN in the code's protograph and the first column of circulants of  $\Gamma$ . The frame error rate (FER) performance of this rate-1/2 code on the binary-input AWGN channel is depicted in Fig. 4. To achieve sufficient reliability, 30 error events were collected for each  $E_b/N_0$  value; only at the highest SNR point  $(E_b/N_0 = 2.5 \text{ dB})$  was the decoder stopped after 15 error events (obtained after more than 300 million transmitted codewords). For the simulations, the maximum number of iterations was set to  $I_{max} = 50$ . The G-LDPC code does not display a floor down to FER  $\simeq 5 \cdot 10^{-8}$ . As shown in the figure, the code's performance is within 1 dB from the random coding bound [25] for (2044,1022) block codes. We point out that the G-LDPC code's software decoder developed for the simulations was able to process approximatively 200K bps on a 3 GHz Pentium IV platform. This observation supports the potential of high-speed hardware decoding of G-LDPC codes based on simple Hamming codes.

#### B. Two G-LDPC Code Families Based on ARA Protographs

As mentioned at the beginning of this section, G-LDPC code design can be based on density evolution. Or, as seen in the previous example, it is possible to design good codes

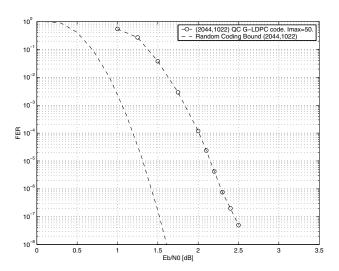
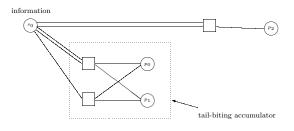


Fig. 4. Frame error rate for the (2044,1022) quasi cyclic G-LDPC code, compared to the random coding bound.  $I_{max}$  was set to 50.

through techniques that are a combination of coding "art" and coding "science." An approach to G-LDPC code design that differs from either of these techniques is based on what we call *code doping*. Code doping was introduced in [12] and consists of the substitution of selected SPC nodes in an LDPC code Tanner graph with more powerful nodes based on less trivial linear block codes. Code doping is conveniently applied at the protograph level: a conventional SPC node in a protograph can be replaced by a more powerful constraint node. The doping will be inherited by the derived graph which will possess q replicas of the constraint node. We begin with LDPC protographs characterized by good decoding thresholds and substitute a fraction of the SPC nodes with stronger constraint nodes, and then we evaluate the resulting graph's decoding performance through density evolution and computer simulations. The choice of the fraction of doped nodes and of the component codes takes into account requirements on the desired code rate and block length. The set of component codes used for doping is limited to codes with low decoding complexity. This pragmatic approach is less thorough than the DE-based search used in the design of LDPC codes. However, it permits the design of G-LDPC codes with both good decoding thresholds and low floors.

Among LDPC codes possessing a protograph representation, accumulate-repeat-accumulate (ARA) codes [15] are among the most interesting for code doping purposes. ARA codes are characterized by good decoding thresholds and simple protograph representations. As an example of code doping on ARA code, we choose the protograph of a rate-1/4 ARA code [15], depicted in Fig. 5 (top). This code has a decoding threshold at  $(E_b/N_0)^*=0.34$  dB, which can be easily computed from the protograph. (In the Appendix we present the EXIT chart technique for obtaining thresholds for G-LDPC codes). By puncturing the nodes corresponding to  $s_0$ , the code rate becomes 1/3, and the threshold is reduced to  $(E_b/N_0)^*=-0.048$  dB, less than 0.5 dB away from the capacity for the binary-input AWGN channel.



Code rate 1/4 - Threshold at  $E_b/N_0 = 0.34dB$ 

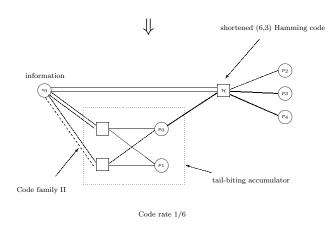


Fig. 5. At the top, the protograph for the rate 1/4 ARA code, and at the bottom, the protographs for the rate 1/6 code families,  $\mathcal{C}^I$  (without dotted edge) and  $\mathcal{C}^{II}$  (with dotted edge).

1) G-LDPC Code Family  $C^I$ : For this protograph, the sub-graph comprising nodes  $s_0$ ,  $p_0$  and  $p_1$  represents the protograph of an irregular repeat-accumulate (IRA) code [13], [26]. The remaining check node is connected to node  $p_2$ , which is in effect precoded by an accumulator [15]. Thus, the overall scheme can be considered to be the concatenation of an accumulator and an IRA code (hence, "ARA"). We chose to dope this protograph by replacing the top SPC node with a more powerful code. In order to maintain a reasonable decoding complexity, we chose a shortened (6,3) Hamming code. This necessitated the addition of another input to that node (we chose  $p_0$ ) and two outputs,  $p_3$  and  $p_4$ , as depicted in Fig. 5 (bottom). This yields a protograph for a rate-1/6code which has a decoding threshold 0.77 dB from the rate-1/6 capacity limit. Higher code rates can be achieved by puncturing the rate-1/6 protograph. As an example, puncturing node  $s_0$  yields a rate-1/5 code whose threshold is only 0.42 dB from the rate-1/5 capacity limit. Also, puncturing nodes  $s_0$  and  $p_4$  yields a rate-1/4 code whose threshold is only 0.48 dB from the rate-1/4 capacity limit. In the following, the codes derived from this protograph will be referred to as G-LDPC Code Family I (denoted  $C^I$ ).

2) G-LDPC Code Family  $\mathcal{C}^{II}$ : The protograph in Fig 5 (bottom) can be modified by adding one edge (dashed line), connecting  $s_0$  to a constraint node of the IRA sub-graph. This rate-1/6 protograph yields a code whose threshold is 0.74 dB from its respective limit. A rate-1/5 code whose threshold is only 0.62 dB from the rate-1/5 capacity limit is obtained by puncturing node  $s_0$ . A rate-1/4 code whose

threshold is only 0.56 dB from the rate-1/4 capacity limit is obtained by puncturing nodes  $s_0$  and  $p_4$ . In the following, the codes derived from this protograph will be referred to as *G-LDPC Code Family II* ( $\mathcal{C}^{II}$ ). Although this modified protograph possesses slightly worse thresholds than that of  $C^I$ , it leads to codes with lower floors, as we now demonstrate.

3) Simulation Results for Families C<sup>I</sup> and C<sup>II</sup>: Simulation results are provided here for the binary-input AWGN channel. In all cases, adjacency matrices that are arrays of circulant permutation matrices were used so that the G-LDPC codes are quasi-cyclic. The decoder used in each case is the standard belief-propation algorithm with maximum a posteriori decoding at each VN and CN (see a decription for the G-LDPC case in [12]). Thus, for the Hamming constraint nodes, soft-outputs are computed using a BCJR decoder [27] working on the BCJR trellis [27], [28] of the component code.

A rate-1/6 code with information block length k=1024 was designed according the  $\mathcal{C}^{II}$  protograph. The BER and FER performance curves are presented in Fig. 6, with a maximum of  $I_{max}=200$  decoding iterations. For comparison with the FER curve, the Gallager random coding bound [25] for (6144,1024) codes was added to the figure. The code exhibits excellent performance, within 0.9 dB from the bound down to FER  $\simeq 3\cdot 10^{-6}$ , owing to the large minimum distance of the code, which is (with high probability)  $d_{min}\simeq 203$  (see the next subsection).

A rate 1/5-code was obtained by puncturing the variable nodes of the previous code corresponding to systematic bits, as described above. The performance of the (5120,1024) G-LDPC code is presented in Fig. 7, and does not show an error floor down to FER  $\simeq 10^{-6}$ . Again, the FER curve is about 0.9 dB from the bound. Similar results are achieved for a rate-1/4 code obtained as described above, as shown in Fig. 8. In this case, the FER curve is about 0.7 dB from the bound.

Two quasi-cyclic rate-1/5 codes with input block length k=1792 have been constructed from the  $\mathcal{C}^I$  and  $\mathcal{C}^{II}$  protographs, the higher rate obtained by puncturing the systematic bits. In Fig. 9, the FER curves (with  $I_{max}=200$ ) for these two codes are compared to the rate 1/6 convolutional turbo code standardized by CCSDS [29], for which k = 1786. For the turbo code, the maximum number of iterations was set to 10 [30]. That value is usually sufficient to achieve a coding gain near the maximum possible for such turbo schemes. The rate-1/5 code based on the  $C^I$  protograph has exceptional performance in the waterfall region: at a  $FER = 10^{-3}$  the code is less than 0.5 dB away from the random coding bound, and exhibits almost the same performance as the rate 1/6turbo code (yet the G-LDPC code has a higher rate). The  $\mathcal{C}^{II}$  protograph has an additional edge relative to the  $\mathcal{C}^{I}$ protograph, resulting in a degraded threshold, however, the introduction of this edge lowers the error floor. As seen in Fig. 9, the  $C^I$  code is 0.2 dB superior to the  $C^{II}$  code at FER =  $10^{-3}$ , but it has an error floor just below this error rate. The  $\mathcal{C}^{II}$  code reaches a frame error rate close to  $10^{-6}$ without a floor, within 0.8 dB from the bound.

In Fig. 6, we compare the performance of the (6144,1024)  $C^{II}$  G-LDPC code using two different decoding algorithms. The first decoder was a belief-propagation decoder applied to the code's Tanner graph based on the adjacency matrix.

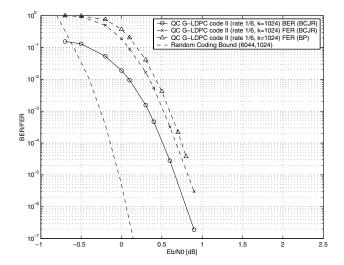


Fig. 6. Frame and bit error rates (with BCJR-based decoding for Hamming nodes) of rate 1/6 (6144,1024) G-LDPC code II, compared to the random coding bound.  $I_{max}$  was set to 200. The performance degradation using BP-based decoding at Hamming nodes can be evaluated from the frame error rate plot.

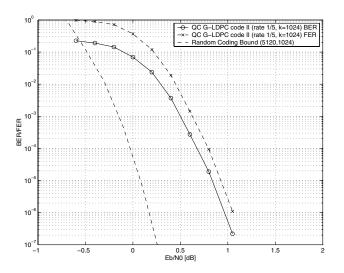


Fig. 7. Frame and bit error rates of rate 1/5 (5120,1024) G-LDPC code II (obtained by puncturing of the rate 1/6 code), compared to the random coding bound.  $I_{max}$  was set to 200.

The BCJR algorithm was employed at each Hamming node. The second decoder was a belief-propagation decoder applied to the code's Tanner graph based on the **H** matrix. The standard LDPC code sum-product algorithm (SPA) was used. The reason that this second decoder is possible is because its **H**-based Tanner graph does not have length-4 loops. This is due to the fact that the parity-check matrix of the shortened (6,3) Hamming code, given by

$$\mathbf{H} = \left[ \begin{array}{ccccc} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{array} \right],$$

and the bipartite graph corresponding to the adjacency matrix are both 4-cycle free. Both simulations used  $I_{max}=200$ . The SPA decoder suffers a loss in the waterfall region of only about  $0.1~\mathrm{dB}$ .

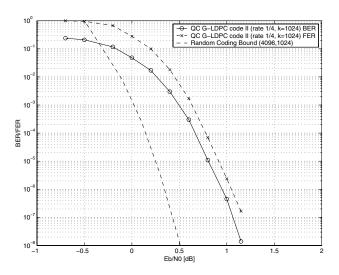


Fig. 8. Frame and bit error rates of rate 1/4 (4096,1024) G-LDPC code II (obtained by puncturing of the rate 1/6 code), compared to the random coding bound.  $I_{max}$  was set to 200.

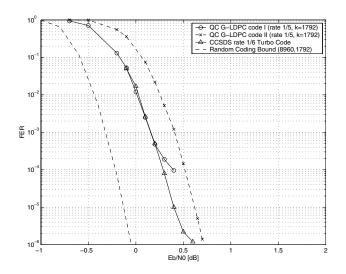


Fig. 9. Frame error rate of rate 1/5 codes obtained by puncturing (10752,1792) G-LDPC codes I and II, compared to the CCSDS turbo code with rate 1/6 (whose performance was taken from [30]) and to the random coding bound.  $I_{max}$  was set to 200 for G-LDPC codes.

4) On Encoders and Minimum Distance for  $C^I$  and  $C^{II}$  Code Families: The two protographs of the  $C^I$  and  $C^{II}$  code families in Fig. 5 suggest an encoding procedure independent of whether or not the codes are quasi-cyclic. The key observation is that the sub-graph involving nodes  $s_0$ ,  $p_0$  and  $p_1$  in Fig. 5 represent the protograph of an IRA code. Thus, an encoder can be configured as a parallel concatenation of an IRA encoder and a shortened (6,3) Hamming code encoder, so that one need not exploit the quasi-cycle properties of the code to devise an encoder. (IRA code encoders are discussed in [13], [14], [26].)

Moreover, because of the observation that the two protographs correspond to the parallel concatenation between an IRA code and a Hamming code, we can explain why the  $\mathcal{C}^{II}$  code floor is lower than the  $\mathcal{C}^{I}$  code floor (see Fig. 9). In the  $\mathcal{C}^{I}$  code family protograph, the repetition rate in the

IRA subgraph is 3 (corresponding to 3 edges connecting the systematic bit  $s_0$  in the protograph and the accumulator). For the  $\mathcal{C}^{II}$  code family, the repetition rate is 4 (corresponding to 4 edges connecting the systematic bit  $s_0$  in the protograph and the accumulator). However, as shown in [13], [14], IRA codes with a repetition rate of 4 tend to have a much larger minimum distance than those with a repetition rate of 3. Additionally, because half of the parity bits for both code families are sent to the Hamming encoder, codeword distances will be increased beyond the distances provided by the IRA component codes.

Computing the true minimum distance of any graph-based code is a formidable task. One attempt to toward this problem was presented in [31], [32] and another was given in [33]. We applied the algorithm proposed in [33] to the parity-check matrix of the rate-1/6  $\mathcal{C}^{II}$  code (k=1024), which we have already noted is 4-cycle free. The estimated minimum distance is 203.

#### IV. CONCLUSION

In this paper a technique for designing quasi-cyclic G-LDPC codes is presented. The quasi-cyclic design permits the construction of G-LDPC codes amenable to efficient encoder and decoders implementations. The G-LDPC codes constructed possess remarkable performance in both the error floor and waterfall regions. A pragmatic approach for designing good codes was proposed which is based on the insertion of powerful constraint nodes in an LDPCC bipartite graphs. Such a doping technique is most easily performed on the protograph of an LDPCC. The performance of low-rate codes constructed in this manner were presented. The density evolution analysis proposed in this paper permits the analysis of the iterative decoding properties of a G-LDPC code in the earlier steps of the code's design, constituting a powerful tool for code designers. The results in this paper motivate further investigation into the design of quasi-cyclic G-LDPC codes (via protographs) for a wider range of code rates and channels. For example, the use of recursive systematic convolutional constraints is considered in [34]. Alternatively, owing to their attractive trellis representations [35], Reed-Muller codes should also be considered for use in the constraint nodes of G-LDPC codes.

#### ACKNOWLEDGMENTS

The authors would like to thanks the Associate Editor and the reviewers, especially Reviewer 1, whose comments vastly improved this paper.

# APPENDIX EXIT CHART ANALYSIS OF G-LDPC CODES

A preliminary step in the development of an EXIT (extrinsic information transfer) chart [2], [36], [37] analysis for G-LDPC codes, based on the Gaussian approximation [38], requires the verification of the output probability density function (p.d.f). for constraint nodes, given a symmetric-Gaussian input. Although the figure count limit preclude us from displaying it, we have produced the p.d.f. estimates for the output of a constraint node based on a (15,11) Hamming code. The histograms have been obtained using a BCJR [27]

decoder for the Hamming code, with an input vector of 15 independent values, all of them Gaussian with mean  $m_{IN}$  and variance  $\sigma_{IN}^2 = 2m_{IN}$ , corresponding to input signal-to-noise ratio  ${\sf SNR}_{IN} = m_{IN}/2$ . Note that the all-zero codeword was assumed. The procedure was repeated  $100\,000$  times for each value of the input SNR. We observed that, at 4 dB and 6 dB, the output p.d.f.'s are nearly Gaussian, but the symmetry condition is generally not satisfied. However, at  ${\sf SNR}_{IN} = 2$  dB the output probability density function is far from Gaussian. For this reason the Gaussian approximation should be used carefully when analyzing codes based on this Hamming code at low SNR values.

This evaluation procedure must be applied to all component codes in order to verify the suitability of the Gaussian approximation for a particular nodes distribution. Regarding the unsatisfied symmetry condition, we can continue to force a Gaussian-symmetric model for output distributions. However, the choice of the tracked parameter can lead to different results in the threshold calculation since the Gaussian symmetric distribution is fit according to measured values. We found that tracking the mean value generally leads to threshold values that were optimistic. Using as the tracked parameter the SNR of the distributions, or the mutual information (MI) [36] between the message and the associated codeword's symbol, we found more reasonable results.

For codes described by a graph with different node types, we averaged all tracked parameters (related to the different messages distributions) over the fraction of edges connected to different node types, as suggested in [37]. Choosing the MI between decoder messages and the corresponding transmitted bits as the tracked parameter, we define  $I_A$  as the *a priori* MI (i.e., the MI between a soft-input of a node and the corresponding codeword bit), while  $I_E$  is used to denote the *extrinsic* mutual information (i.e. the mutual information between a soft-output of a node and the corresponding codeword bit) [36]. In the case of variable nodes, the notation will be particularized to  $I_A^{(v_j)}$  and  $I_E^{(v_j)}$  for the j-th variable node. Similarly,  $I_A^{(c_i)}$  and  $I_E^{(c_i)}$  will be used for constraint nodes.

The soft-input of a variable node is composed of messages both from its neighboring constraint nodes and from the channel. Thus, for a variable node we have

$$I_E^{(v_j)} = I_E^{(v_j)} \left( I_A^{(v_j)}, E_b/N_0 \right),$$
 (3)

where  $I_A^{(v_j)}$  is approximated by the average of all the extrinsic MI values at the output of the constraint nodes over the fraction of edges connected to various constraint node types:

$$I_A^{(v_j)} = I_{Ec} \triangleq \frac{\sum_{i=0}^{m_c - 1} I_E^{(c_i)} \cdot d_c^i}{\sum_{i=0}^{m_c - 1} d_c^i}.$$
 (4)

Constraint nodes accept input messages only from variable nodes, so that

$$I_E^{(c_i)} = I_E^{(c_i)} \left( I_A^{(c_i)} \right),$$
 (5)

with  $I_A^{(c_i)}$  given by

$$I_A^{(c_i)} = I_{Ev} \triangleq \frac{\sum_{j=0}^{n-1} I_E^{(v_i)} \cdot d_v^j}{\sum_{i=0}^{n-1} d_v^j}.$$
 (6)

 $I_{Ev}$  and  $I_{Ec}$  represent the averages over the fractions of edges of the extrinsic mutual information values connected to variable and constraint nodes, respectively.

An EXIT chart [36] can be obtained by plotting on the same chart the EXIT functions

$$I_{Ev} = I_{Ev}(I_{Ec}, E_b/N_0)$$
 (7

and the inverse of

$$I_{Ec} = I_{Ec}(I_{Ev}).$$
 (8)

The EXIT function for variable nodes and SPC-based constraint nodes can be estimated as follows. From [37], (3) can be expressed as

$$I_{Ev_j}(I_{Ec}, E_b/N_0) = J\left(\sqrt{(d_v^j - 1)[J^{-1}(I_{Ec})]^2 + 8R_c \cdot \frac{E_b}{N_0}}\right), (9)$$

where  $R_c$  is the rate of the code described by the bipartite graph and  $J(\sigma)$  [36] is given by

$$J(\sigma) = 1 - \int_{-\infty}^{\infty} \frac{e^{-(s-\sigma^2/2)^2/2\sigma^2}}{\sqrt{2\pi\sigma^2}} \cdot \log_2(1 + e^{-s}) ds. \quad (10)$$

Approximated closed forms for  $J(\sigma)$  and  $J^{-1}(I)$  can be found in [37]. For SPC-based constraint nodes, and approximation of (5) is given as

$$I_{Ec_i}(I_{Ev}) \approx 1 - J\left(\sqrt{d_c^i - 1} \cdot J^{-1}(1 - I_{Ev})\right).$$
 (11)

Constraint nodes based on more complex component codes require a numerical evaluation of (5).

The EXIT function for constraint nodes more complex than SPC may be computed as follows. Assuming that the soft-input is characterized by a signal-to-noise ratio  $SNR_{IN}$ , the *a priori* mutual information is given by [36]:

$$I_{Ac_i} = J\left(\sqrt{4 \cdot \mathsf{SNR}_{IN}}\right).$$

Concerning the soft-output of the constraint node, the probability density function is approximated by a Gaussian symmetric p.d.f. This distribution can be fit to the measured distribution according to different parameters. Namely, they can be constrained to have the same mean, or the same variance, or the same SNR. The choice of the fitting parameter leads to different threshold values, as mentioned above. If we choose the output signal-to-noise ratio  $\mathsf{SNR}_{OUT}$  as the fitting parameter, the computation of the extrinsic mutual information proceeds as follows:

- estimate  $SNR_{OUT}$  by the collection of soft-output samples, for a given  $SNR_{IN}$ ;
- approximate the soft-output distribution as Gaussian p.d.f. with mean  $m_{OUT} = 2 \cdot \mathsf{SNR}_{OUT}$  and variance  $\sigma_{OUT}^2 = 2 \cdot m_{OUT}$ ;
- · calculate the extrinsic mutual information as

$$I_{Ec_i} = J\left(\sqrt{4 \cdot \mathsf{SNR}_{OUT}}\right).$$

*Example:* In Fig. 10 the EXIT chart for a regular G-LDPC code is depicted. The EXIT chart deals with the regular quasicyclic (2190,1022) G-LDPC code introduced in Section III-A.

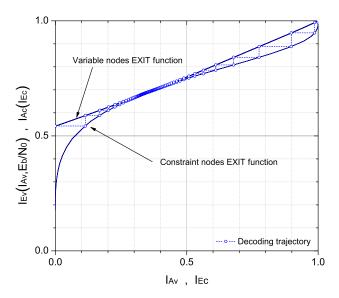


Fig. 10. Mutual information EXIT chart for the a G-LDPC code with constant variable node degree equal to 2 and with constraint nodes based on a (15,11) Hamming code.

Each variable node has degree two and hence identical EXIT characteristics. Also, it can be shown that the two constraint node types have identical characteristics since they are based on different permutations of a (15,11) Hamming code. Thus, for both variable and constraint nodes, the average characteristic is equal to that of a given node. For the Hamming constraint nodes the fitting parameter was chosen to be the SNR. The resulting threshold was  $(E_b/N_0)^* = 1.04$  dB. If the constraint nodes' soft-output distributions were fit using the measured mean value, the threshold would be lower, namely  $(E_b/N_0)^* = 0.75$  dB. This issue, i.e., the higher decoding threshold obtained by measuring the real SNR, was observed in [39]. In [39], the decoding threshold obtained by fitting the Gaussian symmetric distribution of the soft-outputs on the measured mean value was seen to be more precise.

#### REFERENCES

- R. G. Gallager, Low-Density Parity-Check Codes. Cambridge, MA: M.I.T. Press, 1963.
- [2] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.
- [3] Y. Mao and A. Banihashemi, "A heuristic search for good low-density parity-check codes at short block lengths," in *Proc. IEEE Intl. Conf. Commun. (ICC)*, 2001, p. 4144.
- [4] T. Tian, C. Jones, J. D. Villasenor, and R. D. Wesel, "Selective avoidance of cycles in irregular LDPC code construction," *IEEE Trans. Commun.*, vol. 52, pp. 1242–1247, Aug. 2004.
- [5] S. Johnson and S. Weller, "Constraining LDPC degree distributions for improved error floor performance," *IEEE Commun. Lett.*, vol. 10, pp. 103–105, Feb. 2006.
- [6] R. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inform. Theory*, vol. 27, pp. 533–547, Sep. 1981.
- [7] W. W. Peterson and E. J. Weldon Jr., Error-Correcting Codes. Cambridge, MA: M.I.T. Press, 1972.
- [8] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," in *Proc. IEEE Intl. Conf. Commun. (ICC)*, May 1993, pp. 1064–1070.
- [9] 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.
- [10] Y. Wang and M. Fossorier, "Doubly generalized LDPC codes," in Proc. IEEE International Symposium Inform. Theory, July 2006, pp. 669–673.

- [11] L. Ping, W. Leung, and K. Wu, "Low-rate turbo-Hadamard codes," IEEE Trans. Inform. Theory, vol. 49, pp. 3213-3224, Dec. 2003.
- [12] G. Liva and W. E. Ryan, "Short low-error-floor Tanner codes with Hamming nodes," in Proc. IEEE Milcom, Oct. 2005, pp. 208-213.
- [13] M. Yang, Y. Li, and W. Ryan, "Design of efficiently encodable moderatelength high-rate irregular LDPC codes," IEEE Trans. Commun., vol. 52, pp. 564-571, Apr. 2004.
- [14] Y. Zhang and W. E. Ryan, "Structured IRA codes: Performance analysis and construction," to be published.
- [15] A. Abbasfar, K. Yao, and D. Disvalar, "Accumulate repeat accumulate codes," in Proc. IEEE Globecom, Nov. 2004, pp. 509-513.
- [16] R. Pyndiah, "Near optimum decoding of product codes: Block turbo codes," IEEE Trans. Commun., vol. 46, pp. 1003-1010, Aug. 1998.
- [17] J. Boutros, O. Pothier, and G. Zemor, "Generalized low density (Tanner) codes," in Proc. IEEE Intl. Conf. Commun. (ICC), June 1999, pp. 441-445.
- [18] S. Dolinar, "Design and iterative decoding of networks of small codes," in Proc. IEEE International Symp. Inform. Theory, June 2003, p. 346.
- [19] N. Miladinovic and M. Fossorier, "Generalized LDPC codes with Reed-Solomon and BCH codes as component codes for binary channels," in Proc. IEEE Globecom, Nov. 2005, pp. 1239-1244.
- [20] J. Chen and R. Tanner, "A hybrid coding scheme for the Gilbert-Elliot channel," in Proc. 42th Allerton Conf. Commun., Control, Computing, Sep. 2004, pp. 1787-1796.
- [21] J. Thorpe, "Low-density parity-check (LDPC) codes constructed from
- protographs," JPL INP, Tech. Rep., pp. 42–154, Aug. 2003.
  [22] R. Tanner, "On quasi-cyclic repeat-accumulate codes," in *Proc. 37th* Allerton Conf. Communication, Control, Computing, Sep. 1999, pp. 249-259
- [23] J. Xu, L. Chen, L. Zeng, L. Lan, and S. Lin, "Construction of lowdensity parity-check codes by superposition," IEEE Trans. Commun., vol. 53, pp. 243-251, Feb. 2005.
- [24] T. J. Richardson and R. L. Urbanke, "Multi-edge type LDPC codes," IEEE Trans. Inform. Theory [Online]. Available: http://lthcwww.
- [25] R. G. Gallager, Information Theory and Reliable Communication. New York: Wiley, 1968.
- [26] H. Jin, A. Khandekar, and R. McEliece, "Irregular repeat-accumulate codes," in Proc. International Symp. Turbo Codes Related Topics, Sep. 2000, pp. 1-8.
- [27] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," IEEE Trans. Inform. Theory, vol. 20, pp. 284–287, Mar. 1974.
  [28] R. J. McEliece, "On the BCJR trellis for linear block codes," IEEE
- Trans. Inform. Theory, vol. 42, pp. 1072-1092, July 1996.
- [29] TM Synchronization and Channel Coding. Blue Book, Issue 1, Consulative Committee for Space Data Systems (CCSDS) Recommendation for Space Data System Standard 131.0.B.1, Sept. 2003 [Online]. Available: http://www.ccsds.org/CCSDS/documents/131x0b1.pdf
- [30] Telemetry Channel Coding. Green Book, Draft Rev. 5, Consulative Committee for Space Data Systems (CCSDS) Recommendation for Space Data System Standard 1xx.0-G-1, Sep. 2002.
- [31] M. Chiani and A. Ventura, "Some implementation issues and error floor analysis for high-rate low-density parity-check codes," in Proc. IEEE Softcom, Oct. 2001, pp. 49-56.
- -, "Design and performance evaluation of some high-rate irregular low-density parity-check codes," in Proc. IEEE Globecom, Nov. 2001, vol. 12, pp. 990-994.
- [33] X.-Y. Hu, M. P. C. Fossorier, and E. Eleftheriou, "On the computation of the minimum distance of low-density parity-check codes," in Proc. IEEE Intl. Conf. Commun. (ICC), June 2004, pp. 767-771.
- [34] S. Abu-Surra, W. E. Ryan, and G. Liva, "On generalized LDPC codes and their decoders," in IEEE Commun. Theory Workshop, May 2007.
- S. Lin and D. J. Costello, Error Control Coding, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 2004.
- [36] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," IEEE Trans. Commun., vol. 49, pp. 1727-1737, Oct. 2001.
- [37] 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.
- [38] S.-Y. Chung, T. J. Richardson, and R. L. Urbanke, "Analysis of sumproduct decoding of low-density parity-check codes using a Gaussian approximation," IEEE Trans. Inform. Theory, vol. 47, no. 2, pp. 657-670, Feb. 2001.
- [39] D. Divsalar, S. Dolinar, and F. Pollara, "Iterative turbo decoder analysis based on density evolution," IEEE J. Select. Areas Commun., vol. 19, pp. 891-907, May 2001.



Gianluigi Liva was born in Spilimbergo, Italy, on July 23rd, 1977. He received the Laurea in Electronic Engineering in 2002 and the Ph.D. degree in Electronic, Computer Science and Telecommunications Engineering in 2006, both from the University of Bologna. His main research interests include satellite communication systems and error control coding with emphasis on LDPC codes and Turbolike codes. Since 2003 he has been involved in the research of near Shannon limit channel codes for high data rate CCSDS (Consultative Committee

for Space Data Systems) missions, in collaboration with the European Space Operations Centre of the European Space Agency (ESA-ESOC). From October 2004 to April 2005 he was researching at the University of Arizona as visiting student, where he was involved in the design of low-complexity coding systems for space communication systems. He is currently with the Institute of Communications and Navigation, at the German Aerospace Center (DLR), in Wessling, where he is involved in the research of error control and modulation techniques for both near-earth and deep space systems. He is a member of IEEE and he serves as reviewer for several IEEE Transactions, Journals and Conferences.



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 90 publications in the leading conferences and journals in the area of communication theory and channel coding. He is also preparing the textbook Channel Codes: Classical and Modern (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. He was a General Co-Chair of the 2007 Information Theory Workshop in Lake Tahoe.



Marco Chiani was born in Rimini, Italy, in April 1964. He received the Dr. Ing. degree (magna cum laude) in Electronic Engineering and the Ph.D. degree in Electronic and Computer Science from the University of Bologna in 1989 and 1993, respectively. Dr. Chiani is a Full Professor at the II Engineering Faculty, University of Bologna, Italy, where he is the Chair in Telecommunication. During the summer of 2001 he was a Visiting Scientist at AT&T Research Laboratories in Middletown, NJ. He is a frequent visitor at the Massachusetts

Institute of Technology (MIT), where he presently holds a Research Affiliate appointment.

Dr. Chiani's research interests include wireless communication systems, MIMO systems, wireless multimedia, low density parity check codes (LD-PCC), and UWB. He is leading the research unit of CNIT/University of Bologna on Joint Source and Channel Coding for wireless video and is a consultant to the European Space Agency (ESA-ESOC) for the design and evaluation of error correcting codes based on LDPCC for space CCSDS applications.

Dr. Chiani has chaired, organized sessions, and served on the Technical Program Committees at several IEEE International Conferences. He was Co-Chair of the Wireless Communications Symposium at ICC 2004. In January 2006 he received the ICNEWS award "For Fundamental Contributions to the Theory and Practice of Wireless Communications." He is the past chair (2002-2004) of the Radio Communications Committee of the IEEE Communication Society and the current Editor of Wireless Communication for the IEEE TRANSACTIONS ON COMMUNICATIONS.