

# Serial Concatenation of LDPC Codes and Differential Modulations

Michele Franceschini, *Student Member, IEEE*, Gianluigi Ferrari, *Member, IEEE*, Riccardo Raheli, *Member, IEEE*, and Aldo Curtoni

**Abstract**—In this paper, we consider serially concatenated schemes with outer novel and efficient *low-density parity-check* (LDPC) codes and inner modulations effective against channel impairments. With a pragmatic approach, we show how to design LDPC codes tailored for simple and robust modulation formats, such as *differentially encoded* (DE) modulations. The LDPC codes are optimized through the use of a recently proposed analysis technique based on *extrinsic information transfer* (EXIT) charts. In particular, we optimize, through a “clever” *random walk* in the parametric space, the *degree distributions* of the outer LDPC codes, obtaining significant insights on the impact of such distributions on the performance of the proposed concatenated schemes. The optimization is carried out for transmission over both the additive white Gaussian noise channel and a noncoherent channel. The performance predicted by the EXIT chart-based optimization is confirmed by results obtained via computer simulations, considering phase-shift keying and quadrature amplitude modulation at the transmitter side, and iterative demodulation/decoding at the receiver side. The significance of the proposed optimized design of LDPC-coded schemes with DE modulations is validated by the fact that standard nonoptimized LDPC codes perform poorly when used together with inner DE modulations.

**Index Terms**—Differential modulations, extrinsic information transfer (EXIT) charts, iterative decoding, low-density parity-check (LDPC) codes, LDPC code design, LDPC-coded modulations, serial concatenations.

## I. INTRODUCTION

**I**N THE LAST DECADE, techniques for communicating over noisy channels have been greatly improved by the upcoming of new channel coding techniques: turbo codes [1] and low-density parity-check (LDPC) codes<sup>1</sup> [2]–[4]. These codes are well suited for communication over the additive white Gaussian noise (AWGN) channel, allowing near-capacity error-free transmission. Unlike turbo codes, LDPC codes offer a practical way of optimization over a *continuous* space of design parameters, given by the *degree distributions*, corresponding to polynomials with positive real coefficients. In [5], it is shown

how to partition the set of all LDPC codes in equivalence classes characterized by the same degree distributions.

In the literature, the design of schemes given by the concatenation of an encoder (either convolutional or block) with a high-order modulator has received substantial attention. In bit-interleaved coded modulation (BICM) schemes [6], [7], consisting of the concatenation of a binary encoder, a bit interleaver and a high-order memoryless mapper are proposed and analyzed. At the decoder side, a soft demapper generates reliability values for the bits embedded in each modulated symbol, and these values feed a decoder corresponding to the binary encoder used at the transmitter side. In [8]–[11], an extension of BICM schemes, denoted as BICM with iterative decoding (BICM-ID), is proposed: iterative information exchange between the soft demapper and the decoder is considered and performance advantages are observed. In [12], it is shown how to analyze, using extrinsic information transfer (EXIT) charts [13], the performance of LDPC codes transmitted over a multi-input–multi-output (MIMO) channel through a MIMO modulator, with a corresponding soft-input–soft-output (SISO) module<sup>2</sup> at the receiver side. This can be interpreted as a special instance of a BICM-ID scheme, where the interleaver is not required because of the random nature of the LDPC code. In [12], a heuristic optimization technique for the degree distributions of the LDPC code is also proposed—as a matter of fact, the technique considered in [12] was originally introduced in [16] as a way of designing LDPC codes suited to memoryless channels and approaching the capacity limit. Efficient differential modulation schemes suitable to iterative detection/decoding are proposed in [17] and [18], where differentially encoded (DE) phase-shift keying (PSK) is considered.

In this paper, we show how to design good LDPC codes for DE modulations. We analyze the optimized codes, gaining insights into their graph structure and highlighting the differences between LDPC codes for DE modulations and standard LDPC codes. We consider the concatenation of an LDPC code with a differential modulator for both PSK and quadrature amplitude modulation (QAM). At the receiver side, we follow the approach proposed in [12], which enables an accurate EXIT chart-based system performance evaluation. The main features of the proposed scheme, as compared with a standard BICM-ID scheme are: 1) the presence of a modulator *with memory* and 2) the particular subblock decomposition of the receiver. We propose a novel optimization technique to design LDPC codes

Manuscript received April 15, 2004; revised December 17, 2004 and April 30, 2005. This paper was presented in part at the Communication Theory Workshop (CTW 2004), Capri Island, Italy, and in part at the International Symposium on Information Theory and Application (ISITA 2004), Parma, Italy.

M. Franceschini, G. Ferrari, and R. Raheli are with the Dipartimento di Ingegneria dell'Informazione, University of Parma, 43100 Parma, Italy (e-mail: mfrance@tlc.unipr.it; gianluigi.ferrari@unipr.it; raheli@unipr.it).

A. Curtoni is with Selta S.p.A., 29010 Cadeo, Piacenza, Italy (e-mail: baso\_79@libero.it).

Digital Object Identifier 10.1109/JSAC.2005.853797

<sup>1</sup>To be precise, LDPC codes were invented in the 1960s by Gallager [2], and rediscovered in the 1990s [3].

<sup>2</sup>In this paper, we generally define a SISO block as a detection block which *computes the reliabilities of symbols* at the input/output of a time-invariant finite-state machine (FSM) [14], [15].

suitable to DE modulations, based on the use of *EXIT charts* and consisting of a “clever” *random walk* across the parametric space, i.e., the LDPC code degree distributions. In particular, this technique can take into account channel impairments, which are usually neglected for the sake of feasibility by other analysis methods. We compare the performance of codes optimized for DE modulations with the performance of standard LDPC codes, i.e., optimized for transmission over a *memoryless* channel. We show that LDPC codes optimized for DE modulations significantly outperform standard LDPC codes. *Vice versa*, the obtained optimized codes are shown to be tailored *specifically* for the particular DE modulation format and the proposed receiver scheme: in other words, while they perform well if used jointly with DE, they perform poorly with other modulation schemes. This will be shown to depend on the presence of a large fraction of degree-2 variable nodes.

The outline of this paper is as follows. In Section II, the necessary background is provided. In Section III, the considered transmission system is presented and discussed. In Section IV, we introduce the EXIT chart-based analysis of the convergence behavior of the decoding process, we motivate the need for ad hoc optimization of LDPC codes, we propose a simple, yet effective, optimization technique, and we show how to use it to design good LDPC codes to be used with a generic, possibly coded, modulation format. In Section V, we investigate the behavior of serially concatenated schemes with outer LDPC codes and inner DE modulations through numerical simulations. Section VI concludes the paper.

## II. BACKGROUND

In this section, we shortly discuss a few concepts needed in the next sections. First, we briefly introduce LDPC codes and a possible classification method. Then, we describe a recent approximate analysis technique for iterative decoders, based on the use of EXIT charts, which leads to great simplifications with respect to other, more accurate, analysis methods, such as those based on density evolution [4] or simulations of the complete system.

### A. LDPC Codes and Degree Distributions

In [2], LDPC codes were first introduced as linear block codes with a *sparse* parity-check matrix  $H$ . In [2], it is also shown how to build a graph, in a one-to-one correspondence with  $H$ , consisting of two kinds of nodes (also known as bipartite graph): each node of the first kind, denoted as *check node*, is associated to a row of  $H$ ; each node of the second kind, denoted as *variable node*, is associated to a column of  $H$ .

A suboptimal decoding algorithm for LDPC codes has been proposed in [2], characterized by the fact that the nodes act as processors exchanging real-valued messages on the code’s graph [19]. This algorithm achieves optimality if the code graph has no *cycles* [2]. Following the notation in [5], a node has degree  $d$  if it has  $d$  branches departing from it. The decoding algorithm can be formulated in the logarithmic likelihood (LL) domain to significantly simplify the computation of the messages at the variable nodes [2], [4], [19], [20].

The *degree distributions* of an LDPC code are polynomials denoted as  $\lambda(x)$  and  $\rho(x)$ , whose coefficients  $\{\lambda_i\}$  and  $\{\rho_j\}$  correspond to the fraction of branches in the graph connected to degree- $i$  *variable* nodes and degree- $j$  *check* nodes, respectively, [5]. The polynomial  $\rho(x)$  is defined as the *check node degree distribution* and  $\lambda(x)$  is defined as the *variable node degree distribution*. The coefficients  $\{\rho_j\}$  and  $\{\lambda_i\}$  must satisfy the following constraints [5]:

$$\begin{aligned} 0 &\leq \rho_j \leq 1 & j &\geq 1 \\ 0 &\leq \lambda_i \leq 1 & i &\geq 1 \\ \sum_{j=1}^{\infty} \rho_j &= 1 \\ \sum_{i=1}^{\infty} \lambda_i &= 1. \end{aligned} \quad (1)$$

Moreover, the following linear constraint must be satisfied for a degree distribution in order to be compatible with a given code rate  $R$  [5]:

$$\sum_{j=1}^{\infty} \frac{\rho_j}{j} = (1 - R) \sum_{i=1}^{\infty} \frac{\lambda_i}{i}. \quad (2)$$

### B. SISO Detectors and EXIT Charts

In this paper, we will use SISO detectors [14], [15], [17], [18], [21] for DE-PSK and DE-QAM transmitted over an AWGN channel. Noncoherent detection of DE-PSK will also be considered. In the former two cases, the channel is memoryless and the SISO detector is based on the forward-backward (FB) algorithm [21]; in the latter case, the channel has infinite memory and the SISO detector is based on an approximated and very effective *a posteriori* probability (APP) algorithm based on the FB algorithm and described in detail in Section V-D. Due to the suboptimal nature of the iterative decoding schemes, we will prefer the term “reliability” to the term “probability” when referring to the quantities at the input and at the output of SISO blocks, usually referred to as *a priori* and *a posteriori* probabilities.

Without lack of generality, we will assume the reliabilities at the output of SISO blocks as referring to binary symbols. It is possible to compute the mutual information (MI) between each binary symbol and its reliability. Due to the binary nature of the symbols, this MI takes on a value between zero and one.

An EXIT curve for a SISO block  $\mathbf{S}$  is a function  $I_{\mathbf{S}}(I)$ , which quantifies the average relationship between the MI of the *a priori* reliabilities at the input of the block (i.e., the variable  $I$ ) and the MI of the *a posteriori* reliabilities at the output of the block (i.e.,  $I_{\mathbf{S}}$ )—recall that the MI is computed with respect to the transmitted information sequence. EXIT charts allow to predict the system’s performance with a significantly lower computational burden, with respect to the use of standard computer simulations employed to evaluate the bit-error rate (BER) performance of iterative decoders [13], [22].

## III. COMMUNICATION SYSTEM MODEL

The transmitter scheme, shown in Fig. 1, consists of a simple concatenation of an outer LDPC encoder and an inner *coded*

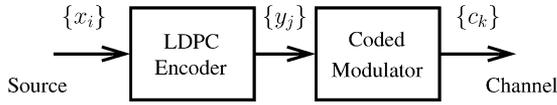


Fig. 1. System model: transmitter side.

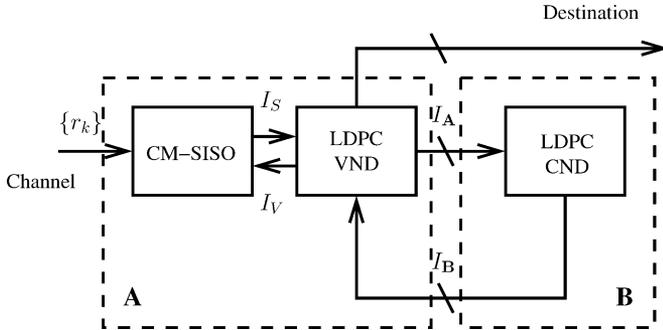


Fig. 2. System model: receiver side.

modulator (CM), which is directly connected to the channel. Without loss of generality, we consider the discrete-time low-pass equivalent model of the communication system. A binary sequence  $\{x_i\}$  at the input of the LDPC encoder is coded into a binary sequence  $\{y_j\}$  (representing a codeword). The binary symbols  $\{y_j\}$  are then coded and mapped to high-order modulated symbols  $\{c_k\}$ . The goal of the inner CM is to make the communication system robust against possible channel impairments. A possible realistic scenario could be a bandpass communication system, where an inner differential encoder is used to solve phase synchronization ambiguities, or to completely avoid phase synchronization problems [23], [24] by means of a differential or noncoherent detection [25], [26]. Another example of inner block could be a modulator which inserts pilot symbols to help synchronization at the receiver side.

The receiver is depicted in Fig. 2. At the input of the receiver, the sequence of channel observations is denoted as  $\{r_k\}$ . For simplicity, we are considering one sample per coded symbol. If two or more samples per symbols are necessary, such as to allow for a time varying channel, the proposed derivation can be extended by considering a suitable vector notation.

The receiver is partitioned into two blocks, denoted as block **A** and block **B**. Block **A** comprises the following subblocks.

- A SISO block matched to the CM and the channel, and referred to as CM-SISO block. This block computes the *a posteriori* reliabilities of the binary symbols  $\{y_j\}$  at the input of the CM on the basis of the channel observations and the relevant *a priori* reliabilities (coming from the block labeled “LDPC VND” and described below).
- An LDPC variable node detector (VND), associated with the variable nodes in the code bipartite graph. This block computes the reliability of each binary symbol  $y_j$  based on the reliabilities from the CM-SISO block and the information received from block **B** and based on the code constraints.

Block **B** includes the LDPC check node detector (CND), associated with the check nodes in the code bipartite graph. The LDPC CND computes the reliability of each binary symbol  $y_j$

based on the *a priori* reliabilities received from the LDPC VND and based on the LDPC code constraints.

The reliabilities at the output of block **A** are computed as follows.

- 1) The VND processes the messages coming from block **B** by performing, at each variable node, a sum of all the incoming messages *excluding* the one coming from the CM-SISO block. The obtained messages are passed to the CM-SISO block as *a priori* input.
- 2) The CM-SISO block computes, based on the observations from the channel and the *a priori* information, reliability values according to its internal algorithm (e.g., FB).
- 3) Finally, the VND computes the messages to be sent to block **B** according to the standard LDPC decoding algorithm, but using, as *a priori* input, the messages from the CM-SISO decoder.

It is important to note that, in all the above computations, only the so-called *extrinsic information* is exchanged between the component blocks [1], [4].

The overall decoding algorithm at the receiver can be described as follows.

- As initialization step, the *a priori* reliabilities of the symbols  $\{y_j\}$  at the input of block **A** (from block **B**) correspond to complete uncertainty (a value equal to 0 in the LL domain).
- Decoding starts from block **A**, which computes output reliabilities and sends them to block **B**. At the first step, since all the messages coming from the CND are 0, the output of block **A** simply consists of the output of the CM-SISO.
- The LDPC CND (i.e., block **B**), thus, computes the extrinsic information to be passed to block **A**.
- The algorithm iterates from the second step until a valid LDPC codeword is obtained or a maximum number of  $N_i$  iteration have been performed.
- In the case a valid LDPC codeword is not obtained, an additional standard LDPC decoding algorithm is applied based on the last extrinsic information at the input of LDPC VND block. This corresponds to iterating information only between LDPC VND and LDPC CND. The maximum number of standard LDPC decoding iterations is  $N_{\text{LDPC}}$ .
- At the end of the process, the *complete* (not extrinsic) reliabilities are computed by the LDPC VND and delivered to the destination.

#### IV. EXIT CHARTS AND OPTIMIZATION ALGORITHM

##### A. EXIT Chart-Based Analysis of the Receiver Performance

For each block shown in Fig. 2, it is possible to draw the corresponding EXIT curve [13], [22]. In Fig. 2, the MI at the output of each block **A** and **B** is denoted as  $I_A$  and  $I_B$ , respectively; within block **A**, the MI at the input and output of the CM-SISO subblock are labeled  $I_V$  and  $I_S$ , respectively. The decoding process can then be represented as a recursive update of the MI in the EXIT charts. If the MI converges to 1, it is possible to predict that the BER will converge to zero.

At this point, we are interested in the computation of the EXIT charts of blocks **A** and **B**. Block **B** is simply characterized by the EXIT curve of the LDPC VND, while the EXIT curve of block **A** is obtained by combining the EXIT curve of the LDPC VND with that of the CM-SISO block. In [12], some formulas are given for the computation of LDPC VND and LDPC VND EXIT curves on the basis of a Gaussian assumption for the exchanged messages, which provides great simplification and good accuracy. Since, in general, the analytical computation of the CM-SISO EXIT curve is a difficult task, approximate computation can be based on Monte Carlo simulations [12].

In the following, approximate formulas are given for the EXIT curves  $I_A$  (of block **A**) and  $I_B$  (of block **B**) [12]:

$$I_B = 1 - \sum_j \rho_j J\left(\sqrt{j-1} J^{-1}(1 - I_A)\right) \quad (3)$$

$$I_A = \sum_i \lambda_i J\left(\sqrt{(i-1)(J^{-1}(I_B))^2 + (J^{-1}(I_S))^2}\right) \quad (4)$$

where the function  $J(\cdot)$  is defined as follows:

$$J(\sigma) \triangleq \int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\frac{\sigma}{2})^2}{2\sigma^2}} \log_2 \frac{2}{1+e^{-x}} dx. \quad (5)$$

The MI  $I_S$  at the output of the CM-SISO block is a function of the MI  $I_V$  of the messages passed by the VND to the CM-SISO block and corresponds to the EXIT function of the CM-SISO. The MI  $I_V$  of the messages passed by the LDPC VND to the CM-SISO block can be approximately computed as follows [12]:

$$I_V = \sum_i \lambda_i J\left(\sqrt{i} \cdot J^{-1}(I_B)\right). \quad (6)$$

### B. Need for Optimization

A reader might wonder if there is a real need for an LDPC code *optimized specifically* for the used inner CM block, or if the use of a powerful LDPC code designed for an AWGN channel might be sufficient to achieve near-channel capacity performance *regardless* of the inner CM block. In general, the LDPC code has to be optimized for the specific application, and an EXIT chart-based analysis can provide additional insights regarding this aspect.

At the beginning of this section, we have chosen to partition the receiver into the two blocks **A** and **B**, including in block **A** both the CM-SISO and the LDPC VND modules. Note that it would have been possible to study the recursive exchange of information of the three distinct component modules: CM-SISO, LDPC VND, and LDPC VND. The CM-SISO and the LDPC VND modules would have one input and one output (not taking into account the channel input for the CM-SISO module, since the corresponding messages are constant throughout the decoding process) and the LDPC VND would have two inputs and one output. Had we analyzed an LDPC code transmitted over a binary input (BI) AWGN channel, there would be no CM-SISO module, but only a memoryless soft demapper, without *a priori* input. Clearly, the EXIT curve of block **A** with the CM-SISO module cannot be equal to the EXIT curve

of block **A** with a soft demapper only. In fact, in the case with the CM-SISO module, every time the CM-SISO module input changes, the reliability values to be sent to the variable nodes are recomputed. On the contrary, these reliability values would be unmodified in the scheme with soft demapper only. It follows that, since the EXIT curves of block **A** (interpreted as functions of the parameters  $\{\lambda_i\}$ ) are different in the two cases and since the EXIT curves of block **B** (interpreted as functions of the parameters  $\{\rho_j\}$ ) are the same, considering an optimization technique based on EXIT charts, in general, an LDPC code optimized for the presence of a CM will be different from an LDPC code optimized for the absence of a CM. Moreover, one can conclude that LDPC codes optimized for the two scenarios should be equal *if and only if* the EXIT curves of the CM-SISO module are constant and independent from the MI at the feedback input. This applies, for example, to a scenario where the CM-SISO module's feedback input is not used, as in the case of BICM, or for every BI memoryless channel, such a binary symmetric channel (BSC), a binary erasure channel (BEC), as well as the BI-AWGN channel.

### C. Optimizing the EXIT Charts

In [12], it is shown that by “eye fitting” the two EXIT curves  $I_A(I)$  and  $I_B^{-1}(I)$  by varying the degree distributions  $(\lambda(x), \rho(x))$ , a significant system performance improvement can be obtained. Since the EXIT curves of VND and VND, relative to the most powerful known LDPC codes for memoryless channels, are very similar at “pinch-off”, i.e., when EXIT curves touch, and considering the good results obtained in [12], at a first glance fitting the EXIT curves seems a good optimization strategy. However, if only low degree nodes are allowed, this similarity of curves becomes less noticeable and usually “low degree only” distributions are desirable in order to keep the LDPC code parity-check matrix as sparse as possible [2], [5]. Moreover, it is important to note that, given a particular signal-to-ratio (SNR) (which will be defined exactly later), convergence of the decoding process can be obtained if the tunnel between the two curves is open. Hence, our actual goal, while performing optimization, *is to keep the tunnel open*.

Our optimization algorithm is based on a simple *random walk* in the degree distribution parametric space. Before describing how this algorithm works, we first provide the reader with some useful considerations and definitions.

Consider, first, two couples of EXIT curves for blocks **A** and **B**, denoted as  $(I_{1,A}(\cdot), I_{1,B}^{-1}(\cdot))$  and  $(I_{2,A}(\cdot), I_{2,B}^{-1}(\cdot))$ , respectively. It can be easily verified that if

$$\begin{aligned} I_{1,A}(I) &\geq I_{2,A}(I) \quad \forall I \in (0, 1) \\ I_{1,B}^{-1}(I) &\leq I_{2,B}^{-1}(I) \quad \forall I \in (0, 1) \end{aligned} \quad (7)$$

i.e.,  $I_{1,A}$  is higher than  $I_{2,A}$  and  $I_{1,B}^{-1}$  is lower than  $I_{2,B}^{-1}$ , then the convergence of the decoding process for the system relative to the EXIT curves  $(I_{1,A}(\cdot), I_{1,B}^{-1}(\cdot))$  will not be slower than the convergence of the system relative to the EXIT curves  $(I_{2,A}(\cdot), I_{2,B}^{-1}(\cdot))$ .

It should be observed that the two EXIT curves touch at point  $(1, 1)$ —a sufficient condition for this is the absence of degree-1 variable nodes in the code, as it can be easily seen

by imposing  $\lambda_1 = 0$  in (3) and letting  $I_{\mathbf{B}} \rightarrow 0$ . The iterative decoding algorithm for a system characterized by the EXIT curves  $(I_{\mathbf{A}}(\cdot), I_{\mathbf{B}}^{-1}(\cdot))$  cannot converge if there exists a value  $I^*$ ,  $0 < I^* < 1$ , such that  $I_{\mathbf{A}}(I^*) < I_{\mathbf{B}}^{-1}(I^*)$ , i.e., the tunnel is closed. We then need to define a functional representative of the tunnel closure: the more the tunnel is closed, the lower this functional must be. A possible choice is the following:

$$f(\lambda, \rho) = \min_{I \in [0,1]} \{I_{\mathbf{A}}(I) - I_{\mathbf{B}}^{-1}(I)\} \quad (8)$$

where we have explicitly indicated the dependence of the functional on the degree distributions. Since, as previously observed, the EXIT curves touch at (1,1), this functional cannot be positive. Moreover, this functional depends also on the particular channel as well as on the CM and the CM-SISO block. As previously observed, it is reasonable to assume that increasing the SNR raises the EXIT curve of block **A**, while decreasing the SNR lowers it. In other words, if the tunnel between the two EXIT curves is at pinch-off, a small SNR increment should be sufficient to open it.

The design parametric space is given by the node degree distributions  $\{\rho_j\}$  and  $\{\lambda_i\}$ . According to (1) and (2), three parameters are linearly dependent on the others. Hence, one has to choose a parameter from the set  $\{\lambda_i\}$ , a parameter from the set  $\{\rho_j\}$ , and an additional parameter from either  $\{\lambda_i\}$  or  $\{\rho_j\}$ . The chosen parameters have then to be expressed as functions of the remaining free parameters. There is no constraint on the numbers of elements of the sets  $\{\lambda_i\}$  and  $\{\rho_j\}$ , provided that these sets are not empty, contain at least four elements and are finite.

We now describe the proposed optimization algorithm. We start with given valid degree distributions associated to a given code rate, according to (2), and determined by a tuple of free parameters. If the tunnel is not closed, i.e.,  $f(\lambda, \rho) = 0$ , we decrease the SNR until the tunnel closes and  $f(\lambda, \rho) < 0$ . New tuples of free parameters are then obtained, by repeatedly adding to the previous tuple a Gaussian increment until all inequalities in (1) are satisfied. The mean of the Gaussian increment is zero and the standard deviation is used to “tune” the optimization algorithm. From the new tuple, we evaluate  $\lambda(x)$  and  $\rho(x)$  and, consequently, the value  $f(\lambda, \rho)$ : if this value is larger than the previous one, we substitute the previous tuple with the new one. If the tunnel opens, the SNR is decreased again, and previous steps are repeated. The algorithm stops when a specific requirement is met, such as, for example, the obtained code ensemble corresponds to an EXIT chart with an open (not closed) tunnel for a desired SNR, or a maximum number of steps (in the random walk) is reached. The steps of the proposed optimization algorithm are summarized in Table I. As a possible improvement for the optimization algorithm, one can diminish the step value, i.e., the standard deviation of the Gaussian increment vector, after a given number of unsuccessful trials. Unlike the EXIT curve fitting optimization algorithm in [12] and [27], the proposed technique offers the advantage of being effective also for small sets of possible node degrees. The proposed algorithm can also be seen as a particular instance of the so called “differential evolution” algorithm [28].

TABLE I  
OPTIMIZATION ALGORITHM: BASIC STEPS

Start	Initialize $\lambda(x)$ and $\rho(x)$ and compute $f(\lambda, \rho)$ .
1	While tunnel is open reduce SNR by small steps and compute the final value of $f(\lambda, \rho)$ .
2	Find a new $(\lambda', \rho')$ compatible with code rate at (small) random distance from $(\lambda, \rho)$ .
3	Compute new $f(\lambda', \rho')$ ; if not larger than previous $f(\lambda, \rho)$ goto step 2, else $(\lambda, \rho) \leftarrow (\lambda', \rho')$ .
4	If stop condition is not reached goto 1 else output $(\lambda, \rho)$ and final SNR.

The proposed algorithm basically performs an optimization of the convergence threshold, defined as the lowest SNR such that the tunnel is open. Within the approximation of the EXIT chart-based analysis, the decoding process converges above this SNR threshold. The simplicity of the proposed optimization algorithm enables a joint optimization of both  $\lambda(x)$  and  $\rho(x)$  in the presence of the CM-SISO block. This would be difficult to perform using analytic optimization techniques. We also observe that, although the predicted thresholds are not very accurate in an absolute sense, they turn out to be in a monotonic relationship with the experimental thresholds obtained by Monte Carlo simulations for actual codes chosen according to optimized degree distributions. This observation, together with the good results shown in the next sections, confirms the validity of the proposed optimization algorithm.

## V. LDPC CODES AND DIFFERENTIAL ENCODING

In this section, we show how the EXIT chart-based analysis presented in Section IV can be used to predict the convergence behavior of the considered serially concatenated coded system for a specific ensemble of LDPC codes. In particular, we present numerical results relative to optimized LDPC codes concatenated with DE-PSK and coherent SISO detection. We then show some results relative to LDPC codes concatenated with QAM with “quadrant” differential encoding and coherent SISO detection [26]. We finally consider LDPC codes serially concatenated with DE-PSK and noncoherent inner SISO detection. For all the considered differential systems, the BER performance will be compared with the performance of BICM systems operating with the same signal constellation and Gray mapping. In the following, we will refer to the proposed iterative systems as *with DE* and to the Gray-mapped PSK and QAM BICM system as *without DE*.

### A. Serial Concatenation of LDPC Codes and PSK With and Without DE

As a representative CM for the transmission system in Fig. 1, we first consider DE-PSK. For coherent detection, the corresponding CM-SISO module implements, with very low complexity, the FB algorithm. The performance of the considered systems, first studied through an EXIT chart-based analysis, is evaluated in terms of BER versus  $E_b/N_0$ , where  $E_b$  is the received energy per bit and  $N_0$  is the one-sided AWGN power

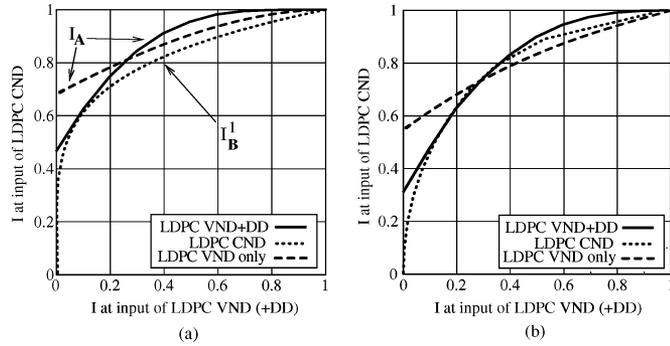


Fig. 3. EXIT chart-based analysis of a system with serial concatenation of an LDPC code and QPSK. (a) EXIT chart of a (3,6) regular LDPC code concatenated with a QPSK with DE ( $E_b/N_0 = 2.5$  dB; tunnel is near *pinch-off*) and QPSK without DE (tunnel is *open*). (b) EXIT chart of an optimized rate 1/2 LDPC code concatenated with a QPSK with DE ( $E_b/N_0 = 0.8$  dB; tunnel is at *pinch-off*) and QPSK without DE (tunnel is *closed*).

spectral density. In all the considered simulations and optimizations, Gray mapping over the PSK constellation is used.

In Fig. 3(a), EXIT charts are shown for a regular rate-1/2 (3,6) LDPC code, characterized by  $\lambda(x) = x^2$  and  $\rho(x) = x^5$ . We preliminarily observe that the use of this code (without DE), mapped to a QPSK modulation format, represents a good tradeoff between complexity and performance for transmission over an AWGN channel. The EXIT curves are computed at  $E_b/N_0 = 2.5$  dB: the solid curve is the EXIT curve of block **A** [LDPC VND and differential detector (DD)] and the dotted curve is the EXIT curve of block **B** (LDPC VND). Note that the SNR does not influence the EXIT curve relative to the LDPC VND (the dotted one in Fig. 3). It is easy to see that the system is at *pinch-off*: convergence at this and lower values of  $E_b/N_0$  is not possible. The dashed curve represents the EXIT curve of the single LDPC VND: this corresponds to the QPSK system *without* DE, i.e., LDPC BICM. It can be immediately seen that at  $E_b/N_0 = 2.5$  dB the tunnel, relative to a transmission scheme without DE is open. The EXIT chart-based analysis then predicts that, for a bit SNR slightly lower than 2.5 dB, the system with DE does not converge as opposed to the system without DE, which instead converges.

We now apply the optimization technique presented in the previous section, forcing the optimization algorithm to use check nodes of degree 3, 4, 8 and 15, and variable nodes of degree 2, 3, and 4 (these are arbitrary and reasonable choices of degrees, but the proposed approach is general). After a few steps, the optimized degree distributions converge to the following:

$$\begin{aligned} \rho_3 &= 0.3157 & \rho_4 &= 0.2259 & \rho_8 &= 0.0273 & \rho_{15} &= 0.4311 \\ \lambda_2 &= 0.5473 & \lambda_3 &= 0.0116 & \lambda_4 &= 0.4411. \end{aligned}$$

Fig. 3(b) shows the EXIT curves for this optimized code ensemble for  $E_b/N_0 = 0.8$  dB: the solid curve corresponds to block **A** and the dotted curve to block **B**. It is immediate to recognize that the tunnel is at *pinch-off*. The dashed curve in Fig. 3(b) is the EXIT curve of the LDPC VND only (i.e., without DD): the tunnel is “heavily” closed, predicting that the system with DE should perform significantly better than the

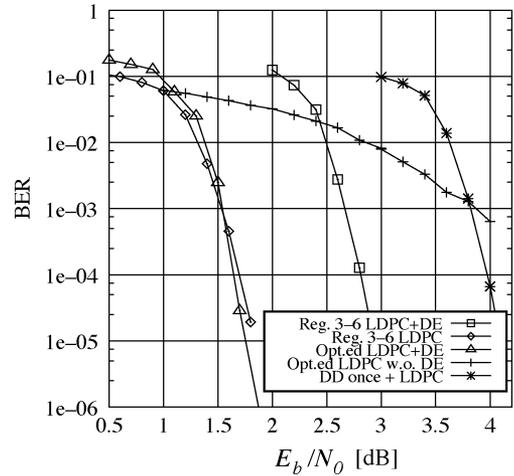


Fig. 4. BER performance of the four communication schemes considered in Fig. 3(a) and (b).

single LDPC code without DE. Note that the convergence SNR threshold predicted by the results in Fig. 3(b) is around 0.9 dB.

In order to closely approximate the degree distributions obtained with the proposed optimization technique, we design LDPC codes with codeword length 6000. In Fig. 4, the performance of both optimized and regular (3,6) LDPC codes with and without DE is shown. For DE systems the maximum number of iterations is  $N_i = 30$  and  $N_{\text{LDPC}} = 30$ , for non-DE systems, a maximum number of 100 standard LDPC iterations is allowed. It can be observed that, for a regular (3,6) LDPC code, while good performance is obtained without DE (curve marked with diamonds), the introduction of DE shifts the BER curve to the right, with an SNR loss of about 1.2 dB (curve marked with squares). When the LDPC code is optimized for DE, i.e., block **A** includes a CM-SISO module based on the FB algorithm relative to the DE modulator, it is possible to see the inversion of performance between the system with and without DE, as predicted by the EXIT chart-based analysis. In other words, the use of the LDPC code optimized for DE, in the system with DE (curve marked with triangles) leads to good performance, i.e., it behaves as the regular (3,6) LDPC code without DE (curve with diamonds). On the other hand, the use of the LDPC code optimized for DE, in the system *without* DE, i.e., LDPC BICM, (curve with crosses), causes a loss, in terms of SNR, of more than 2 dB at a BER equal to  $10^{-3}$ .

It is also possible to use the CM-SISO module, i.e., the DD, only once and then pass the obtained reliability values to a standard LDPC decoder: the corresponding performance, obtained considering a maximum number of 100 LDPC iterations and using the previous regular (3,6) LDPC code, is given by the curve marked with stars. It is easy to recognize that the absence of iteration between the CM-SISO block and the LDPC VND leads to a loss of about 1.2 dB with respect to the system with iterative detection/decoding. This can be interpreted noting that the standard LDPC decoder is based on the assumption that a memoryless channel is used [2]. The presence of DE (and of the corresponding CM-SISO module), instead, heavily correlates the LL passed to the LDPC decoder, thus causing performance degradation.

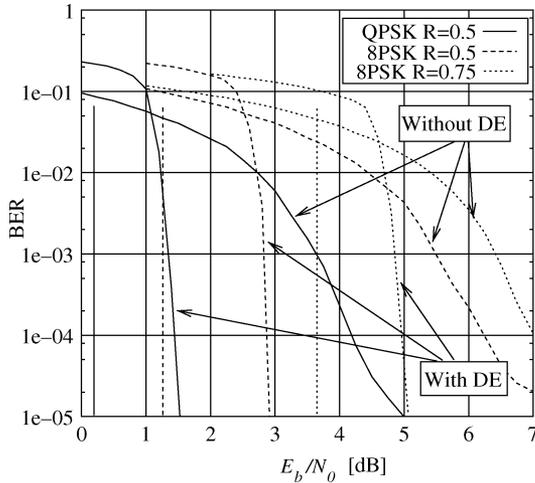


Fig. 5. Simulated BER for three LDPC codes optimized for DE-PSK. Each code is analyzed both with and without DE. In each case, the SNR corresponding to the capacity bound is shown as a vertical line.

### B. Optimized LDPC Codes for PSK Modulation

In order to understand the behavior and limits of the proposed technique, several optimizations have been carried on, both for a system using DE-PSK and a system using PSK without DE. The set of allowed variable node degrees is  $\{2, 3, \dots, 12\}$ . The set of check node degrees is chosen in order to satisfy (2). LDPC codes with codeword length equal to 12 000 have been extracted from the obtained optimized code ensembles. Each LDPC code has been concatenated with both a PSK modulator and a DE-PSK modulator. In both cases, Monte Carlo simulations have been performed. For DE scheme, the decoder's maximum number of iterations is  $N_i = 30$  and  $N_{LDPC} = 30$ , for the scheme *without* DE, a maximum number of 100 standard LDPC iterations is allowed: this makes the complexities of the two different systems very similar. The decoding process stops if a valid codeword is found earlier. In Fig. 5, the BER curves relative to three LDPC codes optimized for the presence of a DE-PSK modulator are shown: the solid curves are relative to an LDPC code with rate  $R = 1/2$  designed for DE-QPSK, the dashed curves are relative to an LDPC code with rate  $R = 1/2$  designed for DE-8PSK, and the dotted curves are relative to an LDPC code with rate  $R = 3/4$  designed for DE-8PSK. For each LDPC code, the BER curve which exhibits a cliff at low SNR corresponds to a system for which the LDPC code has been optimized, i.e., DE-PSK; the other curve represents, instead, the performance of the same LDPC code employed in a PSK LDPC BICM scheme with Gray mapping. For each case, the SNR value corresponding to the capacity limit for the considered coded modulation is shown as a vertical line. The capacity limit for QPSK with code rate  $1/2$ , i.e., with spectral efficiency of 1 bit per channel use, is 0.17 dB; the capacity limit for 8PSK with code rate  $1/2$ , i.e., with spectral efficiency 1.5 bit per channel use, is 1.27 dB; the capacity limit for 8PSK with code rate  $3/4$ , i.e., with spectral efficiency 2.25 bit per channel use, is 3.66 dB. All the DE-PSK systems in Fig. 5 are operating at about  $1 \div 1.5$  dB (in terms of SNR) from capacity. In other words, the optimized codes enable near-capacity performance, even without an exact phase reference.

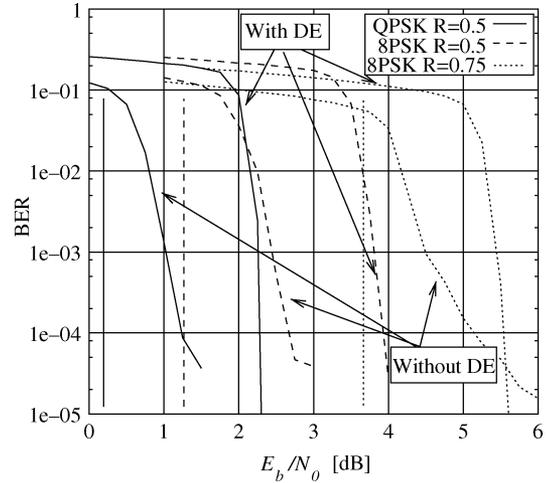


Fig. 6. Simulated BER for three LDPC codes optimized for a memoryless channel. Each code is concatenated with MPSK both with and without DE. In each case, the SNR corresponding to the capacity bound is shown as a vertical line.

In Fig. 6, the performance of LDPC codes optimized for a memoryless PSK modulator is analyzed, both in the presence and absence of DE. For each code, the curve which exhibits a cliff at low SNR corresponds to a system which uses a memoryless PSK modulator, while the other curve represents the performance of the same code concatenated with a DE-PSK modulator. The system without DE shows a performance advantage, in terms of SNR corresponding to the steepest point of the BER curve, of about 1.5 dB with respect to a system with DE. However, it is important to note that LDPC codes optimized for and used with a memoryless PSK modulator exhibit higher “error floor” with respect to that obtained when the same LDPC codes are used with DE-PSK. The presence of the BER floor in the memoryless PSK modulator is due to the nature of the used code, which contains a small amount of *short* cycles. On the other hand, the absence of the floor in the DE-PSK case can be associated to the fact that the DE-PSK modulator can be interpreted as a rate-1 recursive encoder. As shown in [18], the presence of a rate-1 recursive encoder can reduce short error patterns, responsible for the BER curve flattening, by exploiting the so called *interleaving gain*.

In Fig. 7, the coefficients  $\{\rho_j\}$  and  $\{\lambda_i\}$  of several optimized LDPC codes are shown. Different code ensembles with the same constraints are obtained considering different initial seeds of the pseudorandom number generator embedded in the random walk-based optimization algorithm. The code ensembles in Fig. 7(a) and (b) are optimized for DE-QPSK with rate  $1/2$ . The algorithm operates over a limited parametric space, i.e., only a small set of possible node degrees are allowed: the set of variable node degrees is  $\{2, 3, \dots, 12\}$  and the set of check node degrees is  $\{3, 4, \dots, 12\}$ . The variable node degree distributions  $\{\lambda_i\}$  in Fig. 7(c) correspond to realizations of rate- $1/2$  LDPC codes optimized for transmission with BICM PSK. The check node degree distributions appear to give little information, due to the optimization algorithm “residual noise,” this is not surprising since, as stated in [5], the performance of LDPC codes exhibit little dependence on the check node

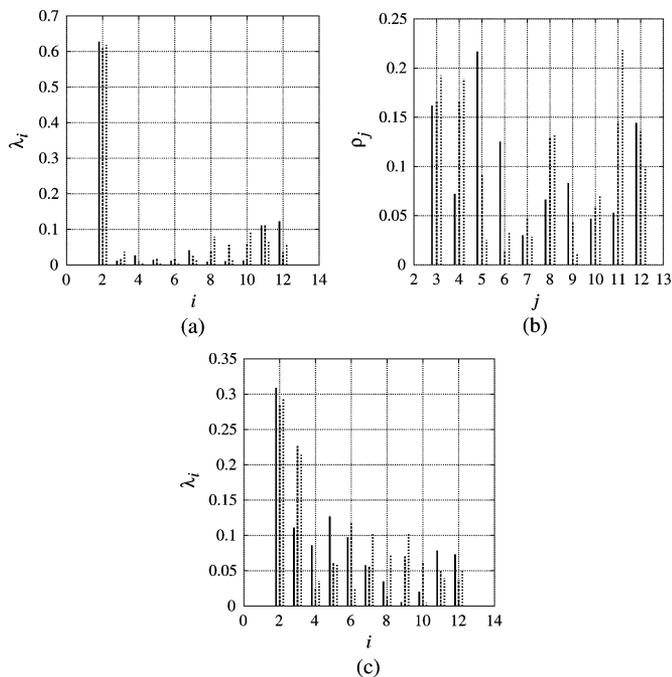


Fig. 7. Bar diagrams of degree distribution coefficients of three realizations of optimized LDPC code ensembles. (a) and (b) Variable and check node degree distributions for three LDPC codes optimized for rate-1/2 DE-QPSK are shown, respectively. (c) Variable node degree distributions of three LDPC codes optimized for rate-1/2 QPSK are shown.

degree distribution. Focusing our attention on the coefficients  $\{\lambda_i\}$ , it is possible to observe that degree-2 variable nodes show a characteristic behavior: in the LDPC code ensembles optimized for DE-QPSK,  $\lambda_2 > 0.5$  and  $\lambda_2 \gg \lambda_i, i > 2$ . Very similar results, in terms of variable node degree distributions with a predominance of  $\lambda_2$ , were obtained also for LDPC codes optimized for rate-1/2 DE-8PSK and rate-3/4 DE-8PSK. In the LDPC code ensembles optimized for a PSK modulator,  $\lambda_2$  is still larger than the other coefficients, but not as large as for differential schemes [see Fig. 7(c)].

In [5], a stability condition on  $\lambda_2$  is given, for standard LDPC decoding algorithm. According to this remark, in order for the BER to approach zero, a necessary condition is  $\lambda_2 < \epsilon$ , where  $\epsilon$  is a parameter which depends on the channel and  $\rho(x)$ . As a reference value, in [5], the authors consider  $\epsilon \simeq 0.4$  for a rate-1/2 standard LDPC code. Our results show that this condition is violated if a CM is inserted between the LDPC code and the channel, allowing in the case of DE-PSK, higher values of  $\lambda_2$ . Note that an LDPC code with large  $\lambda_2$  is a code whose majority of variable nodes have degree 2, and this corresponds to code graphs with a smaller number of edges (for a given code rate and codeword length). Our results (not reported here for lack of space), show a reduction of the order of 20%. Since the computational cost of the decoding algorithm for an LDPC code is proportional to the number of edges in the graph, it follows that LDPC codes optimized for DE-PSK have the pleasant side effect of allowing low-complexity decoding.

It is generally believed that degree-2 variable nodes exhibit weaker error protection than high-order variable nodes [4], [5]. However, considering Fig. 7(a), one notices that the presence of

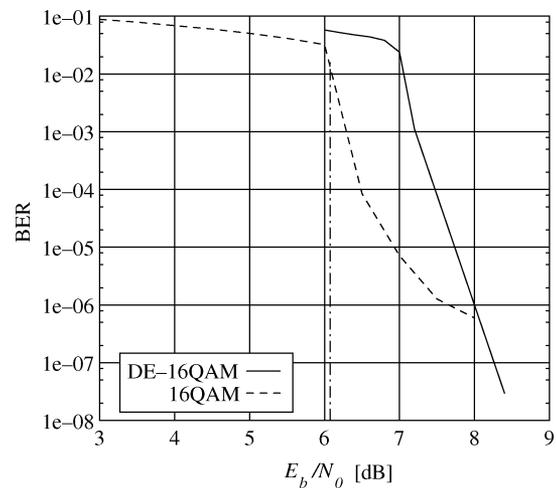


Fig. 8. BER of a rate-7/8 LDPC code optimized for DE-16QAM and concatenated with DE-16QAM (solid line) and with a 16QAM memoryless modulator (dashed line). The vertical (dash-dotted) line indicates the SNR corresponding to the capacity limit for 16-QAM at the considered code rate.

a large percentage of degree-2 variable nodes is associated with an increase of the fractions of high-degree variable nodes. A possible intuitive interpretation for this behavior is that, while a standard LDPC decoder exploits all the available information from the very first iteration, in the proposed iterative detector/decoder the information made available at the “channel input” of the LDPC VND block by the CM-SISO block increases with the iterations. This is possible since, at every iteration, the VND block passes information to the *a priori* input of the CM-SISO block (see Fig. 2). Therefore, the critical part of the decoding algorithm corresponds to the few first iterations, when the information from CM-SISO block is small. High-degree variable nodes seem to help the convergence of the iterative algorithm in the first few iterations.

### C. LDPC Codes for DE-QAM

In Fig. 8, the BER performance for two communication systems with 16QAM, with and without DE, is shown. Both systems use the same rate-7/8 LDPC code with codeword length 65 536. For reference purposes, a vertical dash-dot line is also shown in correspondence to the capacity SNR, equal to approximately 6.16 dB. The LDPC code is chosen from an ensemble of codes optimized for the presence of a DE-16QAM modulator. For DE-QAM, the maximum number of iterations is  $N_i = 30$  and  $N_{\text{LDPC}} = 30$ , for QAM without DE a maximum number of 100 standard LDPC iterations is allowed. The solid curve corresponds to a system with a DE-16QAM and the dotted curve corresponds to the system with a memoryless 16QAM with Gray mapping. In a DE-QAM modulator, two of the 4 bits at its input are encoded by a Gray mapped DE-QPSK modulator: the obtained point is used to rotate, by an angle equal to a multiple of  $\pi/2$ , a first-quadrant 16QAM constellation point selected by the other 2 bits (1 bit per dimension). Fig. 8 shows that the code designed for DE-16QAM performs better if used without DE. A possible interpretation of this result is that the iteration gain of the DE-QAM SISO module is very low. In other words, from an EXIT chart point of view, the considered DE-QAM modulator

is similar to a memoryless, Gray-mapped QAM modulator, and this implies that good codes for DE-QAM may also be good codes for QAM. However, the memory introduced by the DE and the relative CM-SISO block, leads to strong suboptimality of the processing at the LDPC VND and CND, which assume an underlying memoryless channel. Another immediately noticeable fact is that the BER curve relative to the system without DE is characterized by a floor, whereas the curve relative to the system with DE does not show any floor in the considered BER range. The floor in the QAM case can be attributed to the presence of a small amount of short cycles in the code graph, which is typical of *random* LDPC codes. Moreover, the differential encoder can be reinterpreted as a rate-1 recursive encoder (at least for the bits which select the quadrant) which, as observed for DE-PSK, is likely to reduce short errors pattern.

#### D. LDPC Codes for DE-PSK With Noncoherent Detection

Since the proposed design method can take into account the particular channel, as well as the modulation format and the detection algorithm used in the CM-SISO block, the optimization has been carried out also for LDPC codes concatenated with DE-PSK with noncoherent detection. In the presence of phase uncertainty, the received observation can be modeled as

$$r_k = c_k e^{j\theta} + n_k \quad (9)$$

where  $\theta$  is a random variable, constant over the transmitted block and uniformly distributed over  $[0, 2\pi)$ . While coherent detection can be based on the standard FB algorithm in the CM-SISO module, noncoherent MAP symbol detection requires some approximations. Following the approach in [29] and [30], we obtained a detection algorithm based on a quantization of the phase rotation introduced by the channel. First, we compute the APP (through the FB algorithm) conditioned on the channel phase value; then, we average the conditioned APPs over all possible phase values. The *a posteriori* symbol probability can then be written as

$$\begin{aligned} P\{a_k|\mathbf{r}\} &\propto P\{a_k\}p(\mathbf{r}|a_k) \\ &= P\{a_k\} \int_{\theta} p(\mathbf{r}|a_k, \theta)p_{\theta}(\theta)d\theta \end{aligned} \quad (10)$$

where  $\mathbf{r}$  is the vector of all received observations and  $\propto$  means that the first member is equal to the second member times a constant independent of  $a_k$ . In (10),  $p(\mathbf{r}|a_k, \theta)$  can be interpreted as the extrinsic information generated by a coherent FB algorithm, which assumes a phase rotation  $\theta$ . The integral in (10) can be approximated as a sum over a *properly chosen* discrete set  $\mathcal{P}$  of quantized phase values, obtaining

$$P\{a_k|\mathbf{r}\} \simeq P\{a_k\} \sum_{\theta \in \mathcal{P}} p(\mathbf{r}|a_k, \theta)P(\theta). \quad (11)$$

Since DE-PSK is insensitive to rotation of the received signal by multiples of  $2\pi/M$ , where  $M$  is the cardinality of PSK symbols, the set of phases can be a subset of

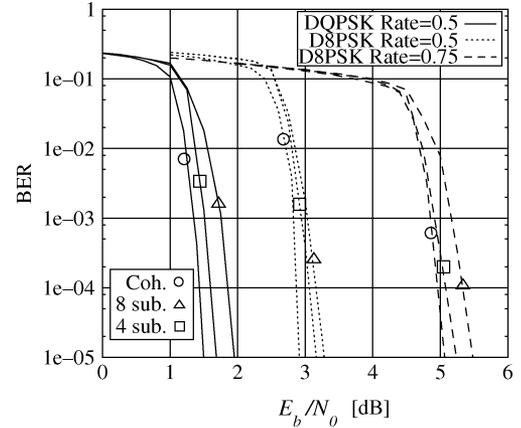


Fig. 9. BER of LDPC codes optimized for DE-QPSK (rate 0.5) and DE-8PSK (rates 0.5 and 0.75) optimized both for AWGN channel and noncoherent channel.

$[0, 2\pi/M)$  [29]. We then choose two possible sets: the first is constituted by eight equally spaced points (in [30], this is shown to lead to negligible performance degradation), i.e.,  $\mathcal{P} = \{0, (1/8)(2\pi/M), \dots, (7/8)(2\pi/M)\}$ , and the second by four equally spaced points, i.e.,  $\mathcal{P} = \{0, (1/4)(2\pi/M), (2/4)(2\pi/M), (3/4)(2\pi/M)\}$ , respectively. The optimization algorithm is then run over the same set of node degrees as in the previous section. The difference between the degree distributions of LDPC code ensembles optimized for DE-PSK and noncoherent detection and those relative to coherent detection is not noticeable. This is true even if the number of quantization levels used for the computation of (11) is reduced to two. An intuitive explanation of this fact is that DE is a technique which makes the communication system insensitive to phase uncertainties, so that the introduction of a further, possibly continuous, phase uncertainty cannot induce a severe system change. Moreover, theoretical results show that, asymptotically, the performance of a noncoherent system approaches that of a coherent system [31]–[34].

In Fig. 9, the performance of optimized LDPC codes for DE-PSK with coherent and noncoherent detection is compared. The considered LDPC codes are optimized for DE-QPSK (with rate 1/2) and DE-8PSK (with rates 1/2 and 3/4); the length of the codeword is 12000 and the maximum allowed iteration number parameters are  $N_i = 30$  and  $N_{\text{LDPC}} = 30$ . The considered numbers of discrete phase values are eight (curves marked with a triangle) and four (curves marked with a square). The curves relative to coherent detection are marked with a circle. It is clear that the phase uncertainty introduces a limited performance loss, as long as the phase quantization is fine enough. Moreover, the results in Fig. 9 show that, while an eight-level phase quantization introduces negligible performance loss, a four-level quantization introduces a performance loss of about 0.4 dB. Further study of the particular considered noncoherent detection algorithm shows that the number of quantization levels can be reduced to a minimum number of 2, causing a performance loss of about 1.7 dB with respect to coherent detection.

## VI. CONCLUSION

In this paper, we have considered communication systems which use, at the transmitter side, LDPC codes concatenated with a CM. In particular, the focus of this paper has been on LDPC codes concatenated with DE modulations. We have made use of the fact that LDPC code families can be characterized by continuous parameters in terms of degree distributions (of check and variable nodes). Upon a particular block decomposition of the receiver, a novel optimization algorithm, based on the use of EXIT charts, has been proposed. This random search algorithm generates good degree distributions which characterize LDPC codes minimizing the SNR convergence threshold of the concatenated LDPC-DE scheme. The analysis of the optimized LDPC codes ensembles in the presence of a DE-PSK modulator has shown that their internal structure significantly differs from that of standard LDPC codes tailored for transmission over an AWGN channel. In particular, it has been shown that the introduction of DE induces a significant increase in the percentage of degree-2 variable nodes: this implies that decoding complexity is likely to reduce. The optimization algorithm has also been used to design LDPC codes optimized for DE-QAM and transmission over the AWGN channel, as well as LDPC codes optimized for DE-PSK and noncoherent detection.

## REFERENCES

- [1] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo codes," *IEEE Trans. Commun.*, vol. 44, no. 10, pp. 1261–1271, Oct. 1996.
- [2] R. G. Gallager, *Low-Density Parity-Check Codes*. Cambridge, MA: MIT Press, 1963.
- [3] D. J. C. MacKay, "Good error correcting codes based on very sparse matrices," *IEEE Trans. Inf. Theory*, vol. 45, no. 2, pp. 399–431, Mar. 1999.
- [4] T. Richardson and R. Urbanke, "The capacity of low-density parity-check codes under message passing decoding," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.
- [5] T. Richardson, A. Shokrollahi, and R. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.
- [6] E. Zehavi, "8-PSK trellis codes for a Rayleigh channel," *IEEE Trans. Commun.*, vol. 40, no. 5, pp. 873–884, May 1992.
- [7] G. Caire, G. Taricco, and E. Biglieri, "Bit-interleaved coded modulation," *IEEE Trans. Inf. Theory*, vol. 44, no. 3, pp. 927–946, May 1998.
- [8] X. Li and J. A. Ritcey, "Bit-interleaved coded modulation with iterative decoding," *IEEE Commun. Lett.*, vol. 1, no. 6, pp. 169–171, Nov. 1997.
- [9] —, "Bit-interleaved coded modulation with iterative decoding using soft feedback," *IEE Electron. Lett.*, vol. 34, no. 10, pp. 942–943, May 1998.
- [10] —, "Trellis-coded modulation with bit interleaving and iterative decoding," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 4, pp. 715–724, Apr. 1999.
- [11] Y. Huang and J. A. Ritcey, "Exit chart analysis of BICM-ID over AWGN channels with SNR mismatch," *IEEE Commun. Lett.*, vol. 8, no. 8, pp. 532–534, Aug. 2004.
- [12] S. ten Brink, G. Kramer, and A. Ashikhmin, "Design of low-density parity-check codes for modulation and detection," *IEEE Trans. Commun.*, vol. 52, no. 4, pp. 670–678, Apr. 2004.
- [13] S. ten Brink, "Convergence of iterative decoding," *IEE Electron. Lett.*, vol. 35, pp. 1117–1119, Jun. 24, 1999.
- [14] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "A soft-input soft-output APP module for iterative decoding of concatenated codes," *IEEE Commun. Lett.*, vol. 1, no. 1, pp. 22–24, Jan. 1997.
- [15] —, "Soft-input soft-output modules for the construction and distributed iterative decoding of code networks," *Eur. Trans. Telecommun.*, vol. 9, no. 2, pp. 155–172, Mar./Apr. 1998.
- [16] M. Ardakani and F. R. Kschischang, "Designing irregular LDPC codes using EXIT charts based on message error rate," in *Proc. IEEE Int. Symp. Inf. Theory*, 2002, p. 454.
- [17] P. Hoeher and J. Lodge, "'Turbo DPSK': Iterative differential PSK demodulation and channel decoding," *IEEE Trans. Commun.*, vol. 47, no. 6, pp. 837–843, Jun. 1999.
- [18] K. R. Narayanan and G. L. Stüber, "A serial concatenation approach in iterative demodulation and decoding," *IEEE Trans. Commun.*, vol. 47, no. 7, pp. 956–961, Jul. 1999.
- [19] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.
- [20] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEE Electron. Lett.*, vol. 33, pp. 457–458, Mar. 1997.
- [21] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inf. Theory*, vol. COM-20, no. 2, pp. 284–287, Mar. 1974.
- [22] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Trans. Commun.*, vol. 49, no. 10, pp. 1727–1737, Oct. 2001.
- [23] M. K. Simon, S. M. Hinedi, and W. C. Lindsey, *Digital Communication Techniques*. Englewood Cliffs, NJ: Prentice-Hall, 1995.
- [24] U. Mengali and A. N. D'Andrea, *Synchronization Techniques for Digital Receivers (Applications of Communications Theory)*. New York: Plenum, 1997.
- [25] M. K. Simon and D. Divsalar, "Multiple symbol differential detection of MPSK," *IEEE Trans. Commun.*, vol. 38, no. 3, pp. 300–308, Mar. 1990.
- [26] G. Colavolpe and R. Raheli, "Noncoherent sequence detection," *IEEE Trans. Commun.*, vol. 47, no. 9, pp. 1376–1385, Sep. 1999.
- [27] M. Tüchler, "Design of serially concatenated systems depending on the block length," *IEEE Trans. Commun.*, vol. 52, no. 2, pp. 209–218, Feb. 2004.
- [28] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic adaptive scheme for global optimization over continuous spaces," *J. Global Optimization*, vol. 11, pp. 341–359, 1997.
- [29] C. Rong-Rong, R. Koetter, U. Madhow, and D. Agrawal, "Joint noncoherent demodulation and decoding for the block fading channel: A practical framework for approaching Shannon capacity," *IEEE Trans. Commun.*, vol. 51, no. 10, pp. 1676–1689, Oct. 2003.
- [30] M. Peleg, S. Shamai (Shitz), and S. Galán, "Iterative decoding for coded noncoherent MPSK communications over phase-noisy AWGN channel," *Proc. IEE Commun.*, vol. 147, pp. 87–95, Apr. 2000.
- [31] G. Colavolpe and R. Raheli, "Theoretical analysis and performance limits of noncoherent sequence detection of coded PSK," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1483–1494, Jul. 2000.
- [32] —, "The capacity of the noncoherent channel," *Eur. Trans. Telecommun.*, vol. 12, no. 4, pp. 289–296, Jul./Aug. 2001.
- [33] G. Ferrari, G. Colavolpe, and R. Raheli, "Finite memory detection: Optimality and reality," in *Proc. Int. Symp. Turbo Codes, Relat. Topics*, Brest, France, Sep. 2003, pp. 1–8.
- [34] M. Peleg and S. Shamai (Shitz), "On the capacity of the blockwise incoherent MPSK channel," *IEEE Trans. Commun.*, vol. 46, no. 5, pp. 603–609, May 1998.



**Michele Franceschini** (S'02) was born in Milan, Italy, in 1977. He received the Dr. Ing. degree ("Laurea," five-year program) in electrical engineering (*summa cum laude*) from the University of Parma, Parma, Italy, in 2002. He is currently working towards the Ph.D. degree in information technology at the University of Parma.

His research interests lie in the area of communication and information theory, with particular emphasis on low-density parity-check code design, advanced signal processing techniques, synchronization, and low-complexity implementation of digital communication systems.

Mr. Franceschini received the Paolo Conti Award in 2003, as the Best Graduate in Information Engineering at the University of Parma in the academic year 2002.



**Gianluigi Ferrari** (S'97–M'03) was born in Parma, Italy, in November 1974. He received the “Laurea” degree (five-year program) (*summa cum laude*) and the Ph.D. degree in electrical engineering from the University of Parma, Parma, Italy, in October 1998 and January 2002, respectively.

From July 2000 to December 2001, he was a Visiting Scholar at the Communication Sciences Institute, University of Southern California, Los Angeles. Since 2002, he has been a Research Professor at the Department of Information Engineering, University of Parma. Between November 2002 and December 2004, he visited, as a Research Associate, the Electrical and Computer Engineering Department, Carnegie Mellon University, Pittsburgh, PA. He has published more than 50 papers in leading international conferences and journals. He is coauthor of *Detection Algorithms for Wireless Communications, With Applications to Wired and Storage Systems* (New York: Wiley, 2004), *Ad Hoc Wireless Networks: A Communication-Theoretic Perspective* (New York: Wiley, 2006), and *Teoria Della Probabilità e Variabili Aleatorie Con Applicazioni* (New York: McGraw-Hill, 2005). He serves as a frequent reviewer for many international journals and conferences. His research interests include digital communication systems design, adaptive signal processing (with particular emphasis on iterative detection techniques for channels with memory), information theory, and ad hoc wireless networking.

Dr. Ferrari is listed in Marquis *Who's Who in the World*, 2005, and in 2000 *Outstanding Intellectuals of the 21st Century*, International Biographical Centre, U.K., 2005. He serves as a Technical Program Member for several international conferences, among them the IEEE International Conference on Communications (ICC'05), Seoul, Korea, 2005, and the IEEE International Conference on Wireless Networks, Communications and Mobile Computing (WirelessCom 2005), Maui, HI, 2005.



**Riccardo Raheli** (M'87) received the Dr. Ing. degree (Laurea) in electrical engineering (*summa cum laude*) from the University of Pisa, Italy, in 1983, the M.S. degree in electrical and computer engineering from the University of Massachusetts, Amherst, in 1986, and the Ph.D. degree (Perfezionamento) in electrical engineering (*summa cum laude*) from the Scuola Superiore di Studi Universitari e di Perfezionamento (now “S. Anna”), Pisa, Italy, in 1987.

From 1986 to 1988, he was with Siemens Telecomunicazioni, Cassina de' Pecchi, Milan, Italy. From 1988 to 1991, he was a Research Professor at the Scuola Superiore di Studi Universitari e di Perfezionamento S. Anna, Pisa, Italy. In 1990, he was a Visiting Assistant Professor at the University of Southern California, Los Angeles. Since 1991, he has been with the University of Parma, Italy, first as a Research Professor, then as an Associate Professor, and currently as a Professor of Communications Engineering. His scientific interests are in the general area of statistical communication theory, with application to wireless, wired and storage systems, and special attention to data detection in uncertain environments, iterative information processing and adaptive algorithms for communications. His research work has lead to numerous scientific publications in leading international journals and conference proceedings, as well as a few industrial patents. In 1990, he conceived (with A. Polydoros) the principle of *Per-Survivor Processing*. He is coauthor of the book *Detection Algorithms for Wireless Communications, with Applications to Wired and Storage systems* (New York: Wiley, 2004).

Dr. Raheli served on the Editorial Board of the IEEE TRANSACTIONS ON COMMUNICATIONS, as an Editor for Detection, Equalization, and Coding from 1999 to 2003. He is currently serving as a Guest Editor of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS (Special Issue on Differential and Noncoherent Wireless Communications), 2005. Since 2003, he has been on the Editorial Board of the *European Transactions on Telecommunications*, as an Editor for Communication Theory.



**Aldo Curtoni** was born in 1979. In 2004 he received the Dr. Ing. degree (Laurea) in telecommunications engineering from the University of Parma with a thesis on multidifferential detection with applications to LDPC codes. Since October, 2004 he has been with Selta S.p.A., Cadeo (PC), Italy. His research interests include iterative detection and telecommunication networks.