Low Complexity Rate Compatible Puncturing Patterns Design for LDPC Codes

Fulvio Babich, Matteo Noschese, Alessandro Soranzo, and Francesca Vatta

Abstract—**In contemporary digital communications design, two major challenges should be addressed: adaptability and flexibility. The system should be capable of flexible and efficient use of all available spectrums and should be adaptable to provide efficient support for the diverse set of service characteristics. These needs imply the necessity of limit-achieving and flexible channel coding techniques, to improve system reliability. Low Density Parity Check (LDPC) codes fit such requirements well, since they are capacity-achieving. Moreover, through puncturing, allowing the adaption of the coding rate to different channel conditions with a single encoder/decoder pair, adaptability and flexibility can be obtained at a low computational cost. In this paper, the design of rate-compatible puncturing patterns for LDPCs is addressed. We use a previously defined formal analysis of a class of punctured LDPC codes through their equivalent parity check matrices. We address a new design criterion for the puncturing patterns using a simplified analysis of the decoding belief propagation algorithm, i.e., considering a Gaussian approximation for message densities under density evolution, and a simple algorithmic method, recently defined by the Authors, to estimate the threshold for regular and irregular LDPC codes on memoryless binary-input continuous-output Additive White Gaussian Noise (AWGN) channels.**

I. Introduction

The main concern in nowadays digital communications is the need of adaptive and flexible communication techniques [1], [2]. The request for higher efficiency, both in bandwidth use and power consumption, furthers the need for limit-achieving, flexible techniques of channel coding. Low Density Parity Check (LDPC) codes fit such requirements well, and puncturing through rate-compatible puncturing patterns can introduce further flexibility (see [3]).

LDPC codes are a class of channel block codes, first introduced in 1960 by Robert Gallager in his doctoral dissertation [4], representing the leading edge in modern channel coding. Due to the technical limitations of that age, LDPC codes were scarcely considered for almost 30 years, a part from Tanner's generalized LDPC definition and graphical representation, presented in his 1981 paper [5] (which was later called Tanner graph), and were re-invented in the mid 1990s by MacKay [6] and Luby et al. [7]. Since it was shown that LDPC performance can approach the Shannon limit as well as Turbo codes, these codes were quickly included in modern communication standards such as IEEE802.11n, 802.16e (Wi-MAX), 10G-BaseT Ethernet, and Digital Video Broadcasting. Recently, they were also proposed as component codes of product code structures [8] for the next generation digital terrestrial broadcasting transmission system [9].

There have been a number of approaches to the so called “one-dimensional (1-D) analysis” of LDPC codes (see, e.g., [10], [11], and [12]), all of them based on the observation that the probability distribution function (pdf) of the decoders log-likelihood ratio (LLR) messages is approximately Gaussian. This approximation is quite accurate for messages sent from variable nodes, but becomes less accurate for messages sent from check nodes, unless the degree distribution polynomial $\rho(x)$ is concentrated on a few degrees, which is verified for regular codes and for some good irregular codes as well [13].

Despite its limits, the use of the Gaussian Approximation (GA) in order to simplify the calculations is very useful when an efficient and low complexity method is needed. See its recent use, e.g., to design low rate LDPC codes in [14], Unequal Error Protection (UEP) LDPC codes in [15], where the authors derive a closed-form expression of the mapping function through GA, and rate-compatible puncturing patterns for LDPC codes in [16], [17], and [18].

There are two types of approach to rate compatibility: code extension [19], [20] where a high-rate “daughter” code is extended to achieve lower rates, and code puncturing. In this paper, rate-compatible LDPC codes are obtained by progressively puncturing a low rate LDPC mother code.

The construction of rate compatible patterns through puncturing may be done in lots of different ways. In [21], for instance, the construction was done optimizing a family of rate-compatible degree distributions as well as the placement of bipartite graph edges. In the already cited [3], rate-compatible puncturing of LDPC codes was considered deriving the density evolution equations for the design of good puncturing degree distributions under the Gaussian Approximation. Ha et al. also proposed an efficient puncturing algorithm for a given mother code in [22]. For finite length (up to several thousand symbols) LDPC codes, Yazdani et al. constructed rate-compatible LDPC codes using puncturing and extending [19].

Manuscript received October 11, 2018; revised November 14, 2018. Date of publication November 30, 2018. The associate editor Prof. Nikola Rožić has been coordinating the review of this manuscript and approved it for publication.

This work has been partially supported by the Italian Ministry of University and Research (MIUR) within the project FRA 2015 (University of Trieste, Italy), entitled “Peer-to-peer millimeter-wave communications in 5G networks: Theoretical modeling and algorithms for massive MIMO systems”

Part of this work was presented at the International Conference on Software, Telecommunications and Computer Networks, SOFTCOM '17, Split, Croatia, September 21-23, 2017.

Fulvio Babich, Matteo Noschese, and Francesca Vatta are with DIA, University of Trieste, Trieste, Italy. A. Soranzo is with DMG, University of Trieste, Trieste, Italy.

E-mails: babich@units.it, matteo.noschese@phd.units.it, soranzo@units.it, vatta@units.it.

Digital Object Identifier (DOI): 10.24138/jcomss.v14i4.639

In [16], we provided a formal analysis of a class of punctured LDPC codes determining their equivalent parity check matrices, given the regular LDPC mother code parity matrix and the puncturing pattern. In [17], we gained further insight into this class addressing a new puncturing design criterion. We used a Gaussian approximation for message densities under density evolution [11], and a simple algorithmic method, recently defined in [23], to estimate the threshold [11] for regular and irregular LDPC codes on memoryless binary-input continuous-output AWGN channels.

In this paper, we greatly simplify the computational complexity of the algorithmic method used in [17] to estimate the Gaussian approximated thresholds, through the results obtained in [24], where, applying the result of [23] to the asymptotical behavior of the recurrent sequence thereby defined, low complexity upper bounds to the exact belief-propagation decoding thresholds have been derived. The analysis in [24] gave rise to a simple algebraic expression of the upper bound on irregular LDPC belief-propagation decoding thresholds using GA, thus allowing its simple determination from the codes parameters.

The paper is organized as follows. In the next section, we recall the class of rate-compatible punctured codes, obtained by puncturing a regular LDPC low rate mother code, addressed in [16], and therein described through their equivalent parity check matrices. Since the punctured codes of this class are irregular, in Section III the Gaussian approximation of message densities is recalled for irregular LDPC codes. In Section IV we recall the simple algorithmic method, recently defined in [23], to estimate the threshold for irregular LDPC codes on memoryless binary-input continuous-output AWGN channels, and, in Section V, the simple algebraic expressions of the upper bounds on irregular LDPC belief-propagation decoding thresholds using GA obtained in [24]. In Section VI, the proposed puncturing design criterion is outlined and the results of the puncturing design are presented. Finally, Section VII summarizes the results of the paper.

II. A CLASS OF RATE-COMPATIBLE PUNCTURED LDPC CODES

In adaptive communication systems, error correcting codes can be required to be flexible with respect to their code rates depending on the current channel state. Rate adaptability can be realized with several pairs of encoders and decoders, one pair for each desired code rate. However, this is undesirable due to the high amount of added complexity. Rate adaptability can also be realized at a low computational cost by puncturing a low rate channel code, resulting in only one encoder and one decoder.

The concept of rate-compatible puncturing has been presented for the first time in [25], where a particular family of convolutional codes, called in the paper rate-compatible punctured convolutional codes, is obtained by adding a rate-compatibility restriction to the puncturing rule. This restriction requires that the rates are organized in a hierarchy, where all coded bits of a higher rate code are used by all lower rate codes. In other words, the high rate codes are embedded into the lower rate codes of the family. The two main applications of these classes of codes are hybrid incremental ARQ/FEC schemes [26] and unequal error protection [27] of an information sequence.

An LDPC code can be fully represented by its Tanner graph [5] or, equivalently, the associated parity check matrix. As far as the Tanner graphical representation is concerned, we assume that, when performing rate-compatible puncturing, the Tanner graph associated with a higher rate code is always a subgraph of all the Tanner graphs associated with lower rate codes of the family. In [16], we provided a formal analysis of this class of punctured LDPC codes determining the equivalent parity matrix for a punctured code, given the mother code parity matrix and the puncturing pattern. The puncturing of a bit from an LDPC codeword is equivalent to the elimination of a variable node from the correspondent Tanner graph. Being k the degree (see [3]) of the bit to be punctured, or equivalently, the number of check equations the bit is involved into, the k parity check equations involving this bit have to be eliminated from the parity matrix (or, equivalently, in the associated Tanner graph, the k edges connecting the corresponding variable node to the check nodes have to be cancelled). To this end, one of its k parity check equations has to be subtracted from the other $k-1$ parity check equations and from itself. This can be done choosing the parity equation to be subtracted (or summed modulo-2) in k different ways, giving origin to k different equivalent parity matrices. Here, as in [16] and [17], to eliminate from the parity matrix the k parity check equations involving the bit to be punctured, we have considered the subtraction of the first parity check equation only.

Example

Consider a $(8, 4)$ linear block code with exactly $d_c = 2$ 1's in each column and $d_v = 4$ 1's in each row, i.e., a regular $(4, 2)$ LDPC code with parity check matrix H:

$$H = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \qquad (1)$$

Observe that variable nodes c_2, c_4, c_5, and c_8 are connected to check node f_1 in accordance to the fact that, in the first row of H, $h_{12} = h_{14} = h_{15} = h_{18} = 1$ (all others are zero). Observe that analogous situations hold for check nodes f_2, f_3, and f_4, which correspond to rows 2, 3, and 4 of H, respectively. Note that, as follows from the fact that $cH^T = 0$, the bit values connected to the same check node must sum to zero. We may also proceed along columns. For example, note that variable node c_1 is connected to check nodes f_2 and f_4 in accordance with the fact that, in the first column of h, $h_{21} = h_{41} = 1$.

Given a codeword $c = (c_1, \ldots, c_8)$, puncturing the bit c_i is equivalent to subtracting one of the $k = 2$ parity check equations it is involved into from matrix H, since $k = 2$ is the degree of bit c_i, $\forall i$. c_i is associated with two check equations, say f_j and f_l. We choose f_j as the equation to subtract: this is equivalent to modulo-2 addition of matrix rows. This results in one all-zero row in the matrix corresponding to the punctured position. Since c_i has degree $k = 2$, there are two different equivalent "punctured" matrices, one for each of the associated check equations f_j and f_l.

Assume that the c_8 bit is punctured. f_1 is subtracted from f_3 and from itself, resulting into the "punctured" $\hat{H}$ parity matrix:

$$\hat{H} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \tag{2}$$

Alternatively, f_3 can be chosen and subtracted from f_1 and from itself. The resulting parity matrix $\hat{\hat{H}}$ is completely equivalent to $\hat{H}$: this is easy to see in this example since the only difference between the two matrices is having the f_1 and f_3 rows swapped.

$$\hat{\hat{H}} = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \tag{3}$$

III. Gaussian Approximation for LDPC Codes

Tanner considered LDPC codes (and a generalization) and showed how they may be represented effectively by a so-called bipartite graph, now called Tanner graph [5]. The Tanner graph of an LDPC code is analogous to the trellis of a convolutional code in that it provides a complete representation of the code and it aids in the descritpion of the decoding algorithm. A *bipartite graph* is a graph (nodes connected by edges) whose nodes may be separated into two types, and edges may only connect two nodes of different types. The two types of nodes in a Tanner graph are the *variable nodes* and the *check nodes*. The Tanner graph of a code is drawn according to the following rule: check node j is connected to variable node i whenever element h_{ij} in H is a 1.

Under belief-propagation decoding of an LDPC code, variable nodes and check nodes exchange "messages" between each other iteratively. A check node gets messages from the variable nodes it is connected to ("neighbours"), processes the messages and sends the result back to its neighbouring variable nodes. Similarly, a variable node receives messages from its check nodes and returns the processed message back to them. We note that each output (both of variable and check nodes) is a function of all incoming messages, except the message coming from the edge where the message will be sent out. This restriction is fundamental to compute the correct marginal *a posteriori* probabilities. This *two-step* procedure is repeated many times. After l iterations, the variable node decodes its associated bit based on all information obtained from its depth-l subgraph of neighbours.

Consider a regular (d_v, d_c) LDPC code, being d_v and d_c the number of neighbors of a variable node and of a check node respectively, under the *local tree assumption*. Under this hypothesis, the girth (i.e., the length of the shortest cycle) is large enough so that the depth-l subgraph forms a tree. The subgraph associated to each variable node is independent from those of other variable nodes, and as a result, so are the messages.

In soft-decision belief-propagation decoding, the messages are the log-likelihood ratios (LLRs) of received bits. The output message v of a variable node is:

$$v = \log\Big(\frac{P\{y|x=+1\}}{P\{y|x=-1\}}\Big)$$

where x is the bit value of the node, and y is all the information available to the node up to the present iteration. The edge carrying the information associated to v is excluded. Similarly, the output message u of a check node is defined as:

$$u = \log\Big(\frac{P\{y'|x'=+1\}}{P\{y'|x'=-1\}}\Big)$$

where x' is the bit value of the variable node getting the message from the check node, and y' is the information available to the check node up to the present iteration, disregarding the edge carrying u.

Consider the first step of the belief-propagation algorithm: the variable nodes send their messages to the connected check nodes. The message v is

$$v = \sum_{i=0}^{d_v-1} u_i$$

where u_i, $i = 1, \ldots, d_v - 1$, are the incoming LLRs from the neighbours of the variable node, and u_0 is the observed LLR of the output bit associated with the variable node. More specifically, at the l^{th} iteration, with $l > 1$ the message is:

$$v^{(l)} = u_0 + \sum_{i=0}^{d_v-1} u_i^{(l-1)}$$

In the second step, the check nodes send their messages to the variable nodes. At the l^{th} iteration, this can be expressed by the following "tanh rule" :

$$\tanh\frac{u^{(l)}}{2} = \prod_{j=1}^{d_c-1} \tanh\frac{v_j^{(l)}}{2}$$

where v_j, $j = 1, \ldots, d_c - 1$, are the incoming LLRs from the $d_c - 1$ neighbours of a check node, and u is the message sent to the remaining neighbour.

This two-stage procedure, called *density evolution*, is used to calculate the thresholds for belief-propagation decoding. Those are defined as the maximum noise level such that the probability of error tends to zero as the number of iterations tends to infinity [11].

Irregular LDPC codes [11] are defined by specifying the distribution of the node degrees in their Tanner graphs. In particular, in the edge-perspective degree distribution, λ_i is the fraction of edges in the Tanner graph connecting to a degree-i variable node, and ρ_j is the fraction of edges connecting to a degree-j check node. To specify the degree distribution, the following polynomials are defined:

$$\lambda(x) = \sum_{i=i_1}^{d_l} \lambda_i x^{i-1} \tag{4}$$

$$\rho(x) = \sum_{j=2}^{d_r} \rho_j x^{j-1} \tag{5}$$

being d_l (respectively d_r) the maximum variable (respectively check) node degree.

The d_l-uple $\{\lambda_i\}$ and d_r-uple $\{\rho_j\}$ both add up to 1. Given the polynomials $\lambda(x)$ and $\rho(x)$, a (λ, ρ)-LDPC code of block length n is defined as a linear code with a Tanner graph in n variable nodes such that the edge-perspective variable-node degree distribution is given by $\lambda(x)$ and the edge-perspective check-node degree distribution is given by $\rho(x)$.

Consider an irregular LDPC code with edge-perspective degree distributions $\lambda(x)$ and $\rho(x)$. The number of edges connecting to each node is not constant: hence, the messages a node receives from its neighbours are a "mixture" of densities from neighbour nodes of different degrees.

Following the analysis conducted in [11], and recalled in [16], and applying the simple algebraic expression of the upper bound on irregular LDPC belief-propagation decoding thresholds using GA obtained in [24], we derive a new puncturing design procedure for LDPC codes, providing a computationally lower cost quality assessment to a puncturing pattern than the one provided in [17]. For completeness and clarity sake, this analysis is briefly recalled here so as to provide the reader with a reference to the mathematical functions we need in the optimization of the puncturing procedure.

Denote with v the output message of a variable node and with u the output message of a check node. Assuming that irregular LDPC codes message distributions for AWGN channels are approximately Gaussian, since a Gaussian is completely specified by its mean and variance, we need only the means and variances of u and v during iterations. Moreover, since the variance σ^2 is shown in [11] to be related to the mean m by the relation $\sigma^2 = 2m$, due to the symmetry condition, we can keep the means only. Denote the means of u and v by $m_u^{(l)}$ and $m_v^{(l)}$ at the l-th iteration, respectively. Moreover, the LLR message u_0 from the channel can be assumed to be Gaussian with mean $m_{u_0} = 2/\sigma_n^2$ and variance $4/\sigma_n^2$, where $\sigma_n^2 = N_0/2$ is the variance of the channel noise.

For a degree-i variable node at the l^{th} iteration, the mean of the output yields:

$$m_{v,i}^{(l)} = m_{u_0} + (i-1)m_u^{(l-1)} \tag{6}$$

where m_{u_0} is the mean of u_0 and $m_u^{(l-1)}$ is the mean of u at the $(l-1)$-th iteration.

Defining $\phi(x)$ as in *Definition 1* in [11], the update rule for an irregular code becomes:

$$m_{u,j}^{(l)} = \phi^{-1}\left(1 - \left[1 - \sum_{i=i_1}^{d_l} \lambda_i \phi(m_{v,i}^{(l)})\right]^{j-1}\right) \tag{7}$$

The output of a variable node is characterized by its mean, which is the sum of the means of incoming densities, since they are independent from each other. The mean $m_u^{(l)}$ can be calculated by linearly combining the means $m_{u,j}^{(l)}$:

$$m_u^{(l)} = \sum_{j=2}^{d_r} \rho_j \phi^{-1}\Big(1 - \Big[1 - \sum_{i=i_1}^{d_l} \lambda_i \phi\big(m_{u_0} + (i-1)m_u^{(l-1)}\big)\Big]^{j-1}\Big) \tag{8}$$

Defining $s = m_{u_0}$ and $t_l = m_u^{(l)}$, (8) may be rewritten as

$$t_l = f(s, t_{l-1}) \tag{9}$$

where the function $f(s,t)$ is defined, through the $f_j(s,t)$, for $0 < s < \infty$ and $0 \le t < \infty$ as:

$$f_j(s,t) := \phi^{-1}\Big(1 - \Big[1 - \sum_{i=i_1}^{d_l} \lambda_i \phi(s + (i-1)t)\Big]^{j-1}\Big) \tag{10}$$

$$f(s,t) := \sum_{j=2}^{d_r} \rho_j f_j(s,t) \tag{11}$$

IV. Low Complexity Approximation of the Exact Belief-propagation Decoding Thresholds

In his introduction to regular LDPC codes [4], Gallager first noticed that these codes exhibit the so called "threshold phenomenon" on binary symmetric channels (BSC), as explained also in [11]: "as the block length tends to infinity, an arbitrarily small bit-error probability can be achieved if the noise level is smaller than a certain threshold". Later Luby [7] showed that *irregular* LDPC codes, performing better than regular ones, exhibit the threshold phenomenon, too.

In [23] we presented a mathematical method to allow the noise thresholds evaluation, exploiting a moderately complex technique. This algorithm replaced the empirical procedure suggested in [11] for regular codes, extending its application to irregular codes, too. Our algorithm is based on the idea that the problem of determining the last value such that a recurrent sequence converges (or, equivalently, the first value such that the sequence diverges), can be seen as a "static" problem that can be assigned to a standard software, leading to a significant computational simplification. This algorithm has been derived using the quadratic degeneracy theory, thus transforming a recurrence relation convergence problem in a problem of mathematical analysis.

Applying the method defined in [23], we derive an approximation of the exact belief-propagation decoding thresholds by means of the Gaussian approximation. Instead of searching the minimum value of the parameter $s = m_{u_0}$ granting the convergence of (8), we solve a problem of quadratic degeneracy which can be assigned to a standard software. When the second derivative $f_{tt}(s,t)$ is $\neq 0$ the problem of quadratic degeneracy is the system of equations

$$\begin{cases} f(s,t) = t \\ f_t(s,t) = 1 \end{cases} \tag{12}$$

where $f_t(s,t)$ is the partial derivative of $f(s,t)$ with respect to t.

The solution of (12) gives the value $s^* = m^*_{u_0}$, the minimum $s = m_{u_0}$ granting the convergence of (8). Defining, in an only formally different way from [11],

$$\Delta(s,t) := f(s,t) - t \tag{13}$$

and computing its first partial derivative with respect to t,

$$\Delta_t(s,t) = f_t(s,t) - 1 \tag{14}$$

(12) may be rewritten as:

$$\begin{cases} \Delta(s,t) = 0 \\ \Delta_t(s,t) = 0 \end{cases} \tag{15}$$

Its solution (s^*, t^*) determines an approximation σ^* of the exact belief-propagation decoding threshold defined as $\sigma^* := \sqrt{\frac{2}{s^*}}$.

To find the solution of (15), an explicitly invertible approximation of the function $\phi(x)$ is needed. In [11], a piecewise defined approximation by elementary functions was given:

$$\phi(x) \approx \begin{cases} \mathrm{e}^{-0.4527\,x^{0.86}+0.0218} & \text{if} \quad 0 \le x \le 10 \\ \sqrt{\frac{\pi}{x}}\,\mathrm{e}^{-x/4}\Big(1 - \frac{10}{7\,x}\Big) & \text{if} \quad x > 10 \end{cases} \tag{16}$$

which is explicitly invertible only for $0 \le x \le 10$. A graph of $\phi(x)$ for $0 \le x \le 10$ may be found, e.g., in [17], where the approximation of $\phi(x)$, called $\tilde{\phi}(x)$, and its inverse, called $\tilde{\phi}^{-1}(y)$ were derived using a numeric integration.

V. Upper Bound on Decoding Thresholds

When t is large, i.e., the probability of error is small, the behaviour of $\Delta(s,t) = f(s,t) - t$ is given in [11] by:

$$\Delta(s,t) = s + (i_1 - 2)t - 4\log\lambda_{i_1} - 4\sum_{j=2}^{d_r} \rho_j \log(j-1) + O(t^{-1}) \tag{17}$$

where λ_{i_1} is the first non-zero coefficient of the polynomial $\lambda(x)$ and i_1 is its index. Using the Jensen's inequality:

$$\prod_{j=2}^{d_r} (j-1)^{\rho_j} \le \sum_{j=2}^{d_r} \rho_j (j-1) \tag{18}$$

and applying it to (17), this can be rewritten as:

$$\Delta(s,t) = s + (i_1 - 2)t - 4\log\lambda_{i_1} - 4\log\Big(\sum_{j=2}^{d_r} \rho_j (j-1)\Big) + O(t^{-1}) \tag{19}$$

Ignoring the $O(t^{-1})$, and solving (15), with $\Delta(s,t)$ obtained in (19), with respect to s, we get:

$$s^* = 4\log\lambda_2 + 4\log\Big(\sum_{j=2}^{d_r} (j-1)\rho_j\Big) \tag{20}$$

Taking $\sigma^* = \sqrt{\frac{2}{s^*}}$, we get an upper bound on decoding threshold.

This is the same result found in [13], but therein obtained following a different way, i.e., determining the upper bound on threshold σ^* as the σ value satisfying the stability condition for the BI-AWGN channel:

$$\lambda'(0)\rho'(1) < e^{\frac{1}{2\sigma^2}} \tag{21}$$

Solving (21) with respect to σ, the following upper bound on the threshold can be obtained:

$$\sigma^*(\lambda, \rho) \le \frac{1}{\sqrt{2\log(\lambda'(0)\rho'(1))}} \tag{22}$$

Being $\lambda'(0) = \sum_{i=i_1}^{d_l} (i-1)\lambda_i x^{i-2}|_{x=0} = \lambda_2$ and $\rho'(1) = \sum_{j=2}^{d_r} (j-1)\rho_j x^{j-2}|_{x=1} = \sum_{j=2}^{d_r} (j-1)\rho_j$, we obtain

$$\sigma^*(\lambda, \rho)^2 \le \frac{1}{2\log(\lambda_2 \sum_{j=2}^{d_r} (j-1)\rho_j)} \tag{23}$$

from which

$$s = \frac{2}{\sigma^*(\lambda, \rho)^2} \ge 4\log(\lambda_2 \sum_{j=2}^{d_r} (j-1)\rho_j) = s^* \tag{24}$$

Namely, (24) gives the same bound we obtained in (20), following a different approach.

VI. Puncturing Criterion

A. Puncturing procedure

Following and integrating the analysis in [11] where, to optimize the degree distribution for an irregular LDPC code, $\Delta t = f(s,t) - t$ is maximized (ignoring the $O(t^{-1})$ term), an optimal design criterion for the puncturing pattern can be derived. Specifically, Δt depends on the punctured LDPC code parameters s, i_1, λ_{i_1}, and all ρ_j, where s is the threshold, λ_{i_1} is the first non-zero coefficient of the polynomial $\lambda(x)$, i_1 is its index, and ρ_j is the j-th coefficient of $\rho(x)$. Notice that s depends, via (12), (11), and (10) on all λ_i's and ρ_j's:

$$s = s(\lambda_{i_1}, ..., \lambda_{d_l}, \rho_2, ..., \rho_{d_r}) = s(\Gamma) \tag{25}$$

where $\Gamma = (\lambda_{i_1}, ..., \lambda_{d_l}, \rho_2, ..., \rho_{d_r})$ is the vector of all the parameters, but here, instead of determining it as solution of (15), we assume $s(\Gamma) = s^*$ given by (20) and (24), since $\sigma^* = \sqrt{\frac{2}{s^*}}$ (called σ^*_{Jensen} in [24]) was shown in [24] to give the tighter upper bound on threshold results.

Thus, $\Delta(s,t)$ in (17) depends on all λ_i's, ρ_j's, and t:

$$\Delta(s,t) = \Delta(\lambda_{i_1}, ..., \lambda_{d_l}, \rho_2, ..., \rho_{d_r}, t) \tag{26}$$

and, distinguishing the variable from the parameters,

$$\Delta(s,t) = \delta_\Gamma(t) = (i_1 - 2)t + q_\Gamma \tag{27}$$

where $q_\Gamma = s(\Gamma) - 4\log\lambda_{i_1} - 4\sum_{j=2}^{d_r} \rho_j \log(j-1)$. Notice that, assuming $s(\Gamma) = s^*$ given by (20) and (24), instead of determining s^* as solution of (15), the computation of q_Γ is greatly simplified.

Using $\delta_\Gamma(t)$ as a function (of t with parameter Γ) measuring the performance, the best candidate vectorial puncturing pattern is the one with the greatest slope $i_1 - 2$, and then the greatest additive constant q_Γ. Thus, the goal of finding the best candidate puncturing pattern can be fulfilled through the following steps:

1) Find the equivalent parity check matrix of the irregular LDPC code obtained applying the candidate puncturing pattern to the regular LDPC mother code.
2) Find its polynomials $\lambda(x)$ and $\rho(x)$ in order to determine Γ and, from it, $s(\Gamma)$, i_1, and q_Γ.

3) Select the best candidate as the one maximizing $\delta_\Gamma(t)$ as said above.

Since, as shown in [24], given an ensemble of LDPC codes with decoding thresholds σ^* in ascending order, the upper bounds on them (see Table I in [24]) keep the same order, the puncturing procedure gave the same results of the ones obtained in [17], i.e., the same puncturing patterns were selected by the procedure, but the computational complexity of (27) was greatly reduced using the bounds on thresholds we found in [24], insted of solving (15) as in [17].

B. The pattern generation algorithm

Given a rate-R_0 mother code with block length n, to attain a target rate R_1, p bits must be punctured, where $p = n(1 - \frac{R_0}{R_1})$. Since the number of possible puncturing patterns is $\binom{n}{p}$, an exhaustive search could be feasible only for short block lengths, but it is well known that the Shannon limit is well approached for block lengths tending to infinity. Thus, the proposed algorithm implements an incremental approach choosing only one incremental punctured bit position at each iteration.

The vectorial value Γ and, from it, $s(\Gamma)$, i_1, λ_{i_1}, and all ρ_j's, are computed for each puncturing position and, within each iteration, the bit position giving place to the best choice of Δt, as explained above, is chosen. Since only one bit position is chosen at each iteration, and, once chosen, is kept for the following iterations of the algorithm, this generates rate-compatible patterns. Specifically, all possible rates between the mother code rate and the final rate. As such, the algorithm may not produce the optimal solution, but go through a number of suboptimal solutions. On the other hand, these intermediate code rates are useful for practical applications (see, e.g., [26] and [27]). Recall that, defining the mother code rate R_0 as $\frac{k}{n}$, the obtained intermediate higher rates are $\frac{k}{n-1}$, $\frac{k}{n-2}$, $\cdots$, $\frac{k}{n-p} = R_1$. It is hence sufficient to run a single instance of the algorithm with a target rate equal to the highest desired code rate to obtain the puncturing patterns associated with all the possible rates in between. It is also possible to define a puncturing pattern for a specific rate (e.g., one that is known to have good performances) as the "starting" pattern to be further punctured, in order to obtain a higher code rate.

C. Simulation results

A custom software based on [28] and [29] was employed to simulate the performances of punctured LDPC codes over an AWGN channel, assuming a BPSK (Binary Phase Shift Keying) modulator. The belief propagation algorithm, also called message passing or sum-product algorithm, commonly employed for LDPC decoding, has been adopted, employing soft decision.

When applying a puncturing pattern to a codeword, some of the bits are removed from it according to the pattern. The removed bits are not transmitted on the channel, resulting in a higher code rate. The puncturing pattern has been defined by listing the positions to be punctured. To perform puncturing, the mother code encoder output symbols corresponding to punctured positions are simply discarded. At the receiver end, the mother code decoder is employed "filling" the punctured positions with "neutral" symbols[1], since no information is provided about the punctured bits. In the specific case of a binary modulation with soft decision decoding, the input of the channel decoder are the LLR for each bit: the "missing" LLR values are replaced with 0. This means no information is known about the coded bit.

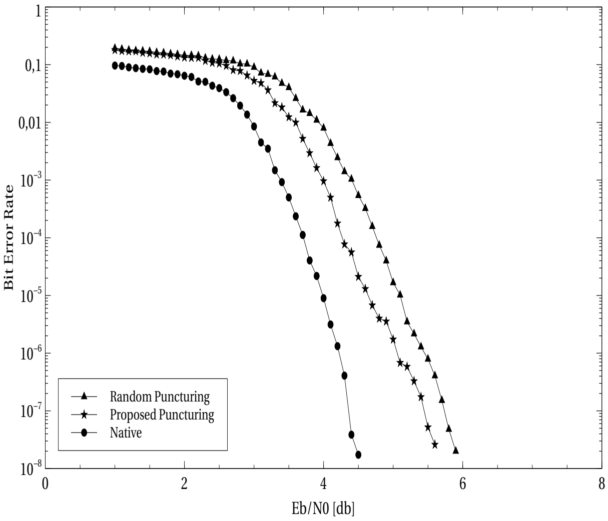

Fig. 1. BER vs. E_b/N_0 in dB of rate-2/3 codes with $n = 1000$. Curve with circle markers: native rate-2/3 code. Curve with triangle markers: punctured rate-2/3 code with random puncturing. Curve with star markers: punctured rate-2/3 code with optimum puncturing.

In the following, the results of the simulations are shown. The performances of the punctured codes generated with the proposed algorithm are compared with those of randomly generated puncturing patterns and with those of the same-rate native codes.

Regular LDPC codes with rate $R_0 = \frac{1}{2} = 1 - \frac{w_c}{w_r}$ have been taken as mother codes, where $w_c = 3$ and $w_r = 6$ are the column and the row weights of their parity check matrices, respectively. Each of them has been punctured to achieve a target rate $R_1 = \frac{2}{3}$. The rate-R_1 native code used for comparison had the same block length $n_1 = n - p$ of the punctured code.

With a block length $n = 1000$, to obtain a rate-2/3 code $p = 250$ bits need to be punctured. Hence, the native code has a block length of $n_1 = 750$. Fig. 1 reports the Bit Error Rate (BER) vs. E_b/N_0 with $n = 1000$ of the native code, of the punctured rate-2/3 code with random puncturing, and of the punctured rate-2/3 code with the proposed puncturing. Similarly, Fig. 2 reports the Block Error Rate (BLER) vs. E_b/N_0 of the native code, of the punctured rate-2/3 code with random puncturing, and of the punctured rate-2/3 code with the proposed puncturing.

The results show that the proposed puncturing design procedure leads to good achievements in coding gain. The best achievement is shown in Fig. 1, where the punctured rate-2/3

[1] It is always assumed that the decoder knows the locations of punctured coded symbols.

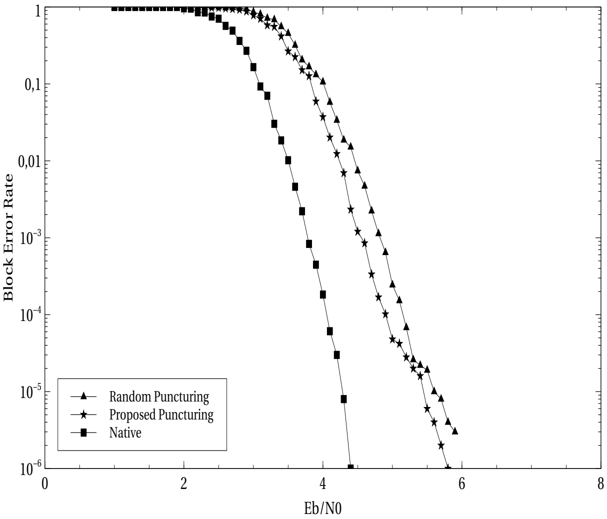

Fig. 2. BLER vs. E_b/N_0 in dB of rate-2/3 codes with $n = 1000$. Curve with square markers: native rate-2/3 code. Curve with triangle markers: punctured rate-2/3 code with random puncturing. Curve with star markers: punctured rate-2/3 code with optimum puncturing.

code performance, obtained applying the outlined procedure, is very close to the native rate-2/3 code performance. As far as Fig. 2 is concerned, the same good achievements in coding gain may be seen, but not as significant as those reported in Fig. 1.

The performance results shown in Figs. 1 and 2 have been obtained assuming a BPSK modulator. However, when bandwidth is constrained, higher order modulations, such as, e.g., quadrature PSK (QPSK), typical of most deep space missions, may be needed, in order to more effectively trade off power efficiency, bandwidth efficiency, and complexity [30]. For these applications, system engineers may also be interested in knowing the performance of LDPC codes when used with other higher order modulations, such as, e.g., M-PSK, with $M > 4$, and quadrature amplitude modulation (QAM).

The performance of a code when used with a nonbinary modulation may be approximated from its BPSK performance by its code *imperfectness*, defined in [31]. First, the imperfectness of the code in conjunction with BPSK is determined by measuring the difference between the code's required bit signal-to-noise ratio E_b/N_0 to attain a given word error probability (BLER) and the minimum possible E_b/N_0 required to attain the same BLER as implied by the sphere-packing bound[2] for codes with the same block size n and code rate R_1 [31]. The imperfectness evaluated in this way may then be applied with respect to the channel capacity curve for higher order modulations to get an approximated performance of the code when used in conjunction with these modulations. The imperfectness approximation has generally been found to be fairly accurate, to within about 0.5 dB, over a wide variety of codes and modulations [32].

To reduce the complexity of this evaluation, the approximation of the channel capacity curve $C(\gamma)$ for BPSK modulation we found in [23], where γ is the symbol signal-to-noise ratio, may be useful. In particular, in [23], considering an AWGN channel with binary input and soft output, the channel capacity $C(\gamma)$ for BPSK modulation [33] (see also [34]), and its inverse $C^{-1}(r)$, have been approximated by the following couple of functions, one inverse of the other, with forms, respectively,

$$1 - \mathrm{e}^{u\,\gamma^{w}+v} \qquad \left(\frac{\log(1-r)-v}{u}\right)^{\frac{1}{w}} \tag{28}$$

with

$$u = -1.286 \qquad v = 0.01022 \qquad w = 0.9308 \tag{29}$$

for which we found also the limitations for the absolute and relative errors.

The function $C_1(\gamma) = 1 - \mathrm{e}^{u\,\gamma^{w}+v}$ approximates $C(\gamma)$ very accurately in the monodimensional case (BPSK modulation) even if it is not correct for γ very close to 0, due to the positive value of the parameter v. It can be extended to the M-dimensional case giving:

$$C_M(\gamma) = M(1 - \mathrm{e}^{u\,(\frac{\gamma}{M})^{w}+v}) \tag{30}$$

Opposite to the standard AWGN model, impulsive noise[3] is a non-stationary signal and may represent the main disturbing signal, such is the case of Power Line Communications Systems (PLC) [35]. This undesirable signal is also reported on Digital Television [36], audio broadcasting [37], xDSL technologies [38] as well as Underwater Acoustic systems [39]. In the frequency range from some hundreds of kHz up to 20MHz, the measurement of impulsive noise shows that it exhibits a duration of some microseconds up to a few milliseconds [40], and it could be as high as 40 dB above the background noise [41].

As explained by the pioneer work of Middleton [42], the modelling of impulsive noise allows to provide a realistic and quantitative description of this kind of interference, to specify and guide experiments for measuring its effects. Middleton's Class A noise model[4] describes the probability density function (pdf) of impulsive noise as the linear superposition of Gaussian functions with different variances [43]:

$$p_X(x) = \sum_{m=0}^{\infty} \frac{A^m \mathrm{e}^{-A}}{m!} \frac{\exp(-x^2/2\sigma_m^2)}{\sqrt{2\pi\sigma_m^2}} \tag{31}$$

with

$$\sigma_m^2 = \left(1 + \frac{1}{\Gamma}\right) \frac{k/A + \Gamma}{1 + \Gamma} \sigma_N^2 \tag{32}$$

where the parameter A is called impulsive index. The parameter Γ is the background-to-impulsive noise ratio, namely, the value of Γ gives how strong the impulsive noise is with respect to the AWGN component [44]. The value of $\sigma_N^2 = \sigma_n^2 + \sigma_I^2$

[2] The sphere-packing bound would be reached with equality only if the code were a *perfect* code for this channel, i.e., if equal-size nonintersecting cones could be drawn around every codeword to completely fill the 2^n dimensional space.

[3] Impulsive noise is mainly comprised by short duration noisy pulses. This noise is caused by switching devices, channel environment, clicks from computer keyboards, loads not synchronous with the power frequency, etc.

[4] Class A noise channel proposed by Middleton [43] is one of the non-Gaussian noise channels and is currently applied to the modelling of man-made impulsive noise channels, such as the wireless channel, the power line channel, and so on.

is the total power given by the contribution of the impulsive noise power σ_I^2 and the AWGN noise power σ_n^2.

As may be seen from (31) and (32), the statistical characteristic of Class A noise is much different from that of Gaussian noise, since the occurrence of the impulsive noise may cause error bursts in data transmission. Therefore, the conventional decoders optimized for the additive white Gaussian noise (AWGN) channel, in general, are not suitable for Class A noise environment. In particular, as far as LDPC codes decoding is concerned, in the literature a number of modified sum-product decoding algorithms suitable for this channel, called Additive White class A Noise (AWAN) channel, may be found (see, e.g., [45], [46], and [47]).

VII. Conclusions

This paper was focused on the search for limit-achieving and flexible channel coding techniques. With this goal in mind, the design of rate-compatible puncturing patterns for LDPC codes was addressed. A low complexity performance assessment of the puncturing patterns was presented and formally justified, thanks to the work in [11] and to the bounds on thresholds found in [24]. Namely, we greatly simplified the computational complexity of the algorithmic method used in [17] to estimate the Gaussian approximated thresholds, through the results obtained in [24], where, applying the results of [23] to the asymptotical behavior of the recurrent sequence thereby defined, low complexity upper bounds to the exact belief-propagation decoding thresholds have been derived. The analysis in [24] gave rise to a simple algebraic expression of the upper bound on irregular LDPC belief-propagation decoding thresholds using GA, thus allowing its simple determination from the codes parameters.

A possible follow up of this work could be the determination of upper bounds on the word residual error probability for the punctured codes object of this study. This result could give rise to an extension of the puncturing criterion including the minimization of the word residual error probability, thus optimizing the error floor performance.

References

[1] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. K. Soong, and J. C. Zhang, "What will 5G be?", *IEEE Journal on Selected Areas in Communications*, Vol. 32, No. 6, June 2014, pp. 1065-1082. DOI: 10.1109/JSAC.2014.2328098.

[2] C. Koller, A. Graell i Amat, J. Kliewer, F. Vatta, K. Sh. Zigangirov, and D. J. Costello, Jr., "Analysis and design of tuned turbo codes", *IEEE Trans. on Information Theory*, Vol. 58, No. 7, July 2012, pp. 4796-4813. DOI:10.1109/TIT.2012.2195711.

[3] J. Ha, J. Kim, and S.W. McLaughlin, "Rate-compatible puncturing of low-density parity-check codes", *IEEE Trans. on Information Theory*, Vol. 50, No. 11, November 2004, pp. 2824-2836. DOI: 10.1109/TIT.2004.836667.

[4] Robert G. Gallager, "Low-density parity-check codes", MIT press Cambridge, 1963.

[5] R. M. Tanner, "A recursive approach to low complexity codes", *IEEE Transactions on Information Theory*, vol. 27, no. 5, September 1981, pp. 533-547. DOI: 10.1109/TIT.1981.1056404.

[6] David J. C. Mackay, "Good error correcting codes based on very sparse matrices", *IEEE Transactions on Information Theory*, Vol. 45, No. 2, March 1999, pp. 399-431. DOI: 10.1109/18.748992.

[7] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, "Analysis of Low Density Codes and improved design using Irregular Graphs", *IEEE Transactions on Information Theory*, Vol. 47, No. 2, February 2001, pp. 585-598. DOI: 10.1109/18.910576.

[8] V. Sidorenko, M. Bossert, and F. Vatta, "Properties and encoding aspects of direct product convolutional codes", *Proc. of the 2012 IEEE International Symposium on Information Theory, ISIT'12*, Boston, U.S.A., July 1-6, 2012, pp. 2351-2355. DOI: 10.1109/ISIT.2012.6283934.

[9] B. Liu, Y. Li, B. Rong, L. Gui, and Y. Wu, "LDPC-RS Product Codes for Digital Terrestrial Broadcasting Transmission System", *IEEE Trans. on Broadcasting*, Vol. 60, No. 1, March 2014, pp. 38-49. DOI: 10.1109/TBC.2013.2291359.

[10] I. Land, S. Huettinger, P. A. Hoeher, and J. B. Huber, "Bounds on information combining", *IEEE Transactions on Information Theory*, Vol. 51, No. 2, Feb. 2005, pp. 612-619. DOI: 10.1109/TIT.2004.840883.

[11] S.-Y. Chung, T. J. Richardson, and R. Urbanke, "Analysis of Sum-Product Decoding of Low-Density Parity-Check Codes Using a Gaussian Approximation", *IEEE Trans. on Inf. Theory*, Vol. 47, No. 2, Feb. 2001, pp. 657-670. DOI: 10.1109/18.910580.

[12] F. Lehmann and G. M. Maggio, "Analysis of the Iterative Decoding of LDPC and Product Codes Using the Gaussian Approximation", *IEEE Trans. on Inf. Theory*, Vol. 49, No. 11, Nov. 2003, pp. 2993-3000. DOI: 10.1109/TIT.2003.819335.

[13] T. J. Richardson, A. Shokrollahi, and R. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes", *IEEE Trans. on Inf. Theory*, Vol. 47, No. 2, Feb. 2001, pp. 619-637. DOI: 10.1109/18.910578.

[14] M.-Y. Xu, X.-Y. Wang, S. Han, and S. Yu, "Designing low rate LDPC codes with iterative threshold close to the channel capacity", *Proc. of the 4th Int. Conf. on Electronics, Communications and Networks*, Beijing, China, Dec. 12-15, 2015.

[15] X. Chen and F. C. M. Lau, "Optimisation of Low-Density Parity-Check codes with Deterministic Unequal Error Protection Properties", *IET Communications*, Vol. 5, No. 11, 2011, pp. 1560-1565. DOI: 10.1049/iet-com.2010.0677.

[16] F. Babich, M. Noschese, and F. Vatta, "Analysis and Design of Rate Compatible LDPC Codes", *Proc. of the 27th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC'16*, Valencia, Spain, September 4-8, 2016, pp. 1-6. DOI: 10.1109/PIMRC.2016.7794684.

[17] F. Babich, M. Noschese, A. Soranzo, and F. Vatta, "Low Complexity Rate Compatible Puncturing Patterns Design for LDPC Codes", *Proc. of the 2017 Int. Conf. on Software, Telecommunications and Computer Networks, SoftCOM'17*, Split, Croatia, Sept. 21-23, 2017, pp. 1-5. DOI: 10.23919/SOFTCOM.2017.8115558.

[18] G. Hosoya, K. Osada, and M. Goto, "Rate-compatible punctured LDPC codes with two subgraphs", *Far East Journal of Electronics and Communications*, vol. 10, no. 2, 2013, pp. 83-104.

[19] M. R. Yazdani and A. H. Banihashemi, "On construction of rate-compatible low-density parity-check codes", *IEEE Communication Letters*, Vol. 8, No. 3, March 2004, pp. 159-161. DOI: 10.1109/LCOMM.2004.825728.

[20] N. Jacobsen and R. Soni, "Design of rate-compatible irregular LDPC codes based on edge growth and parity splitting", *Proc. of the 2007 IEEE 66th Vehicular Technology Conference*, Baltimore, MD, U.S.A., Sept. 30 - Oct. 3, 2007. DOI: 10.1109/VETECF.2007.228.

[21] T. Tian and C. R. Jones, "Construction of rate-compatible LDPC codes utilizing information shortening and parity puncturing", *EURASIP Journal on Wireless Communications and Networking*, December 2005, pp. 789-795. DOI: 10.1155/WCN.2005.789.

[22] J. Ha, J. Kim, D. Klinc, and S. W. McLaughlin, "Rate-compatible punctured Low-Density Parity-Check codes with short block lengths", *IEEE Trans. on Information Theory*, Vol. 52, No. 2, February 2006, pp. 728-738. DOI: 10.1109/TIT.2005.862118

[23] F. Babich, A. Soranzo, and F. Vatta, "Useful mathematical tools for capacity approaching codes design", *IEEE Communications Letters*, Vol. 21, No. 9, Sept. 2017, pp. 1949-1952. DOI: 10.1109/LCOMM.2017.2714684.

[24] F. Vatta, A. Soranzo and F. Babich, "Low-complexity bound on irregular LDPC belief-propagation decoding thresholds using a Gaussian approximation", *Electronics Letters*, Vol. 54, No. 17, August 23rd, 2018, pp. 1038-1040. DOI: 10.1049/el.2018.0478.

[25] J. Hagenauer, "Rate-compatible punctured convolutional codes (RCPC codes) and their applications", *IEEE Trans. on Communications*, Vol. 36, No. 4, April 1988, pp. 389-400. DOI: 10.1109/26.2763.

[26] F. Babich and F. Vatta,"Turbo codes construction for robust hybrid multitransmission schemes", *Journal of Communication Software and Systems (JCOMSS)*, Vol. 7, No. 4, December 2011, pp. 128-135. DOI: 10.24138/jcomss.v7i4.174.

[27] F. Babich, M. D'Orlando, and F. Vatta, "Distortion estimation algorithms for real-time video streaming: an application scenario", *Proc. of the 2011*

Int. Conf. on Software, Telecommunications and Computer Networks, SoftCOM'11, Split, Croatia, Sept. 15-17, 2011, pp. 291-295.

[28] A. Boscolo, F. Vatta, F. Armani, E. Viviani, and D. Salvalaggio,"Physical AWGN channel emulator for Bit Error Rate test of digital baseband communication", *Applied Mechanics and Materials*, Vols. 241-244, 2013, pp. 2491-2495. DOI: 10.4028/www.scientific.net/AMM.241-244.2491.

[29] M. Rigo, C. Travan, F. Vatta, and F. Babich, "Implementation of a burst error and burst erasure channel emulator using an FPGA architecture", *Proc. of the 2014 Int. Conf. on Software, Telecommunications and Computer Networks, SOFTCOM'14*, Split, Croatia, September 17-19, 2014, pp. 414-418. DOI: 10.1109/SOFTCOM.2014.7039095.

[30] K. S. Andrews, D. Divsalar, S. Dolinar, J. Hamkins, C. R. Jones, and F. Pollara, "The Development of Turbo and LDPC Codes for Deep-Space Applications", *Proceedings of the IEEE*, Vol. 95, No. 11, Nov. 2007, pp. 2142-2156. DOI: 10.1109/JPROC.2007.905132.

[31] S. Dolinar, D. Divsalar, and F. Pollara, "Turbo Code performance as a Function of Code Block Size", *Proc. of the 1998 IEEE International Symposium on Information Theory*, Cambridge, MA, U.S.A., Aug. 16-21, 1998. DOI: 10.1109/ISIT.1998.708612.

[32] J. Hamkins, "Performance of Low-Density Parity-Check Coded Modulation", *Proc. of the 2010 IEEE Aerospace Conference*, Big Sky, MT, U.S.A., March 6-13, 2010. DOI: 10.1109/AERO.2010.5446927.

[33] F. Babich, "A sphere-packing exponent approximation", *Proc. of the 2013 IEEE 77th Vehicular Technology Conference (VTC Spring)*, Dresden, Germany, 2-5 June 2013. DOI: 10.1109/VTCSpring.2013.6691845.

[34] S. Benedetto, E. Biglieri, *Principles of Digital Transmission with Wireless Applications*, Ed. Kluwer Academic Plenum Publishers, New York, Boston, 1999. DOI: 10.1007/b117711.

[35] V. Degardin, M. Lienard, P. Degauque, A. Zeddam, and F. Gauthier, "Impulsive noise on indoor power lines: characterization and mitigation of its effect on PLC systems", *Proc. of the 2003 IEEE International Symposium on Electromagnetic Compatibility, EMC '03*, Istanbul, Turkey, May 11-16, 2003. DOI: 10.1109/ICSMC2.2003.1428221.

[36] P. Guedes Esperante, C. Akamine, and G. Bedicks, "Comparison of Terrestrial DTV Systems: ISDB-TB and DVB-T2 in 6 MHz", *IEEE Latin America Transactions*, Vol. 14, No. 1, Jan. 2016, pp. 45-56. DOI: 10.1109/TLA.2016.7430060.

[37] I. Landa, M. M. Vélez, A. Arrinda, R. Torre, and M. Fernández, "Impulsive noise characterization and its effect on digital audio quality", *Proc. of the 2015 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting, BMSB'15*, Ghent, Belgium, June 17-19, 2015. DOI: 10.1109/BMSB.2015.7177238.

[38] J. Krejci and T. Zeman, "Impulse noise influencing xDSL technologies", *Proc. of the 2012 15th International Conference MECHATRONIKA*, Prague, Czech Republic, Dec 5-7, 2012.

[39] X. Kuai, H. Sun, and S. Zhou, "Impulsive Noise Mitigation in Underwater Acoustic OFDM Systems", *IEEE Transactions on Vehicular Technology*, Vol. 65, No. 10, Oct. 2016, pp. 8190-8202. DOI: 10.1109/TVT.2016.2516539.

[40] S.-Y. Huang, C.-S. Chang, and T.-H. Tan, "Markov Model Parameters Optimization for Asynchronous Impulsive Noise over Power Line Communication Network", *Proc. of the 2006 IEEE International Conference on Systems, Man and Cybernetics, SMC '06*, Taipei, Taiwan, Oct 8-11, 2006. DOI: 10.1109/ ICSMC.2006.384941.

[41] G. Laguna-Sanchez and M. Lopez-Guerrero, "On the Use of Alpha-Stable Distributions in Noise Modeling for PLC", *IEEE Transactions on Power Delivery*, Vol. 30, No. 4, Aug. 2015, pp. 1863-1870. DOI: 10.1109/TPWRD.2015.2390134.

[42] D. Middleton, "Procedures for determining the parameters of the first-order canonical models of class A and class B electromagnetic interference", *IEEE Transactions on Electromagnetic Compatibility*, Vol. EMC-21, No. 3, Aug. 1979, pp. 190-208. DOI: 10.1109/TEMC.1979.303731.

[43] D. Middleton, "Statistical-physical models of electro-magnetic interference", *IEEE Transactions on Electromagnetic Compatibility*, Vol. EMC-19, No. 3, Aug. 1977, pp. 106-127. DOI: 10.1109/TEMC.1977.303527.

[44] G. Ndo, F. Labeau, and M. Kassouf, "A Markov-Middleton Model for Bursty Impulsive Noise: Modeling and Receiver Design", *IEEE Transactions on Power Delivery*, Vol. 28, No. 4, Oct. 2013, pp. 2317-2325. DOI: 10.1109/TPWRD.2013.2273942.

[45] H. Nakagawa, D. Umehara, S. Denno, and Y. Morihiro, "A decoding for low density parity check codes over impulsive noise channels", *Proc. of the International Symposium on Power Line Communications and its Applications, 2005*, Vancouver, BC, Canada, April 6-8, 2005. DOI: 10.1109/ISPLC.2005.1430471.

[46] N. Andreadou and F.-N. Pavlidou, "Mitigation of Impulsive Noise Effect on the PLC Channel With QC-LDPC Codes as the Outer Coding Scheme", *IEEE Transactions on Power Delivery*, Vol. 25, No. 3, July 2010, pp. 1440-1449. DOI: 10.1109/TPWRD.2010.2041370.

[47] Z. Mei, M. Johnston, S. Le Goff, and L. Chen, "Finite Length Analysis of Low-Density Parity-Check Codes on Impulsive Noise Channels", *IEEE Access*, Vol. 4, 2016, pp. 9635-9642. DOI: 10.1109/ACCESS.2017.2649571.

Fulvio Babich received the doctoral degree, (Laurea), cum laude, in Electrical Engineering, at the University of Trieste, on July 1984. After graduation he was with Telettra at the Research and Development Laboratories, where he was engaged in optical fiber communications. Then he joined Zeltron, where he was a communication system engineer, responsible of the activities within the ESPRIT program. In 1992 he joined the Department of Electrical Engineering (DEEI) of the University of Trieste, where he is Professor of Signals and Systems and Wireless Networks. Fulvio Babich is vice director of Department of Engineering and Architecture. He is the coordinator of the Ph.D. in Industrial and Information Engineering of the University of Trieste. His current research interests are in the field of wireless networks and personal communications. He is involved in channel modeling, multiple access techniques, channel encoding, error control techniques and cross-layer design. Fulvio Babich serves as reviewer for many international journals, has served as co-chair for the Communication Theory Symposium, ICC 2005, Seul, ICC 2014, Sydney, and ICC 2017, Paris, for the Wireless Communication Symposium, ICC 2011, Kyoto, and for the Wireless Communication Symposium, WCSP 2012, Huangshan, China. He has served as General chair of the 13th IEEE IFIP Annual Mediterranean Ad Hoc Networking Workshop, Med-Hoc-Net 2014 June 2-4, 2014 Piran, Slovenia. Fulvio Babich is Senior Member of IEEE.

Matteo Noschese received his B.Sc. And M.Sc. in Telecommunication Engineering from the University of Trieste, in 2009 and 2015 respectively. From 2010 to 2015 he has worked as an embedded systems developer. He is currently working toward his Ph.D. in Telecommunications at the University of Trieste. His research interests include positioning, signal processing, cellular networks, LDPC channel codes, and network protocols.

Alessandro Soranzo received the M.S. Degree in Mathematics from the University of Trieste, Trieste, Italy, in 1991. He joined the Department of Mathematics and Geosciences (DMG) of the University of Trieste in 1994, where he is Assistant Professor of Mathematical Analysis. He is an author of about 20 papers, published on international journals and conference proceedings, in the fields of convex tomography, topology of natural numbers, fluidodynamics, and the approximation of special functions of statistics. His current research interests are in the area of the mathematical aspects of coding theory, mostly concerning capacity achieving coding schemes.

Francesca Vatta received the M.S. Degree in Electronic Engineering and the Ph.D. Degree in Telecommunications from the University of Trieste, Trieste, Italy, in 1992 and 1998, respectively. She joined the Department of Engineering and Architecture (DIA) of the University of Trieste in 1999, where she is Assistant Professor of Information Theory and Error-Control Coding. Starting in 2002, she spent several months as visiting scholar at the University of Notre Dame, Notre Dame, IN, U.S.A., cooperating with the Coding Theory Research Group under the guidance of Prof. D. J. Costello, Jr. Starting in 2005, she spent several months as visiting scholar at the University of Ulm, Germany, cooperating with the Telecommunications and Applied Information Theory Research Group under the guidance of Prof. M. Bossert. She is an author of more than 80 papers published on international journals and conference proceedings. Her current research interests are in the area of channel coding concerning, in particular, the analysis and design of capacity achieving coding schemes.