International Symposium on Information Theory and its Applications, ISITA2004
Parma, Italy, October 10–13, 2004

Serial Concatenation of LDPC Codes and Differentially Encoded Modulations

Michele Franceschini, Gianluigi Ferrari, Riccardo Raheli and Aldo Curtoni

Università di Parma
Dipartimento di Ingegneria dell'Informazione
Parco Area delle Scienze 181A
I-43100, Parma, Italy
E-mail: mfrance@tlc.unipr.it, gianluigi.ferrari@unipr.it, raheli@unipr.it

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, or *LDPC coded modulations*. With a pragmatic approach, we show how to design LDPC codes tailored for simple and robust modulation formats, like *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 the *degree distributions* of the LDPC codes, obtaining significant insights into the impact of such distributions on the performance of the proposed concatenated schemes. The EXIT chart-based optimization is confirmed by numerical simulations, considering Differential M-ary Phase Shift Keying (DMPSK) at the transmitter side, and iterative demodulation/decoding at the receiver side. The obtained optimized codes show poor performance if not concatenated with the inner DE. The analysis of the optimized codes shows that the decoding complexity of these codes is lower, with respect to that of standard LDPC codes, i.e., optimized for the additive white Gaussian noise (AWGN) channel.

1. Introduction

New channel coding techniques, like Turbo Codes (TC) [1] and Low Density Parity Check (LDPC) codes (originally invented in [2], and recently rediscovered [3]) are finding increasing applications in practical communication systems. In particular, TC have been thoroughly investigated, expecially with regard to their internal structure, consisting of well known convolutional encoders [4] and interleavers. Based on their simple internal structure, the parametric space of TC is discrete, in the sense that TC design parameters are related to the component convolutional codes (and the interleaver). This prevents the use of simple code optimization algorithms. LDPC codes, on the other hand, offer a practical way of optimization over a continuous space of design parameters, i.e., the *degree distributions*, which correspond to polynomials with positive real coefficients. In [5], it is shown how to partition the set of all LDPC codes in equivalence classes on the basis of their degree distributions. In [6], the authors optimize LDPC codes for transmissions over a Multi-Input Multi-Output (MIMO) channel through a MIMO modulator, in correspondence of which there is, at the receiver side, a Soft-Input Soft-Output (SISO) demodulator.

The goal of this paper is to show that there exist good LDPC codes to be used with modulations robust to phase uncertainties, typically present at the detector input in bandpass systems, and how to design such codes. As an example, in a coherent system, the detector has to deal with phase ambiguities due to the symmetry of the signal constellation. Indeed, experience shows that the use of a code designed for communication over the Binary Input Additive White Gaussian Noise Channel (BI-AWGN), i.e., the most investigated channel for LDPC codes, yields poor performance in the presence of an inner coded modulator [7]. In this paper, we concentrate on conventional Differential Encoding (DE) for M-ary Phase Shift Keying (MPSK) because of its well known robustness against most common channel impairments and its wide range of applications.

The proposed optimization technique is based on the approach presented in [6] and on the use of EXtrinsic Information Transfer (EXIT) charts, originally introduced in [8] for the convergence analysis of TC. This approach can take into account channel impairments, which are usually neglected in other simpler and approximated analysis methods. The obtained "ad-hoc" LDPC codes optimized for DE perform in the near-capacity region, confirming the validity of our design methodology.

The paper outline is as follows. In Section 2, we briefly introduce the necessary background. In Section 3, the considered transmission system is presented and discussed. In Section 4, we introduce the EXIT chart-based analysis of the convergence behavior of the decoding process. In this section, we also make some remarks about the necessity of ad-hoc optimization of LDPC codes and we outline how to use this technique to design good LDPC codes to be used with a generic, possibly coded, modulation format. In Section 5, we investigate, on the basis of numerical simulations, the behavior of serially concatenated schemes with outer LDPC codes and inner DE-PSK. Section 6 concludes the paper.

2. Background

In this section, we first introduce LDPC codes and a classification method. We then describe an approximate analysis technique for iterative decoders, based on EXIT charts, which leads to great simplifications with respect to other analysis methods such as those based on density evolution [9] or Monte Carlo simulations.

2.1. LDPC Codes and Degree Distributions

In linear block codes, a vector of bits $\boldsymbol{y}$ is a codeword iff $H\boldsymbol{y} = 0$, in which H is a binary matrix known as *parity check matrix*. In [2], LDPC codes are introduced as linear block codes whose parity check matrix H is sparse. In [2], it is also shown how to define a graph, in a one-to-one correspondence with H, consisting of two kinds of nodes (also known as bipartite graph or Tanner graph). Each node of the first kind, called *check node*, is associated with a row of H; each node of the second kind, called *variable node*, is associated with a column of H. A *variable node* is connected to a *check node* if there is a "1" at the intersection of the respective row and column of H. This graphical interpretation of a linear block code was used in [2] as the basis to obtain an asymptotically (in the codeword length) optimal decoding algorithm, in which each node sends and receives real-valued messages through the graph branches. This algorithm has recently been described as a particular instance of a broader class of graph-based algorithms also known as *sum-product algorithms* [10].

Following the notation in [5], a node has degree d if it has d branches departing from it. The *degree distributions* of a code, indicated as $\lambda(x)$ and $\rho(x)$, are polynomials defined as $\rho(x) \triangleq \sum_j \rho_j x^{j-1}$ and $\lambda(x) \triangleq \sum_i \lambda_i x^{i-1}$ respectively, where ρ_j is the fraction of branches in the graph connected to degree-j *check* nodes and λ_i is the fraction of branches in the graph connected to degree-i *variable* nodes. The polynomial $\rho(x)$ is also known as the *check node degree distribution* and $\lambda(x)$ is also known as the *variable node degree distribution.* These polynomials can be used to represent ensembles of codes, whose behavior becomes statistically equivalent, for increasing codeword length [5, 9]. Hence, optimization can be performed over these continuous-valued polynomials, assuming that the codeword length is sufficiently large.

Since $\{\rho_j\}$ and $\{\lambda_i\}$ correspond to fractions of the number of branches, they must satisfy the following conditions:

$$\begin{array}{rcccl} 0 & \leq & \rho_j & \leq & 1 \qquad j \geq 1 \\ 0 & \leq & \lambda_i & \leq & 1 \qquad i \geq 1 \\ \multicolumn{3}{r}{\sum_{j=1}^{\infty} \rho_j} & = & 1 \\ \multicolumn{3}{r}{\sum_{i=1}^{\infty} \lambda_i} & = & 1\,. \end{array} \tag{1}$$

A coefficient ρ_j of a check node degree distribution can be equivalently interpreted as the probability of finding a degree-j check node when picking at random one branch of the graph entering into the check nodes, and similarly for a coefficient λ_i.

In [5], the following linear constraint on the degree distributions, guaranteeing a given code rate R, is introduced:

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

The constraint (2) will be embedded in the optimization algorithm presented in Section 4.

2.2. SISO detectors and EXIT Charts

A SISO detector for a specific channel is a block which computes, based on the a priori probabilities of the transmitted symbols, their a posteriori probabilities, given some constraints on the received signal (or sequence) due to the channel or the transmitter structure. A common example of a SISO detector is based on the use of the forward-backward (FB) algorithm [4]. Without lack of generality, we will consider transmission of binary symbols —these symbols are then modulated (with high-order modulations) before being transmitted over the channel. In the following, we will also refer to the quantities at the input and output of a SISO detector as *reliabilities.* If a SISO module implements a maximum a posteriori (MAP) symbol strategy, the generated reliability values correspond to the a posteriori probabilities (APPs). More generally, the generated values approximate the APPs.

Since the observations at the output of the SISO block are representative of the transmitted binary sequence, it is possible to compute the Mutual Information (MI) between each transmitted bit and its a posteriori reliability at the output of the SISO block. A possible, and very generic, method for computing the MI between the output of a SISO block and the transmitted binary sequence can be based on Monte Carlo

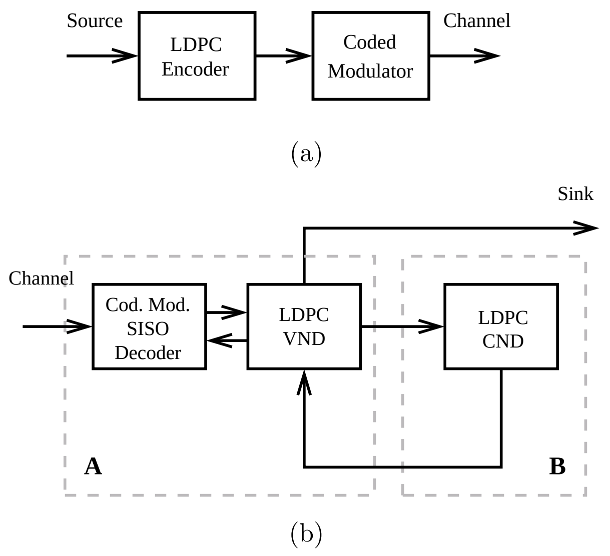

Figure 1: *System model: (a) transmitter side and (b) receiver side.*

simulations. An EXIT chart for a SISO block **S** is a function $I_{\mathbf{S}}(I)$ which plots the average relationship between the MI of the a priori reliabilities at the input of the block – indicated by I– and the MI of the a posteriori reliabilities at the output of the block – denoted by $I_{\mathbf{S}}$. Note that the MI values (both at the input and at the output) are computed with respect to the original transmitted sequence.

3. Communication System Model

In Figure 1(a), the model of the transmitter is shown: it consists of a simple concatenation of an outer LDPC encoder with an inner *coded modulator* which is directly connected to the channel. The goal of the inner block is to make the communication system robust against possible channel impairments.

In Figure 1(b), the receiver structure is depicted. It comprises the following blocks:

- A SISO block relative to the coded modulator and the channel, which computes a posteriori reliabilities for the binary symbols at the input of the coded modulator based on the channel observations and a priori reliabilities on the symbols, generated by the block labeled "LDPC VND" and described below.

- The LDPC Variable Node Detector (VND), relative to the variable nodes in the code bipartite graph, computes the reliabilities of each binary symbol. These reliabilities are sent to the "LDPC CND" block, described below.

- The LDPC Check Node Detector (CND), relative to the check nodes in the code bipartite graph, computes reliabilities of each binary symbol based on the a priori reliabilities received from the LDPC VND block and based on the relevant code constraints.

Note that, in all the computations involved in the decoding process, only the so-called *extrinsic information*, i.e. the a posteriori probability of a symbol computed assuming complete a priori uncertainty about that symbol, is exchanged [1]

The iterative decoding algorithm can be described as follows. As initialization step, the a priori reliabilities at the input of the coded modulator SISO block correspond to complete uncertainty (the reliability values from the LDPC VND to the SISO block). At the first step, the coded modulator SISO block generates and passes extrinsic information to the LDPC VND block, which, in turn, computes the extrinsic information to be sent to the LDPC CND block. The LDPC CND block computes extrinsic information values to be passed to the LDPC VND block, which thus computes extrinsic information to be passed to the coded modulation SISO block. This algorithm is iterated from the first step until some stopping condition is satisfied (e.g., an LDPC codeword is decoded). Finally, a posteriori reliabilities, i.e., not simply extrinsic information, are computed by the LDPC VND block and delivered to the destination.

4. LDPC Code Optimization Algorithm

In Figure 1(b), the receiver is divided in two superblocks labeled **A** and **B**, respectively: block **A** includes the SISO decoder and LDPC VND block; block **B** includes the LDPC CND block. These superblocks can be viewed as SISO blocks exchanging extrinsic information. For each superblock it is possible to plot the corresponding EXIT chart [8]: $I_{\mathbf{A}}(I)$ for block **A** and $I_{\mathbf{B}}(I)$ for block **B**. Plotting $I_{\mathbf{A}}(I)$ and $I_{\mathbf{B}}^{-1}(I)$ in the same diagram, the decoding process can be visualized as a recursive update of the MI in the two EXIT curves. If the MI eventually goes to 1, one can expect that the bit error rate (BER) will eventually be zero. In order for the decoding process to converge, the $I_{\mathbf{A}}(I)$ curve should be always above the $I_{\mathbf{B}}^{-1}(I)$ curve. In other words, in this case there is an open path between $I_{\mathbf{A}}(I)$ and $I_{\mathbf{B}}^{-1}(I)$ which leads to the point (1,1), and we refer to this situation by saying that the *tunnel is open*. If these curves touch, we say that the *tunnel is at pinch-off*. Otherwise, the *tunnel is closed*. If the LDPC code satisfies some conditions (i.e., absence of short cycles in the code graph), one can show that the EXIT charts are functions only of the degree distributions, so that it is possible to optimize the overall system by finding a good set of degree distributions.

In [6], a heuristic code optimization technique is proposed, based on fitting the two EXIT chart curves $I_{\mathbf{A}}(I)$ and $I_{\mathbf{B}}^{-1}(I)$ as functions of the degree distributions $(\lambda(x), \rho(x))$. This technique seems appealing for the following reason. If, for a given signal-to-noise ratio (SNR), the tunnel between $I_{\mathbf{A}}(I)$ and $I_{\mathbf{B}}^{-1}(I)$ is at pinch-off, i.e., for a particular value I^{po} the EXIT curves touch but are separated anywhere else, then a small[1] variation of the degree distributions, maintaining the same code rate, might open the tunnel at I^{po}. A possible observation about the fitting optimization approach can be that it leads to a class of algorithms that try to generate EXIT curves with the smallest possible distance.[2] However, *our actual goal is to keep the tunnel open.* It is reasonable to suppose that increasing the SNR raises the EXIT curve of block **A**, and this is confirmed by experience. Hence, if the distance of the two EXIT curves, when the tunnel is closed, is small, an SNR increment *should* open the tunnel. However, this could not be true if the tunnel were closed around the point $(I_{\mathbf{A}}, I_{\mathbf{B}}) = (1, 1)$: in this region, in fact, an SNR increment would lead to smaller increments of $I_{\mathbf{A}}$. This technique is then effective when very accurate fits are possible, i.e., when the parameter space has a large dimensionality. In practice, the maximum degrees of check and variable nodes need to be limited in order to keep the girth of the graph sufficiently large.

We choose to optimize the degree distributions by means of a simple algorithm performing a *random walk* in the parametric space. Other techniques, such as linear programming over the check node degree distribution [11], could be applied with better computational performance. Nevertheless the chosen technique is simple and efficient enough to identify a set of "good" degree distributions. The random walk algorithm can be described as follows.

We define a functional that has to be optimized. This functional has to be representative of the tunnel opening: the more the channel is closed, the lower the functional must be. A possible choice is the following:

$$f(\lambda, \rho) \triangleq \min_{I} \left\{ I_{\mathbf{A}}(I) - I_{\mathbf{B}}^{-1}(I) \right\} \tag{3}$$

where we have explicitly indicated the dependence of the functional on the degree distributions. This choice guarantees that the optimization algorithm will first try to open the tunnel, if closed. In fact, it first finds the point $I^{\min}$ in correspondence to which the tunnel is most closed, and then returns the difference between $I_{\mathbf{A}}(I^{\min})$ and $I_{\mathbf{B}}^{-1}(I^{\min})$. This difference is negative if and only if the tunnel is closed. Indeed, this functional can not be larger than zero because in $(I_{\mathbf{A}}, I_{\mathbf{B}}) = (1, 1)$ the EXIT charts necessarily touch.

Every time a step in the random walk in the design parameter space leads to a point in correspondence to which the tunnel is not closed, i.e., $f(\lambda, \rho) = 0$, the channel is worsened (i.e., the SNR is diminished) until the tunnel closes.

The design parameter space is represented by the two sets of polynomial coefficients $\{\rho_j\}$ and $\{\lambda_i\}$. According to relations (1) and (2), three parameters are linearly dependent from the others. Hence, one has first to choose (i) a parameter from $\{\lambda_i\}$, (ii) a parameter from $\{\rho_j\}$ and (iii) an additional parameter, either from $\{\lambda_i\}$ or $\{\rho_j\}$. Then, these three parameters have to be expressed as functions of the remaining *free* parameters. The numbers of elements of the sets $\{\lambda_i\}$ and $\{\rho_j\}$ can be any, provided that these sets are not empty and contain at least four elements.

The random walk can be described as follows. Staring from a tuple which satisfies the inequalities in (1), a new tuple is obtained by adding to the previous tuple a Gaussian random increment vector with zero mean and standard deviation s. If this tuple does not satisfy all the inequalities in (1), a new tuple is generated starting from the previous tuple. This procedure is repeated until a tuple which satisfies all the inequalities in (1) is found. From this new tuple, one can compute $\lambda(x)$ and $\rho(x)$ and then the new trial value of $f(\lambda, \rho)$: if this value is bigger than the previous one, we substitute the previous tuple with the new one. 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 tunnel for a desired SNR, or a maximum number of iterations is reached. The steps of the proposed optimization algorithm are summarized in Table 1.

As a possible improvement, one can diminish the step value s (standard deviation of the Gaussian increment vector) after a given number of unsuccessful trials. It is also possible to repeat the last successful increment vector as a first trial, and, if unsuccessful, use a random Gaussian increment vector. This technique offers the advantage of being effective also for small sets of possible node degrees. Refined versions of this optimization technique are currently under investigation.

5. Numerical Results

We applied the EXIT chart-based optimization al-

[1] We remark that it is possible to consider a small variation owing to the continuous nature of LDPC codes families in terms of degree distributions.

[2] This "distance" could be any reasonable distance: mean square distance, maximum distance, etc.

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 (λ',ρ') compatible with code rate at (small) random distance from (λ,ρ).
	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 (λ,ρ) and final SNR.

Table 1: *Optimization Algorithm.*

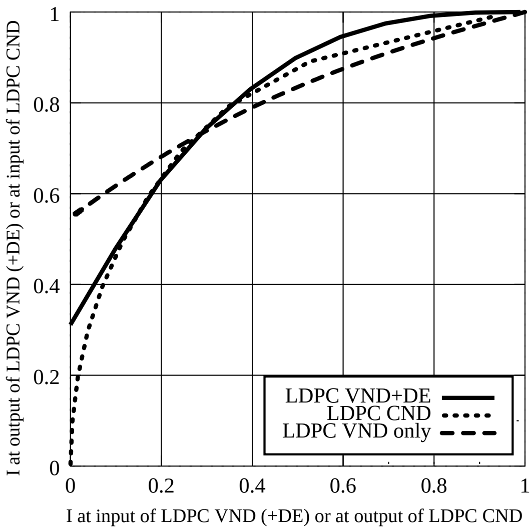

Figure 2: *EXIT chart-based analysis of an optimized LDPC code concatenated with a QPSK with DE ($E_b/N_0 = 0.8$dB: tunnel at* pinch-off*) and QPSK without DE (tunnel is* closed*).*

gorithm to LDPC codes concatenated with Differential encoded M-ary PSK (DMPSK) and constellations of order 4 and 8. The channel is AWGN, but, due to the nature of DMPSK, the resulting codes are insensitive to discrete phase ambiguities which may arise from the use of a phase synchronization algorithm. The optimization has been performed allowing only a small set of degrees in the degree distributions to be nonzero. In particular, the set of allowed variable node degrees is $d_v \in \{2,\ldots,12\}$, for all considered code rates. The optimization algorithm allows to perform 100 iterations (a path of 100 successful points in the parametric space). Figure 2 shows the EXIT chart of a D4PSK (Quaternary DPSK, DQPSK) system, serially concatenated with the relative optimized LDPC code: the solid curve corresponds to block **A** (cfr. Figure 1) and the dotted curve to block **B**. The SNR E_b/N_0, where E_b is the bit energy and N_0 is the noise power spectral density, is set to 0.8 dB. It is immediate to recognize that the tunnel is at pinch-off. The dashed curve in Figure 2 is the EXIT chart of the LDPC VND only (i.e., without DE): the tunnel is “heavily” closed, predicting that the DE system should perform significantly better than the single LDPC code without DE. Note that the convergence SNR threshold for DQPSK predicted by the results in Figure 2 is around 0.9 dB.

In Figure 3, the performance of various DMPSK systems using ad-hoc optimized LDPC codes is shown. The LDPC binary codeword length is 12000 and the code rates (indicated in Figure 3) are chosen so that the overall spectral efficiencies are 1 bps/Hz, 1.5 bps/Hz, and 2.25 bps/Hz, respectively. In Figure 3, the performance of the same codes is plotted both with DE (considered in the optimization process) and without DE. It is possible to see that, while the performance without DE is very poor (thus meaning that these codes are not powerful codes in a usual sense), the systems using DE perform at about 1 dB from their respective AWGN channel capacities, still enabling high robustness against channel impairments, such as phase noise, discrete phase uncertainty and small carrier frequency offsets.

Obviously, in order to obtain good performance under possible impairments it is mandatory to design a SISO detector for DMPSK which takes into account the presence of such impairments. We remark that the use of conventional LDPC codes designed for the AWGN channel, without an inner coded modulator, in this concatenated scheme would entail a performance loss, with respect to the AWGN channel capacity, between 2 dB and 3 dB.

An interesting property of the variable nodes degree distributions optimized for the presence of DE is that they are characterized by the presence of a very high fraction ($0.6 \div 0.7$) of degree-2 variable nodes. Bounds for the degree-2 variable nodes fraction for irregular LDPC codes directly “connected” to the channel, i.e., without an inner coded modulator, were investigated in [5]. The degree distributions optimized for the presence of DE violate these bounds, leading to codes with poor performance if used without DE. A useful remark is that, LDPC codes with high fraction of degree-2 variable nodes are very efficiently decodable codes. In fact,

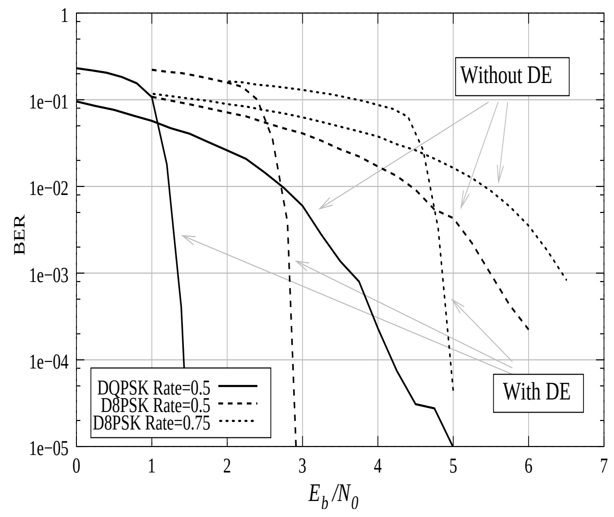

Figure 3: *BER, as a function of the SNR, for three systems using codes optimized for DE-PSK. The performance is shown with and without the DE.*

since a high fraction of the edges is connected to degree-2 variable nodes, these codes are characterized by a small number of edges in their Tanner graph. Since the decoding complexity is known to be proportional to the number of edges in the graph, these codes are therefore decodable with low complexity.

6. Conclusions

In this paper, we have considered communication systems which use, at the transmitter side, an LDPC encoder concatenated with a DMPSK modulator. We have made use of the fact that LDPC codes families can be characterized by continuous parameters in terms of degree distributions (of check and variable nodes). Using a particular sub-block decomposition of the receiver, a novel optimization algorithm, based on the use of EXIT charts, has been proposed. By means of a semi-random search in the space of the degree distributions, this algorithm generates good degree distributions which characterize LDPC codes minimizing the SNR convergence threshold of the concatenated LDPC-DMPSK scheme. The analysis of the optimized LDPC code ensembles in the presence of a DMPSK modulator has shown that their internal structure significantly differs from that of standard LDPC codes designed 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 decrease. For all the obtained code ensembles, the BER performance, evaluated by means of Monte Carlo simulations, turned out to be close to the channel capacity, indicating that the proposed optimization procedure is effective.

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, October 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,” *IEE Electronics Letters*, vol. 33, pp. 457–458, March 1997.

[4] 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, pp. 22–24, January 1997.

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

[6] 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, April 2004.

[7] H. Jin and T. Richardson, “Design of low-density parity-check codes for noncoherent mpsk communication,” in *Proc. IEEE Symp. Inform. Theory*, Lausanne, Switzerland, June 2002, p. 169.

[8] S. ten Brink, “Convergence of iterative decoding,” *IEE Electronics Letters*, vol. 35, pp. 1117–1119, 24th June 1999.

[9] T. Richardson and R. Urbanke, “The capacity of low density parity check codes under message passing decoding,” *IEEE Trans. Inform. Theory*, vol. 47, pp. 599–618, February 2001.

[10] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, “Factor graphs and the sum-product algorithm,” *IEEE Trans. Inform. Theory*, vol. 47, pp. 498–519, February 2001.

[11] S.-Y. Chung, T. Richardson, and R. Urbanke, “Analysis of sum-product decoding of low-density parity-check codes using Gaussian approximation,” *IEEE Trans. Inform. Theory*, vol. 47, pp. 657–670, February 2001.