Optimized Concatenated LDPC Codes for Joint Source-Channel Coding

Maria Fresia[†], Fernando Pérez-Cruz[†‡], H. Vincent Poor[†]
[†]Department of Electrical Engineering, Princeton University, Princeton, New Jersey 08544
[‡]Department of Signal Theory and Communications, University Carlos III, Leganés (Madrid) 28911.
Email: {mfresia, fp, poor}@princeton.edu

***Abstract*—In this paper a scheme for joint source-channel coding based on LDPC codes is investigated. Two concatenated independent LDPC codes are used in the transmitter: one for source and the other for channel coding, with joint belief propagation decoder. The asymptotic behavior is analyzed using EXtrinsic Information Transfer (EXIT) charts and this approximation is corroborated with illustrative experiments. The optimization of the degree distributions for our sparse code to maximize the information transmission rate is also considered.**

I. Introduction

The separation principle states that there is no-loss of optimality from disjoint design and decoding of the source and channel codes as the block length tends to infinity [1]. Thereby, these two problems have traditionally been addressed independently of each other. On one side, source coding relies on the Lempel-Ziv (LZ) algorithm [2] which compresses any stationary and ergodic source to its entropy as the number of symbols increases. On the other, channel coding builds on low-density parity-check (LDPC) codes [3] achieving channel capacity as the number of coded bits tends to infinity. However, for finite-length codes, the separation principle does not apply and the residual redundancy left by the source code should be employed by the channel decoder to reduce its error rate.

For finite length codes, the nature of LPDC codes and the LZ algorithm makes impractical the use of a joint decoder. Furthermore, for disjoint decoding, the LZ algorithm may be problematic: unless the block length is sufficiently long and the channel decoding error rate is extremely low [4]. Therefore, some systems, such as third-generation wireless data transmission, do not compress the redundant source prior to channel encoding and, consequently, there has been several proposals [4]-[9] that exploit the source statistics at the channel decoder. These schemes use the redundancy at the source (uncompressed or partially compressed) to improve the channel decoder performance.

We propose to add an additional LDPC-based source encoder between the uncompressed source and the LDPC channel encoder, which are decoded jointly. This approach has the advantage of (further) compressing the source prior to adding the redundancy bits and the joint decoder allows the channel decoder to exploit the redundancy left by the source encoder. This structure was mentioned as a possible solution in [10, Sec. 2.1], but was not investigated further in favor of the LOTUS codes. LOTUS codes use a single sparse graph for joint source-channel coding, in a similar fashion to [4]-[9], with the code optimized for the source statistics.

The double structure in the transmitter side is of particular interest for very low entropy sources. In this case in fact, the overall rate of the separated source and channel codes compresses the source, while the structures in [4]-[9] can only add redundancy to this already highly redundant source. Therefore, the separated structure in the transmitter side increases the flexibility of our communication system, allowing to compress and protect the source. For high entropy sources, we can trade-off the redundancy left in the source by the redundancy introduced by the channel encoder to achieve higher information transmission rates.

The rest of the paper is organized as follows. In Section II the structure of the proposed JSC decoder is described. We analyze its asymptotic behavior in Section III using the EXtrinsic Information Transfer (EXIT) chart approximation [11], while an optimization procedure for maximizing its transmission rate is described in Section IV. The experimental results are reported in Section V.

II. Joint Source and Channel Decoder

The transmitter first compresses the source with an LDPC-based code, as proposed in [12]. The compressed sequence $\mathbf{b} = \mathbf{H}_{sc}\mathbf{s}$, where $\mathbf{H}_{sc}$ is a sparse $\ell \times n$ parity-check matrix ($R_{sc} = \ell/n < 1$). Then it protects the encoded bits with another LPDC code ($R_{cc} = \ell/m < 1$) and it finally transmits the codeword $\mathbf{x} = \mathbf{G}_{cc}\mathbf{b}$ over the channel.

Two sparse bipartite graphs compose the decoder, as shown in Fig. 1, where each check node of the source code (left) is connected to a single variable node of the channel code (right). The joint decoder runs in parallel. First the variable nodes inform the check nodes about their log-likelihood ratio (LLR) and then the check nodes respond with their LLR constraints for each variable node. Let us consider the k^{th} iteration of the decoder. For the sake of clarity, we describe the two decoders with separated notation. $m_{v,c}^{sc,(k)}$ and $m_{v,c}^{cc,(k)}$ are, respectively, the message passed from the v^{th} variable node to the c^{th} check node of the source code ($\mathcal{C}_{sc}$) and channel code ($\mathcal{C}_{cc}$). $m_{c,v}^{sc,(k)}$

This research was supported in part by the National Science Foundation under Grant CNS-06-25637. Fernando Pérez-Cruz is supported by Marie Curie Fellowship 040883-AI-COM

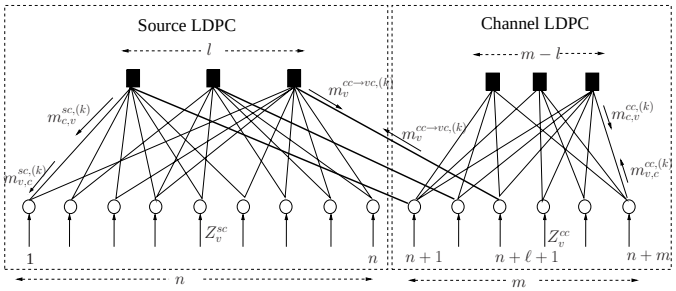

Fig. 1. Joint decoder scheme

and $m_{c,v}^{cc,(k)}$ are, respectively, the message passed from the c^{th} check node to the v^{th} variable node of $\mathcal{C}_{sc}$ and $\mathcal{C}_{cc}$. $m_v^{sc\to cc,(k)}$ is the message passed from the check node in $\mathcal{C}_{sc}$ connected to the v^{th} variable node in $\mathcal{C}_{cc}$, while $m_v^{cc\to sc,(k)}$ is the message passed from the v^{th} variable node in $\mathcal{C}_{cc}$ connected to the c^{th} check node in $\mathcal{C}_{sc}$. These last two messages are indexed only by v, because there is a single connection between every check node in $\mathcal{C}_{cs}$ with a variable node in $\mathcal{C}_{cc}$

Z_v^{sc} and Z_v^{cc} represent, respectively, the LLRs for the variable nodes for $v = 1, \ldots, n$ (i.e. the variable nodes of the source decoder) and for $v = n+1, \ldots, n+m$ (i.e. the variable nodes of the channel decoder). For independent binary sources transmitted over a BIAWGN channel, $Z_v^{sc} = \log(\frac{1-p_v}{p_v})$ (where $p_v = \mathbb{P}[s_v = 1]$), and $Z_v^{cc} = \frac{2r_v}{\sigma_n^2}$, where $r_v = (1-2x_v) + n_v$, and σ_n^2 is the channel noise variance.

The messages between variable nodes and check nodes follow the same procedure that standard belief propagation. First the variable nodes send their LLRs to the check nodes and the corresponding messages are given by

$$m_{v,c}^{sc,(k)} = Z_v^{sc} + \textstyle\sum_{c'\neq c} m_{c',v}^{sc,(k-1)}, \quad (1)$$

$$m_{v,c}^{cc,(k)} = Z_v^{cc} + m_v^{sc\to cc,(k-1)} + \textstyle\sum_{c'\neq c} m_{c',v}^{cc,(k-1)}, \quad (2)$$

$$m_v^{cc\to sc,(k)} = Z_v^{cc} + \textstyle\sum_{c'} m_{c',v}^{cc,(k-1)} \quad \text{and} \quad (3)$$

$$m_{v,c}^{cc,(k)} = Z_v^{cc} + \textstyle\sum_{c'\neq c} m_{c',v}^{cc,(k-1)}, \quad (4)$$

where (1) runs for $v = 1, \ldots, n$; (2) and (3) for $v = n+1, \ldots, \ell$; and (4) for $v = n+\ell+1, \ldots, n+m$. Notice that $m_{c',v}^{sc,(0)} = 0$, $m_{c',v}^{cc,(k-1)} = 0$ and $m_v^{sc\to cc,(0)} = 0$.

The messages between the check nodes and the variables nodes are given by

$$\tanh\left(\frac{m_{c,v}^{sc,(k)}}{2}\right) = \tanh\left(\frac{m_v^{cc\to sc,(k)}}{2}\right) \prod_{v'\neq v} \tanh\left(\frac{m_{v',c}^{sc,(k)}}{2}\right), \quad (5)$$

$$\tanh\left(\frac{m_v^{sc\to cc,(k)}}{2}\right) = \prod_{v'} \tanh\left(\frac{m_{v',c}^{sc,(k)}}{2}\right) \quad \text{and} \quad (6)$$

$$\tanh\left(\frac{m_{c,v}^{cc,(k)}}{2}\right) = \prod_{v'\neq v} \tanh\left(\frac{m_{v',c}^{cc,(k)}}{2}\right), \quad (7)$$

where (5) and (6) run for $c = 1, \ldots, \ell$, while (7) runs for $c = \ell+1, \ldots, m$. After K iterations of the decoding process, the v^{th} source bit is estimated by computing the LLR of the source bit s_v (i.e., $\hat{s_v} = 0$ if $LLR(s_v) \geq 0$, and $\hat{s_v} = 1$ otherwise, where $LLR(s_v) = Z_v^{sc} + \sum_c m_{c,v}^{sc,(K)}$).

III. Asymptotic Analysis: EXIT charts

The belief propagation algorithm allows analyzing finite-length codes, but is impractical for studying the asymptotic behavior of sparse codes. EXIT charts [11], an approximation of density evolution (DE) [13], [14], are a simple way to analyze this asymptotic behavior. Specifically the EXIT chart with Gaussian approximation assumes that the belief propagation messages are Gaussians having a particular symmetry condition which imposes that $\sigma^2 = 2\mu$.

Since the decoder is composed of two separated LDPC decoders that exchange information, it is not possible to combine the evolution of the two decoders in a single input-output function. Even if they run in parallel exchanging information, we need to describe the evolution of the source and channel decoders separately.

The following notation is used in the rest of the section. $\mathsf{x}_{i_{sc}}$ [$\mathsf{x}_{i_{cc}}$] denotes the mutual information between a message sent along an edge (v,c) with "left-degree" i_{sc} [i_{cc}] and the symbol corresponding to the bitnode v for the LDPC source [channel] decoder; and x_{sc} [x_{cc}] denotes the average of $\mathsf{x}_{i_{sc}}$ [$\mathsf{x}_{i_{cc}}$] over all edges (v,c). $\mathsf{y}_{j_{sc}}$ [$\mathsf{y}_{j_{cc}}$] denotes the mutual information between a message sent along an edge (c,v) with "right-degree" j_{sc} [j_{cc}] and the symbol corresponding to the bitnode v for the LDPC source [channel] decoder; and y_{sc} [y_{cc}] denotes the average of $\mathsf{y}_{i_{sc}}$ [$\mathsf{y}_{i_{cc}}$] over all edge (c,v).

We consider the class of EXIT functions that make use of Gaussian approximation of the BP messages, which considers the well-known fact that the family of Gaussian random variables is closed under addition (i.e. the sum of Gaussian random variables is also Gaussian, and its mean is the sum of the means of the addends). Imposing the symmetry condition and Gaussianity, the conditional distribution of each message $\mathcal{L}$ in the direction $v \to c$ is Gaussian $\sim \mathcal{N}(\mu, 2\mu)$, for some value $\mu \in \mathbb{R}_+$. Hence, letting V denote the corresponding bitnode variable, we have

$$I(V;\mathcal{L}) = 1 - \mathbb{E}\left[\log_2\left(1+e^{-\mathcal{L}}\right)\right] \triangleq J(\mu),$$

where $\mathcal{L} \sim \mathcal{N}(\mu, 2\mu)$. Notice that, by using the function $J(\cdot)$, the capacity of a BIAWGN channel with noise variance σ_n^2 can be expressed as $C = J(2/\sigma_n^2)$.

Let us consider the "two-channel" scenario induced by the proposed JSC scheme. Generally, an LDPC code is defined by $\lambda(x) = \sum_i \lambda_i x^{i-1}$ [$\Lambda(x) = \sum_i \Lambda_i x^i$], and $\rho(x) = \sum_j \rho_j x^{j-1}$ [$P(x) = \sum_j P_j x^j$] which represent the degree distribution of the variable nodes and the check nodes respectively in the edge [node] perspective.

For the source decoder, the message x_{sc} is given by

$$\mathsf{x}_{sc} = \sum_{i_{sc}} \lambda_{i_{sc}} J_{BSC}\left((i_{sc}-1)J^{-1}(\mathsf{y}_{sc}), p\right), \quad (8)$$

where $J_{BSC}(\cdot)$ is a manipulation of the function $J(\cdot)$ to take into account that the source is binary and i.i.d. with $p = \mathbb{P}[s_v = 1]$, i.e. the equivalent channel is a binary

symmetric channel (BSC) with crossover probability p (then capacity $1-H(p)$). In particular, the probability density function (pdf) of the LLR output from the equivalent channel is given by $p\delta(x+L)+(1-p)\delta(x-L)$. Therefore, the function J_{BSC} can be expressed by

$$J_{BSC}(\mu,p) = (1-p)I(V;\mathcal{L}^{(1-p)}) + pI(V;\mathcal{L}^{(p)}),$$

where $\mathcal{L}^{(1-p)} \sim \mathcal{N}(\mu+L, 2\mu)$, and $\mathcal{L}^{(p)} \sim \mathcal{N}(\mu-L, 2\mu)$.

The message y_{sc} is given by

$$\mathsf{y}_{sc} = 1 - \sum_{i_{cc},j_{sc}} \Lambda_{i_{cc}} \rho_{j_{sc}} J\big((j_{sc}-1)J^{-1}(1-\mathsf{x}_{sc}) + J^{-1}(1-\uparrow c_{i_{cc}})\big), \tag{9}$$

where $\uparrow c_{i_{cc}} = J(i_{cc}J^{-1}(\mathsf{y}_{cc})) + J^{-1}(C))$ is the message generated by a variable node of degree i_{cc} of the channel decoder. Notice that we average over all possible values of i_{cc} through $\Lambda_{i_{cc}}$.

For the channel decoder, the message x_{cc} is given by

$$\mathsf{x}_{cc}^{(k)} = R_{cc} \sum_{j_{sc},i_{cc}} P_{j_{sc}} \lambda_{i_{cc}} J\big((i_{cc}-1)J^{-1}(\mathsf{y}_{cc}) + J^{-1}(C) + J^{-1}(\downarrow c_{j_{sc}})\big) + (1-R_{cc}) \sum_{i_{cc}} \lambda_{i_{cc}} J\left((i_{cc}-1)J^{-1}(\mathsf{y}_{cc}) + J^{-1}(C)\right), \tag{10}$$

where $\downarrow c_{j_{sc}} = 1 - J\left(j_{sc}J^{-1}(1-\mathsf{x}_{sc})\right)$ is the message generated by a check node of degree j_{sc} of the source decoder. We average over all possible values of j_{sc} through $P_{j_{sc}}$. Notice that (10) is composed of two parts to take into account the fact that a fraction of R_{cc} variable nodes of the channel decoder are connected to the check nodes of the source decoder (i.e. they have the extra message $\downarrow c_{j_{sc}}$), while the remaining $1-R_{cc}$ are connected only to the transmission channel. Since the data are transmitted over a BIAWGN channel, then $C = J(2/\sigma_n^2)$, therefore $J^{-1}(C) = 2/\sigma_n^2$.

Finally, the message y_{cc} is given by

$$\mathsf{y}_{cc} = 1 - \sum_{j_{cc}} \rho_{j_{cc}} J\left((j_{cc}-1)J^{-1}(1-\mathsf{x}_{cc})\right). \tag{11}$$

After K iterations, in order to compute the bit error rate (BER), we need to obtain the conditional pdf of the LLRs output by the source bits (i.e. the variable nodes of the source code). Without taking into account the message generated by the equivalent channel, these LLRs are Gaussian, i.e. for a variable node with degree i_{sc}, $N(\mu_{i_{sc}}, 2\mu_{i_{sc}})$, where $\mu_{i_{sc}} = i_{sc}J^{-1}(\mathsf{y}_{sc})$. Since the equivalent channel is modeled as a BSC, the pdf of the overall message is a Gaussian mixture weighted by the value of p, i.e. $pN(\mu_{i_{sc}}-L, 2\mu_{i_{sc}}) + (1-p)N(\mu_{i_{sc}}+L, 2\mu_{i_{sc}})$. Averaging over all possible values we have that the bit error rate BER is equal to

$$P_e = \sum_{i_{sc}} \lambda_{i_{sc}} \left[pQ(x_{i_{sc}}^{p}) + (1-p)Q(x_{i_{sc}}^{1-p})\right], \tag{12}$$

where $Q(\cdot)$ is the Gaussian tail function and where $x_{i_{sc}}^{p} = \sqrt{\mu_{i_{sc}}/2} - L/\sqrt{2\mu_{i_{sc}}}$, and $x_{i_{sc}}^{1-p} = \sqrt{\mu_{i_{sc}}/2} + L/\sqrt{2\mu_{i_{sc}}}$.

As described in Section II, in the finite length simulations the two decoders run in parallel. On the contrary, in the infinite length case we adopt a conceptually easier schedule: for each iteration of the LDPC source decoder, a large number of iterations on the channel decoder are performed, in order to reach the fixed point equilibrium; the generated messages are incorporated as "additive messages" to the check nodes of the source LDPC decoder; all the check nodes of the channel LDPC code are activated. This provides a complete cycle of scheduling, which is repeated an arbitrarily large number of times. The reason for adopting this scheduling instead of the practical one is related to the fact that the EXIT charts can be seen as a multidimensional dynamical system with state variables. Since we are interested in studying the fixed points and the trajectories of this system, we need to reduce the problem to an input-output function and then reduce the number of variables.

IV. JSC Code optimization

In this section we present an optimization procedure for maximizing the transmission rate of the proposed JSC code. We suggest a suboptimal procedure that gives optimal codes when the source and channel rates tend, respectively, to the entropy of the source and the capacity of the channel. First, we compute the optimal channel code assuming the input bits are i.i.d. and equally likely (worse case). This is the standard LPDC optimization for channel coding [3], [15]. Given the optimized channel code, we compute optimal degree distributions for the variables and check nodes in the source code.

Substituting (9) into (8), we can express the input-output function of the source code as

$$\mathsf{x}_{sc}^{(k)} = F_{sc}(x_{sc}^{(k-1)}, p, f_{cc}(x_{sc}^{(k-1)}, C)), \tag{13}$$

where $f_{cc}(x_{sc}^{(k-1)}, C)$ is the input-output function related to the channel code and is derived by substituting (11) into (10).

In a density evolution analysis, the convergence is guaranteed if $F_{sc}(\mathsf{x}_{sc}, p, f_{sc}(\mathsf{x}_{sc}, \sigma^2)) > \mathsf{x}_{sc}$ for $\mathsf{x}_{sc} \in [0,1]$, which ensures convergence at the fixed point $\mathsf{x}_{sc} = 1$.

Given $(\lambda_{cc}(x), \rho_{cc}(x))$ (i.e. fixing the channel code) and $\rho_{sc}(x)$[1], Eq. (13) is linear with respect to the coefficients of $\lambda_{sc}(x)$, and thus the optimization problem can be written as

$$\max \textstyle\sum_{i_{sc}\geq 2} \frac{\lambda_{i_{sc}}}{i_{sc}} \tag{14}$$

$$\text{subject to} \quad \textstyle\sum_{i_{sc}} \lambda_{i_{sc}\geq 2} = 1,\ 0 \geq \lambda_{i_{sc}} \leq 1 \tag{15}$$

$$F_{sc}(\mathsf{x}_{sc}, p, f_{cc}(\mathsf{x}_{sc}, C)) > x_{sc} \tag{16}$$

$$\lambda_2 < \frac{1}{2\sqrt{p(1-p)}} \cdot \frac{1}{\sum_{j_{sc}} \rho_{j_{sc}}(j-1)}, \tag{17}$$

where (17) represents the stability condition [16].

The optimization procedure sketched above, is based on the optimization procedure proposed in [16]. In contrast to [16], we deal with two codes that iterate in parallel and then we add the input-output function of the channel code (i.e. $f_{cc}(x_{sc}^{(k-1)}, C)$) to the optimization.

[1]According to [16], we consider a concentrated right degree distribution of the form $\rho(x) = \rho x^{k-1} + (1-\rho)x^k$ for some $k \geq 2$ and $0 \leq \rho \leq 1$.

V. Experimental Results

In this section, we illustrate the advantages of using two concatenated LPDC codes for joint source-channel coding instead of one structure as proposed in [10]. We have performed three sets of experiments with regular LDPC codes with three ones per column and an overall coding rate of 2. In the first, we illustrate the advantage of using two concatenated LDPC codes, one for source coding and the other for channel coding, instead of a single sparse code. In the second experiment, we show why joint decoding is better than cascade decoding: a channel decoder followed by an independent source decoder. In the final experiment, we explore the use of optimized irregular LDPC codes instead of regular ones. For all the experiments, we use additive white Gaussian noise channels and the sources are assumed to be i.i.d. with $\mathbb{P}[s_v = 1] = p$.

For the first experiment we have used three codes. The first scheme, denoted as LDPC-8-2-4, consists of two concatenated LPDC codes with rates $R_{sc} = 2/8$ and $R_{cc} = 2/4$ for source and channel coding respectively. The second scheme, LDPC-8-3-4, consists of two concatenated LPDC codes with rates $R_{sc} = 3/8$ and $R_{cc} = 3/4$. The last scheme, LDPC-8-4, consists of a single LDPC code whose compression rate is $R_{sc} = 4/8$. This code only compresses the source and does not add any redundancy to the transmitted bits.

In Fig. 2, we show the bit error rate (BER) as a function of the E_b/N_0 for i.i.d. input bits with $p = 0.02$. For each code there is a set of three plots: the BER predicted by the EXIT chart (dash-dotted line), the BER for codewords with 3200 bits (solid lines) and the BER for codewords with 1600 bits (dashed lines). The three leftmost plots are for LDPC-8-2-4 ($\circ$), the three middle plots are for LDPC-8-3-4 ($\times$) and the rightmost plots for LDPC-8-4 ($\square$).

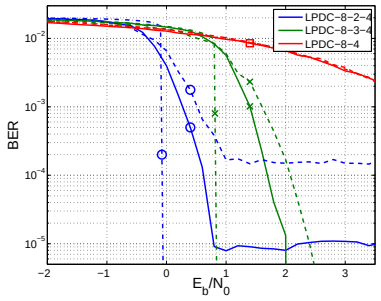

Fig. 2. BER versus E_b/N_0 for LDPC-8-2-4, LDPC-8-3-4 and LDPC-8-4.

In Fig. 2 we observe the standard behavior of the joint decoder for two concatenated LDPC codes, one for channel coding and the other for source coding. As E_b/N_0 increases there is a sharp decline in the BER due to the channel code operating below capacity and, as expected, this transition is sharper as the code length increases. There is a residual BER at high E_b/N_0, only observable for LDPC-8-2-4 in Fig. 2, due to the source decoder not being able to correctly detect all the compressed words for finite-length codes. This residual BER tends to zero as the codeword length increases, because R_{sc} is above the entropy of the source. The residual BER does not show for LDPC-8-3-4, because its R_{sc} is higher.

In the plots we observe about $1dB$ gain when we compare LDPC-8-2-4 with LDPC-8-3-4 for BER larger than the residual BER. This gain is due to the additional redundancy in LDPC-8-2-4. The price we pay for this gain is a higher residual BER. We can trade off the expected gain and the residual BER by parameterizing the source $R_{sc} = \ell/8$ and channel $R_{cc} = \ell/4$ code rates with ℓ. The value of $\ell \in (0, 4]$ is inversely proportional to the gain and to the residual BER. We can also decrease the residual BER by increasing the code length, as illustrated in Fig. 2.

For LPDC8-4, the BER does not improve as we increase the code length (the three lines superimpose). A single sparse graph for source and channel coding with $n > m$ does not possess error correcting capabilities, because equally likely inputs present outputs whose distance is constant and independent of the code length. Thereby, there cannot be a gain as we increase the code length.

In the second experiment, we decode the LPDC-8-2-4 scheme with a cascade decoder and compare it with the joint scheme proposed in this paper. The cascade decoder first decodes the channel code assuming the compressed bits are equally likely i.i.d. bits and then it decodes the source bits using the LLRs output by the channel decoder.

In Fig. 3, we plot the BER for both decoders as a function of the E_b/N_0 for i.i.d. input bits with $p = 0.02$. For each decoder there are two plots: the BER estimated by the EXIT chart (dash-dotted line) and the BER for codewords with 3200 bits (solid line). The two leftmost plots are for the joint decoding ($\circ$) and the rightmost plots for the cascade decoding ($\times$).

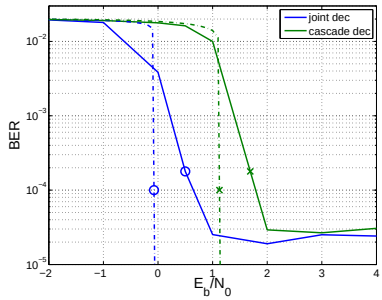

Fig. 3. BER versus E_b/N_0 for the joint and cascade decoder for LDPC-8-2-4.

For low E_b/N_0 neither decoding procedure is able to decode the transmitted word and they provide chance level performance, BER $= p$. The signal to noise ratio is below capacity and the redundancy is not high enough to decode correctly the transmitted words. For high E_b/N_0 both decoding procedures return the same residual BER. There are no errors due to the channel decoder and the residual BER is solely due to the source decoder failing to return the correct word. There is a range in E_b/N_0 between 0 and $1dB$, in which the joint decoder returns the residual BER (low BER) and the cascade decoder returns chance level performance (high BER).

That the channel and source decoders work together to return the correct word explains the difference in performance

in this E_b/N_0 range. The redundancy not removed by the source encoder gives additional information to the channel decoder to return the correct word. This information is not present in the cascade decoder and therefore the channel decoder is unable to decode the correct word. The joint decoder provides in this particular example a $1dB$ gain with respect to the cascade decoder. This gain remains unchanged in the EXIT charts. This gain disappears only if the rates of the source and channel encoders tend to the entropy and capacity, respectively, as the codeword length increases. But for finite-length codes, the rate of the source cannot approach capacity and we obtain a gain from a joint source-channel decoder.

Finally, we present the performance of irregular optimized LDPC codes obtained by using the method described in Section IV. For the channel code, we adopt the first LDPC code with $R_{cc} = 1/2$ in [15].

For the source code, we fix $p = 0.03$ and the degree distribution of the check nodes to $\rho(x) = 0.5x^{21} + 0.5x^{22}$ and using the optimization procedure described in Section IV, we obtain the degree distribution for the variable nodes: $\lambda(x) = 0.098x + 0.274x^3 + 0.025x^7 + 0.292x^9 + 0.075x^{33} + 0.234x^{34}$.

The rate of the source code is $R_{sc} \approx 0.24$ and the overall coding rate is around 2.08. We denote this scheme as LDPCi-8-2-4 and we compare it with LPDC-8-2-4, as their rates are similar. In Fig. 4, we plot the BER for the two codes as a function of the E_b/N_0 for i.i.d. bits with $p = 0.03$. For each code there is a set of three plots: the BER for codewords with 3200 bits (solid lines), the BER for codewords with 6400 bits (dash-dotted lines) and the BER for codewords with 12800 bits (dashed lines). The three top plots are for LDPC-8-2-4 ($\circ$), the three bottom plots are for LDPCi-8-2-4 ($\times$).

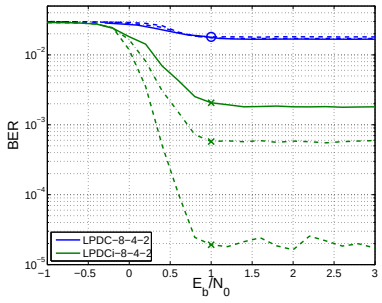

Fig. 4. BER versus E_b/N_0 for the codes LDPC-8-2-4 and LDPCi-8-2-4.

In Fig. 4 we observe that for the regular code as the codeword length increases the residual BER remains constant, while the irregular code is able to reduce its residual BER gradually with the code length. This result is similar to the typical results for LPDC codes for channel coding. Regular codes cannot approach capacity and need a margin in their rate to be able to reduce the BER towards zero, while irregular LDPC can achieve capacity as the code length increases. Therefore, if the source code rate approaches the entropy of the source, we need optimized irregular codes in order to reduce the BER with the code length.

We have also verified our double LDPC transmitter with high redundancy sources using a rate 1/2 source encoder and a rate 1/4 channel encoder and we have compare it with a rate 1/2 channel encoder with a joint decoder that knows the source statistics. The results were similar to the shown results in the first experiment. Moreover, we have also considered Markovian correlated sources and achieved gains holds. These results have not been reported here for lack of space.

VI. Conclusions

We have proposed a new procedure for joint source-channel coding, using two concatenated LDPC codes. We have studied these codes using EXIT charts and our simulations results confirm the agreement between the EXIT chart predictions and the performance of these finite-length codes. Finally, we have suggested a procedure for optimizing the degree distribution of our code with EXIT charts that achieves good performance when R_{sc} tends to the entropy and R_{cc} to capacity.

References

[1] S. Vembu, S. Verdú, and Y. Steinberg, "The source-channel separation theorem revisited," *IEEE Trans. on Inform. Theory*, vol. 41, no. 1, pp. 44–54, Jan. 1995.

[2] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," *IEEE Trans. on Inform. Theory*, vol. 23, no. 3, pp. 337–343, May 1977.

[3] T. Richardson and R. Urbanke, *Modern Coding Theory*. Cambridge University Press, Cambridge, UK, 2008.

[4] E. Ordentlich, G. Seroussi, S. Verdú, and K. Viswanathan, "Universal algorithms for channel decoding of uncompressed sources," *IEEE Trans. on Inform. Theory*, vol. 54, no. 5, pp. 2243–2262, May 2008.

[5] J. Garcia-Frias and J. D.Villasenor, "Combining hidden Markov source models and parallel concatenated codes," *IEEE Commun. Lett.*, vol. 1, no. 7, pp. 111–113, Jul. 1997.

[6] T. Hindelang, J. Hagenauer, and S. Heinen, "Source-controlled channel decoding: Estimation of correlated parameters," in *Proc. 3rd Int. ITG Conf. on Source and Channel Coding*, Munich, Germany, 2000, pp. 259–266.

[7] J. Garcia-Frias and J. D.Villasenor, "Joint turbo decoding and estimation of hidden Markov models," *IEEE J. Sel. Areas Commun.*, vol. 19, no. 9, pp. 1671–1679, Sep. 2001.

[8] G.-C. Zhu and F. Alajaji, "Joint source-channel turbo coding for binary markov sources," *IEEE Trans. on Wireless Comm.*, vol. 5, no. 5, pp. 1065–1075, May 2006.

[9] A. Guyader, E. Fabre, C. Guillemot, and M. Robert, "Joint source-channel turbo decoding of entropy coded sources," *IEEE J. Sel. Areas Commun.*, vol. 19, no. 9, pp. 1680–1696, Sep. 2001.

[10] G. Caire, S. Shamai, and S. Verdú, "Almost-noiseless joint source-channel coding-decoding of sources with memory," in *Proc. 5th Int. ITG Conf. on Source and Channel Coding*, Munich, Germany, Jan. 2004.

[11] S. T. Brink, "Designing iterative decoding schemes with the extrinsic information transfer chart," *AEU Int. J. Electron. Commun.*, vol. 54, pp. 389–398, Dec. 2000.

[12] G. Caire, S. Shamai, and S. Verdú, "Noiseless data compression with low-density parity-check codes," in *Advances in Network Information Theory*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, American Mathematical Society, Piscataway, NJ, 2004.

[13] M. Luby, M. Mitzenmacher, A. Shokrollahi, and D. Spielman, "Analysis of low-density codes and improved designs using irregular graphs," in *Proc. 30th ACM Symp. Theory of Computing*, Dallas, TX, 1998, pp. 249–258.

[14] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. on Inform. Theory*, vol. 47, pp. 619–637, Feb. 2001.

[15] R. Urbanke, "http://lthcwww.epfl.ch/research/ldpcopt/," 2002.

[16] S.-Y. Chung, T. J. Richardson, and R. Urbanke, "Analysis of sum-product decoding of low-density-parity-check codes using gaussian approximation," *IEEE Trans. on Inform. Theory*, vol. 47, pp. 657–670, Feb. 2001.