Multi-edge Optimization of Low-Density Parity-Check Codes for Joint Source-Channel Coding

H. V. Beltrão Neto and W. Henkel
Jacobs University Bremen
Campus Ring 1
D-28759 Bremen, Germany
Email: {h.beltrao, w.henkel}@jacobs-university.de

***Abstract*—We present a novel joint source-channel coding system based on low-density parity-check codes where the amount of information about the source bits available at the decoder is increased by improving the connection profile between the factor graphs that compound the joint system. Furthermore, we propose an optimization strategy for the component codes based on a multi-edge-type joint optimization. Simulation results show a significant improvement in the performance compared to existent joint systems based on low-density parity-check codes.**

I. Introduction

The "separation principle" between source and channel coding states that there is no loss in asymptotic performance when source and channel coding are performed separately. It is though widely observed that for communication systems transmitting in the non-asymptotic regime with limited delay constraints, the separation principle may not be applicable and gains in complexity and fidelity may be obtained by a joint design strategy [1].

In this paper, we investigate a joint system which performs linear encoding of sources by means of error-correcting codes. The strategy of such schemes is to treat the source output $\mathbf{u}$ as an error pattern and perform compression calculating the syndrome generated by $\mathbf{u}$, i.e., the source encoder calculates $\mathbf{s} = \mathbf{u}\mathbf{H}^T$, where $\mathbf{H}$ is the parity-check matrix of the linear error-correcting code being considered as source encoder, and the syndrome $\mathbf{s}$ represents the compressed sequence. Herein we consider only binary memoryless sources, since it is widely known that linear source codes achieve the entropy rate for this kind of sources. Nevertheless, the optimality of linear source compression can be extended to very general sources with memory and nonstationarity [2].

Compression schemes based on syndrome encoding for binary memoryless sources were developed in the context of variable-to-fixed length algorithms in [3] and [4]. Afterwards, Ancheta [5] developed a fixed-to-fixed linear source code based on syndrome formation. Due to the limitations of the practical error-correcting codes known at that time, this line of research was left aside by the advent of Lempel-Ziv coding. Nevertheless, due to a lack of resilience of state-of-the-art data compressors to transmission errors and to the fact that such compression algorithms just have an efficient performance with block sizes much longer than the ones typically specified in some modern wireless standards, there are state-of-the-art applications that do not apply data compression.

In order to cope with such limitations of some modern data compression algorithms, the authors in [2] proposed the use of syndrome-source compression by means of low-density parity-check (LDPC) codes together with belief propagation decoding, which was further extended in [6] to cope with a noisy channel. In contrast to general linear codes, an LDPC code has a sparse parity-check matrix and can thus be used as a linear compressor with linear complexity in the block length. In addition, syndrome source-coding schemes can be naturally extended to joint source-channel (JSC) encoding and decoding configurations.

One of the schemes proposed in [6] for JSC consists of a serial concatenation of two LDPC codes, where the outer code works as a syndrome-source compressor and the inner code as the channel code. The codeword resulting from such a concatenation is then jointly decoded using the source statistics and channel information by means of the belief propagation algorithm applied to the joint source-channel factor graph. In spite of its introduction in [6], it was in [7] that this scheme was first studied for a JSC application. However, the bit error-rate curves presented in [7] showed considerably high error floors for source output sequences with moderate block length. The proposed solutions to cope with such high error floors were either to reduce the source compression rate or to increase the codeword size, but such solutions have the following drawbacks.

First of all, increasing the size of the codeword would undermine one of the advantages of the JSC scheme, namely the possibility of a better performance in a non-asymptotic scenario. Second, reducing the compression rate is also not desirable, since it pushes the system performance away from its asymptotically achievable capacity. In this paper, we propose the construction of an LDPC-based joint source-channel coding scheme which significantly lowers the error floor resulting from the compression of source sequences that correspond to uncorrectable error patterns of the LDPC codes used as syndrome-source encoder.

This work was funded by the DFG project number HE-3654/16-1.

II. LDPC-BASED JOINT SOURCE-CHANNEL SYSTEM

In [6], the authors proposed a configuration for a joint source-channel encoding system using LDPC codes for both source compression and channel coding. This structure is based on a serial concatenation of two LDPC codes where the outer and the inner codes perform syndrome-source compression and channel coding, respectively. In this concatenated approach, a codeword $\mathbf{c}$ is defined by

$$\mathbf{c} = \mathbf{s} \cdot \mathbf{G}_{cc} = \mathbf{u} \cdot \mathbf{H}_{sc}^T \cdot \mathbf{G}_{cc} \ ,$$

where $\mathbf{G}_{cc}$ is the $l \times m$ LDPC generator matrix of the channel code, $\mathbf{H}_{sc}$ is the $l \times n$ parity-check matrix of the LDPC code applied for source coding, $\mathbf{s}$ is the $1 \times l$ source compressed sequence, and $\mathbf{u}$ is the $1 \times n$ source output.

Considering a binary memoryless source and performing standard belief propagation decoding, the simulation results in [7] showed the presence of error floors in the error-rate curves, which are a consequence of the fact that some output sequences emitted by the source form error patterns that cannot be corrected by the LDPC code used as source compressor.

Our idea to cope with this problem is to improve the amount of information about the source bits available at decoding. We do it by inserting new edges connecting the check nodes of the channel code to the variable nodes of the source code in the factor graph that represents the serial concatenated system introduced in [6]. The reasoning of this strategy is that such an edge insertion will provide an extra amount of extrinsic information to the variable nodes of the source LDPC which will significantly lower the error floor due to uncorrectable source output patterns. We depict this idea in Fig. 1, where the new inserted edges in the concatenated JSC system of [6] are represented by the dashed lines.

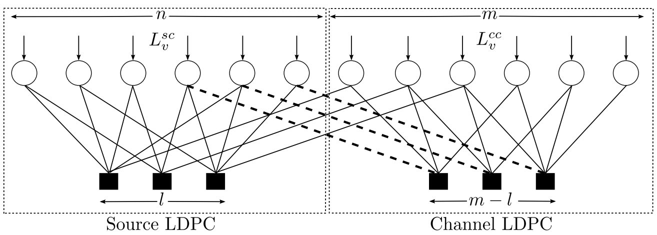

Fig. 1. Joint source-channel factor graph with inserted edges.

The variable and the check nodes of the source LDPC (left) represent the source output and the compressed source sequence, respectively. Since we will consider only binary sources, the variable nodes represent binary symbols. In this system, each check node of the source LDPC is connected to a single variable node of the channel code (right) forming the systematic part of the channel codeword. We consider that only m variable nodes are transmitted (the n source output symbols are punctured prior to transmission). Thus, the overall rate is n/m. Furthermore, L_v^{sc} and L_v^{cc} denote the log-likelihood ratios representing the intrinsic information received by the source ($v = 1, \dots, n$) and channel ($v = n+1, \dots, n+m$) variable nodes, respectively. In this work, we will limit our investigation to memoryless binary sources.

A. Encoder

To understand our proposed serial encoding strategy, consider the representation of the factor graph depicted in Fig. 1 by a $m \times (n+m)$ matrix $\mathbf{H}$. This matrix can be written as

$$\mathbf{H} = \left[\begin{array}{c|c|c} \mathbf{H}_{sc} & \mathbf{I} & \mathbf{0} \\ \hline \mathbf{L} & \multicolumn{2}{c}{\mathbf{H}_{cc}} \end{array}\right],$$

where $\mathbf{H}_{sc}$ is the $l \times n$ source encoder parity-check matrix, $\mathbf{H}_{cc}$ is the $(m-l) \times m$ parity-check matrix of the channel code, $\mathbf{I}$ is an $l \times l$ identity matrix, and $\mathbf{L}$ is an $(m-l) \times n$ matrix, to which we will refer as *linking* matrix. The linking matrix $\mathbf{L}$ represents the connections among the check nodes of the channel code to the variable nodes of the source code.

The encoding of our proposed system diverts slightly from the serial approach of [7]. The difference lies in the fact that the message to be encoded before the transmission is formed by the concatenation of the source output $\mathbf{u}$ and its syndrome $\mathbf{s}$ computed by the source code, i.e., a codeword $\mathbf{c}$ is defined by

$$\mathbf{c} = [\mathbf{u}, \mathbf{s}] \cdot \mathbf{G}_L = [\mathbf{u}, \mathbf{u} \cdot \mathbf{H}_{sc}^T] \cdot \mathbf{G}_L \ , \quad (1)$$

where $\mathbf{G}_L$ is an $(n+l) \times (n+m)$ matrix constructed such that the row space of $\mathbf{G}_L$ is the null space of $[\mathbf{L}, \mathbf{H}_{cc}]$, i.e., $\mathbf{G}_L$ is the generator matrix of a linear systematic code whose parity-check matrix is given by the horizontal concatenation of the matrices $\mathbf{L}$ and $\mathbf{H}_{cc}$. In the following, we show that every codeword of the code spanned by $\mathbf{G}_L$ is a codeword of the code spanned by the null space of $\mathbf{H}$.

Proposition 1: Let $\mathbf{H} = \left[\ [\mathbf{H}_{sc}, \mathbf{I}, \mathbf{0}]^T, [\mathbf{L}, \mathbf{H}_{cc}]^T \ \right]^T$ denote the matrix whose factor graph representation corresponds to the joint system depicted in Fig. 1, $\mathbf{H}_L = [\mathbf{L}, \mathbf{H}_{cc}]$, and $[\mathbf{u}, \mathbf{s}]$ be the concatenation of the source output $\mathbf{u}$ and its syndrome-compressed sequence $\mathbf{s}$. A codeword $\mathbf{c}$ formed by the encoding of the vector $[\mathbf{u}, \mathbf{s}]$ by the linear code spanned by the null space of the matrix $\mathbf{H}_L$ is also a codeword of the linear code spanned by the null space of $\mathbf{H}$.

Proof: Let $\mathbf{G}_L$ denote the systematic generator matrix of the null space of the matrix $\mathbf{H}_L$. Since the code spanned by the rows of $\mathbf{G}_L$ is systematic, its codewords can be written as $\mathbf{c} = [\mathbf{d}, \mathbf{p}]$, where $\mathbf{d}$ is the systematic part of the codeword. Let $\mathbf{d} = [\mathbf{u}, \mathbf{s}]$, then we can write $\mathbf{c} = [\mathbf{u}, \mathbf{s}, \mathbf{p}]$, where $\mathbf{u} = [u_0, \dots, u_{n-1}]$ represents the source output, $\mathbf{s} = [s_0, \dots, s_{l-1}]$ denotes the syndrome compressed sequence, and $\mathbf{p} = [p_0, \dots, p_{m-l-1}]$ is a vector whose elements are the parity bits generated by the inner product between $[\mathbf{u}, \mathbf{s}]$ and $\mathbf{G}_L$. For every codeword $\mathbf{c}$, we have

$$\mathbf{c} \cdot \mathbf{H}_L^T = \mathbf{c} \cdot [\mathbf{L}, \mathbf{H}_{cc}]^T = \mathbf{0} \ . \quad (2)$$

Recall now that according to our compression rule, and since our operations are defined over GF(2), we can write

$$\begin{aligned} [u_0, \dots, u_{n-1}] \cdot \mathbf{H}_{sc}^T &= [s_0, \dots, s_{l-1}] \\ [u_0, \dots, u_{n-1}] \cdot \mathbf{H}_{sc}^T + [s_0, \dots, s_{l-1}] \cdot \mathbf{I} &= \mathbf{0} \ , \end{aligned} \quad (3)$$

where $\mathbf{I}$ is an $l \times l$ identity matrix, and $\mathbf{0}$ is a vector whose elements are all equal to zero. Note that Eq. (3) can be written as

$$[u_0, \dots, u_{n-1}, s_0, \dots, s_{l-1}] \cdot [\mathbf{H}_{sc}, \mathbf{I}]^T = \mathbf{0} \ . \quad (4)$$

Consider now the $l \times (n+m)$ matrix $[\mathbf{H}_{sc}, \mathbf{I}, \mathbf{0}]$. According to Eq. (4), for every vector $\mathbf{p} = [p_0, \ldots, p_{m-l-1}]$, we can write

$$[u_0, \ldots, u_{n-1}, s_0, \ldots, s_{l-1}, p_0, \ldots, p_{m-l-1}] \cdot [\mathbf{H}_{sc}, \mathbf{I}, \mathbf{0}]^T = \mathbf{0} ,$$

i.e.,

$$\mathbf{c} \cdot [\mathbf{H}_{sc}, \mathbf{I}, \mathbf{0}]^T = \mathbf{0} . \tag{5}$$

Finally, consider the inner product

$$\begin{aligned} \mathbf{c} \cdot \mathbf{H}^T &= \mathbf{c} \cdot \left[\ [\mathbf{H}_{sc}, \mathbf{I}, \mathbf{0}]^T, [\mathbf{L}, \mathbf{H}_{cc}]^T \ \right] \\ &= \left[\ \mathbf{c} \cdot [\mathbf{H}_{sc}, \mathbf{I}, \mathbf{0}]^T, \mathbf{c} \cdot [\mathbf{L}, \mathbf{H}_{cc}]^T \ \right] . \end{aligned} \tag{6}$$

Substituting eqs. (2) and (5) into Eq. (6), we have

$$\mathbf{c} \cdot \mathbf{H}^T = \mathbf{0} ,$$

i.e., a codeword $\mathbf{c}$ of the code spanned by the null space of $\mathbf{H}_L$ is also a codeword of the code spanned by the null space of $\mathbf{H}$.

$\square$

The encoding algorithm of our proposed joint source-channel system can be summarized as follows:

1) Given a source output vector $\mathbf{u}$, compute $\mathbf{s} = \mathbf{u} \cdot \mathbf{H}_{sc}^T$.
2) Compute $\mathbf{v} = [\mathbf{u}, \mathbf{s}]$, i.e., the horizontal concatenation of vectors $\mathbf{u}$ and $\mathbf{s}$.
3) Generate the codeword $\mathbf{c} = \mathbf{v} \cdot \mathbf{G}_L$.
4) Transmit $\mathbf{c}$ after puncturing its first n bits.

Steps 1 and 3 are the source and channel encoding steps, respectively. Since $\mathbf{H}_{sc}$ is sparse, the source encoding has a complexity that is linear with respect to the block length. Furthermore, applying the technique presented in [8] for encoding LDPC codes by means of their parity-check matrix, the complexity of the channel encoding can be made approximately linear.

B. Decoder

The decoding of the LDPC-based joint source-channel system is done by means of the belief propagation algorithm applied to the factor graph of Fig. 1, whose structure is known to both the encoder and the decoder. We assume that the decoder knows the statistics of the source.

Herein, we assume that the source is a memoryless Bernoulli source with success probability p_v, and that the transmission takes place through a binary input AWGN channel. Within this framework, we can write $L_v^{sc} = \log\left(\frac{1-p_v}{p_v}\right)$ and $L_v^{cc} = \frac{2y_v}{\sigma_n^2}$ where y_v is the received BPSK modulated codeword transmitted through an AWGN with noise variance σ_n^2 (consequently L_v^{cc} has variance $\sigma_{ch}^2 = 4/\sigma_n^2$).

III. Multi-edge notation

A. Multi-edge-type LDPC codes

Multi-edge-type LDPC codes [9] are a generalization of irregular and regular LDPC codes. Diverting from standard LDPC ensembles where the graph connectivity is constrained only by the node degrees, in the multi-edge setting, several edge classes can be defined, and every node is characterized by the number of connections to edges of each class. Within this framework, the code ensemble can be specified through two node-perspective multinomials associated to variable and check nodes, which are defined respectively by [9]

$$\nu(\mathbf{r}, \mathbf{x}) = \sum \nu_{\mathbf{b},\mathbf{d}} \mathbf{r}^{\mathbf{b}} \mathbf{x}^{\mathbf{d}} \quad \text{and} \quad \mu(\mathbf{x}) = \sum \mu_{\mathbf{d}} \mathbf{x}^{\mathbf{d}}, \tag{7}$$

where $\mathbf{b}$, $\mathbf{d}$, $\mathbf{r}$, and $\mathbf{x}$ are vectors which are explained as follows. First, let m_e denote the number of edge types used to represent the graph ensemble and m_r the number of different received distributions. The number m_r represents the fact that the different bits can go through different channels and thus, have different received distributions. Each node in the ensemble graph has associated to it a vector $\mathbf{x} = (x_1, \ldots, x_{m_e})$ that indicates the different types of edges connected to it and a vector $\mathbf{d} = (d_1, \ldots, d_{m_e})$ referred to as *edge degree vector* which denotes the number of connections of a node to edges of type i, where $i \in (1, \ldots, m_e)$.

For the variable nodes, there is additionally the vector $\mathbf{r} = (r_0, \ldots, r_{m_r})$, which represents the different received distributions and the vector $\mathbf{b} = (b_0, \ldots, b_{m_r})$, which indicates the number of connections to the different received distributions (b_0 is used to indicate a variable node with no available intrinsic information at the decoder). We use $\mathbf{x}^{\mathbf{d}}$ to denote $\prod_{i=1}^{m_e} x_i^{d_i}$ and $\mathbf{r}^{\mathbf{b}}$ to denote $\prod_{i=0}^{m_r} r_i^{b_i}$. Finally, the coefficients $\nu_{\mathbf{b},\mathbf{d}}$ and $\mu_{\mathbf{d}}$ are non-negative reals such that, if n is the total number of variable nodes, $\nu_{\mathbf{b},\mathbf{d}} n$ and $\mu_{\mathbf{d}} n$ represent the number of variable nodes of type ($\mathbf{b}$,$\mathbf{d}$) and check nodes of type[1] $\mathbf{d}$, respectively.

B. Multi-edge notation for joint source-channel factor graphs

In order to being able to quantify the amount of information exchanged by the individual factor graphs representing the channel and source codes during decoding, we define herein a multi-edge framework for the JSC system. Within this framework, we define four edge types within the corresponding graph, i.e., $m_e = 4$. Additionally, now we also have two different received distributions corresponding to the source statistics and channel information, respectively. Figure 2 depicts the four edge types and two received distributions. The solid and dashed lines depict type-1 and type-2 edges, respectively. The type-3 and type-4 edges are depicted by the dash-dotted and dotted lines, respectively. Moreover, the received distributions of the source and channel variable nodes are depicted by solid and dashed arrows, respectively. Note that the source and channel code factor graphs exchange information solely through type-3 and type-4 edges. Since the variable nodes have access to two different observations, the vector $\mathbf{r} = (r_0, r_1, r_2)$ has three components. The first component (r_0) represents a bit with no available intrinsic information, second component (r_1) corresponds to the observation accessible to the n source LDPC variable nodes, and the third component (r_2) denotes the channel observations, which are available only to the m channel LDPC variable nodes. Furthermore, since each variable node has access to either the source statistics or the channel observation, we can write $\mathbf{b} = (0, 1, 0)$ for the source and $\mathbf{b} = (0, 0, 1)$ for the channel variable nodes, respectively.

[1]We will frequently refer to nodes with edge degree vector $\mathbf{d}$ as "type $\mathbf{d}$" nodes.

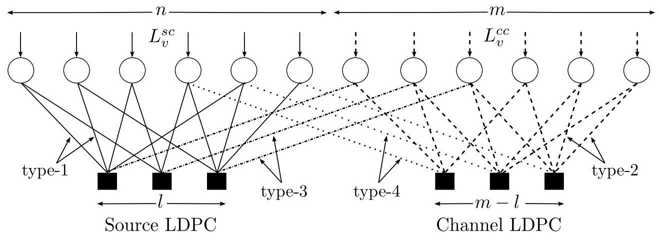

Fig. 2. Multi-edge joint source-channel factor graph.

IV. Asymptotic analysis

In this section, we derive the multi-edge-type mutual information evolution equations for LDPC-based joint source-channel coding systems. We will use the edge-perspective degree distributions $\lambda^{(j)}(\mathbf{r},\mathbf{x})$ and $\rho^{(j)}(\mathbf{x})$ to describe the evolution of the mutual information between the messages sent through type-j edges and the associated variable node values. The edge-perpective multi-edge degree distributions can be written as

$$\lambda^{(j)}(\mathbf{r},\mathbf{x}) = \frac{\nu_{x_j}(\mathbf{r},\mathbf{x})}{\nu_{x_j}(\mathbf{1},\mathbf{1})}, \quad \rho^{(j)}(\mathbf{x}) = \frac{\mu_{x_j}(\mathbf{x})}{\mu_{x_j}(\mathbf{1})}, \tag{8}$$

where $\nu_{x_j}(\mathbf{r},\mathbf{x})$ and $\mu_{x_j}(\mathbf{x})$ are the derivatives of $\nu(\mathbf{r},\mathbf{x})$ and $\mu(\mathbf{x})$ with respect to x_j, respectively.

Note that, since we are dealing with syndrome-source encoding (a framework where the source output is analogous to an error pattern) of memoryless Bernoulli sources with a probability of emitting a one equal to p_v, we can model the received distributions of the source code variable nodes as the distribution of the output of a BSC with crossover probability p_v [5].

Let $I_{v,l}^{(j)}$ ($I_{c,l}^{(j)}$) denote the mutual information (MI) between the messages sent through type-j edges at the output of variable (check) nodes at iteration l and the associated variable node value. Assuming Gaussian approximation [10] of the messages exchanged through the joint factor-graph during BP decoding, we can express the mutual information equation for the channel code variable nodes, i.e., for $j \in \{2,3\}$ as

$$I_{v,l}^{(j)} = \sum_{\mathbf{d}} \lambda_{\mathbf{d}}^{(j)} J(\sigma_{ch}^2 + (d_j-1)[J^{-1}(I_{c,l-1}^{(j)})] + \sum_{s\neq j} d_s[J^{-1}(I_{c,l-1}^{(s)})]) , \tag{9}$$

where σ_{ch}^2 is the variance of the received channel message, $\lambda_{\mathbf{d}}^{(j)}$ is the probability of a type-j edge being connected to a variable node with edge degree vector $\mathbf{d}$, and the function $J(\cdot)$ relates all the MI quantities to the variance of LLR messages and is defined as [11]

$$J(\sigma^2) = 1 - \int_{-\infty}^{\infty} \frac{e^{\frac{(\xi-\sigma^2/2)^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}} \cdot \log_2[1+e^{-\xi}]d\xi \ .$$

In addition, for the source code variable nodes, i.e., for $j \in \{1,4\}$, we can write

$$I_{v,l}^{(j)} = \sum_{\mathbf{d}} \lambda_{\mathbf{d}}^{(j)} J_{BSC}((d_j-1)[J^{-1}(I_{c,l-1}^{(j)})] + \sum_{s\neq j} d_s[J^{-1}(I_{c,l-1}^{(s)})], p_v) , \tag{10}$$

with the function J_{BSC} defined as [7]

$$J_{BSC}(\sigma^2,p_v) = (1-p_v)I(x_v;\mathcal{L}^{(1-p_v)}) + p_v I(x_v;\mathcal{L}^{(p_v)}) ,$$

where x_v denotes the corresponding bitnode variable, $\mathcal{L}^{(1-p_v)} \sim \mathcal{N}(\frac{\sigma^2}{2}+L_v^{sc},\sigma^2)$, and $\mathcal{L}^{(p_v)} \sim \mathcal{N}(\frac{\sigma^2}{2}-L_v^{sc},\sigma^2)$.

Finally, the mutual information between the messages sent by a check node through a type-j edge and its associated variable value for both source and channel LDPC codes (i.e., for all j) can be written as

$$I_{c,l}^{(j)} = 1 - \sum_{i=1}^{d_{c_{max}}^{(j)}} \sum_{\mathbf{d}:d_j=i} \rho_{\mathbf{d}}^{(j)} \cdot J((d_j-1)[J^{-1}(1-I_{v,l}^{(j)})] + \sum_{s\neq j} d_s[J^{-1}(1-I_{v,l}^{(s)})]) , \tag{11}$$

where $\rho_{\mathbf{d}}^{(j)}$ is the probability of a type-j edge being connected to a check node with edge degree vector $\mathbf{d}$, and $d_{c_{max}}^{(j)}$ is the maximum number of type-j edges connected to a check node.

In order to limit the search space of the optimization algorithm, we consider only check-regular source and channel LDPC codes. Furthermore, the check nodes of source and channel LDPC codes are considered to have edge degree vectors $\mathbf{d} = (d_{c_1},0,1,0)$ and $\mathbf{d} = (0,d_{c_2},0,1)$, respectively. As a consequence, the multi-edge check node degree distributions of the source and channel LDPC codes are given by $\rho^{(1)}(\mathbf{x}) = x_1^{d_{c_1}-1}$ and $\rho^{(2)}(\mathbf{x}) = x_2^{d_{c_2}-1}$, respectively.

A. Source code mutual information evolution

For the source code factor graph, the variable nodes only have connections to type-1 and type-4 edges, i.e., all source code variable nodes have an edge degree vector $\mathbf{d} = (d_1,0,0,d_4)$ where $d_1 \in \{2,\ldots,d_{v_{max}}^{(1)}\}$, and $d_4 \in \{0,1\}$. We can summarize the set of mutual information evolution equations as follows:

- variable nodes messages update:
$$I_{v,l}^{(1)} = \sum_{\mathbf{d}} \lambda_{\mathbf{d}}^{(1)} J_{BSC}((d_1-1)[J^{-1}(I_{c,l-1}^{(1)}(\mathbf{d}))] + d_4[J^{-1}(I_{c,l-1}^{(4)}(\mathbf{d}))], p_v) \tag{12}$$
- check nodes messages update:
$$I_{c,l}^{(1)}(\mathbf{d}) = 1 - J((d_{c_1}-1)[J^{-1}(1-I_{v,l}^{(1)})] + [J^{-1}(1-I_{v,l}^{(3)}(\mathbf{d}))]) \tag{13}$$
- source to channel decoder messages update:
$$I_{v,l}^{(4)}(\mathbf{d}) = d_4 \cdot J_{BSC}(d_1[J^{-1}(I_{c,l-1}^{(1)}(\mathbf{d}))], p_v) \tag{14}$$
$$I_{c,l}^{(3)} = 1 - J(d_{c_1}[J^{-1}(1-I_{v,l}^{(1)})]) \tag{15}$$

- channel decoder messages update:

$$I_{v,l}^{(3)}(\mathbf{d}) = T_v(I_{c,l-1}^{(3)}, I_{v,l-1}^{(4)}(\mathbf{d})) \tag{16}$$

$$I_{c,l}^{(4)}(\mathbf{d}) = T_c(I_{c,l}^{(3)}, I_{v,l}^{(4)}(\mathbf{d})) \tag{17}$$

where $T_v(\cdot)$ and $T_c(\cdot)$ are the transfer functions of the channel decoder, which is considered to be fixed. Given the channel code degree distribution $\lambda^{(2)}(\mathbf{r},\mathbf{x})$ and $\rho^{(2)}(\mathbf{x})$, those functions can be explicitly computed by means of eqs. (9) and (11) for every edge degree vector $\mathbf{d}$ [2]. In the computation of $T_v(\cdot)$ and $T_c(\cdot)$, the rightmost sum in Eq. (11) will be zero if $I_{v,l}^{(4)}(\mathbf{d}) = 0$, since the corresponding check node is not receiving any information through type-4 edges in this case.

Combining eqs. (12) - (17) we can summarize the mutual information evolution for the source code as a function of the mutual information in the previous iteration, the source statistics, the channel condition, and the degree distributions:

$$I_{v,l}^{(1)} = F_1(\underline{\lambda}, \underline{d_c}, I_{v,l-1}^{(1)}, p_v, \sigma_{ch}) \ , \tag{18}$$

where $\underline{d_c} = [d_{c_1}, d_{c_2}]$, and $\underline{\lambda} = [\underline{\lambda^{(1)}}, \underline{\lambda^{(2)}}]$ with $\underline{\lambda^{(j)}}$ denoting the sequence of coefficients $\lambda_{\mathbf{d}}^{(j)}$ for all $\mathbf{d}$ and $j \in \{1,2\}$. The initial conditions are $I_{v,0}^{(4)}(\mathbf{d}) = I_{c,0}^{(4)}(\mathbf{d}) = I_{c,0}^{(1)}(\mathbf{d}) = 0 \ \forall \ \mathbf{d}$, and $I_{c,0}^{(3)} = 0$.

By means of Eq. (18), given a channel LDPC code, we can predict the convergence behavior of the iterative decoding for the source code and then optimize the multi-edge edge-perspective variable node degree distributions $\lambda^{(1)}(\mathbf{r},\mathbf{x})$ under the constraint that the mutual information must be increasing as the number of iterations grows.

V. Optimization

In the proposed algorithm herein, we first compute the rate optimal channel LDPC code assuming a transmission over an AWGN channel with noise variance σ_n^2. This is a standard irregular LDPC optimization [12] and since we are not considering any connection to the source code in this first step, it can be done by means of eqs. (9) and (11) with $\mathbf{d} = (0, d_2, 0, 0)$ and $d_2 \in \{2, \ldots, d_{v_{max}}^{(2)}\}$, where $d_{v_{max}}^{(j)}$ denotes the maximum number of type-j edges connected to a variable node. The optimized degree distribution obtained at this step will be denoted as $\lambda_0^{(2)}(\mathbf{r},\mathbf{x})$.

After having optimized the channel code variable nodes degree distribution, we assign the variable nodes of higher degree to the message bits. This is done in order to better protect the compressed message transmitted through the channel, since the more connected a variable node, the better its error error rate performance [13]. This can be done as follows,

1) Given $\lambda_0^{(2)}(\mathbf{r},\mathbf{x})$, compute the node-perspective multi-edge degree distribution $\nu_0(\mathbf{r},\mathbf{x}) = \frac{\int \lambda_0^{(2)}(\mathbf{r},\mathbf{x})dx_2}{\int_0^1 \lambda_0^{(2)}(\mathbf{r},\mathbf{x})dx_2}$.
2) Assign a fraction R_{cc} of nodes (the ones with higher degree) to the systematic part of the codeword, where R_{cc} is the rate of the channel code. This is done by turning a variable node with edge degree vector $\mathbf{d} = (0, d_2, 0, 0)$ into a variable node with edge degree vector $\mathbf{d} = (0, d_2, 1, 0)$. This gives rise to a modified node-perspective degree distribution $\nu(\mathbf{r},\mathbf{x})$, where a fraction of R_{cc} nodes have one connection to type-3 edges.
3) Given $\nu(\mathbf{r},\mathbf{x})$, compute the new edge-perspective multi-edge variable node degree distribution $\lambda^{(2)}(\mathbf{r},\mathbf{x}) = \frac{\nu_{x_2}(\mathbf{r},\mathbf{x})}{\nu_{x_2}(\mathbf{1},\mathbf{1})}$.

Once we have optimized the channel code, we optimize (maximizing its rate) the source LDPC code considering its connections to the channel LDPC code graph.

Let $\underline{d_{v_{max}}} = [d_{v_{max}}^{(1)}, d_{v_{max}}^{(2)}, d_{v_{max}}^{(3)}, d_{v_{max}}^{(4)}]$ be a vector whose components $d_{v_{max}}^{(j)}$ represent the maximum number of connections of a single variable node to type-j edges. Also, recall that the components of the vector $\underline{d_c} = [d_{c_1}, d_{c_2}]$ define the number of connections of the source code check nodes to type-1 edges (d_{c_1}) and the number of connections of the channel code check nodes to type-2 edges (d_{c_2}). Additionally, $\underline{\lambda^{(j)}}$ denote the sequence of the coefficients of $\lambda^{(j)}(\mathbf{r},\mathbf{x})$. Given $\underline{d_{v_{max}}}$, $\underline{d_c}$, p_v, and $\sigma_n^2 = \sigma_{ch}^2/4$, the optimization problem can be written as shown in Algorithm 1.

Algorithm 1 Joint source-channel code optimization

1) Optimize the rate of the channel LDPC code without considering the connections to the factor graph of the source LDPC code. Save the obtained the degree distribution $\lambda_0^{(2)}(\mathbf{r},\mathbf{x})$.
2) Compute $\lambda^{(2)}(\mathbf{r},\mathbf{x})$ by assigning as systematic bits a fraction of the variable nodes with higher degrees of the optimized channel LDPC code.
3) Considering $\underline{\lambda} = [\underline{\lambda^{(1)}}, \underline{\lambda^{(2)}}]$, maximize $\sum_{s=2}^{d_{v_{max}}^{(1)}} \sum_{\mathbf{d}:d_1=s} \lambda_{\mathbf{d}}^{(1)}/s$ under the following constraints,
$\mathcal{C}_1 \ : \sum_d \lambda_{\mathbf{d}}^{(1)} = 1 \ ,$
$\mathcal{C}_2 \ : F_1(\underline{\lambda}, \underline{d_c}, I, p_v, \sigma_{ch}) > I, \forall \ I \ \in \ [0,1) \ ,$
$\mathcal{C}_3 \ : \sum_{\mathbf{d}:d_1=2} \lambda_{\mathbf{d}}^{(1)} < \frac{1}{2\sqrt{p_v(1-p_v)}} \cdot \frac{1}{(d_{c_1}-1)} \ ,$
$\mathcal{C}_4 \ : \sum_{\mathbf{d}:d_4>0} \frac{\lambda_{\mathbf{d}}^{(1)}}{d_1} = 1/(d_{c_1} d_{c_2} \sum_{\mathbf{d}:d_3=1} \frac{\lambda_{\mathbf{d}}^{(2)}}{d_2}) \ .$

where $\mathcal{C}_1$ and $\mathcal{C}_2$ are the proportion and convergence constraints, respectively. Since we are considering the convergence only through edges of type-1, the stability condition $\mathcal{C}_3$ remains the same as for standard LDPC codes ensembles with codewords transmitted over a BSC with transition probability p_v [9]. Furthermore, the rate constraint $\mathcal{C}_4$ must be considered due to the fact that the number of type-4 edges connected to the source code variable nodes must be equal to the number of channel code check nodes (since we assume that every channel code check nodes is connected to only one type-4 edge).

For given $\underline{\lambda^{(2)}}$, $\underline{d_c}$, p_v, and σ_{ch}, the constraints $\mathcal{C}_1$, $\mathcal{C}_2$, $\mathcal{C}_3$, and $\mathcal{C}_4$ are linear in the parameter $\underline{\lambda^{(1)}}$. This means that the optimization of both source and channel codes can be solved by linear programming. For a given channel condition, every different set of vectors $\underline{d_{v_{max}}}$, $\underline{d_c}$ will give rise to systems with a different overall rate. In practice, we fix the vector $\underline{d_{v_{max}}}$ and vary d_{c_1} and d_{c_2} to obtain the joint system with maximum overall rate for a binary symmetric source with

[2] For the computation of $T_v(\cdot)$, note that by means of Eq. (8) we can write $\lambda_{\mathbf{d}}^{(3)}(\mathbf{r},\mathbf{x}) = \left[\frac{\int \lambda_{\mathbf{d}}^{(2)}(\mathbf{r},\mathbf{x})}{\int_0^1 \lambda_{\mathbf{d}}^{(2)}(\mathbf{r},\mathbf{x})}\right]'_{x_3}$, where f'_x denotes the partial derivative of f with respect to x.

transition probability p_v and an AWGN with noise variance σ_n^2.

VI. SIMULATION RESULTS

Herein, we present simulation results obtained with an LDPC-based JSC coding system constructed according to the degree distributions optimized by the algorithm previously proposed. We optimized a system with the following parameters: $p_v = 0.03$, $\sigma_n^2 = 0.95$, $\underline{d_{v_{max}}} = [30, 30, 1, 1]$, and $\underline{d_c} = [22, 6]$. For such source and channel conditions, the asymptotically optimal Shannon limit is $C/H(S) \simeq 2.58$ source symbols per channel use, where C is the channel capacity and $H(S)$ is the source entropy. The compression rate obtained for the source code was $R_{sc} = 0.2361$, and the transmission rate obtained for the channel LDPC code was $R_{cc} = 0.4805$ giving an overall coding rate of $R_{over} = R_{cc}/R_{sc} \simeq 2.03$.

In order to show the merits of the proposed optimization, we compare the performance of our proposed system with the LDPC-based JSC systems with the same overall rate $R_{over} = 2.03$ introduced by Caire *et al.* in [6] to which we will refer as System I. This system has the edges between the check nodes of the source LDPC code and the systematic variable nodes of the channel LDPC code as the only connections between the factor graphs of the source and channel codes. Furthermore, it consists of a source code jointly optimized with a fixed channel code previously optimized for the AWGN channel. Every simulation point presented was obtained considering BPSK modulated signal transmitted over an AWGN channel and a total of 50 decoding iterations.

Since we are interested in an almost-noiseless compression scenario, we chose the word-error rate (WER) as figure of merit. The simulation results for our optimized system (referred to as JSC opt) and System I with source message of lengths $n = 3200$ and $n = 6400$ are depicted in Fig. 3. As mentioned previously, the results for System I show an very high error floor for high SNR's which are a consequence of the compression of source codewords that form error patterns not correctable by the source LDPC code. Figure 3 shows that our proposed system managed to significantly lower this error floor while keeping the overall rate constant.

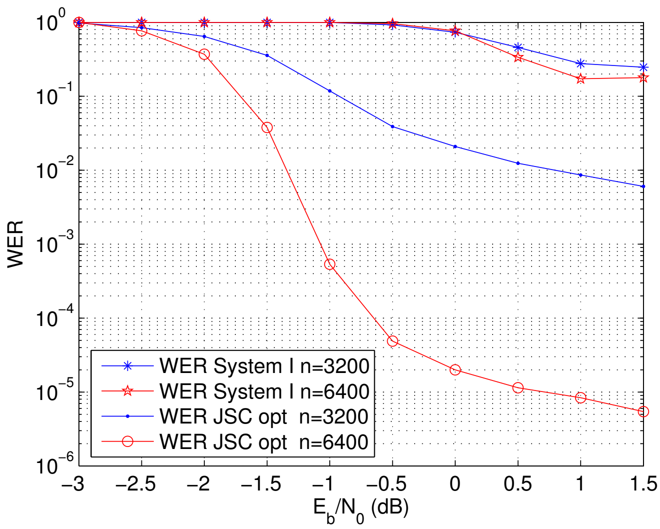

Fig. 3. Performance of joint source-channel coded systems with $R_{over} = 2.03$ for $n = 3200$ and $n = 6400$.

VII. CONCLUDING REMARKS

We proposed an LDPC-based joint source-channel coding scheme and, by means of a multi-edge analysis, proposed an optimization algorithm for such systems. Based on a syndrome source-encoding, we presented a novel configuration where the amount of information about the source bits available at the decoder is increased by improving the connection profile between the factor graphs of the source and channel codes that form the joint system. The presented simulation results show a significant reduction of the error floor caused by the encoding of messages that correspond to uncorrectable error patterns of the LDPC code used as source encoder in comparison to existent LDPC-based joint source-channel coding systems.

The next step will be to improve our design by placing infinite reliability on some source variable nodes. This was done in the context of pure source compression in [2] and will be the subject of our future investigations in order to lower even more the error floor presented in our simulations.

REFERENCES

[1] J. L. Massey, "Joint source and channel coding," *Communications Systems and Random Process Theory*, vol. 11, pp. 279–293, Sijthoff and Nordhoff, 1978.

[2] G. Caire, S. Shamai, and S. Verdú, "A new data compression algorithm for sources with memory based on error correcting codes," in *IEEE Workshop on Information Theory*, Paris, France, Mar. 30 - Apr. 4 2003, pp. 291–295.

[3] P. E. Allard and A. W. Bridgewater, "A source encoding technique using algebraic codes," in *Proc. Canadian Computer Conference*, 1972, pp. 201–213.

[4] K. C. Fung, S. Tavares, and J. M. Stein, "A comparison of data compression schemes using block codes," in *Proc. IEEE Int. Electrical Electronics Conf.*, October 1973, pp. 60–61.

[5] T. C. Ancheta, "Syndrome-source-coding and its universal generalization," *IEEE Transactions on Information Theory*, vol. 22, no. 4, pp. 432–436, July 1976.

[6] G. Caire, S. Shamai, and S. Verdú, "Almost-noiseless joint source-channel coding-decoding of sources with memory," in *Proc. 5th International ITG Conference on Source and Channel Coding*, January 2004, pp. 295–304.

[7] M. Fresia, F. Pérez-Cruz, and H. V. Poor, "Optimized concatenated LDPC codes for joint source-channel coding," in *Proc. IEEE Int. Symposium on Information Theory*, Seoul, South Korea, 2009.

[8] T. J. Richardson and R. L. Urbanke, "Efficient encoding of low-density parity-check codes," *IEEE Transactions on Information Theory*, vol. 47, pp. 638–656, Feb. 2001.

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

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

[11] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Trans. Communications*, vol. 49, pp. 1727–1737, Oct. 2001.

[12] T. Richardson, M. Shokrollahi, and R. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.

[13] H. V. Beltrão Neto, W. Henkel, and V. C. da Rocha Jr., "Multi-edge type unequal error protecting low-density parity-check codes," in *Proc. IEEE Information Theory Workshop (ITW '11)*, Paraty, Brazil, Oct. 2011.