Information shortening for joint source-channel coding schemes based on low-density parity-check codes

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 propose a simple shortening algorithm for joint source-channel coding schemes based on low-density parity-check codes. Due to finite-length effects, such systems present high error floors when optimised for low-entropy binary symmetric sources and transmission over an AWGN channel. To mitigate such effects, we propose the reduction of the compression rate by inserting infinite reliability variable nodes which form cycles with length equal to the girth of the code used as source compressor. Simulation results show that such a strategy is able to efficiently trade a slight reduction of the compression rate for a significant performance enhancement.**

I. Introduction

Following the idea of joint source-channel coding based on the application of low-density parity-check (LDPC) codes for source and channel coding introduced in [1], we proposed in [2] a novel LDPC-based joint source-channel (JSC) coding scheme and developed an optimisation algorithm for such systems based on a multi-edge-type analysis [4]. Simulations in [2] and [3] showed a significant enhancement of the performance in comparison with the results presented in [5] for the LDPC-based JSC system of [1]. Nevertheless, a relatively high error floor was still present in the simulations we presented for a low-entropy (0.19 bits) binary symmetric memoryless source and medium block lenghts ($n = 3200$ bits).

Further investigations indicated that the main cause of such high error floors were the short cycles present in the Tanner graph corresponding to the optimised joint system. For medium block lengths and low entropy sources, it is not possible to construct Tanner graphs with large girth following the degree distributions optimised for the LDPC-based joint source-channel proposed in [2]. This occurs due to the fact that for low-entropy memoryless binary symmetric sources, the optimised check node degree distributions are very concentrated around high degrees, which results in a Tanner graph with a high number of edges.

A straightforward approach to cope with this problem is to increase the length of the source output. However, since we are interested mainly in a non-asymptotic scenario, we decided to follow a different strategy. Specifically, we propose an algorithm based on information shortening to mitigate the effect of short cycles in the joint system performance. Information shortening (also known in the literature as symbol doping) is a technique usually applied in the context of rate-compatible LDPC codes and consists of placing infinite reliability on some systematic variable nodes of the system's Tanner graph. This is accomplished by fixing the value of those nodes and sharing the information about their values and positions with the decoder. Note however that there is no need for an extra channel to share this information, since it can be agreed between the encoder and decoder prior to the transmission.

In what follows, we briefly review the LDPC-based JSC systems we consider herein, the multi-edge notation applied in the analysis of such systems, and present some multi-edge degree distributions optimised for the compression of low-entropy memoryless binary symmetric sources and posterior transmission over an AWGN channel. Afterwards, we describe the proposed shortening algorithm and present some simulation results.

II. LDPC-based joint source-channel coding system

For the sake of clarity, we start describing the LDPC-based JSC coding system introduced in [1] and the modifications we proposed in order to cope with the high error floors present in its error-rate curves. Consider a JSC coding system formed by the serial concatenation of two LDPC codes. The outer encoder performs source compression computing the syndrome $\mathbf{s}$ corresponding to the source output $\mathbf{u}$. The inner code is a systematic channel code utilized prior to the transmission of the compressed sequence through a noisy channel. That is, for this system, a codeword $\mathbf{c}$ is defined as

$$\mathbf{c} = \mathbf{s} \cdot \mathbf{G}_{cc} = (\mathbf{u} \cdot \mathbf{H}_{sc}^T) \cdot \mathbf{G}_{cc} \,, \tag{1}$$

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. The factor graph defined by such an encoding scheme is depicted in Fig. 1. The variable and the check nodes of the source LDPC (left) represent the source output and the compressed source sequence,

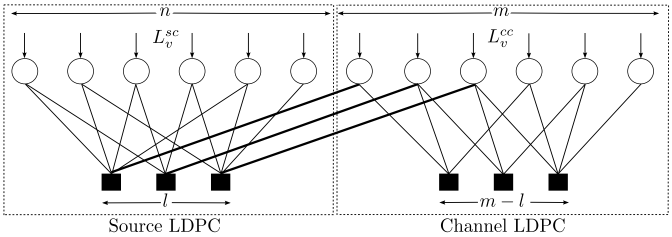

Fig. 1. Joint source-channel factor graph.

respectively. Since we are only considering binary symmetric sources, the variable nodes represent binary symbols. Note that only m variable nodes are transmitted, which means that the overall rate is n/m. Furthermore, L_v^{sc} and L_v^{cc} represent the intrinsic information (in form of log-likelihood ratio - LLR) available to the source ($v = 1, \ldots, n$) and channel ($v = n+1, \ldots, n+m$) variable nodes, respectively.

Simulation results for the system depicted in Fig. 1 [5] 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 correspond to error patterns that cannot be corrected by the outer LDPC code. In [2], we presented an idea of coping with this problem by providing an extra amount of extrinsic information to the variable nodes of the source LDPC code. The way we envisaged to achieve this is equivalent to the insertion of extra edges to the JSC system factor graph. This idea is depicted in Fig. 2, where the extra edges are represented by dashed lines.

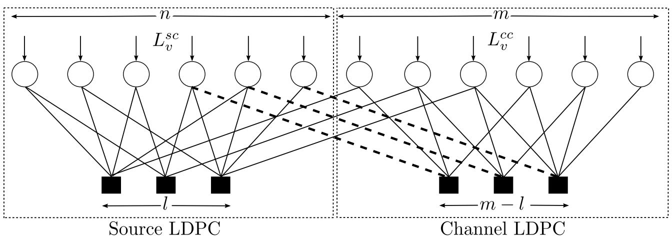

Fig. 2. Joint source-channel factor graph with inserted edges.

The first problem that arises with the configuration of Fig. 2 concerns the encoding algorithm. Note that as opposed to the system of Fig. 1, one cannot perform the serial encoding as described by Eq. (1) since in the extended system of Fig. 2, not only the compressed bits but also the source output must be considered for channel encoding.

In order to understand how to perform encoding for the system of Fig. 2, consider its representation as a $m \times (n+m)$ incidence matrix $\mathbf{H}$. This matrix can be written as

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

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. In [2], we showed that a codeword of the system in Fig. 2 can be defined as

$$\mathbf{c} = [\mathbf{u}, \mathbf{s}] \cdot \mathbf{G}_L = [\mathbf{u}, \mathbf{u} \cdot \mathbf{H}_{sc}^T] \cdot \mathbf{G}_L \ , \qquad (2)$$

where $[\mathbf{u}, \mathbf{s}]$ is the concatenation of the source output $\mathbf{u}$ and its syndrome $\mathbf{s}$ computed by the source code, and $\mathbf{G}_L$ 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 summary, the encoding algorithm of the JSC system (Fig. 2) has the following steps:

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. Note that since $\mathbf{H}_{sc}$ is sparse, the source encoding has a complexity that is linear with respect to the block length.

Once the encoding algorithm is set, it is necessary to determine how to construct the linking matrix $\mathbf{L}$ and how to optimize the matrices $\mathbf{H}_{sc}$ and $\mathbf{H}_{cc}$ aiming at achieving the lowest possible error floor. In order to cope with this problem, we proposed a joint optimization algorithm for such matrices in [2] making use of the multi-edge framework presented in the following section.

III. Multi-edge framework

Multi-edge-type LDPC codes [4] are a generalization of irregular and regular LDPC codes where the graph connectivity is constrained not 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

$$\nu(\mathbf{r}, \mathbf{x}) = \sum \nu_{\mathbf{b},\mathbf{d}} \mathbf{r}^{\mathbf{b}} \mathbf{x}^{\mathbf{d}} \quad \text{and} \quad \mu(\mathbf{x}) = \sum \mu_{\mathbf{d}} \mathbf{x}^{\mathbf{d}}, \qquad (3)$$

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 $\mathbf{r} = (r_0, \dots, r_{m_r})$, which represents the different received distributions and the vector $\mathbf{b} = (b_0, \dots, b_{m_r})$, which indicates the number of connections to the different received distributions. We use $\mathbf{x^d}$ to denote $\prod_{i=1}^{m_e} x_i^{d_i}$ and $\mathbf{r^b}$ 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 $\mathbf{d}$, respectively.

Within this framework, an LDPC-based JSC system can be seen as the combination of the Tanner graphs of two LDPC codes: one used for source compression and the other for channel coding. In such a graph, we define four edge types, i.e., $m_e = 4$. Additionally, now we also have two different received distributions corresponding to the source statistics and channel information, respectively. Figure 3 depicts the Tanner graph of an LDPC-based JSC system where the source output and the transmitted sequence have lengths n and m, respectively. It has 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.

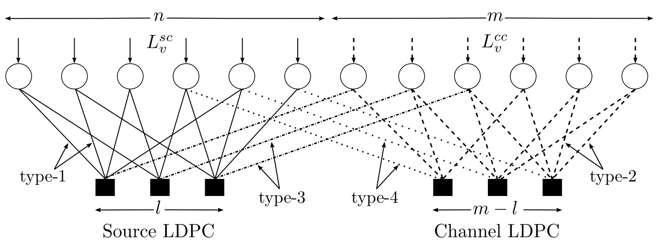

Fig. 3. Multi-edge joint source-channel factor graph.

IV. Optimised multi-edge degree distributions

In [2], we presented an optimisation strategy for the component codes of an LDPC-based JSC system following a multi-edge framework. Therein, assuming transmission over a binary input AWGN channel and a binary symmetric source, we constructed LDPC-based JSC systems showing that our optimization strategy was able to lower the error floor present in the simulation of the LDPC-based JSC systems introduced in [1] and investigated in [5].

In order to depict the advances previously obtained with the system proposed in [2], we present here a set of simulations for the systems proposed in [1] and the LDPC-based JSC systems optimised following the strategy of [2]. We will refer to such systems as System I and II, respectively. For such simulations, we assumed transmission over a binary AWGN channel with capacity $C = 0.5$ bits/channel use and binary symmetric sources with three different entropies ($H(S) = 0.28$, 0.40 and 0.49 bit).

Table I lists the overall rates R_{over} of the simulated codes. On the one hand, it is true that there is still a significant gap to the asymptotically achievable JSC capacity (listed in bold). On the other hand, the LDPC-based JSC systems proposed in [2] present a performance significantly better than the systems presented in [5] as can be seen in Fig. 4.

TABLE I
Overall rates (bits/channel use) of the LDPC-based JSC systems constructed following the optimization algorithm of [2]. In bold, the asymptotic system capacity $C/H(S)$ in bits/channel use.

$H(S)$	0.28	0.40	0.49
R_{over}	1.38 **(1.74)**	1.03 **(1.24)**	0.84 **(1.00)**

The results depicted in Fig. 4 were obtained for a source output with length $n = 2048$ bits and 50 belief-propagation decoding iterations.

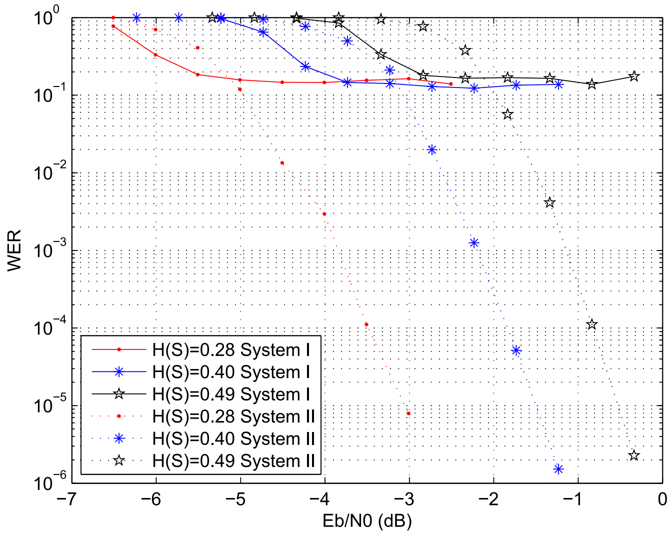

Fig. 4. Performance comparison between the LDPC-based JSC systems proposed in [1] and [2] for three different source entropies.

In spite of the good results presented in Fig. 4, it was still possible to detect error floors in the performance of the JSC systems when the compressed output of binary symmetric sources with $H(S) = 0.19$ are transmitted through AWGN channels with capacity 0.6 bits/channel use as depicted in Fig. 5. As pointed out in the introduction, preliminary simulations indicated that the main cause of such high error floors were the short cycles present in the Tanner graph corresponding to the optimised joint system for medium block lengths and low-entropy sources.

In order to further investigate such effect, we applied the algorithm previously proposed in [2] to optimise the degree distributions of an LDPC-based JSC system considering a binary input AWGN channel with capacity $C = 0.6$ as transmission channel and memoryless binary symmetric sources with entropies $H(S) = 0.14$, $H(S) = 0.19$, and $H(S) = 0.28$ (which correspond to a binary symmetric source with the probability of emitting a one $p_v = 0.02$, 0.03, and 0.05, respectively).

Let $\mathbf{d_v}$ and $\mathbf{d_c}$ be the maximum variable and check

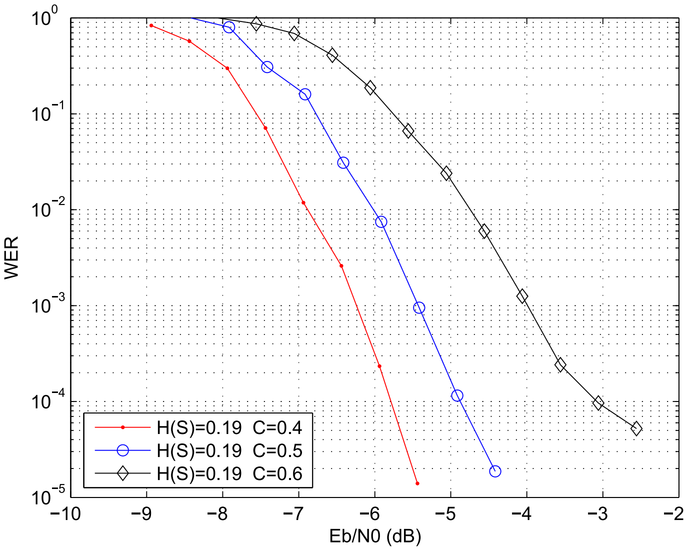

Fig. 5. Binary symmetric source with $p_v = 0.03$ and output length $n = 4096$ bits.

node degree vectors, respectively. We define those vectors as $\mathbf{d_v} = (d^{(1)}_{v_{max}}, d^{(2)}_{v_{max}}, d^{(3)}_{v_{max}}, d^{(4)}_{v_{max}})$ and $\mathbf{d_c} = (d^{(1)}_{c_{max}}, d^{(2)}_{c_{max}}, d^{(3)}_{c_{max}}, d^{(4)}_{c_{max}})$, where $d^{(j)}_{v_{max}}$ and $d^{(j)}_{c_{max}}$ represent the maximum number of type-j edges connected to variable and check nodes, respectively. In our optimisation algorithm, for a given source entropy and channel capacity, we fix the vector $\mathbf{d_v}$ and adjust the vector $\mathbf{d_c}$ in order to obtain the maximum overall rate R_{over} under the optimisation constraints. Table II shows the resulting maximum check node degree vector and overall rate for the considered source entropies. A detailed description of the optimisation algorithm is out of the scope of this paper but can be found in [2] and [3].

TABLE II
MAXIMUM VARIABLE AND CHECK NODE EDGE DEGREE VECTORS FOR BINARY SYMMETRIC SOURCE WITH DIFFERENT ENTROPIES.

Source Entropy	$\mathbf{d_v}$	$\mathbf{d_c}$	R_{over}
$H(S) = 0.14$ bits	(10,10,1,2)	(24,9,1,8)	3.27 bits/channel use
$H(S) = 0.19$ bits	(10,10,1,2)	(18,9,1,6)	2.38 bits/channel use
$H(S) = 0.28$ bits	(10,10,1,2)	(11,9,1,4)	1.66 bits/channel use

Note that the lower the source entropy, the higher are $d^{(1)}_{c_{max}}$ and $d^{(4)}_{c_{max}}$. Those high check node degrees are the reason behind the formation of a large number of short cycles in the Tanner graph of our LDPC-based JSC system for short medium block lengths. Information shortening can help to break such short cycles, since variable nodes with infinite reliability can be removed from the corresponding Tanner graph [8]. This can be better understood if we consider the check node update rule of the belief-propagation decoding algorithm of LDPC codes. Log-likelihood ratio values (L-values) with infinite reliability do not need to be considered, since they are equivalent to the neutral element of the check node update rule of the belief-propagation algorithm.

The application of shortening techniques to eliminate short cycles was first proposed in [6] and investigated in the context of rate-compatible LPDC codes in diverse publications (see [8] and references therein). In the next section, we propose a simple shortening algorithm and apply it to the LDPC-based JSC systems of [2].

V. SHORTENING ALGORITHM

Let $\mathbf{H}$ be the incidence matrix corresponding to the Tanner graph of an LDPC-based joint source-channel system. The information shortening strategy we propose herein can be stated as in Algorithm 1.

Algorithm 1 Shortening algorithm

Given the incidence matrix $\mathbf{H}$ of an LDPC-based JSC system and a desired number of shortened symbols d:

1) Find the length of the shortest cycle running through all variable nodes i, for $i = 1, \ldots, n$, where n is the length of the source output
2) Select all variable nodes which are part of a cycle with size equal to the girth of $\mathbf{H}$ as candidates for doping
3) If the number of doping candidates (ε) is smaller than d, randomly select $d - \varepsilon$ variable nodes among the $n - \varepsilon$ not yet in the list of candidates
4) Shorten all the candidate variable nodes.

One of the drawbacks of this algorithm is the high complexity of Step 1, since the complexity of cycle counting algorithms typically grows exponentially with the number of edges present in the graph. Nonetheless, the shortening algorithm can be executed offline, and the shortened positions can be stored together with the corresponding incidence matrix $\mathbf{H}$. Another disadvantage is a slight loss in the compression rate, since the shortened positions cannot be utilized to transmit useful information. Nevertheless, our simulations indicate that as few as 5% of the source variable nodes are enough to significantly lower the error floor caused by finite-length effects.

VI. SIMULATION RESULTS

Figure 6 shows the results achieved using our shortening strategy for the LDPC-based JSC system optimised for a binary symmetric source with entropy $H(S) = 0.19$ and two AWGN channels with capacity $C = 0.6$ and $C = 0.75$. All the codes were constructed assuming sources with outputs of length $n = 4096$ bits. In all simulations, we considered a total of 50 belief-propagation decoding iterations and that all the transmitted unit-energy signals were BPSK modulated.

Note that for $d = 205$ shortened positions (which represents only a 5% reduction in the overall rate, since $n = 4096$) there is already a significant error-floor reduction for both AWGN channels considered. Furthermore, for $d = 410$, the error floor was not detectable in our simulation range for $C = 0.6$. This shows that our shortening algorithm is able to efficiently trade a small loss in the compression rate for a significant reduction of the error floor present in the performance curves.

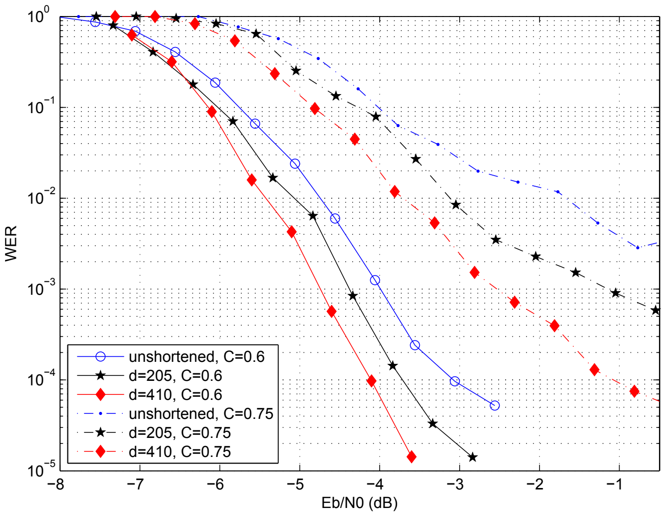

Fig. 6. Error-floor mitigation by means of the proposed shortening strategy for an LDPC-based JSC system optimised for a binary symmetric source with entropy $H(S) = 0.19$ bits ($p_v = 0.03$) and AWGN channels with capacities $C = 0.6$ and $C = 0.75$ bits/channel use.

VII. Concluding remarks

We presented a shortening algorithm aiming at lowering the error floor caused by the presence of short cycles in the Tanner graph of medium-length joint source-channel coding schemes based on low-density parity-check codes. Considering a low-entropy binary symmetric source and two AWGN channels with different capacities, we optimised and constructed two different LDPC-based joint source-channel coding schemes and applied the proposed shortening algorithm to them. Simulation results showed that the proposed algorithm was successful in lowering the error-floor of the considered LDPC-based JSC systems in exchange of a small reduction in the overall compression rate.

Acknowledgment

This work was funded by the German Research Foundation (DFG).

References

[1] G. Caire, S. Shamai, and S. Verdú, "Almost-Noiseless Joint Source-Channel Coding-Decoding of Sources with Memory," *Proc. 5th International ITG Conference on Source and Channel Coding*, 2004, pp. 295-304.
[2] H. V. Beltrão Neto and W. Henkel, "Multi-Edge Optimization of Low-Density Parity-Check Codes for Joint Source-Channel Coding," *Proc. of 9th International ITG conference on Systems, Communications and Coding*, Munich, Germany, Jan. 2013.
[3] H. V. Beltrão Neto, "Analysis and Design of Modern Coding Schemes for Unequal Error Protection and Joint Source-Channel Coding," Ph.D. dissertation, Jacobs University Bremen, School of Engineering and Science, 2012.
[4] T. Richardson and R. Urbanke, "Modern Coding Theory," *Cambridge University Press*, 2008.
[5] M. Fresia, F. Pérez-Cruz, H. V. Poor, and S. Verdú, "Joint Source/Channel Coding with Low Density Parity Check Matrices," *IEEE Signal Processing Magazine*, 2010, **27** (6), pp. 104-113.
[6] O. Milenkovic, N. Kashyap, and D. Leyba, "Shortened Array Codes of Large Girth,"*IEEE Trans. Inf. Th.*, **52** (8), pp.3707-3722, Aug. 2006.
[7] T. Tian and C. R. Jones, "Construction of rate-compatible LDPC codes utilizing information shortening and parity puncturing," *EURASIP J. Wirel. Commun. Netw.*, Oct. 2005.
[8] M. Beermann, and P. Vary, "Breaking Cycles with Dummy Bits: Improved Rate-Compatible LDPC Codes with Short Block Lengths," *Proc. of 9th International ITG conference on Systems, Communications and Coding*, Munich, Germany, Jan. 2013.