Non-Linear Graph-Based Codes for Joint Source-Channel Coding

David Matas[1], Meritxell Lamarca[1,2], and Javier Garcia-Frias[2]
[1]Signal Theory and Communications Department, Technical University of Catalonia
Email: (dmatas, xell)@gps.tsc.upc.edu
[2]Department of Electrical and Computer Engineering, University of Delaware
Email: (lamarca, jgarcia)@ee.udel.edu

***Abstract*—We study the behavior of a new family of non-linear graph-based codes, previously introduced for compression of asymmetric binary memoryless sources, for the joint source-channel coding scenario in which the codewords are transmitted through an additive white Gaussian noise channel. We focus on low entropy sources (with high redundancy) and compression rates. Monte Carlo simulation and density evolution results show that the proposed family, with a regular and simple parametrization of the degree profiles, outperforms linear codes.**

I. Introduction

Non-linear source codes are potentially more powerful than linear ones, since they include the latter as a particular case. The fact that non-linear codes may have different distance properties for different codewords can be exploited to guarantee better distance profiles for the most likely information sequences, whereas linear codes always possess identical distances profiles for all codewords. In spite of this potential advantage, there has been relatively little work in the literature on non-linear codes, since linear codes are known to be asymptotically optimum for infinite codeword length.

In [1] a new family of non-linear binary codes based on graphs was presented. They were observed to outperform linear codes for compression, considering a regular scheme, while maintaining many of the desirable features of low density parity check (LDPC) codes. Namely, i) the proposed non-linear codes can be graphically represented by means of a factor graph, ii) they can be decoded using belief propagation, and iii) their performance can be predicted using density evolution (DE), and thus they can be easily designed when long codewords are considered.

In this paper we study the application of the aforementioned non-linear codes to the problem of joint source-channel coding, where the the generated coded sequence is directly transmitted through an additive white Gaussian noise (AWGN) channel. The standard approach for the transmission of non-uniform sources through a noisy channel is to separate the encoding process in source and channel coding, and this process is optimal if complexity and delay are not an issue. However, when they are at stake the overall performance can be improved if the tasks of source and channel coding are blended together by means of a joint source-channel encoder. In this way, the joint decoder can employ some of the inherent redundancy of the source to alleviate the requirements of the channel encoder.

Previous work on this area includes techniques that utilize a standard linear channel code and exploit the source statistics in the decoding process [2], [3], [4], [5]. It also includes techniques that, besides decoding modifications, modify the encoder based on the source statistics [6], [7], [8], [9]. The use of non-linear encoders, as proposed here, leads to significant performance gain, especially for the case of very low entropies. In this case, the best existing linear schemes utilize asymmetric energy allocation, but are still far from the theoretical limits.

II. System set-up

We consider the problem of transmitting an asymmetric memoryless binary source with entropy H through an AWGN channel. We assume $p(1) > p(0)$ at the source bits and consider fixed-length block codes, where a sequence of K information bits, $b_1 b_2 \dots b_K$, is encoded into a codeword of N bits, so that it results a (compression) code rate $R_{co} = N/K$. Note that this magnitude is always greater than 1 for a channel code, which adds additional redundancy, but might be smaller for a joint source-channel code, which also compresses the information. We will focus in the latter. This scenario makes sense when the entropy of the source is such that, even if we compress the input sequencce, there is still redundancy left to protect the information from the errors introduced by the channel. This leads us to recall the theoretical limits for a reliable communication. The required condition to retrieve the source bits without error at the receiver is:

$$H/R_{co} < C \tag{1}$$

where C is the capacity of the channel.
We consider the problem of joint source channel coding when the coded bits are transmitted employing a Binary Phase-Shift Keying (BPSK) modulation through an AWGN channel with a given E_s/N_o and a corresponding energy per information bit over noise ratio (E_b/N_o) defined as:

$$E_b/N_o = \frac{E_s/N_o \cdot R_{co}}{H} .$$

This work has been partially funded by the Catalan/Spanish Science and Technology Commissions and by FEDER funds from the European Commission: 2009SGR-1236, TEC2007-68094-C02-02, CSD2008-00010.

978-1-4244-6746-4/10/$26.00 ©2010 IEEE

The capacity of this input constrained channel is well known and can be computed numerically. However, since we focus our study at low SNR values, we can rely on the Gaussian capacity. Then the limit for the E_b/N_o is

$$E_b/N_{o_{lim}} = \frac{R_{co}}{2H}(2^{2R_{co}\cdot H} - 1) \tag{2}$$

III. LDPC-LDPrC codes

A. Code description

We employ the new family of non-linear codes introduced in [1]: the hybrid LDPC-LDPrC codes, based on a generalized syndrome source coding approach. These codes are constructed as a parallel concatenation of two block codes: a fraction α of coded bits is generated by a non-linear Low Density Product Check (LDPrC) code and the reminder fraction, $1-\alpha$, by a linear LDPC code.

The linear block is encoded as in a standard LDPC code. Defining a generator matrix $\mathbf{G}$ of size $K \times N_2$, where $N_2 = (1-\alpha)N$, the encoding process can be expressed as:

$$\mathbf{c} = \mathbf{bG}, \quad \mathbf{c} = [c_1 \ldots c_{N_2}], \, \mathbf{b} = [b_1 \ldots b_K] \,, \tag{3}$$

For the LDPrC code, each coded bit, p_j, is obtained as the product (AND) of a few information bits b_i. Thus, we generate a codeword of length $N_1 = \alpha N$ as

$$p_j = \prod_{i \in S_j} b_i, \qquad j = 1 \ldots N_1,$$

where S_j is the set of d_{p_j} indices ($1 \leq d_{p_j} \leq K$) that defines which information bits are used to generate each product bit p_j.

Analogously to the LDPC code, the encoding process can be described in a compact form by defining a $K \times N_1$ generator matrix $\mathbf{P}$ and the $\boxdot$ operator:

$$\mathbf{p} = \mathbf{b} \boxdot \mathbf{P}, \;\; \mathbf{p} = [p_1 \ldots p_{N_1}], \;\; \mathbf{b} = [b_1 \ldots b_K]. \tag{4}$$

The LDPC-LDPrC codeword is built as $[\mathbf{p}\,\mathbf{c}]$. Therefore, matrices $\mathbf{G}$ and $\mathbf{P}$ fully characterize the hybrid LDPC-LDPrC code. Fig. 1 depicts the factor graph of this code, which has bit, product, and parity-check nodes. Note that they include both LDPC (setting $\alpha = 0$) and LDPrC ($\alpha = 1$) codes as particular cases.

Following the usual convention for graph-based codes, we characterize matrices $\mathbf{G}$ and $\mathbf{P}$ with the profiles corresponding to the degree of the bit nodes, $d_{b_i}^L$ and $d_{b_i}^{NL}$ (we specify two separate degrees corresponding to the edges going to the linear and non-linear check nodes, respectively), and the degrees of the parity check, d_{c_j}, and product check nodes d_{p_j}.

We focus our attention into a very simple case in which the degree of the information bit nodes is regular, d_b^L and d_b^{NL} for the LDPC and the LDPrC codes, respectively. Then, for a given code rate $R_{co} = N/K$ and parameter α, the average degree of the product and the check nodes must be equal to $d_p = d_b^{NL}/R_{co}\alpha$, $d_c = d_b^L/R_{co}(1-\alpha)$. Since these may result to be fractional, we will set the products/checks to have degrees equal to the closest lower and upper integers. Note that these codes are totally characterized by the set of parameters $(N, K, d_b^{NL}, d_b^L, \alpha)$.

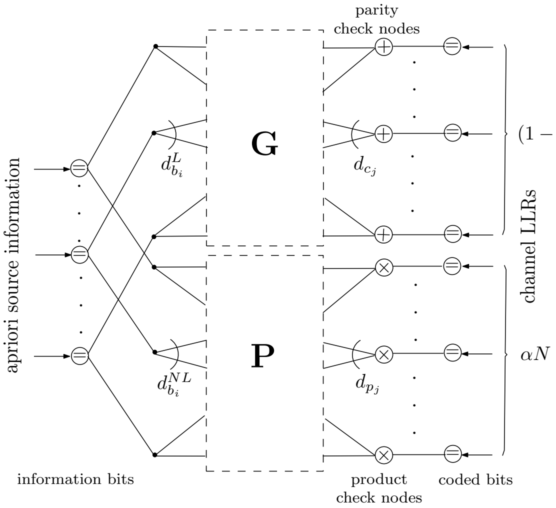

Figure 1. Factor graph of the proposed LDPC-LDPrC codes.

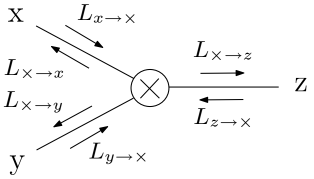

Figure 2. LLR messages exchanged in a product node where $z = x \cdot y$.

B. Decoding and Performance Analysis

The proposed codes are constructed using sparse matrices $\mathbf{P}$ and $\mathbf{G}$. Hence, if the codeword is long enough and the matrix has been properly designed, there will be few cycles in the graph and belief propagation will provide a quite accurate approximation of maximum-a-posteriori decoding. The message passing equations for the information bit nodes and parity check nodes will be the same as in LDPC codes, whereas new equations are required for the product nodes.

The decoding equations for the log-likelihood ratios (LLR) of a product node of degree two are [1] (see Fig. 2 for variable definition):

$$L_{\times\to y} = log\left(\frac{1+2e^{L_{x\to\times}+L_{z\to\times}}}{1+2e^{L_{x\to\times}}}\right) \tag{5}$$

$$L_{\times\to z} = L_{x\to\times} + L_{y\to\times} - log\left(1+e^{L_{x\to\times}}+e^{L_{y\to\times}}\right) \tag{6}$$

For product nodes of higher degree, the messages can be computed recursively from the expressions above.

As indicated in [1], the performance of hybrid LDPC-LDPrC codes for source coding with belief propagation decoding can be accurately predicted by density evolution (DE). However, the standard DE procedure had to be modified to take into account the fact that, due to the code non-linearity, the performance depends on the information message, and thus the analysis cannot be made assuming an all-zero sequence. The generalized DE equations for the analysis of hybrid LDPC-LDPrC in the source coding scenario were presented in [1]. In this paper, DE has also been employed to predict the performance of these codes in the joint source-channel coding scenario. The same equations presented in [1] are valid in this case, the only difference is that we introduce the channel probability density function as the imput of the parity and product check nodes. The results obtained by DE in this case also provide accurate predictions of the performance of these codes. Figure 3 depicts the performance of the hybrid code $(d_b^{NL}, d_b^L, \alpha) = (10, 20, 0.4)$ and codeword length $N = 40000$ bits and compares it with the BER predicted by DE. The results are quite accurate, the differences can be attributed to the finite codeword length and to the fact that random matrices were not optimized to avoid cycles in the graph. Taking into account this accuracy, all simulations in sections IV and V have been obtained using DE.

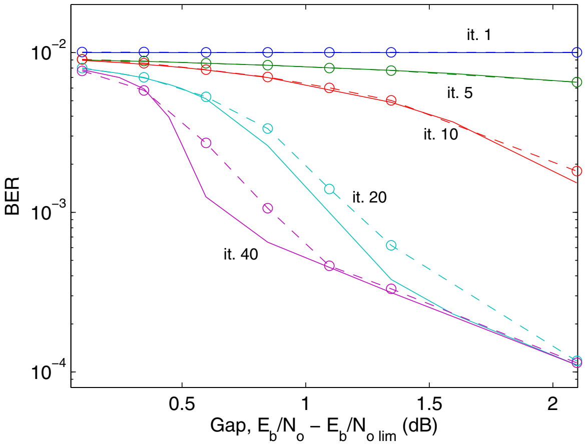

Figure 3. BER vs E_b/N_o gap top the limit for the regular LDPC-LDPrC code $(d_b^{NL}, d_b^L, \alpha) = (10, 20, 0.4)$ for iterations $1, 5, 10, 20$ and 40 comparing DE with Monte Carlo simulation results, with continous and dashed lines respectively.

IV. Simulation Results

A. LDPrC codes performance

In [1] it was observed that, for source coding, the BER of LDPrC codes decreases very slowly with the source entropy. Although the slope of the curve BER vs. entropy increased with the degree of the product nodes, it was still too small and LDPrC codes did not exhibit the classic threshold/waterfall behavior of LDPC codes. This fact was atributed to the specific characteristics of the AND node. As also discussed in that work, the performance could be significantly improved concatenating in parallel LDPrC with LDPC codes in an hybrid architecture, outperforming linear codes. Figure 4 depicts the BER vs E_b/N_o performance of LDPrC codes of rate 0.5 and diverse bit degrees for the compression of a source with $p(0) = 0.01$ (an entropy of 0.0808). As shown in the figure, the performance of these codes in the joint source-channel coding scenario is also very poor: the slope of the curve is very small even for large E_b/N_o gaps, although it increases with the degree of the bit node.

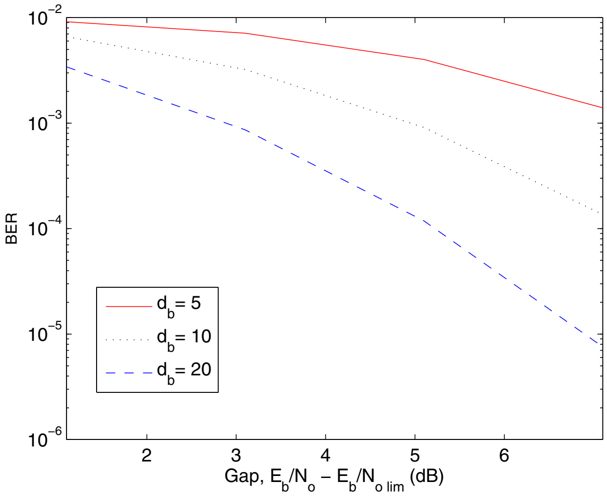

Figure 4. BER vs gap to the E_b/N_o limit for different LDPrC codes of compression rate 0.5 and bit degree d_b when the source has entropy 0.0808.

B. LDPC codes performance

As indicated in the introduction, the use of LDPC codes for the joint source-coding problem has been analyzed by different researchers [10], although to the best of authors knowledge the work in the literature has focused in the case of codes with rates of 1 or larger. As the simulations in this paper focus in the case of code rate 0.5, Figure 5 illustrates the performance of regular LDPC codes of rate 0.5 with different degrees when they are employed to compress a source with entropy 0.0808. A tradeoff between convergence and error correcting capability can be observed depending on the degree: the higher the degree the lower the error floor but the higher the convergence threshold. In all cases the gap to the theoretical limit is above 1dB, and it is much larger when the error floor gets lower. In the next section it is shown that the combination of LDPrC and LDPC codes can outperform these LDPC codes.

C. Hybrid LDPC-LDPrC codes performance

In this section the performance of hybrid LDPC-LDPrC codes has been evaluated and compared to that one of LDPrC and LDPC codes. We have focused in the compression of a source with entropy 0.0808, since it is well known that linear code performance degrades for low entropy sources and LDPrC codes are expected to have a high compression capacity thanks to the use of the AND gates. An exhaustive search has been carried out over the optimum values of d_b^{NL}, d_b^L, and α. Some of the best codes are depicted in Figure 6, where each code is described by its parameters $(d_b^{NL}, d_b^L, \alpha)$.

The first conclusion that can be drawn from this figure is that all codes with good performance have large degrees. This is due to the fact that they are operating at very low E_b/N_o

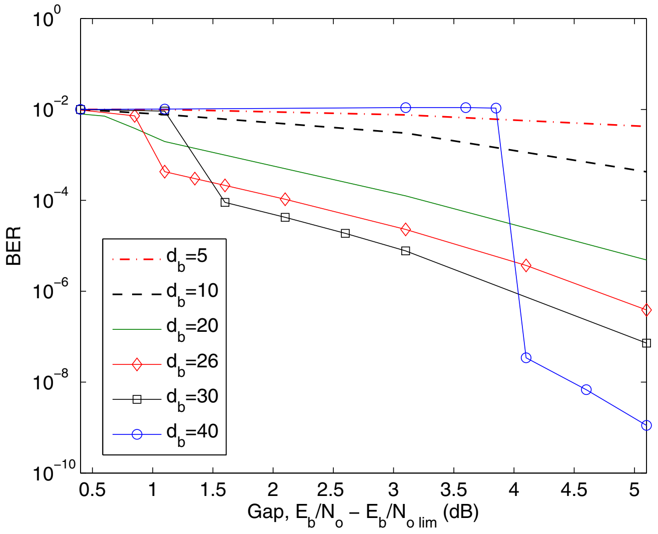

Figure 5. BER vs E_b/N_o for LDPC codes of compression rate 0.5 and different bit degrees d_b when the source has entropy 0.0808.

(the E_b/N_o limit for a source of $H = 0.08$ and code rate 0.5 is -1.1dB).

It is also worth mentioning that in the joint source coding scenario the fraction of coded bits generated by the non-linear code (α) is reduced with respect to values that provided good performance in the source coding scenario. The codes in [1] achieved good performance for α values close to 0.6, whereas now the optimum values for the codes depicted are around 0.4. This difference can be attributed to the behavior of the AND operator: while in the source coding problem the knowledge that a coded bit was '1' provided perfect knowledge of all input bits to that product, in joint source-channel coding the reliability of the received bits is reduced, and this hinders the compression capability of the non-linear code.

Figure 6 depicts the codes with best performance for $d_b^L = 18, 20$ and 22. For each degree in the linear code, three or four codes corresponding to different α values are depicted, and in all cases a search was made over the codes with different non-linear degrees to select that one with best performance among those with the same linear degree and α. As shown in the figure, the same behavior observed for LDPC codes is reproduced here: a larger value of d_b^L results in a smaller error floor and a higher convergence threshold. Among those codes with the same linear degree, the modification of α allows to slightly change the trade-off between the error floor and the convergence threshold. Finally, it is also worth mentioning that the optimum non-linear degree depends on the value of the linear degree: the higher it is d_b^L the lower it is d_b^{NL} (so the total degree of the bit node is kept approximately constant).

The hybrid codes depicted in Figure 6 have a gap that is significantly lower than that of the LDPC codes in Figure 5: convergence is achieved for gaps between 0.4 and 0.7 dB. However, the error floor of these codes is very high. It is expected that an irregular design of the degree profiles in the hybrid code would allow to obtain a better trade-off between the error floor and the convergence threshold. However, as illustrated in the next section, this error floor can also be reduced, while trying to maintain the regular architecture, by connecting the bit nodes to a few other check nodes with high degree, i.e. by concatenating the hybrid LDPC-LDPrC code with a high rate linear code.

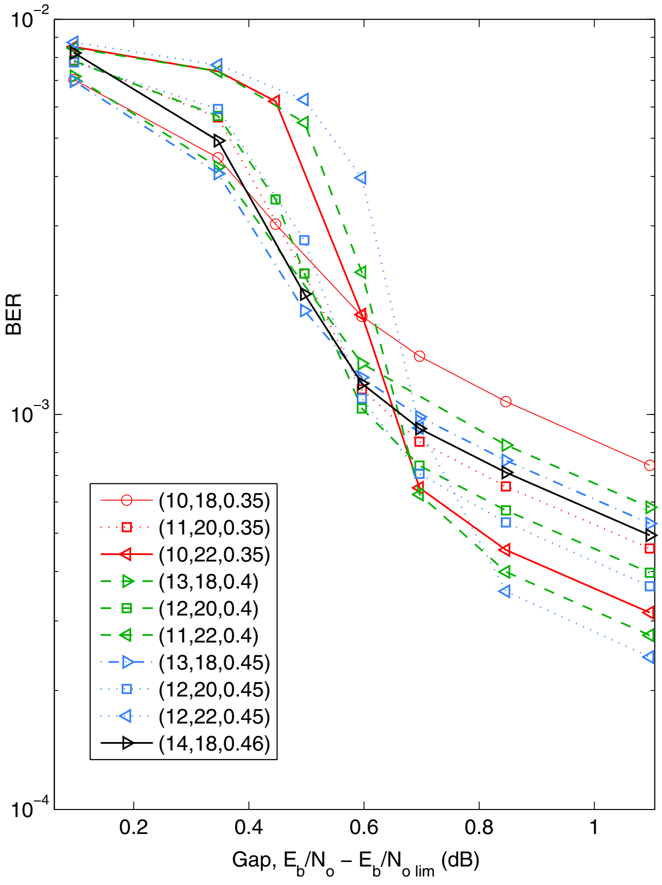

Figure 6. BER vs E_b/N_o gap top the limit for different regular LDPC-LDPrC codes specified as $(d_b^{NL}, d_b^L, \alpha)$ when the source has entropy 0.0808.

V. Error floor reduction by parallel concatenation

With the aim of reducing the high error floor seen in the previous section, we add another linear block code in parallel to the hybrid LDPC-LDPrC code, obtaining the factor graph represented in Figure 7. The codeword length is increased with the bits associated to the new parity check nodes.

Our purpose is to reduce the error floor of the hybrid LDPC-LDPrC code while preserving its good convergence behavior. In order to obtain a low error floor large bit node degrees are required in the new code $\mathbf{G}'$. In order to avoid degrading the convergence threshold the rate of the new code must be kept low, because for a fixed E_b/N_o per information bit the total power allocated to the hybrid code decreases when the rate of $\mathbf{G}'$ increases, and this degrades its convergence threshold.

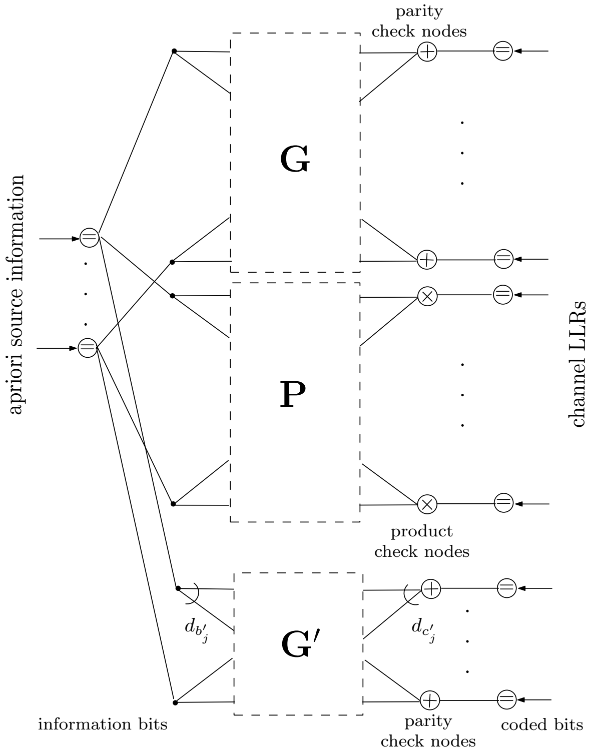

Figure 7. Factor graph of the parallel concatenation of an hybrid LDPC-LDPrC code with a linear LDPC block.

This effect is shown in Figure 8. We use the LDPC-LDPrC code with parameters $(d_b^{NL}, d_b^L, \alpha) = (10, 20, 0.4)$ and concatenate in parallel an LDPC code of rate r_L with regular bit and check degrees, d'_b and d'_c respectively. Assuming a fixed check degree equal to 250, we vary the rate and observe how the E_b/N_o gap increases with the code rate. On the contrary, lowering the code rate naturally lowers the bit degree and therefore results in a higher error floor.

The performance of two codes with the same bit degree ($d'_b = 25$) is also shown, being better the one with lower rate. Note that the degree of the parity check nodes takes very large values in this case.

The simulations in Figure 8 evidence that very good performance can be achieved by the architecture depicted in Figure 7 even if each stage is designed using regular codes: the source of entropy 0.0808 can be encoded with a compression rate of 0.54 with only 0.9dB of gap and a BER of 10^{-5}.

VI. Conclusion

We have studied the performance of the non-linear hybrid LDPC-LDPrC codes with compression rates for the transmission of highly redundant binary memoryless souces through an AWGN channel. Simulation results confirm that the resulting performance is better than that of linear codes, but still subject to high error floor due to the degradation introduced by the channel. A trade-off between the error floor and the convergence threshold to approach capacity has been observed and linked to the degree profile definition. The error floor can be greatly improved by parallel concatenation of a high rate LDPC code, obtaining codes with very good performance even if codes with regular degrees are employed.

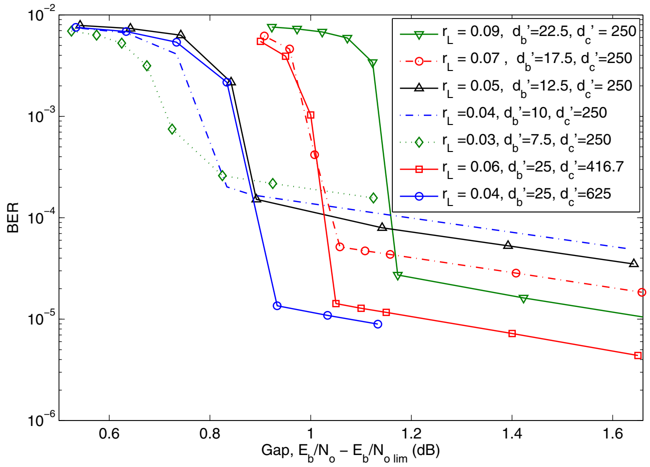

Figure 8. BER vs E_b/N_o for the concatenation of the (0.4, 10, 20) hybrid LDPC-LDPrC code with different linear blocks.

The behavior of the proposed structure can be accurately predicted using an extension of density evolution. Thus, we expect to be able to apply density evolution to design even better LDPC-LDPrC codes by incorporating irregular degree profiles and stage concatenation.

References

[1] D. Matas, M. Lamarca, and J. Garcia-Frias, "Non-linear graph-based codes for source coding," *Information Theory Workshop (ITW)*, Taormina, Italy, October 2009.

[2] J. Hagenauer, "Source controlled channel decoding," *IEEE Trans. on Communications*, pp. 2449-2457, 1995.

[3] J. Garcia-Frias and J. Villasenor, "Combining Hidden Markov Source Models and Parallel Concatenated Codes," *IEEE Communication Letters*, pp. 111-113, July 1997.

[4] J. Garcia-Frias and J. Villasenor, "Joint Turbo Decoding and Estimation of Hidden Markov Sources," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 9, pp. 1671-1679 September 2001.

[5] J. Garcia-Frias and Y. Zhao, "Compression of Binary Memoryless Sources using Punctured Turbo Codes," *IEEE Communication Letters*, pp. 394-396, September 2002.

[6] G.C. Zhu and F. Alajaji, "Turbo Codes for Non-Uniform Memoryless Sources over Noisy Channels," *IEEE Communications Letters*, pp. 64-66, February 2002.

[7] G. C. Zhu, F. Alajaji, J. Bajcsy, and P. Mitran, "Transmission of Nonuniform Memoryless Sources via Nonsystematic Turbo Codes," *IEEE Trans. Communications*, vol. 52, no. 8, pp. 1344-1354, August 2004.

[8] F. Cabarcas, R. D. Souza, and J. Garcia-Frias, "Source-Controlled Turbo Coding of Non-Uniform Memoryless Sources Based on Unequal Energy Allocation," *IEEE International Symposium on Information Theory*, June 2004.

[9] F. Cabarcas, R. D. Souza, and J. Garcia-Frias, "Turbo Coding of Strongly Non-Uniform Memoryless Sources with Unequal Energy Allocation and PAM Signaling," *IEEE Trans. on Signal Processing*, vol. 54, no. 5, pp. 1942-1946, May 2006.

[10] G. Caire, S. Shamai, and S. Verdu, "Almost-Noiseless Joint Source-Channel Coding-Decoding of Sources with Memory," *Proc. of 5th International Conf. on Source and Channel Coding*, pp. 295-302, January 2004.