# Non-Systematic LDPC Codes for Redundant Data

Gil I. Shamir, Joseph J. Boutros, Amira Alloum, and Li Wang

***Abstract*— Non-systematic channel encoding can be superior to systematic encoding in the presence of redundancy in the transmitted data. We consider classes of non-systematic *low-density parity-check* (LDPC) codes based on *scrambling* or *splitting* redundant data bits into coded bits. Scrambling and splitting are achieved by cascading a sparse matrix or an inverse of a sparse matrix, respectively, with an LDPC code. Such codes exhibit excellent performance in the presence of redundancy in the transmitted data, which is far superior to that of systematic LDPC codes. We study the theoretical limits of such codes, and present a *density evolution* (DE) method to find the threshold values of splitting based codes. We show that the advantage of these codes is even more significant for high channel rate transmission. Simulations, supporting the results, are presented.**

## I. Introduction

In many channel coding applications, redundancy is left in channel coded data (see, e.g., [10] and references therein). Our goal is to design channel codes whose structure allows best utilization of this redundancy. As shown in [10] for turbo codes, non-systematic encoding, in which the transmitted codeword does not contain duplications of the bits of the original message, is superior to standard systematic encoding in such scenarios. The reason is that with non-systematic encoding, the set of *typical* data sequences can be better mapped into the code space. In other words, non-systematic codes still allow attaining the capacity achieving distribution of the channel, whereas systematic encoding forces a constraint on the channel input distribution, leading to a distribution that is not the capacity achieving one. The situation becomes even more extreme as the channel code rate increases.

Non-systematic LDPC-like codes were first proposed in a pioneering work by MacKay and Neal [5]. These codes were later referred to as MN codes. In this paper, we summarize some results on a new family of non-systematic LDPC based codes, which we recently proposed in [7], and continue to study this family. The family proposed in [7] was wide and allowed various code configurations, one of which yields MN codes. However, we have focused (see, e.g., [1], [8], [9]) on two particular configurations: *scramble-LDPC* and *split-LDPC* codes. These codes consist of a pre-coding *scrambler* or *splitter*, respectively, concatenated by a standard systematic LDPC code. A scrambler is a low density square matrix that scrambles the message bits. A scramble-LDPC code transmits the scrambled bits and parities generated by systematic encoding of the LDPC code on the scrambled bits. The decoder combines the equations of the scrambler and the LDPC code into one decoding graph. A split-LDPC code is similar to a scramble-LDPC code, except that it uses an inverse of a sparse square matrix to initially *split* the message bits. To the best of our knowledge, the split-LDPC code structure yields the best performance in presence of redundancy. This paper presents the structure of scramble-LDPC and split-LDPC codes. We then study capacity bounds on these code structures, and present a *density evolution* (DE) method, first presented in [1], to find threshold values of split-LDPC codes. Finally, we turn some attention to high rate codes from these families [8].

The paper is organized as follows: In Section II, we define the system. Section III describes the structure of scramble-LDPC and split-LDPC codes. Next, Section IV contains a comparative study of the best achievable mutual information by the different code structures considered. Next, Section V describes DE for split-LDPC codes. Finally, Section VI contains simulation results including results for high rate codes.

## II. System Description and Notation

Let $\mathbf{s} \stackrel{\triangle}{=} s_1^K \stackrel{\triangle}{=} (s_1, s_2, \ldots, s_K)^T$; $s_i \in \{0, 1\}$, be a bit sequence of length $K$. The superscript $T$ denotes the transpose operator. Assume that $\mathbf{s}$ is generated by some i.i.d. source that generates 1 with probability $\pi_1$ and 0 with probability $\pi_0 = 1 - \pi_1$, and has entropy $H_s = -\pi_0 \log \pi_0 - \pi_1 \log \pi_1$. If $\pi_0 \neq 1/2$, $\mathbf{s}$ contains redundancy. Then, $\mathbf{s}$ is encoded non-systematically with a code of rate $R_c = K/N$ into $\mathbf{c} \stackrel{\triangle}{=} (c_1, c_2, \ldots, c_N)^T$ of length $N > K$. The code sequence $\mathbf{c} = [\mathbf{u}^T | \boldsymbol{\vartheta}^T]^T$ consists of $K$ pre-coded bits $\mathbf{u}$ and $N - K$ parity bits $\boldsymbol{\vartheta}$. The vector $\mathbf{c}$ is BPSK modulated to the vector $\mathbf{x}$, that is transmitted over an AWGN channel with spectral density $N_0/2$, and received as the noisy vector $\mathbf{y}$. We will use $E_{br}$ to denote the average energy per (redundant) data bit $s_i$. The decoder, which receives $\mathbf{y}$, estimates $\mathbf{s}$ utilizing $\pi_1$.

## III. Scramble-LDPC and Split-LDPC Codes

Let $\mathbf{A}$ be a randomly generated sparse matrix of dimensions $K \times K$. For a regular scrambler or splitter, $\mathbf{A}$ has row and column weight $d_s$. A scramble-LDPC encoder first encodes (scrambles) the source vector $\mathbf{s}$ into $\mathbf{u}$ by $\mathbf{u} = \mathbf{A}\mathbf{s}$. Then, the vector $\mathbf{u}$ is encoded by a systematic LDPC generator matrix $\mathbf{G}$ of dimensions $N \times K$ to the code vector $[\mathbf{u}^T | \boldsymbol{\vartheta}^T]^T = \mathbf{G}\mathbf{u}$. The vector $\boldsymbol{\vartheta}$ is the parity vector. For a regular LDPC code, the parity-check matrix $\mathbf{H}$ has column weight $d_b$ and row weight $d_c$. A split-LDPC encoder is very similar, except that the scrambling operation is replaced by splitting performed by

[1]G. Shamir and L. Wang are with Department of Electrical and Computer Engineering, University of Utah, Salt Lake City, UT 84112, U.S.A., e-mails: gshamir@ece.utah.edu, liw@eng.utah.edu. J. Boutros is with Communications and Electronics Department, ENST Paris, 46 Rue Barrault, 75634 Paris, FRANCE, e-mail: boutros@comelec.enst.fr. A. Alloum is with France Telecom R&D, 92130 Issy-Les-Moulineaux, France, e-mail: amira.alloum@francetelecom.com. The work of G. Shamir and L. Wang was supported by NSF Grant CCF-0347969.

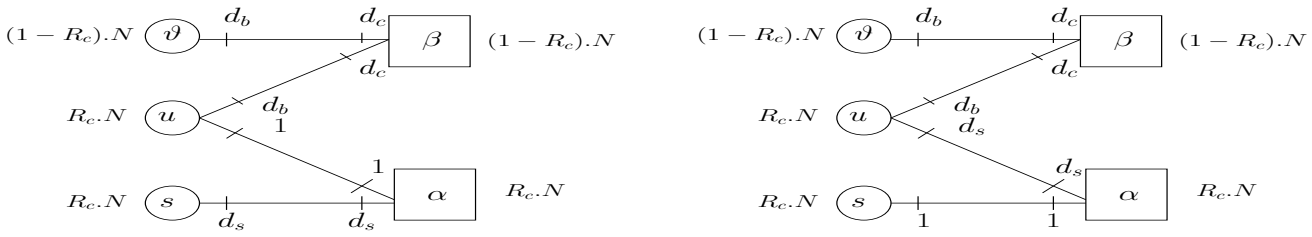


Fig. 1: Graph structures for non-systematic cascade of regular $d_s$ degree scrambler (left) and splitter (right) with a regular $(d_b, d_c)$ binary LDPC code. Scrambler/splitter check nodes are denoted by $\alpha$, and LDPC check nodes by $\beta$. Similar graph representations are valid for scrambler/splitter cascaded with an irregular $(\lambda(x), \rho(x))$ LDPC code.

$\mathbf{u} = \mathbf{A}^{-1}\mathbf{s}$. The term scrambler is used since the operation of a sparse square matrix on incoming bits can be viewed as if bits are scrambled together. The term splitter is used because the multiplication by an inverse of a sparse matrix can be viewed as if an incoming bit is split into several code bits.

Decoding graphs for scramble-LDPC and split-LDPC codes that combine parity checks of the LDPC codes, denoted by $\beta$, with those obtained from the scrambler or splitter, denoted by $\alpha$, are shown in Figure 1. These graphs are valid since a scrambled node $u_j$ satisfies a parity check with $d_s$ source nodes $s_i$ for a scramble-LDPC code, and node $s_i$ satisfies a parity check with $d_s$ split nodes $u_j$ for a split-LDPC code. In either case, the graph is a decoding graph of an LDPC code, whose code word $\mathbf{c}' \stackrel{\triangle}{=} [\mathbf{s}^T, \mathbf{c}^T]^T$ consists of a concatenation of $\mathbf{s}$ and $\mathbf{c}$, where the bits of $\mathbf{s}$ have been punctured. For scramble-LDPC, the degree of the $K$ systematic nodes $\mathbf{s}$ is $d_s$, the degree of the $K$ scrambled nodes $\mathbf{u}$ is $d_b + 1$, and the degree of the $N - K$ parity nodes $\boldsymbol{\vartheta}$ is $d_b$. For the split-LDPC, the nodes in $\mathbf{s}$ have degree 1, the nodes in $\mathbf{u}$ have degree $d_b + d_s$, and the nodes of $\boldsymbol{\vartheta}$ degree $d_b$. The parity check nodes $\alpha$ have degree $d_s + 1$, and the LDPC parity check nodes $\beta$ degree $d_c$ in both cases. *A-priori* information is available and passed from the nonuniform nodes of $\mathbf{s}$, and channel information is available at the code bit nodes of $\mathbf{c}$ ($\mathbf{u}$ and $\boldsymbol{\vartheta}$). The equations of the iterative decoding process can be found in [7].

## IV. Mutual Information

For a good channel code, the mutual information between the channel input vector $\mathbf{X}$ and the channel output vector $\mathbf{Y}$, $I(\mathbf{X};\mathbf{Y}) \stackrel{\triangle}{=} H(\mathbf{Y}) - H(\mathbf{Y}|\mathbf{X})$, should be large. We use capital letters to denote random variables and vectors. For a given channel and channel input distribution, the theoretically achievable channel code rate $R_c$ satisfies $NH_sR_c = I(\mathbf{X};\mathbf{Y})$. For a memoryless AWGN channel, the capacity achieving input distribution (of $X$) is Gaussian. Then, $I(X;Y) = C_{AWGN} = 0.5\log(1 + 2R_cE_{br}/N_0)$, and the theoretical minimum achievable SNR is $E_{br}/N_0 = \left(2^{2H_sR_C} - 1\right)/(2R_c)$. For an AWGN channel with BPSK input, the maximal achievable mutual information is

$$I(X;Y) = C_{BPSK} = 1 - \int_{-\infty}^{\infty} \frac{1}{\sqrt{\pi N_0}} e^{-\frac{(y-A)^2}{N_0}} \log\left(1 + e^{-\frac{4Ay}{N_0}}\right) dy, \quad (1)$$

where $A \stackrel{\triangle}{=} \sqrt{R_cE_{br}}$. This capacity is achieved with a uniform memoryless input distribution.

For a systematic code, if the source is nonuniform, the sequence $\mathbf{x}_s$, representing the modulation points for the bits in $\mathbf{s}$ is nonuniform. Thus, for every component $X_s$ of this sequence with BPSK modulation,

$$I(X_s;Y_s) = -\pi_0 \int_{-\infty}^{\infty} \frac{1}{\sqrt{\pi N_0}} e^{-\frac{(y+A)^2}{N_0}} \log\left(\pi_0 + \pi_1 e^{\frac{4Ay}{N_0}}\right) dy - \pi_1 \int_{-\infty}^{\infty} \frac{1}{\sqrt{\pi N_0}} e^{-\frac{(y-A)^2}{N_0}} \log\left(\pi_1 + \pi_0 e^{-\frac{4Ay}{N_0}}\right) dy. \quad (2)$$

Hence, the best achievable average mutual information for a systematic code with BPSK modulation is

$$I_{sys}(\mathbf{X},\mathbf{Y})/N = R_cI(X_s;Y_s) + (1 - R_c)\, C_{BPSK} \le C_{BPSK}. \quad (3)$$

It is achieved only if all parity bits are uniformly distributed. A theoretically optimal non-systematic code, on the other hand, if well designed, can generate uniform distributions for all components of $\mathbf{X}$, thus achieving $C_{BPSK}$. A split-LDPC code multiplies the nonuniform sequence $\mathbf{s}$ by a dense matrix (since the inverse of a sparse matrix is dense [2]), generating a split vector that due to the dense matrix has distribution very close to uniform. Hence, the best possible split-based code may be close to achieving the BPSK capacity.

For a regular scramble-based code, $d_s$ systematic nonuniform bits are scrambled into a code bit. Assuming that the parity bits that are generated by the LDPC code part are uniformly distributed, the best achievable mutual information can be computed using (2)-(3), where $\pi_1$ and $\pi_0$ in (2) are replaced by $q_0$ and $q_1$, respectively, which can be obtained, using Gallager's lemma [4], by $q_1 = 0.5\left[1 - (1 - 2\pi_1)^{d_s}\right]$, and $q_0 = 1 - q_1$. The probabilities $q_0$ and $q_1$ denote the probability of 0 and 1 in the scrambled sequence $\mathbf{u}$. For an irregular scrambler, similar computation can be done, considering all the different scrambling degrees.

Figure 2 shows the theoretical minimum achievable $E_{br}/N_0$ as function of $H_s$ for channel code rates $R_c = 0.5$, $R_c = 0.8$ and $R_c = 0.9$, and $E_{br}/N_0$ as a function of $R_c$ for $\pi_1 = 0.1$ ($H_s \approx 0.47$). The curves demonstrate the losses in systematic encoding, which increase with the code rate, or with the decrease of $H_s$ (increase in non-uniformity of $\mathbf{s}$). A well

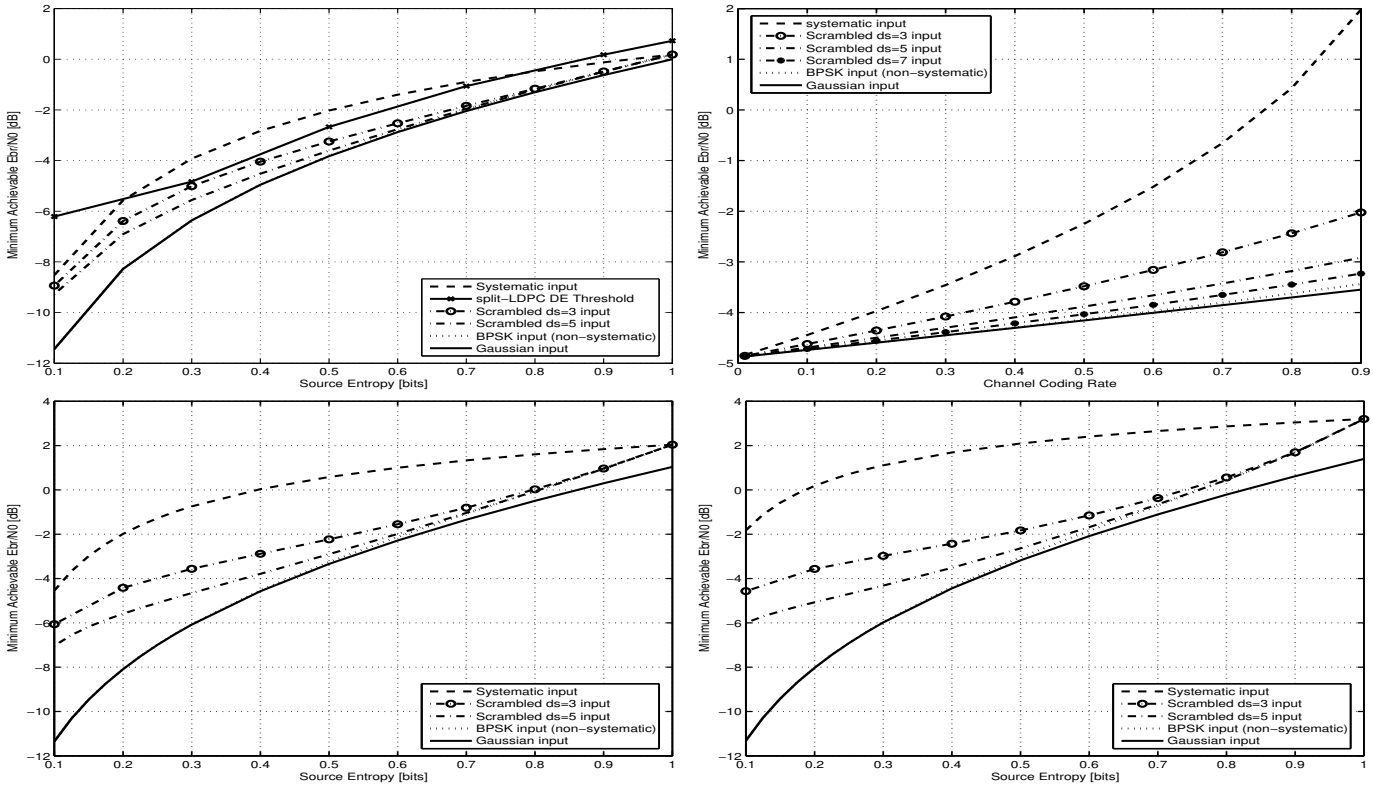


Fig. 2: Minimum achievable $E_{br}/N_0$ vs. source entropy $H_s$ and channel coding rate $R_c$ for different coding methods. The top left graph shows $E_{br}/N_0$ vs. $H_s$ for $R_c = 0.5$. The top right graph shows $E_{br}/N_0$ vs. $R_c$ for $\pi_1 = 0.1$. The bottom graphs show $E_{br}/N_0$ vs. $H_s$ for $R_c = 0.8$ (left) and $R_c = 0.9$ (right).

designed split-based code can avoid these losses. Scramble-based codes also incur some loss, but this decreases with increasing the scrambling degree. This is also consistent with the splitter being a dense scrambler. The increase in loss of systematic codes with $R_c$ points out to the significant benefit in using non-systematic high rate codes over systematic ones. This is expected, since high rate systematic codes contain more non-uniformly distributed bits, that have distribution far from the capacity achieving one.

## V. Density Evolution for Split-LDPC Codes

In this section, we describe how density evolution (DE) can be performed on a split-LDPC code with a nonuniform source in order to determine the code threshold under iterative decoding on a binary input AWGN channel. We use the discretized version of DE proposed in [3]. Let $p(x)$ denote a probability mass function (PMF) quantized from the probability distribution function (PDF) of the logarithmic ratio (LR) $\log(P(0)/P(1))$ messages, sent on a graph. Let the two PMF input $R$-operator $R(p_a, p_b)$ give a new PMF $p_c = R(p_a, p_b)$ defined as follows. For values $a$ and $b$, let $f(a,b) = 2\tanh^{-1}\left(\tanh\left(\frac{a}{2}\right)\tanh\left(\frac{b}{2}\right)\right)$. Then, $p_c(v) = \sum_{a,b:f(a,b)=v} p_a(a)p_b(b)$, i.e., the new distribution assigns the sum of $p_a(a)p_b(b)$ to $p_c(v)$ for all combinations of $a$ and $b$ that are mapped to $v$ by $f(a,b)$. Denote by $R^{j-1}p \triangleq R(p,\ldots,R(p,p))$ $j-1$ applications of the operator $R$ on distribution $p(x)$. Now, let the polynomial $\rho(x) \triangleq \sum_{j=2}^{d_r} \rho_j x^j$ denote a degree distribution of edges entering a check node. Then, $\rho(p) \triangleq \sum_{j=2}^{d_r} \rho_j R^{j-1}p$ is the LR message distribution resulting from averaging $R^{j-1}p$ over the degree distribution $\rho(x)$. Let $\lambda(x) \triangleq \sum_{i=2}^{d_l} \lambda_i x^i$ denote a degree distribution of edges entering a variable node. Then, define the distribution $\lambda(p) \triangleq \sum_{i=2}^{d_l} \lambda_i \otimes^{i-1} p$, where $\otimes^{i-1}p$ denotes $i-1$ convolutions of distribution $p(x)$ with itself.

Like DE for standard codes, messages propagating on graph edges are of LR-type, and will be characterized by their quantized PDF (or PMF). However, unlike codes for uniform sequences, we need to design DE that takes into account the nonuniform probabilities of the $s$ nodes, and the unique structure of the split-LDPC decoding graph. This yields different types of variable nodes for $u$ and $\vartheta$ nodes as well as different type of nodes for check nodes $\alpha$ and $\beta$. Fortunately, for split-LDPC codes, $s$ nodes can be absorbed in the $\alpha$ nodes. Furthermore, since the all 0 code word is not a typical code word, we cannot assume that this is the code word that was sent. However, we can perform DE on $u$ and $\vartheta$ nodes that take value 0 modulated at point $+1$. By symmetry, we have mirror image distributions around 0 on the LR messages for the same nodes taking value 1. We have three different types of messages propagating from variable to check nodes and two types of messages propagating from check nodes to variable

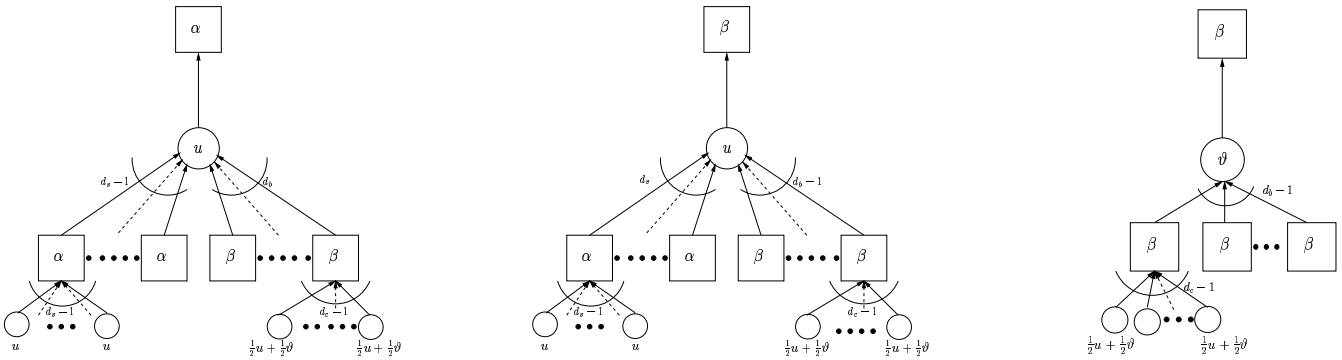


Fig. 3: Tree representations for type-1 (left), type-2 (middle), and type-3 (right) messages for rate $R_c = 1/2$ split-LDPC. (For different channel rates, the fractions on the nodes going to $\beta$ will be $R_c$ and $1 - R_c$.)

nodes. These are described below.

Let $p_1(x)$ denote the PMF of a *type-1* LR-message going from $u$ nodes to $\alpha$ nodes. Let $p_2(x)$ denote the PMF of a *type-2* LR-message going from $u$ nodes to $\beta$ nodes, and let $p_3(x)$ be the PMF of a *type-3* LR-message going from $\vartheta$ nodes to $\beta$ nodes. The PMF of LR-messages propagating from $\alpha$ check nodes to $u$ variable nodes is denoted $p_\alpha(x)$, and that of messages generated by $\beta$ is $p_\beta(x)$. Figure 3 shows the propagation of messages for the different types. Type-1 messages have $d_s - 1$ incoming extrinsics from check nodes $\alpha$ and $d_b$ incoming extrinsics from check nodes $\beta$. Type-2 messages have $d_s$ incoming extrinsics from check nodes $\alpha$ and $d_b - 1$ incoming extrinsics from check nodes $\beta$. A type-3 message has $d_b - 1$ incoming extrinsics from check nodes $\beta$.

The nonuniform distribution of s must be taken into account by two means: First, source *a-priori* LR-information is represented by a probability mass $p_s(x) = \delta(x-t)$ located at the LR $t = \log\left(\frac{1-\pi_1}{\pi_1}\right)$. Since $s$ nodes are absorbed in $\alpha$ checks, the addition of this LR is performed inside $\alpha$ check nodes while propagating $p_1(x)$ and $p_2(x)$. Second, assuming we propagate over variable nodes that take value 0, if an $s$ node is 1 (with probability $\pi_1$), we must have an odd number of $u$ nodes that equal 1 connected to check node $\alpha$ propagating to the 0 valued $u$ node. Conversely, if the $s$ node is 0, there must be an even number of 1 valued $u$ nodes connected to $\alpha$ at the bottom of the tree. Since $R[p(x), p(x)] = R[p(-x), p(-x)]$, we are only concerned with one $u$ node connected to $\alpha$, if $s = 1$, this $u$ node has to be 1, and otherwise 0. Thus the propagation depends on the value of $s$, which is dictated by $\pi_1$. Hence, a unique $u$ bit connected to $\alpha$ will be equal to 1 with probability $\pi_1$, and other $u$ bits connected to $\alpha$ can be forced to zero. The iterative equations for the DE on split-LDPC codes are described below, where variable nodes considered are those taking value 0 (modulated at $+1$).

*Proposition 1:* Consider a split-LDPC code built by the simple cascade of a $d_s$-splitter and a $(\lambda(x), \rho(x))$ binary LDPC code. For a nonuniform binary i.i.d. source characterized by $\pi_1$, density evolution is performed as follows:

$$p_\alpha^m(x) = R\left\{p_s(x), \right.$$
$$\left. R\left[(1-\pi_1)\ p_1^m(x)\ +\ \pi_1\ p_1^m(-x)\ ,\ \rho'_\alpha\left(p_1^m(x)\right)\right]\right\}$$

$$p_\beta^m(x) = \rho\left(R_c\ p_2^m(x)\ +\ (1-R_c)\ p_3^m(x)\right)$$
$$p_1^{m+1}(x) = p_0(x)\ \otimes\ \lambda_{1\alpha}\left(p_\alpha^m(x)\right)\ \otimes \lambda_1\left(p_\beta^m(x)\right)$$
$$p_2^{m+1}(x) = p_0(x)\ \otimes\ \lambda_{2\alpha}\left(p_\alpha^m(x)\right)\ \otimes \lambda\left(p_\beta^m(x)\right)$$
$$p_3^{m+1}(x) = p_0(x)\ \otimes\ \lambda\left(p_\beta^m(x)\right)$$

where the superscript $m$ represents the decoding iteration index, the symbol $\otimes$ represents classical convolution, $p_0(x)$ is the Gaussian quantized distribution of LR conditioned on a $+1$ transmitted symbol, and the polynomials are given by $\rho_\alpha(x) = \lambda_{1\alpha}(x) = x^{d_s-1}$, $\rho'_\alpha(x) = \rho_\alpha(x)/x$, $\lambda_{2\alpha}(x) = x\lambda_{1\alpha}(x)$ and $\lambda_1(x) = x\lambda(x)$. Termination is achieved when the total probability of error obtained from $p_1(x)$ (by integration over the negative region of $x$) is below a desired threshold.

We note that the proposed DE reduces to standard DE if $\pi_1 = 0.5$ [1]. This implies that a split-LDPC code has the same threshold as the original LDPC code for uniform sequences. The proposed DE method is demonstrated in Figure 2 (top left), where thresholds have been computed for $d_s = 3$, $\lambda(x) = 0.32660x + 0.11960x^2 + 0.18393x^3 + 0.36988x^4$, $\rho(x) = 0.78555x^5 + 0.21445x^6$ (see [6]), and source entropies varying from $0.1$ up to 1. The discretized LR interval was $[-15 \ldots + 30]$, with quantization step that equaled to $0.0025$. We note that the DE thresholds for this code are better than the best achievable SNR's for systematic codes in a wide region of low entropies. We also note that this may not be a very good split-LDPC code for certain values of $\pi_1$. In particular, it is likely that codes must be optimized for specific values of $\pi_1$. EXIT chart analysis [9] based on the message propagation structure described here has been used to compare between different regular codes with different splitting degrees. In particular, there are regular codes with $d_b = 3$ and splitting degrees between 3 and 5 whose thresholds are likely to be below $-2$dB for $\pi_1 = 0.1$ for code rates between $0.4$ and $0.9$. As Figure 2 (top-right) shows such codes are clearly better than the theoretical limit of systematic codes for rates $0.5$ or greater.

## VI. Simulation Results

Figure 4 shows simulation results for the different codes including systematic codes, MN codes, scramble-LDPC and

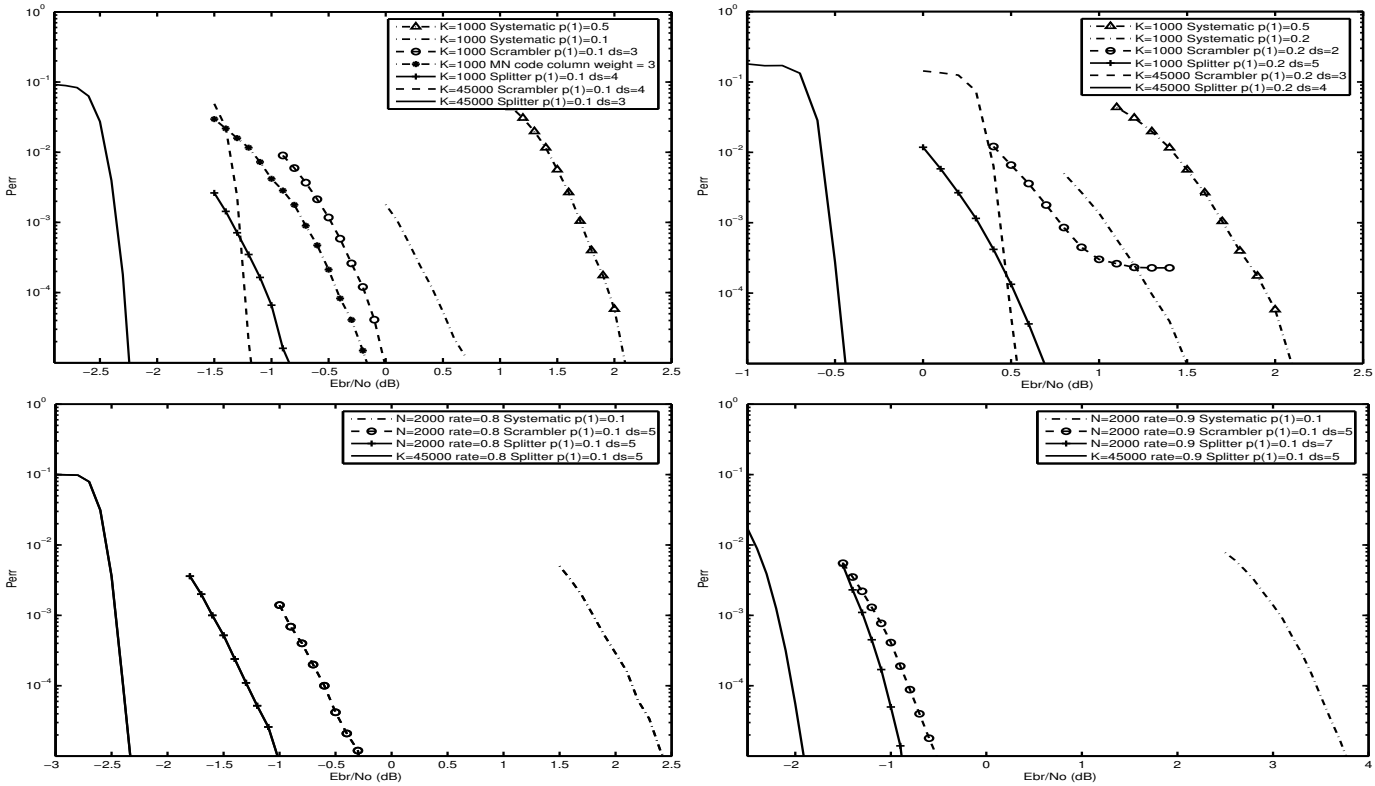


Fig. 4: Bit error probability vs. $E_{br}/N_0$ for different codes, rates, and source distributions, for blocks of $N = 2000$ and $K = 45000$. The top graphs are for $R_c = 0.5$ with $\pi_1 = 0.1$ (left) and $\pi_1 = 0.2$ (right). The bottom graphs are for $\pi_1 = 0.1$ and $R_c = 0.8$ (left) and $R_c = 0.9$ (right). All LDPC codes are regular with $d_b = 3$.

split-LDPC codes. All codes simulated have a regular structure, and for the scramble- and split-LDPC codes, regular scrambler/splitter. Only the best $d_s$ is shown for a specific code. The results show the advantage of the split-LDPC codes on the other codes. The waterfall regions for large blocks are within range of thresholds predicted. The disadvantage of systematic codes is clear, and is very significant at high rates. While the gain of the split based code over a systematic code is about $1.5$dB for BER $10^{-5}$ at rate $0.5$, this gain increases to $3.5$dB for $R_c = 0.8$, and to over $4.5$dB for $R_c = 0.9$. Such gains are in agreement with the minimum achievable SNR gains shown in Figure 2. MN codes and scramble-LDPC codes have very close performance. Scramble-LDPC codes achieve close performance to split-LDPC codes at high rates since their $d_s$ values at these rates are rather large.

## VII. Summary and Conclusions

We studied scramble-LDPC and split-LDPC codes for channel coding of nonuniform sequences. We showed theoretically that such codes have better potential than systematic codes, and that split-LDPC codes are better than other codes. We proposed a DE procedure to find threshold values of split-LDPC codes, and we showed simulation results illustrating the advantage of split-LDPC codes over other codes for coding nonuniform sequences. Specifically, the gain over systematic codes was shown to increase significantly at high rates.


## Acknowledgment

We would like to thank Siddhartha Mallik and Kai Xie for their precious help during this research.


## References


[1] A. Alloum, J. J. Boutros, G. I. Shamir, L. Wang, "Non-systematic LDPC codes via scrambling and splitting", in *Proc. Allerton Conference*, Monticello, IL, U.S.A., Sept. 28 - 30, 2005.

[2] G.Battail, "On Gallager's low-density parity-check codes," in *Proc. ISIT-2000*, Sorrento, Italy, June 2000.

[3] S.-Y. Chung, J. D. Forney, Jr., T. J. Richardson, R. L. Urbanke, "On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit," *IEEE Communications Letters*, vol. 5, no. 2, pp. 58-60, February 2001.

[4] R. G. Gallager, *Low Density Parity Check Codes*, MIT Press, 1963.

[5] D. J. C. Mackay and R. M. Neal, "Good codes based on very sparse matrices," *in Cryptography and Coding, 5th IMA Conference*, no. 1025 in lecture Notes in Computer Science, pp. 100-111, Berlin, Springer, 1995.

[6] T. J. Richardson, M. A. Shokrollahi, R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Transactions on Information Theory*, vol. 47, no. 2, February 2001.

[7] G. I. Shamir and J. J. Boutros, "Non systematic low-density parity-check codes for nonuniform sources," in *Proc. ISIT-2005*, Adelaide, Australia, pp. 1898-1902, Sept. 2005.

[8] G. I. Shamir, L. Wang, and J. J. Boutros, "High rate non-systematic LDPC codes for nonuniform sources," to appear in *4th International Symposium on Turbo Codes*, Munich, Germany, April 3-7, 2006.

[9] K. Xie, L. Wang, G. I. Shamir, and J. J. Boutros, "EXIT chart analysis for split-LDPC codes," submitted to the *ISIT-2006*, July 2006.

[10] G. C. Zhu, F. Alajaji, J. Bajcsy, and P. Mitran, "Transmission of nonuniform memoryless sources via nonsystematic turbo codes," *IEEE Trans. Commun*, vol. 52, no. 8, pp. 1344-1354, Aug., 2004.