Protograph-Based Raptor-Like LDPC Codes with Low Thresholds

Tsung-Yi Chen
Department of Electrical Engineering
University of California, Los Angeles

Dariush Divsalar
Jet Propulsion Laboratory
California Institute of Technology

Richard D. Wesel
Department of Electrical Engineering
University of California, Los Angeles

***Abstract*—This paper presents a new construction of punctured-node protograph-based Raptor-like (PN-PBRL) codes that is suitable for long-blocklength applications. As with the Raptor codes, additional parity bits can be easily produced by exclusive-OR operations on the precoded bits, providing extensive rate compatibility. The new construction provides low iterative decoding thresholds that are within 0.45 dB of the capacity for all code rates studied, and the construction is suitable for long blocklengths. Comparing at the same information block size of $k = 16368$ bits, the PN-PBRL codes are as good as the best known AR4JA codes in the waterfall region. The PN-PBRL codes also perform comparably to DVB-S2 LDPC codes even though the DVB-S2 codes have longer blocklength and outer BCH codes.**

I. Introduction

Low-density parity-check (LDPC) codes are a prominent class of error correcting codes. They were proposed by Gallager [1] in the early 1960s but did not receive much attention until decades later [2]. LDPC codes have a sparse parity check matrix and are decoded efficiently by the iterative belief propagation (BP) algorithm. (See for example [3].) In recent years, a new class of LDPC codes was introduced by Thorpe [4] and studied extensively in [5]. These protograph-based LDPC codes (protograph codes) use a relatively small graph (the protograph) that is replicated many times. This allows efficient decoder implementation in hardware [5].

Introduced by Luby [6] and Shokrollahi [7], LT codes and Raptor codes share many similarities with LDPC codes and are shown to achieve the capacity of the binary erasure channel (BEC) universally. Etesami et al. [8] explore the application of Raptor codes to binary memoryless symmetric channels and derive various results on the output degree distribution of LT codes. Results on Raptor codes such as [7] and [8] rely heavily on the assumption of large information blocks.

Rate-compatible punctured codes including rate-compatible punctured convolutional (RCPC) codes and rate-compatible punctured turbo (RCPT) codes are widely used in incremental redundancy (IR) schemes. These code families use a good "mother" code at the lowest rate and obtain the higher-rate codes by puncturing. One must carefully choose the puncturing patterns to avoid undue performance degradation as the rate increases. One drawback of rate-compatible puncturing is the difficulty of optimization over a large number of possible puncturing patterns.

Chen et al. proposed two new classes of protograph codes [9]: protograph-based Raptor-like (PBRL) codes that perform well in the short-blocklength regime and punctured-node protograph-based Raptor-like (PN-PBRL) codes that achieve improved thresholds by introducing punctured nodes. Similar to Raptor codes, the PBRL and PN-PBRL codes are rate-compatible and lend themselves to the IR applications. There is a similar construction technique for rate-compatible LDPC codes called "extending" in the literature. See for example [10] and [11] and the references therein.

This paper proposes a new construction of PN-PBRL codes that further improves the thresholds obtained in [9] and is particularly suitable for constructing long blocklength codes.

The paper is organized as follows: Section II reviews the preliminaries of LT codes and Raptor codes. Section III reviews the structure of protograph-based LDPC codes and introduces the construction of PBRL and PN-PBRL codes. The new construction and optimization technique is detailed in Section IV. Section V provides the simulation results. Finally, Section VI concludes the paper.

II. Raptor Codes

This section reviews the preliminaries of LT codes [6] and Raptor codes [7]. An LT code is described by the output degree distribution $\mathbf{\Omega}$ on its output symbols. Let k be a positive integer that denotes the number of the input symbols and $\mathbf{\Omega} = [\Omega_1, \Omega_2, \ldots, \Omega_k]$ be a distribution on the set of integers $\{1, 2, \ldots, k\}$ such that Ω_j denotes the probability of the value j being chosen. For the ith encoded bit, the encoder first chooses integer d_i randomly according to the distribution $\mathbf{\Omega}$. It then chooses d_i input symbols uniformly (with replacement) from $\{1, 2, \ldots, k\}$ and generates the i th output symbol by XORing the chosen input symbols. This encoding process continues indefinitely for $i = 1, 2, \ldots$, often concluding only when all interested receivers have been able to decode the message.

Let C be an (n, k) linear block code. A Raptor code is a serial concatenation of a code C, also called the "precode", and an LT code. A Raptor code is described by the parameters $(k, C, \Omega(x))$, where $\Omega(x) = \sum_{i=k}^{n} \Omega_i x^i$ is the generator polynomial of the output degree distribution of the LT code.

This research was carried out in part at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with NASA.

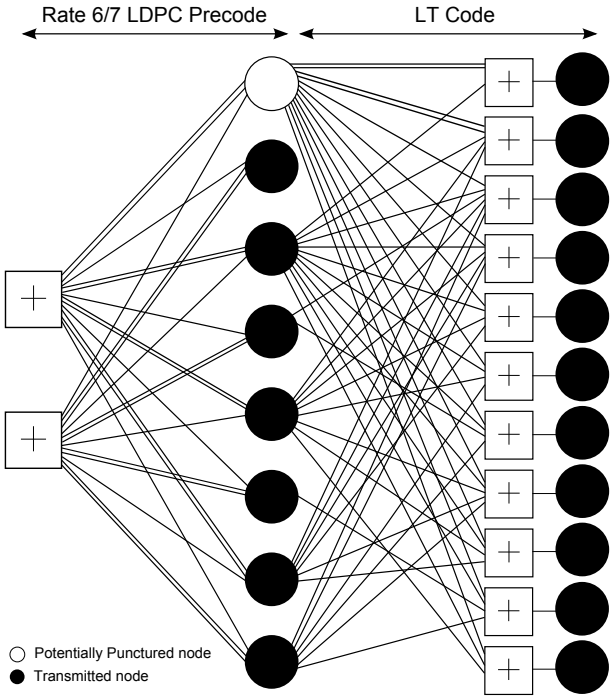

Fig. 1. PN-PBRL code with a rate-$6/7$ precode producing thresholds shown in Table I. The first node in the precode is always punctured. Lower-rate codes are obtained by transmitting the variable nodes in the LT code protograph starting from the top node.

The decoding of the Raptor code is performed in two stages: the decoder first decodes the LT code and recovers a fraction of its input symbols. The decoder then attempts to recover the remaining symbols by decoding the precode.

III. Protograph-Based Raptor-Like LDPC Code

This section reviews the structure of a protograph-based LDPC code and introduces the protograph-based Raptor-like (PBRL) LDPC codes.

A. Code Construction

A protograph-based LDPC code is constructed by a "copy-and-permute" operation (also called "lifting") from a Tanner graph with a relatively small number of nodes. The lifting operation first makes N copies of the protograph and then the edges of the same type among the protograph replicas are permuted.

Fig. 1 shows the protograph of a PBRL code. This protograph consists of two parts: (1) a relatively simple protograph code (on the left) representing the protograph of the precode, and (2) a number of check nodes (on the right) that are each connected to several variable nodes of the first part and an additional degree-one variable node. The second part represents the protograph of the LT code.

After the lifting operation, the first part can be seen as an LDPC precode, and the degree-one variable nodes of the second part can be efficiently encoded with the precoded symbols in a manner similar to the LT code. The structure of this protograph code resembles a Raptor code, but with a deterministic (rather than random) encoding rule for combining the precoded symbols. The rate of the precode in this example is $3/4$ if all variable nodes are transmitted, and as we increase the number of transmitted degree-one variable nodes in the LT code part, the code rate is reduced gradually.

Fig. 1 shows a punctured version of the PBRL code in which the top variable node of the precode is punctured. We will refer this class of codes as Puncture-Node Protograph-Based Raptor-Like (PN-PBRL) codes for the rest of the paper. EXIT chart analysis and experimental results [5] show that puncturing a variable node in the protograph can improve the iterative decoding threshold of the protograph. Note that puncturing the first variable node of the precode protograph gives a rate-$6/7$ precode. The subsequent code rates of $6/8, 6/9, \ldots, 6/18$ are obtained by transmitting the variable nodes of the LT code protograph from top to bottom. Regardless of the operating rate, the first variable node of the precode protograph is always punctured for the PN-PBRL codes.

Both PBRL and PN-PBRL codes share the same optimization procedure as we will show in Section IV. PN-PBRL codes are the main focus for the rest of the paper.

B. Decoding of Protograph-Based Raptor-Like LDPC Codes

Consider the decoding of a traditional Raptor code that collects the precoded symbols and encodes them with an LT code. In the case of an LDPC precode used with an LT code, decoding proceeds as follows: the decoder first performs BP decoding on the LT code and then performs BP decoding on the precode. The two-stage decoding implies the use of two different BP decoders, each exchanging their extrinsic information after the iterative decoding.

The PN-PBRL code always transmits the output symbols of the precode (except for the punctured nodes), allowing joint decoding of the LT code and the LDPC precode. This property also guarantees that the BP algorithm will always work for the initial transmission as well as the lower-rate codewords comprised of the original transmission and additional incremental redundancy. For traditional Raptor codes, the initial transmission may not contain enough information for BP decoding to succeed even in a noiseless setting.

For high-rate codes, the decoder can deactivate those check nodes in the LT portion for which the neighboring degree-one variable node is punctured. At the highest rate, when only the precode is transmitted, none of the check nodes in the LT portion need to be activated, offering possible complexity reduction.

IV. Design of PBRL LDPC Codes

This section proposes a design technique for finding good PBRL LDPC codes. Iterative belief propagation (BP) decoding is assumed.

A. Density Evolution with Reciprocal Channel Approximation

For the BI-AWGN channel, the asymptotic *iterative decoding threshold* characterizes the performance of the ensemble of LDPC codes based on a specified protograph. This threshold

$$H_p = \begin{bmatrix} \sigma^0+\sigma^1 & \sigma^3 & \sigma^1+\sigma^0 & \sigma^2 & \sigma^0+\sigma^1 & \sigma^0 & \sigma^2+\sigma^0 & \sigma^2 \\ \sigma^0 & \sigma^0+\sigma^1 & \sigma^0 & \sigma^0+\sigma^1 & \sigma^0 & \sigma^0+\sigma^1 & \sigma^0 & \sigma^0+\sigma^1 \end{bmatrix}. \quad (1)$$

indicates the minimum SNR required to transmit reliably with the underlying ensemble of codes as the blocklength grows to infinity.

Computing the exact iterative decoding threshold for BI-AWGN requires a large amount of computation. The reciprocal channel approximation (RCA) [12] [5] provides a fast and accurate approximation to the density evolution originally proposed by Richardson et al. [13], [14]. Experimental results [5] [12] show that the deviation from the exact density evolution is less than 0.01 dB. The following subsection describes an optimization process that uses the approximated threshold.

B. New PN-PBRL Code Construction for Long Blocklengths

For a fixed code rate, the design begins by finding a protograph LDPC code that has the linear minimum distance growth property [5] to serve as the precode. To construct the LT code part, first add a new check node and a new degree-one variable node to the protograph. Connect the new check node and the new degree-one variable node with an edge. Edges are added between the new check node and the precoded variable nodes according to the degree and placement that will optimize the iterative decoding threshold. This process continues until the combined precode/LT protograph reaches the lowest rate desired.

To obtain good thresholds at each subsequent code rate, the punctured node of the precode must connect to every additional check node with at least a single edge. These edges induce high-degree punctured variable nodes at low rates. The removal of these edges would result in a notable degradation of the thresholds.

This approach was originally avoided in the design of short blocklength PN-PBRL codes [9]. For codes with short blocklengths (the focus of [9]), the high-degree punctured node creates undesirable structures under BP decoding. Thus for short-blocklength codes, the resulting performance of the lifted codes does not correspond to the threshold results in the cases we studied, making the optimization process ad hoc rather than systematic. However, for longer blocklengths we see good agreement between the thresholds and the simulated performance.

Additional parallel edges between the punctured variable node in the precode protograph and the check nodes in the LT code protograph can further reduce the thresholds. Table I summarizes the thresholds of the code that has parallel edges between the punctured node and every check node in the LT code part. The gaps between the thresholds and the capacities are all less than 0.34 dB except the highest rate (the precode). The threshold of the rate-$1/3$ code is -0.266 dB and the gap to BI-AWGN capacity is only 0.229 dB.

The lifted codes for low rates, however, do not manifest the gain obtained in threshold. This is because that the connections of parallel edges hinder the flow of information from the degree-one variable nodes to the pre-coded variable nodes, preventing the BP decoder from converging.

The thresholds will slightly increase by limiting the number of parallel edges between the punctured variable node and the check nodes in the LT code part, but the lifted code graph will allow BP decoder to converge with a reasonable number of iterations.

For a given protograph precode with the linear minimum distance growth property, the optimization procedure of the LT code part is summarized as follows:

1) Add a new check and a new variable node that are connected to each other into the protograph.
2) Perform density evolution on the new protograph to determine the optimal degree distribution and the connections of the new check node to the precoded symbols.
3) Start again with step 1 if the lowest rate desired is not yet obtained.
4) Lift the resulting protograph with circulant permutations to match the desired initial blocklength.
5) Assign the permutation of each edge such that small cycles are avoided. (Based on the circulant PEG algorithm [15])

Note that parallel edges are allowed in step 2 but should be kept to a small number in the optimization process to avoid convergence problem in the lifted code graph. The search of the connections in step 2 is an exhaustive search. Fig. 1 is an example of having at most two parallel edges between the punctured variable node in the precode and the check nodes in the LT code part. The resulting thresholds obtained from the protograph in Fig. 1 are shown in Table II. The BER/FER simulations are shown in Sec. V.

A two-step lifting process is necessary to obtain good PN-PBRL codes with long blocklengths. The first lifting step, also known as pre-lifting, uses a relatively small lifting number and aims to remove the parallel edges in the protograph. The second lifting step then lifts the protograph resulting from the previous step to the desired blocklength. Suppose that the code is pre-lifted 4 times and then further lifted 682 times, the resulting information blocklength is then $6 \times 4 \times 682 = 16368$.

With a step size of 4×682=2728, subsequent code rates $6/8, 6/9, \ldots, 6/18$ are obtained by transmitting the output symbols of the LT code from each successive group of variable nodes starting from the top. The corresponding thresholds of each code rate are summarized in Table II. All possible rates from $1/3$ to $6/7$ with a resolution of 1 bit are indeed feasible by adding one variable node at a time.

V. Numerical Results

This section presents the simulations of the PN-PBRL codes of Fig. 1 shown in Section III. The lifting process of the protograph is accomplished by circulant permutation of each edge, which allows efficient hardware implementation of the

TABLE I
THRESHOLDS OF THE PN-PBRL CODES WITH ADDITIONAL PARALLEL EDGES TO PUNCTURED NODE. (E_b/N_0 IN DECIBELS).

Rate	Threshold	Capacity	Gap
6/7	3.077	2.625	0.452
6/8	1.956	1.626	0.330
6/9	1.314	1.059	0.255
6/10	0.948	0.679	0.269
6/11	0.678	0.401	0.277
6/12	0.422	0.187	0.235
6/13	0.270	0.0179	0.252
6/14	0.118	-0.122	0.240
6/15	0.005	-0.238	0.243
6/16	-0.102	-0.337	0.235
6/17	-0.172	-0.422	0.250
6/18	-0.266	-0.495	0.229

TABLE II
THRESHOLDS OF THE PN-PBRL CODES SHOWN IN FIG. 1. (E_b/N_0 IN DECIBELS).

Rate	Threshold	Capacity	Gap
6/7	3.077	2.625	0.452
6/8	1.956	1.626	0.330
6/9	1.392	1.059	0.333
6/10	1.078	0.679	0.399
6/11	0.798	0.401	0.397
6/12	0.484	0.187	0.297
6/13	0.338	0.018	0.320
6/14	0.144	-0.122	0.266
6/15	0.072	-0.238	0.310
6/16	0.030	-0.337	0.367
6/17	-0.024	-0.422	0.398
6/18	-0.150	-0.495	0.345

decoder. The design of the circulant permutation is based on circulant PEG algorithm. The minimum loop size of the precode part is 10 while the minimum loop size of the LT code part is 8.

The pre-lifted protograph of the PN-PBRL code is described as follows: Let H_p be the parity check matrix of the precode and H_{LT} be the parity matrix of the LT code excluding the degree-one variable nodes. Let σ be a 4×4 identity matrix shifted to the left by 1 and let 0 represent the 4×4 all-zero matrix. The parity check matrix of the precode is given by

$$H = \begin{bmatrix} H_p & O \\ H_{LT} & I \end{bmatrix}$$

where H_p and H_{LT} are given in equations (1) and (2). I is the identity matrix and O is the all-zero matrix with proper dimensions. Entries with multiple terms of σ indicate parallel edges in the protograph.

$$H_{LT} = \begin{bmatrix} \sigma^3+\sigma^0 & 0 & \sigma^0 & 0 & 0 & 0 & 0 & 0 \\ \sigma^1+\sigma^0 & 0 & \sigma^0 & 0 & \sigma^0 & 0 & \sigma^0 & 0 \\ \sigma^2 & 0 & \sigma^0 & \sigma^0 & \sigma^0 & 0 & \sigma^0 & \sigma^0 \\ \sigma^3 & 0 & \sigma^0 & 0 & \sigma^0 & 0 & \sigma^0 & \sigma^0 \\ \sigma^0 & 0 & \sigma^0 & 0 & \sigma^0 & 0 & \sigma^0 & 0 \\ \sigma^0 & 0 & \sigma^0 & 0 & \sigma^0 & 0 & 0 & \sigma^0 \\ \sigma^2 & 0 & \sigma^0 & 0 & \sigma^0 & 0 & \sigma^0 & \sigma^2 \\ \sigma^0 & 0 & \sigma^0 & \sigma^0 & 0 & 0 & 0 & \sigma^0 \\ \sigma^0 & 0 & \sigma^0 & 0 & \sigma^0 & 0 & \sigma^0 & 0 \\ \sigma^1 & 0 & \sigma^0 & 0 & 0 & \sigma^0 & 0 & \sigma^0 \\ \sigma^0 & 0 & \sigma^0 & 0 & \sigma^0 & 0 & 0 & 0 \end{bmatrix}. \quad (2)$$

The pre-lifted protograph contains no parallel edges and is further lifted by 682 in a similar fashion but using a larger

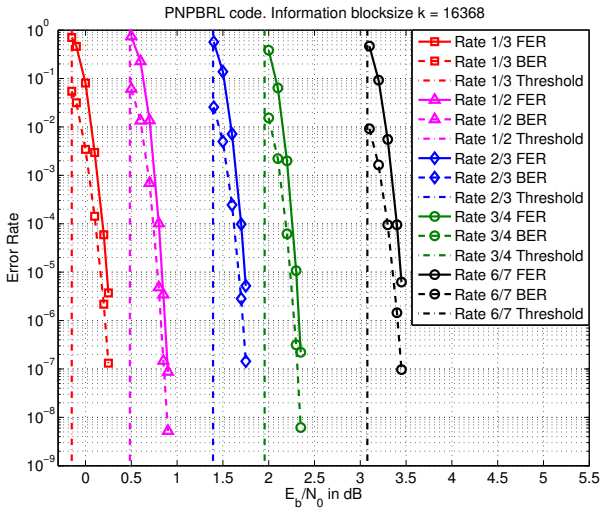

Fig. 2. Frame error rates and bit error rates for codes in the PN-PBRL code family of Example 1. The information block is fixed to 16368 and the blocklength varies for different code rates with a step size of 2728 for each increment of the denominator by one (with the numerator fixed at 6). For example, the rate $1/2$ code has blocklength 32736 and rate $1/3$ code has blocklength 49104.

matrix Σ: a 682×682 identity matrix shifted to the left by 1. The powers of the matrix Σ for all edges in the pre-lifted protograph are omitted due to space limitations.

All simulations use sum product algorithm with flooding. The iterative procedure will terminate if the decoding is successful or if it reaches the maximum number of iterations. The maximum iteration is 100 for all simulations if not otherwise stated. Simulations results with 50 iterations is also presented in the following for reader's convenience to compare with the DVB-S2 standard [16].

Fig. 2 shows the simulations of the PN-PBRL code family with information block size $k = 16368$ and rates $6/7, 3/4, 2/3, 1/2$ and $1/3$. The blocklengths are 19096, 21824, 24552, 32736 and 49104, respectively. At a fixed frame error rate of 10^{-5}, the estimated gaps of these codes to the BI-AWGN capacity are summarized in Table III.

TABLE III
SNRS REQUIRED TO ACHIEVE FER 10^{-5} FOR CODE EXAMPLE 1.

Rate	Required SNR	Capacity	Gap to capacity
6/7	3.39	2.625	0.765
6/8	2.30	1.626	0.674
6/9	1.74	1.059	0.681
6/12	0.83	0.187	0.643
6/18	0.23	-0.495	0.725

Fig. 3 compares the PN-PBRL codes to the LDPC codes in the DVB-S2 standard. Note that the DVB-S2 LDPC codes have a fixed blocklength of 64800 bits for each code rate but the PN-PBRL codes still have comparable (often better) performance. Take the rate $1/2$ PN-PBRL code for example: with almost half of the blocklength, the PN-PBRL still outperforms the DVB-S2 code.

Fig. 4 shows that in the waterfall region, the PN-PBRL

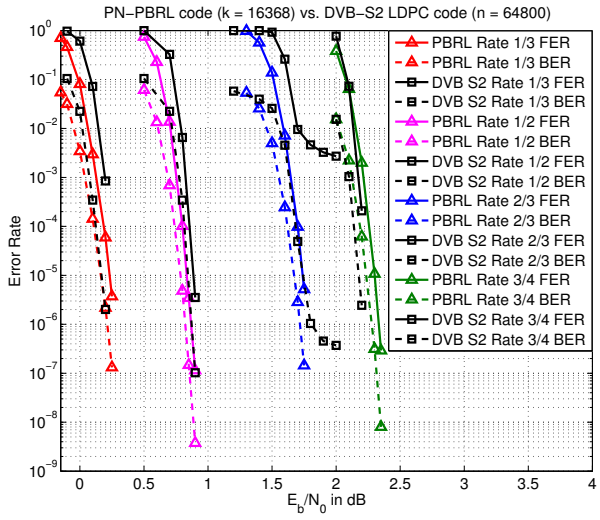

Fig. 3. Frame error rates and bit error rates for both DVB-S2 (LDPC codes only) and PN-PBRL codes. The blocklengths of the DVB-S2 codes are fixed to 64800 where the PN-PBRL codes have blocklength 21824, 24552, 32736 and 49104 for rates $3/4, 2/3, 1/2$ and $1/3$, respectively.

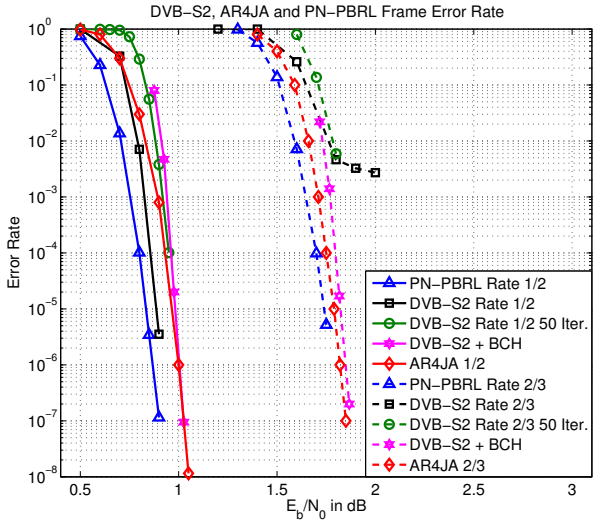

Fig. 4. Frame error rates for DVB-S2 LDPC only, DVB-S2 LDPC concatenated with an outer BCH code, AR4JA codes and PN-PBRL codes at rates $1/2$ and $2/3$. Note that the maximum iteration for LDPC+BCH is 50. The blocklengths of the DVB-S2 codes are fixed to 64800 where the PN-PBRL and AR4JA codes have blocklength 32736 and 24552 for rates $1/2, 2/3$, respectively. The overall rates of DVB-S2 codes after concatenation are 0.497 and 0.6642.

codes outperform the best known AR4JA codes [17]. Fig. 4 also shows the performance of the DVB-S2 LDPC codes concatenated with BCH codes (with maximum 50 iterations). With shorter blocklengths and higher rates, the PN-PBRL codes still outperform the concatenated codes in the waterfall region. The error floor performance of PN-PBRL codes needs to be simulated by FPGA and is left as future work.

VI. Conclusion

This paper proposes a new construction of punctured-node protograph-based Raptor-like (PN-PBRL) LDPC codes and provides a procedure for optimizing the codes. Optimization of the protograph is based on asymptotic iterative decoding thresholds of the protograph LDPC codes. Instead of the original density evolution, the reciprocal channel approximation is used to obtain a fast and accurate approximation to the thresholds of PN-PBRL codes. The assignment of the circulants is based on circulant PEG algorithm.

The proposed PN-PBRL codes have several advantages in terms of complexity: first, the protograph structure allows low-complexity encoding and decoding. Second, the systematic structure allows joint decoding of the precode and LT code with the same decoder. Finally, the code is rate-compatible and can be readily applied to incremental redundancy schemes with feedback and other schemes that require rate-compatible channel codes.

The long-blocklength codes shown in the paper demonstrate excellent capacity-approaching performance and do not have error floors down to FER as low as 10^{-7}. The gaps are all within 0.8 dB to the BI-AWGN capacity across a variety of rates.

References

[1] R. G. Gallager, "Low-density parity-check codes," Ph.D. dissertation, MIT, Cambridge, MA, 1963.

[2] D. J. C. Mackay and R. M. Neal, "Near shannon limit performance of low density parity check codes," *Electronics Letters*, vol. 32, Issue 18, p. 1645, Aug. 1996.

[3] M. R. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inf. Theory*, vol. 27, No. 5, pp. 533–547, 1981.

[4] J. Thorpe, "Low Density Parity Check (LDPC) codes constructed from protographs," *JPL IPN Progress Report*, vol. 42–154, Aug. 2003.

[5] D. Divsalar, S. Donlinar, C. R. Jones, and K. Andrews, "Capacity-approaching protograph codes," *IEEE J. Sel. Areas Commun.*, vol. 27, No. 6, pp. 876–888, Aug. 2009.

[6] M. Luby, "LT codes," in *Proc. 43rd Annu. IEEE Symp. Foundations of Computer Science (FOCS)*, Vancouver, BC, Canada, Nov. 2002.

[7] A. Shokorollahi, "Raptor codes," *IEEE Trans. Inf. Theory*, vol. 52, pp. 2551–2567, Jun. 2006.

[8] O. Etesami and A. Shokorollahi, "Raptor codes on binary memoryless symmetric channels," *IEEE Trans. Inf. Theory*, vol. 52, pp. 2033–2051, May 2006.

[9] T.-Y. Chen, D. Divsalar, J. Wang, and R. D. Wesel, "Protograph-based raptor-like LDPC codes for rate-compatibility with short blocklengths," in *Proc. GLOBECOM '11*, Houston, TX, Dec. 2011.

[10] T. V. Nguyen, A. Nosratinia, and D. Divsalar, "The deisgn of rate-compatible protograph LDPC codes," in *Proc. 48th Annual Allerton Conference*, Allerton, IL, USA, Sep. 2010.

[11] M. R. Yazdani and A. H. Banihashemi, "On construction of rate-compatible low-density parity-check codes," *IEEE Commun. Lett.*, vol. 8, No. 3, pp. 159–161, Mar. 2004.

[12] S. Y. Chung, "On the construction of some capacity-approaching coding schemes," Ph.D. dissertation, MIT, Cambridge, MA, 2000.

[13] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inf. Theory*, vol. 47, No.2, pp. 599–618, Feb. 2001.

[14] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 47, No.2, pp. 618–637, Feb. 2001.

[15] K. Andrews, S. Dolinar, D. Divsalar, and J. Thorpe, "Design of low-density parity-check (LDPC) codes for deep-space applications," *JPL IPN Progress Report*, vol. 42–159, Nov. 2004.

[16] "Digital video broadcasting; second generation framing structure, channel coding and modulation systems for broadcasting, interactive services, news gathering and other broadband satellite applications," *ETSI EN 302 307 V1.2.1*, Aug. 2009.

[17] "Low density parity check codes for use in near-earth and deep space. orange book." *CCSDS standard, Issue No.2*, Sep. 2007.