Construction of Rate-Compatible LDPC Codes Utilizing Information Shortening and Parity Puncturing

Tao Tian

QUALCOMM Incorporated, San Diego, CA 92121, USA
Email: ttian@qualcomm.com

Christopher R. Jones

Jet Propulsion Laboratory, California Institute of Technology, NASA, CA 91109, USA
Email: crjones@jpl.nasa.gov

This paper proposes a method for constructing rate-compatible low-density parity-check (LDPC) codes. The construction considers the problem of optimizing a family of rate-compatible degree distributions as well as the placement of bipartite graph edges. A hybrid approach that combines information shortening and parity puncturing is proposed. Local graph conditioning techniques for the suppression of error floors are also included in the construction methodology.

Keywords and phrases: rate compatibility, shortened codes, punctured codes, irregular low-density parity-check codes, density evolution, extrinsic message degree.

1. INTRODUCTION

Complexity-constrained systems that undergo variations in link budget may benefit from the adoption of a rate-compatible family of codes. Code symbol puncturing has been widely used to construct rate-compatible convolutional codes [1], parallel concatenated codes [2, 3], and serially concatenated codes [4]. Techniques for implementing rate compatibility in the context of LDPC coding have primarily pursued parity puncturing [5, 6]. In particular, a density evolution model for an additive white Gaussian noise (AWGN) channel with puncturing was developed by Ha et al. [5]. The model was used to find asymptotically optimal puncturing fractions (in a density evolution sense) for each variable node degree of a mother code distribution to achieve given (higher) code rates. Li and Narayanan [7] show that puncturing alone is insufficient for the formation of a sequence of capacity-approaching LDPC codes across a wide range of rates. In addition to puncturing, the authors in [7, 8] used extending (adding columns and rows to the code's parity matrix) to achieve rate compatibility.

In contrast to prior work that has focused primarily on puncturing and extending, this paper proposes a rate-compatible scheme that carefully combines parity puncturing and information shortening. In addition to providing good asymptotic distributions with which to achieve rate compatibility, we also present a column weight assignment strategy that seeks to adhere to the weight distribution goal provided by each rate. The parity puncturing portion of our method leverages the work of Ha et al. [5] while the information shortening part of the approach introduces a novel technique for "fitting" an optimal degree distribution for each component rate to the portion of the graph that effectively implements this rate. Simulation results show that a hybrid scheme achieves close-to-capacity performance with low error floors across a wide range (0.1 to 0.9) of code rates.

Shortening and puncturing techniques can affect the rate that a given graph implements by forcing what would otherwise be channel reliability values on variable node inputs to distinct extreme values. Shortening (rate reduction) is achieved by placing infinite reliability on the corresponding graph variable node. Puncturing (rate expansion) is achieved by placing 50% reliability on variable nodes in the decoding graph that correspond to punctured code symbols. At the transmitter, both techniques are implemented through the omission of the shortened or punctured code symbols during the transmission of the codeword.

Motivation to implement a rate-compatible approach that employs both shortening and puncturing stems from a few simple observations. First, if an approach uses only
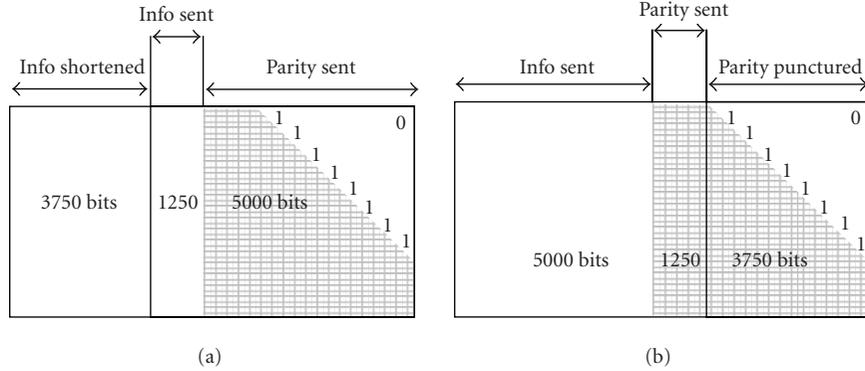
FIGURE 1: Parity matrix of the proposed rate-compatible scheme for center rate $R_0 = 0.5$. The lower triangular structure speeds up encoding and suppresses error floor, as explained below; (a) information shortening to achieve $R = 0.2$ and (b) parity puncturing to achieve $R = 0.8$.

information shortening to reduce rate, then the mother code that is used should have a relatively high rate and will contain a relatively large number of columns compared to its number of rows. The girth of the high-rate mother code is likely impaired and structures that have low extrinsic message degree [9] may dominate code performance.

The puncturing technique from [5] achieves good results for $0.5 \leq R \leq 0.9$. However, high-performance rate compatibility across $0.1 \leq R \leq 0.9$ is difficult to achieve with puncturing alone since 88.9% of the columns of a rate 0.1 mother code matrix would need to be punctured to achieve rate 0.9. In such an approach, avoidance of stopping set puncturing at the highest rate would dictate a parity matrix structure that would yield relatively poor low rate performance. Our hybrid rate-compatible scheme achieves results similar to those of [5] in rates ranging from $0.5 \leq R \leq 0.9$. This is to be expected since the puncturing profile for this range of rates has been borrowed from [5]. However, the proposed technique also gracefully extends the useful rate range down to $R = 0.1$. In general, the hybrid scheme can achieve rate compatibility across a rate range $R_L \leq R \leq R_H$ by setting the mother code rate to $R_0 = (R_L + R_H)/2$.

Figure 1 shows an example of how the proposed method achieves low rate 0.2 and high rate 0.8 from a length-10^4 mother code that has rate $R_0 = 0.5$. Information bits are on the left side (white area) and parity bits on the right side (shaded area).

The above rate-compatible LDPC code can be used within the framework of a single iterative encoder/decoder pair. To achieve $R = 0.2$ from the rate 0.5 mother code, zeros are used instead of payload data for the leftmost 3750 information bits in the encoding/decoding process. To achieve $R = 0.8$ from the rate 0.5 mother code, the rightmost 3750 parity bits are punctured and the decoder initializes the punctured variables with 50% reliability. The number of information bits shortened and the number of parity bits punctured can be varied to achieve a wide range of code rates. Rates above R_0 are achieved exclusively through parity puncturing and rates below R_0 exclusively through information shortening. In Section 2, we propose a column degree

assigning algorithm that has been designed to fit the degree distribution associated with a given code rate to the desired degree distribution for that rate. In Section 3, we discuss how to generate the desired degree distributions that achieve good shortening performance across $[R_L, R_0]$.

2. DEGREE DISTRIBUTION SELECTION AND COLUMN ASSIGNMENT STRATEGY

Our construction methodology first obtains a degree distribution for each of the target rates and then constructs the parity matrix using a greedy approach that tries to best match each subportion of the matrix with the degree distribution that is associated with the corresponding rate.

We denote the *node-wise* variable degree distribution by $\tilde{\lambda}$, whose relationship with the edge-wise variable degree distribution λ is

$$\tilde{\lambda}_i = \frac{\lambda_i/i}{\sum_{j=2}^{d_v} \lambda_j/j}, \quad i = 2, 3, \ldots, d_v, \tag{1}$$

where d_v is the highest variable degree.

Similarly, the *node-wise* constraint degree distribution $\tilde{\rho}$ is related to the edge-wise constraint degree distribution ρ by

$$\tilde{\rho}_i = \frac{\rho_i/i}{\sum_{j=2}^{d_c} \rho_j/j}, \quad i = 2, 3, \ldots, d_c, \tag{2}$$

where d_c is the highest constraint degree.

A sequence of node-wise variable degree distributions such as the following will be used:

$$\tilde{\lambda}^{(R_L)}, \ldots, \tilde{\lambda}^{(R_\alpha)}, \ldots, \tilde{\lambda}^{(R_0)}, \ldots, \tilde{\lambda}^{(R_\beta)}, \ldots, \tilde{\lambda}^{(R_H)},$$
$$R_L < \cdots < R_\alpha < \cdots < R_0 < \cdots < R_\beta < \cdots < R_H, \tag{3}$$

where R_0 denotes the code rate of the mother code, and $[R_L, R_H]$ denotes the code rate range of the rate-compatible scheme.

At code rates $R_\alpha < R_0$, degree distributions are found using a linear program whose constraints and objective are determined by Chung's Gaussian approximation [10]. Both Urbanke and Chung [10, 11] have indicated that the selection of a uniform or nearly uniform constraint node degree yields good threshold performance. Throughout the rest of the paper, the constraint degree distribution will be concentrated at a level that is optimal for the mother code at rate R_0.

Shortened LDPC codes have the property of generic LDPC codes, therefore, the *node-wise* average constraint degree of a shortened code can be calculated from the variable degree distribution of the corresponding code rate

$$
\begin{aligned}
\bar{d}_c^{(R_\alpha)} &= \sum_{j=2}^{d_c} j \tilde{\rho}_j^{(R_\alpha)} = \frac{1}{\sum_{j=2}^{d_c} \rho_j^{(R_\alpha)}/j} \\
&= \frac{1}{(1 - R_\alpha) \sum_{j=2}^{d_v} \lambda_j^{(R_\alpha)}/j} \\
&= \frac{\sum_{j=2}^{d_v} j \tilde{\lambda}_j^{(R_\alpha)}}{1 - R_\alpha},
\end{aligned}
\tag{4}
$$

where a well-known relationship $R = 1 - ((\sum_{j=2}^{d_c} \rho_j/j)/(\sum_{j=2}^{d_v} \lambda_j/j))$ is applied (see [11]). It should be noted that when we generate the mother code parity matrix, we control the row budget in a way such that the constraint degree distributions of shortened codes are as concentrated as possible.

The simplicity in the design of concentrated constraint degree distributions is not shared by that of variable degree distributions, which vary with rate. First, we normalize these distributions with respect to the dimensions of the mother code matrix (as the component distributions must "fit" the mother code matrix),

$$
\tilde{\Lambda}^{(R_\alpha)} = \frac{1 - R_0}{1 - R_\alpha} \tilde{\lambda}^{(R_\alpha)}.
\tag{5}
$$

For code rate $R_\beta > R_0$, we puncture $\tilde{\lambda}^{(R_0)}$ using the technique suggested by Ha et al. in [5]. Ha uses the notation $\pi_i^{(R_\beta)}$ to define the puncturing fraction on degree-i variable nodes at rate $R_\beta > R_0$. In summary, we use the following definition for the normalized node-wise degree distribution of the rate-compatible code family:

$$
\tilde{\Lambda}_i^{(R)} = \begin{cases} \dfrac{1 - R_0}{1 - R} \tilde{\lambda}_i^{(R)} & \text{if } 0 \le R \le R_0, \\ \tilde{\lambda}_i^{(R_0)} \left(1 - \pi_i^{(R)}\right) & \text{if } R_0 < R \le 1. \end{cases}
\tag{6}
$$

Note that an essentially continuously parameterized (in rate) $\tilde{\Lambda}_i^{(R)}$ can be achieved by interpolation.

The mother code degree distribution we use is a rate 0.5 code from [5]: $\lambda(x) = 0.25105x + 0.30938x^2 + 0.00104x^3 + 0.43853x^9$ and $\rho(x) = 0.63676x^6 + 0.36324x^7$. We plot $\tilde{\Lambda}_i$ of a rate-compatible scheme based on this mother code in
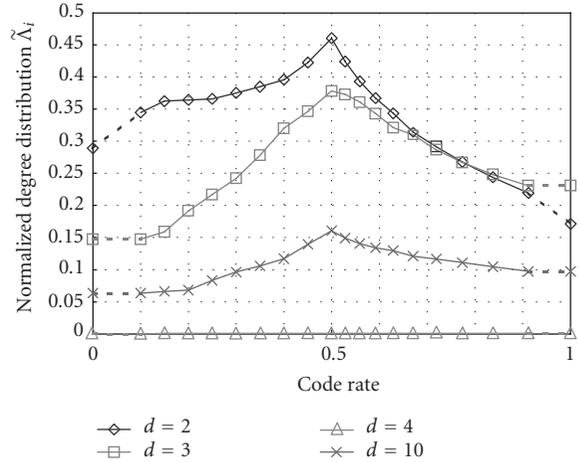

FIGURE 2: Normalized node-wise variable degree distribution $\tilde{\Lambda}_i$.

Figure 2. Distributions for the shortened portion ($R < R_0$) of the scheme are generated with a constrained density evolution algorithm to be discussed in the next section.

The curves in Figure 2 must be extrapolated to code rates 0 and 1 for the allocation of columns in the middle of the mother code matrix (where either shortening or puncturing reach their respective maximum levels). Because an application is only interested in a certain code rate range $[R_L, R_H]$, the allocation of columns out of the interesting rate range is arbitrary to some extent. However, the extrapolation must satisfy

(i) monotonicity, $\tilde{\Lambda}_i$ is nondecreasing for $R < 0.5$ and nonincreasing for $R > 0.5$,

(ii) continuity,

$$
\tilde{\Lambda}_i^{(0)} + \tilde{\Lambda}_i^{(1)} = \tilde{\Lambda}_i^{(R_0)}.
\tag{7}
$$

Equation (7) can be understood in the following way: $\tilde{\Lambda}_i^{(0)}$ describes the normalized distribution of the parity portion of H; $\tilde{\Lambda}_i^{(1)}$ describes the normalized distribution of the information portion of H; the sum of $\tilde{\Lambda}_i^{(0)}$ and $\tilde{\Lambda}_i^{(1)}$ is equal to the overall distribution of the mother code (at rate $R_0 = 0.5$). We use an extrapolation strategy that optimizes the threshold signal-to-noise ratio (SNR) at the lowest shortened code rate R_L while simultaneously satisfying the above two criteria. These ideas will be discussed in more detail in the next section.

Next we present a greedy algorithm (see Algorithm 1) to assign column degrees in a way that is meant to minimize the discrepancy between the distribution realized in the final matrix and the distribution goal shown in Figure 2. The number of columns that have been assigned to degree-i is denoted by n_i and code block length by n.

The column being constructed is allocated the degree where the two distributions have the largest mismatch. The first part of the column assignment strategy, columns up to

```
           Column degree allocation
n_i = 0, i = 2, 3, ..., d_v - 1;
for (column j = 1; j ≤ n; j + +)
    x = j/n;
    if (x < R_0)
        p_i = n × {Λ̃_i^(R_0) - Λ̃_i^(R_0-x)} - n_i,
        i = 2, 3, ..., d_v - 1;
    else
        p_i = n × Λ̃_i^(1-x+R_0) - n_i, i = 2, 3, ..., d_v - 1;
    endif
    η = arg max_i{p_i};
    Assign the degree of column j to η;
    n_η++;
end
```

ALGORITHM 1: The greedy algorithm.

index $j = nR_0$, is assigned degrees W_j according to

$$W_j = \arg\max_i \left[n\left(\tilde{\Lambda}_i^{(R_0)} - \tilde{\Lambda}_i^{(R_0-x)} \right) - n_i \right]. \qquad (8)$$

To understand the above objective, note that the first columns assigned correspond to columns in the shortening portion of the matrix with rates close to R_0. As the column index approaches nR_0, the portion of the matrix to the right must implement a code with rate close to zero (which occurs when nR_0 columns have been nulled (shortened)). When column assignment begins, the target rate is R_0. As the assignment index increases, the distribution target in Figure 2 moves left toward rate 0. Per the objective in (8), node-wise distributions for variable degrees that fall off more rapidly as code rate decreases from R_0 to 0 are assigned with higher priority.

After the first nR_0 column indices have been assigned variable degrees, the target rate of the graph switches from zero to one with a single index step. As previously mentioned, a discontinuity in the target degree distribution that might otherwise occur is avoided by enforcing the continuity condition of (7). The second part of the column assignment strategy, columns in the index range $j \in \{nR_0 + 1, n\}$, is assigned degrees W_j according to

$$W_j = \arg\max_i \left[n\tilde{\Lambda}_i^{(1-x+R_0)} - n_i \right]. \qquad (9)$$

The first columns assigned under this objective (columns with j indices slightly larger than nR_0) correspond to codes with rate close to 1 (which occurs when nR_0 columns have been punctured). As the column index approaches n, the entire matrix implements a code with rate close to R_0 (exactly when no columns are punctured). As the assignment index increases, the distribution target in Figure 2 moves left from rate 1 toward rate R_0. Per the objective in (9), node-wise distributions for variable degrees that rise more rapidly as code rate decreases from 1 to R_0 are assigned with higher priority.

In addition to the column degree assignment strategy, we also use the lower triangular structure in Figure 1b. Reasons

for this are twofold. First, the parity matrix satisfies the structure proposed by [12] and hence has an almost linear time encoder. Second, the proposed structure can suppress error floors. We know from [13] that to form a stopping set, each constraint neighbor of a variable set must connect to this variable set at least twice. Any column subset of the rightmost portion of the matrix in Figure 1b is not a stopping set, because the leftmost column of this subset is by construction only singly connected to this set.

3. CONSTRAINED DENSITY EVOLUTION

We need to design the edge-wise degree distributions $\lambda(x) = \sum_{i=2}^{d_v} \lambda_i x^{i-1}$ (for variables) and $\rho(x) = \sum_{i=2}^{d_c} \rho_i x^{i-1}$ (for constraints), where d_v and d_c are the highest variable degree and the highest constraint degree, respectively. Our construction shall employ node-wise degree distributions:

$$\tilde{\lambda}_i = \frac{\lambda_i/i}{\sum_{j=2}^{d_v} \lambda_j/j}, \quad i = 2, 3, \ldots, d_v,$$

$$\tilde{\rho}_i = \frac{\rho_i/i}{\sum_{j=2}^{d_c} \rho_j/j}, \quad i = 2, 3, \ldots, d_c. \qquad (10)$$

The well-known work of Chung et al. [10] presented a technique that approximates the true evolution of densities in an iterative decoding procedure with a mixture of Gaussian densities. The following equations describe the recursions provided by Chung:

$$\bar{u}_l = \sum_j \rho_j \Theta^{-1}\left(\bar{T}_{l-1}^{j-1} \right),$$

$$\bar{T}_l = \sum_i \lambda_i \Theta\left(\bar{u}_0 + (i - 1)\bar{u}_l \right),$$

$$\Theta(x) = \begin{cases} \frac{1}{\sqrt{4\pi x}} \int_R \tanh\left(\frac{u}{2}\right) \exp\left(-\frac{(u-x)^2}{4x}\right) du & \text{if } x > 0, \\ 0 & \text{if } x = 0. \end{cases}$$

$$\bar{u}_1 = 0 \text{ (initial condition)}, \qquad (11)$$

where $\bar{u}_l$ is the mean of the log-likelihood ratio (LLR) generated by constraint nodes after the lth iteration, $\bar{T}_l = E(\tanh(v_l/2))$, v_l is the LLR generated by variable nodes after the lth iteration, and $\bar{u}_0 = 2/\sigma^2$ is the mean of the *a priori* LLRs.

Using the above recursions in conjunction with bisection on initial mean value ($\bar{u}_0$), an irregular degree distribution can be optimized for a given code rate as in Algorithm 2, where inequality (d) is the stability constraint that enforces code convergence at high LLR (see [11]). From (1) and (6), we can obtain

$$\tilde{\Lambda}_i^{(R)} = (1 - R_0)\frac{\int \lambda^{(R)}}{\int \rho^{(R)}} \times \frac{\lambda_i^{(R)}/i}{\int \lambda^{(R)}} = \frac{1 - R_0}{i \int \rho^{(R)}} \lambda_i^{(R)}. \qquad (12)$$

For fixed ρ, maximize $1/(1 - R) = \int \lambda / \int \rho$
such that
(a) $\sum_{j=2}^{d_v} \lambda_j = 1$,
(b) $\lambda_j \geq 0$,
(c) $\sum_i \lambda_i \Theta(\bar{u}_0 + (i - 1)\bar{u}) > \bar{T}$ for many $(\bar{T}, \bar{u})$
    pairs that satisfy $\bar{u} = \sum_j \rho_j \Theta^{-1}(\bar{T}^{j-1})$,
    $\Theta(\bar{u}_0) < \bar{T} < 1$,
(d) $\lambda_2 < (\exp(\bar{u}_0/4))/\rho'(1)$.

ALGORITHM 2: Traditional optimization algorithm.

The monotonicity constraint can be expressed as

$$\tilde{\Lambda}_i^{(R_1)} \leq \frac{1 - R_0}{i \int \rho^{(R)}} \lambda_i^{(R)} \leq \tilde{\Lambda}_i^{(R_2)}, \quad \forall R \in (R_1, R_2), \qquad (13)$$

where $R_1 \geq R_L$ and $R_2 \leq R_0$.

The continuity constraint can be expressed as

$$\frac{1 - R_0}{i \int \rho^{(R_L)}} \lambda_i^{(R_L)} \leq \tilde{\Lambda}_i^{(R_0)} - \tilde{\Lambda}_i^{(R_H)}. \qquad (14)$$

We assume that the mother code distribution is given, and the distribution at the highest rate R_H is fixed (the optimization on puncturing component rates is conducted before the optimization on shortening component rates). Then (13) and (14) can be applied to density evolution of any shortening component code rate within $[R_L, R_0)$. It should be noted that the code rate range is closed on the left and open on the right, because R_L is a rate subject to optimization, while the distribution at R_0 is prescribed. The concentrated row distribution $\rho^{(R)}$ is chosen so it maximizes the code rate in density evolution.

No known research focuses on the problem of simultaneously optimizing all code rates in the shortening code rate range. To define the optimality of a rate-compatible shortened LDPC code, we first discuss the existence of "dominant solutions."

Definition 1. A series of normalized variable degree distribution $\tilde{\Lambda}_D^{(R_L)}, \ldots, \tilde{\Lambda}_D^{(R_\alpha)}, \ldots, \tilde{\Lambda}_D^{(R_0)}$ is called dominant if it satisfies monotonicity and continuity, and for all $R \in [R_L, R_0)$, the corresponding iterative decoder converges at the highest Gaussian noise power, that is, $\sigma(\tilde{\Lambda}_D^{(R)}) \geq \sigma(\tilde{\Lambda}^{(R)})$, where $\tilde{\Lambda}^{(R_L)}, \ldots, \tilde{\Lambda}^{(R_\alpha)}, \ldots, \tilde{\Lambda}^{(R_0)}$ is any other series of normalized variable degree distribution that satisfies monotonicity and continuity.

If a dominant solution exists, Theorem 1 explains how to find it.

Therom 1. *If density evolution with the constraint*

$$\tilde{\Lambda}_i^{(R_L)} \leq \tilde{\Lambda}_{Di}^{(R)} \leq \tilde{\Lambda}_i^{(R_0)} - \tilde{\Lambda}_i^{(R_H)} \qquad (15)$$

yields a series of $\tilde{\Lambda}_D^{(R)}$ within $[R_L, R_0)$ that satisfy the monotonicity constraint, then this series of $\tilde{\Lambda}_D^{(R)}$ is a dominant solution as defined in Definition 1.

Proof. Distribution $\tilde{\Lambda}_D^{(R)}$ is obtained with the loosest monotonicity constraint that only considers boundary code rates. Therefore, its corresponding iterative decoder converges at equal or higher Gaussian noise power than any other feasible solution at rate R. □

Theorem 1 indicates that if a dominant solution exists, the above optimization process should yield at least one series of distributions that satisfies the monotonicity constraint. For the test mother code distribution, we try to individually optimize code rates of interest. However, the resulting series of distributions do not satisfy the monotonicity constraint, which suggests that at least for some cases, there is no dominant solution.

Without a dominant solution, we resort to a strategy that optimizes code rates close to R_L and those close to R_0 before it optimizes code rates close to $(R_L + R_0)/2$. Figure 2 was generated this way and our experiment shows that although suboptimal, this method nevertheless gives a good solution to the shortening component rates.

4. SIMULATION RESULTS

Bit error rate (BER) and frame error rate (FER) results for additive white Gaussian noise (AWGN) channels are shown in Figures 3 and 4, respectively. The degree distribution profile of the mother code is described by Figure 2. The mother code is generated by the ACE algorithm proposed in [9] with the further constraint that columns be allocated per the degree assignment of the previous section. The parity matrix is also constructed to have a semilower triangular form as this prevents stopping set activation due to parity puncturing.

The ACE algorithm [9] targets cycles in the bipartite graph corresponding to an LDPC code. The algorithm has two parameters, d_{ACE} and η_{ACE}. The design criterion is such that for all cycles of length $2d_{ACE}$ or less, the number of extrinsic edge connections (edges that do not participate in the cycle) is at least η_{ACE}. This approach increases the connectivity between any portion of the bipartite graph with the rest of the graph, and therefore prevents the occurrence of isolated cycles (cycles with poor variable node connectivity in the graph form stopping sets [9]). The ACE parameters achieved by the designed rate-compatible scheme are $d_{ACE} = 10$ and $\eta_{ACE} = 4$.

Figure 5 plots the proposed code performance (at BER = 10^{-5}) together with binary-input AWGN (BIAWGN) channel capacity threshold, the density evolution threshold, and the Shannon sphere-packing bound at FER = 10^{-4}. It should be noted that the density evolution threshold for punctured code rates $R > 0.5$ are borrowed from [5], and the density evolution threshold for shortened code rates are generated with the proposed optimization algorithm. The density evolution thresholds are achieved with Gaussian approximation
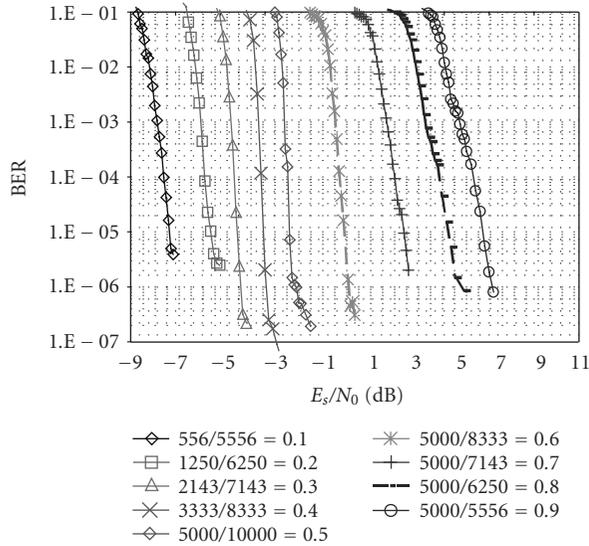
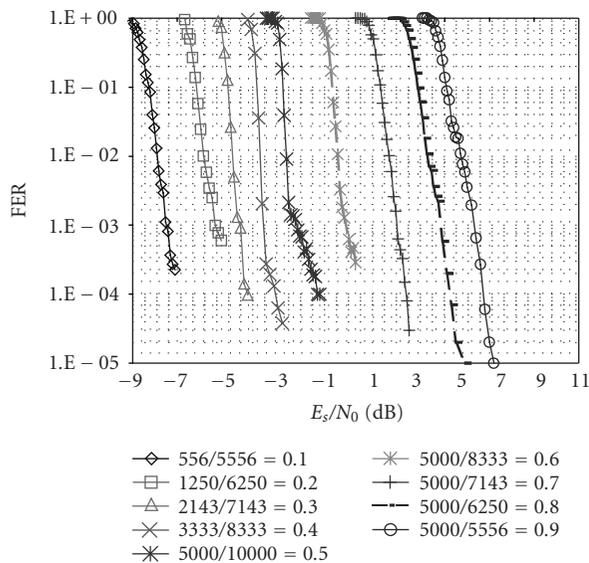FIGURE 3: BER simulation results and AWGN channel.

FIGURE 5: Code performance compared to theoretical bounds.

SNR to capacity at either rate extreme is approximately 3 dB at the designed block size.

5. CONCLUSION

A hybrid rate-compatible scheme for irregular LDPC codes that achieve good performance across a wide range of rates has been presented. The hybrid approach complements Ha and McLaughlin's puncturing technique by extending rate compatibility to the lower rate regime.

FIGURE 4: FER simulation results and AWGN channel.

at infinity block length, while the sphere-packing threshold is achieved with finite (n, k) pairs for generic BIAWGNC. Shannon sphere-packing bound is included here to account for the information-bits reduction for shortened codes, and the block-size reduction for punctured codes. We evaluate code performance at BER $= 10^{-5}$ instead of at FER $= 10^{-4}$ because some low rate (shortened) codes have error floors higher than FER $= 10^{-4}$.

The figure shows that the threshold degrades gracefully around $R_0 = 0.5$. For example, the simulation threshold SNR is 0.66 dB worse than the density evolution threshold for the mother code ($R_0 = 0.5$). This difference is 2.58 dB at $R = 0.1$ and 3.19 dB at $R = 0.9$, respectively. Therefore, the excess
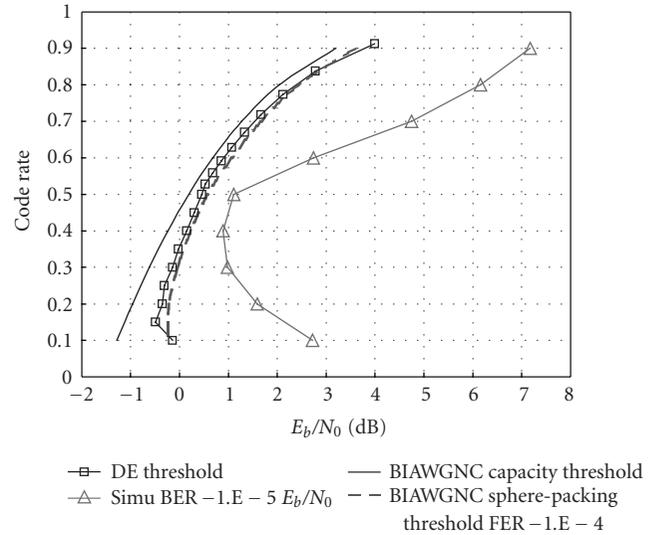
REFERENCES

[1] J. Hagenauer, "Rate-compatible punctured convolutional codes (RCPC codes) and their applications," *IEEE Trans. Commun.*, vol. 36, no. 4, pp. 389–400, 1988.

[2] A. S. Barbulescu and S. S. Pietrobon, "Rate compatible turbo codes," *IEE Electronics Letters*, vol. 31, no. 7, pp. 535–536, 1995.

[3] D. N. Rowitch and L. B. Milstein, "On the performance of hybrid FEC/ARQ systems using rate compatible punctured turbo (RCPT) codes," *IEEE Trans. Commun.*, vol. 48, no. 6, pp. 948–959, 2000.

[4] F. Babich, G. Montorsi, and F. Vatta, "Rate-compatible punctured serial concatenated convolutional codes," in *Proc. IEEE Global Telecommunications Conference (GLOBECOM '03)*, vol. 4, pp. 2062–2066, San Francisco, Calif, USA, December 2003.

[5] J. Ha, J. Kim, and S. W. McLaughlin, "Rate-compatible puncturing of low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 50, no. 11, pp. 2824–2836, 2004.

[6] H. Pishro-Nik and F. Fekri, "Results on punctured LDPC codes," in *Proc. IEEE Information Theory Workshop*, pp. 215–219, San Antonio, Tex, USA, October 2004.

[7] J. Li and K. R. Narayanan, "Rate-compatible low density parity check codes for capacity-approaching ARQ schemes in packet data communications," in *Proc. IASTED International Conference. Communications, Internet, and Information Technology (CIIT '02)*, pp. 201–206, St.Thomas, Virgin Islands, USA, November 2002.

[8] 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, 2004.

[9] T. Tian, C. R. Jones, J. D. Villasenor, and R. D. Wesel, "Selective avoidance of cycles in irregular LDPC code construction," *IEEE Trans. Commun.*, vol. 52, no. 8, pp. 1242–1247, 2004.

[10] S.-Y. Chung, T. J. Richardson, and R. L. Urbanke, "Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 657–670, 2001.

[11] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 619–637, 2001.

[12] T. J. Richardson and R. L. Urbanke, "Efficient encoding of low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 638–656, 2001.

[13] C. Di, D. Proietti, I. E. Telatar, T. J. Richardson, and R. L. Urbanke, "Finite-length analysis of low-density parity-check codes on the binary erasure channel," *IEEE Trans. Inform. Theory*, vol. 48, no. 6, pp. 1570–1579, 2002.

Tao Tian received B.S. degree from Tsinghua University, Beijing, China, in 1999, and M.S. and Ph.D. degrees from University of California, Los Angeles (UCLA) in 2000 and 2003, all in electrical engineering. From 2003 to 2004, he worked with MediaWorks Integrated Systems Inc. in Irvine, Calif. Since April 2004, he has been with QUALCOMM Incorporated in San Diego, Calif, where he works on problems related to multimedia signal processing and communications.

Christopher R. Jones received B.S., M.S., and Ph.D. degrees in electrical engineering from University of California, Los Angeles (UCLA) in 1995, 1996, and 2003. From 1997 to 2002, he worked with Broadcom Corporation in the area of VLSI architectures for communications systems. He has been with the Jet Propulsion Laboratory in Pasadena since January 2004 where he works on problems related to iterative coding.