GLOBAL DESIGN METHODS FOR RAPTOR CODES USING BINARY AND HIGHER-ORDER MODULATIONS

Richard J. Barron, Caleb K. Lo, Jerome M. Shapiro

MITRE Corporation {rbarron, clo, jshapiro}@mitre.org

ABSTRACT

We develop accurate and robust one-dimensional EXIT chart methods to design Raptor codes with binary and higher-order modulations on AWGN channels. We formulate the checknode degree distribution optimization as a linear program that directly accounts for the pre-code error correction capability and simultaneously optimizes over any desired range of rate operating points. The throughput curves predicted by the analysis are easily compared across different pre-code rates and pre-code error-correcting capabilities. For higher-order modulations we employ BICM on a standard Gray-mapped 16-QAM constellation. We further enhance the optimization by applying separate check-node degree distributions to the mapped MSBs and LSBs, which results in significant gains compared to a single check-node degree distribution.

1. INTRODUCTION

Fountain codes are a class of codes for which the encoder produces a virtually endless stream of bits and the decoder may use any unordered collection of channel bits to decode a codeword. Fountain codes are rateless codes, i.e. they can dynamically transmit any rate R = k/n, where k is the fixed input block size and n is the number of bits transmitted. A good rateless code is one that attains capacity of the channel without any prior knowledge of the channel parameters, e.g.the channel SNR. Any rateless code does not necessarily have the aforementioned fountain property, i.e. decoding can be done on any unordered collection of bits. For example, an incremental redundancy code is a rateless code that requires that codeword bits be received in a specific order starting from a particular start bit.

The fountain property is particularly useful in applications where a receiver or multiple receivers listen to the channel at different start times and/or intermittently. Fountain codes were originally applied to digital networks subject to dropped packets, for which the erasure channel model applies. Indeed, raptor codes which are a class of fountain codes, have been shown to approach the capacity of the erasure channel asymptotically. More recently fountain codes have been developed for wireless applications including rateless pointto-point hybrid-ARQ; rateless multi-cast; distributed network storage[1]; and collaborative relay networks [2].

Raptor codes have also been applied to the AWGN and Rayleigh channels [3, 4], but there has been little progress on universal design methods for these channels. Several proposed methods are accurate only over a certain narrow range of SNRs [3, 5, 6]. The Extrinsic Information Transfer (EXIT) method originally introduced by ten Brink [7] has proven to be a highly accurate and robust design tool for LDPC and IRA codes [8]. The method has produced fixed-rate codes with thresholds within 0.1dB of capacity. Raptor codes are amenable to the same analysis and design [9]. Existing designs for raptor codes have been limited to a fixed SNR, which is at odds with the signature feature of fountain/rateless codes that they operate over virtually any SNR/rate. Our contribution is to define the optimization problem using EXIT chart analysis so that performance is optimized over any desired range of SNRs/rates. Furthermore, we directly incorporate the error-correcting capability of the pre-code into the optimization. We simulate the throughput of designed codes and show improved performance compared to previous results. The throughput curves predicted by the analysis are easily compared across different pre-code rates and pre-code error-correcting capabilities and are useful for exploring various design tradeoffs.

There have been some attempts at applying raptor codes to higher order modulation with limited success [10]. Bitinterleaved coded modulation (BICM) is a highly effective method for coding over high-order constellations. We design spectrally-efficient fountain codes using raptor-coded BICM designed by our EXIT chart methods. We further enhance the optimization by applying separate check-node degree distributions to the mapped MSBs and LSBs, which results in significant gains compared to a single check-node degree distribution.

2. LT AND RAPTOR CODES

The Tanner graph for the raptor code is shown in Figure 1. A raptor code is a pre-coded LT code. The payload bits are encoded by the pre-code of rate $R_{\rm pre}$ to form what are called

Approved for Public Release; Distribution Unlimited. 09-1879



Fig. 1. Tanner graph of pre-code and LT code. Circles represent variables and squares represent checks. The variable $x^{(l)}$ represents the mutual information of the message passed from the check nodes to the variable nodes at iteration l of sum product decoding.

input bits. The LT code draws d bits in a uniformly random manner from the set of input bits and computes their parity. The outputs of the parity checks are called *output bits* and they are put on the channel. The value for d is drawn from a distribution called the check-node degree distribution for each new output bit. The more output bits that are put on the channel, the lower the rate of the code. We see from Figure 2 that the Tanner graph for the LT code is a bi-partite graph comprised of bit nodes and check nodes, which makes it amenable to sum-product decoding, similar to LDPC code decoding.

The check-node degree distribution is denoted Ω_i , $i = 1, 2, ..., d_c$, and it represents the probability that a check node is degree *i*. A convenient representation of the distribution is by the degree polynomial

$$\Omega(x) = \sum_{i=1}^{d_c} \Omega_i x^i.$$

The check-node degree distribution can be defined with respect to the nodes themselves as with $\Omega(x)$ or with respect to the edges. We let ω_i , $i = 1, 2, ..., d_c$ represent the check-node degree distribution with respect to the edges, i.e., the probability that a given edge is of degree *i*. The degree polynomial with respect to the edges is defined by

$$\omega(x) = \sum_{i=1}^{d_c} \omega_i x^{i-1}$$

Degree distributions are similarly defined with respect to the input bit nodes. We denote by $\Lambda(x)$ and $\lambda(x)$ the degree polynomials for the input bits with respect to the nodes and the edges, respectively. Recall that the edges to the input bit nodes are created by selecting bits in a uniformly random manner. Etesami [3] proves that in the limit of a large number of input bits k, we have $\Lambda(x) = \lambda(x) = e^{\alpha(1-x)}$, where α is the average degree of the input nodes. By Taylor expansion, we see that Λ is Poisson distributed in this limit. We will be assuming that Λ is Poisson distributed for the remainder of this paper. The formulas for converting between the node-view and the edge-view distributions and vice versa are straighforward to derive: $\omega_i = \frac{i\Omega_i}{\sum_j j\Omega_j}$

and

$$\Omega_i = \frac{\omega_i/i}{\sum_j \omega_j/j}.$$
(1)

Letting $R_{\text{LT}} = k/n$ be the rate of the LT code where *n* is the number of check nodes, we have

$$R_{\rm LT} = \frac{\sum_j \lambda_j / j}{\sum_j \omega_j / j}$$
(2)

$$= \frac{\sum_{j} j\Omega_{j}}{\sum_{j} j\Lambda_{j}} = \frac{\beta}{\alpha},$$
(3)

where β denotes the average check node degree. The check node degree distribution is fixed, and hence β is fixed. In rateless mode therefore, from the perspective of degree distributions, the rate of the LT code is reduced solely by increasing the input node degree α . The rate of the raptor code is $R = R_{\rm pre}R_{\rm LT}$

3. EXIT CHART TECHNIQUE

Iterative decoders pass extrinsic information messages between constituent elements of the code. For turbo codes the messages are computed by the BCJR algorithm running on turbo codes. For LDPC-like codes consisting of a bipartite graph with bit nodes and check nodes – an LT code is one of these – the messages are computed by the sum-product algorithm.

The sum-product algorithm is run in iterations, where l is the iteration index. At each iteration, an input bit node uses incoming messages from the output bits to compute an outgoing message on each of its edges to corresponding output bits. For the *i*th input bit node of degree d we denote the outgoing message from the input bit to the output bit by $m_{i,o}^{(l)}$, o = 1, 2, ..., d, where o is the outgoing edge index. The outgoing messages from the output bits to the input bits in round l are denoted similarly by $m_{o,i}^{(l)}$. These outgoing messages are computed as follows:

$$m_{o,i}^{(l)} = 2 \tanh\left(\tanh\left(\frac{Z_o}{2}\right) \cdot \prod_{i' \neq i} \tanh\left(\frac{m_{i',o}^{(l+1)}}{2}\right)\right) \quad (4)$$
$$m_{i,o}^{(l+1)} = \sum_{o' \neq o} m_{o',i}^{(l)}, \quad (5)$$

where Z_o is the log-likelihood ratio (LLR) of the output bit determined from the channel measurement. At the last iteration *l* the input bit LLR is computed by $\sum_{o} m_{o,i}^{(l)}$. The fundamental assumption of EXIT chart analysis is that the messages passed between nodes have a symmetric Gaussian

$$x^{(l)}\left(x^{(l-1)}\right) = 1 - \sum_{i=1}^{d_c} \omega_i J\left(J^{-1}\left(1 - J\left(2/\sigma^2\right)\right) + (i-1)J^{-1}\left(1 - \sum_{j=1}^{d_v} \lambda_j J\left((j-1)J^{-1}\left(x^{(l-1)}\right)\right)\right)\right).$$
(10)

distribution $\mathcal{N}(\mu, 2|\mu|)$ for some mean μ . A second important assumption is that the bipartite graph for the LT code can be represented by a tree graph. From (5) we see that for the bit nodes, by the sum rule, symmetric Gaussian input messages will yield symmetric Gaussian output messages.

At the check nodes, (4) does not indicate that input Gaussian messages will create Gaussian output messages, and indeed this is not the case. However, empirically the Gaussian approximation has proven to accurately model the mutual information I of the messages passed between nodes, i.e. the mutual information between the message on an edge and a channel input that is equally likely a zero or a one. EXIT chart analysis views iterative decoding as an evolution of the message mutual informations. When the mutual information equals one, there are no bit errors.

For a symmetric Gaussian distribution, mutual information J is parameterized by its mean $\mu > 0$:

$$J(\mu) = \frac{1}{\sqrt{4\pi\mu}} \int_{-\infty}^{\infty} e^{\frac{-(y-\mu)^2}{4\mu}} \log_2\left(1+e^{-y}\right) dy.$$
 (6)

Eq.(6) is monotonically increasing in μ and defines a one-toone mapping between any distribution and a symmetric Gaussian distribution defined by μ . For the remainder of this paper we will assume that all message distributions are symmetric Gaussian.

We denote the mutual information on an edge incoming to a node by I_A and the mutual information outgoing from a degree d node by $I_E(d)$. For a bit node, we see from the sum rule in (5) that

$$I_E(d) = J((d-1)J^{-1}(I_A)).$$
(7)

At the check nodes EXIT chart analysis employs the reciprocal channel approximation (RCA), which states that from a mutual information perspective we can view the check node as a bit node with new incoming mutual informations $1 - I_A$ and outgoing mutual informations $1 - I_E$ [11]. Hence for a degree d check node, from the RCA and (5) we have the relation

$$I_E(d) = 1 - J((d-1)J^{-1}(1-I_A)).$$
(8)

For both the bit nodes and the check nodes the mutual information for any outgoing edge is the average mutual information over the degrees:

$$I_E = \sum_{i=1}^{d_v} \lambda_i I_E(i) \tag{9}$$

We denote by $x^{(l)}$ the mutual information outgoing from the check nodes at iteration l (illustrated in Fig.1). Using the RCA and equations (7)-(9) we have the mutual information evolution equation for the LT code given in (10), where $2/\sigma^2$ is the mean LLR value off the channel of noise variance σ^2 . We note that (10) is linear in the coefficients of $\omega(x)$.

We define the function $\phi(x) = x^{(l)}(x)$. At any point x, $0 \le x \le 1$, for which $\phi(x) = x$, we have a fixed point in the evolution. As long as $x^{(l)}(x) > x$, $0 \le x \le x_0$ for some fixed point x_0 , then we are assured that mutual information will evolve up to x_0 . The evolution of mutual information essentially follows a tunnel between the function $\phi(x)$ and identity function x, $0 \le x \le x_0$. Figure 2 shows a plot of $\phi(x)$ at a channel SNR that is 0.9 dB from capacity using

$$\Omega(x) = 0.05x + 0.5x^2 + 0.25x^4 + 0.05x^6 + 0.1x^8.$$
(11)

Note that the simulated mutual information trajectory for the LT code tracks $x^{(l)}(x)$ remarkably closely, exhibiting the claimed accuracy of the EXIT chart analysis.

The fixed point of the trajectory in Fig.2 is $x^{(l)} = 0.387 < 1$, and yet the codeword is decoded without error. The reason for this is that $x^{(l)}$ represents the extrinsic information for the check bits. The process of summing messages at the bit nodes amplifies the mutual information of the input bits, denoted $y^{(l)} = \gamma(x^{(l)})$, where

$$\gamma(x) = \sum_{i=1}^{d_v} i\Lambda_i J^{-1}(x).$$
(12)

For the scenario of Fig.2, $y^{(l)} = 0.919$ at the end of the iterations, which is enough mutual information for the pre-code of rate-0.9 to decode without error. Based on the rate of the outer code $R_{\rm pre}6$, we can determine the required fixed point of the iterations for $x^{(l)}$. Note that increasing the average input bit degree α increases the amplification of (12), but decreases the rate of the LT code.

We denote the precode mutual information decoding threshold by $t(R_{\rm pre})$, which is either determined by simulation or EXIT chart analysis. The messages feeding the pre-code are not necessarily Gaussian (although they are assumed to be) and they are not identically distributed, so a mutual information threshold may not necessarily be sufficient to guarantee decoding, but we still make this reasonable assumption.





Fig. 2. EXIT chart of $\Omega(x) = 0.05x + 0.5x^2 + 0.25x^4 + 0.05x^6 + 0.1x^8$: Rate 0.9 pre-code; k = 65000; n = 139100; SNR is 0 dB which is 0.9 dB from capacity. The stair step trajectory is the output of an actual LT simulation with these parameters.

4. LINEAR PROGRAMMING FORMULATION

Recall from (2) that for a fixed α the rate of the LT code is inversely proportional to $\sum \omega_i/i$. Moreover, $\phi(x)$ is a linear function of the ω_i s. Thus for a fixed channel SNR and fixed α we maximize the rate of the LT code by the following linear program (LP):

maximize
$$\sum_{i=1}^{d_c} \omega_i / i$$
 (13)

subject to the constraints:

$$\begin{split} &(\mathbf{C}_1) \ \sum_{i=1}^{d_c} \omega_i = 1, \, \omega_i \geq 0, \forall i \\ &(\mathbf{C}_2) \ \phi(x) > x + \epsilon \ \forall x \in [0, \theta \gamma^{-1}(t(R_{\text{pre}}))] \text{ for some } \epsilon > \\ &0, 0 < \theta \leq 1 \end{split}$$

(C₃)
$$\phi(x) > x \, \forall x \in (\theta \gamma^{-1}(t(R_{\rm pre})), \gamma^{-1}(t(R_{\rm pre}))]$$

Condition (C₂) is used to enforce a minimum tunnel spacing ϵ for a fraction θ of the required tunneling interval. Note that the condition at x = 0 is necessary to get the decoder started. Condition (C₃) is used to enforce the condition that tunnel stay open over the remaining fraction of the required interval. In contrast to (C₂) there is no minimum tunnel separation enforced, which allows the tunnel to taper at the end of the interval. Note that the LP in (13) is guaranteed to be good only for a single SNR and rate, which is at odds with the signature feature of fountain codes.

Let $C(\sigma^2)$ denote the BPSK channel capacity as a function of channel variance. We define a set of rates $\tilde{\mathcal{R}} = \{\tilde{R}_i\}$ j = 1, 2, ..., J corresponding to a set of channel variances $S = \{\sigma_i^2\} \ j = 1, 2, ..., J$, defined by

$$\sigma_j^2 = C^{-1}(\tilde{R}_j) / \Delta$$

where $\Delta > 1$ is some fixed gap to capacity. The rates in \mathcal{R} are essentially target rates for the optimization which returns the actual rates $\mathcal{R} = \{R_j\} \ j = 1, 2, ..., J$. The gap Δ could be a function of j but we usually consider a fixed gap between 0.5dB and 1.5dB, depending how close we think the raptor code will be to capacity.

Once the check node distribution $\omega(x)$ is selected, it defines the LT code and it applies to all rates and SNRs. It follows from this fact and (3) that α is inversely proportional to R. We define the set $\mathcal{A} = {\alpha_i} j = 1, 2, ..., J$ where

$$\alpha_j = \frac{\beta}{R_{\rm pre}\tilde{R}_j} = \frac{\beta_0}{\tilde{R}_j},\tag{14}$$

where $\beta_0 = \beta/R_{\text{pre}}$ is a free parameter. Indeed α_j will be a function of the actual rates R_j returned by the optimization, but defining the α_j s by (14) turns out to be very accurate. We express the implicit dependence of the tunneling function $\phi(x)$ on α_j and σ_j^2 by the notation $\phi_j(x)$.

We define a global objective function $f(\mathcal{R}; \tilde{\mathcal{R}})$ subject to maximization that treats low-rate codes and high-rate codes equally by weighting the rates by the inverse of their target rates:

$$f(\tilde{\mathcal{R}}, \mathcal{R}) = \sum_{j=1}^{J} \frac{R_j}{\tilde{R}_j}.$$
 (15)

Each term in the sum reduces as follows:

$$\frac{R_j}{\tilde{R}_i} = \frac{R_{\rm pre} \sum_i \lambda_{j,i}/i}{\tilde{R}_j \sum_i \omega_i/i} \tag{16}$$

$$= \frac{R_{\rm pre} \left(1 - e^{-\alpha_j}\right)}{\tilde{R}_j \alpha_j \sum_i \omega_i / i} \tag{17}$$

$$\approx \frac{R_{\rm pre}}{\tilde{R}_j \alpha_j \sum_i \omega_i / i} \tag{18}$$

$$= \frac{R_{\rm pre}}{\beta_0 \sum_i \omega_i / i},\tag{19}$$

where (17) follows from

$$\sum_{i} \lambda_i / i = \int_0^1 \lambda(x) dx = \frac{1}{\alpha} \left(1 - e^{-\alpha} \right)$$
(20)

for $\lambda(x) = \exp(\alpha(1-x))$; (18) is a highly accurate approximation by the fact that α is always greater than 3 for the highest rates in empirical studies; and (19) uses (14).

From (19)we see that we are justified to use the slightly modified objective function:

$$\hat{f}(\omega(x)) = \frac{1}{\sum_{i} \omega_i / i},$$
(21)

$$\phi(x) = 1 - \sum_{i=1}^{d_c} \omega_i^{MSB} J\left(J^{-1} \left(1 - I_{MSB}(\text{SNR})\right) + (i-1)J^{-1} \left(1 - \sum_{j=1}^{d_v} \lambda_j J\left((j-1)J^{-1}(x)\right)\right)\right) - \sum_{i=1}^{d_c} \omega_i^{LSB} J\left(J^{-1} \left(1 - I_{LSB}(\text{SNR})\right) + (i-1)J^{-1} \left(1 - \sum_{j=1}^{d_v} \lambda_j J\left((j-1)J^{-1}(x)\right)\right)\right)$$
(24)

which turns out to be the same objective function used in (13). Thus, the only modification to the LP is to the constraints:

- $\begin{array}{l} ({\rm C}_2') \ \ \phi_j(x) > x + \epsilon \ \forall x \in [0, \theta \gamma^{-1}(t(R_{\rm pre}))], \mbox{ for some } \epsilon > \\ 0, 0 < \theta \leq 1, j = 1, 2, ..., J \end{array}$
- $(\mathbf{C}_3') \ \phi_j(x) > x \ \forall x \in (\theta \gamma^{-1}(t(R_{\mathrm{pre}})), \gamma^{-1}(t(R_{\mathrm{pre}}))], j = 1, ..., J$

We have observed empirically that the LP optimized objective function is unimodal in the free parameter β_0 . Thus, we find the globally optimal solution in a straighforward manner by searching the LP optimized objective function over β_0 ; each $beta_0$ requires the solution of new LP.

5. BIT-INTERLEAVED CODED MODULATION

In this section we extend our design tools to perform with BICM. We focus on standard gray-coded 16-QAM, which is equivalent to gray-coded 4-PAM mapped independently to the I and Q components. The extension of this technique to higher-order constellations and arbitrary mappings is straightforward. The bit pairs {01, 00, 10, 11} are mapped to the constellation points {-3, -1, +1, +3}. We see that the MSBs have more channel protection than the LSBs. Fig. 3 shows the mutual information attained by the MSBs and by the LSBs – $I_{MSB}(SNR)$ and $I_{LSB}(SNR)$, respectively – as a function of SNR on the AWGN channel; the comparison shows the dramatic effect of the unequal error protection. Using EXIT chart analysis, we design separate check distributions for the MSB output bits and the LSB output bits.

We define the degree polynomial

$$\omega(x) = \omega^{MSB}(x) + \omega^{LSB}(x), \qquad (22)$$

where $\omega^{MSB}(x)$ and $\omega^{LSB}(x)$ are not valid degree polynomials, because their coefficients do not add up to one. We constrain $\omega^{MSB}(x)$ and $\omega^{LSB}(x)$ so that half of the check nodes are MSBs and half are LSBs. Using (1) we have

$$\frac{\sum_{i} \omega_{i}^{MSB}/i}{\sum_{i} \omega_{i}/i} = \frac{1}{2}$$
$$\frac{\sum_{i} \omega_{i}^{LSB}/i}{\sum_{i} \omega_{i}/i} = \frac{1}{2},$$



Fig. 3. Plots of $I_{MSB}(SNR)$ and $I_{LSB}(SNR)$. Their sum equals the BICM capacity and is also plotted.

which yield

$$\sum_{i} \omega_i^{MSB} / i - \sum_{j} \omega_j^{LSB} / j = 0$$
 (23)

Again using the same objective function (21) that we have been using throughout this paper, we develop an LP that maximizes it subject to new and old constraints. The only new constraint is (23). In (24) we rewrite (10) to take into account the two polynomials $\omega^{MSB}(x)$ and $\omega^{LSB}(x)$ and the differing channel mutual informations. We define $\phi_j(x)$ in the same manner as above using the new definition for $\phi(x)$. The LT proceeds by using constraints (C₁), (C'₂), (C'₃), (C'₄), and (23).

6. SIMULATIONS

In this section we simulate fountain codes in rateless operation, and compute throughput curves as a function of channel SNR. At any given SNR, the number of output bits n required to decode varies from codeword to codeword. Throughput is therefore computed as $T = k/\bar{n}$, where \bar{n} is the average number of received output bits per codeword. We used k = 65000

for all simulations and used 50 iterations for the LT code and a maximum of 100 iterations for the LDPC code.

6.1. Binary codes

We simulated three raptor codes applied to the binary-input AWGN channel. The pre-codes were (3,30), (3,15), (3,60) regular LDPC codes with rates of $R_{\rm pre} = 0.9$, $R_{\rm pre} = 0.8$, and $R_{\rm pre} = 0.95$, respectively. The first two pre-codes were used in [?] with an LT distribution given by (11), and we use these raptor codes as a baseline against which to compare our codes. To highlight the utility of our design technique we also design our codes with a high outer code rate $R_{\rm pre} = 0.95$, for which (11) is ineffective.

The check distributions were optimized with the algorithm developed in this paper for rates 0.2, 0.3, 0.4, 0.5, 0.6, and 0.7 bits/dimension. The designed LT check distributions are as follows:

$$\begin{split} \Omega_{0.9}(x) &= 0.06756x + 0.38357x^2 + 0.34884x^3 + \\ &\quad 0.00820x^4 + 0.01266x^8 + 0.17917x^9, \\ \Omega_{0.8}(x) &= 0.051x + 0.47x^2 + 0.242x^3 + \\ &\quad 0.237x^5, \\ \Omega_{0.95}(x) &= 0.08627x + 0.36680x^2 + 0.32288x^3 \\ &\quad + 0.15562x^7 + 0.02294x^{19} + 0.04548x^{20} \end{split}$$

with β_0 equal to 0.74, 0.54, and 0.87, respectively.

The throughput curves corresponding to the first two precodes are shown in Figs.4 and 5, respectively. For comparison purposes, we also simulated the same pre-codes with the check distribution given by (11), which was presented as a heuristic design in [6]. Interestingly, for $R_{\rm pre} = 0.9$ our results lie virtually on top of the heuristic design up to rate-0.7, which we believe is an indication that the heuristic design is a good one. Note that the performance of our code tails off at higher SNRs/rates because the optimization was only performed up to rate-0.7. The $R_{\rm pre} = 0.8$ raptor code shows significant throughput improvement over the heuristic design.

Fig. 6 shows the throughput curves for $R_{\rm pre} = 0.95$ for both the design and for the heuristic LT code given by (11). The heuristic is clearly ineffective at these high pre-code rates, as the performance is even worse than the $R_{\rm pre} = 0.9$ case. The design case has very minor improvement at medium and low rates, but is clearly superior at high rates. We expect this behavior at high rates as the high-rate precode system saturates at a higher rate.

6.2. BICM

We simulated raptor-coded BICM with a gray-mapped 16-QAM and a $R_{\rm pre} = 0.9$ turbo code, the throughput results for which are shown in Fig. 7. The reason for using a turbo code was that it permitted easy encoding; one cannot transmit the all-zeros codeword to obtain simulated throughput. As



Fig. 4. Throughput simulation plot for raptor code on binary input AWGN channel. $R_{\text{pre}} = 0.9$. k = 65000. Heuristic is from (11).



Fig. 5. Throughput simulation plot for raptor code on binary input AWGN channel. $R_{\text{pre}} = 0.8$. k = 65000. Heuristic is from (11).



Fig. 6. Throughput simulation plot for raptor code on binary input AWGN channel. $R_{\text{pre}} = 0.95$. k = 65000. Heuristic is from (11).

a comparison we also used the heuristic degree distribution in (11) to code the BICM. We note improved throughput at lower SNRs, but little or no improvement at high SNRs. We postulate that the smaller effect at higher SNRs is due to the fact that the LSB and MSB have little difference in mutual information at high SNR.

7. CONCLUSION

We developed a highly accurate and robust design method based on EXIT charts for designing raptor codes for BPSK and BICM modulations. We derived computationally simple linear programs that optimized rates over any desired range of SNRs. We further generalized the algorithm to assign different distributions to different bits for higher-order modulations.

8. REFERENCES

- S. Aly, Z. Kong, and E. Soljanin, "Fountain Codes Based Distributed Storage Algorithms for Large-scale Wired Sensor Networks," *International Conference on Information Processing in Sensor Networks*, 2008.
- [2] A. Molisch, N. Mehta, J. Yedidia, and J. Zhang, "Performance of fountain codes in collaborative relay networks," *IEEE Trans. on Wireless Communications*, November 2007.
- [3] O. Etesami and A. Shokrollahi, "Raptor codes on inary memoryless symmetric channels," *IEEE Trans. Inform. Theory*, May 2006.
- [4] Palanki, R.; Yedidia, J.S., "Rateless Codes on Noisy Channels", *IEEE International Symposium on Information Theory (ISIT)*, June 2004.



Fig. 7. Throughput simulation plot for raptor-encoded BICM with gray-mapped 16-QAM. $R_{\rm pre} = 0.9$. k = 65000. Heuristic is from (11).

- [5] Cheng, Zhong; Castura, Jeff; Mao, Yongyi, "On the Design of Raptor Codes for Binary-Input Gaussian Channels," *ISIT 2007*, June 2007.
- [6] E. Soljanin, N. Varnica, and P. Whiting, "Punctured vs Rateless Codes for Hybrid ARQ," *IEEE Inform. Theory Workshop*, Punta del Este, Uruguay, March 13-16, 2006.
- [7] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Trans. on Communications*, October 2001.
- [8] A. Roumy, S.Guemgar, G. Caire, and S. Verdu, "Design methods for irregular repeat-accumulate codes",*IEEE Trans. Inform. Theory*, August 2004.
- [9] A. Venkiah, C. Poulliat, and D. Declercq, "Analysis and design of raptor codes for joint decoding using information content evolution," *ISIT 2007*, June 2007.
- [10] Castura, J, Yongyi Mao, "Rateless coding and modulation scheme for unknown Gaussian channels," Information Theory, 2007. CWIT '07. 10th Canadian Workshop on Information Theory, June 2007.
- [11] Alexei E. Ashikhmin, Gerhard Kramer, Stephan ten Brink, "Extrinsic information transfer functions: model and erasure channel properties." *IEEE Transactions on Information Theory*, 2004.