On the Design of Raptor Codes for Binary-Input Gaussian Channels

Zhong Cheng, Jeff Castura, and Yongyi Mao, *Member, IEEE*

Abstract

This paper studies the problem of Raptor-code design for binary-input AWGN (BIAWGN) channels using the mean-LLR-EXIT chart approach presented in [1]. We report that there exist situations where such a design approach may fail or fail to produce capacity-achieving codes, for certain ranges of channel SNR. Suggestions and discussions are provided pertaining the design of Raptor codes for BIAWGN channels using the mean-LLR-EXIT chart.

I. Introduction

The great success of fountain codes [2], [3] — including LT codes and Raptor codes — over erasure channels has inspired reviving interest in the use of incremental redundancy schemes [4] and the idea of "rateless coding" [5], [6] for communication under channel uncertainty [7]. In particular, beyond their applications in erasure channels, Raptor codes have recently been applied in various noisy channel models including binary symmetric channels, AWGN channels and fading channels [1], [6], [8], as well as in multi-user channels such as wireless relay channels [9]. A milestone in the analysis of Raptor codes over noisy channels is the work of [1], where properties of capacity-achieving Raptor codes are derived and additionally a framework of code construction is presented. This paper follows up on the results of [1] and studies the design issues of Raptor codes using the method in [1].

As Raptor codes are extensions of LT codes to including an LDPC precode and their decoding is on the factor-graph representation of the global code by Belief Propagation (BP), a key of designing a Raptor code is to determine the distribution of the output-symbol degrees in the factor graph. A fundamental result of [1] is that over noisy channels (like BIAWGN channels) there exists no universally capacity-achieving output degree distribution. This implies that for BIAWGN channels the output degree distribution of a Raptor code needs to be designed based on the channel SNR. The authors of [1] then further suggest a design methodology for determining the output degree distribution based on the extrinsic information transfer (EXIT) chart [10], similar to those for LDPC codes [11], [12]. It is however worth noting that unlike its counterparts in LDPC codes (e.g., [12]), the EXIT chart of [1] uses the expected value of the log-likelihood-ratio (LLR) in the messages (under certain Gaussianality assumption) and tracks its trajectory over BP iterations. For this reason, we refer to the EXIT chart of [1] as the mean-LLR-EXIT chart or simply EXIT chart in short. Under this framework, the authors of [1] show that the design of a Raptor code — in fact more precisely, of its output-symbol degree distribution — is a linear programming problem, whereby efficient algorithms exist.

This paper follows up on the EXIT chart approach of [1] for constructing Raptor codes. In particular, we are interested in effectiveness of this approach for various values of channel SNRs, in terms of whether it is capable of producing capacity-achieving codes. The main results of this paper is summarized as follows.

We prove that for any given parameter setting in the linear program, there exist two bounds of channel SNR — $\mathrm{SNR}^*_{\mathrm{low}}$ and $\mathrm{SNR}^*_{\mathrm{high}}$ (depending on the parameter setting of the linear program) — such that when the channel SNR is lower than $\mathrm{SNR}^*_{\mathrm{low}}$, the linear program fails to give a solution, and when the channel SNR is higher than $\mathrm{SNR}^*_{\mathrm{high}}$, the constructed Raptor codes necessarily fail to achieve the channel capacity. For channel SNR between these two bounds, we show by simulations that for relatively low channel SNR, the EXIT-chart based linear program is capable of producing capacity-achieving Raptor codes, but as SNR increases, the constructed Raptor codes perform at a visible gap away from the capacity. We also present suggestions for constructing Raptor codes based on the EXIT-chart approach.

The remainder of this paper is organized as follows. To be self-contained, in Section II, we first introduce Raptor codes with its decoding algorithm. Then in Section III, we review the main results of [1], upon which this paper primarily follows up. In Section IV, we present the two SNR bounds and their implications on the design of Raptor codes. In Section V, we report a simulation study of codes designed for channel with SNR between the two bounds, and present a detailed discussion on our observations. The paper is briefly concluded in Section VI.

II. Raptor Codes

A binary Raptor code is a concatenation of an LDPC precode $\mathtt{C}$ with an LT code. In the LDPC precode, a k-bit information vector is first mapped to a k'-bit codeword of $\mathtt{C}$, where the codeword bits are usually referred to as the *input symbols*. Then via the LT code, a randomly selected fraction of input symbols are used to generate a new bit via the XOR operation and as this process repeats, a potentially infinite stream of bits — usually referred to as the *output symbols* — are generated and transmitted. The process terminates when the receiver is able to decode the k-bit information vector and signals an ACK via a feedback channel. Fig. 1 shows an example of Raptor code represented by a factor graph, in which each c_i', $i \in \{1, 2, \dots, k'\}$, represents an input symbol, and each c_j, $j \in \{1, 2, \dots, n\}$, represents an output symbol. A compact form of the factor graph is shown in Fig. 1(b), where the output symbol c_j has been identified with its connected parity check. Throughout this paper, Raptor code factor graphs will be represented using this form.

A Raptor code is parameterized by $(k, \mathtt{C}, \Omega(x))$, where $\Omega(x)$ is the output-symbol degree distribution. Specifically $\Omega(x) := \sum_{d=1}^{\infty} \Omega_d x^d$, where Ω_d is the fraction of output symbols with degree d. A related notion of output degree distribution is $\omega(x) := \sum_{d=1}^{\infty} \omega_d x^{d-1}$, where ω_d is the fraction of edges in the factor graph of the Raptor code connecting to a degree-d output symbol. It is then easy to show that $\omega(x) = \Omega'(x)/\Omega'(1)$, or $\omega_d = d\Omega_d / \sum_d d\Omega_d$.

On the factor-graph representation of Raptor codes, the sum-product algorithm can be easily developed for memoryless channels. Over binary input memoryless channels, let $y^n := (y_1, y_2, \dots, y_n)$ be the channel output vector up to time n, where for every $j \in \{1, 2, \dots, n\}$, y_j is the channel output for symbol c_j. The channel LLR message of c_j is defined as

$$m_{0,j} := \log \frac{P(c_j = 0|y_j)}{P(c_j = 1|y_j)}, \tag{1}$$

where throughout this paper, logarithm takes base e.

The sum-product algorithm operates in an iterative way where messages are passed bidirectionally along each edge between the neighboring input symbols and output symbols. One may follow the schedule of message passing on the global factor graph involving both the LT part and the LDPC part. For the ease of analysis, we consider a schedule that first passes messages in the LT part of the graph and upon convergence passes messages in the LDPC part of the graph. For message passing in the LT part of the graph, in each iteration, messages are first

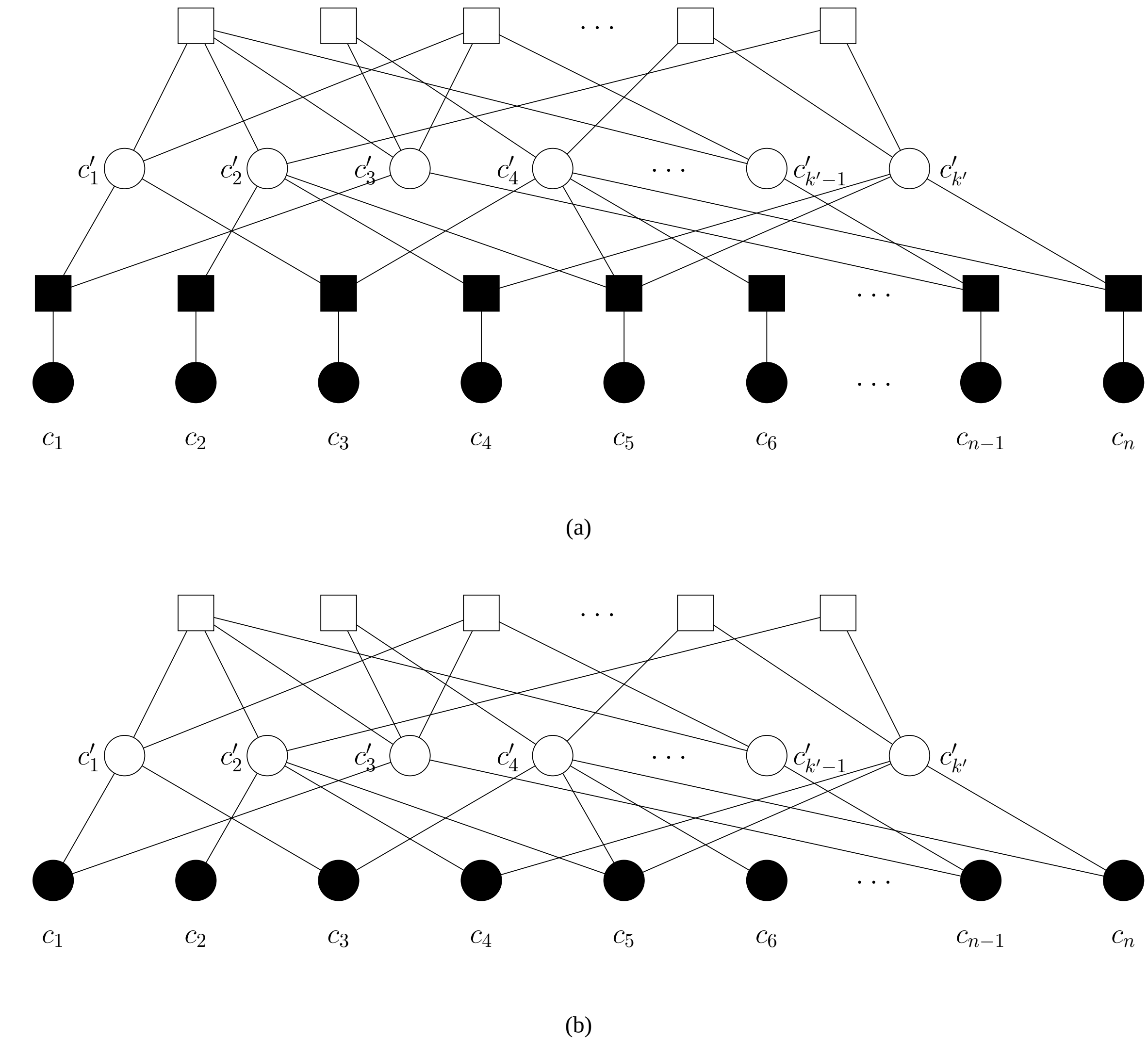

Fig. 1. (a) The standard factor graph of a Raptor code truncated at block length n (white squares represent the parity checks of LDPC code, white circles represent LDPC codeword bits, black squares represent the parity checks of LT code, and black circles represent Raptor codeword bits). (b) The compact form of the factor graph in (a).

passed from output symbols to input symbols, and then from input symbols to output symbols. At iteration l, we denote the message passed from output symbol c_j to input symbol c'_i by $m^{(l)}_{j\to i}$, and the message passed from input symbol c'_i to output symbol c_j by $m^{(l)}_{i\to j}$. In each iteration, each symbol passes messages to its neighbors along its edges. For convenience, we denote the set of neighbors of a particular node v by $\mathcal{N}(v)$. Then the message passed from output symbol

c_j to input symbol c'_i at each iteration l is

$$m_{j\to i}^{(l)} = 2\text{atanh}\left(\tanh\left(\frac{m_{0,j}}{2}\right)\prod_{c'_{i'}\in\mathcal{N}(c_j)\setminus\{c'_i\}}\tanh\left(\frac{m_{i'\to j}^{(l-1)}}{2}\right)\right). \tag{2}$$

The message passed from input symbol c'_i to output symbol c_j is

$$m_{i\to j}^{(l)} = \sum_{c_{j'}\in\mathcal{N}(c'_i)\setminus\{c_j\}} m_{j'\to i}^{(l)}. \tag{3}$$

After a pre-determined criterion is satisfied (such as convergence or that a maximum number of iterations are reached), the summary message at each input symbol c'_i can be computed as

$$m_i = \sum_{c_j\in\mathcal{N}(c'_i)} m_{j\to i}^{(l)}. \tag{4}$$

We note that these update equations ((2)-(4)) are simply the log-domain implementation of the standard BP or sum-product algorithm. For a complete treatment on how these equations are derived, the reader is referred to [13].

III. Existing Results

Following [1], for a binary-input memoryless symmetric channel (BIMSC) $\mathcal{C}$, we denote by $\text{Cap}(\mathcal{C})$ the capacity of channel $\mathcal{C}$ and by $\text{B}(\mathcal{C})$ the expectation $\text{E}(\tanh(Z/2))$, where random variable Z is the channel LLR $m_{0,j}$ defined in (1). We note that when $m_{0,j}$, or, Z, is treated as a random variable, it is independent of j, since the channel statistics we consider are time-invariant. An important parameter $\Pi(\mathcal{C})$ is then defined as

$$\Pi(\mathcal{C}) := \frac{\text{Cap}(\mathcal{C})}{\text{B}(\mathcal{C})}. \tag{5}$$

A sequence of Raptor codes $(k, \mathtt{C}, \Omega^{(k)}(x))$ indexed by increasing k is said to be capacity-achieving over a given BIMSC $\mathcal{C}$, if BP decoding applied to $k/\text{Cap}(\mathcal{C}) + o(k)$ output symbols gives rise to decoding error probability approaching zero as k approaches infinity. In [1], it is shown that a necessary condition for a Raptor-code sequence to be capacity-achieving over a BIMSC $\mathcal{C}$ is that Ω_1 and Ω_2 are respectively lower-bounded and converge as follows.

$$\Omega_1^{(k)} > 0 \text{ and } \lim_{k\to\infty}\Omega_1^{(k)} = 0, \tag{6}$$

and

$$\Omega_2^{(k)} > \frac{\Pi(\mathcal{C})}{2} \text{ and } \lim_{k\to\infty}\Omega_2^{(k)} = \frac{\Pi(\mathcal{C})}{2}, \tag{7}$$

provided that $\mathrm{B}(\mathcal{C}) \neq 0$ and $\mathrm{Cap}(\mathcal{C}) \neq 0$. Following [1], the lower bound and limit $\Pi(\mathcal{C})/2$ of Ω_2 in (7) is denoted by $\Omega_2(\mathcal{C})$. Since except for binary erasure channels, $\Omega_2(\mathcal{C})$ depends on the channel parameters for most BIMSCs — including binary-input AWGN (BIAWGN) channels, an immediate consequence of this result is that there exists no single output-symbol degree distribution $\Omega(x)$ of Raptor codes that is universally capacity-achieving for any of these channel families.

Upon this development, the authors of [1] then present a framework of constructing Raptor codes for a *given* BIAWGN channel, which we refer to as the *mean-LLR-EXIT chart* (or simply EXIT chart) approach. Similar to density evolution (DE) [11] and to the one-dimensional approximation of DE [12], there are two key assumptions involved in this method of construction.

1) Cycle-free assumption The factor-graph representation of the Raptor code is *locally cycle-free*, so that all incoming BP messages arriving at a given node in the graph can be treated as being statistically independent.
2) Semi-Gaussian assumption The probability density of a message passed from an input symbol to an output symbol along a randomly chosen edge in the graph is a mixture of *symmetric* Gaussian distributions. We note that a probability density function (pdf) $f(x)$ is said to be symmetric if $f(x) = e^x f(-x)$ [11], and that a uni-variate symmetric Gaussian distribution is parameterized by its mean only, as under the symmetric condition, the variance of the Gaussian is twice its mean [12].

The first assumption is well-justified when the graph is large and sparsely connected. For the second assumption, since the BP message sent from an input symbol is the sum of the incoming messages arriving at the symbol, and when each input symbol has relatively high degree, the message sent from the symbol is approximately Gaussian, under the Central Limit Theorem. The Gaussian Mixture model in the second assumption is then a consequence of the irregularity of the input-symbol degrees, namely that input symbols have different degree.

Now we denote by $\mu^{(l)}$ the mean of message $m_{i\to j}^{(l)}$ along a randomly chosen edge in BP iteration l, and by α the average degree of the input symbols in the LT component code. Under the above assumptions and that the all-zero codeword is transmitted, it is possible to show for BIAWGN channels that $\mu^{(l+1)}$ and $\mu^{(l)}$ are related by

$$\mu^{(l+1)} = \alpha \sum_{d} \omega_d f_d(\mu^{(l)}), \tag{8}$$

with function f_d defined as follows.

$$f_d(\mu) := 2\mathrm{E}\left(\mathrm{atanh}\left(\tanh\left(\frac{Z}{2}\right)\prod_{q=1}^{d-1}\tanh\left(\frac{X_q}{2}\right)\right)\right), \tag{9}$$

where X_q $(q = 1, \ldots, d-1)$ is the symmetric Gaussian random variable with mean μ describing the message $m_{i\to j}^{(l)}$, and where $X_1, \cdots, X_{d-1}, Z$ are mutually independent. We note that it is easy to verify that Z has symmetric Gaussian pdf with mean $2\mathrm{SNR}$, where SNR is the signal-to-noise ratio of the BIAWGN channel[1]. Here we note that throughout this paper, SNR is in linear scale unless otherwise specified in unit of dB.

For each d, we refer to $f_d(\mu)$ as an "elementary" EXIT chart, which can be interpreted as the expected value of a message passed from a degree-d output symbol.

It is then straight-forward to show that a given value of α induces a design rate (of the LT code) $R_{\text{design}} = 1/(\alpha\sum\omega_d/d)$, and that $\mu^{(l+1)} > \mu^{(l)}$ implies that the bit error probability (BER) of the input symbols (upon a hard decision on the messages) at iteration $l+1$ is lower than that in iteration l.

Thus the authors of [1] formulate the problem of designing a Raptor code (or the LT component code) as finding $\Omega(x)$ — or equivalently $\omega(x)$ — that maximizes the design rate R_{design} subject to the constraint that BER must decrease as BP iterates. Specifically, in the design framework of [1], one first fixes a choice of average input-symbol degree α, maximal output-symbol degree D, and targeted maximal message mean μ_0, and then solve the linear program

$$\begin{aligned} \text{minimizing} \quad & \alpha\sum_{d=1}^{D}\omega_d/d \\ \text{subject to} \quad & \forall i = 1, \ldots, N: \ \alpha\sum_{d=1}^{D}\omega_d f_d(\mu_i) > \mu_i \\ & \sum_{d=1}^{D}\omega_d = 1 \\ & \forall d = 1, \cdots, D: \ \omega_d \geq 0, \end{aligned} \tag{10}$$

[1]Without loss of generality, the BIAWGN channel is modelled as $Y = X + N$, where X is the input to the channel, taking values from $\{1, -1\}$ under a one-to-one correspondence with codeword symbol alphabet $\{0, 1\}$, Y is the output from the channel, and N is the real Gaussian noise with variance σ^2 independently drawn for each channel use. Thus the channel SNR is defined as $1/\sigma^2$. Mapping back to AWGN channel models characterized in terms of noise power spectral density equal to $N_0/2$, this definition of SNR is equivalent to $2E_s/N_0$, where E_s is the input symbol energy.

where $\{\mu_i : i = 1, \ldots, N\}$ is a set of uniformly spaced values in range $(0, \mu_0]$.

IV. Bounds of Channel SNR, $\mathrm{SNR}^*_{\mathrm{low}}$ and $\mathrm{SNR}^*_{\mathrm{high}}$

The linear-programming framework based on the EXIT chart provides a useful tool for the design of Raptor codes for BIAWGN channels, and the authors of [1] presented good code examples constructed using this approach. However, the use of this approach requires the code designer to supply to the linear program a few parameters, namely, α, μ_0, and D for any given channel SNR, and to date there has been no serious effort in deriving optimal choice of these parameters. Heuristically, value D can be chosen as a relatively large number to include a sufficient space of $\Omega(x)$. It is also understood that the value of μ_0 should be chosen depending on the targeted errors to be erased by the LDPC precode, although such a dependency yet requires a careful characterization. The choice of α to date also remains mostly heuristic [14] (see also Appendix IV), with the reason being that the best choice of α depends on solving for the best $\omega(x)$ in the linear program.

This paper is motivated by studying the effectiveness of the linear program for various channel SNRs. Specifically we investigate, what is the consequence, in terms of the solution of the linear program, of a given choice of parameter setting across the range of all channels. In particular, we are interested in whether a given parameter setting for the linear program is capable of producing capacity-achieving codes for every channel.

Formally, we say that a given choice of $(\alpha, \mu_0, D, \mathrm{SNR})$ is *feasible* if the linear program (10) has a solution, namely, if the constraints in the linear program define a non-empty set of $\omega(x)$. For any choice of (α, μ_0, D), we define the following threshold values of SNR.

$$\mathrm{SNR}^*_{\mathrm{low}}(\alpha, \mu_0) := \mu_0 / 2\alpha.$$

$$\mathrm{SNR}^*_{\mathrm{high}}(d; \alpha, \mu_0) := \inf \Big\{ \mathrm{SNR} : (\alpha, \mu_0, D, \mathrm{SNR}) \text{ feasible and } \forall \mu \in (0, \mu_0], \frac{f_2(\mu) - f_d(\mu)}{2\mathrm{SNR} - f_d(\mu)} < \frac{d-2}{2d-2} \Big\}, 2 < d \leq D$$

and

$$\mathrm{SNR}^*_{\mathrm{high}}(\alpha, \mu_0, D) := \min \left\{ \mathrm{SNR}^*_{\mathrm{high}}(d; \alpha, \mu_0) : 2 < d \leq D \right\}.$$

Our analysis relies on the following lemmas, the proof of which are provided in Appendices I, II, and III.

Lemma 1: Let X be a symmetric Gaussian random variable with mean μ, and $h(x)$ be an increasing function of x with the property that $h(-x) = -h(x)$, then the expectation of the function $\mathrm{E}\left(h(X)\right)$ increases with μ.

Lemma 2: Suppose the multivariate function $y = g(x_1, x_2, \ldots, x_n)$ has the following properties,

1) $\forall i \in (1, \ldots, n)$, when $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n$ are fixed,

$$g(x_1, \ldots, -x_i, \ldots, x_n) = -g(x_1, \ldots, x_i, \ldots, x_n);$$

2) If $x_1 \geq 0, x_2 \geq 0, \ldots, x_n \geq 0$, then $y \geq 0$;
3) If $x_1 \geq 0, x_2 \geq 0, \ldots, x_n \geq 0$, then $\forall i \in (1, \ldots, n)$, when $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n$ are fixed, the function y strictly increases with x_i.

Let Y be a random variable defined as $Y = g(X_1, X_2, \ldots, X_n)$, where $X_1, X_2, \ldots, X_n$ are all independent symmetric Gaussian random variables with mean $\mu_1, \mu_2, \ldots, \mu_n$ respectively, then the expectation $\mathrm{E}(Y)$ increases with μ_i provided that $\mu_1, \ldots, \mu_{i-1}, \mu_{i+1}, \ldots, \mu_n$ are all fixed.

Lemma 3: The elementary EXIT chart $f_d(\mu)$ as defined in (9), satisfies the following properties.

1) $f_d(\mu) > 0$ for all $\mu > 0$ and $d > 0$.
2) $f_1(\mu) = 2\mathrm{SNR}$.
3) $f_2(\mu) \leq \mu$ and $f_2(\mu) \leq 2\mathrm{SNR}$.
4) For any $d > 1$ and SNR, $f_d(\mu)$ increases with μ.
5) For any μ and SNR, $f_d(\mu)$ decreases with d.
6) For any d and μ, $f_d(\mu)$ increases with SNR.

Theorem 1: Any $(\alpha, \mu_0, D, \mathrm{SNR})$ is infeasible if and only if $\mathrm{SNR} < \mathrm{SNR}^*_{\text{low}}(\alpha, \mu_0)$.

We note that an equivalent result to this theorem was first observed by Shokrollahi [14].

Proof: First recall the constraint of the linear program, namely that for any μ_i, $\mu_i < \alpha \sum_d \omega_d f_d(\mu_i)$. But by Lemma 3, $f_d(\mu)$ decreases with d, we have, for every μ_i,

$$\sum_d \omega_d f_d(\mu_i) < \sum_d \omega_d f_1(\mu_i) = f_1(\mu_i) = 2\mathrm{SNR}.$$

Thus we have $\mu_i < 2\alpha\mathrm{SNR}$. Taking μ_i to its maximum μ_0 gives that $\mathrm{SNR} > \mu_0/2\alpha = \mathrm{SNR}^*_{\text{low}}(\alpha, \mu_0)$, which is a necessary condition for $(\alpha, \mu_0, D, \mathrm{SNR})$ to be feasible. Thus, $\mathrm{SNR} < \mathrm{SNR}^*_{\text{low}}(\alpha, \mu_0)$ is sufficient to ensure $(\alpha, \mu_0, D, \mathrm{SNR})$ infeasible.

Now suppose $\mathrm{SNR} \geq \mathrm{SNR}^*_{\mathrm{low}}(\alpha, \mu_0)$, then we are to show $(\alpha, \mu_0, D, \mathrm{SNR})$ is feasible. Since $\mathrm{SNR}^*_{\mathrm{low}}(\alpha, \mu_0) = \mu_0/2\alpha$, we have

$$2\alpha\mathrm{SNR} \geq \mu_0.$$

It is easy to verify that degree distribution $\omega(x) = 1$ satisfies all the constraints of the linear program, so there exists at least one feasible solution. Therefore, $(\alpha, \mu_0, D, \mathrm{SNR})$ is feasible. ■

Theorem 2: If $\mathrm{SNR} > \mathrm{SNR}^*_{\mathrm{high}}(\alpha, \mu_0, D)$, then the solution of the linear program is such that $\omega_2 = 0$.

Prior to proving the theorem, we note that given (α, μ_0, D), the theorem holds irrespective of the choice of the LDPC precode.

Proof: We will first prove that if $\mathrm{SNR} > \mathrm{SNR}^*_{\mathrm{high}}(d; \alpha, \mu_0)$ for some $d > 2$, then the solution of the linear program has $\omega_2 = 0$, which implies, if $\mathrm{SNR} > \mathrm{SNR}^*_{\mathrm{high}}(\alpha, \mu_0, D)$, the linear program has solution $\omega_2 = 0$. To prove this by contradiction, suppose that the solution of the linear program is $\omega(x)$ with $\omega_2 > 0$. We construct another $\tilde{\omega}(x)$ differing from $\omega(x)$ only in degrees $1, 2$ and d by splitting the mass ω_2 to degree 1 and degree d. That is, let $\tilde{\omega}_i = \omega_i$ for all $i \neq 1, 2, d,\ i \leq D$. Let $\tilde{\omega}_1 = \omega_1 + \lambda\omega_2,\ \tilde{\omega}_2 = 0,\ \tilde{\omega}_d = \omega_d + (1-\lambda)\omega_2$ for some $\lambda \in (0, 1)$ to be determined. Under the condition $\mathrm{SNR} > \mathrm{SNR}^*_{\mathrm{high}}(d; \alpha, \mu_0)$, clearly

$$\frac{d-2}{2d-2} > \frac{f_2(\mu) - f_d(\mu)}{2\mathrm{SNR} - f_d(\mu)}.$$

Therefore there exists some λ such that

$$\frac{d-2}{2d-2} > \lambda > \frac{f_2(\mu) - f_d(\mu)}{2\mathrm{SNR} - f_d(\mu)}, \text{for all } \mu \in (0, \mu_0].$$

Then by Lemma 3, we have

$$\frac{\omega_1}{1} + \frac{\omega_2}{2} + \frac{\omega_d}{d} > \frac{\omega_1 + \lambda\omega_2}{1} + \frac{\omega_d + (1-\lambda)\omega_2}{d}$$

and

$$\omega_1 f_1(\mu) + \omega_2 f_2(\mu) + \omega_d f_d(\mu) < (\omega_1 + \lambda\omega_2) f_1(\mu) + (\omega_d + (1-\lambda)\omega_2) f_d(\mu).$$

Equivalently,

$$\sum_d \frac{\omega_d}{d} > \sum_d \frac{\tilde{\omega}_d}{d}$$

and

$$\sum_d \omega_d f_d(\mu) < \sum_d \tilde{\omega}_d f_d(\mu)$$

for all $\mu \in (0, \mu_0]$. That is, there exists λ such that the constructed $\tilde{\omega}(x)$ satisfies all constraints of the linear program and results in a lower value of the objective function. Thus $\omega(x)$ can not be the solution of the linear program. ■

The essence of this theorem is simply suggesting that for any given (α, μ_0, D), there is an upper bound $\mathrm{SNR}^*_{\mathrm{high}}(\alpha, \mu_0, D)$ of SNR above which linear program (10) fails to construct capacity-achieving Raptor codes — noting the necessary condition of Ω_2 for capacity-achieving Raptor codes developed in [1] and re-stated in Section III. It remains to verify that $\mathrm{SNR}^*_{\mathrm{high}}(\alpha, \mu_0, D)$ has some practical significance and to assure the boundedness of $\mathrm{SNR}^*_{\mathrm{high}}(\alpha, \mu_0, D)$. To that end, define

$$\gamma(d) = \sup_{\mu \in (0, \mu_0]} \frac{f_2(\mu)(2d-2) - d f_d(\mu)}{2(d-2)}. \tag{11}$$

We note that $\gamma(d)$ is in fact a function of SNR. Let

$$I(d) := \{\mathrm{SNR} : \mathrm{SNR} > \gamma(d)\}$$

and

$$I := \bigcup_{2<d\leq D} I(d).$$

This allows us to simplify the formulation of $\mathrm{SNR}^*_{\mathrm{high}}(\alpha, \mu_0, D)$, as in the following lemma.

Lemma 4: $\mathrm{SNR}^*_{\mathrm{high}}(\alpha, \mu_0, D) = \inf\left(I \cap (\mu_0/2\alpha, +\infty)\right)$.

Proof: By definition,

$$\begin{aligned}
\mathrm{SNR}^*_{\mathrm{high}}(d; \alpha, \mu_0) &= \inf\left\{\mathrm{SNR} : \mathrm{SNR} > \mu_0/2\alpha,\ \text{and } \forall \mu \in (0, \mu_0], \frac{f_2(\mu) - f_d(\mu)}{2\mathrm{SNR} - f_d(\mu)} < \frac{d-2}{2d-2}\right\} \\
&= \inf\left\{\mathrm{SNR} : \mathrm{SNR} > \mu_0/2\alpha,\ \text{and } \mathrm{SNR} > \gamma(d)\right\}.
\end{aligned}$$

Therefore,

$$\begin{aligned}
\mathrm{SNR}^*_{\mathrm{high}}(\alpha, \mu_0, D) &= \min_{2<d\leq D} \mathrm{SNR}^*_{\mathrm{high}}(d; \alpha, \mu_0) \\
&= \min_{2<d\leq D} \left(\inf\left\{\mathrm{SNR} : \mathrm{SNR} > \mu_0/2\alpha,\ \text{and } \mathrm{SNR} > \gamma(d)\right\}\right) \\
&= \min_{2<d\leq D} \left(\inf\left((\mu_0/2\alpha, +\infty) \cap I(d)\right)\right) \\
&= \inf\left(\bigcup_{2<d\leq D} \left((\mu_0/2\alpha, +\infty) \cap I(d)\right)\right) \\
&= \inf\left(I \cap (\mu_0/2\alpha, +\infty)\right).
\end{aligned}$$

■

This lemma provides a simpler formulation of the bound $\mathrm{SNR}^*_{\mathrm{high}}(\alpha, \mu_0, D)$. Following this, it is easy to see that $\mathrm{SNR}^*_{\mathrm{high}}(\alpha, \mu_0, D)$ is bounded due to

$$\gamma(d) < \frac{f_2(\mu)(2d-2)}{2(d-2)} \leq \frac{\mu(2d-2)}{2(d-2)}.$$

Clearly, $\mathrm{SNR}^*_{\mathrm{high}}(\alpha, \mu_0, D)$ is non-increasing with α, since I is independent of α. Furthermore, for any given μ_0, if at some $\alpha = \hat{\alpha}$, $\mathrm{SNR}^*_{\mathrm{high}}(\hat{\alpha}, \mu_0, D) = \inf(I)$, then $\mathrm{SNR}^*_{\mathrm{high}}(\alpha, \mu_0, D) = \inf(I)$ for any $\alpha > \hat{\alpha}$. This corresponds to cases of practical significance, which we outline next.

For $\alpha = 1$, $\mathrm{SNR}^*_{\mathrm{low}}(\alpha, \mu_0)$ and $\mathrm{SNR}^*_{\mathrm{high}}(\alpha, \mu_0, D)$ are plotted as functions of μ_0 in Fig. 2, where $\mathrm{SNR}^*_{\mathrm{high}}(\alpha, \mu_0, D)$ is computed numerically. (We note that in our computation of $\mathrm{SNR}^*_{\mathrm{high}}(\alpha, \mu_0, D)$, we observe that $\mathrm{SNR}^*_{\mathrm{high}}(\alpha, \mu_0, D)$ is independent of D and in addition $\mathrm{SNR}^*_{\mathrm{high}}(\alpha, \mu_0, D) \approx \mathrm{SNR}^*_{\mathrm{high}}(d = 3; \alpha, \mu_0)$). The following remarks are in order.

1) We note that $\alpha = 1$ is the minimal value that α is allowed to take since every input symbol must contribute to generating at least one of the output symbols. As a consequence, the $\mathrm{SNR}^*_{\mathrm{low}}(\alpha, \mu_0)$ curve plotted for $\alpha = 1$ will be uniformly higher than the $\mathrm{SNR}^*_{\mathrm{low}}(\alpha, \mu_0)$ curves plotted for any other value of α.
2) From Fig. 2, it is clear that at every μ_0 in the plotted range $\mathrm{SNR}^*_{\mathrm{high}}(\alpha = 1, \mu_0, D) > \mathrm{SNR}^*_{\mathrm{low}}(\alpha = 1, \mu_0)$. By Lemma 4,

$$\begin{aligned} \mathrm{SNR}^*_{\mathrm{high}}(\alpha = 1, \mu_0, D) &= \inf\left(I \cap \left(\mathrm{SNR}^*_{\mathrm{low}}\left(\alpha = 1, \mu_0\right), +\infty\right)\right) \\ &= \inf(I). \end{aligned}$$

   Using the argument following the lemma, for every $\alpha > 1$, $\mathrm{SNR}^*_{\mathrm{high}}(\alpha, \mu_0, D) = \inf(I)$ at every μ_0 in the plotted range. That is, the plotted curve $\mathrm{SNR}^*_{\mathrm{high}}(\alpha = 1, \mu_0, D)$ is in fact $\mathrm{SNR}^*_{\mathrm{high}}(\alpha, \mu_0, D)$ for all $\alpha > 1$.
3) The figure then suggests that for a reasonable choice of μ_0, say between 20 and 30, the EXIT-chart approach will fail to produce capacity-achieving codes for channel SNR higher than 12-13 dB, based on $\mathrm{SNR}^*_{\mathrm{high}}(\alpha, \mu_0, D)$.

Some insights may be obtained from these results.

First, the ability of the EXIT chart based approaches in constructing capacity-achieving Raptor codes for BIAWGN channels is fundamentally limited. This limitation is enhanced when one recognizes that the bound $\mathrm{SNR}^*_{\mathrm{high}}(\alpha, \mu_0, D)$ only provides a *sufficient* condition for failing to achieve capacity. Specifically, the bound corresponds to a condition that *drastically* violates the capacity-achieving requirement of Ω_2 — by setting it to zero. We expect that considerably earlier before channel SNR increases to pass the bound $\mathrm{SNR}^*_{\mathrm{high}}(\alpha, \mu_0, D)$, the solution of the linear program has already departed from the capacity-achieving requirement of Ω_2, even without considering the possibility of violating the requirement of Ω_1. However, we would also like to note that such a limitation may not have significant practical relevance, since for high-SNR channels, one would rarely consider binary signaling after all.

Second, when constructing Raptor codes for high SNR BIAWGN channels, it is necessary to design the LDPC precode and the LT code jointly. Previous results appear to have overlooked this aspect by choosing an LDPC code mostly arbitrarily as long as the rate loss is not significant. Theorem 2 suggests that such a casual consideration is inadequate for high-SNR channels. Specifically, the rate and structure of the LDPC precode approximately determine the residual error left by the LT code for which the LDPC code is responsible. This residual error induces a choice of μ_0 in the design of LT codes using the EXIT chart. The choice of μ_0 — approximately increasing with rate of LDPC code (assuming LDPC code is optimally designed) — in turn sets an upper bound of SNR — also increasing with μ_0 — below which capacity-achieving codes can be designed. Thus for a given relatively-high channel SNR, it is necessary to carefully blueprint the rate and structure of the LDPC code together with the design of the LT code, possibly using μ_0 as their interface.

Finally, combining Theorem 1, it is necessary to choose the parameters (α, μ_0, D) carefully in designing Raptor codes for a given channel, provided that a fixed choice of LDPC code has been specified. This necessity deserves an attention particularly because it is not yet clear up to what SNR level a given choice of (α, μ_0, D) fails to produce capacity-achieving codes.

At this end, we have fully characterized the behavior of the mean-LLR-EXIT chart based linear program for $\mathrm{SNR} < \mathrm{SNR}^*_{\mathrm{low}}(\alpha, \mu_0)$ and $\mathrm{SNR} > \mathrm{SNR}^*_{\mathrm{high}}(\alpha, \mu_0, D)$. It remains for investigation how the linear program behaves for SNR between the two thresholds. In next section, we present some preliminary results along this direction via a simulation study.

V. The case of $\mathrm{SNR}^*_{\mathrm{low}} < \mathrm{SNR} < \mathrm{SNR}^*_{\mathrm{high}}$

To partially eliminate the effect due to the sub-optimality of the choice of α supplied to the linear program, we make a modest modification as to how α is determined. In our modified approach, we consider the optimization problem (10) also including α as a variable, and perform

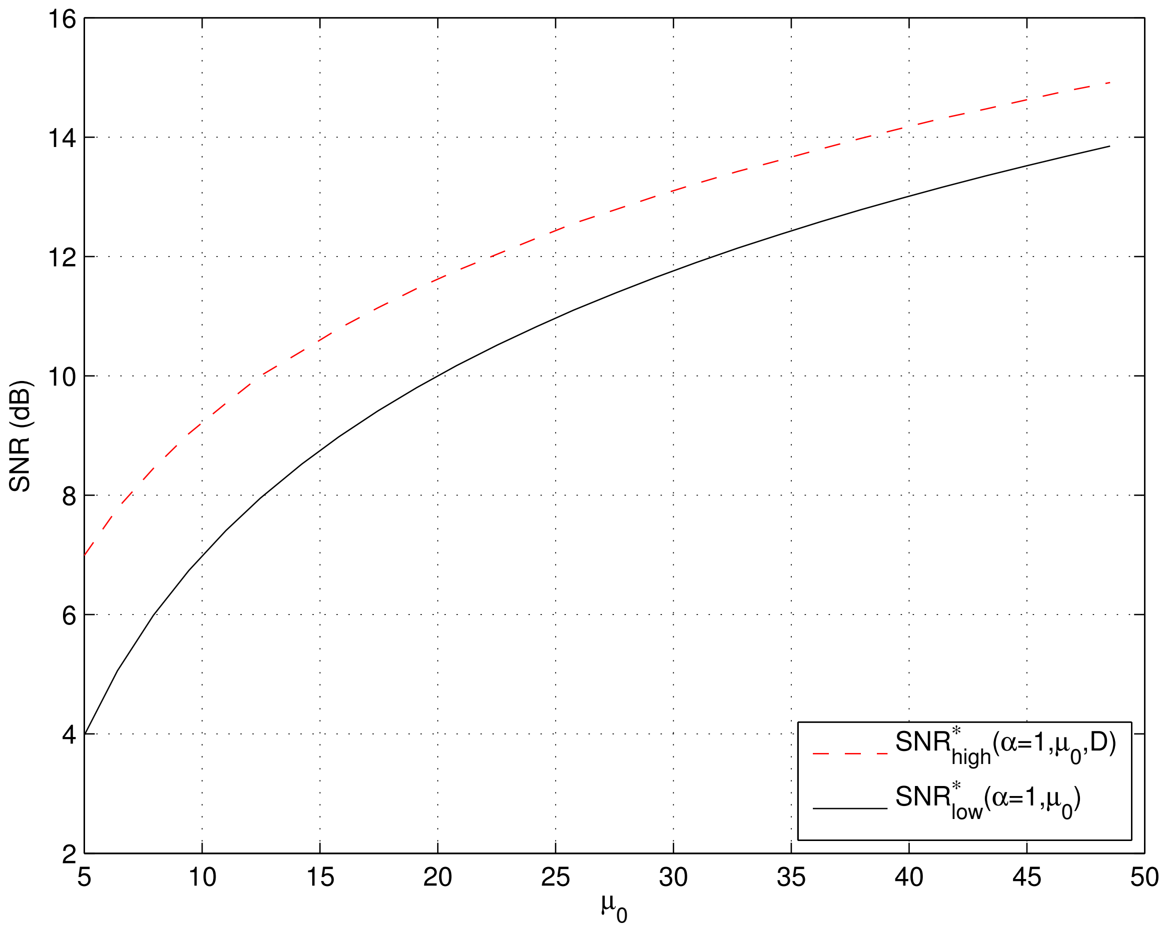

Fig. 2. Bounds $\mathrm{SNR}^*_{\mathrm{high}}(\alpha, \mu_0, D)$ and $\mathrm{SNR}^*_{\mathrm{low}}(\alpha, \mu_0)$ as functions of μ_0 for $\alpha = 1$.

optimization jointly over α and $\omega(x)$. The comparison between the modified approach and the original approach is shown in Appendix IV, and one can see that the modified approach results in better codes. We then restrict our simulations to using only the modified approach.

Code design and simulations were carried out for different choices of parameters and LDPC precodes. For each value of channel SNR, μ_0 is chosen to be 30 and 60 respectively, and D is chosen to be 200. Additionally, in simulations, for each code designed with different μ_0, we selected precode to be a rate-0.95 left-4-regular right-Poisson code as presented in [8], and a randomly constructed rate-0.7 left-4-regular right-Poisson code respectively. The number of input symbols k' is set to be 10000 and the number of information bits is set to be 9500 and 7000

corresponding to different LDPC precodes. We also verified that the channel SNR at which the code is designed is within the range of $[\mathrm{SNR}^*_{\mathrm{low}}, \mathrm{SNR}^*_{\mathrm{high}}]$.

Each constructed Raptor code is simulated, under BPSK modulation, over the AWGN channel for which the code is designed. The performances of these codes are evaluated in terms of their *realized rates*. We note here that the realized rate of a Raptor code over a given channel is defined as $k/\mathrm{E}[n]$ over all transmitted codewords, where n is the earliest time at which the codeword can be correctly decoded.

Fig. 3 plots respectively the realized rates achieved by Raptor codes designed with different μ_0 and LDPC precode (Note that each point in the plots corresponds to a code designed specifically for that SNR). Also plotted in the figure is the capacity of the BIAWGN channels. First one may identify from the figure that codes designed with μ_0 to be 30 perform uniformly better than those with μ_0 at 60. This may be reasoned by noting that larger μ_0 corresponds to more inequality constraints on the feasible configurations of the linear program and hence a reduced space of feasible configurations. This leads to a higher optimal value of the linear program, which corresponds to a lower designed rate. In addition, for each chosen μ_0, codes with the rate-0.7 precode perform rather poorly comparing with the capacity, particularly at the high-SNR end. This is due to the severe rate loss in the low-rate precoding. We then from here on restrict our discussion to codes with rate-0.95 precode.

At the lower-end of the simulated channel SNR, the constructed codes with rate-0.95 precode perform fairly closely to the capacity. As SNR increases, the realized rates of these codes gradually depart from the capacity curve, and the gap to capacity becomes more visible. We note that this behaviour is also observed for codes with precode having higher rates (for example we also simulated codes with rate-0.98 LDPC precode, and nearly identical results are seen – data not shown). The behaviour of these codes diverging from the capacity curve may be explained using their corresponding values of Ω_2, which are plotted in Fig. 4. In the figure, the lower-bound $\Omega_2(\mathcal{C})$ for capacity-achieving Raptor codes is also plotted. Clearly, at the lower end of simulated range of SNR, the generated values of Ω_2 stay above or close to the lower-bound $\Omega_2(\mathcal{C})$. As channel SNR increases to exceed certain value, the generated Ω_2 quickly drops below the lower-bound, violating the capacity-achieving conditions. It is worth noting that for channel with 5dB SNR, the resulting Ω_2 is already 0. This confirms our earlier statement that the violation of capacity-achieving condition may happen much earlier before SNR reaches the $\mathrm{SNR}^*_{\mathrm{high}}$ bound,

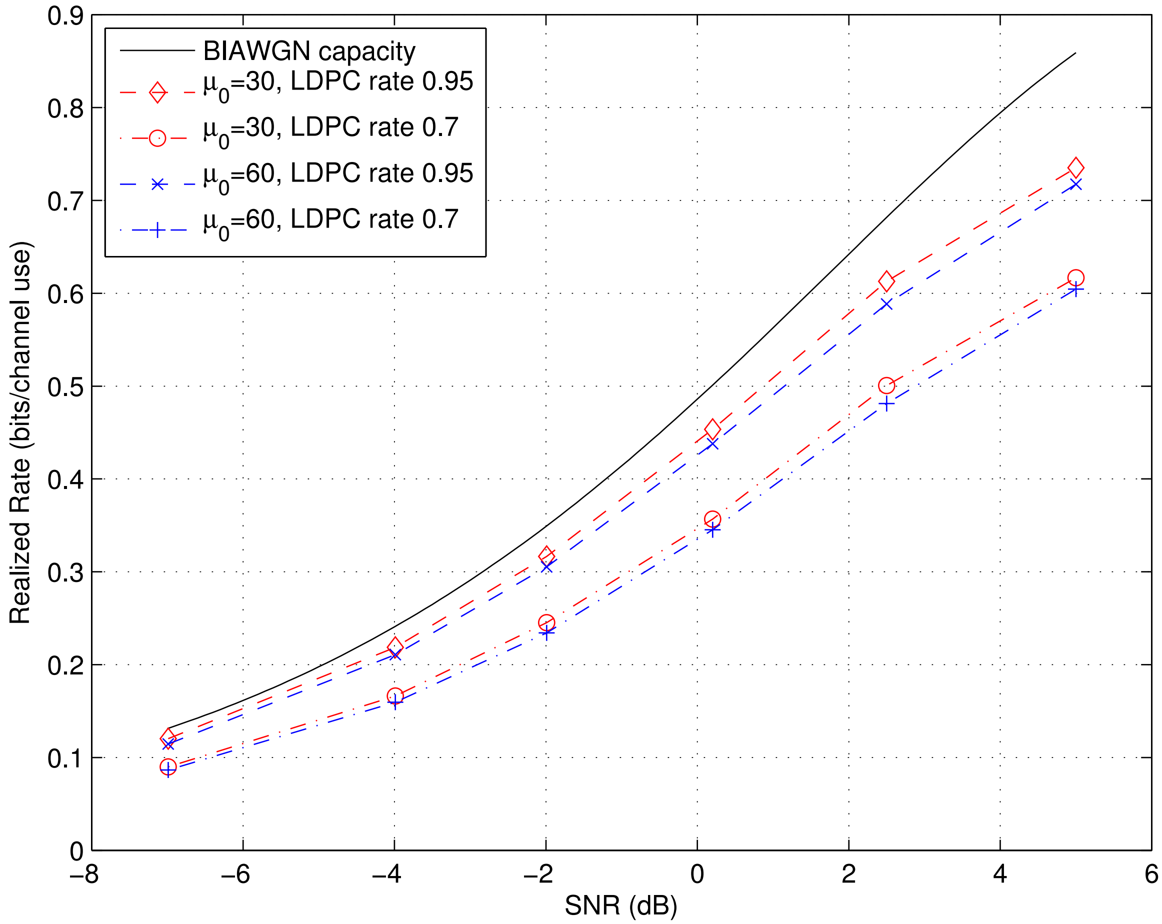

Fig. 3. Realized rates of Raptor codes constructed using different parameters.

which is about 13 dB.

One may wish to explore the reasons for which the EXIT chart based approach fails to produce the desired Ω_2 at these SNR values. Undoubtedly, what is responsible may include imperfect choice of the rate and structure of the LDPC precode as well as the sub-optimal choice of μ_0. There is another factor which we believe also contributes significantly to the failure of the EXIT chart based approaches. — Recall in Section III, we noted that the validity of mean-LLR-EXIT chart approach relies on the semi-Gaussian assumption, which only holds approximately true when the input symbols mostly have relatively high degrees. In the higher-SNR regime, it is necessary that on average the input symbols have low degrees, or equivalently that the capacity-

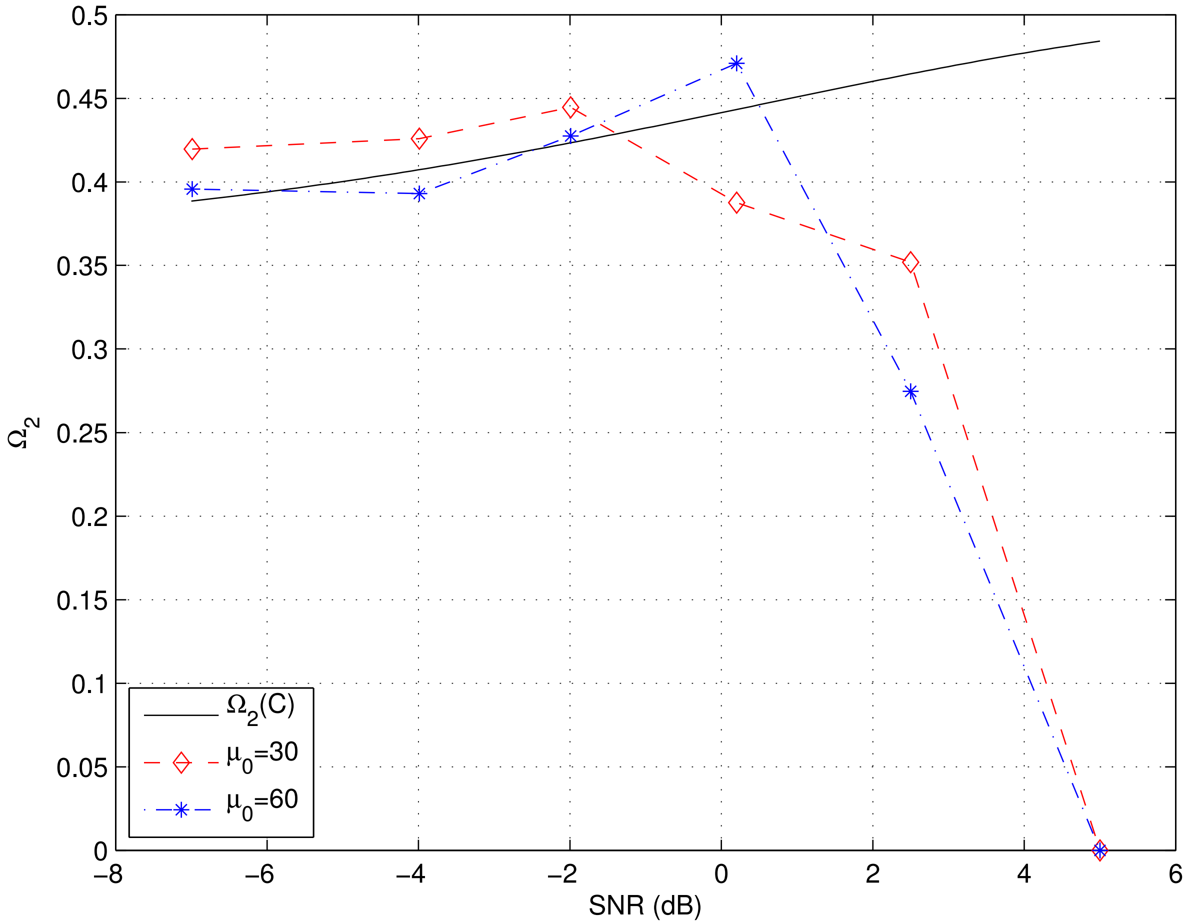

Fig. 4. Determined Ω_2 with different μ_0 at various values of channel SNR.

achieving α is relatively small. This is because the capacity-achieving Raptor codes are decodable at a shorter code length, resulting in fewer edges in the factor graph and hence smaller α for the same number of input symbols. This fact makes the semi-Gaussian assumption invalid which in turn limits the accuracy of the EXIT-chart formulation. For example, in the codes constructed for $\mu_0 = 30$, the determined α is 5.2 for $\mathrm{SNR} = 5$dB, and is 24.4 for $\mathrm{SNR} = -2$dB. This should, to a good extent, indicate the validity of the semi-Gaussian assumption for the EXIT-chart approaches and explain why the capacity-achieving condition of Ω_2 is violated in the first case, but satisfied in the second case. Additionally we notice that when α is determined to be rather small, the resulting designed rate R_{design} (after factoring out the rate loss due to the LDPC precode) may exceed the capacity. This observation from another perspective suggests that in these cases, the EXIT-chart formulation of the code design problem is no longer valid.

VI. Conclusion

This paper reports a study of the EXIT-chart-based linear-programming approach to the construction of Raptor codes for binary-input AWGN channels [1]. Giving suggestions on Raptor code design, we establish a result that there are two SNR bounds, $\mathrm{SNR}^*_{\mathrm{low}}$ and $\mathrm{SNR}^*_{\mathrm{high}}$, for any given parameter setting of the linear program. We show that if the channel SNR is outside the interval $[\mathrm{SNR}^*_{\mathrm{low}}, \mathrm{SNR}_{\mathrm{high}}]$, the linear program either fails to produce solutions or produces Raptor codes failing to achieve the capacity. Via simulations, we also show that when the channel SNR is at the lower end of this interval, the constructed codes perform closely to the capacity, whereas as SNR increases to close to the higher end, the constructed codes suffer from performance degradation.

Raptor codes are appealing communication schemes under channel uncertainty. We hope that this work inspire more research in the design of Raptor codes, or more generally, rateless codes, for high SNR channels.

Acknowledgment

The authors are indebted to Dr. Amin Shokrollahi for his valuable insights.

Appendix I
Proof of Lemma 1

We express the expectation of the function $h(X)$ as

$$\mathrm{E}\left(h(X)\right) = \int_{-\infty}^{+\infty} h(x) f_X(x) dx,$$

where $f_X(x)$ is the pdf of the random variable X.

By the symmetric condition of X, $f_X(-x) = e^{-x} f_X(x)$ holds. Thus,

$$\begin{aligned}
\mathrm{E}(h(X)) &= \int_{-\infty}^{0} h(x) f_X(x) dx + \int_{0}^{+\infty} h(x) f_X(x) dx \\
&= -\int_{0}^{+\infty} h(x) f_X(-x) dx + \int_{0}^{+\infty} h(x) f_X(x) dx \\
&= \int_{0}^{+\infty} h(x)(f_X(x) - e^{-x} f_X(x)) dx \\
&= \int_{0}^{+\infty} h(x)(1 - e^{-x}) f_X(x) dx.
\end{aligned}$$

Since $h(x)$ is an increasing function of x, and the term $1 - e^{-x}$ also increases with x, we construct a new function for convenience

$$H(x) = \begin{cases} h(x)(1 - e^{-x}) & \text{if } x > 0; \\ 0 & \text{if } x \leq 0. \end{cases}$$

Obviously $H(x)$ strictly increases when $x > 0$.

By such definition of $H(x)$, $\mathrm{E}(h(X))$ can be rewritten as

$$\mathrm{E}(h(X)) = \mathrm{E}(H(X)).$$

For an arbitrary positive δ, define

$$X' = (X - \mu)\sqrt{(\mu + \delta)/\mu} + \mu + \delta.$$

Then X' and X are jointly Gaussian and in particular one can verify that X' is symmetric Gaussian with mean $\mu + \delta$. Since $X' > X$ with probability 1 and because $H(x)$ is increasing with x, $\mathrm{E}(H(X')) > \mathrm{E}(H(X))$. We note that as this inequality depends only on the marginal distributions of X and X', independent of their joint distribution, we have proved that $\mathrm{E}(H(X))$ increases with mean of X, hence the lemma.

Appendix II
Proof of Lemma 2

It is sufficient to prove the lemma only for $i = 1$. First we consider the conditional expectation $\mathrm{E}(Y|X_2 = x_2, \ldots, X_n = x_n)$ when $x_2, \ldots, x_n$ all take positive values. Clearly

$$\mathrm{E}(Y|X_2 = x_2, \ldots, X_n = x_n) = \mathrm{E}(g(X_1, x_2, \ldots, x_n)).$$

For this choice of $x_2, \ldots, x_n$, denote $g^*(x_1) = g(x_1, x_2, \ldots, x_n)$. Then g^* increases with x_1, and $g^*(-x_1) = -g^*(x_1)$. By Lemma 1, $\mathrm{E}(g^*(X_1))$ increases with μ_1, i.e., $\mathrm{E}(Y|X_2 = x_2, \ldots, X_n = x_n)$ increases with μ_1, if $x_2, \ldots, x_n$ are all positive.

Now

$$\mathrm{E}(Y) = \int_{-\infty}^{+\infty} \cdots \int_{-\infty}^{+\infty} \mathrm{E}(Y|X_2 = x_2, \ldots, X_n = x_n) f_{X_2,\ldots,X_n}(x_2, \ldots, x_n) dx_2 \ldots dx_n.$$

Due to the properties of the function g and the independence among all X_i's, and by splitting the integration interval for each variable, it can be obtained that

$$\begin{aligned}\mathrm{E}(Y) &= \int_0^{+\infty} \cdots \int_0^{+\infty} \mathrm{E}(Y|X_2 = x_2, \ldots, X_n = x_n) \\ &\quad \times (f_{X_2}(x_2) - f_{X_2}(-x_2)) \cdots (f_{X_n}(x_n) - f_{X_n}(-x_n)) dx_2 \ldots dx_n.\end{aligned}$$

Note that for $x_i > 0$, the term $f_{X_i}(x_i) - f_{X_i}(-x_i)$ is always greater than 0 (since X_i is a Gaussian with positive mean).

The only term that involves μ_1 is the conditional expectation

$$\mathrm{E}(Y|X_2 = x_2, \ldots, X_n = x_n),$$

which increases with μ_1. Therefore, with all the other parameters $\mu_2, \ldots, \mu_n$ fixed, $\mathrm{E}(Y)$ also increases with μ_1.

APPENDIX III
PROOF OF LEMMA 3

Let function $y = g(z, x_1, \ldots, x_{d-1})$ be defined as

$$y := 2\mathrm{atanh}\left(\tanh\left(\frac{z}{2}\right)\prod_{q=1}^{d-1}\tanh\left(\frac{x_q}{2}\right)\right).$$

One can verify that all properties assumed for g in Lemma 2 hold. Clearly for all $\mu > 0$ and $d > 0$, $f_d(\mu) > 0$ due to $f_d(\mu)$ being defined as the expected value of a message and the channel being AWGN. When $d = 1$, it is easy to verify that $f_1(\mu) = \mathrm{E}(Z)$, and thus $f_1(\mu) = 2\mathrm{SNR}$.

For any $d > 1$ and fixed SNR, the distribution of Z is fixed. Since $\mathrm{E}(Y)$ is a function of $\mu_1, \mu_2, \ldots, \mu_{d-1}$, we denote it by $\mathrm{E}(Y; \mu_1, \mu_2, \ldots, \mu_{d-1})$. For any $\mu' < \mu''$, by Lemma 2, we have

$$\begin{aligned} \mathrm{E}(Y;\mu_1=\mu',\mu_2=\mu',\ldots,\mu_{d-1}=\mu') &< \mathrm{E}(Y;\mu_1=\mu'',\mu_2=\mu',\ldots,\mu_{d-1}=\mu') \\ &< \mathrm{E}(Y;\mu_1=\mu'',\mu_2=\mu'',\ldots,\mu_{d-1}=\mu') \\ &< \mathrm{E}(Y;\mu_1=\mu'',\mu_2=\mu'',\ldots,\mu_{d-1}=\mu''), \end{aligned}$$

i.e., $f_d(\mu)$ increases with μ.

Similarly, for any d and μ, the distribution of X_q, $\forall q \in (1,2,\ldots,d-1)$, is fixed. Since $\mathrm{E}(Y)$ is a function of μ_Z, where μ_Z is the mean of Z, we denote it by $\mathrm{E}(Y;\mu_Z)$. Clearly $\mu_Z = 2\mathrm{SNR}$. For any $\mathrm{SNR}' < \mathrm{SNR}''$, by Lemma 2, we have

$$\mathrm{E}(Y;\mu'_Z) < \mathrm{E}(Y;\mu''_Z),$$

i.e., $f_d(\mu)$ increases with SNR.

When d decreases to $d-1$, we equivalently change the random variable X_d with mean μ to satisfy $\tanh(X_d/2) = 1$ with probability 1, which means an increasing of μ to infinity and that by Lemma 2, $f_d(\mu)$ increases. This proves that $f_d(\mu)$ decreases with d.

Then we have $f_2(\mu) < f_1(\mu)$, which results in $f_2(\mu) < 2\mathrm{SNR}$. Considering Z and X_1 are symmetric in the expression

$$f_2(\mu) = 2\mathrm{E}\left(\mathrm{atanh}\left(\tanh\left(\frac{Z}{2}\right)\tanh\left(\frac{X_1}{2}\right)\right)\right),$$

since $f_2(\mu)$ is upper bounded by $\mathrm{E}(Z)$, we have $f_2(\mu)$ upper bounded by μ, the mean of X_1.

APPENDIX IV

Previously suggested by Shokrollahi [14], α is chosen approximately corresponding to that giving rise to $\mathrm{SNR}^*_{\mathrm{low}} = \mathrm{SNR}$, namely, $\alpha \approx \mu_0/2\mathrm{SNR}$. This strategy, which we refer to as the *heuristic*-α approach, although capable of constructing good Raptor codes in some cases [1], is clearly not optimal. Here we suggest embedding the determination of α in the optimization problem prescribed by (10). That is, instead of supplying a pre-selected α to the linear program of (10), the linear program is converted to an optimization problem with the same objective function and constraints as in (10) but optimizing jointly over all pairs $(\alpha, \omega(x))$. One can verify that this modified optimization problem is no longer a linear program and in fact it is not even convex. Nevertheless, one can solve the problem by searching for the optimal α over a relatively

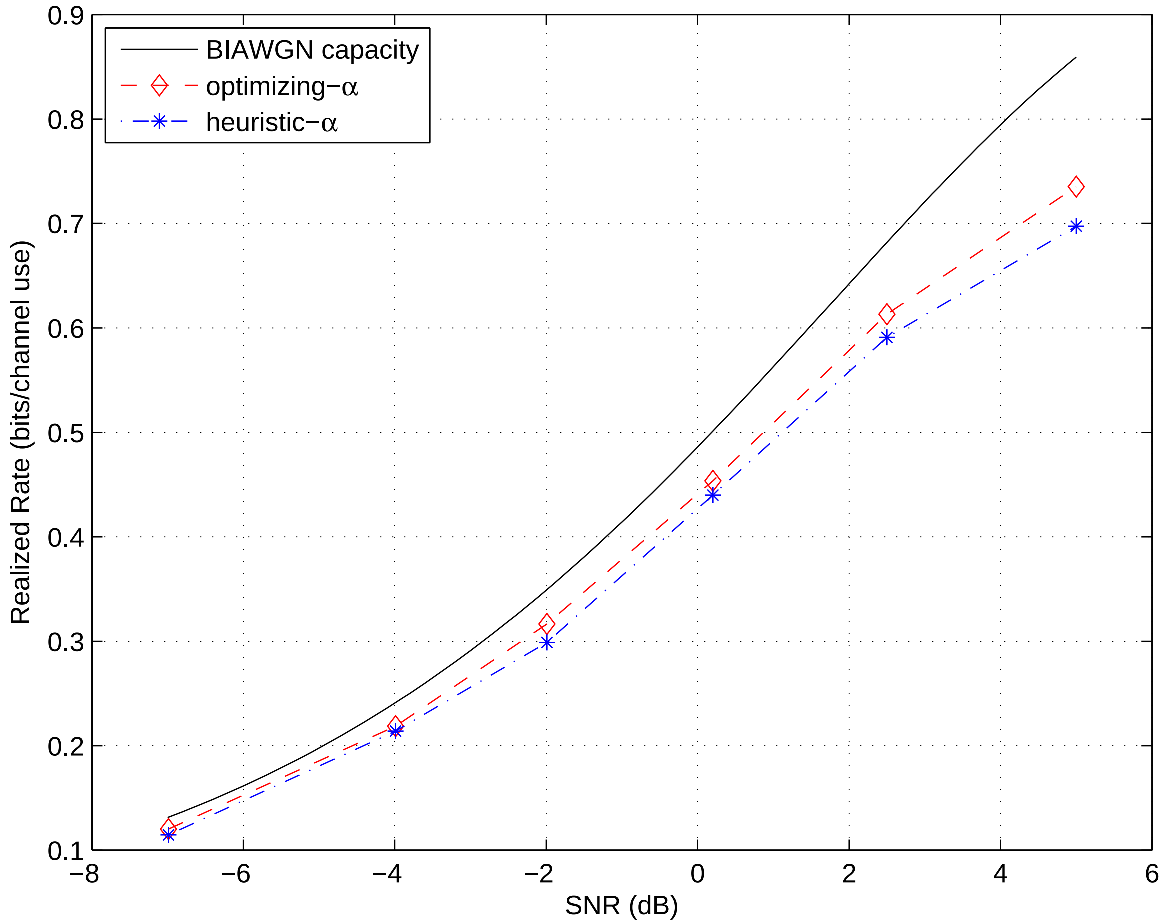

Fig. 5. Realized rates of Raptor codes constructed respectively using heuristic-α and optimizing-α approaches.

large range of discretized values of α (the sampling interval of α is chosen to be 0.00001 in our approach), where for each α the optimization problem reduces to the original linear program. We refer to this strategy as the *optimizing-*α approach.

We performed code design and simulations for both the heuristic-α and optimizing-α approaches for various values of channel SNR where μ_0 is chosen to be 30 and D chosen to be 200. The LDPC precode selected is a rate-0.95 left-4-regular right-Poisson code. The number of input symbols k' is set to $10,000$ and the information block length k is set to $9,500$. We note that although the choice of LDPC code and μ_0 are rather arbitrary, we believe that they are reasonable for the selected channel SNR values.

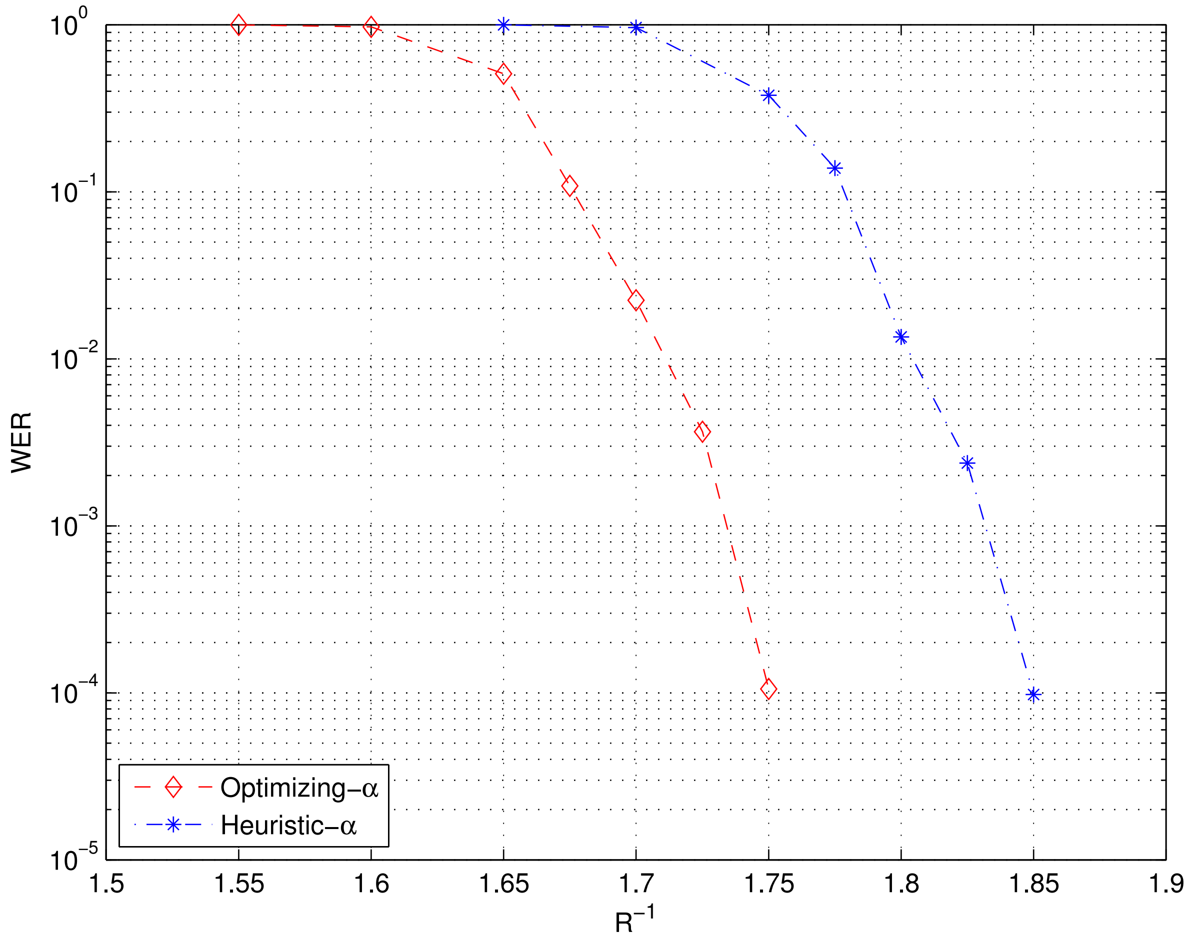

Fig. 6. Word error rate of the Raptor code as a function of $1/R$ for the two approaches over channel with SNR=2.5 dB.

The realized rate of Raptor codes designed using the heuristic-α and optimizing-α approaches is compared with each other in Fig. 5. It can be seen that the optimizing-α approach indeed uniformly outperforms the heuristic-α approach across all channel SNRs. In fact, the advantage of the optimizing-α approach over the heuristic-α approach is seen more pronounced when inspecting their word error rates, as in Fig. 6 at any given truncated code length. Here the word error rate (WER) at a given code rate R is defined as the probability of word error when the code is truncated to length k/R. Each point in the WER curves is obtained by simulating the transmission of 30,000 codewords.

References

[1] O. Etesami and A. Shokrollahi, "Raptor codes on binary memoryless symmetric channels," *IEEE Trans. Inform. Theory*, vol. 52, no. 5, pp. 2033–2051, May 2006.

[2] M. Luby, "LT codes," in *The 43rd Annual IEEE Symp. Found. Comp. Sci.*, 2002, pp. 271–282.

[3] A. Shokrollahi, "Raptor codes," *IEEE Trans. Inform. Theory*, vol. 52, no. 6, pp. 2551–2567, Jun. 2006.

[4] S. Lin and D. Costello, *Error control coding*. Prentice Hall, Apr. 2004.

[5] S. C. Draper, B. J. Frey, and F. R. Kschischang, "Efficient variable length channel coding for unknown DMCs," in *Proc. IEEE Int. Symp. Inform. Theory*, 2004, p. 379.

[6] J. Castura and Y. Mao, "Rateless coding over fading channels," *IEEE Commun. Lett.*, vol. 10, no. 1, Jan. 2006.

[7] A. Lapidoth and P. Narayan, "Reliable communication under channel uncertainty," *IEEE Trans. Inform. Theory*, vol. 44, no. 6, pp. 2148–2175, Oct. 1998.

[8] R. Palanki and J. S. Yedidia, "Rateless codes on noisy channels," in *Proc. IEEE Int. Symp. on Inform. Theory*, 2004, p. 37.

[9] J. Castura and Y. Mao, "Rateless coding for wireless relay channels," in *Proc. IEEE Int. Symp. Inform. Theory*, Adelaide, Australia, Sep. 2005.

[10] S. ten Brink, "Designing iterative decoding schemes with the extrinsic information transfer chart," *AEÜ Int. Jour. Electron. Commun.*, vol. 54, no. 6, pp. 389–398, Nov. 2000.

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

[12] M. Ardakani and F. Kschischang, "A more accurate one-dimensional analysis and design of irregular LDPC codes," *IEEE Trans. Commun.*, vol. 52, no. 12, pp. 2106–2114, Dec. 2004.

[13] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 498–519, Feb 2001.

[14] A. Shokrollahi, 2006, private communication.