Optimal Puncturing Ratios and Energy Distribution for Multiple Parallel Concatenated Codes

Fredrik Brännström, Lars K. Rasmussen, and Alex J. Grant

Abstract

Extrinsic information transfer (EXIT) charts have been used extensively for designing concatenated coding schemes with iterative decoding. The area between the two transfer curves has been shown to approximate the gap to channel capacity. A curve-fitting procedure on EXIT charts was previously suggested for designing low density parity check codes. In this paper, we develop a similar approach for shaping the EXIT charts of multiple parallel concatenated codes (MPCCs) with two or more constituent codes. Random puncturing and unequal energy distributions across parallel coding streams provide additional degrees of freedom for manipulating the EXIT functions of the constituent codes. A search over all rate-one convolutional codes of memory length four or less is performed, identifying all codes with unique EXIT functions. Another search for good combinations of constituent codes is subsequently conducted. Optimal constituent codes, puncturing ratios, and energy distributions are found in terms of minimizing the average signal-to-noise ratio threshold required for convergence, leading to simple MPCCs over a wide range of code rates. The best rate-1/2 code found has a 0.15 dB gain over the original turbo code with only half the decoding complexity. Another example shows a 0.5 dB gain obtained just by optimizing the energy distribution.

Index Terms

Code search, EXIT chart, iterative decoding, parallel concatenated codes, puncturing, unequal energy distribution.

F. Brännström and L. K. Rasmussen are supported by the Swedish Research Council under Grant 621-2001-2976. F. Brännström is also partly supported by Personal Computing and Communication (PCC++) under Grant PCC-0201-09. L. K. Rasmussen and A. J. Grant are supported by the Australian Government under ARC Grant DP0344856. The material in this paper was presented in part at the IEEE International Symposium on Information Theory, Chicago, IL, June/July 2004, and at the IEEE International Conference on Communications, Seoul, Korea, May 2005.

F. Brännström was with the Department of Computer Engineering, Chalmers University of Technology, Sweden. He is now with the Department of Signals and Systems, Chalmers University of Technology, SE-412 96 Göteborg, Sweden (e-mail: fredrikb@chalmers.se).

L. K. Rasmussen was with the Department of Computer Engineering, Chalmers University of Technology, SE-412 96 Göteborg, Sweden. He is now with the Institute for Telecommunications Research, University of South Australia, Mawson Lakes, SA 5095, Australia (e-mail: Lars.Rasmussen@unisa.edu.au).

A. J. Grant is with the Institute for Telecommunications Research, University of South Australia, Mawson Lakes, SA 5095, Australia (e-mail: Alex.Grant@unisa.edu.au).

I. INTRODUCTION

Since the invention of parallel concatenated turbo codes [1], the turbo principle has been extended to symmetric multiple parallel concatenated codes (MPCCs)[1] [2, 3] where all constituent codes are identical, and to asymmetric MPCCs [4] where the constituents can be different. The original turbo code [1] transmits all uncoded (systematic) bits and punctures half of the coded bits from the two constituents to raise the code rate from $1/3$ to $1/2$. MPCCs can have systematic doping [5], where some of the parity bits are replaced by systematic bits preserving the code rate. The ratio between the number of systematic bits replacing parity bits and the number of coded bits is usually referred to as the doping ratio, d. Usually the doping ratio is fixed to a small number [3–5], chosen to achieve some satisfactory level of performance. In [6], the performance for different doping ratios, $0 \leq d \leq 1$, is evaluated using approximated error-floors for maximum likelihood decoding.

Parallel concatenated codes (PCCs) with two constituent codes can be analyzed using two-dimensional extrinsic information transfer (EXIT) charts [7]. The signal-to-noise ratio (SNR) convergence threshold is estimated by tracing the evolution of mutual information (MI) in the EXIT chart. The EXIT chart for an MPCC with N constituents is N-dimensional, and thus, conventional EXIT chart analysis is complicated. Such N-dimensional charts have been used for analysis of $N = 3$ symmetric and asymmetric MPCCs [8]. Convergence analysis for codes with $N \geq 3$ constituents is simplified by "projecting" the multi-dimensional EXIT charts onto a single two-dimensional chart as shown in [9–12].

In [13] a curve-fitting procedure on EXIT charts was suggested for optimizing joint coding and modulation schemes based on irregular low-density parity-check codes. The approach was inspired by the area theorem for the erasure channel, stating that the area between the two transfer curves is equal to the gap to channel capacity [14]. The EXIT curve for the inner code was shown to be a weighted sum over the EXIT curves for each variable node degree. Curve fitting was then used for minimizing the area between the two EXIT curves, leading to a corresponding variable node degree distribution.

In this paper, we develop a similar approach for shaping the EXIT charts of parallel concatenated convolutional codes. Random puncturing across parallel coding streams provides N degrees of freedom for manipulating the EXIT functions of the constituent codes. Each of the N constituent codes in an MPCC can have their own puncturing ratio independent of each other and independent of the puncturing ratio of the systematic bits [11, 15, 16]. Furthermore, in previous work binary phase-shift keying (BPSK) symbols with equal symbol energy was used (uniform energy distribution). However, the constituent codes can have unequal energy distributions where each of the bit sequences (systematic or parity) use different transmitted energy under an average overall transmitted energy constraint. Finally, a search over all rate-one convolutional codes of memory length four or less is performed, identifying all codes with unique EXIT functions. The search shows that there are 98 classes of rate-one convolutional codes (CCs) of memory four or less that should be considered as constituent codes in multiple concatenated codes (MCCs). This set of CCs and their EXIT functions are subsequently used to jointly optimize the energy distribution and the $N + 1$ puncturing ratios, subject to minimizing the required

[1]An MPCC contains an arbitrary number ($N \geq 2$) of constituent codes concatenated in parallel.

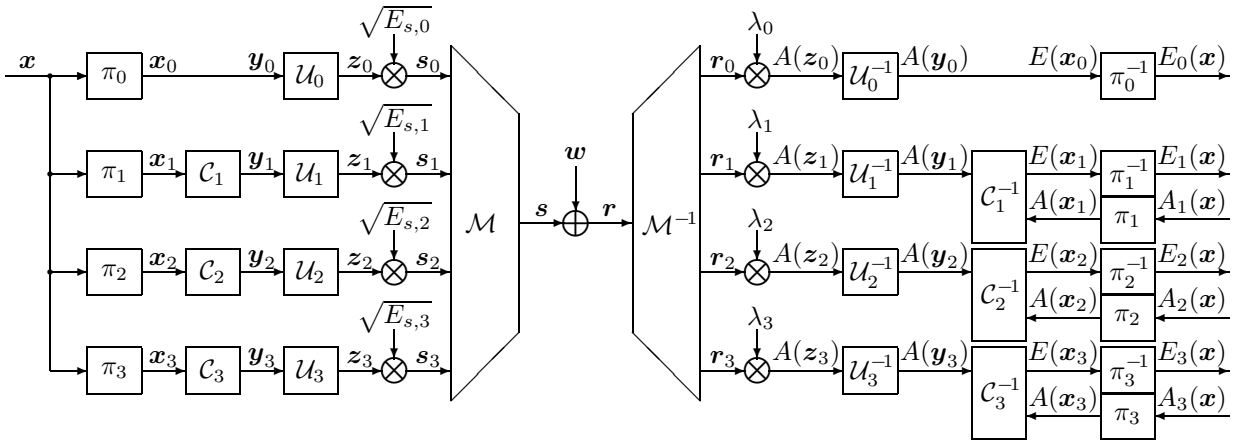

Fig. 1. Three parallel concatenated codes with puncturing.

SNR to reach a target bit-error rate (BER). A specific choice of N constituents will then have an achievable SNR-rate region within which it is possible to construct MPCCs with low decoding complexity for any desired code rate.

The remainder of the paper is organized as follows. In Section II the notation is introduced and the system model is described. In Section III it is shown that the prior MIs for punctured sequences are simple linear functions of the prior MIs of the un-punctured sequences. It is also shown that the extrinsic MIs for randomly punctured sequences can be expressed by the EXIT functions of the un-punctured constituent codes. The code search for identifying all classes of rate-one CCs is discussed in Section IV. Within each class, three different types of CCs are recognized. The optimization problem formulated for finding the optimal puncturing ratios and energy distributions to minimize the required SNR to reach a target BER is stated and applied to different example scenarios. The results show several MPCCs with low decoding complexity and good performance in terms of BER. Concluding remarks are presented in Section V.

II. System Model

Consider a system with N parallel concatenated constituent codes $\mathcal{C}_n$, transmitting binary data over an additive white Gaussian noise (AWGN) channel. Figure 1 shows an example with three components concatenated in parallel. The source bits are divided into blocks of L bits, $\boldsymbol{x} \in \{-1,+1\}^L$ and $N+1$ interleavers[2] permute the source sequence into $N+1$ different sequences $\boldsymbol{x}_n = \pi_n(\boldsymbol{x})$, $n = 0, 1, \ldots, N$. Encoder $\mathcal{C}_n$ maps a sequence of L input bits $\boldsymbol{x}_n \in \{-1,+1\}^L$ to a sequence of L/R_n output bits $\boldsymbol{y}_n \in \{-1,+1\}^{L/R_n}$, where R_n is the rate of code $n = 1, 2, \ldots, N$. The uncoded (systematic) bit sequence is denoted $\boldsymbol{x}_0$ and $R_0 = 1$, since it corresponds to the "code rate" of the systematic bits. Individual elements of these sequences are denoted by $x_{n,i}$, $i = 1, 2, \ldots, L$ and $\boldsymbol{x}_n = [x_{n,1}, x_{n,2}, \ldots, x_{n,L}]$, $n = 0, 1, \ldots, N$. This notation is naturally extended to all other sequences.

With reference to Figure 1, $\mathcal{U}_n$ denotes a random *puncturer* for the sequence $\boldsymbol{y}_n$ with a *puncturing ratio* $\delta_n \in [0, 1]$, for $n = 0, 1, \ldots, N$. $1 - \delta_n$ denotes the fraction of bits in $\boldsymbol{y}_n$ that are randomly punctured [11]. If

[2]Note that π_0 and π_1 can be removed since it is the relative interleaving between the encoders that is important [1].

$\delta_n = 0.8$, it means that 20% of the bits in $\boldsymbol{y}_n$ are removed, i.e., $\boldsymbol{z}_n$ contains only 80% of the bits from $\boldsymbol{y}_n$, namely $\boldsymbol{z}_n \in \{-1,+1\}^{L_n}$, where $L_n \triangleq \frac{L\delta_n}{R_n}$.

The output from each puncturer is multiplied by individual signal amplitudes, $\sqrt{E_{s,n}}$ for $n = 0, 1, \ldots, N$, where $E_{s,n} \neq 0$ is the transmitted symbol energy for the output bits from $\mathcal{U}_n$, i.e., $\boldsymbol{s}_n = E_{s,n}\boldsymbol{z}_n$ for $n = 0, 1, \ldots, N$. In Figure 1, $\mathcal{M}$ represents a multiplexer, converting the $N+1$ streams of BPSK symbols into a single stream of symbols to be transmitted successively over the AWGN channel, $\boldsymbol{s} = [\boldsymbol{s}_0, \boldsymbol{s}_1, \ldots, \boldsymbol{s}_N]$. The overall code rate of the punctured parallel system is [11]

$$R = \frac{L}{\sum_{n=0}^{N} L_n} = \left(\sum_{n=0}^{N} \frac{\delta_n}{R_n} \right)^{-1}. \tag{1}$$

The receiver's matched filter output is $\boldsymbol{r} = \boldsymbol{s} + \boldsymbol{w}$, where each element in $\boldsymbol{w}$ is a zero-mean Gaussian noise sample with variance $\sigma_w^2 = N_0/2$. The average energy per source bit is

$$E_b = \frac{1}{L} \sum_{n=0}^{N} L_n E_{s,n} = \sum_{n=0}^{N} \frac{\delta_n}{R_n} E_{s,n}, \tag{2}$$

and the average SNR is defined as

$$\gamma_b \triangleq \frac{E_b}{N_0} = \sum_{n=0}^{N} \frac{\delta_n}{R_n} \gamma_{s,n}, \tag{3}$$

where $\gamma_{s,n} = E_{s,n}/N_0$. Define

$$\psi_n \triangleq \frac{E_{s,n}}{\sum_{k=0}^{N} E_{s,k}} = \frac{\gamma_{s,n}}{\sum_{k=0}^{N} \gamma_{s,k}} = \frac{\gamma_{s,n}}{\Gamma}, \tag{4}$$

as the fraction of transmitted average symbol energy for the output bits from $\mathcal{U}_n$, where $\Gamma \triangleq \sum_{k=0}^{N} \gamma_{s,k}$. Note that $0 < \psi_n < 1$ for all $n = 0, 1, \ldots, N$ and $\sum_{n=0}^{N} \psi_n = 1$. From now on this will be referred to as the energy distribution. Using (3) and (4), the average SNR can be rewritten as

$$\gamma_b = \Gamma \sum_{j=0}^{N} \frac{\delta_j}{R_j} \psi_j. \tag{5}$$

Combining (4) and (5) and solving for $\gamma_{s,n}$ gives

$$\gamma_{s,n} = \frac{\psi_n}{\sum_{j=0}^{N} \frac{\delta_j}{R_j} \psi_j} \gamma_b. \tag{6}$$

For later use, collect the code rates, the puncturing ratios, and the energy distributions in three vectors with $N+1$ elements each,

$$\mathbf{R} \triangleq [R_0, R_1, \ldots, R_N], \tag{7}$$

$$\mathbf{\Delta} \triangleq [\delta_0, \delta_1, \ldots, \delta_N], \tag{8}$$

$$\mathbf{\Psi} \triangleq [\psi_0, \psi_1, \ldots, \psi_N]. \tag{9}$$

Let $A(\boldsymbol{x}_n) = [A(x_{n,1}), A(x_{n,2}), \ldots, A(x_{n,L})] \in \mathbb{R}^L$ denote a sequence of *a priori* information corresponding to $\boldsymbol{x}_n$. Likewise, $E(\boldsymbol{x}_n) = [E(x_{n,1}), E(x_{n,2}), \ldots, E(x_{n,L})] \in \mathbb{R}^L$ is a sequence of *extrinsic* information for $\boldsymbol{x}_n$. Priors and extrinsics are here represented as sequences of log-likelihood ratios (LLRs) [11]. This notation is naturally extended to other sequences, $\boldsymbol{x}$, $\boldsymbol{y}_n$ and $\boldsymbol{z}_n$, $n = 0, 1, \ldots, N$.

The demultiplexer, $\mathcal{M}^{-1}$ splits the matched filter outputs $\boldsymbol{r} = [\boldsymbol{r}_0, \boldsymbol{r}_1, \ldots, \boldsymbol{r}_N]$ into the corresponding $N+1$ streams as illustrated in Figure 1. The prior LLRs from the channel, $A(\boldsymbol{z}_n)$, $n = 0, 1, 2, \ldots, N$, are calculated from the matched filter outputs as depicted in Figure 1

$$A(z_{n,i}) \triangleq \ln\left(\frac{p_{r|z}(r_{n,i}|z_{n,i}=+1)}{p_{r|z}(r_{n,i}|z_{n,i}=-1)}\right) = \lambda_n r_{n,i}, \tag{10}$$

where $p_{r|z}(r_{n,i}|z_{n,i})$ is the conditional probability density function of the matched filter output $r_{n,i} = \sqrt{E_{s,n}}z_{n,i} + w_i$ and λ_n is the channel reliability constant defined as [1, 7]

$$\lambda_n \triangleq \frac{\sqrt{16E_{s,n}}}{N_0}. \tag{11}$$

To be able to calculate or approximate the prior channel LLRs in (10), $E_{s,n}$ and N_0 need to be known or estimated [17]. Combining (10) and (11) it is clear that $A(\boldsymbol{z}_n)$ is a mixed Gaussian random variable that is constant during the decoding process

$$A(z_{n,i}) = \lambda_n(\sqrt{E_{s,n}}z_{n,i} + w_i) = \frac{\sigma_n^2}{2}z_{n,i} + v_i, \tag{12}$$

where $\sigma_n^2 = 8\gamma_{s,n}$ is the variance of the zero-mean Gaussian v_n [7].

To create $A(\boldsymbol{y}_n)$, $n = 0, 1, \ldots, N$, the *depuncturer* $\mathcal{U}_n^{-1}$ in Figure 1 inserts zeros at the positions in $A(\boldsymbol{z}_n)$ where the punctured bits are located. The decoder consists of N *a posteriori* probability (APP) decoders $\mathcal{C}_n^{-1}$ [18], interconnected by *interleavers* π_n and *deinterleavers* π_n^{-1}, $n = 1, 2, \ldots, N$. Upon activation[3], decoder $\mathcal{C}_n^{-1}$ uses its code constraint and the most recent priors $A(\boldsymbol{x}_n) = \pi_n(A_n(\boldsymbol{x}))$ and $A(\boldsymbol{y}_n)$ [19] to update the extrinsics on the source bits [18], $E_n(\boldsymbol{x}) = \pi_n^{-1}(E(\boldsymbol{x}_n))$, where

$$A_n(\boldsymbol{x}) = \sum_{\substack{j=0 \\ j\neq n}}^{N} E_j(\boldsymbol{x}). \tag{13}$$

Prior to decoding, the extrinsic values are set to zero, $E_n(\boldsymbol{x}) = \{0\}^L$, $n = 1, 2 \ldots, N$. The only extrinsic sequence with non-zero elements prior to decoding is $E_0(\boldsymbol{x}) = \pi_0^{-1}(A(\boldsymbol{y}_0))$. In contrast to a serially concatenated code, the extrinsic output from the decoder, $E(\boldsymbol{y}_n)$, in a parallel concatenated code is never used and is therefore not included in Figure 1 [10, 11].

Let $D(\boldsymbol{x}) \in \mathbb{R}^L$ denote the LLR for the decision statistics of the source bits $\boldsymbol{x}$. The decision statistics are updated after each activation according to

$$D(\boldsymbol{x}) = \sum_{j=0}^{N} E_j(\boldsymbol{x}). \tag{14}$$

The hard decision $\hat{x}_i$ on source bit x_i is $D(x_i) \underset{\hat{x}_i=-1}{\overset{\hat{x}_i=+1}{\gtrless}} 0$ and the system performance is measured in BER, i.e., the probability $P_b \triangleq \Pr(\hat{x}_i \neq x_i)$.

[3]The term *activation* is used instead of iteration. In a system with two constituents, one iteration is the same as two activations, one for each of the two decoders.

III. Mutual Information and EXIT Functions

Let

$$I_{A(x_n)} \triangleq \frac{1}{L}\sum_{i=1}^{L} I(x_{n,i}; A(x_{n,i})), \tag{15}$$

$$I_{E(x_n)} \triangleq \frac{1}{L}\sum_{i=1}^{L} I(x_{n,i}; E(x_{n,i})) \tag{16}$$

denote the average MI between the input bits and the prior values, and the extrinsic values, respectively, (similarly for $\boldsymbol{y}_n$ and $\boldsymbol{z}_n$). We shall refer to these as prior and extrinsic MIs [11, 12]. Since we are only dealing with bits, all MIs are between zero and one, $0 \le I_{A(x_n)} \le 1$ and $0 \le I_{E(x_n)} \le 1$, for all $n = 0, 1, \ldots, N$.

A. Maximum Rate and Minimum Required SNR

A code is characterized by its EXIT functions [11, 12],

$$T_{x_n} : [0,1]^2 \mapsto [0,1], \tag{17}$$

$$T_{y_n} : [0,1]^2 \mapsto [0,1], \tag{18}$$

where, for example, the extrinsic MI, $I_{E(x_n)}$ for decoder $n = 1, 2, \ldots, N$ is the EXIT function of the prior MIs $I_{A(x_n)}$ and $I_{A(y_n)}$

$$I_{E(x_n)} = T_{x_n}\big(I_{A(x_n)}, I_{A(y_n)}\big). \tag{19}$$

In practice, this EXIT function is obtained by Monte-Carlo simulations[4] of the constituent code for all values of $0 \le I_{A(x_n)} \le 1$ and $0 \le I_{A(y_n)} \le 1$ by modelling the priors as Gaussian [7], similar to (10).

It can be shown that

$$I_{A(z_n)} = J\big(\sqrt{8\gamma_{s,n}}\big), \quad \text{where} \tag{20}$$

$$J(\sigma) = 1 - \frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{+\infty} e^{-\frac{(\xi-\sigma^2/2)^2}{2\sigma^2}} \log_2\big(1+e^{-\xi}\big)\, d\xi, \tag{21}$$

according to [7]. $J(\sigma)$ is monotonically increasing and therefore has a unique inverse, $\sigma = J^{-1}(I)$. Unfortunately, J and J^{-1} can not be expressed in closed form, but they can be closely approximated as suggested in [11, 12]. Note that (20) is also the constellation-constrained capacity, $0 \le C_{\text{BPSK}} \le 1$, of a system using BPSK modulation with symbol energy $E_{s,n}$ over an AWGN channel, $C_{\text{BPSK}} = J\big(\sqrt{8\gamma_{s,n}}\big)$.

Since the system in Figure 1 uses $N+1$ different symbol energies (L_n symbols with $E_{s,n}$ for $n = 0, 1, \ldots, N$), the maximum rate of the concatenated code can be expressed as

$$\begin{aligned} C(R, \boldsymbol{\Delta}, \boldsymbol{\Psi}, \gamma_b) &= \frac{1}{\sum_{k=0}^{N} L_k}\sum_{n=0}^{N} L_n J\big(\sqrt{8\gamma_{s,n}}\big) \\ &= \frac{1}{\sum_{k=0}^{N} \frac{\delta_k}{R_k}}\sum_{n=0}^{N} \frac{\delta_n}{R_n} J\left(\sqrt{8\frac{\psi_n}{\sum_{j=0}^{N}\frac{\delta_j}{R_j}\psi_j}\gamma_b}\right). \end{aligned} \tag{22}$$

[4]For certain codes and simple channel models, it is possible to compute the EXIT functions [20, 21].

A special case of (22) is when the energy distribution is uniform, $E_{s,n} = RE_b$, i.e., $\psi_n = 1/(N+1)$ for all $n = 0, 1, \ldots, N$. This uniform energy distribution is denoted by $\mathbf{\Psi}_0$,

$$\mathbf{\Psi}_0 \triangleq \left[\frac{1}{N+1}, \frac{1}{N+1}, \ldots, \frac{1}{N+1}\right]. \tag{23}$$

The systems treated in [11, 15] use $\mathbf{\Psi}_0$, where the maximum rate is equal to C_{BPSK}

$$C(R, \mathbf{\Delta}, \mathbf{\Psi}_0, \gamma_b) = J\left(\sqrt{8\frac{1}{\sum_{j=0}^{N}\frac{\delta_j}{R_j}}\gamma_b}\right) = J\left(\sqrt{8R\gamma_b}\right). \tag{24}$$

Let $\widetilde{\gamma}_b$ denote the minimum required SNR for a given $\mathbf{R}$, $\mathbf{\Delta}$ and $\mathbf{\Psi}$. $\widetilde{\gamma}_b$ can be found by letting $C(R, \mathbf{\Delta}, \mathbf{\Psi}, \widetilde{\gamma}_b) = R$ and solving for $\widetilde{\gamma}_b$. If the energy distribution is uniform, $\widetilde{\gamma}_b$ can be expressed in closed form using (24)

$$\widetilde{\gamma}_b = \frac{J^{-1}(R)^2}{8R}. \tag{25}$$

B. EXIT Functions

The puncturer, $\mathcal{U}_n$ in Figure 1, only removes bits from randomly chosen positions of the sequence $\boldsymbol{y}_n$, and $\mathcal{U}_n^{-1}$ adds zeros in $A(\boldsymbol{y}_n)$ at these positions. Therefore, the relationship between $I_{A(y_n)}$ and $I_{A(z_n)}$ is linear [11]

$$I(y_{n,i}; A(y_{n,i})) \triangleq \begin{cases} I(z_{n,i}; A(z_{n,i})) & \text{if } y_{n,i} \text{ is transmitted,} \\ 0 & \text{if } y_{n,i} \text{ is punctured.} \end{cases} \tag{26}$$

Since δ_n denotes the fraction of non-zero elements in $A(\boldsymbol{y}_n)$, the average MI $I_{A(y_n)}$ can be expressed as

$$I_{A(y_n)} = \frac{R_n}{L}\sum_{i=1}^{L} I(y_{n,i}; A(y_{n,i})) = \delta_n I_{A(z_n)} = \delta_n J\left(\sqrt{8\gamma_{s,n}}\right), \tag{27}$$

for all $n = 0, 1, \ldots, N$. The linear relationship in (27) assumes random puncturing and infinitely large interleavers, similar to the assumptions for the EXIT chart analysis [7]. The average MI is not affected by an interleaver or a deinterleaver. Therefore, $I_{E(x_n)} = I_{E_n(x)}$ and $I_{A(x_n)} = I_{A_n(x)}$. Since the prior values are sums of N extrinsic values (13), they are modelled as sums of N biased Gaussian random variables. Using (21) and its inverse,

$$I_{A(x_n)} = J\left(\sqrt{\sum_{\substack{j=0 \\ j\neq n}}^{N} J^{-1}\left(I_{E(x_j)}\right)^2}\right), \tag{28}$$

since $I_{E(x_j)}$ need to be added in the *variance domain* [8, 11, 12]. Letting $I_{D(x)} \triangleq \frac{1}{L}\sum_{i=0}^{L} I(x_i; D(x_i))$, then similarly

$$I_{D(x)} = J\left(\sqrt{\sum_{j=0}^{N} J^{-1}\left(I_{E(x_j)}\right)^2}\right), \tag{29}$$

where $I_{D(x)} = 1.0$ corresponds to having full information about the source bits, i.e., P_b is close to zero. If the decision statistics are close to a Gaussian model as in (12), the BER can be approximated by [7],

$$P_b \approx Q\left(\frac{J^{-1}\left(I_{D(x)}\right)}{2}\right), \tag{30}$$

where

$$Q(\phi) \triangleq \frac{1}{\sqrt{2\pi}} \int_{\phi}^{\infty} e^{-\xi^2/2}\, d\xi \tag{31}$$

is the Gaussian Q-function.

The N EXIT functions for a parallel concatenated system can therefore be expressed as

$$I_{E(x_n)} = T_{x_n}\left(\overbrace{J\left(\sqrt{J^{-1}\left(\delta_0 J\left(\sqrt{8\frac{\psi_0}{\sum_{j=0}^{N}\frac{\delta_j}{R_j}\psi_j}\gamma_b} \right) \right)^2 + \sum_{\substack{j=1 \\ j\neq n}}^{N} J^{-1}\left(I_{E(x_j)}\right)^2} \right)}^{I_{A(x_n)}}, \overbrace{\delta_n J\left(\sqrt{8\frac{\psi_n}{\sum_{j=0}^{N}\frac{\delta_j}{R_j}\psi_j}\gamma_b} \right)}^{I_{A(y_n)}} \right) \tag{32}$$

for all $n = 1, 2, \ldots, N$, using (6), (27), (28) and the fact that $I_{E(x_0)} = \delta_0 J\left(\sqrt{8\gamma_{s,0}}\right)$ [11]. Note that if $\mathbf{\Delta}$, $\mathbf{\Psi}$, and γ_b, together with the EXIT functions of the codes, T_{x_n}, are known[5], $I_{E(x_n)}$ in (32) depends only on the remaining $I_{E(x_j)}$, $j \neq n$.

A uniform energy distribution (23) together with a fixed γ_b gives $N+1$ degrees of freedom in $\mathbf{\Delta}$ to change the mutually dependence of the N EXIT functions in (32) over a range of code rates R [11, 15]. From (32) it is also clear that the energy distribution $\mathbf{\Psi}$ provides additional N degrees of freedom[6] to modify the mutually dependence of the EXIT functions facilitating possible performance improvements of the system [16].

IV. Code Design

It has been shown that, under certain conditions, the area between the two curves in an EXIT chart is related to the SNR loss as compared to the capacity [14, 21, 22]. A small area, without intersections of the two curves in the EXIT chart, corresponds to a low convergence threshold close to the capacity limit. The design of good concatenated codes is therefore similar to a curve-fitting problem minimizing the area between the two curves.

With conventional doping, some of the coded bits are replaced by uncoded (systematic) bits, thus preserving the code rate [5, 6]. The doping ratio, d, is defined as the ratio between the number of transmitted systematic bits and the total number of transmitted bits. Changing the doping ratio influences the shape of the EXIT chart for the code, hence affecting the performance and the convergence threshold. This means that there is one or several optimal doping ratios, which give the lowest possible convergence threshold. The doping ratio is usually just fixed to an arbitrary small number yielding satisfactory performance [4, 5]. In contrast to previous approaches, having only one single parameter in the doping ratio d, there are a total of $2N$ degrees of freedom in $\mathbf{\Delta}$ and $\mathbf{\Psi}$, for a specific code rate, to shape the curves in the EXIT chart.

Consider an MPCC with N components using puncturing and a BPSK mapper, as in Figure 1. For this system there are N EXIT functions as stated in (32). The evolution of MI can be tracked using the EXIT functions for a fixed γ_b, $\mathbf{\Delta}$, and $\mathbf{\Psi}$. All extrinsic MIs converge to fixed values, independent of activation schedule, as long as the EXIT functions are monotonically non-decreasing and an unlimited number of decoder activations is

[5]Each T_{x_n} corresponds to $\mathcal{C}_n$ with code rate R_n, i.e., $\mathbf{R}$ is known if all T_{x_n} are known.

[6]The sum of all ψ_n, $n = 0, 1, \ldots, N$, is equal to one, and that results in N degrees of freedom.

allowed [9, 11, 12]. This implies that the MI on the decision statistics also converges to a fixed value according to (29). This fixed value is defined as the convergence point, $I_D^\star$, and for a specific set of codes and code rate R, $I_D^\star$ is a function of $\mathbf{\Delta}$, $\mathbf{\Psi}$, and γ_b, i.e.,

$$I_D^\star \triangleq f(\mathbf{\Delta}, \mathbf{\Psi}, \gamma_b)\,. \tag{33}$$

A. Constituent Codes

CCs can be classified according to their *input-output weight enumeration function* (IOWEF) [23]. In a system with a single CC, the CC that has the maximum free distance for a specified code rate and complexity is usually chosen. Tables of these optimal CCs are easily found in the literature [24]. However, designing concatenated codes based on EXIT chart analysis only rely on the EXIT functions of the constituent codes. A CC with a low free distance can result in better performance when concatenated with other CCs, than a CC with larger free distance. Further, two CCs with different IOWEF can have almost identical EXIT functions, resulting in identical convergence behavior predicted by EXIT charts. For these reasons, it is important consider all possible CCs of a certain rate and complexity, not only the ones that have the largest free distance.

In this paper, only CCs with $R_n = 1$ for $n = 0, 1, \ldots, N$, are considered as constituent codes, i.e., $\mathbf{R} = \mathbf{1}$. It is straightforward to include other code rates or types of constituent codes. A CC with $R_n = 1$ is represented by two generator polynomials, one numerator and one denominator, both given in octal representation. A CC with ν delay elements is represented by $\nu + 1$ binary digits. With this description, $13 \equiv 1011$ (octal and binary form, respectively) represents $1 + \mathrm{D}^2 + \mathrm{D}^3$ [7]. Note that sometimes the octal representation of the generator polynomials are defined the reversed way, e.g., as in [25]. A *unitary polynomial* is just a '1'. The binary representation of a unitary polynomial for a CCs with memory ν is therefore a one followed by ν zeros. In octal representation the unitary polynomials are 2, 4, 10 and 20 for $\nu = 1$, 2, 3 and 4, respectively.

There are three types of CCs with code rate $R_n = 1$: feed-forward, feed-backward and combined feed-forward/feed-backward. They are here referred to as type F, B, and C, respectively. A CC of type F has a unitary denominator, a CC of type B has a unitary numerator, while a CC of type C has no unitary polynomials as its numerator or denominator. Using this notation, CC$(7/4)$ is type F with generator polynomial $1 + \mathrm{D} + \mathrm{D}^2$, CC$(4/7)$ is type B with generator polynomial $1/(1 + \mathrm{D} + \mathrm{D}^2)$, and CC$(7/5)$ is type C with generator polynomial $(1 + \mathrm{D} + \mathrm{D}^2)/(1 + \mathrm{D}^2)$. Figure 2 shows the EXIT functions for these three CCs generated by Monte Carlo Simulations [11, 12]. If these three CCs are used as constituent codes in Figure 1, the MPCC is denoted by PCC$(1 + 7/4 + 4/7 + 7/5)$, where '1' represents the systematic part. The difference between the three types are here easily visualized by the EXIT functions. All three types have $T_x\big(I_{A(x)}, 0\big) = T_y\big(0, I_{A(y)}\big) = 0$ and $T_x\big(I_{A(x)}, 1\big) = T_y\big(1, I_{A(y)}\big) = 1$. CCs of type F and C have $T_x\big(0, I_{A(y)}\big) = 0$ and $T_y\big(I_{A(x)}, 1\big) = 1$, while CCs of type B and C have $T_x\big(1, I_{A(y)}\big) = 1$, and $T_y\big(I_{A(x)}, 0\big) = 0$. The impact of the various EXIT functions characteristics on the decoding convergence is discussed later in this section.

For each memory length ν there are 2^ν different polynomials and therefore $2^\nu 2^\nu$ ways of describing a rate-one CC with memory ν. This can be reduced to $2^\nu(2^\nu - 1)$ by removing all combinations with identical numerator and denominator. The number of type F CCs with memory ν is $2^\nu - 1$, since there is only one unitary

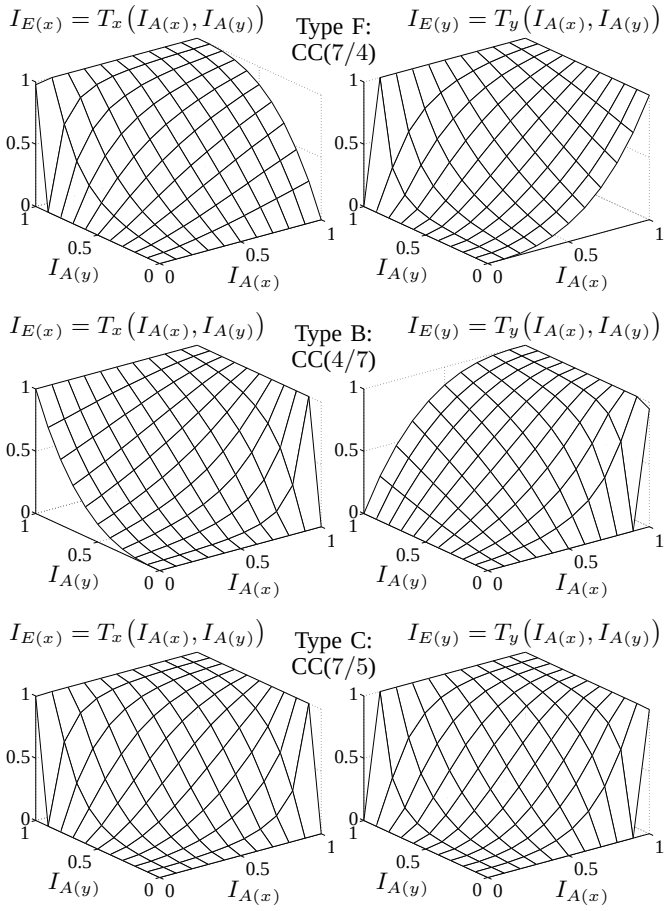

Fig. 2. EXIT functions for CC(7/4), CC(4/7), and CC(7/5).

polynomial for each ν that can be combined with all the other $2^\nu - 1$ polynomials. For the same reason, the number of type B CCs is also $2^\nu - 1$. Hence, the number of type C CCs with memory ν is $(2^\nu - 1)(2^\nu - 2)$.

In Table I, the three first columns under "Total" shows the number of CCs with a specific ν for type F, B and C, respectively. The fourth column shows the total number of CCs with a specific ν and the last row shows the total number of CCs with $\nu \leq 4$ for the three different types. This means that there are 310 different rate-one CCs with $\nu \leq 4$, where 258 of them are of type C and 26 each of type F and B.

The classification of CCs to be used as constituents in concatenated codes are here not based on the IOWEF, but instead on the volume of the difference between the EXIT functions describing two CCs. The question is now how many unique sets of EXIT functions, $\{T_x, T_y\}$, are there among the 310 CCs in Table I? In the Appendix, classes of codes with identical (or almost identical) EXIT functions are identified leading to the four last column of Table I. This classification reveals 98 sets of EXIT functions for CCs with $\nu \leq 4$. 82 being of type C, and 8 of type F and B, respectively.

To get good performance of the MPCC, i.e., a low BER in (30), $I_{D(x)} \approx 1.0$. This can only be obtained if at least one of the $I_{E(x_i)}$ in (29) is close to one. For this to happen, Figure 2 shows that one of the input arguments of (32), $I_{A(x_n)}$ or $I_{A(y_n)}$, must be close to one. The second argument, $I_{A(y_n)}$, is only close to one

TABLE I

NUMBER OF CCS WITH $R = 1$ AND $\nu = 1, 2, 3, 4$.

	Total				Unique			
ν	#F	#B	#C	Σ	#F	#B	#C	Σ
1	1	1	0	2	1	1	0	2
2	3	3	6	12	1	1	4	6
3	7	7	42	56	2	2	13	17
4	15	15	210	240	4	4	65	73
Σ	26	26	258	310	8	8	82	98

for high SNR, and the first argument is only close to one if one of the $I_{E(x_j)}$ $(j \neq n)$ in (32) is close to one. Type B and C codes have $T_{x_n}\big(1, I_{A(y_n)}\big) = 1$, but type F codes have $T_{x_n}\big(1, I_{A(y_n)}\big) \leq 1$, as illustrated in Figure 2. Consequently, at least two of the constituent codes must be of type B or C, otherwise the MPCC will not converge at low SNR.

Type F and C codes have $T_{x_n}\big(0, I_{A(y_n)}\big) = 0$, but type B codes have $T_{x_n}\big(0, I_{A(y_n)}\big) \geq 0$. If no systematic bits are transmitted ($\delta_0 = 0$), the first argument of (32) is initially be zero. This means that type B codes can be used even when no systematic bits are transmitted while type C and F codes always require that some systematic bits, i.e. $I_{A(x_n)} \neq 0$, are transmitted to converge, even for high SNR.

B. Classification of Rate-1 Convolutional Codes

Consider a MPCC where the rate of the component codes, $\mathbf{R} = [R_1, R_2, \ldots, R_N]$, the puncturing ratios, $\mathbf{\Delta} = [\delta_0, \delta_1, \ldots, \delta_N]$, and the energy distribution, $\mathbf{\Psi} = [\psi_0, \psi_1, \ldots, \psi_N]$, are fixed. The overall code rate, R, is then also fixed according to (1), since it depends only on $\mathbf{R}$ and $\mathbf{\Delta}$. Good combinations of constituent codes using these fixed $\mathbf{\Delta}$ and $\mathbf{\Psi}$ can be found by performing a search over a restricted set of constituent codes. The criterion for a good combination of constituent codes can, for example, be low convergence threshold, or a minimum number of activations required to reach a certain target BER. The restricted set of codes defines the type and rate of the constituent codes and also constraints the decoding complexity.

Tables VI–IX in the Appendix contain all classes of rate-one CCs with different EXIT functions for $\nu \leq 4$. Any CC among those listed in the second columns can be used as component codes in the design of MPCCs. The number of different combinations of N CCs, using the rule stated above that at least two CCs need to be of type B or C, can be calculated as follows. The number of ways to choose k elements from n possible elements with replacement, disregarding the order they are chosen, is [26]

$$\binom{n+k-1}{k} \triangleq \frac{(n+k-1)!}{k!(n-1)!}. \tag{34}$$

Let N_{BC} denote the number of available CCs of type B or C and let n_{BC} denote the number of CCs (of type B or C) to be included in the MPCC. The number of possible combinations can then be expressed using (34) as,

$$\binom{N_{\text{BC}} + n_{\text{BC}} - 1}{n_{\text{BC}}}. \tag{35}$$

TABLE II

TOTAL NUMBER OF WAYS, N_{TOT}, TO COMBINE N CCS WITH A MEMORY, ν, LESS THAN OR EQUAL TO 1, 2, 3 AND 4, RESPECTIVELY.

$\nu \leq$	N_{BC}	N_{F}	$N=2$	$N=3$	$N=4$	$N=5$
1	1	1	1	2	3	4
2	6	2	21	98	301	756
3	21	4	231	2 695	20 020	117 964
4	90	8	4 095	158 340	4 071 795	83 261 178

TABLE III

RATE-ONE CCS WITH DIFFERENT EXIT FUNCTIONS AND MEMORY $\nu \leq 2$.

CC	$3/2$	$7/4$	$2/3$	$4/7$	$7/5$	$7/6$	$5/7$	$6/7$
ν	1	2	1	2	2	2	2	2
type	F	F	B	B	C	C	C	C

In a similar way, let N_{F} denote the number of available CCs of type F and let n_{F} denote the number of chosen CCs of type F. The number of possible combinations is then

$$\binom{N_{\text{F}} + n_{\text{F}} - 1}{n_{\text{F}}}. \tag{36}$$

Using n_{BC} CCs of type B or C and n_{F} CCs of type F gives a total of $N = n_{\text{BC}} + n_{\text{F}}$ CCs. The total number of combinations of CCs, N_{tot}, where at least two CCs are of type B or C is then finally expressed as the product of (35) and (36) summed over all $n_{\text{BC}} = 2, 3, \ldots, N$,

$$N_{\text{tot}} = \sum_{n_{\text{BC}}=2}^{N} \binom{N_{\text{BC}} + n_{\text{BC}} - 1}{n_{\text{BC}}} \binom{N_{\text{F}} + N - n_{\text{BC}} - 1}{N - n_{\text{BC}}}. \tag{37}$$

Table II shows N_{tot} for $N = 2, 3, 4$ and 5. The second column, N_{BC}, contains the number of CCs of type B or C (collected from Table I) with a memory less than ν. The third column is also collected from Table I and contains the number of CCs of type F with a memory less than ν. From Table II it is clear that performing an exhaustive search over all combinations of CCs to find good MPCCs is computationally challenging for high ν and N. For example, there are over 83 million combinations of $N = 5$ CCs with memory $\nu \leq 4$ and $R_n = 1$ for $n = 1, 2, 3, 4, 5$.

The code search in this paper is therefore restricted to low-complexity CCs with $\nu \leq 2$. Tables VI–VIII in the Appendix contain the eight classes of rate-one CCs up to memory $\nu = 2$ that have different EXIT functions. The class representative of these eight classes are also listed in Table III. According to Table II, there are 21 combinations of $N = 2$ CCs, 98 combinations of $N = 3$ CCs and 301 combinations of $N = 4$ CCs that can be chosen from Table III.

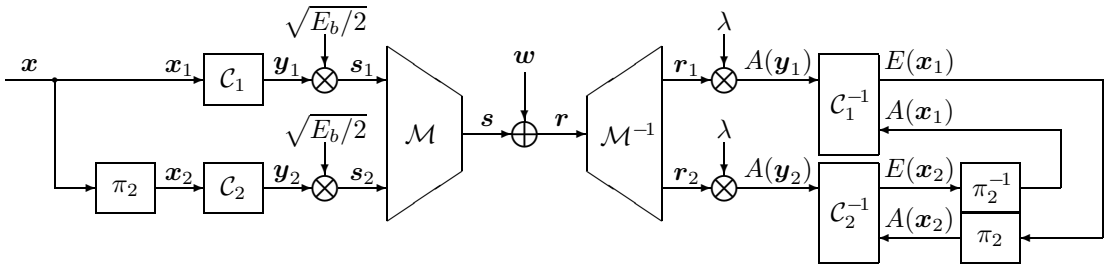

Fig. 3. Non-systematic PCC.

C. Non-Systematic PCC with Uniform Energy distribution

The code search in this sub-section finds the best[7] non-systematic PCC with uniform energy distribution at $R = 1/2$, constructed of two CCs with memory $\nu \leq 2$. This is equivalent to selecting the puncturing ratios $\mathbf{\Delta} = [0, 1, 1]$ and $\mathbf{\Psi} = \mathbf{\Psi}_0$ (defined in (23)). The simple structure of this non-systematic PCC is shown in Figure 3, where $E_{s,1} = E_{s,2} = E_b/2$ and therefore $\lambda = \sqrt{8E_b}/N_0$.

Instead of searching over all 21 combinations of $N = 2$ CCs from Table III, the search can here be restricted to the 11 combinations with at least one CC of type B and no CC of type F, since the code is non-systematic. Inspection of these 11 EXIT functions shows that PCC($1 + 4/7 + 7/5$) has the lowest convergence threshold among the 11 combinations of CCs with $S = 4 + 4$ states and $\mathbf{\Delta} = [0, 1, 1]$. This PCC can be written as PCC($4/7 + 7/5$), since no systematic bits are transmitted. Its convergence threshold is found to be around 0.4 dB [11].

A similar search is performed in [4], also using EXIT functions, but for fixed puncturing ratios. The search in [4] is made over $1\,540$ different combinations of $N = 3$ CCs and $8\,855$ different combinations of $N = 4$ CCs, in both cases with rate-one CCs that have $\nu \leq 3$. The rules for restricting the set of CCs for the search in [4] is different from the approach used here, since there are $2\,695$ and $20\,020$ combinations of $N = 3$ and $N = 4$ CCs with $\nu \leq 3$ that have different EXIT functions, according to Table II. The puncturing ratio in [4] for the systematic bits is fixed to a small number $\delta_0 = \epsilon$ (systematic doping [27]) and the remaining puncturing ratios, δ_n for $n = 1, 2, \ldots, N$, are chosen to give a code rate of $R = 1/N$ according to (1).

D. Optimal Puncturing and Energy Distribution

The previous sub-section showed that PCC($4/7 + 7/5$) gives the lowest convergence threshold among all combinations of two rate-one CCs with $\nu \leq 2$, uniform energy distribution $\mathbf{\Psi}_0$, at a fixed $\mathbf{\Delta} = [0, 1, 1]$, i.e., $R = 1/2$. However, there might be other puncturing ratios and energy distributions that give a lower convergence threshold for PCC($1 + 4/7 + 7/5$) at $R = 1/2$. Such schemes can be found through an exhaustive search of all combinations of δ_n for $n = 0, 1, \ldots, N$ that fulfills the code rate in (1), and all energy distributions $\mathbf{\Psi}$ where the sum is one.

For a given code rate R, the minimum required SNR, $\gamma_b^\star$ (to reach a target BER, $P_b^\star$), together with the optimal puncturing, $\mathbf{\Delta}^\star$, and the optimal energy distribution, $\mathbf{\Psi}^\star$, can be found as a solution to the following

[7]Lowest possible convergence threshold.

minimization problem.

$$[\mathbf{\Delta}^\star, \mathbf{\Psi}^\star, \gamma_b^\star] = \arg \min_{[\mathbf{\Delta}, \mathbf{\Psi}, \gamma_b]} g(\mathbf{\Delta}, \mathbf{\Psi}, \gamma_b) \tag{38}$$

subject to

$$\mathbf{\Delta} \in [0,1]^{N+1}, \quad R = \left(\sum_{n=0}^{N} \frac{\delta_n}{R_n} \right)^{-1}, \tag{39}$$

$$\mathbf{\Psi} \in (0,1)^{N+1}, \quad \sum_{n=0}^{N} \psi_n = 1, \tag{40}$$

$$\gamma_b \geq \widetilde{\gamma}_b, \quad \text{or equivalent } R \leq C(R, \mathbf{\Delta}, \mathbf{\Psi}, \gamma_b). \tag{41}$$

The objective function in (38) is defined as

$$g(\mathbf{\Delta}, \mathbf{\Psi}, \gamma_b) = \begin{cases} \gamma_b & f(\mathbf{\Delta}, \mathbf{\Psi}, \gamma_b) \geq J\big(2Q^{-1}(P_b^\star)\big) \\ \infty & \text{otherwise,} \end{cases} \tag{42}$$

where $f(\mathbf{\Delta}, \mathbf{\Psi}, \gamma_b)$ is defined in (33). The code rate constraint in (39) is the code rate for a punctured MPCC, (1). The upper bound for R in (41) is taken from the maximum rate stated in (22). The threshold for $I_D^\star = f(\mathbf{\Delta}, \mathbf{\Psi}, \gamma_b)$ in (42) is taken from (30), which corresponds to a target BER $P_b^\star$ if the decision statistics were Gaussian [7, 11].

This constrained optimization problem is challenging. Since the arguments, $\mathbf{\Delta}$, $\mathbf{\Psi}$, and γ_b of the objective function are continuous within their corresponding interval of support, an exhaustive search is not possible. However, a grid search over δ_n and ψ_n for $n = 0, 1, \ldots, N$ can be performed. If δ_1 is quantized into N_δ uniformly spaced levels between 0 and 1, the resolution is $\epsilon_\delta = 1/(N_\delta - 1)$, or equivalent

$$N_\delta \triangleq \frac{1}{\epsilon_\delta} + 1. \tag{43}$$

If the same quantization is made for all δ_n, $n = 1, 2, \ldots, N$, there are N_δ^N combinations where the last N elements in $\mathbf{\Delta}$ are different. Define $\mathbf{\Delta}_k \triangleq \left[\delta_0^k, \delta_1^k, \delta_2^k, \ldots, \delta_N^k\right]$, for $k = 1, 2, \ldots, N_\delta^N$, to be all these combinations. Note that the first element in $\mathbf{\Delta}_k$, δ_0^k, is so far not specified. For a specific code rate $\mathbf{R}$, the first element in $\mathbf{\Delta}_k$ can be caluculated as

$$\delta_0^k = \frac{1}{R} - \sum_{i=1}^{N} \frac{\delta_i^k}{R_i}, \tag{44}$$

using (1) and the fact that $R_0 = 1$ (the rate of the systematic bits). Note that the calculated value in (44) may not be valid for this code rate, due to the constraint in (39).

The energy distribution can, in a similar way, also be uniformly quantized into N_ψ levels. Define $\mathbf{\Psi}_j \triangleq \left[\psi_0^j, \psi_1^j, \psi_2^j, \ldots, \psi_N^j\right]$, for $j = 1, 2, \ldots, N_\psi^N$, to be all the combinations of different $\mathbf{\Psi}$, where ψ_0 is calculated as

$$\psi_0^k = 1 - \sum_{i=1}^{N} \psi_i^k. \tag{45}$$

Let ϵ_γ be a small constant and let N_δ and N_ψ be the number of levels per δ_n^k and ψ_n^j, respectively. Construct all the $\mathbf{\Delta}_k$ and $\mathbf{\Psi}_j$ as described above, using (44) and (45). The search for optimal puncturing ratios and energy

distribution to reach a specified target BER, $P_b^\star$, for a specific set of N CCs concatenated in parallel with an overall code rate of R can now be performed by the following procedure:

Algorithm 1 (Optimal puncturing and energy distribution):

1) Set $j=-1$, $k=0$, and initialize γ_b with $\gamma_b = Q^{-1}(P_b^\star)^2/2$, corresponding to uncoded BPSK.
2) Increment j with one. If $j > N_\psi^N$, go to Step 7. If $0 < \psi_0^j < 1$, go to Step 3. Otherwise, go to Step 2.
3) Increment k with one. If $k > N_\delta^N$, set $k=0$ and go to Step 2. If $0 \le \delta_0^k \le 1$, go to Step 4. Otherwise, go to Step 3.
4) If $R \le C(R, \mathbf{\Delta}_k, \mathbf{\Psi}_j, \gamma_b)$, go to Step 5. Otherwise, go to Step 3.
5) Update the extrinsic MIs in (32) for the N decoders in an arbitrary order using $\mathbf{\Delta}_k$, $\mathbf{\Psi}_j$, and γ_b until $I_{D(x)}$ has converged to $I_D^\star = f(\mathbf{\Delta}_k, \mathbf{\Psi}_j, \gamma_b)$.
6) If $I_D^\star \ge J\big(2Q^{-1}(P_b^\star)\big)$, save $\mathbf{\Delta}^\star = \mathbf{\Delta}_k$, $\mathbf{\Psi}^\star = \mathbf{\Psi}_j$, $\gamma_b^\star = \gamma_b$, set $\gamma_b = \gamma_b - \epsilon_\gamma$ and go to Step 4. Otherwise, go to Step 3.
7) Output $\mathbf{\Delta}^\star$, $\mathbf{\Psi}^\star$, and $\gamma_b^\star$ from Step 6 as the optimal puncturing ratios, energy distribution, and the corresponding convergence threshold.

The small value ϵ_γ, used in Step 6, is chosen arbitrarily to give a certain resolution of the convergence threshold, $\gamma_b^\star$. In Step 1, $j=-1$ so that the search always starts with the uniform energy distribution $\mathbf{\Psi}_0$ from (23). The algorithm can be improved by sorting the candidate puncturing ratios and energy distributions, $\mathbf{\Delta}_k$ and $\mathbf{\Psi}_j$, according to the maximum rate $C(R, \mathbf{\Delta}_k, \mathbf{\Psi}_j, \gamma_b)$ in (22). The algorithm now terminates in Step 4 instead of Step 2, since once the rate is above $C(R, \mathbf{\Delta}_k, \mathbf{\Psi}_j, \gamma_b)$ it will stay above for all remaining candidate puncturing ratios and energy distributions, hence the search can be terminated earlier.

In summary, Algorithm 1 first initializes γ_b with a high value corresponding to uncoded BPSK. The algorithm then evaluates $f(\mathbf{\Delta}, \mathbf{\Psi}, \gamma_b)$ for all values of $\mathbf{\Delta}$ and $\mathbf{\Psi}$ that satisfy the constraints in (39)–(41). Whenever $f(\mathbf{\Delta}, \mathbf{\Psi}, \gamma_b)$ is above the threshold, γ_b can be decreased with an arbitrary small step size until $f(\mathbf{\Delta}, \mathbf{\Psi}, \gamma_b)$ is below the threshold. The search is continued until there is no $\mathbf{\Delta}$ and $\mathbf{\Psi}$ that satisfy the constraints at the same time as $f(\mathbf{\Delta}, \mathbf{\Psi}, \gamma_b)$ is above the threshold.

E. Achievable SNR-rate Region

Using Algorithm 1 with $N_\psi = 0$ finds the optimal puncturing ratios with a uniform energy distribution, since the algorithm will only use $\mathbf{\Psi}_0$ and terminate when $j=1$. For example, the optimal puncturing ratios for PCC($1 + 4/7 + 7/5$) with uniform energy distribution and $R = 1/2$ can be found using Algorithm 1 with $N_\psi = 0$. The results show that $\mathbf{\Delta}^\star = [0, 1, 1]$ are in fact the optimal puncturing ratios for this code at $R = 1/2$. Hence, the convergence threshold for PCC($1 + 4/7 + 7/5$) at $R = 1/2$ cannot be lowered by changing the puncturing ratios. It may, however, be possible to find other combinations of two component codes with other puncturing ratios and energy distribution, providing an even lower convergence threshold at $R = 1/2$.

With N rate-one CCs, the code rate can be chosen arbitrarily within a range $1/(N+1) \le R \le 1$ by simply changing the puncturing ratios $\mathbf{\Delta}$. The lower limit is when no bits are punctured, i.e., $\delta_n = 1$ for all $n = 0, 1, \ldots, N$, and the upper limit is the uncoded case when $\delta_0 = 1$ and $d_n = 0$ for $n = 1, 2, \ldots, N$. There are other combinations of puncturing ratios that also give $R = 1$, but their performance cannot be better than

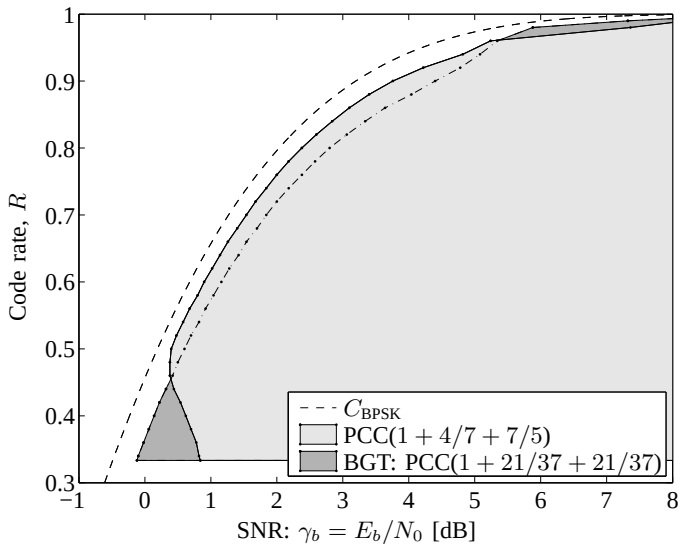

Fig. 4. SNR-rate regions for $P_b^\star = 10^{-5}$ with uniform energy distribution.

the uncoded case. Applying (38) or using Algorithm 1 with $N_\psi = 0$ (uniform energy distribution) the optimal puncturing ratios and the corresponding convergence threshold for PCC$(1 + 4/7 + 7/5)$ with $1/3 \leq R \leq 1$ is found. The chosen threshold for $f(\mathbf{\Delta}, \mathbf{\Psi}, \gamma_b)$ in (42) is set to 0.9999 which corresponds to a BER $P_b^\star \approx 10^{-5}$ if the decision statistics are Gaussian (30). The resolution is fixed to $\epsilon_\delta = \epsilon_\gamma = 0.02$. The result is an SNR-rate region for PCC$(1 + 4/7 + 7/5)$, shown as the light gray shaded region in Figure 4. All points that are inside the light gray shaded region are according to Algorithm 1 achievable. As a reference, the maximum rate for BPSK (24) is also included as the dashed curve.

Also, as a reference, the SNR-rate region for the original turbo code: PCC$(1 + 21/37 + 21/37)$, here called the BGT code referring to its inventors [1], is included in Figure 4 as the dark gray shaded region with dash-dotted boundaries. For $R \leq 0.45$, the BGT code has a lower $\gamma_b^\star$ than PCC$(1 + 4/7 + 7/5)$, but for $0.45 < R \leq 0.96$ PCC$(1 + 4/7 + 7/5)$ has a lower $\gamma_b^\star$. The optimal puncturing for the BGT code at $R = 1/2$ is $\mathbf{\Delta}^\star = [0.74, 0.56, 0.70]$, which gives $\gamma_b^\star = 0.60$ dB. The puncturing used in [1], $\mathbf{\Delta} = [1.0, 0.5, 0.5]$ gives a threshold of $\gamma_b^\star = 0.66$ dB. Note that in [1], a fixed puncturing pattern, puncturing every other parity bit from both encoders is used. However, the convergence threshold predicted here assuming random puncturing is close to the convergence threshold found by simulations in [1]. According to the achievable SNR-rate regions in Figure 4, PCC$(1+4/7+7/5)$ with $R = 1/2$ and optimal puncturing has better performance than the BGT code, which is also confirmed in the next sub-section. Figure 4 also shows that $\gamma_b^\star = 0.40$ dB for PCC$(1+4/7+7/5)$ at $R = 1/2$ is lower than $\gamma_b^\star = 0.84$ for $R = 1/3$. This means that PCC$(1+4/7+7/5)$ has better performance with $\mathbf{\Delta} = [0, 1, 1]$ ($R = 1/2$) than with $\mathbf{\Delta} = [1, 1, 1]$ ($R = 1/3$) [11].

The optimal puncturing ratios $\mathbf{\Delta}^\star$ for PCC$(1 + 4/7 + 7/5)$ for $1/3 \leq R \leq 1$ are shown in Figure 5. The conclusion from Figure 5 is that no systematic bits should be transmitted for $R \geq 1/2$. Since δ_n and γ_b are quantized, several different $\mathbf{\Delta}^\star$ can be found for each R. This explains the discontinuous behavior in Figure 5 for high R. The optimal values for δ_0, δ_1 and δ_2 are almost monotonically decreasing (disregarding the points

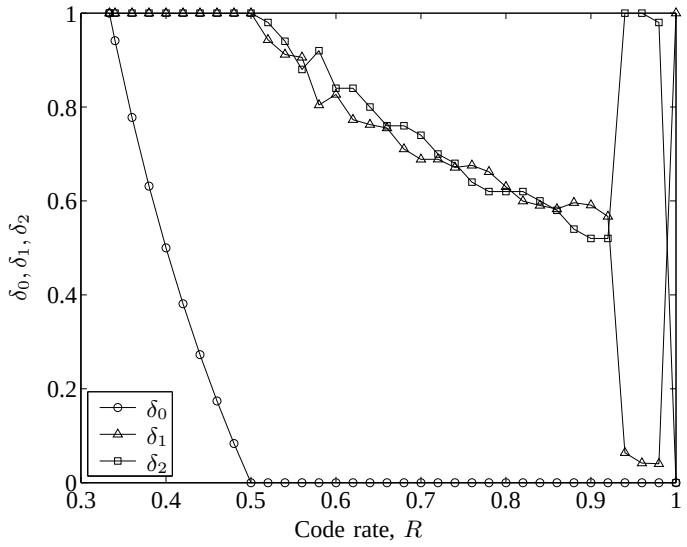

Fig. 5. $\boldsymbol{\Delta}^\star = [\delta_0, \delta_1, \delta_2]$ for PCC$(1 + 4/7 + 7/5)$ with uniform energy distribution and $P_b^\star = 10^{-5}$.

at high R). The PCC is therefore rate-compatible [28], i.e., the code rate can be changed from a higher R to a lower R just by transmitting more bits from, $\boldsymbol{y}_0$, $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ in Figure 1, according to the puncturing ratios in Figure 5.

F. Code Search with Uniform Energy Distribution

According to Table II, there are 21 ways of combining two of the rate-one CCs with $\nu \leq 2$ listed in Table III. In this sub-section Algorithm 1 is used to find and compare the SNR-rate region for all these 21 combinations using the uniform energy distribution.

The results show that no single combination has the lowest $\gamma_b^\star$ for all $R \geq 1/3$. However, PCC$(1+5/7+7/6)$ has the lowest $\gamma_b^\star$ for all $R \geq 1/2$ and the achievable SNR-rate region is shown in Figure 6 as the light gray shaded region. PCC$(1 + 5/7 + 5/7)$ has the lowest $\gamma_b^\star \approx 0$ for $R = 1/3$ (not shown in the figure), but a higher $\gamma_b^\star$ than PCC$(1 + 5/7 + 7/6)$ when $R \geq 0.45$.

Figure 7 shows the EXIT chart for PCC$(1 + 5/7 + 7/6)$ with optimal puncturing $\boldsymbol{\Delta}^\star = [0.04, 1.0, 0.96]$ ($R = 1/2$) at $\gamma_b = 0.3$ dB. The area between the two EXIT functions is very small since 0.3 dB is close to the minimum required SNR of 0.19 dB (25). The EXIT chart in Figure 7 indicates that the performance of this PCC is better than both the best non-systematic PCC$(4/7 + 7/5)$ (found in the previous sub-section) and the BGT code with $R = 1/2$.

The BGT code with $R = 1/2$ in [1] uses $\boldsymbol{\Delta} = [1.0, 0.5, 0.5]$ and blocks of $L = 2^{16} = 65\,536$ information bits. Each CC in the BGT code has memory $\nu = 4$ (16 states) while PCC$(1 + 5/7 + 7/6)$ only has memory $\nu = 2$ (4 states) for each CC. The computational decoding complexity per activation is roughly proportional to the number of states, i.e., approximately 4 times higher for the BGT code than for the PCC$(1 + 5/7 + 7/6)$.

Figure 8 shows the performance of the BGT code after 40 activations, when the performance has converged. The performance of PCC$(1 + 5/7 + 7/6)$ after 40, 60 and 80 activations, using optimal puncturing ratios

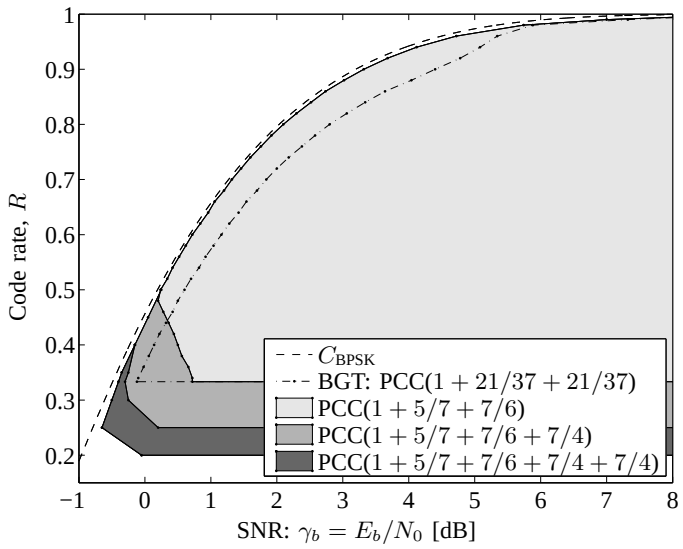

Fig. 6. SNR-rate regions for $P_b^\star = 10^{-5}$ with uniform energy distribution.

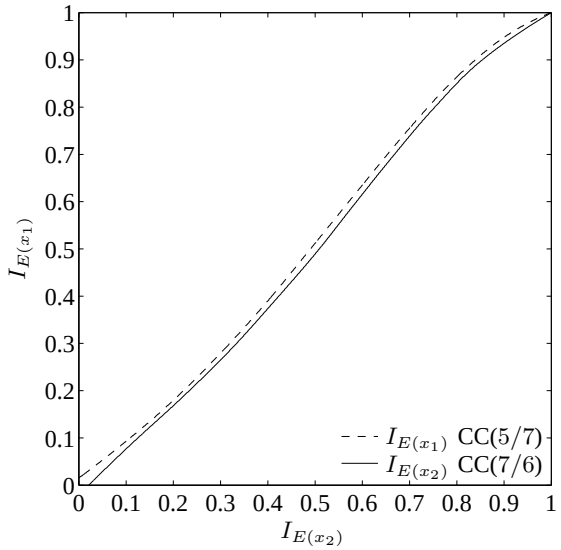

Fig. 7. EXIT chart for PCC$(1 + 5/7 + 7/6)$ with $\mathbf{\Delta}^\star = [0.04, 1.0, 0.96]$ and uniform energy distribution, at SNR: $\gamma_b = 0.3$ dB.

$\mathbf{\Delta}^\star = [0.04, 1.0, 0.96]$ for $R = 1/2$, is also shown in Figure 8. Due to the difference in decoding complexity, 40, 60 and 80 activations corresponds to using approximately 25%, 38% and 50% of the decoding complexity used by the decoder for the BGT code after 40 activations. It is observed that the performance of PCC$(1+5/7+7/6)$ is better than the performance of the BGT code, even though only around 25% of the decoding complexity is used. After 80 activations, the performance of PCC$(1 + 5/7 + 7/6)$ has converged. The difference compared to the BGT code is then 0.15 dB at $P_b = 10^{-5}$, even though PCC$(1 + 5/7 + 7/6)$ is then only using around $1/2$ of the complexity that the BGT code is using. The performance of PCC$(4/7 + 7/5)$ is also included in Figure 8 as the dashed-dotted curves. The complexity of the decoder for PCC$(4/7 + 7/5)$ is slightly lower than the complexity for PCC$(1 + 5/7 + 7/6)$, since no systematic bits or puncturer/depuncturer are needed in

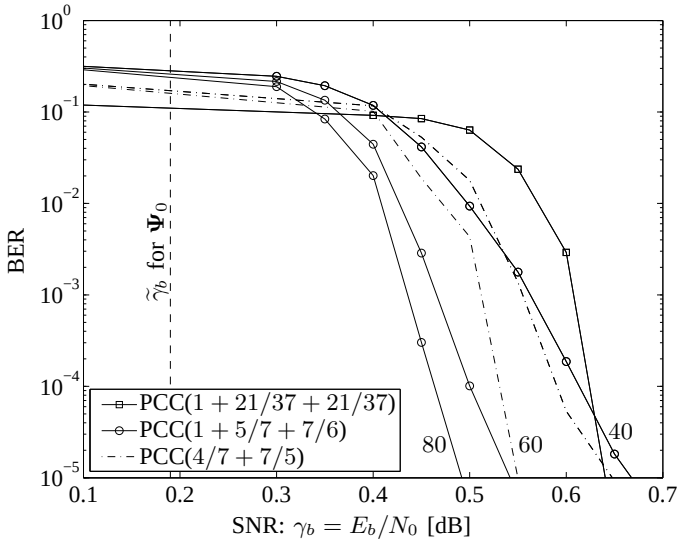

Fig. 8. Performance of BGT with $\mathbf{\Delta} = [1.0, 0.5, 0.5]$ after 40 activations, PCC($1 + 5/7 + 7/6$) with $\mathbf{\Delta}^\star = [0.04, 1.0, 0.96]$ after 40, 60 and 80 activations, and PCC($4/7 + 7/5$) after 40 and 60 activations. All three PCCs have $R = 1/2$ and $L = 2^{16} = 65\,536$.

PCC($4/7 + 7/5$) (refer to Figure 3). The performance of PCC($4/7 + 7/5$) and the BGT code, both after 40 activations, are similar. After 60 activations, PCC($4/7 + 7/5$) has similar performance as PCC($1 + 5/7 + 7/6$). The performance of PCC($4/7 + 7/5$) has converged after 60 activations.

Algorithm 1 can also be used to find the SNR-rate regions of MPCCs with $N > 2$ CCs. In place of performing a full search over all 98 combinations of $N = 3$ CCs, we can examine all possible extensions of PCC($1 + 5/7 + 7/6$) using the eight candidates in Table III. The resulting MPCC has an SNR-rate region that includes the previous region since the new MPCC comprises PCC($1 + 5/7 + 7/6$) by choosing $\delta_3 = 0$. The search is therefore only made for $1/4 \leq R \leq 1/2$ since PCC($1 + 5/7 + 7/6$) is already so close to the capacity for $R \geq 1/2$. The lowest $\gamma_b^\star$ for $R \geq 1/3$ among the eight combinations is found when $CC(7/4)$ is added to PCC($1 + 5/7 + 7/6$). The SNR-rate region for PCC($1 + 5/7 + 7/6 + 7/4$) is shown in Figure 6 as the second lightest shaded region together with the region for PCC($1 + 5/7 + 7/6$). As a reference, the SNR-rate region for the BGT code is included in Figure 6 as the dashed-dotted curve. It is clear that the BGT code has a smaller SNR-rate region than PCC($1 + 5/7 + 7/6$) for all $R > 0.45$ and a smaller SNR-rate region than PCC($1 + 5/7 + 7/6 + 7/4$) for all $R > 1/3$.

The search can be continued by adding another CC to PCC($1 + 5/7 + 7/6 + 7/4$). The best choice for all $1/4 \leq R \leq 1/2$, among the eight CCs from Table III, is to add another $CC(7/4)$ to PCC($1 + 5/7 + 7/6 + 7/4$). This MPCC has an SNR-rate region shown in Figure 6 as the darkest shaded region together with the two lighter regions. As for PCC($1 + 5/7 + 7/6 + 7/4$), PCC($1 + 5/7 + 7/6 + 7/4 + 7/4$) includes the SNR-rate region of the previous codes since PCC($1 + 5/7 + 7/6 + 7/4 + 7/4$) comprises PCC($1 + 5/7 + 7/6 + 7/4$) when $\delta_4 = 0$ and PCC($1 + 5/7 + 7/6$) when $\delta_3 = \delta_4 = 0$.

The SNR-rate region in Figure 6 for $N = 2$ is found using Algorithm 1 with $\epsilon_\delta = \epsilon_\gamma = 0.02$. For $N = 3$ and $N = 4$, a more coarse resolution is used, $\epsilon_\delta = \epsilon_\gamma = 0.05$, due to the larger number of N_δ^N. In order to find the best combination of $N = 4$ CCs of memory up to $\nu = 2$, an exhaustive search over all 301 combinations

TABLE IV

OPTIMAL PUNCTURING AND ENERGY DISTRIBUTION FOR PCC$(1+5/7+7/6+7/4)$ TO REACH $P_b^\star = 10^{-5}$.

R	$\boldsymbol{\Delta}^\star$	$\boldsymbol{\Psi}^\star$	$\boldsymbol{\Psi}_0$	$\gamma_b^\star$ for $\boldsymbol{\Psi}^\star$	$\gamma_b^\star$ for $\boldsymbol{\Psi}_0$	$\widetilde{\gamma}_b$ for $\boldsymbol{\Psi}^\star$	$\widetilde{\gamma}_b$ for $\boldsymbol{\Psi}_0$
0.25	$[1.00, 1.00, 1.00, 1.00]$	$[0.08, 0.34, 0.24, 0.34]$	$[0.25, 0.25, 0.25, 0.25]$	-0.36 dB	0.16 dB	-0.65 dB	-0.79 dB
0.30	$[0.35, 1.00, 1.00, 0.98]$	$[0.20, 0.30, 0.20, 0.30]$	$[0.25, 0.25, 0.25, 0.25]$	-0.30 dB	-0.22 dB	-0.58 dB	-0.61 dB

of four CCs from Table III should be performed. PCC$(1+5/7+7/6+7/4+7/4)$ shown in Figure 6 is just a good MPCC with $N=4$, but may not be the best combination.

Algorithm 1 can also be used to find the optimal puncturing in a multiple serial concatenated code (MSCC). The difference is then the expression for the code rate and how the EXIT functions are mutually dependent [11, 28]. Algorithm 1 is also easily modified to find the optimal puncturing ratios for MCCs with a fixed number of activations. This can be done by limiting the number of activations in Step 5. Instead of limiting the number of activations, the computational decoding complexity can be limited to find the optimal scheduling [10–12], in Step 5 of Algorithm 1.

G. Optimal Energy Distribution

All codes in Figure 6 are close to the maximum rate using BPSK for all $R \geq 1/N$. For code rates below $1/N$, the convergence threshold, $\gamma_b^\star$, increases as R tends to its lower bound $1/(N+1)$. By optimizing the energy distribution using (38) (or Algorithm 1) the convergence threshold can be reduced, which in turn reduces the gap between $\gamma_b^\star$ and $\widetilde{\gamma}_b$.

Consider PCC$(1+5/7+7/6+7/4)$ for $R=1/4$ ($\boldsymbol{\Delta} = \boldsymbol{\Delta}^\star = \mathbf{1}$) in Figure 6. With a uniform energy distribution $\gamma_b^\star = 0.16$ dB. Optimizing the energy distribution using (38) gives $\boldsymbol{\Psi}^\star = [0.08, 0.34, 0.24, 0.34]$ and $\gamma_b^\star = -0.36$ dB. This is a gain of 0.52 dB compared to uniform energy distribution. The optimal puncturing and energy distribution for PCC$(1+5/7+7/6+7/4)$ at $R=0.25$ and $R=0.30$ are listed in Table IV. The same table also lists $\gamma_b^\star$ for $\boldsymbol{\Psi}_0$ together with $\widetilde{\gamma}_b$ for $\boldsymbol{\Psi}_0$ and $\boldsymbol{\Psi}^\star$. The predicted gain for $0.25 \leq R \leq 0.30$ is in the range of 0.08–0.52 dB using $\boldsymbol{\Psi}^\star$ instead of $\boldsymbol{\Psi}_0$. At the same time, the minimum required SNR, $\widetilde{\gamma}_b$, increases with 0.03–0.14 dB when $\boldsymbol{\Psi}^\star$ is used instead of $\boldsymbol{\Psi}_0$.

The EXIT chart projection [9–12] for PCC$(1+5/7+7/6+7/4)$ at $R=1/4$ is shown in Figure 9. Here it is clear that changing the energy distribution affects the EXIT chart. Figure 9 shows that using $\boldsymbol{\Psi}^\star$ the tunnel is open at $\gamma_b = -0.36$ dB, while it is closed using $\boldsymbol{\Psi}_0$. Figure 10 shows the simulated performance of PCC$(1+5/7+7/6+7/4)$ in BER using both uniform and optimal energy distribution at $R=1/4$. The gain in performance, just by changing the energy distribution is around 0.5 dB and it corresponds well with the predicted gain listed in Table IV. In the same figure the minimum required SNRs from (22) for the two schemes are reported. It is clear that even though the minimum required SNR increases by 0.14 dB when non-uniform energy distribution is used, the gain in performance is around 0.5 dB.

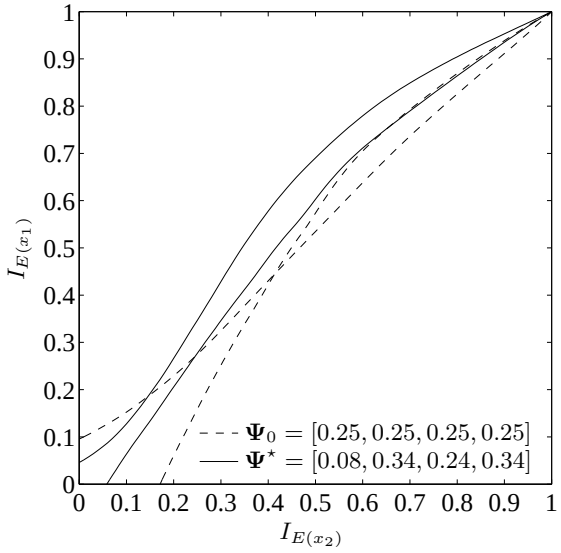

Fig. 9. EXIT chart projection of PCC($1 + 5/7 + 7/6 + 7/4$) with $R = 1/4$ ($\boldsymbol{\Delta}^\star = [1, 1, 1, 1]$) at SNR: $\gamma_b = -0.36$ dB.

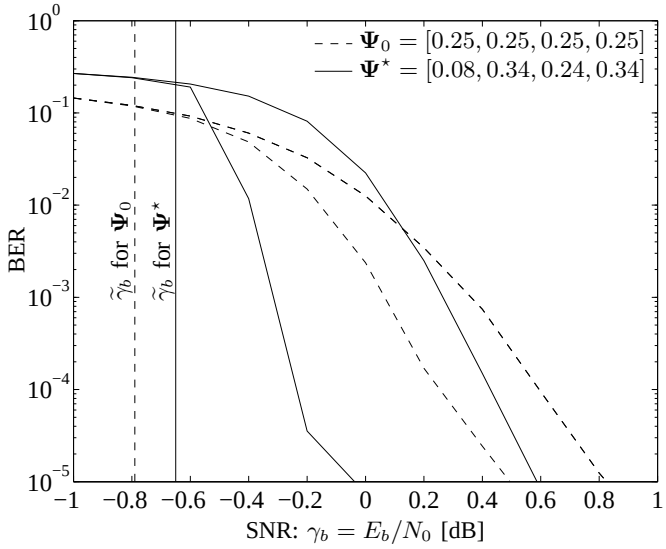

Fig. 10. Performance of PCC($1+5/7+7/6+7/4$) with $R = 1/4$ ($\boldsymbol{\Delta}^\star = [1, 1, 1, 1]$) and $L = 2^{16} = 65\,536$ after 20 and 80 activations.

V. Conclusions

We considered the problem of finding the optimal puncturing ratios together with the optimal energy distribution for MPCCs, using EXIT functions to minimize the average SNR convergence threshold. For such codes, the individual puncturing ratios for the constituent codes can be chosen freely within a desired code rate. The energy distribution can also be chosen arbitrary without affecting the code rate, while preserving the average bit energy. We have shown how to jointly optimize the energy distribution and the puncturing ratios by utilizing the EXIT functions of the constituents. The result is an SNR-rate region for any combination of an arbitrary number of constituent codes. For MCCs it is important to investigate all combinations of constituent codes that provide different performance. A code search is therefore made to identify all rate-one CCs with unique EXIT

TABLE V

FACTORIZATION IN GF(2).

ν	binary	octal	factors
1	10	2	–
1	11	3	–
2	100	4	$2 \cdot 2$
2	101	5	$3 \cdot 3$
2	110	6	$2 \cdot 3$
2	111	7	–
3	1000	10	$2 \cdot 2 \cdot 2$
3	1001	11	$3 \cdot 7$
3	1010	12	$2 \cdot 3 \cdot 3$
3	1011	13	–
3	1100	14	$2 \cdot 2 \cdot 3$
3	1101	15	–
3	1110	16	$2 \cdot 7$
3	1111	17	$3 \cdot 3 \cdot 3$
4	10000	20	$2 \cdot 2 \cdot 2 \cdot 2$
4	10001	21	$3 \cdot 3 \cdot 3 \cdot 3$
4	10010	22	$2 \cdot 3 \cdot 7$
4	10011	23	–
4	10100	24	$2 \cdot 2 \cdot 3 \cdot 3$
4	10101	25	$7 \cdot 7$
4	10110	26	$2 \cdot 13$
4	10111	27	$3 \cdot 15$
4	11000	30	$2 \cdot 2 \cdot 2 \cdot 3$
4	11001	31	–
4	11010	32	$2 \cdot 15$
4	11011	33	$3 \cdot 3 \cdot 7$
4	11100	34	$2 \cdot 2 \cdot 7$
4	11101	35	$3 \cdot 13$
4	11110	36	$2 \cdot 3 \cdot 3 \cdot 3$
4	11111	37	–

functions. The result is a list of 98 classes of CCs with memory four or less. To demonstrate the approach, an example is included showing a performance gain can be obtained using the optimal energy distribution, even though the minimum required SNR, given by information theory, increases. The ideas presented here are easily extended to other constituent codes and also to MSCCs.

APPENDIX

In this appendix all CCs with $R = 1$ and $\nu \leq 4$ according to their set of EXIT functions $\{T_x, T_y\}$ are classified. Table V shows all generator polynomials with $\nu \leq 4$, given in both binary and octal notation. It also shows the factorization of the numbers under Galois field with 2 elements, denoted by GF(2). Let 1001_2 denote a binary number and 11_8 its equivalent number in octal notation, according to Table V. For example,

$1001_2 \equiv 11_8 = 3_8 \cdot 7_8 \equiv (1+\mathrm{D})(1+\mathrm{D}+\mathrm{D}^2) = 1+\mathrm{D}+\mathrm{D}+\mathrm{D}^2+\mathrm{D}^2+\mathrm{D}^3 = 1+\mathrm{D}^3 \equiv 11_8$, since $\mathrm{D}^k+\mathrm{D}^k = 0$ for any $k = 0, 1, 2, \ldots$, under GF(2). A factor 2_8 adds a zero to the right side of the binary representation, e.g., $2_8 \cdot 3_8 \cdot 7_8 = 2_8 \cdot 11_8 = 10010_2 \equiv 22_8$. Using GF(2) arithmetic, it can be concluded that there are eight prime factors for $\nu \leq 4$, i.e., $2, 3, 7, 13, 15, 23, 31$ and 37 (given in octal notation).

Note also that some of the binary numbers in Table V are reversible, i.e., they turn into a different number if they are reversed, e.g., $1101_2 \equiv 15_8$ turns into $1011_2 \equiv 13_8$. Other reversible numbers are 31_8 (turns into 23_8), and 35_8 (turns into 27_8). Further, note that if 32_8 is reversed (disregarding the ending zero) it turns into 26_8, since $32_8 = 2_8 \cdot 15_8$ turns into $2_8 \cdot 13_8 = 26_8$. From now on, only the octal notation of the generator polynomials and their factorization are used. Therefore, the subindex for all numbers are omitted. Collect the four reversible numbers in a vector $\boldsymbol{\alpha}$ and their corresponding reversed numbers in a vector $\boldsymbol{\beta}$,

$$\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \alpha_3, \alpha_4] \triangleq [15, 31, 35, 32]\,, \tag{46}$$

$$\boldsymbol{\beta} = [\beta_1, \beta_2, \beta_3, \beta_4] \triangleq [13, 23, 27, 26]\,. \tag{47}$$

This feature is used later in this appendix.

Table I shows the total number of rate-one CCs of all types (F, B and C) with memory $\nu \leq 4$. Each of the 310 CCs can be given a unique index k starting with CC$(3/2)$ ($k = 1$), and ending with CC$(36/37)$ ($k = 310$). This means that a CC with index $k < n$ has a lower or equal ν than the CC with index n.

To identify all unique sets of EXIT functions, $\{T_x, T_y\}$, among these 310 different CCs with $\nu \leq 4$, the volume of the difference between their EXIT functions can be investigated. Define the following metric

$$V(k,n) \triangleq \tag{48}$$
$$\int_0^1\int_0^1 |T_{x_k}(\phi,\xi) - T_{x_n}(\phi,\xi)| + |T_{y_k}(\phi,\xi) - T_{y_n}(\phi,\xi)| d\phi\, d\xi,$$

where T_{x_k} and T_{y_k} are the EXIT functions for the CC with index $k = 1, 2, \ldots, 310$ and T_{x_n} and T_{y_n} are the EXIT functions for the CC with index $n = 1, 2, \ldots, 310$. Figure 2 shows that the EXIT functions are unstable in the points $T_x(1,0)$, $T_x(0,1)$, $T_y(1,0)$, and $T_y(0,1)$. These points are therefore excluded when $V(k,n)$ in (48) is evaluated.

The minimum volume of the difference for the CC with index k to any of the other CCs with index $n < k$ is then

$$V_{\min}(k) \triangleq \min_{n<k}\{V(k,n)\}\,. \tag{49}$$

After calculating all $V_{\min}(k)$ for $k = 1, 2, \ldots, 310$ they are sorted in decreasing order and plotted in Figure 11 (note that $1000V_{\min}(k)$ is plotted). Only the minimum difference in volume for the CCs with index $k = 92, 93, \ldots, 310$ is shown. For $k = 1, 2, \ldots, 91$, $V_{\min}(k) > 0.004$. Choosing a threshold $V_{\text{th}} = 0.0025$ gives 98 unique sets of EXIT functions $\{T_x, T_y\}$. The CC with lowest index among all CCs within each of the 98 unique classes is chosen as the representative for that class. This assures that one of the CCs with the lowest decoding complexity in that class is the representative. Tables VI–IX lists all classes of CCs with unique EXIT functions. The first column, ν, is the number of memory elements for the class representative, the second column, CC, is

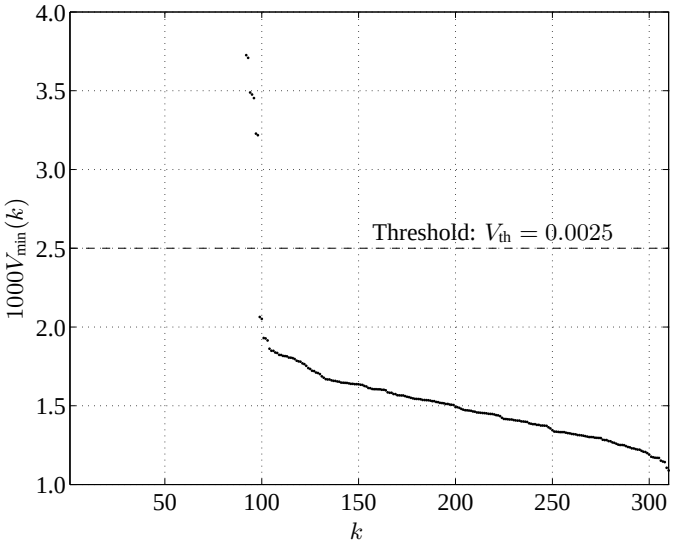

Fig. 11. The minimum volume of the difference for the 310 CCs.

TABLE VI

ALL 8 CLASSES OF $R = 1$ CCS OF TYPE F WITH $\nu \leq 4$.

ν	CC	Members
1	$3/2$	$5/4$, $6/4$, $5/6$, $11/10$, $12/10$, $14/10$, $17/12$, $12/14$, $17/14$, $11/16$, $21/20$, $22/20$, $24/20$, $30/20$, $33/22$, $21/24$, $36/24$, $35/26$, $24/30$, $33/30$, $36/30$, $27/32$, $22/34$, $33/34$, $21/36$
2	$7/4$	$16/10$, $11/14$, $25/20$, $34/20$, $33/24$, $22/30$, $25/34$
3	$13/10$	$15/10$, $26/20$, $32/20$, $27/30$, $35/30$
3	$17/10$	$36/20$, $21/30$
4	$23/20$	$31/20$
4	$27/20$	$35/20$
4	$33/20$	-
4	$37/20$	-

the class representative, and the third column contains all other members of that class. For example, CC($5/4$) belongs to the class represented by CC($3/2$) in Table VI. This is also denoted by $5/4 \equiv 3/2$ and basically means that the volume of the difference, according to (48), between these two CCs is less than 0.0025.

The following *ad hoc* rules can be applied to find the unique classes of rate-one CCs with $\nu \leq 4$ listed in Tables VI–IX, (some of them more obvious than others):

1) All k/k-factors can be removed, e.g., $17/12 = 2/2 \cdot 3/3 \cdot 3/2 \equiv 3/2$.
2) Doubles and quadruples can be reduced to one, e.g., $25/21 = 7/5 \cdot 7/5 \equiv 7/5$ and $21/20 = 3/2 \cdot 3/2 \cdot 3/2 \cdot 3/2 \equiv 3/2$. Note that this is not true for triples, e.g., $17/10 = 3/2 \cdot 3/2 \cdot 3/2$ is a class representative.
3) A CC of the form β_j/α_k from (46)–(47) can be replaced by its corresponding α_j/β_k, for $j = 1, 2, 3, 4$ and $k = 1, 2, 3, 4$, e.g., $13/15 \equiv 15/13$ and $23/35 \equiv 31/27$.
4) A CC of the form α_j/α_k can be replaced by its corresponding β_j/β_k, for $j \neq k$, e.g., $35/31 \equiv 27/23$.

TABLE VII

ALL 8 CLASSES OF $R = 1$ CCS OF TYPE B WITH $\nu \leq 4$.

ν	CC	Members
1	$2/3$	$4/5$, $6/5$, $4/6$, $10/11$, $16/11$, $10/12$, $14/12$, $10/14$, $12/17$, $14/17$, $20/21$, $24/21$, $36/21$, $20/22$, $34/22$, $20/24$, $30/24$, $32/27$, $20/30$, $22/33$, $30/33$, $34/33$, $26/35$, $24/36$, $30/36$
2	$4/7$	$14/11$, $10/16$, $30/22$, $20/25$, $34/25$, $24/33$, $20/34$
3	$10/13$	$10/15$, $20/26$, $30/27$, $20/32$, $30/35$
3	$10/17$	$30/21$, $20/36$
4	$20/23$	$20/31$
4	$20/27$	$20/35$
4	$20/33$	-
4	$20/37$	-

TABLE VIII

ALL 17 CLASSES OF $R = 1$ CCS OF TYPE C WITH $\nu \leq 3$.

ν	CC	Members
2	$7/5$	$16/12$, $11/17$, $25/21$, $33/21$, $34/24$, $25/33$, $22/36$
2	$7/6$	$11/12$, $16/14$, $25/22$, $22/24$, $25/24$, $34/30$, $33/36$
2	$5/7$	$17/11$, $12/16$, $36/22$, $21/25$, $33/25$, $21/33$, $24/34$
2	$6/7$	$12/11$, $14/16$, $24/22$, $22/25$, $24/25$, $36/33$, $30/34$
3	$13/11$	$15/11$, $26/22$, $32/22$, $27/33$, $35/33$
3	$13/12$	$15/12$, $26/24$, $32/24$, $27/36$, $35/36$
3	$11/13$	$11/15$, $22/26$, $33/27$, $22/32$, $33/35$
3	$12/13$	$12/15$, $24/26$, $36/27$, $24/32$, $36/35$
3	$14/13$	$14/15$, $30/26$, $24/27$, $30/32$, $24/35$
3	$15/13$	$13/15$, $32/26$, $35/27$, $26/32$, $27/35$
3	$16/13$	$16/15$, $34/26$, $22/27$, $34/32$, $22/35$
3	$17/13$	$17/15$, $36/26$, $21/27$, $36/32$, $21/35$
3	$13/14$	$15/14$, $27/24$, $35/24$, $26/30$, $32/30$
3	$13/16$	$15/16$, $27/22$, $35/22$, $26/34$, $32/34$
3	$17/16$	$21/22$, $36/34$
3	$13/17$	$15/17$, $27/21$, $35/21$, $26/36$, $32/36$
3	$16/17$	$22/21$, $34/36$

5) Any reversible number α_j in the denominator can be changed to its corresponding reversed number β_j unless the numerator is a reversed or reversible number, e.g., $25/35 \equiv 25/27$. Note that, e.g., $32/35$ is not equivalent to $32/27$ since 32 is a reversible number.
6) Any reversible number α_j in the numerator can be changed to its corresponding reversed number β_j unless the denominator is a reversed or a reversible number, e.g., $31/21 \equiv 23/21$. Note that, e.g., $31/27$ is not equivalent to $23/27$ since 27 is a reversed number.
7) Two special cases: $3/2 \cdot 7/4 \equiv 3/2$ and $2/3 \cdot 4/7 \equiv 2/3$, e.g., $11/10 = 3/2 \cdot 7/4 \equiv 3/2$.

TABLE IX

ALL 65 CLASSES OF $R = 1$ CCS OF TYPE C WITH $\nu = 4$.

ν	CC	Members
4	$23/21$	$31/21$
4	$26/21$	$32/21$
4	$34/21$	-
4	$37/21$	-
4	$23/22$	$31/22$
4	$37/22$	-
4	$21/23$	$21/31$
4	$22/23$	$22/31$
4	$24/23$	$24/31$
4	$25/23$	$25/31$
4	$26/23$	$32/31$
4	$27/23$	$35/31$
4	$30/23$	$30/31$
4	$31/23$	$23/31$
4	$32/23$	$26/31$
4	$33/23$	$33/31$
4	$34/23$	$34/31$
4	$35/23$	$27/31$
4	$36/23$	$36/31$
4	$37/23$	$37/31$
4	$23/24$	$31/24$
4	$37/24$	-
4	$23/25$	$31/25$
4	$26/25$	$32/25$
4	$27/25$	$35/25$
4	$30/25$	-
4	$36/25$	-
4	$37/25$	-
4	$21/26$	$21/32$
4	$23/26$	$31/32$
4	$25/26$	$25/32$
4	$27/26$	$35/32$
4	$31/26$	$23/32$

ν	CC	Members
4	$33/26$	$33/32$
4	$37/26$	$37/32$
4	$23/27$	$31/35$
4	$25/27$	$25/35$
4	$26/27$	$32/35$
4	$31/27$	$23/35$
4	$34/27$	$34/35$
4	$37/27$	$37/35$
4	$23/30$	$31/30$
4	$25/30$	-
4	$37/30$	-
4	$23/33$	$31/33$
4	$26/33$	$32/33$
4	$37/33$	-
4	$21/34$	-
4	$23/34$	$31/34$
4	$27/34$	$35/34$
4	$37/34$	-
4	$23/36$	$31/36$
4	$25/36$	-
4	$37/36$	-
4	$21/37$	-
4	$22/37$	-
4	$23/37$	$31/37$
4	$24/37$	-
4	$25/37$	-
4	$26/37$	$32/37$
4	$27/37$	$35/37$
4	$30/37$	-
4	$33/37$	-
4	$34/37$	-
4	$36/37$	-

Note that these rules are constructed only to give the same result as $V_{\text{th}} = 0.0025$. They are not theoretically justified. Note also that rule 1 can be applied together with one of the other rules, e.g., $10/12 = 2/2 \cdot 2/3 \cdot 2/3 \equiv 2/3$ (using rule 1 and 2), $32/34 = 2/2 \cdot 15/16 \equiv 13/16$ (using rule 1 and 6), and $33/30 = 3/3 \cdot 3/2 \cdot 7/4 \equiv 3/2$ (using rule 1 and 7).

Using the rules above (or the threshold $V_{\text{th}} = 0.0025$ in Figure 11) gives 8 classes of CCs of type F listed in Table VI, 8 classes of CCs of type B listed in Table VII, and 82 classes of CCs of type C listed in Tables VIII–IX. Note that each CC(k/n) of type F in Table VI has a corresponding CC(n/k) of type B in Table VI. If

CC(k/n) is a class representative in Tables VIII–IX, CC(n/k) is also a class representative, unless $k = \alpha_j$ from (46) for $j = 1, 2, 3, 4$. For example, CC($23/35$) is not a class representative in Table IX even though CC($35/23$) is, since $23/35 \equiv 31/27$ according to rule 3 above.

Each of the possible 310 CCs can be found once in Tables VI–IX, either as a class representative or as a member of a class. For example, instead of using CC($20/21$) (with $\nu = 4$, i.e., 16 states) its class representative CC($2/3$) (with $\nu = 1$, i.e., 2 states) from Table VII should be used. These CCs have similar EXIT functions and therefore also similar performance in the waterfall region if they are used in a concatenated scheme, even though CC($20/21$) has 8 times more states than CC($2/3$). However, it should be noted that these two CCs have different IOWEF [23], so their performance in the error floor region is different.

REFERENCES

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," in *Proc. IEEE Int. Conf. Commun. (ICC '93)*, vol. 2, Geneva, Switzerland, May 1993, pp. 1064–1070.

[2] D. Divsalar and F. Pollara, "Multiple turbo codes for deep-space communications," *TDA Progress Report 42-121*, Jet Propulsion Laboratory, Pasadena, CA, May 1995.

[3] S. Huettinger, S. ten Brink, and J. Huber, "Turbo-code representation of RA-codes and DRS-codes for reduced decoding complexity," in *Proc. Conf. Inform. Sciences and Syst. (CISS '01)*, The Johns Hopkins University, Mar. 2001.

[4] S. Huettinger and J. Huber, "Design of "multiple-turbo-codes" with transfer characteristics of component codes," in *Proc. Conf. Inform. Sciences and Syst. (CISS '02)*, Princeton University, Mar. 2002.

[5] S. ten Brink, "Rate one-half code for approaching the Shannon limit by 0.1 dB," *IEE Electron. Lett.*, vol. 36, pp. 1293–1294, July 2000.

[6] I. Land and P. Hoeher, "Partially systematic rate 1/2 turbo codes," in *Proc. Int. Symp. on Turbo Codes and Rel. Topics*, Brest, France, Sept. 2000, pp. 287–290.

[7] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Trans. Commun.*, vol. 49, pp. 1727–1737, Oct. 2001.

[8] ——, "Convergence of multi-dimensional iterative decoding schemes," in *Proc. Thirty-Fifth Asilomar Conference on Signals, Systems and Computers*, vol. 1, Pacific Grove, CA, Nov. 2001, pp. 270–274.

[9] F. Brännström, L. K. Rasmussen, and A. Grant, "Optimal scheduling for iterative decoding," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT '03)*, Yokohama, Japan, June/July 2003, p. 350.

[10] ——, "Optimal scheduling for multiple serially concatenated codes," in *Proc. Int. Symp. on Turbo Codes and Rel. Topics*, Brest, France, Sept. 2003, pp. 383–386.

[11] F. Brännström, "Convergence analysis and design of multiple concatenated codes," Ph.D. dissertation, Chalmers University of Technolgy, Göteborg, Sweden, Mar. 2004.

[12] F. Brännström, L. K. Rasmussen, and A. J. Grant, "Convergence analysis and optimal scheduling for multiple concatenated codes," *IEEE Trans. Inform. Theory*, vol. 51, no. 9, pp. 3354–3364, Sept. 2005.

[13] S. ten Brink, G. Kramer, and A. Ashikhmin, "Design of low–density parity–check codes for modulation and detection," *IEEE Trans. Commun.*, vol. 52, no. 4, pp. 670–678, Apr. 2004.

[14] A. Ashikhmin, G. Kramer, and S. ten Brink, "Code rate and the area under extrinsic information transfer curves," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT '02)*, Lausanne, Switzerland, June/July 2002, p. 115.

[15] F. Brännström, L. K. Rasmussen, and A. Grant, "Optimal puncturing for multiple parallel concatenated codes," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT '04)*, Chicago, IL, June/July 2004, p. 154.

[16] F. Brännström and L. K. Rasmussen, "Multiple parallel concatenated codes with optimal puncturing and energy distribution," in *Proc. IEEE Int. Conf. Commun.(ICC '05)*, Seoul, Korea, May 2005, pp. 622–626.

[17] ——, "Non-data-aided parameter estimation in an additive white Gaussian noise channel," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT '05)*, Adelaide, Australia, Sept. 2005, pp. 1446–1450.

[18] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "A soft-input soft-output APP module for iterative decoding of concatenated codes," *IEEE Commun. Lett.*, vol. 1, pp. 22–24, Jan. 1997.

[19] J. Han and O. Y. Takeshita, "On the decoding structure for multiple turbo codes," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT '01)*, Washington, DC, June 2001, p. 98.

[20] A. Ashikhmin, G. Kramer, and S. ten Brink, "Extrinsic information transfer functions, information functions, support weights, and duality," in *Proc. Int. Symp. on Turbo Codes and Rel. Topics*, Brest, France, Sept. 2003, pp. 223–226.

[21] ——, "Extrinsic information transfer functions: model and erasure channel properties," *IEEE Trans. Inform. Theory*, vol. 50, pp. 2657–2673, Nov. 2004.

[22] S. ten Brink, "Exploiting the chain rule of mutual information for the design of iterative decoding schemes," in *Proc. 39th Allerton Conference on Communications, Control and Computing*, Monticello, Urbana-Champaign, Oct. 2001.

[23] S. Benedetto and G. Montorsi, "Unveiling turbo codes: some results on parallel concatenated coding schemes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 409–428, Mar. 1996.

[24] S. Lin and D. J. Costello Jr., *Error Control Coding: Fundamentals and Applications*. Prentice-Hall, 1983.

[25] D. Divsalar, S. Dolinar, and F. Pollara, "Iterative turbo decoder analysis based on density evolution," *IEEE J. Selected Areas Commun.*, vol. 19, pp. 891–907, May 2001.

[26] L. Rade and B. Westergren, *Beta Mathematics Handbook*, 2nd ed. Studentlitteratur, 1993.

[27] S. ten Brink, "Code doping for triggering iterative decoding convergence," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT '01)*, Washington, DC, June 2001, p. 235.

[28] E. Uhlemann, "Adaptive concatenated coding for wireless real-time communications," Ph.D. dissertation, Chalmers Univ. of Techn., Göteborg, Sweden, Sept. 2004.