Puncturing Strategies for Incremental Redundancy Schemes Using Rate Compatible Systematic Serially Concatenated Codes

Elisabeth Uhlemann[1,2], Lars K. Rasmussen[3] and Fredrik Brännström[4]

[1]Centre for Research on Embedded Systems, Halmstad University, Halmstad, Sweden, bettan@ide.hh.se
[2]Volvo Technology Corporation, Transport and Telematics Services, Dept. 6600, M1:6, Göteborg, Sweden
[3]Inst. for Telecommunications Research, University of South Australia, Australia, lars.rasmussen@unisa.edu.au
[4]Dept. of Signals and Systems, Chalmers University of Technology, Göteborg, Sweden, fredrikb@chalmers.se

Abstract

A rate compatible puncturing strategy intended for incremental redundancy retransmission schemes is derived. In particular, systematic serially concatenated codes with two or more component codes are considered, although the method is applicable to any type of concatenated code. Extrinsic information transfer charts are used to select a puncturing strategy such that the signal-to-noise ratio required to reach a target bit error rate is minimized.

1 Introduction

To enable efficient communication over channels with highly time-varying characteristics, a common strategy is to adapt the rate of the error control code to the current channel state. In order to reduce decoder complexity, this rate adaptation is often accomplished through puncturing of a common mother code. The transmitter and receiver only share a series of puncturing tables, determining which code bits are to be transmitted for a specific code rate. The receiver then simply inserts erasures for all code bits that are not received. This approach is also used for applications requiring unequal error protection. If the puncturing is made with a rate compatible constraint, implying that all the code bits of a punctured high rate code are used by all lower rate codes, the approach can also be used for incremental redundancy hybrid ARQ (IR-HARQ) schemes. Rate compatible codes imply that, starting with a high rate code in the first transmission, t_1, the transmitter only needs to transmit complementary code bits in t_2, to get to the code with the next highest rate. Rate compatible punctured turbo codes were first considered in [1] for unequal error protection and then independently in [2], [3] and [4] for IR-HARQ schemes.

In this paper, we develop a rate compatible puncturing strategy intended for use in IR-HARQ schemes. Systematic serially concatenated codes are considered, although the method is applicable to any type of concatenated code. Given a family of rate compatible codes that should be obtained through puncturing of a mother code, good puncturing patterns in terms of minimizing the signal-to-noise (SNR) ratio required to reach a target bit error rate (BER) are desirable. Such good puncturing patterns will also maximize the average code rate of IR-HARQ schemes using these patterns.

Selecting good puncturing patterns for concatenated codes by means of simulations is computationally demanding and dependent on the particular interleaver used. Several analytical methods for selecting puncturing patterns have been suggested. In [5] the joint distance spectrum based on uniform interleavers were determined for punctured parallel concatenated codes and used to select good puncturing patterns. These results were extended in [6] for the case where systematic bits are allowed to be punctured. The evaluation of the joint distance spectrum for punctured codes is, however, computationally demanding. The ideas presented in [7] for finding good component encoders for turbo codes, were extended in [8] and [9] to obtain good rate compatible puncturing patterns for parallel and serially concatenated codes, respectively. In these cases the search space is reduced and the calculation of the distance spectrum simplified since the puncturing of each component code is optimized separately. The search may, however, still be computationally demanding and the union bound corresponding to the distance spectrum may not be tight for low SNRs. Since IR-HARQ schemes generate retransmissions at low SNRs, puncturing patterns suitable for this region are of interest.

In [10] extrinsic information transfer (EXIT) charts were used to determine good puncturing strategies for parallel concatenated codes, assuming random puncturing within each component code. An EXIT chart is a powerful semi-analytical tool used to determine the convergence behavior of an iterative decoder as a function of the number of iterations [11]. The EXIT chart is two-dimensional when two component codes are concatenated, but multi-dimensional when more than two

E. Uhlemann was partly funded by the Knowledge Foundation, the national Swedish Real-Time Research initiative ARTES supported by the Swedish Foundation for Strategic Research and by PCC++ under Grant PCC-0201-09. L. K. Rasmussen was supported by the Australian Government under ARC Grant DP0558861. F. Brännström and L. K. Rasmussen were funded in parts by the Swedish Research Council under grants 621-2001-2976 and 723-2002-4533.

component codes are used. Hence, projections onto two dimensions can be applied in these cases [12].

In this paper, EXIT functions are constructed for each component code. These functions are then used to search for a candidate rate compatible puncturing strategy resulting in the lowest possible SNR threshold required for convergence to a target BER. However, since EXIT charts assume infinite interleavers and the particular search procedure used assumes random puncturing within each component code we do not know exactly which bits to puncture. Therefore, rather than a puncturing pattern, the search results in a *puncturing ratio*, δ, for each encoder detailing the percentage of bits remaining after puncturing of each component code. In order to obtain a good puncturing pattern, the puncturing ratios obtained using EXIT charts can be combined with the individual component code puncturing patterns found using the methods in [8, 9]. For some simple codes such as a single parity check (SPC) code, the order of puncturing bits from a particular encoder is irrelevant as long as random interleavers are used. Therefore, for SPC codes the puncturing ratios also give the puncturing pattern. This paper further extends the work in [10] by considering *rate compatible* puncturing ratios for systematic *serially* concatenated codes. Due to their simplicity, SPC codes are used as example codes to show performance results.

2 System Model

The system model is derived assuming block component codes. However, extending the model to any type of component code is straightforward. Fig. 1 shows an encoder for a binary serially concatenated code $C(n,k)$ with three systematic component codes, $C_l(n_l,k_l)$. Each component encoder has as its input the vector $\mathbf{x}$ or an interleaved version of it, $\mathbf{x}_i$ for $i=0,1,2,3$. Note that $\mathbf{x}=\mathbf{x}_0=\mathbf{x}_1$. The output of C_l, for $l=1,2,3$ in Fig. 1 is the n_l-k_l redundant bits from component code l. Further, since the component codes are serially concatenated, the output bits, $\mathbf{y}_l$, from an encoder C_l are used as input to all encoders C_j, where $j>l$. If we want to use identical codes for each of the component codes, the length k of the input frame $\mathbf{x}$ must be a multiple of $(k_1\cdot k_2\cdot k_3)$. The output of each component encoder is a vector, $\mathbf{y}_l\in\{+1,-1\}^{(n_l-k_l)m_l}$, where m_l is an integer. Thus, $\mathbf{x}_2=\Pi_1\left(\left[\mathbf{x}_0,\mathbf{y}_1\right]\right)$ and $\mathbf{x}_3=\Pi_2\left(\left[\mathbf{x}_0,\mathbf{y}_1,\mathbf{y}_2\right]\right)$. The outputs from all component encoders are collected in a vector $\mathbf{y}=\left[\mathbf{x}_0,\mathbf{y}_1,\mathbf{y}_2,\mathbf{y}_3\right]$, which also includes the systematic part. This vector is then sent through individual puncturers. A puncturer Θ_i removes selected bits from its input sequence according to the puncturing pattern for the chosen code rate. The punctured sequences $\boldsymbol{\theta}_i$, $i=0,1,2,3,$ are then multiplexed, $\boldsymbol{\theta}=[\boldsymbol{\theta}_0,\boldsymbol{\theta}_1,\boldsymbol{\theta}_2,\boldsymbol{\theta}_3]$, and forwarded to a BPSK modulator with symbol energy $r_C E_b$. Here E_b is the information bit energy, and r_C is the code rate of C, leading to the transmitted sequence $\mathbf{z}=\boldsymbol{\theta}\sqrt{r_C E_b}$. The sequence is transmitted over an additive white Gaussian noise (AWGN) channel with single-sided spectral density N_0.

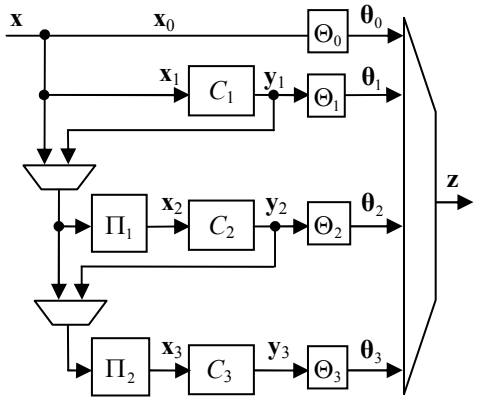

Fig. 1. Encoder for serially concatenated code with three components.

In Fig. 2 the components of an iterative decoder with three component codes are depicted. However, since the number of connections between the different components is large, they are omitted in the figure for clarity. Let $C(\boldsymbol{\theta})=[C(\boldsymbol{\theta}_0),C(\boldsymbol{\theta}_1),C(\boldsymbol{\theta}_2),C(\boldsymbol{\theta}_3)]$ be the sequence of log-likelihood ratios (LLRs) of the corresponding matched filter output at the receiver. The *depuncturers* Θ_i^{-1} insert zeros in the positions in $\mathbf{y}$ corresponding to punctured bits, leading to full-length sequences $C(\mathbf{y})=[C(\mathbf{x}_0),C(\mathbf{y}_1),C(\mathbf{y}_2),C(\mathbf{y}_3)]$ for use in the iterative decoder. The component decoders, marked C_l^{-1} in Fig. 2, take as inputs all available prior information in terms of LLRs of $\mathbf{x}_i$ and $\mathbf{y}_l$ respectively, denoted by $A(\mathbf{x}_i)$ and $A(\mathbf{y}_l)$. Each decoder produces *extrinsic information* of the corresponding vectors in terms of LLRs, denoted by $E(\mathbf{x}_i)$ and $E(\mathbf{y}_l)$. Note that $E(\mathbf{x}_3)$ contains information relevant for the *a priori* input $A(\mathbf{x}_1)$ in the first part of the vector. Hence, let $E(\mathbf{x}_3)=[E(\mathbf{x}_3^{\mathbf{x}_0}),E(\mathbf{x}_3^{\mathbf{y}_1}),E(\mathbf{x}_3^{\mathbf{y}_2})]$ denote the different parts of the vector of extrinsic information obtained from C_3^{-1}. Using the appropriately interleaved version

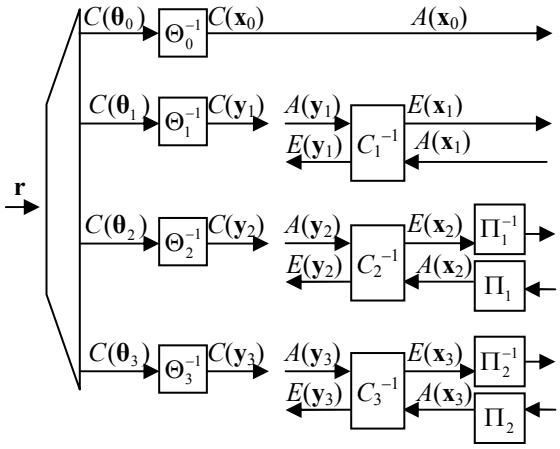

Fig. 2. Decoder for serially concatenated code with three components. All connections are omitted for clarity, and given below as equations.

of each vector, this yields the following equations or connections related to Fig. 2:

$$A(\mathbf{x}_1) = C(\mathbf{x}_0) + \Pi_1^{-1}\left(E\left(\mathbf{x}_2^{\mathbf{x}_0}\right)\right) + \Pi_2^{-1}\left(E\left(\mathbf{x}_3^{\mathbf{x}_0}\right)\right), \quad (1)$$

$$A(\mathbf{y}_1) = C(\mathbf{y}_1) + \Pi_1^{-1}\left(E\left(\mathbf{x}_2^{\mathbf{y}_1}\right)\right) + \Pi_2^{-1}\left(E\left(\mathbf{x}_3^{\mathbf{y}_1}\right)\right), \quad (2)$$

$$\begin{aligned} A(\mathbf{x}_2) = \Pi_1\Big(\Big[\Big(C(\mathbf{x}_0) + E(\mathbf{x}_1) + \Pi_2^{-1}\left(E\left(\mathbf{x}_3^{\mathbf{x}_0}\right)\right)\Big), \\ \Big(C(\mathbf{y}_1) + E(\mathbf{y}_1) + \Pi_2^{-1}\left(E\left(\mathbf{x}_3^{\mathbf{y}_1}\right)\right)\Big)\Big]\Big), \end{aligned} \quad (3)$$

$$A(\mathbf{y}_2) = C(\mathbf{y}_2) + \Pi_2^{-1}\left(E\left(\mathbf{x}_3^{\mathbf{y}_2}\right)\right), \quad (4)$$

$$\begin{aligned} A(\mathbf{x}_3) = \Pi_2\Big(\Big[\Big(C(\mathbf{x}_0) + E(\mathbf{x}_1) + \Pi_1^{-1}\left(E\left(\mathbf{x}_2^{\mathbf{x}_0}\right)\right)\Big), \\ \Big(C(\mathbf{y}_1) + E(\mathbf{y}_1) + \Pi_1^{-1}\left(E\left(\mathbf{x}_2^{\mathbf{y}_1}\right)\right)\Big), \big(C(\mathbf{y}_2) + E(\mathbf{y}_2)\big)\Big]\Big), \end{aligned} \quad (5)$$

$$A(\mathbf{y}_3) = C(\mathbf{y}_3). \quad (6)$$

Finally, the decision statistics on $\mathbf{x}$ is given by

$$\begin{aligned} D(\mathbf{x}) = C(\mathbf{x}_0) + E(\mathbf{x}_1) + \\ \Pi_1^{-1}\left(E\left(\mathbf{x}_2^{\mathbf{x}_0}\right)\right) + \Pi_2^{-1}\left(E\left(\mathbf{x}_3^{\mathbf{x}_0}\right)\right). \end{aligned} \quad (7)$$

3 EXIT Functions

Let $I_{A(x_i)}$ denote the mutual information (MI) between the input bits $\mathbf{x}_i$ and the prior LLRs $A(\mathbf{x}_i)$ and similarly for $\mathbf{y}_l$. Further, let $I_{E(x_i)}$ denote the MI between the input bits $\mathbf{x}_i$ and the extrinsic LLRs $E(\mathbf{x}_i)$ and similarly for $\mathbf{y}_l$. For notational simplicity, we define

$$I_{E(x_1)} \oplus I_{E(x_2)} \triangleq J\left(\sqrt{J^{-1}\left(I_{E(x_1)}\right)^2 + J^{-1}\left(I_{E(x_2)}\right)^2}\right), \quad (8)$$

where the J function is defined in [11] as

$$J(\sigma) = \int_{-\infty}^{+\infty} \frac{e^{-\left(\alpha - \frac{\sigma^2}{2}\right)/2\sigma^2}}{\sqrt{2\pi\sigma^2}} \left(1 - \log_2\left(1 + e^{-\alpha}\right)\right) d\alpha. \quad (9)$$

$A(\mathbf{x}_1)$ is determined based on input from the channel and from the two other component codes according to (1). Further, let $0 \le \delta_0 \le 1$ denote the fraction of bits that remains after puncturing in $C(\mathbf{x}_0)$, i.e., the fraction of $\mathbf{x}_0$ that is transmitted and hence also the fraction of elements in $A(\mathbf{x}_1)$ that has information from $C(\mathbf{x}_0)$, $E(\mathbf{x}_2)$ and $E(\mathbf{x}_3)$. Consequently, $(1-\delta_0)$ is the fraction of elements in $A(\mathbf{x}_1)$ that has information only from $E(\mathbf{x}_2)$ and $E(\mathbf{x}_3)$, since puncturing is made only on the prior information received from the channel. Note that the different dimensions in a serially concatenated system are not identical. Each dimension adds more redundancy than the previous, thus $\delta_1 = 0$ implies fewer punctured parity bits than if $\delta_2 = 0$. Hence [13],

$$\begin{aligned} I_{A(x_1)} = \delta_0\left(I_{C(x_0)} \oplus I_{E(x_2)} \oplus I_{E(x_3)}\right) + \\ + (1-\delta_0)\left(I_{E(x_2)} \oplus I_{E(x_3)}\right), \end{aligned} \quad (10)$$

where $I_{C(x_0)} = J(\sqrt{8 r_C E_b/N_0})$, and similarly from (2)

$$\begin{aligned} I_{A(y_1)} = \delta_1\left(I_{C(y_1)} \oplus I_{E(x_2)} \oplus I_{E(x_3)}\right) + \\ + (1-\delta_1)\left(I_{E(x_2)} \oplus I_{E(x_3)}\right). \end{aligned} \quad (11)$$

Since $\mathbf{x}_2 = \Pi_1([\mathbf{x}_0, \mathbf{y}_1])$ it means that k_1/n_1 of the elements in $A(\mathbf{x}_2)$ comes from the systematic bits, $\mathbf{x}_0$, and $(n_1 - k_1)/n_1$ of the elements come from redundant bits, $\mathbf{y}_1$. Further, δ_0 of the elements coming from the systematic bits have information from $C(\mathbf{x}_0)$, $E(\mathbf{x}_1)$ and $E(\mathbf{x}_3)$, whereas $(1-\delta_0)$ of the elements have information only from $E(\mathbf{x}_1)$ and $E(\mathbf{x}_3)$. The same reasoning can be made for the $(n_1 - k_1)/n_1$ elements coming from redundant bits, but here with δ_1 instead. Using (3) we have

$$\begin{aligned} I_{A(x_2)} = \delta_0 \frac{k_1}{n_1}\left(I_{C(x_0)} \oplus I_{E(x_1)} \oplus I_{E(x_3)}\right) + \\ + (1-\delta_0)\frac{k_1}{n_1}\left(I_{E(x_1)} \oplus I_{E(x_3)}\right) + \\ + \delta_1 \frac{n_1 - k_1}{n_1}\left(I_{C(y_1)} \oplus I_{E(y_1)} \oplus I_{E(x_3)}\right) + \\ + (1-\delta_1)\frac{n_1 - k_1}{n_1}\left(I_{E(y_1)} \oplus I_{E(x_3)}\right), \end{aligned} \quad (12)$$

and similarly from (4)

$$I_{A(y_2)} = \delta_2\left(I_{C(y_2)} \oplus I_{E(x_3)}\right) + (1-\delta_2) I_{E(x_3)}. \quad (13)$$

Finally, for C_3^{-1} we have $\mathbf{x}_3 = \Pi_2([\mathbf{x}_0, \mathbf{y}_1, \mathbf{y}_2])$, hence using (5) together with the above reasoning yields

$$\begin{aligned} I_{A(x_3)} = \delta_0 \frac{k_1}{n_1}\cdot\frac{k_2}{n_2}\left(I_{C(x_0)} \oplus I_{E(x_1)} \oplus I_{E(x_2)}\right) + \\ + (1-\delta_0)\frac{k_1}{n_1}\cdot\frac{k_2}{n_2}\left(I_{E(x_1)} \oplus I_{E(x_2)}\right) + \\ + \delta_1 \frac{n_1 - k_1}{n_1}\cdot\frac{k_2}{n_2}\left(I_{C(y_1)} \oplus I_{E(y_1)} \oplus I_{E(x_2)}\right) + \\ + (1-\delta_1)\frac{n_1 - k_1}{n_1}\cdot\frac{k_2}{n_2}\left(I_{E(y_1)} \oplus I_{E(x_2)}\right) + \\ + \delta_2 \frac{n_2 - k_2}{n_2}\left(I_{C(y_2)} \oplus I_{E(y_2)}\right) + \\ + (1-\delta_2)\frac{n_2 - k_2}{n_2} I_{E(y_2)}. \end{aligned} \quad (14)$$

and similarly from (6)

$$I_{A(y_3)} = \delta_3 I_{C(y_3)}. \quad (15)$$

The extrinsic MIs for decoders $l = 1, 2, 3$ are functions of the prior MIs according to

$$I_{E(x_l)} = f_{x_l}\left(I_{A(x_l)}, I_{A(y_l)}\right), \quad (16)$$

$$I_{E(y_l)} = f_{y_l}\left(I_{A(x_l)}, I_{A(y_l)}\right), \tag{17}$$

where f_{x_l} and f_{y_l} are the EXIT functions for the input and output bits for decoder l, respectively [12, 13], obtained using Monte Carlo simulations [11]. The MI between the input symbols $\mathbf{x}$ and the decision statistics $D(\mathbf{x})$ in (7) converges to a fixed value denoted by I_D given a sufficient number of iterations have been performed [12],

$$I_D = I_{C(x_0)} \oplus I_{E(x_1)} \oplus I_{E(x_2)} \oplus I_{E(x_3)}. \tag{18}$$

4 Search Procedure

We consider three different scenarios when searching for good puncturing ratios:

1. Search with no constraints on puncturing.
2. Search with the constraint that no systematic bits are allowed to be punctured.
3. Search for rate compatible puncturing ratios.

For search scenario 1 the following procedure is used. For a given code rate, r_C, and a number of component codes U, we conduct a search to find the optimal values of $\Delta = [\delta_0, \delta_1, \ldots, \delta_U]$, leading to the lowest possible SNR value $\gamma_b \triangleq E_b / N_0$ required for reaching a target BER, P_b^*. The target BER corresponds to a target MI, I_D^*, assuming Gaussian behavior, $I_D^* = J(2Q^{-1}(P_b^*))$, where Q is the integral of the tail of a Gaussian probability density function [11]. Further, the MI of the decision statistics I_D is a function of Δ and γ_b, $I_D = f_{I_D}(\Delta, \gamma_b)$ as described in Section 3. We formulate an objective function according to

$$g(\gamma_b, \Delta) = \begin{cases} \gamma_b & f_{I_D}(\Delta, \gamma_b) \geq J\left(2Q^{-1}\left(P_b^*\right)\right), \\ \infty & \text{otherwise.} \end{cases} \tag{19}$$

Following the approach in [10], an optimization problem is formulated for finding optimal values of γ_b, Δ :

$$\left[\gamma_b^*, \Delta^*\right] = \arg\min_{[\gamma_b, \Delta]} g(\gamma_b, \Delta) \tag{20}$$

subject to code rate and puncturing ratio constraints

$$r_C = \left(\sum_{i=0}^{U} \frac{\delta_i}{\eta_i}\right)^{-1}, \qquad \Delta \in [0,1]^{U+1} \tag{21}$$

with $\eta_0 = 1$ and $\eta_i = n_l (k_l / n_l)^i$ for $i = 1, 2, \ldots, U$ if the same component code C_l with parameters (n_l, k_l) is used in all dimensions. Thus, r_C is the code rate of the concatenated code expressed as a function of Δ.

For a given code rate, r_C, we evaluate I_D after convergence for all values of Δ that yield the required code rate. Starting with the first valid Δ and an initial value of $\gamma_b = Q^{-1}(P_b^*)^2 / 2$, we update the extrinsic MI for all the component decoders until the decision statistics has converged. As long as I_D is above the target MI $I_D^* = J(2Q^{-1}(P_b^*))$, we reduce γ_b by a small amount, $\varepsilon_\gamma > 0$ until I_D falls below I_D^*. The search continues until all valid Δ have been checked and the minimum γ_b and corresponding Δ have been found. More details about the search procedure can be found in [10].

For search scenario 2, we conduct the search with the constraint that $\delta_0 = 1$. For scenario 3 we simply never allow the δ-values to be increased once decreased.

5 Performance Results

Let 2D SCSPC(8,7) denote a two-dimensional (2D), i.e., two component codes, serially concatenated (SC) code, encoded in each dimension with a systematic SPC(8,7) code. We focus on two example codes, the 2D SCSPC(8,7) and the 3D SCSPC(8,7). Note that the 2D code is equivalent to the 3D code when $\delta_3 = 0$.

The optimal values of Δ for the 3D code using search scenario 1 are plotted in Fig. 3 for $P_b^* = 10^{-5}$ at different code rates. It can be seen that it is preferable to puncture systematic bits for medium code rates. Also, when puncturing parity bits we should start with δ_1, corresponding to the outermost code, C_1.

In Fig. 4 the γ_b needed to reach P_b^* is plotted as a function of the code rate for the 2D and the 3D SCSPC(8,7) codes. Curves corresponding to four different values of P_b^* are plotted together with the capacity for the binary input AWGN channel, C^*. It is possible to achieve rates above the binary input capacity when the target BER is sufficiently high. The curves for the 3D code start at rate $(7/8)^3$ and the 2D codes at $(7/8)^2$ since these are the lowest possible code rates of the two schemes, respectively.

We can see from Fig. 4 that the performance of the 3D code is significantly better than that of the 2D code. The performance of the 2D and 3D codes is only equal for code rates close to one. The knee on all the 2D curves with a low target BER, is due to the simple structure of the 2D SCSPC example code. For example, above rate 7/8 not all information blocks of 7 can be

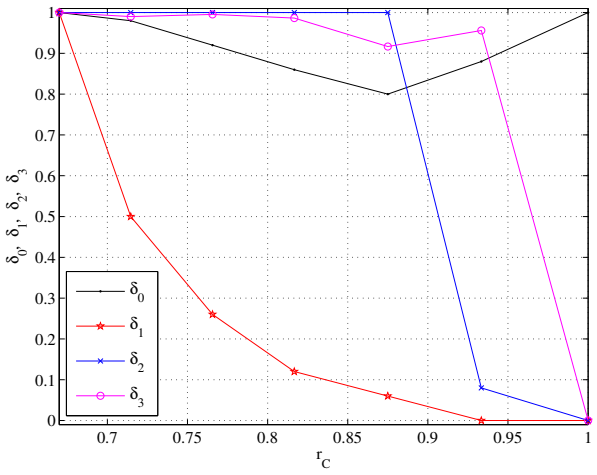

Fig. 3. Optimal puncturing ratios for the 3D SCSPC(8,7) code when the target BER is 10^{-5} and search scenario 1 is used.

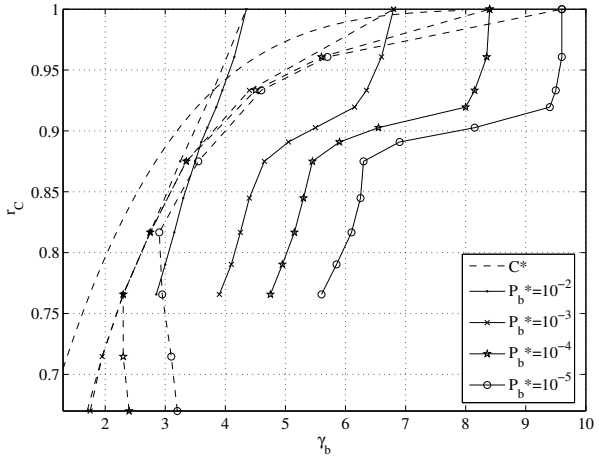

Fig. 4. Code rate versus minimum SNR required to reach P_b^* for 2D (solid lines) and 3D (dashed lines) codes using Δ^*. Search scenario 1.

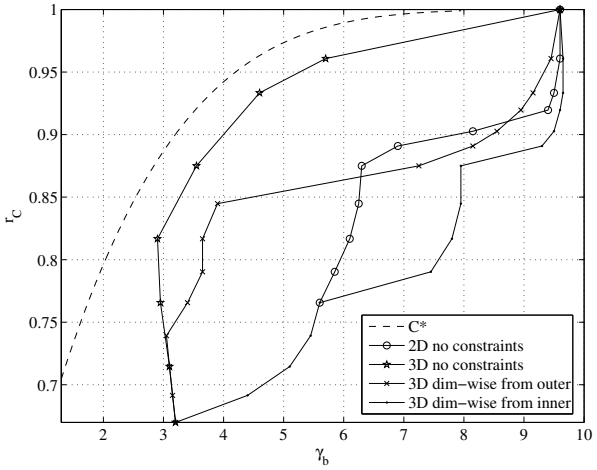

Fig. 5. Code rate versus minimum SNR required to reach $P_b^* = 10^{-5}$ for a 2D and a 3D SCSPC(8,7) code that use optimal Δ-values without constraints (search scenario 1), and for a 3D SCSPC(8,7) code with different constraints (search scenario 2 and 3).

protected by 1 whole parity bit, which affects the minimum distance of the code. Each additional parity bit around this code rate thus yields significant improvements. With a relaxed constraint on the target BER, however, more systematic bits can be punctured, thus increasing the unequal error protection. If too many systematic bits are punctured, some information bits may not be possible to recreate. When this happens the code is said to have lost its invertibility.

The *local* invertibility is lost when two bits pertaining to the same *component* codeword in an SPC code are punctured. Depending on the interleavers, the invertibility may or may not be restored by the component codes in the remaining dimensions. Since EXIT charts assume infinite interleavers, the probability of pairing up several punctured bits in the same component code approaches zero, as the component code frame length, k_l, is much smaller than the interleaver size. This may explain why the EXIT chart analysis suggests puncturing systematic bits. With knowledge of the particular random interleavers used, the puncturing pattern can be adapted so that the local invertibility is guaranteed. In this case, systematic bits should be punctured to increase the performance. However, if we want to guarantee the local invertibility regardless of which interleavers are used, puncturing of systematic bits should be avoided for SCSPC codes.

Hence, we consider search scenario 2 where systematic bits are not allowed to be punctured. The optimal puncturing ratios for the 3D SCSPC(8,7) code found using the constraint search in scenario 2 are dimension-wise puncturing starting with the outermost code. That is to say that for optimal performance with $\delta_0 = 1$, we should start puncturing bits pertaining to C_1, the outermost code, and continue doing so until all these bits have been punctured. Thereafter, we should start puncturing bits pertaining to C_2, and continue doing so until all these bits have been punctured, and so forth. This result is intuitive since the inner codes contain sufficient information for activating the outermost decoders regardless of whether their parity bits are punctured or not. Note that dimension-wise puncturing starting with the outermost code also implies rate compatible puncturing ratios. The performance of the 3D SCSPC(8,7) code using dimension-wise puncturing starting with the outermost code is reported in Fig. 5. The performance of the 2D and 3D codes using search scenario 1 are also plotted in Fig. 5 for reference. As can be seen, the puncturing ratios found with no search constraints perform much better than the puncturing ratios found when not allowing systematic bits to be punctured, i.e., dimension-wise puncturing starting from the outermost code. For medium code rates, when the 2D code punctures systematic bits, it is superior to the 3D code with dimension-wise puncturing. For high and low code rates, however, the 3D scheme is superior.

When considering search scenario 3, i.e., searching for *rate compatible* puncturing ratios, we start from the lowest possible code rate and start puncturing, i.e., start decreasing the δ-values , making sure that none of them are allowed to be increased once decreased. This search results in puncturing ratios with a performance that follows the optimal curve with no constraints on Δ in Fig. 5 up until $r_C = 0.875$. The puncturing ratios thus require more and more systematic bits to be punctured as the code rate increases. However, when reaching a sufficiently high code rate it is no longer possible to find a Δ for which $I_D \geq I_D^*$. To allow for higher code rates within this particular code family, the BER performance must be compromised. This results in poor performance for high code rates, but optimal for low rates. If code rates up to one are required with $P_b^* = 10^{-5}$ no systematic bits can be punctured. The search for rate compatible puncturing ratios under the constraint that $\delta_0 = 1$, results in dimension-wise puncturing starting with the outermost code with performance as reported in Fig. 5. This means that search scenario 2 and 3 yield the same result in this example.

It may be argued that for an IR-HARQ scheme, it is preferable to optimize the performance at high code rates since that may increase the average code rate of the IR-ARQ scheme. Hence, the search could alternatively start at the highest possible code rate. Searching for the best puncturing ratios for a 3D SCSPC(8,7) code at $r_C = 1$ yields $\Delta = [1,0,0,0]$, i.e., only systematic bits should be transmitted. Consequently, in this case the search starts with $\delta_0 = 1$ which will inherently result in dimension-wise puncturing, starting with the outermost code. This is again the same result as search scenario 2, with performance according to Fig. 5. The only way to get different rate compatible puncturing ratios is if the search starts at a code rate $r_C < 1$, implying that not all high rates are included in the rate compatible code family. The particular rate compatible code family chosen is thus important since a different set of code rates used in the search may result in different rate compatible puncturing ratios. For an IR-HARQ scheme this implies that a longer packet is sent in the initial transmission, t_1. For example, if the highest code rate required is $r_C = 0.875$, then the optimal Δ-values for $0.67 \le r_C \le 0.875$ shown in Fig. 3 are rate compatible since they are all decreasing. Hence, rate compatible puncturing ratios including punctured systematic bits would now instead be obtained. Consequently, dimension-wise puncturing starting with the outermost code is only guaranteed to be the best for the rate compatible code family including all possible rates. Therefore, with knowledge of the particular code rate requirements better puncturing ratios can be found.

Dimension-wise puncturing starting with the outermost code implies that the decoding complexity is fixed for all rates in the code family since all decoders have to be activated as soon as parity bits pertaining to the innermost decoder arrive. A puncturing pattern that reduces decoding complexity at higher rates is dimension-wise puncturing starting with the innermost code, since only the component decoders that has received parity bits so far needs to be activated. Hence, even if the mother code is based on three component codes concatenated serially, only the first or the first and second component decoders may need to be activated for the first few transmissions. The performance in terms of γ_b required for convergence to a target BER using dimension-wise puncturing starting with the innermost code is reported in Fig. 5. Dimension-wise puncturing starting with the outermost requires a lower γ_b for convergence, and thus fewer retransmissions, which may in turn result in a lower average decoder complexity. However, if performance in terms of decoding complexity per retransmission is important, dimension-wise puncturing starting with the innermost code should be selected. Consequently, the puncturing pattern should be selected according to the specific application requirements.

6 Conclusions

Rate compatible puncturing strategies intended for IR-HARQ schemes have been derived using EXIT charts. The selected puncturing ratios minimize the SNR required to reach a target BER. If the search for rate compatible puncturing ratios starts from the lowest possible code rate, the performance of the low rates in the code family will be good, i.e., close or equal to the performance when using optimal unconstrained puncturing ratios. However, the performance at high rates in the code family may be poor. Conversely, starting the search from a high rate results in puncturing ratios with good performance for this rate, but not necessarily good performance for low rates as compared to the case with unconstrained puncturing. Given a set of component codes, the particular interleavers used, the specific rate compatible code rates needed and performance requirements all affect the choice of puncturing strategy.

References

[1] A. S. Barbulescu and S. S. Pietrobon, "Rate compatible turbo codes," *Electronics Letters*, vol. 31, pp. 535-536, Mars 1995.

[2] P. Jung, J. Plechinger, M. Doetsch, and F. M. Berens, "A pragmatic approach to rate compatible punctured turbo-codes for mobile radio applications," in *Proc. Int. Conf. on Advances in Commun. and Control*, Corfu, Greece, June 1997.

[3] J. Li and H. Imai, "Performance of hybrid-ARQ protocols with rate compatible turbo codes," in *Proc. Int. Symp. on Turbo Codes & Related Topics*, Brest, France, Sept. 1997, pp. 188-191.

[4] D. N. Rowitch and L. B. Milstein, "Rate compatible punctured turbo (RCPT) codes in a hybrid FEC/ARQ system," in *Proc. Commun. Theory Mini-Conf. of GLOBECOM '97*, Phoenix, AZ, Nov. 1997, pp. 55-59.

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

[6] E. Uhlemann and L. K. Rasmussen, "Analytical approach for maximizing the average code rate of incremental redundancy schemes," in *Proc. Asia-Pacific Conf. on Commun.*, Perth, Australia, Oct. 2005, pp. 481-485.

[7] S. Benedetto, R. Garello, and G. Montorsi, "A search for good convolutional codes to be used in the construction of turbo codes," *IEEE Trans. on Commun.*, vol. 46, no. 9, Sept. 1998.

[8] F. Babich, G. Montorsi, and F. Vatta, "Design of rate-compatible punctured turbo (RCPT) codes," in *Proc. IEEE Int. Conf. on Commun.,* New York City, NY, April 2002, pp. 1701-1705.

[9] A. Graell i Amat, G. Montorsi, and F. Vatta, "Analysis and design of rate compatible serial concatenated convolutional codes," in *Proc. IEEE Int. Symp. on Information Theory*, Adelaide, Australia, Sept. 2005, pp. 607-611.

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

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

[12] F. Brännström, L. K. Rasmussen, and A. J. Grant, "Convergence analysis and optimal scheduling for multiple concatenated codes," *IEEE Trans. on Information Theory*, vol. 51, pp. 3354–3364, Sept. 2005.

[13] E. Uhlemann, *Adaptive Concatenated Coding for Wireless Real-Time Communications*, Ph.D. thesis, Chalmers University of Technology, Göteborg, Sweden, Sept. 2004.