

Analysis and Design of Punctured Rate-1/2 Turbo Codes Exhibiting Low Error Floors

Ioannis Chatzigeorgiou, *Member, IEEE*, Miguel R. D. Rodrigues, *Member, IEEE*,
Ian J. Wassell, and Rolando A. Carrasco

Abstract

The objective of this paper is two-fold. Initially, we present an analytic technique to rapidly evaluate an approximation to the union bound on the bit error probability of turbo codes. This technique exploits the most significant terms of the union bound, which can be calculated straightforwardly by considering the properties of the constituent convolutional encoders. Subsequently, we use the bound approximation to demonstrate that particular punctured rate-1/2 turbo codes can achieve a lower error floor than that of their rate-1/3 parent codes. In particular, we propose pseudo-random puncturing as a means of improving the bandwidth efficiency of a turbo code and simultaneously lowering its error floor.

Index Terms

Turbo codes, puncturing, transfer function, performance evaluation, error floor

I. INTRODUCTION

In certain applications, such as satellite communications, link reliability is of prime importance and, consequently, low rate codes are often used. However, bandwidth occupancy is of much

This research was supported by EPSRC under Grant GR/S46437/01. Parts of this paper have been presented at the 2007 Conference on Information Sciences and Systems and at the 2007 International Symposium on Information Theory.

I. Chatzigeorgiou and I. J. Wassell are with the Computer Laboratory, University of Cambridge, Cambridge CB3 0FD, United Kingdom (e-mail: ic231@cam.ac.uk; ijw24@cam.ac.uk).

Miguel R. D. Rodrigues is with the Instituto de Telecomunicações and the Department of Computer Science, University of Porto, Porto 4169-007, Portugal (e-mail: mrodrigues@dcc.fc.up.pt).

Rolando A. Carrasco is with the School of Electrical, Electronic and Computer Engineering, University of Newcastle-upon-Tyne, Newcastle NE1 7RU, United Kingdom (e-mail: r.carrasco@newcastle.ac.uk).

greater importance in wireless communications and so high rate codes are preferred. A high rate binary convolutional code can be obtained by periodic elimination, known as puncturing, of particular code bits from the output of a parent low rate convolutional encoder. Extensive analysis on punctured convolutional codes has shown that their performance is always inferior to the performance of their low rate parent codes (e.g. see [1], [2]).

The performance of punctured parallel concatenated convolutional codes (PCCCs), also known as punctured turbo codes, has also been investigated. Design considerations have been derived by semi-analytical [3]–[6] and simulation-based approaches [7]–[9], while upper bounds on the bit error probability (BEP) were evaluated in [5], [10]. Punctured turbo codes are usually classified as systematic (S), partially systematic (PS) or nonsystematic (NS) depending on whether all, some or none of their systematic bits are transmitted [8]. Recent papers [8]–[10] have demonstrated that partially systematic PCCCs yield lower error floors than systematic PCCCs of the same rate.

The average performance of a PCCC can be evaluated using the union bound on its BEP, which coincides with the error floor of the PCCC for high E_b/N_0 values [11]. Calculation of the bound requires knowledge of the transfer function of the PCCC under investigation. However, computation of the transfer function becomes intensive as the interleaver size increases, especially in the case of punctured PCCCs [10]. The motivation for this paper is to propose a framework to rapidly evaluate an accurate approximation to the union bound, which will aid in the analysis and design of punctured PCCCs operating in the error floor region. The proposed framework also enables us to investigate whether particular punctured PCCCs achieve a lower error floor, and thus a better performance at high E_b/N_0 values, than that of their parent codes.

More specifically, in Section II we introduce definitions for the case of rate-1/3 PCCCs, whilst in Section III we provide expressions for the computation of the union bound. In Section IV we explain that, for large interleaver sizes, only particular dominant terms of the union bound need to be considered to closely approximate the overall union bound. We also study the properties of the constituent encoders and demonstrate that the dominant terms can be expressed as functions of the corresponding code characteristics. In Section V we use the bound approximation to derive criteria for the performance evaluation of turbo codes. Based on these criteria, we compare specific PCCC configurations and we demonstrate that pseudo-random puncturing can be used to obtain rate-1/2 PCCCs that exhibit a lower error floor than that of their parent rate-1/3 PCCCs. The paper concludes in Section VI with a summary of the main contributions.

II. BINARY TURBO CODES: DEFINITIONS AND NOTATION

A rate-1/3 binary turbo code is the parallel concatenation of two rate-1/2 recursive systematic convolutional (RSC) encoders separated by an interleaver of size N [12]. The information bits are input to the upper constituent encoder, while an interleaved version of the information bits are input to the lower encoder. The output of the rate-1/3 turbo encoder consists of the systematic bits of the upper RSC encoder, which are identical to the input information bits, the parity check bits of the upper RSC encoder and the parity check bits of the lower RSC encoder.

Rates higher than 1/3 can be obtained by periodic elimination of specific code bits from the output of a rate-1/3 turbo encoder. Row vectors \mathbf{P}_{U_1} , \mathbf{P}_{Z_1} and \mathbf{P}_{Z_2} , which determine the bits that will be eliminated from the systematic and the two parity check outputs of the encoder, compose the puncturing pattern \mathbf{P} , which is repeated periodically every M time steps:

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{U_1} \\ \mathbf{P}_{Z_1} \\ \mathbf{P}_{Z_2} \end{bmatrix} = \begin{bmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,M} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,M} \\ p_{3,1} & p_{3,2} & \cdots & p_{3,M} \end{bmatrix}, \quad (1)$$

where $p_{i,m} \in \{0, 1\}$, with $i = 1, 2, 3$ and $m = 1, \dots, M$. For $p_{i,m} = 0$, the corresponding output bit is punctured, otherwise it is transmitted.

Throughout this paper we use the notation $W^w U^u Z^z$ to describe the weight properties of a code sequence of a systematic encoder, either convolutional or turbo. The code sequence, which has been generated by an information sequence of Hamming weight w and length N , comprises a systematic sequence and a parity check sequence having weights u and z , respectively; W , U and Z are indeterminate variables. If the output of the encoder is not punctured, then $w = u$.

We consider a turbo encoder that accepts input sequences of length N , or equivalently uses an interleaver of size N , and we denote as \mathcal{P} the corresponding turbo code. The transfer function $B^{\mathcal{P}}(W, U, Z)$ of \mathcal{P} enumerates all code sequences that correspond to trellis paths of length N , which start from the zero state, can re-visit it more than once and can remain at it for an indefinite period of time. Tail-bits, appended to each input sequence, and a double terminating interleaver [13] ensure that trellis paths return to the zero state. A generic form for the transfer function is

$$B^{\mathcal{P}}(W, U, Z) = \sum_{w,u,z} B_{w,u,z}^{\mathcal{P}} W^w U^u Z^z, \quad (2)$$

where $B_{w,u,z}^{\mathcal{P}}$ denotes the number of code sequences having weights w , u and z . The transfer function can be re-written as a sum of conditional weight enumerating functions (CWEFs), each

of which provides all code sequences generated by input information sequences of a particular weight w . More specifically, the transfer function $B^{\mathcal{P}}(W, U, Z)$ can be computed by taking the CWEF for a particular value of w , defined as [11]

$$B_w^{\mathcal{P}}(U, Z) = \sum_{u,z} B_{w,u,z}^{\mathcal{P}} U^u Z^z, \quad (3)$$

and summing over all possible values of w , i.e.,

$$B^{\mathcal{P}}(W, U, Z) = \sum_w B_w^{\mathcal{P}}(U, Z) W^w. \quad (4)$$

III. CONVENTIONAL PERFORMANCE EVALUATION

Computation of the CWEF of a turbo code \mathcal{P} from the CWEFs of its constituent RSC codes, \mathcal{C}_1 and \mathcal{C}_2 , is exceedingly complex since it depends on the permutation scheme employed by the interleaver. However, by adopting the probabilistic concept of uniform interleaving of size N , it is possible to calculate the CWEF of the turbo code from [11]

$$B_w^{\mathcal{P}}(U, Z) = \frac{B_w^{\mathcal{C}_1}(U, Z) \cdot B_w^{\mathcal{C}_2}(U=1, Z)}{\binom{N}{w}}, \quad (5)$$

where $B_w^{\mathcal{C}_1}(U, Z)$ and $B_w^{\mathcal{C}_2}(U, Z)$ are the CWEFs of the constituent RSC codes. Note that the systematic output sequence of the lower constituent encoder is not transmitted, so it is eliminated by setting $U=1$ in $B_w^{\mathcal{C}_2}(U, Z)$. We shall assume throughout the paper, when punctured turbo codes are considered, \mathcal{C}_1 and \mathcal{C}_2 denote the constituent RSC codes *after* puncturing. Methods for the transfer function calculation of RSC codes have been described in [10], [11] and [14].

The bit error probability (BEP) P_b of a PCCC employing maximum-likelihood (ML) soft decision decoding on an additive white Gaussian noise (AWGN) channel, is upper bounded as follows

$$P_b \leq P_b^u = \sum_w P(w) \quad (6)$$

where P_b^u is the union bound on the BEP. The sum runs over all possible values of information weight w , with $P(w)$ being the contribution to the union bound of only those erroneously decoded code sequences which correspond to information sequences of a specific weight w . The contribution $P(w)$ is given by [11]

$$P(w) = \sum_d \frac{w}{N} B_{w,d}^{\mathcal{P}} Q \left(\sqrt{\frac{2R^{\mathcal{P}} \cdot E_b}{N_0} \cdot d} \right), \quad (7)$$

where N is the interleaver size, $R^{\mathcal{P}}$ is the code rate of the turbo code \mathcal{P} , E_b is the energy per information bit, N_0 is the noise spectral density and $B_{w,d}^{\mathcal{P}}$ denotes the number of code sequences having overall weight d , which were generated by input information sequences of weight w . Taking into account that a code sequence comprises a systematic sequence of weight u and a parity check sequence of weight z , its overall weight is $u + z = d$. Hence, coefficients $B_{w,d}^{\mathcal{P}}$ for a particular value of d can be derived from the coefficients $B_{w,u,z}^{\mathcal{P}}$ of the transfer function, using

$$B_{w,d}^{\mathcal{P}} = \sum_{\substack{u,z \\ u+z=d}} B_{w,u,z}^{\mathcal{P}}. \quad (8)$$

It is important to note that the union bound on the BEP of a PCCC using a uniform interleaver of size N coincides with the average of the union bounds obtainable from the whole class of deterministic interleavers of size N [11]. For small values of N , the union bound can be loose compared with the exact BEP of turbo codes using deterministic interleavers. However, for $N \geq 1000$, deterministic interleaver designs do not perform as well as randomly generated interleavers [15], which in turn perform similarly to uniform interleavers. Improved bounds, tighter than the union bound at low E_b/N_0 values, have also been presented in [16]. For our work, we have elected to use the much simpler union bound argument because it provides a good indication of the performance of a PCCC in the error floor region, when N is large.

IV. RAPID PERFORMANCE EVALUATION

Computation of the transfer function and, consequently, the union bound on the BEP of a punctured turbo code becomes highly intensive as the interleaver size N and the puncturing period M increase [10]. In this section we revisit the structure of the constituent RSC encoders in order to develop a rapid method to calculate the most significant terms of the transfer function of a punctured turbo code, which can then be used to obtain a close approximation to the union bound for medium to low BEP values.

A. A Union Bound Approximation

Benedetto *et al.* [17] and Chatzigeorgiou *et al.* [18] investigated the performance of nonpunctured rate-1/3 turbo codes and punctured rate-1/2 turbo codes, respectively. It was observed that code sequences having the minimum possible information weight w_{\min} are the main contributors to the bit error performance, as the size N of the uniform interleaver increases. Owing to the

structure of the constituent RSC encoders, the minimum information weight of an input sequence to a PCCC is equal to two, i.e., $w_{\min}=2$.

Consequently, for large interleaver sizes, it follows that $P(2)$ is the dominant contribution to the union bound P_b^u on the average BEP, so that

$$P_b^u \approx P(2), \quad (9)$$

where $P(2)$ assumes the form

$$P(2) = \sum_d \frac{2}{N} B_{2,d}^P Q \left(\sqrt{\frac{2R^P \cdot E_b}{N_0} \cdot d} \right), \quad (10)$$

based on (7). The coefficients $B_{2,d}^P$ can be derived from (8), provided that the transfer function $B^P(W, U, Z)$ of the turbo code is known. However, the CWEF for $w=2$, denoted as $B_2^P(U, Z)$ and referred to as the *dominant CWEF*, is sufficient to extract the same necessary information. The expression for $B_2^P(U, Z)$ is derived from (5) for $w=2$.

This indicates that to compute the union bound approximation for a turbo code that uses a long interleaver, it suffices to calculate the dominant CWEFs of the constituent convolutional codes, rather than their complete transfer functions. In the following subsections we analyze the structure of the RSC encoder and devise a simple method to compute the less computationally demanding dominant CWEF.

B. Nonpunctured Constituent Codes

We consider a constituent rate-1/2 RSC code \mathcal{C} , where \mathcal{C} can be either \mathcal{C}_1 or \mathcal{C}_2 . The respective encoder consists of ν memory elements in the form of registers, r_1, r_2, \dots, r_ν , as shown in Fig. 1. The memory state of the encoder can be expressed in decimal form as $s = \sum_{i=1}^{\nu} r_i 2^{\nu-i}$. Based on the design considerations presented in [17], the feedback generator polynomial G_R is taken to be primitive, whilst the feedforward generator polynomial G_F is taken to be monic; the degree of both G_R and G_F is ν . Initially, the encoder is in the zero memory state, i.e., $r_i=0$ for all i .

We denote as x_t the input to the RSC encoder at time step t and y_t^u, y_t^z the systematic and parity check bits, respectively, generated by the encoder. Note that $y_t^u = x_t$ for every value of t . For convenience, we assume that at time step $t=0$, the first nonzero bit $x_0=1$ is input to the encoder. Based on the block diagram in Fig. 1, we observe that the input to the first register becomes $r_0=1$ and the parity check bit $y_0^z=1$ is produced. At the end of the time step, the

content of each register is shifted to the successive register and the memory state changes from $s_0=0$ at $t=0$ to $s_1=2^{\nu-1}$ at $t=1$, as is illustrated in the trellis diagram of Fig. 2.

The dominant CWEF of the RSC code, namely $B_2^c(U, Z)$, only considers weight-2 input sequences, which correspond to paths in the trellis diagram that diverge from the zero state and re-merge with it only once after a number of time steps. It is important to note that as long as a trail of zeroes follows the first nonzero input bit, the value of r_0 depends solely on the feedback generator vector G_R and the bits stored in the registers, hence the RSC encoder behaves like a pseudo-random number generator. According to [19], pseudo-random number generators using primitive generator polynomials periodically revisit all distinct states, except the zero state. Hence the period of the polynomial, denoted as L , is given by [19]

$$L = 2^\nu - 1. \quad (11)$$

The part of the path of an information weight-2 code sequence which was generated when the RSC encoder was operating as a pseudo-random number generator, is depicted by a dashed line in the trellis diagram shown in Fig. 2.

Due to the memory state periodicity, the initial state of the pseudo-random number generator, $s_1=2^{\nu-1}$, is also repeated every L time steps, i.e., $s_{kL+1}=s_1$, where k is a positive integer that satisfies the condition $kL+1 \leq N$; therefore the value of k can range from 1 to $\lfloor (N-1)/L \rfloor$, where the floor function $\lfloor \xi \rfloor$ gives the largest integer that does not exceed ξ . When the encoder is in state $s_{kL+1}=2^{\nu-1}$, the first register is set, i.e., $r_1=1$, while all other registers are clear, i.e., $r_2=\dots=r_\nu=0$. Bearing in mind that the input bit is zero, the bits stored in the registers at the immediately preceding time step should have been $r_1=\dots=r_{\nu-1}=0$ and $r_\nu=1$. Consequently, the state previous to $s_{kL+1}=2^{\nu-1}$ is always $s_{kL}=1$. Note that during the state transition, the logical value of both r_0 and r_ν is “1”, thus the generated parity check bit is $y_{kL}^z=0$ according to the schematic in Fig. 1. The state transition from 1 to $2^{\nu-1}$ as well as the associated input and output bits are depicted in Fig. 2.

The RSC encoder stops behaving like a pseudo-random number generator only when the second nonzero bit of the weight-2 information sequence is input to the encoder forcing a return to the zero state. To achieve this, the second nonzero bit can be input to the encoder only when a particular state s_t , that precedes the zero state $s_{t+1}=0$, is reached. Based on the block diagram in Fig. 1, we observe that when $x_t=1$, the state of the encoder goes to zero only if the preceding

state is $s_t=1$. Only then is the logical “1”, stored in r_ν , canceled out with the input bit, giving $r_0=0$ and, consequently, forcing the encoder to return to the zero state. We demonstrated in the previous paragraph that $s_{kL}=1$. Therefore, when the second nonzero bit $x_{kL}=1$ is input to the encoder, the parity check bit $y_{kL}^z=1$ is generated while the memory state changes to $s_{kL+1}=0$, as is illustrated in Fig. 2.

Based on the trellis structure of an RSC code, it is straightforward to identify the properties of code sequences, generated by weight-2 information sequences, in order to compute the dominant CWEF, namely $B_2^c(U, Z)$. We observe in Fig. 2 that the part of the trellis path which was generated while the RSC encoder was operating as a pseudo-random number generator can be obtained by the alternate succession of a path starting from state $2^{\nu-1}$ and ending in state 1, and a branch connecting states 1 and $2^{\nu-1}$. The parity check stream generated during the sequence of transitions from state $2^{\nu-1}$ to state 1 has a constant weight, which we denote as z_{core} , whilst the parity check bit y_{kL}^z generated during the transition from state 1 to state $2^{\nu-1}$ is zero, as we have previously explained. Consequently, as we can deduce from Fig. 2, the trail of zeroes between the two nonzero information bits in the weight-2 input sequence causes the encoder to generate a stream of parity check bits having weight kz_{core} . However, we have also previously demonstrated that each of the two nonzero information bits causes the encoder to output a nonzero parity check bit. Therefore, the overall weight of the parity check sequence generated by a weight-2 information sequence is given by

$$z(k) = kz_{\text{core}} + 2. \quad (12)$$

The weight of the systematic sequence is identical to the weight of the information sequence, since we are considering a nonpunctured code, hence

$$u(k) = 2. \quad (13)$$

If the weight-2 information sequence has total length N , the corresponding trellis path will also have length N . Depending on the positions of the two nonzero bits in the information sequence, the trellis path could diverge from the all-zero path at *any* time step and re-merge

with it $kL + 1$ time steps later¹. In particular, we find that there are

$$B_k = N - (kL + 1) + 1 = N - kL \quad (14)$$

possible time steps, thus there are B_k code sequences of the form $W^2 U^{u(k)} Z^{z(k)}$. The dominant CWEF, $B_2^C(U, Z)$, is the sum of all code sequences having weights $u(k)$ and $z(k)$, over all valid values of k , hence

$$\begin{aligned} B_2^C(U, Z) &= \sum_k B_k U^{u(k)} Z^{z(k)} \\ &= \sum_{k=1}^{\lfloor (N-1)/L \rfloor} (N - kL) U^2 Z^{kz_{\text{core}}+2}. \end{aligned} \quad (15)$$

C. Punctured Constituent Codes

Let $\mathbf{P}_U = [p_{1,1} \ p_{1,2} \ \dots \ p_{1,M}]$ and $\mathbf{P}_Z = [p_{2,1} \ p_{2,2} \ \dots \ p_{2,M}]$ be the row vectors used to puncture the systematic output and the parity check output, respectively, of a constituent rate-1/2 RSC encoder. Note that vectors \mathbf{P}_U and \mathbf{P}_Z are circularly repeated every M time steps, in such a way that $p_{1,m+jM} = p_{1,m}$ and $p_{2,m+jM} = p_{2,m}$, where $m = 1, \dots, M$ and j is a positive integer. In order to compute the dominant CWEF, $B_2^C(U, Z)$, of the punctured RSC encoder, we need to express the weights of its two output sequences as functions of the puncturing elements, $p_{1,m}$ and $p_{2,m}$, and the output bits, y_t^u and y_t^z , of the parent rate-1/2 RSC encoder.

In particular, we note that if the m -th puncturing elements are active at time step t , the systematic and parity check weights of the output bits of the punctured RSC encoder will be $y_t^u \cdot p_{1,m}$ and $y_t^z \cdot p_{2,m}$, respectively. Although input sequences with information weight $w = 2$ generate paths of length $kL + 1 \leq N$, as we have previously shown, we will first consider paths of length $L + 1$, i.e., $k = 1$, for simplicity. The systematic weight $u(k=1, m)$ of a code sequence, whose path diverges from the zero state when $p_{1,m}$ is active, is given by

$$u(k=1, m) = \sum_{t=0}^L (y_t^u \cdot p_{1,m+t}), \quad (16)$$

which reduces to

$$u(k=1, m) = p_{1,m} + p_{1,m+L}, \quad (17)$$

¹The problem of enumerating all possible time steps at which a path can diverge from the all-zero path of length N and re-merge with it $kL + 1$ time steps later is identical to the problem of enumerating all possible ways that $kL + 1$ balls of the same color can be placed successively in N slots.

since $y_t^u = x_t$ and the two only nonzero information bits occur at the very beginning and at the very end of the path, i.e., $x_t = 1$ for $t = 0$ and $t = L$, otherwise $x_t = 0$.

Similarly, the weight $z(k=1, m)$ of the parity check sequence, whose path diverges from the zero state when $p_{2,m}$ is active, assumes the form

$$z(k=1, m) = \sum_{t=0}^L (y_t^z \cdot p_{2,m+t}). \quad (18)$$

Although z_{core} has a fixed value when a nonpunctured code is considered, a puncturing pattern of period M creates M variants of z_{core} , denoted as $z_{\text{core}}^1, \dots, z_{\text{core}}^m, \dots, z_{\text{core}}^M$, where index m indicates that the m -th column of the puncturing pattern is active at time step $t = 1$, when the RSC encoder starts behaving as a pseudo-random number generator. Consequently, the weight $z(k=1, m)$ of the parity check sequence can be written as

$$z(k=1, m) = p_{2,m} + z_{\text{core}}^{m+1} + p_{2,m+L}, \quad (19)$$

since the parity check bits y_0^z and y_L^z are nonzero, as illustrated in Fig. 3. In order to calculate $z(k=1, m)$ for every value of m , we first need to derive the M variants of z_{core} by applying the M circularly shifted versions of the puncturing vector \mathbf{P}_Z to the corresponding output parity check bits of the parent rate-1/2 RSC encoder, i.e.,

$$z_{\text{core}}^m = \sum_{t=1}^{L-1} (y_t^z \cdot p_{2,m+t-1}). \quad (20)$$

If we extend our analysis to code sequences associated with paths of length $kL+1$, we observe that the systematic weight $u(k, m)$ still depends on the puncturing elements which are active at the very beginning and the very end of the input sequence, hence

$$u(k, m) = p_{1,m} + p_{1,m+kL}, \quad (21)$$

for $k = 1, \dots, \lfloor (N-1)/L \rfloor$ and $m = 1, \dots, M$. In order to compute the parity check weight $z(k, m)$ of a sequence that corresponds to a path of length $kL+1$, we need to consider the weights of the k consecutive parity check streams comprising the parity check output sequence. Therefore, $z(k, m)$ can be expressed as

$$z(k, m) = p_{2,m} + \sum_{t=0}^{k-1} z_{\text{core}}^{m+tL+1} + p_{2,m+kL}. \quad (22)$$

In the previous subsection we demonstrated that a weight-2 information sequence of length N , which forces the encoder to return to the zero state, generates one of a total of $N - kL$

possible code sequences. When a puncturing pattern of period M is used to increase the rate of the code, the $N - kL$ code sequences of the form $W^2U^2Z^{z(k)}$ are mapped to $N - kL$ punctured code sequences of the form $W^2U^{u(k,m)}Z^{z(k,m)}$. Consequently, if $B_{k,m}$ denotes the number of punctured code sequences whose trellis paths diverge from the all-zero path when the m -th column of the puncturing pattern is active, the sum of $B_{k,m}$ over all values of m should give

$$\sum_{m=1}^M B_{k,m} = N - kL. \quad (23)$$

Owing to the puncturing period M , the $N - kL$ punctured code sequences can be divided into M subsets; for a given k , the m -th subset comprises code sequences having the same weights $u(k, m)$ and $z(k, m)$. Thus, the number of code sequences $B_{k,m}$ in the m -th subset is $(N - kL)/M$, if M exactly divides $N - kL$. In general, we find that

$$B_{k,m} = \begin{cases} \left\lfloor \frac{N - kL}{M} \right\rfloor, & \text{if } ((N - kL) \bmod M) < m \\ \left\lfloor \frac{N - kL}{M} \right\rfloor + 1, & \text{otherwise,} \end{cases} \quad (24)$$

where $(\xi_1 \bmod \xi_2)$ denotes the remainder of division of ξ_1 by ξ_2 .

Therefore, the dominant CWEF, $B_2^C(U, Z)$, of a punctured RSC code can be computed using

$$B_2^C(U, Z) = \sum_{k=1}^{\lfloor (N-1)/L \rfloor} \sum_{m=1}^M B_{k,m} U^{u(k,m)} Z^{z(k,m)}, \quad (25)$$

where quantities $u(k, m)$, $z(k, m)$ and $B_{k,m}$ have been defined in (21), (22) and (24), respectively.

D. Numerical Results

Given the two constituent RSC codes \mathcal{C}_1 and \mathcal{C}_2 of a turbo code \mathcal{P} , we can use (15) or (25), depending on whether puncturing is applied or not, to calculate $B_2^{\mathcal{C}_1}(U, Z)$ and $B_2^{\mathcal{C}_2}(U, Z)$. Consequently, the dominant CWEF, $B_2^{\mathcal{P}}(U, Z)$, of the turbo code can be obtained from (5) under the assumption that a uniform interleaver of size N is used. Knowledge of the dominant CWEF of \mathcal{P} allows us to compute the probability $P(2)$, which is a good approximation to the union bound P_b^u when long interleavers are used.

In Fig. 4, we compare the union bound P_b^u with its approximation $P(2)$, for various turbo code configurations. In all cases, the parent code is the rate-1/3 PCCC(1,5/7,5/7). As expected, $P(2)$ deviates from P_b^u when interleavers of size $N = 100$ are considered. However, for $N = 1000$, the approximation closely matches the union bound at low to medium E_b/N_0 values, while for

$N = 10000$, $P(2)$ is a good bound approximation over a wider range of E_b/N_0 values. If we compare Fig. 4(a) with Fig. 4(b), we also note the effect of puncturing the systematic bits; for the same interleaver size, $P(2)$ is closer to P_b^u when all systematic bits are transmitted. In the complete absence of systematic information, the closeness of $P(2)$ to the union bound deteriorates, particularly at high E_b/N_0 values; nevertheless, $P(2)$ becomes the most significant contribution over all E_b/N_0 values as $N \rightarrow \infty$ [18].

V. PSEUDO-RANDOM PUNCTURING

So far, we have introduced a rapid method to compute the dominant CWEFs of the constituent RSC codes of a PCCC, which can be used to obtain an accurate bound approximation on its BEP, or equivalently, to gain insight into its performance in the error floor region. Therefore, we can compute bound approximations for turbo codes of a given rate (higher than 1/3) which are obtainable by puncturing a parent rate-1/3 PCCC using various patterns of a specific period. We can then identify optimal puncturing patterns, i.e., patterns that minimize the BEP predicted by the proposed union bound approximation.

As an example, Fig. 5 shows bound approximations for three punctured PCCCs, one from each class (i.e., S, PS and NS). The corresponding puncturing patterns are presented in Table I. For comparison, the bound approximation for the parent rate-1/3 PCCC(1,5/7,5/7) has also been plotted. We observe that the parent rate-1/3 PCCC performs better than the rate-1/2 S-PCCC but both PS and NS codes exhibit lower bound approximations than that of their parent code. Even though the NS scheme achieves the lowest bound approximation, it has been demonstrated that very long interleavers ($N = 10^6$) and an impractical number of decoding iterations are required to drive the BEP performance of NS-PCCCs into the error floor region [20]. On the other hand, the performance of iteratively decoded PS-PCCCs converges to low BEPs for practical interleaver sizes ($N \approx 1000$) after 8 iterations [20]; nevertheless, PS-PCCCs yielding low BEPs can be found only by means of an exhaustive search, for a given parent code. Consequently, it is important to conceive alternative methods to construct good punctured PCCCs.

In this section, we introduce a special case of a rate-1/2 PS-PCCC, which we call a pseudo-randomly punctured PCCC (PRP-PCCC) [18]. The bound approximation and the puncturing pattern of a PRP-PCCC for the above example are also presented in Fig. 5 and Table I, respectively. We will use performance criteria to demonstrate that a rate-1/2 PRP-PCCC, which

(contrary to good PS-PCCCs) can be obtained in a straightforward manner from a parent turbo code, will always exhibit a lower error floor than that of its parent rate-1/3 PCCC.

A. Criteria for Performance Comparison

We will now use the union bound approximation as the basis to compare by analysis the performance of turbo codes. In particular, if \mathcal{P} and \mathcal{P}' are two PCCCs using long interleavers of identical size, we say that \mathcal{P} yields a lower error floor than that of \mathcal{P}' when their bound approximations, $P^{\mathcal{P}}(2)$ and $P^{\mathcal{P}'}(2)$, respectively, satisfy $P^{\mathcal{P}}(2) < P^{\mathcal{P}'}(2)$. This condition can also be expressed using (10) as follows

$$\sum_d B_{2,d}^{\mathcal{P}} Q \left(\sqrt{\frac{2R^{\mathcal{P}} E_b}{N_0} d} \right) < \sum_d B_{2,d}^{\mathcal{P}'} Q \left(\sqrt{\frac{2R^{\mathcal{P}'} E_b}{N_0} d} \right). \quad (26)$$

It was demonstrated in [17] and [21] that the effective free distance, d_{ef} , which conveys the minimum weight of a code sequence for a weight-2 input information sequence, has a major impact on the performance of a turbo code in the error floor region. Consequently, if $d_{\text{ef}}^{\mathcal{P}}$ and $d_{\text{ef}}^{\mathcal{P}'}$ denote the effective free distances of \mathcal{P} and \mathcal{P}' respectively, condition (26) collapses to

$$B_{2,d_{\text{ef}}}^{\mathcal{P}} Q \left(\sqrt{\frac{2R^{\mathcal{P}} E_b}{N_0} d_{\text{ef}}^{\mathcal{P}}} \right) < B_{2,d_{\text{ef}}}^{\mathcal{P}'} Q \left(\sqrt{\frac{2R^{\mathcal{P}'} E_b}{N_0} d_{\text{ef}}^{\mathcal{P}'}} \right), \quad (27)$$

which only considers the first nonzero, that is the most significant, term of each sum.

Function $Q(\xi)$ is a monotonically decreasing function of ξ , where ξ is a real number. Therefore, if ξ_1 and ξ_2 are real numbers, with $\xi_1 > \xi_2$, we deduce that $Q(\xi_1) < Q(\xi_2)$, and vice versa. Consequently, inequality (27) holds true if

$$R^{\mathcal{P}} d_{\text{ef}}^{\mathcal{P}} > R^{\mathcal{P}'} d_{\text{ef}}^{\mathcal{P}'}, \quad (28)$$

$$B_{2,d_{\text{ef}}}^{\mathcal{P}} \leq B_{2,d_{\text{ef}}}^{\mathcal{P}'}. \quad (29)$$

Note that when the code rates are equal, the effective free distance of a turbo code plays a role similar to that of the free distance of a convolutional code, since (28) simplifies to $d_{\text{ef}}^{\mathcal{P}} > d_{\text{ef}}^{\mathcal{P}'}$.

Hence, if our objective is to obtain turbo codes of any rate yielding low error floors, the product of the code rate and the effective free distance should be maximized, whilst the multiplicity of information weight-2 code sequences should be minimized. However, it is important to note that puncturing influences the convergence behavior of a turbo code [20]; thus we also need to investigate whether the performance of the suboptimal iterative decoder eventually converges to the error floor region at high E_b/N_0 values, using analysis tools such as EXIT charts [22].

B. Rate-1/3 PCCCs

Criteria (28) and (29) require knowledge of the effective free distance d_{ef} and the coefficient $B_{2,d_{\text{ef}}}$ of each PCCC. In the remainder of the paper, we use the abbreviation ‘‘Par’’ to denote a rate-1/3 parent PCCC. Its effective free distance $d_{\text{ef}}^{\text{Par}}$ can be expressed as the sum of the minimum weight $d_{\text{min}}^{\mathcal{C}_1}$ of the code sequence generated by the upper constituent encoder, and the minimum weight $z_{\text{min}}^{\mathcal{C}_2}$ of the parity check sequence generated by the lower constituent encoder, when a sequence of information weight $w=2$ is input to the PCCC, i.e.,

$$\begin{aligned} d_{\text{ef}}^{\text{Par}} &= d_{\text{min}}^{\mathcal{C}_1} + z_{\text{min}}^{\mathcal{C}_2} \\ &= \min_k \left(u^{\mathcal{C}_1}(k) + z^{\mathcal{C}_1}(k) \right) + \min_k \left(z^{\mathcal{C}_2}(k) \right). \end{aligned} \quad (30)$$

The parent turbo code is taken to be symmetric, thus $z^{\mathcal{C}_1}(k) = z^{\mathcal{C}_2}(k) = z(k)$. Furthermore, a weight-2 information sequence generates an identical weight-2 systematic sequence, hence $u^{\mathcal{C}_1}(k) = 2$ for every value of k . Consequently, we can write

$$\begin{aligned} d_{\text{ef}}^{\text{Par}} &= 2 + 2 \min_k (z(k)) \\ &= 2 + 2z(k=1) \\ &= 6 + 2z_{\text{core}}, \end{aligned} \quad (31)$$

since $z(k=1) = z_{\text{core}} + 2$. However, when the generator polynomials of the constituent RSC code are selected such that $G_R \neq G_F$ and G_R is primitive, it can be shown that z_{core} achieves its maximum value [17], [19], that is $z_{\text{core}} = 2^{\nu-1}$, thus

$$d_{\text{ef}}^{\text{Par}} = 6 + 2^{\nu}. \quad (32)$$

The number $B_{2,d_{\text{ef}}}^{\text{Par}}$ of code sequences having weight $d_{\text{ef}}^{\text{Par}}$ can be associated with the number $B_{2,d_{\text{min}}}^{\mathcal{C}_1}$ of code sequences having weight $d_{\text{min}}^{\mathcal{C}_1}$ and the number $B_{2,z_{\text{min}}}^{\mathcal{C}_2}$ of parity check sequences having weight $z_{\text{min}}^{\mathcal{C}_2}$, if we elaborate on (5). In particular, we obtain

$$B_{2,d_{\text{ef}}}^{\text{Par}} = \frac{B_{2,d_{\text{min}}}^{\mathcal{C}_1} \cdot B_{2,z_{\text{min}}}^{\mathcal{C}_2}}{\binom{N}{2}}, \quad (33)$$

if a uniform interleaver of size N is assumed, where $B_{2,d_{\text{min}}}^{\mathcal{C}_1}$ and $B_{2,z_{\text{min}}}^{\mathcal{C}_2}$ return the same value since they both consider the same trellis paths of minimum weight $z(k=1)$. Based on (14), we

find that for $k = 1$ the number of trellis paths, or equivalently the number of code sequences, having minimum parity check weight is

$$B_{2,d_{\min}}^{C_1} = B_{2,z_{\min}}^{C_2} = N - L, \quad (34)$$

where $L = 2^\nu - 1$ is the period of the primitive feedback generator polynomial. In the convenient case when the size N of the interleaver is an integer multiple of the period L , i.e., $N = \mu L$, coefficient $B_{2,d_{\text{ef}}}^{\text{Par}}$ assumes the form

$$B_{2,d_{\text{ef}}}^{\text{Par}} = \frac{2L(\mu - 1)^2}{\mu(\mu L - 1)}. \quad (35)$$

C. Pseudo-randomly Punctured Rate-1/2 PCCCs

In the case of rate-1/2 PS-PCCCs, the effective free distance and the multiplicity of code sequences having information weight $w = 2$ can only be derived using the semi-analytical expressions (21), (22) and (24). Here, we demonstrate that it is possible to express the above quantities in a more analytically tractable form, if we properly design the puncturing pattern.

Let us consider again a rate-1/3 PCCC that uses primitive feedback generator polynomials of period L . As previously explained in Section IV-B, an input sequence $1, 0, \dots, 0$ of length $L+1$ will cause the upper constituent RSC encoder to generate a parity check sequence of the form $1, y_1^z, \dots, y_L^z$. If a puncturing pattern \mathbf{P} of period $M = L$, which has the following structure

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{U_1} \\ \mathbf{P}_{Z_1} \\ \mathbf{P}_{Z_2} \end{bmatrix} = \begin{bmatrix} 1 - y_L^z & 1 - y_1^z & \dots & 1 - y_{L-1}^z \\ y_L^z & y_1^z & \dots & y_{L-1}^z \\ 1 & 1 & \dots & 1 \end{bmatrix}, \quad (36)$$

is used to reduce the rate of the PCCC from 1/3 to 1/2, we refer to this type of PS-PCCC as a pseudo-randomly punctured PCCC [18]. For an example case of PRP-PCCC, see Table I.

Let us now consider the upper constituent punctured RSC code C'_1 ; we observe that (21) and (22) can be further reduced if we take into account $M = L$ and $p_{1,m} = 1 - p_{2,m}$. In particular,

$$u^{C'_1}(k, m) = 2p_{1,m} = 2 - 2p_{2,m}, \quad (37)$$

$$z^{C'_1}(k, m) = kz_{\text{core}}^{m+1} + 2p_{2,m}, \quad (38)$$

respectively. Note that $p_{j,m+kM} = p_{j,m}$ for $j = 1, 2$ and $z_{\text{core}}^{m+tM} = z_{\text{core}}^m$, due to the periodicity of the puncturing pattern. Consequently, the minimum weight $d_{\min}^{C'_1}$ of the code sequence generated

by the upper constituent encoder is given by

$$\begin{aligned} d_{\min}^{\mathcal{C}'_1} &= \min_{k,m} \left(u^{\mathcal{C}'_1}(k, m) + z^{\mathcal{C}'_1}(k, m) \right) \\ &= \min_{k,m} \left(k z_{\text{core}}^{m+1} + 2 \right). \end{aligned} \quad (39)$$

Exploiting the properties of pseudo-random sequences [19], [23], we can show that

$$z_{\text{core}}^{m+1} = \begin{cases} 2^{\nu-1}, & \text{if } m = 1 \\ 2^{\nu-2}, & \text{if } 2 \leq m \leq M. \end{cases} \quad (40)$$

A detailed proof is presented in the Appendix. Thus, we can rewrite (39) as follows

$$d_{\min}^{\mathcal{C}'_1} = 2 + 2^{\nu-2}. \quad (41)$$

The parity check sequence of the lower constituent code \mathcal{C}'_2 is not punctured, since $p_{3,m} = 1$ for every value of m . Therefore, the minimum weight $z_{\min}^{\mathcal{C}'_2}$ can be obtained using the expressions for nonpunctured RSC codes, i.e.,

$$\begin{aligned} z_{\min}^{\mathcal{C}'_2} &= \min_k \left(z^{\mathcal{C}'_2}(k) \right) \\ &= z_{\text{core}} + 2 \\ &= 2 + 2^{\nu-1}. \end{aligned} \quad (42)$$

In a similar fashion to rate-1/3 PCCCs, the effective free distance $d_{\text{ef}}^{\text{PRP}}$ of the rate-1/2 PRP-PCCC can be expressed as

$$\begin{aligned} d_{\text{ef}}^{\text{PRP}} &= d_{\min}^{\mathcal{C}'_1} + z_{\min}^{\mathcal{C}'_2} \\ &= 4 + 3(2^{\nu-2}). \end{aligned} \quad (43)$$

In order to facilitate our analysis, we now assume that the interleaver size N is an integer multiple of the puncturing period M , i.e., $N = \mu M$, where μ is a positive integer. Based on (24), we find that for $k=1$, $(\mu-1)$ code sequences can be generated for each value of $m \in [1, M]$. We observe however, that $z_{\text{core}}^{m+1} = 2^{\nu-2}$ only when $m \in [2, M]$; since m can assume $(M-1)$ integer values in the range between 2 and M , there are $(M-1)$ code sequences per puncturing period M that achieve a weight equal to $d_{\min}^{\mathcal{C}'_1}$. Consequently, we deduce that the total number of minimum-weight code sequences obtained at the output of the upper constituent encoder during the N time steps of the coding process, assumes the value

$$\begin{aligned} B_{2,d_{\min}^{\mathcal{C}'_1}} &= (\mu - 1)(M - 1) \\ &= (\mu - 1)(L - 1), \end{aligned} \quad (44)$$

since M and L are equal quantities and can be used interchangeably.

The lower constituent RSC encoder of the rate-1/2 PRP-PCCC generates exactly the same number of sequences having minimum parity check weight as that of the rate-1/3 PCCC, since the parity check output of both constituent codes is not punctured. Therefore, we obtain

$$B_{2,z'_{\min}}^{C'_2} = N - L. \quad (45)$$

The coefficient $B_{2,d_{\text{ef}}}^{\text{PRP}}$ of a rate-1/2 PRP-PCCC can be expressed as a function of $B_{2,d'_{\min}}^{C'_1}$ and $B_{2,z'_{\min}}^{C'_2}$, in a similar manner to (33). Consequently, using (44), (45) and taking into account that $N = \mu M$ and $M = L$, we find that

$$B_{2,d_{\text{ef}}}^{\text{PRP}} = \frac{2(L-1)(\mu-1)^2}{\mu(\mu L-1)}. \quad (46)$$

D. Analytical and Numerical Results

We are now in the position to explore whether a rate-1/2 PRP-PCCC exhibits a lower bound approximation than that of its rate-1/3 parent PCCC. We observe that $d_{\text{ef}}^{\text{PRP}}$ can be expressed in terms of $d_{\text{ef}}^{\text{Par}}$, if we subtract (32) from (43)

$$d_{\text{ef}}^{\text{PRP}} = d_{\text{ef}}^{\text{Par}} - (2 + 2^{\nu-2}). \quad (47)$$

Coefficient $B_{2,d_{\text{ef}}}^{\text{PRP}}$ can also be represented in terms of $B_{2,d_{\text{ef}}}^{\text{Par}}$, if we divide (46) by (35)

$$B_{2,d_{\text{ef}}}^{\text{PRP}} = \left(\frac{L-1}{L} \right) B_{2,d_{\text{ef}}}^{\text{Par}}. \quad (48)$$

According to (28) and (29), if both conditions

$$\frac{1}{2}d_{\text{ef}}^{\text{PRP}} > \frac{1}{3}d_{\text{ef}}^{\text{Par}}, \quad (49)$$

$$B_{2,d_{\text{ef}}}^{\text{PRP}} \leq B_{2,d_{\text{ef}}}^{\text{Par}} \quad (50)$$

are satisfied, a rate-1/2 PRP-PCCC yields a lower bound approximation than that of its rate-1/3 parent code. We deduce from (48) that $B_{2,d_{\text{ef}}}^{\text{PRP}}$ is always less than $B_{2,d_{\text{ef}}}^{\text{Par}}$, thus the second condition holds true. If we combine (47) with (49), the first condition assumes the following form

$$3 \left(d_{\text{ef}}^{\text{Par}} - (2 + 2^{\nu-2}) \right) > 2d_{\text{ef}}^{\text{Par}}, \quad (51)$$

which collapses to

$$d_{\text{ef}}^{\text{Par}} > 6 + 3(2^{\nu-2}). \quad (52)$$

Nevertheless, we have shown in (32) that the effective free distance of the parent PCCC is given by $d_{\text{ef}}^{\text{Par}} = 6 + 2^\nu$, which can be rewritten as $d_{\text{ef}}^{\text{Par}} = 6 + 4(2^{\nu-2})$. Therefore, $d_{\text{ef}}^{\text{Par}}$ is always greater than $6 + 3(2^{\nu-2})$, and hence, both conditions are satisfied. The outcome of this investigation reveals that rate-1/2 PRP-PCCCs using long interleavers are expected to always yield a lower bound approximation and, presumably, a lower error floor than that of their rate-1/3 parent codes.

Fig. 6 compares bound approximations to simulation results for punctured rate-1/2 PCCCs and their parent rate-1/3 PCCC(1,5/7,5/7), over the AWGN channel. The component decoders employ the conventional exact log-MAP algorithm [24]. A moderate interleaver size of $N = 1,000$ has been chosen, so as to allow the performance curves of the PCCCs to approach the corresponding bound approximations at BEPs in the region of 10^{-6} . As expected [20], iterative decoding of the rate-1/2 NS-PCCC hits an error floor, which is much higher than that defined by the bound approximation. On the other hand, transmission of all systematic bits helps the iterative decoder converge to the error floor region at low E_b/N_0 ratios. However, the rate-1/2 S-PCCC exhibits a higher error floor than that of its parent rate-1/3 PCCC. Most importantly, however, Fig. 6 confirms that the performance of the iteratively decoded rate-1/2 PRP-PCCC approaches the bound approximation and, for E_b/N_0 values greater than 2.5 dB, the BEP of the rate-1/2 PRP-PCCC is indeed lower than that of the parent rate-1/3 PCCC. Hence, a rate-1/2 turbo code obtained by using pseudo-random puncturing, outperforms its own parent rate-1/3 PCCC, when they both operate in the error floor region.

VI. CONCLUSIONS

In this paper we introduce a rapid method, which exploits the properties of constituent RSC encoders, to obtain their dominant CWEFs, which only consider code sequences having information weight equal to two. This is in contrast to their transfer functions, which enumerate all code sequences having nonzero information weight. Using the dominant CWEFs, we are able obtain union bound approximations that are accurate when long interleavers are employed and, hence, can give us insight into the performance of turbo codes operating in the error floor region. We exploit this technique to establish that pseudo-randomly punctured rate-1/2 PCCCs, which form a subset of partially systematic PCCCs, not only approach the error floor region for an increasing number of iterations but always yield a lower error floor than that of their rate-1/3 parent codes. Consequently, pseudo-random puncturing can be used to reduce the rate of a PCCC

from 1/3 to 1/2 and at the same time achieve a coding gain at low bit error probabilities.

APPENDIX

CALCULATION OF z_{CORE} IN PSEUDO-RANDOMLY PUNCTURED CONVOLUTIONAL CODES

Let y_t^z be the parity check bit generated at time step t by an RSC encoder using a primitive feedback polynomial of degree ν and period $L = 2^\nu - 1$. If y_1^z, \dots, y_L^z is a sequence of parity check bits, the autocorrelation function of the equivalent bipolar sequence $(2y_1^z - 1), \dots, (2y_L^z - 1)$ is given by

$$\phi(i) = \sum_{t=1}^L (2y_t^z - 1)(2y_{t+i}^z - 1) \quad (53)$$

where i is a non-negative integer with $0 \leq i < L$. In the event that the binary parity check sequence is a pseudo-noise (PN) sequence, i.e., it is generated when the RSC encoder operates as a pseudo-random number generator, the autocorrelation function $\phi(i)$ reduces to [19], [23]

$$\phi(i) = \begin{cases} 2^\nu - 1, & \text{if } i = 0 \\ -1, & \text{if } 1 \leq i < L. \end{cases} \quad (54)$$

If we expand (53), we obtain

$$\begin{aligned} \phi(i) &= 4 \sum_{t=1}^L y_t^z y_{t+i}^z - 2 \sum_{t=1}^L y_t^z - 2 \sum_{t=1}^L y_{t+i}^z + L \\ &= 4 \sum_{t=1}^L y_t^z y_{t+i}^z - 2^\nu - 1, \end{aligned} \quad (55)$$

where

$$\sum_{t=1}^L y_{t+i}^z = 2^{\nu-1} \quad (56)$$

for $0 \leq i < L$, since there are $2^{\nu-1}$ nonzero bits in a PN sequence of length $L = 2^\nu - 1$ [19].

Combining (54) and (55), we find that

$$\sum_{t=1}^L (y_t^z \cdot y_{t+i}^z) = \begin{cases} 2^{\nu-1}, & \text{if } i = 0 \\ 2^{\nu-2}, & \text{if } 1 \leq i < L. \end{cases} \quad (57)$$

We have demonstrated in par. 4 of Section IV-B that $y_L^z = 0$, subsequently $y_L^z \cdot y_{L+i}^z = 0$. From this observation, we can deduce that the weight of the generated sequence remains unchanged if we take the sum of only the first $L-1$ terms. If pseudo-random puncturing is used to increase the rate of the RSC encoder, the elements of the puncturing vector for the parity check output

satisfy $p_{2,m+1} = y_m^z$ with $m \in [1, M]$ where $M = L$, as we described in Section V. Consequently, if we replace y_{t+i}^z in (57) with its equivalent, $p_{2,t+i+1}$, and substitute $(i+1)$ with m , we obtain

$$\sum_{t=1}^{L-1} (y_t^z \cdot p_{2,m+t}) = \begin{cases} 2^{\nu-1}, & \text{if } m = 1 \\ 2^{\nu-2}, & \text{if } 2 \leq m \leq M. \end{cases} \quad (58)$$

Note that the quantity on the left hand side of (58) corresponds to z_{core}^{m+1} , according to (20).

REFERENCES

- [1] J. Hagenauer, "Rate compatible punctured convolutional codes and their applications," *IEEE Trans. Commun.*, vol. 36, no. 4, pp. 389–400, Apr. 1988.
- [2] D. Haccoun and G. Bégin, "High-rate punctured convolutional codes for Viterbi and sequential decoding," *IEEE Trans. Commun.*, vol. 37, no. 11, pp. 1113–1125, Nov. 1989.
- [3] Ö. Açikel and W. E. Ryan, "Punctured turbo-codes for BPSK/QPSK channels," *IEEE Trans. Commun.*, vol. 47, no. 9, pp. 1315–1323, Sep. 1999.
- [4] F. Babich, G. Montorsi, and F. Vatta, "Design of rate-compatible punctured turbo (RCPT) codes," in *Proc. Int. Conf. Comm.*, New York, USA, Apr. 2002, pp. 1701–1705.
- [5] M. A. Kousa and A. H. Mugaibel, "Puncturing effects on turbo codes," *Proc. IEE Comm.*, vol. 149, no. 3, pp. 132–138, Jun. 2002.
- [6] F. Brännström, L. K. Rasmussen, and A. Grant, "Optimal puncturing for multiple parallel concatenated codes," in *Proc. IEEE International Symposium on Information Theory*, Chicago, USA, Jun. 2004, p. 151.
- [7] M. Fan, S. C. Kwatra, and K. Junghwan, "Analysis of puncturing pattern for high rate turbo codes," in *Proc. Military Comm. Conf.*, New Jersey, USA, Oct. 1999, pp. 547–550.
- [8] I. Land and P. Hoeher, "Partially systematic rate 1/2 turbo codes," in *Proc. Int. Symp. Turbo Codes*, Brest, France, Sep. 2000, pp. 287–290.
- [9] Z. Blazek, V. K. Bhargava, and T. A. Gulliver, "Some results on partially systematic turbo codes," in *Proc. Vehicular Tech. Conf.*, Vancouver, Canada, Sep. 2002, pp. 981–984.
- [10] I. Chatzigeorgiou, M. R. D. Rodrigues, I. J. Wassell, and R. Carrasco, "A novel technique for the evaluation of the transfer function of punctured turbo codes," in *Proc. Intl. Conf. Comm.*, Istanbul, Turkey, Jul. 2006.
- [11] S. Benedetto and G. Montorsi, "Unveiling turbo codes: Some results on parallel concatenated coding schemes," *IEEE Trans. Inf. Theory*, vol. 42, no. 2, pp. 409–429, Mar. 1996.
- [12] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo codes," *IEEE Trans. Commun.*, vol. 44, no. 2, pp. 1261–1271, Oct. 1996.
- [13] M. Breiling, S. Peeters, and J. Huber, "Class of double terminating turbo code interleavers," *Electronics Letters*, vol. 35, no. 5, pp. 389–391, Mar. 1999.
- [14] D. Divsalar, S. Dolinar, R. J. McEliece, and F. Pollara, "Transfer function bounds on the performance of turbo codes," JPL, Cal. Tech., TDA Progr. Rep. 42-121, Aug. 1995.
- [15] E. K. Hall and S. G. Wilson, "Design and analysis of turbo codes on Rayleigh fading channels," *IEEE J. Sel. Areas Commun.*, vol. 16, pp. 160–174, Feb. 1998.

- [16] I. Sason and S. Shamai, *Performance Analysis of Linear Codes under Maximum-Likelihood Decoding: A Tutorial*. Delft, the Netherlands: Foundations and Trends in Communications and Information Theory, NOW Publishers, 2006.
- [17] S. Benedetto and G. Montorsi, "Design of parallel concatenated convolutional codes," *IEEE Trans. Commun.*, vol. 44, no. 5, pp. 591–600, May 1996.
- [18] I. Chatzigeorgiou, M. R. D. Rodrigues, I. J. Wassell, and R. Carrasco, "A union bound approximation for rapid performance evaluation of punctured turbo codes," in *Proc. Conf. on Information Sciences and Systems*, Baltimore, USA, Mar. 2007.
- [19] F. J. MacWilliams and N. J. A. Sloane, "Pseudo-random sequences and arrays," *Proc. IEEE*, vol. 64, no. 12, pp. 1715–1729, Dec. 1976.
- [20] I. Chatzigeorgiou, M. R. D. Rodrigues, I. J. Wassell, and R. Carrasco, "Can punctured rate-1/2 turbo codes achieve a lower error floor than their rate-1/3 parent codes?" in *Proc. IEEE Information Theory Workshop*, Chengdu, China, Oct. 2006.
- [21] D. Divsalar and R. J. McEliece, "Effective free distance of turbo codes," *Electronics Letters*, vol. 32, no. 5, pp. 445–446, Feb. 1996.
- [22] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Trans. Commun.*, vol. 49, no. 10, pp. 1727–1737, Oct. 2001.
- [23] D. V. Sarwate and M. B. Pursley, "Crosscorrelation properties of pseudorandom and related sequences," *Proc. IEEE*, vol. 68, no. 5, pp. 593–618, May 1980.
- [24] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimising symbol error rate," *IEEE Trans. Inf. Theory*, vol. IT-20, pp. 284–287, Mar. 1974.

TABLE I

PUNCTURING PATTERNS FOR THE RATE-1/2 PCCC CONFIGURATIONS IN FIG. 5. THE PARENT CODE IS PCCC(1,5/7,5/7).

S-PCCC	PRP-PCCC	PS-PCCC	NS-PCCC
$\begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$

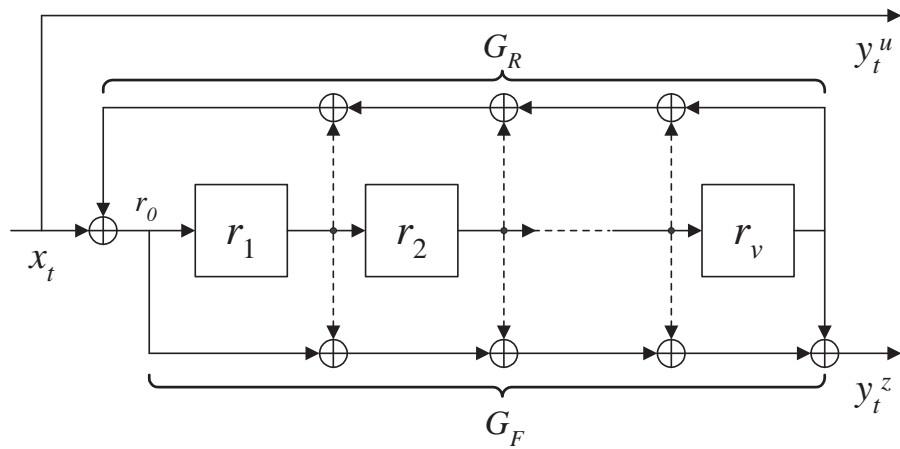


Fig. 1. Block diagram of a rate-1/2 constituent RSC encoder.

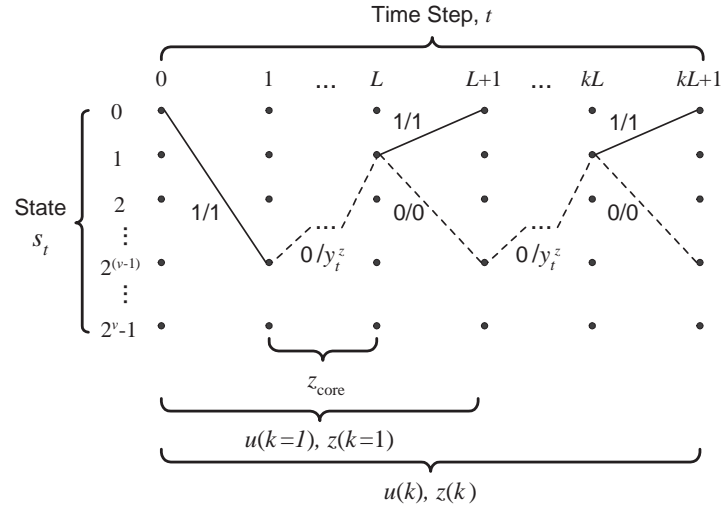


Fig. 2. Trellis diagram for information weight-2 code sequences. The dashed line corresponds to the path generated when the rate-1/2 RSC encoder was operating as a pseudo-random number generator. A pair x_t/y_t^z next to a branch originating from a state at time step t , corresponds to the input bit x_t and the output parity check bit y_t^z generated at the end of the state transition.

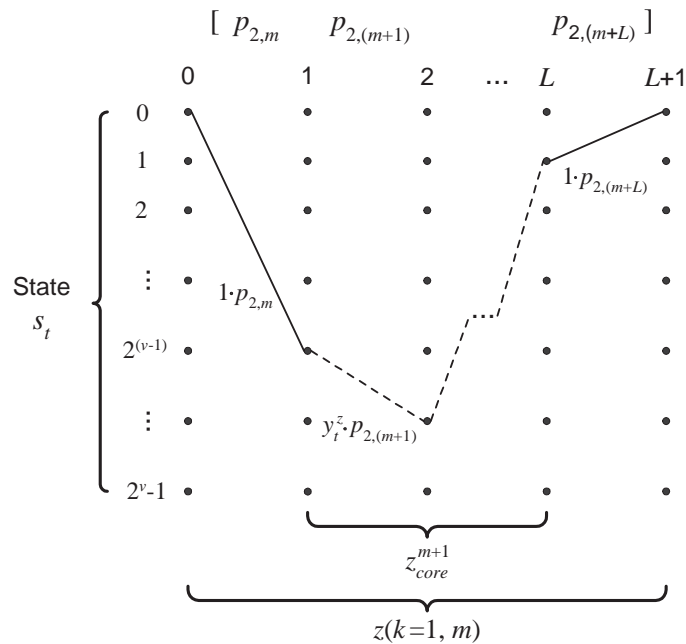
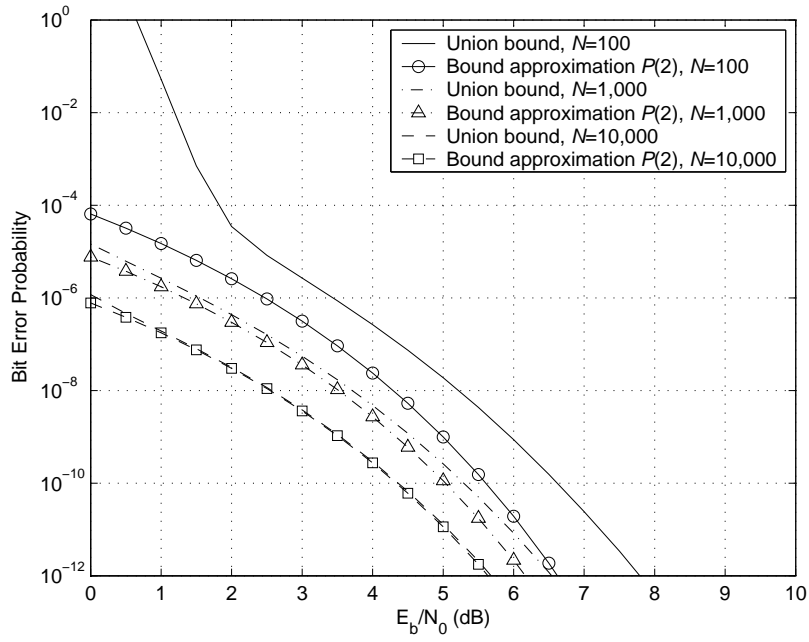
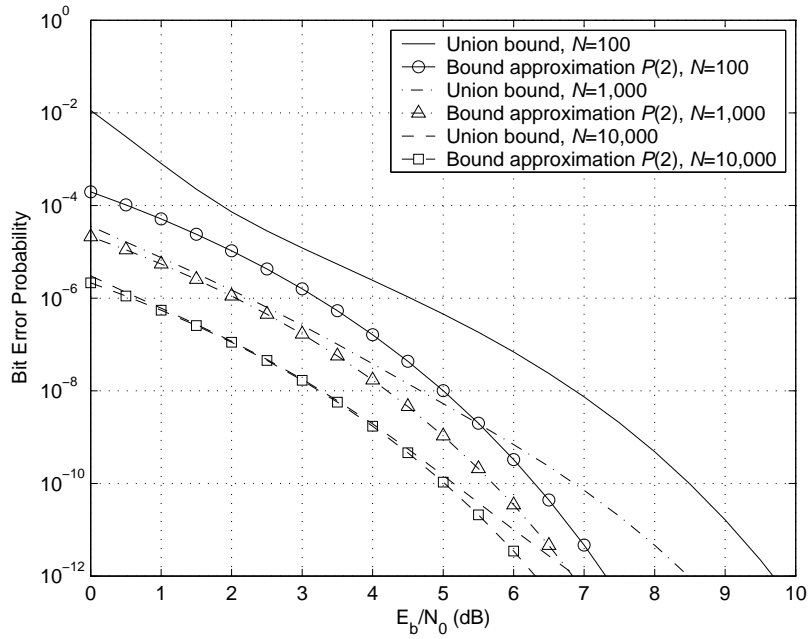


Fig. 3. Trellis diagram for the parity check weight calculation of a punctured RSC code ($k=1$).



(a) Rate-1/2 S-PCCC(1,5/7,5/7)



(b) Rate-1/2 NS-PCCC(1,5/7,5/7)

Fig. 4. Exact union bounds and their approximation for punctured rate-1/2 systematic (S-PCCC) and nonsystematic (NS-PCCC) turbo codes employing an interleaver of size 100, 1000 or 10000.

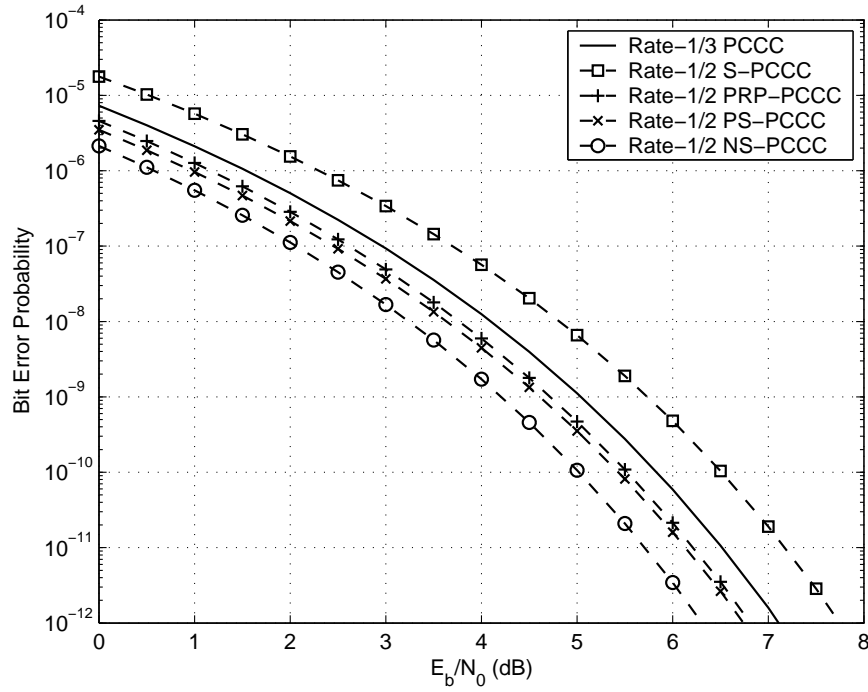


Fig. 5. Bound approximations for the rate-1/3 PCCC(1,5/7,5/7) and various rate-1/2 child configurations. The size of the interleaver is 10,000.

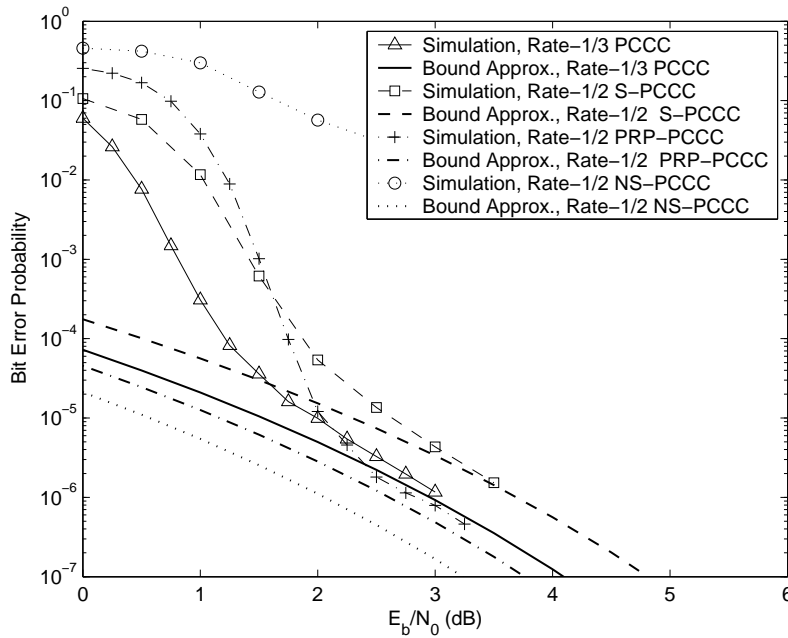


Fig. 6. Comparison of bound approximations to simulation results. The parent code is the rate-1/3 PCCC(1,5/7,5/7). The exact log-MAP algorithm is applied over 8 iterations and an interleaver size of 1000 bits is used.