Some Extended Results on the Design of Punctured Serially Concatenated Convolutional Codes
Massimiliano Laddomada and Bartolo Scanavino

Abstract—The aim of this paper is twofold. On one hand, it presents the results of the search for good punctured systematic recursive convolutional encoders suitable for application in serially concatenated convolutional codes (SCCCs) operating in two different target regions: at low-to-moderate signal-to-noise ratios (SNRs), i.e., in the so-called waterfall region, and at high SNRs. On the other hand, it provides some useful design guidelines for choosing the constituent encoders in an SCCC.

The results of the search for good SCCCs operating in the waterfall region rely upon an effective algorithm, based on density evolution technique, first proposed in a companion paper. Good punctured SCCCs were obtained through considerations deduced by the behaviour of the bit error probability of an SCCC for high values of both SNR and interleaver length, i.e., through asymptotic considerations.

The mother codes in the serial concatenation are rate \( \frac{1}{2} \) recursive convolutional encoders (RCC) found by an exhaustive search for encoders tailored to SCCC schemes, using two different selection criteria. Extensive tables of optimized puncturing patterns for various mother codes and SCCCs are presented along with sample simulation results.

Index Terms—Convolutional codes, high rate, punctured, serially concatenated convolutional codes, turbo codes.

I. INTRODUCTION

The idea of serial concatenation of interleaved codes dates back to 1966 upon a seminal work of Forney [1]. Such codes have been in use for a while in many different applications from deep space to compact discs for the music industry. The recent explosion of research activities has largely surrounded Parallel Concatenated Convolutional Codes (PCCCs) and SCCCs [2]-[4] and iterative soft decoding. A block diagram of a typical SCCC with some useful notation that will be used throughout the paper is depicted in Fig. 1.

In wireless band-limited channels spectral efficiency calls for high-rate channel codes with rate \( \frac{1}{2} \) to limit the bandwidth expansion inherent in channel coding. Cain et al. in [5] first suggested puncturing a mother code of lower rate for obtaining higher rate codes with simplified Viterbi decoding. Since then, many papers have addressed the problem of obtaining optimal punctured convolutional codes, but most of the literature has been devoted to finding optimal puncturing patterns (PPs) for non-recursive convolutional encoders. The recent research activities related to Turbo codes suggest that there is a great need for suitable high-rate recursive convolutional encoders to be used in such applications. Among the various papers in this field, several are of relevance in connection with the current work. The survey of the literature we present in the following is by no means exhaustive and is meant to simply provide a sampling of the literature in this fertile area.

Classical papers [6]-[17] deal with the problem of obtaining optimal PPs for non-recursive convolutional encoders. Another interesting approach to puncturing was introduced by Hagenauer in 1988. In detail, in [18] Hagenauer introduced the so called Rate-Compatible Punctured Convolutional Codes (RPCCCs). In RPCCC, a low rate \( \frac{1}{n} \) mother convolutional code is punctured periodically with period \( P \). RPCCCs are essentially a class of punctured convolutional codes (CCs) with rate \( \frac{1}{n} \), whereby the obtained puncturing table is such that all code bits of higher rate codes are used by the lower rate codes. The concept of RPCCC has been extended in [19]-[21] to both PCCCs and SCCCs. Papers [21]-[25] deal with the design of high-rate SCCCs, while articles [26]-[28] address the design of good constituent codes for turbo codes applications. In particular, in [23] the authors proposed a design algorithm for SCCCs aiming at minimizing the effects of both weight-2 and weight-3 input error events on the SCCC distance spectra. Also related to the current work are papers [29]-[31], in which the authors proposed techniques to analyze the convergence behavior of parallel and serially concatenated convolutional codes.

This paper presents the results of an extensive search for SCCC schemes constituted by punctured encoders optimized for two different SCCC target SNR operating regions, i.e., at low-to-moderate SNRs and at high SNRs.

Before proceeding, let us review the mathematical notation on punctured CCs used throughout the paper. Puncturing is obtained by regularly deleting some output bits of a mother code with rate \( 1/n \). As a result of puncturing, the trellis of the punctured code becomes periodically time-varying. In the case of concatenated schemes, there is no loss of generality if we consider a rate \( 1/2 \) systematic mother encoder, specified by a \( 1 \times 2 \) generator matrix \( G(D) = \begin{bmatrix} 1, g_1(D) \end{bmatrix} \) composed of two polynomials \( g_1(D) \) and \( g_0(D) \) specifying the connections of the finite state encoder. In this formula, \( g_i(D) = g_i^1 + g_i^2 D + \ldots + g_i^\nu D^\nu \), where \( i = 0, 1 \), \( g_i^j \in \{0, 1\} \) and \( \nu = 0, \ldots, \nu \). The variable \( \nu \) specifies the memory of the convolutional encoder (the encoder constraint length is \( 1+\nu \)). The form of the generator matrix \( G(D) \) expresses the fact that the considered encoders are recursive (i.e., we have ratio of polynomials).

The rest of the paper is organized as follows. Section II is devoted to the formulation of the problem and presents the

Manuscript received February 25, 2006; revised March 1, 2006 and August 25, 2006. This work has been partially supported by PRIMO (Piattaforme Riconfigurabili per Interoperabilità in Mobilità) and by progetto PRIN2005 (ICONA).
Authors are with the Dep. of Electronics, Politecnico di Torino, Italy. (e-mail: {laddomada, scanavino}@polito.it).
code search technique we have employed for the design of asymptotically good punctured SCCCs composed by punctured constituent encoders of rate $\frac{k}{k+s}$, for $k=2, \ldots, 8$ and $s=1, \ldots, k-1$. In section III, we deal with the technique for the design of good SCCCs, operating in the low-to-moderate SNR region, designed through an effective algorithm first proposed in a companion paper [32]. In section IV, we present the results of our code search and propose their potential applications to SCCCs. Conclusions are drawn in section V.

II. PROBLEM FORMULATION AND CODE SEARCH TECHNIQUE FOR ASYMPTOTICALLY GOOD SCCCs

This section deals with the design techniques adopted in our search for punctured encoders to be used as outer and inner encoders for asymptotically good SCCCs. For clarity of exposition, we consider the design of good constituent codes for SCCCs separately from the design of the related puncturing patterns.

A. Design of Constituent Encoders for Asymptotically Good SCCCs

In this section, the emphasis is on the design of the constituent encoders in SCCCs.

Based on the asymptotic analysis accomplished in [4], we can state that the inner encoder in an SCCC should be a recursive convolutional encoder (no matter it is systematic or not) in order for the interleaver to yield a coding gain, irrespective of whether it is recursive or not.

The design approach pursued in this work is strictly related to the average asymptotic bit error probability of SCCCs, that, for interleaver sizes $N$ very large and for high values of $\frac{E_b}{N_0}$, are given by [4]:

$$P_{b_{SCCC}} \approx C_o N^{-\frac{d_f}{2}} Q \left( \sqrt{d_f^2 d_2 R_s \frac{E_b}{N_0}} \right)$$

(1)

for even values of $d_f$, and

$$P_{b_{SCCC}} \approx C_o N^{-\frac{d_f+1}{2}} Q \left( \sqrt{\left( d_f^2 - 3 \right) d_2 + 2 d_3} R_s \frac{E_b}{N_0} \right)$$

(2)

for odd values of $d_f$. In both equations, $C_c$ and $C_o$ are two terms which do not depend on the interleaver length $N$, $d_f$ is the free distance of the outer encoder, $d_2$ is the effective free distance of the inner encoder (i.e., the minimum weight of the inner codewords generated by weight-2 input sequences), $R_s$ is the rate of the SCCC, $d_3$ is the minimum weight of the inner code codewords generated by weight-3 input sequences, and $Q(\cdot)$ is the Gaussian integral function defined by $Q(t_o) = \frac{1}{\sqrt{2\pi}} \int_{t_o}^{\infty} e^{-\frac{t^2}{2}} dt$.

Equations (1) and (2) suggest that the outer encoder should be a convolutional encoder with maximum free distance: the interleaver gain goes asymptotically as $N^{-\frac{d_f}{2}}$, for odd values of $d_f$, and as $N^{-\frac{d_f^2}{2}}$ for even values of $d_f$. In particular, compatibly with the desired rate $R_s$ of the SCCC, it is better to choose outer encoders with odd values of $d_f$. It is not necessary for the outer encoder to be neither recursive, nor systematic. Since a large body of literature has addressed the design of good non-recursive convolutional encoders, in this paper we only focus on good punctured codes obtained from recursive, systematic convolutional encoders.

Based on observations deduced above, the code search strategy adopted for obtaining good outer encoders consists in minimizing the required SNR for an appropriately chosen target bit error rate (BER). In fact, as suggested by Lee in [14]-[16], the criteria to only maximize the minimum distance of a code may be valuable if the target BER is extremely small. When on the other hand the target BER is in the moderate range, it is insufficient to only consider the first distance terms of the code’s distance spectrum. Furthermore, minimization of the BER based on the first few terms of the distance spectrum would require the knowledge of the CC operating SNR, which is not always given or known specially if the CC must be embedded in a PCCC or SCCC scheme. Given these considerations, it is preferable to minimize the required SNR for a given target BER. This is indeed the strategy adopted for the code search presented in this paper. Notice that, from a practical point of view, the minimization of the required SNR for a small value of the target BER often yields codes which give distance spectra with maximum free distance. This is essentially due to the fact that the codeword distances appear in the $Q(\cdot)$ function defining the BER probability as shown in (3). The main difference between our approach and that of maximizing only the minimum distance, is that in our approach the codeword multiplicities and the distance profile has a major impact on the choice of the codes.

As a cost function for BER minimization, we employed the upper-bound on the performance of a convolutional code of rate $R_c = \frac{k}{n}$ under soft-decision maximum likelihood Viterbi decoding, defined as follows:

$$P_b \leq \frac{1}{k} \sum_{d=d_{free}}^{\infty} w_d Q \left( \sqrt{\frac{2 E_b}{N_0} R_s d} \right)$$

(3)

where $d_{free}$ is the minimum distance of the CC, also called free distance, $w_d$ is the cumulative input Hamming weight associated with all the incorrect paths (also so-called error events) in the trellis departing from the correct path at a given node and merging with the correct path later, whose Hamming weight of the coded sequence is $d$, and $Q(\cdot)$ is the Gaussian integral function defined above.
Recursive mother encoders selected as constituent encoders in the serial concatenation have been obtained through a search for good constituent recursive, systematic, rate 1/2 convolutional encoders by minimizing the SNR for a given target BER\cite{[14]-[16]}. In order to resolve potential ties, in a second phase, between all the encoders yielding similar performances we have chosen the encoder with the best $d_2$, and, subsequently, the best $d_3$.

The mother encoders obtained by minimization of the SNR can be used as outer codes in SCCC schemes, since the optimization approach has the goal to maximize the minimum distance of the punctured code and minimize the overall weight associated with the input patterns yielding the minimum distance, followed by maximization of the successive low distance terms and minimization of their corresponding input weights for the first six minimum distance terms used in the formulation of the cost function in (3). Extensive analysis suggested us the use of the target BER=$10^{-6}$: such a small target value makes low-weight codewords to dominate the cost function in (3). In the following, we will identify this criterion with the acronym $C_1$.

In connection with the design of inner encoders, we used as objective function the maximization of the effective free distance $d_2$. In a second phase, among the encoders yielding the same $d_2$ (if several), we have chosen the one requiring the minimum SNR for achieving a predefined target BER, and then the one with maximum $d_3$. In the following, we will identify this criterion with the acronym $C_2$.

The results of this search are shown in Table I: the table contains the best rate 1/2, systematic, recursive constituent encoders with memory $\nu$ equal to 2, 3, 4 and 5. Each row presents the optimization results for a given memory $\nu$. In any given row, we show two encoders found using the two different objective functions presented above. In particular, in the first line of a given row we present the best encoder found maximizing $d_2$, while in the second line we present the best encoder found by minimizing the required SNR for achieving a predefined target BER. The second column shows the generator matrices of the optimal encoders, the third column lists the first few terms of the code distance spectra (each triplet represents the Hamming weight of the codewords $d_i$, the multiplicity $m_i$ of all the input patterns with overall weight $w_i$ leading to codewords with weight $d_i$, and the overall input weight $w_i$) and the last column shows the effective free distance $d_2$ and the distance generated by weight-3 sequences denoted $d_3$ (the entry $d_3 = \infty$ is used to signify the fact that there are no weight-3 input patterns leading to error events in the trellis of the examined encoder).

<table>
<thead>
<tr>
<th>$\nu$</th>
<th>$d_2$</th>
<th>$d_3$</th>
<th>$G(D)$</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>$d_2$</td>
<td>$d_3$</td>
<td>$G(D)$</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>$d_2$</td>
<td>$d_3$</td>
<td>$G(D)$</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>$d_2$</td>
<td>$d_3$</td>
<td>$G(D)$</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>$d_2$</td>
<td>$d_3$</td>
<td>$G(D)$</td>
<td></td>
</tr>
</tbody>
</table>

In particular, equality is achieved when the denominator polynomial $g_0(D)$ is primitive and under two additional conditions which require that $g_0(D) \neq g_1(D)$ and that $\deg [g_i(D)] \leq \nu$ for $i = 0, 1$.

B. Design of Punctured Outer Encoders for Asymptotically Good SCCCs

In this section, the emphasis is on the design of punctured convolutional encoders suitable as outer encoders in asymptotically good SCCCs.

The design criteria relies upon the considerations deduced above from Equ.s (1) and (2). Indeed, the code search strategy adopted for obtaining good PPs for outer encoders consists in minimizing the required SNR for the target BER=$10^{-6}$. We searched for the best PPs with this criterion, by using the BER upper bound as expressed in (3), but, for complexity reasons, based on the first four dominant terms of the distance spectrum. Furthermore, we have verified experimentally that the minimization of the required SNR for this target BER practically yields PPs which give distance spectra with maximum free distance. The main difference between our approach and that of maximizing only the minimum distance, is that in our approach the codeword multiplicities and the distance profile has a major impact on the choice of the PPs. In order to resolve potential ties, in a second phase, between all the PPs yielding similar performances we have chosen the PP with the best $d_2$ and subsequently the best $d_3$. In the following, we will identify this criterion with the acronym $C_1$ when applied to PPs.

The mother encoders to which puncturing is applied are the best codes found under criterion $C_1$ discussed in the previous section. We have used these codes as mother codes by following the generally accepted rule that good mother codes lead to good punctured codes. Indeed, practical documented results show that PPs with maximum possible $d_{free}$ are derived from mother encoders having maximum $d_{free}$ [24], [25].

C. Design of Punctured Inner Encoders for Asymptotically Good SCCCs

In this section, the emphasis is on the design of punctured convolutional encoders suitable as inner encoders in asymptotically good SCCCs.
Equations (1) and (2) suggest that good punctured inner encoders in an SCCC are the ones which maximize the effective free distance \( d_2 \). In other words, the effective free distance \( d_2 \) of the inner encoder acts as the free distance of the outer encoder, \( d_0^i \), in the case in which \( d_0^i \) is even. In the case in which \( d_0^i \) is odd, we should also maximize the \( d_3 \) in (2) (\( d_3 \) is the minimum weight of the inner code codewords generated by weight-3 input sequences). Note that these considerations are valid for the design of both mother encoders and related PPs.

Furthermore, the mother encoders designed by maximizing the effective free distance can also be used as constituent encoders in parallel concatenated convolutional codes (PCCCs), since it is known that, asymptotically, the upper-bound on the BER of a PCCC can be expressed as:

\[
P_{b_{PCCC}} \leq \frac{1}{N} Q \left( \sqrt{\frac{2 E_b}{N_0} R_o d_2} \right)
\]

where \( R_o \) is the code rate of the PCCC, and \( N \) is the interleaver length. Since this paper is devoted to the design of good punctured encoders for SCCC schemes, we will not mention PCCCs any further in what follows. The interested readers can refer to [27], [33] for details on this topic.

The search for good PPs for inner encoders has been accomplished by choosing the PP with the best \( d_2 \), and, subsequently, the best \( d_3 \). In order to resolve potential ties, in a second phase, between all the PPs yielding similar performances in terms of \( d_2 \) and \( d_3 \), we have chosen the PP that minimizes the SNR to achieve the target BER \( 10^{-6} \) as expressed in (3), but based on the first four dominant terms of the distance spectrum. Obviously, this approach yields asymptotically good PPs for inner codes in SCCC operating at high SNRs. In the following, we will identify this criterion with the acronym \( C_2 \).

The mother encoders to which puncturing is applied are the best codes found under criterion \( C_2 \) discussed above. Practical documented results show that PPs with maximum possible \( d_2 \) are derived from mother encoders having maximum \( d_2 \) [24], [25].

III. DESIGN OF GOOD SCCCS THROUGH DENSITY EVOLUTION CODE MATCHING

The design of good SCCC operating in the low-to-moderate SNR range (i.e., in the waterfall region before the error floor shows its effects) cannot be based on the asymptotic considerations deduced by (1) and (2). During the search for good SCCC, we recognized that an effective design algorithm for punctured inner encoders in SCCC operating in the low-to-moderate SNR range could be based on the density evolution technique [29], [30], [34], whose main merit is to highlight the convergence behavior of the iterative decoder. In a companion paper [32], we proposed an effective algorithm for matching both outer and inner encoders in an SCCC scheme in such a way as to obtain good punctured SCCC operating in the so-called waterfall region. In order to understand the rationales beyond the proposed algorithm, we will briefly recall some results presented in [32]. In particular, first we will describe a reasonable model for characterizing the extrinsic information exchanged during iterative decoding of SCCC, and, then, we will discuss a nonlinear model for describing the behavior of the iterative decoder.

Consider the iterative decoder scheme of an SCCC shown in Fig. 1, whereby the extrinsic information, denoted by \( \Pi(\cdot) \), is passed from one Soft-In Soft-Out (SISO) decoder to the other. Under the assumption of using large, random interleavers, the extrinsic information messages \( \Pi(\cdot) \) can be considered as statistically independent and identically distributed random variables approaching Gaussian-like distributions \( p_{\Pi}(\xi) \) with increasing number of iterations of the iterative decoder. As shown in [31], the probability density function \( p_{\Pi}(\xi) \) is symmetric and, as a consequence, the statistical mean \( \mu_{\Pi} = E(\Pi) \) represents the discrimination between the two probability density functions \( p_{\Pi}(\xi) \) and \( p_{\Pi}(-\xi) \). Based on these assumptions, the logarithmic version of the SISO input/output can be modelled as [29]:

\[
\Pi = \log \frac{P(1)}{P(0)} = \mu_{\Pi} \cdot a + n
\]

where \( a \) is a binary antipodal symbol \((\pm 1)\), \( n \) is a Gaussian noise with zero mean and variance \( \sigma_{\Pi}^2 \), and \( \mu_{\Pi} = \frac{\sigma_{\Pi}^2}{2} \) is the mean of \( \Pi \).

The random variable \( \Pi \) can be analyzed by specifying a signal-to-noise ratio defined as \( SNR = \frac{\mu_{\Pi}^2}{\sigma_{\Pi}^2} \). The latter approach was first proposed in [30]. Under the assumption that \( p_{\Pi}(\xi) \) is both Gaussian and symmetric, it is possible to observe that \( \sigma_{\Pi}^2 = 2 \cdot \mu_{\Pi} \). The previous relation yields \( SNR = \mu_{\Pi}/2 \).

The iterative decoder shown in Fig. 1 can be described by a nonlinear dynamical feedback system, involving two SISO decoders whose behavior is described by two nonlinear functions defined as follows. The nonlinear function accomplished by the inner decoder on its input variables, namely \( SNR_i \)
and $E_b/N_o$, can be described as:

$$SNRI_o = G_{in}(SNRI_i, \frac{E_b}{N_o})$$

(7)

whereby $SNRI_i$ measures the quality of the extrinsic information exchanged between the two SISO decoders during the iterative decoding algorithm, while $E_b/N_o$ measures the quality of the channel symbols normalized per information bit.

In the same manner, we can identify the input-output relation for the outer SISO decoder by the following equation:

$$SNRO_o = G_{out}(SNRO_i),$$

(8)

and the input-output relation between the input variable $SNRO_i$ and the $SNRO$ evaluated on the soft output, $I^o(\nu; O)$, of the information bits, as follows:

$$SNRO = G_{dec}(SNRO_i).$$

(9)

Since mathematical expressions for characterizing the functions $G_{in}(\cdot)$, $G_{out}(\cdot)$, and $G_{dec}(\cdot)$ are very hard to define, in the proposed algorithm these functions are empirically evaluated by Monte Carlo simulation.

Starting from the nonlinear model of the iterative decoder, the design of good SCCCs operating in the waterfall region

<table>
<thead>
<tr>
<th>TABLE I</th>
<th>Optimized rate $\frac{1}{2}$ recursive constituent encoders for concatenated channel codes.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R = \frac{1}{4}$</td>
<td>$G(D) = \frac{2}{4}$</td>
</tr>
<tr>
<td>$ν = 2, d_2$</td>
<td>$d_2 = 2, d_3 = 3$</td>
</tr>
<tr>
<td>$ν = 2, 3, d_2$</td>
<td>$d_2 = 2, d_3 = 3$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE II</th>
<th>Puncturing Table for 4-state recursive constituent encoders.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G(D) = \frac{1}{2}$</td>
<td>$R = \frac{1}{2}$</td>
</tr>
<tr>
<td>$ν = 1, d_2, d_3$</td>
<td>$d_2 = 2, d_3 = 3$</td>
</tr>
<tr>
<td>$ν = 2, d_2$</td>
<td>$d_2 = 2, d_3 = 3$</td>
</tr>
<tr>
<td>$ν = 3, d_2$</td>
<td>$d_2 = 2, d_3 = 3$</td>
</tr>
<tr>
<td>$ν = 4, d_2$</td>
<td>$d_2 = 2, d_3 = 3$</td>
</tr>
</tbody>
</table>

and $E_b/N_o$, can be described as:

$$SNRI_o = G_{in}(\frac{SNRI_i E_b}{N_o})$$

(7)

whereby $SNRI_i$ measures the quality of the extrinsic information exchanged between the two SISO decoders during the iterative decoding algorithm, while $E_b/N_o$ measures the quality of the channel symbols normalized per information bit.

In the same manner, we can identify the input-output relation for the outer SISO decoder by the following equation:

$$SNRO_o = G_{out}(SNRO_i),$$

(8)

and the input-output relation between the input variable $SNRO_i$ and the $SNRO$ evaluated on the soft output, $I^o(\nu; O)$, of the information bits, as follows:

$$SNRO = G_{dec}(SNRO_i).$$

(9)

Since mathematical expressions for characterizing the functions $G_{in}(\cdot)$, $G_{out}(\cdot)$, and $G_{dec}(\cdot)$ are very hard to define, in the proposed algorithm these functions are empirically evaluated by Monte Carlo simulation.

Starting from the nonlinear model of the iterative decoder, the design of good SCCCs operating in the waterfall region
can be accomplished in two steps. The first step consists in
choosing a punctured outer encoder which maximizes the
SCCC interleaver gain: the constituent outer encoder and the
related PP for achieving a specific code rate, both of them
satisfying the criterion $C_1$, suit well this purpose. The second
step aims at matching the outer and the inner encoders (the
latter ones belonging to the codes found by applying criterion
$C_2$) by choosing the PP for the inner encoder which minimizes
the BER after a target number of iterations $N_{it}$. The choice of
the inner encoder PP is done by tracing for each PP (belonging
to the set of candidate PPs guaranteeing the desired inner
code rate) the SNR exit-charts described by (7) and (8), and,
then, by choosing the PP yielding the maximum value of
$SNRO(N_{it})$ after $N_{it} = 10$ iterations. The bit error probability
$P_b$ can then be evaluated as follows:

$$P_b \approx \frac{1}{2} \text{erfc} \left( \frac{1}{\sqrt{2}} \frac{\mu_{P_U}(u)}{\sigma_{P_U}(u)} \right) = \frac{1}{2} \text{erfc} \left( \frac{\sqrt{SNRO(u)}}{2} \right).$$

(11)

This equation shows that the $SNRO(N_{it})$ can be used as a
cost function the maximization of which can lead to lower
BER.

Punctured SCCCs designed with this technique, present a
low error-floor due to the good distance spectra of the outer
encoders embedded in the serial concatenation, supported by early
converging punctured inner encoders.

Before concluding the section, let us discuss the parameters
employed for the proposed coding search technique. The block
size used for tracing exit-charts was set to 200,000: such a
high value is needed for guaranteeing the hypotheses discussed
above on the statistical behaviour of the extrinsic messages
$\Pi(\cdot)$. Notice that such values of block size have been also
proposed by ten Brink [29] in relation to the density evolution

TABLE III

<table>
<thead>
<tr>
<th>$G(D)$</th>
<th>$\cdot$</th>
<th>$R = \frac{4}{5}$</th>
<th>$R = \frac{3}{4}$</th>
<th>$R = \frac{2}{3}$</th>
<th>$R = \frac{1}{2}$</th>
<th>$R = \frac{2}{5}$</th>
<th>$R = \frac{1}{5}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>BER</td>
<td>$\theta$</td>
<td>$d_2 = 4 $</td>
<td>$d_3 = 5 $</td>
<td>$d_4 = 6 $</td>
<td>$d_5 = 7 $</td>
<td>$d_6 = 8 $</td>
<td>$d_7 = 9 $</td>
</tr>
<tr>
<td>$d_2, BER_{opt}$</td>
<td>$p_7(3,1,4)$</td>
<td>$p_7(2,1,4)$</td>
<td>$p_7(1,2,4)$</td>
<td>$p_7(1,3,4)$</td>
<td>$p_7(2,2,4)$</td>
<td>$p_7(1,4,4)$</td>
<td>$p_7(2,3,4)$</td>
</tr>
</tbody>
</table>

For non-systematic outer encoders, we can only state that $\Pi^o(u; O)$ is a function, say $f(\cdot)$, of $\Pi^o(c; I) + \Pi^o(c; O)$.

Based on the assumptions deduced above, both soft infor-

mation $\Pi^o(c; I)$ and $\Pi^o(c; O)$ are Gaussian distributed. As

a consequence, the outer decoder soft output $\Pi^o(u; O)$ is

supposed to be Gaussian distributed with variance $\sigma_{P_U}^2$ and

mean value $\mu_{P_U}(u; O) = \frac{\sigma_{P_U}(u; O)}{2}$. The bit error probability

$P_b$ can then be evaluated as follows:

$$P_b \approx \frac{1}{2} \text{erfc} \left( \frac{1}{\sqrt{2}} \frac{\mu_{P_U}(u)}{\sigma_{P_U}(u)} \right) = \frac{1}{2} \text{erfc} \left( \frac{\sqrt{SNRO(u)}}{2} \right).$$

(11)
TABLE IV
Puncturing Table for the 16-state recursive constituent encoders.

<table>
<thead>
<tr>
<th>$G(D)$</th>
<th>$R = \frac{d}{4}$</th>
<th>$R = \frac{d}{2}$</th>
<th>$R = \frac{d}{8}$</th>
<th>$R = \frac{d}{16}$</th>
<th>$R = \frac{d}{32}$</th>
<th>$R = \frac{d}{64}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>BER</td>
<td>$p_1(1,3)$, $d_2 = 4$</td>
<td>$p_1(3,5)$, $d_2 = 4$</td>
<td>$p_1(5,8)$, $d_2 = 4$</td>
<td>$p_1(7,10)$, $d_2 = 4$</td>
<td>$p_1(9,12)$, $d_2 = 4$</td>
<td>$p_1(11,14)$, $d_2 = 4$</td>
</tr>
<tr>
<td>$d_2$, BER, opt</td>
<td>$p_1(1,3)$, $d_2 = 4$</td>
<td>$p_1(3,5)$, $d_2 = 4$</td>
<td>$p_1(5,8)$, $d_2 = 4$</td>
<td>$p_1(7,10)$, $d_2 = 4$</td>
<td>$p_1(9,12)$, $d_2 = 4$</td>
<td>$p_1(11,14)$, $d_2 = 4$</td>
</tr>
<tr>
<td>$d_2$, BER, opt</td>
<td>$p_1(1,3)$, $d_2 = 4$</td>
<td>$p_1(3,5)$, $d_2 = 4$</td>
<td>$p_1(5,8)$, $d_2 = 4$</td>
<td>$p_1(7,10)$, $d_2 = 4$</td>
<td>$p_1(9,12)$, $d_2 = 4$</td>
<td>$p_1(11,14)$, $d_2 = 4$</td>
</tr>
<tr>
<td>BER</td>
<td>$p_1(3,5)$, $d_2 = 6$</td>
<td>$p_1(5,8)$, $d_2 = 6$</td>
<td>$p_1(7,10)$, $d_2 = 6$</td>
<td>$p_1(9,12)$, $d_2 = 6$</td>
<td>$p_1(11,14)$, $d_2 = 6$</td>
<td>$p_1(13,16)$, $d_2 = 6$</td>
</tr>
<tr>
<td>$d_2$, BER, opt</td>
<td>$p_1(3,5)$, $d_2 = 6$</td>
<td>$p_1(5,8)$, $d_2 = 6$</td>
<td>$p_1(7,10)$, $d_2 = 6$</td>
<td>$p_1(9,12)$, $d_2 = 6$</td>
<td>$p_1(11,14)$, $d_2 = 6$</td>
<td>$p_1(13,16)$, $d_2 = 6$</td>
</tr>
<tr>
<td>BER</td>
<td>$p_1(5,8)$, $d_2 = 6$</td>
<td>$p_1(7,10)$, $d_2 = 6$</td>
<td>$p_1(9,12)$, $d_2 = 6$</td>
<td>$p_1(11,14)$, $d_2 = 6$</td>
<td>$p_1(13,16)$, $d_2 = 6$</td>
<td>$p_1(15,18)$, $d_2 = 6$</td>
</tr>
<tr>
<td>$d_2$, BER, opt</td>
<td>$p_1(5,8)$, $d_2 = 6$</td>
<td>$p_1(7,10)$, $d_2 = 6$</td>
<td>$p_1(9,12)$, $d_2 = 6$</td>
<td>$p_1(11,14)$, $d_2 = 6$</td>
<td>$p_1(13,16)$, $d_2 = 6$</td>
<td>$p_1(15,18)$, $d_2 = 6$</td>
</tr>
</tbody>
</table>

Each PP listed, we show the minimum distance $d_m$, the number of nearest neighbors $M_m$ yielding the minimum distance $d_m$, and the total weight $W_m$ of these input patterns as the triplet $(d_m, M_m, W_m)$. In the lower part of any given row, we specify the effective free distance $d_2$, and the weight-3 distance $d_3$, of the punctured encoders. Where $d_3$ is not specified, it means that there are no weight-3 input patterns leading to error events of the punctured encoder.

As an example of how to read the table entries, consider the rate 2/3 PP 13 shown in Table II satisfying the criterion $C_1$ obtained by using the 4-state mother encoder with generator matrix $G(D) = [1, 5]$. This PP leads to a encoder whose minimum distance 3 is due to one input pattern with weight 3. The effective free distance $d_2$ of the encoder is 4 and the distance $d_3$ is 3.

The PPs are represented in octal notation. A given PP should be read from right to left by collecting $k$-pairs of systematic-parity bits. As an example, the PP in Table II which yields a code with rate 2/3 for the 4-state encoder, should be interpreted as follows: $p = 138 = 1011_2 = \langle x_1, y_1, x_2, y_2 \rangle$ (the subscript denotes the base of the numbers). In this case the PP leaves the encoder systematic and deletes the first parity bit associated with every two input bits.

Tables VI-IX show the best PPs for inner encoders obtained by density evolution (criterion $C_3$) as explained in section III. The tables are organized as follows. In the first column, we technique applied to the analysis of the iterative decoder of PCCCs. In a practical setting, the decorrelation between the extrinsic messages is guaranteed by the interleaver in the serial concatenation. The value of $E_b/N_0$ used for code design was set to 0.5dB from the Shannon channel capacity assuming BPSK modulation and AWGN channel. This value was chosen based on extensive tests. Suffice to say that it guarantees early converging SCCCs. We invite the interested reader to refer to [32] for the details of the algorithm along with the set of parameters used for code search.

In the following, the results obtained by applying this algorithm will be denoted by criterion $C_3$.

IV. CODE SEARCH RESULTS AND APPLICATIONS

The results of the search for good PPs are presented in Tables II-IX. In particular, Tables II-V contain the PPs obtained by applying the criteria $C_1$ and $C_2$. Each table collects the results related to encoders with the same number of states. Generator matrices are specified in the upper leftmost corner of each table. For any encoder, we show the best PPs specified in the appropriate column yielding a punctured encoder with rate $\frac{k}{k+s}$ (with $k \in \{2, \ldots, 8\}$ and $s \in \{1, \ldots, k - 1\}$) satisfying the design criteria $C_1$ and $C_2$ discussed above. In particular, the upper row specifies the best PPs obtained with the criterion $C_1$. The lower row, identified by $d_2$, BER, opt, specifies the best PP obtained with the criterion $C_2$. Next to
show the rate $R_o = \frac{k_1}{k_2 + 1}$ of the punctured outer encoder along with the respective PP, both of them obtained upon application of criterion $C_1$. The intersection between a given row indexed by the value of the rate of the punctured outer encoder (i.e., $k_2$ where $k_2 \in \{1, \ldots, 8\}$), and a column, indexed by the value $R_i = \frac{k_1}{k_2 + 1}$, indicates the optimal PP which yields an inner encoder with rate $\frac{k_1}{k_2 + 1}$ where $k_1 \in \{2, \ldots, 8\}$. In this case, by neglecting the effects of the encoder terminations, the overall SCCC presents a rate $R_o = \frac{k_1}{k_2 + 1}$ inner encoder concatenated with the rate-$2/3$ outer encoder punctured with the PP $15$, is $56$. Simulation results shown in Fig. 2 where the best PP found by criterion $C_3$ is compared to other candidate PPs yielding a rate-3/4 inner encoder, confirm the effectiveness of the design approach $C_3$. In the legend we show the $SNR$ obtained after $10$ iterations for each PP examined. We invite the interested reader to refer to [32] for further simulation results on SCCCs designed with the criterion $C_3$.

V. CONCLUSIONS

In this paper we presented extensive optimized puncturing pattern tables for designing serially concatenated convolutional codes (SCCCs) operating in two different objective regions: in the waterfall region and in the so-called error-floor region.

The optimization has been conducted using three different techniques each one of which is suited to a certain application in connection with the design of SCCCs.

We have further conducted exhaustive search for mother encoders of rate $1/2$ to be used for puncturing using two different selection criteria. These encoders have then been used as mother encoders to which puncturing is applied.

Sample simulation results were conducted to verify the functionality of the designed encoders in the concatenated coding context.

<table>
<thead>
<tr>
<th>$G(D) =$</th>
<th>$R_o = \frac{4}{15}$</th>
<th>$R_o = \frac{4}{9}$</th>
<th>$R_o = \frac{4}{7}$</th>
<th>$R_o = \frac{4}{5}$</th>
<th>$R_o = \frac{4}{3}$</th>
<th>$R_o = \frac{4}{2}$</th>
<th>$R_o = \frac{4}{1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>BER</td>
<td>$p \in \mathbb{P}(0.06)$</td>
<td>$d_g = 6$</td>
<td>$\rho = \frac{4}{9}$</td>
<td>$\rho = \frac{4}{7}$</td>
<td>$\rho = \frac{4}{5}$</td>
<td>$\rho = \frac{4}{3}$</td>
<td>$\rho = \frac{4}{2}$</td>
</tr>
<tr>
<td>$d_2,BER, opt$</td>
<td>$p \in \mathbb{P}(0.32)$</td>
<td>$d_g = 6$</td>
<td>$\rho = \frac{4}{9}$</td>
<td>$\rho = \frac{4}{7}$</td>
<td>$\rho = \frac{4}{5}$</td>
<td>$\rho = \frac{4}{3}$</td>
<td>$\rho = \frac{4}{2}$</td>
</tr>
</tbody>
</table>

TABLE V: Puncturing Table for 32-state recursive constituent encoders.
### TABLE VI

Puncturing Tables for SCCC schemes composed by 4-state recursive constituent encoders.

<table>
<thead>
<tr>
<th>( G_1(D) = { 1, D } )</th>
<th>( G_2(D) = { 1, D } )</th>
<th>( d_0 )</th>
<th>( d_1 )</th>
<th>( d_2 )</th>
<th>( d_3 )</th>
<th>( \Delta )</th>
<th>( \Phi )</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>35</td>
<td>652</td>
<td>355</td>
<td>4865</td>
<td>25253</td>
<td>152552</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>67</td>
<td>1721</td>
<td>4487</td>
<td>25466</td>
<td>110973</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>56, 3</td>
<td>175</td>
<td>2627</td>
<td>32432</td>
<td>12168</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>33</td>
<td>352</td>
<td>671</td>
<td>4638</td>
<td>24467</td>
<td>55227</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>47</td>
<td>265</td>
<td>871</td>
<td>2853</td>
<td>25846</td>
<td>116331</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>15253, 2</td>
<td>352</td>
<td>3932</td>
<td>18944</td>
<td>18944</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>47</td>
<td>151</td>
<td>326</td>
<td>3452</td>
<td>14647</td>
<td>117321</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>125253, 2</td>
<td>1172</td>
<td>2732</td>
<td>14647</td>
<td>151146</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( G_2(D) = { 1, D } )</td>
<td>( G_3(D) = { 1, D } )</td>
<td>( d_0 )</td>
<td>( d_1 )</td>
<td>( d_2 )</td>
<td>( d_3 )</td>
<td>( \Delta )</td>
<td>( \Phi )</td>
</tr>
<tr>
<td>15</td>
<td>33</td>
<td>263</td>
<td>1392</td>
<td>5391</td>
<td>29852</td>
<td>125283</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>50</td>
<td>354</td>
<td>1057</td>
<td>7311</td>
<td>7236</td>
<td>104235</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>50</td>
<td>353</td>
<td>36</td>
<td>47</td>
<td>7235</td>
<td>104332</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>272</td>
<td>1222</td>
<td>226</td>
<td>117066</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>272</td>
<td>1222</td>
<td>226</td>
<td>117066</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>272</td>
<td>1222</td>
<td>226</td>
<td>117066</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>272</td>
<td>1222</td>
<td>226</td>
<td>117066</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>272</td>
<td>1222</td>
<td>226</td>
<td>117066</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>272</td>
<td>1222</td>
<td>226</td>
<td>117066</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>272</td>
<td>1222</td>
<td>226</td>
<td>117066</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( G_4(D) = { 1, D } )</td>
<td>( \Delta )</td>
<td>( \Phi )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>50</td>
<td>63, 3</td>
<td>( \infty )</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>50</td>
<td>63, 3</td>
<td>( \infty )</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>50</td>
<td>63, 3</td>
<td>( \infty )</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>50</td>
<td>63, 3</td>
<td>( \infty )</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>50</td>
<td>63, 3</td>
<td>( \infty )</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>50</td>
<td>63, 3</td>
<td>( \infty )</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>50</td>
<td>63, 3</td>
<td>( \infty )</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>50</td>
<td>63, 3</td>
<td>( \infty )</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### REFERENCES


### Table IX
Puncturing Table for the 32-state recursive constituent inner encoder with generator matrix \( G(D) = [1, \frac{d_1}{d_2}, \frac{d_2}{d_1}] \). The outer encoder in the SCCC has generator matrix \( G(D) = [1, \frac{d_1}{d_2}, \frac{d_2}{d_1}] \).

<table>
<thead>
<tr>
<th>( p )</th>
<th>( d_1 )</th>
<th>( d_2 )</th>
<th>( p \times (5, 2, 7) )</th>
<th>( d_1 = 9, d_2 = 4 )</th>
<th>( d_1 = 7, d_2 = 4 )</th>
<th>( p \times (4, 2, 7) )</th>
<th>( d_1 = 7, d_2 = 4 )</th>
<th>( p \times (3, 2, 7) )</th>
<th>( d_1 = 7, d_2 = 4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>35</td>
<td>1266</td>
<td>5253</td>
<td>29526</td>
<td>1272552</td>
<td>1272552</td>
<td>1272552</td>
<td>1272552</td>
<td>1272552</td>
</tr>
<tr>
<td>16</td>
<td>35</td>
<td>1266</td>
<td>5253</td>
<td>29526</td>
<td>1272552</td>
<td>1272552</td>
<td>1272552</td>
<td>1272552</td>
<td>1272552</td>
</tr>
<tr>
<td>17</td>
<td>35</td>
<td>1266</td>
<td>5253</td>
<td>29526</td>
<td>1272552</td>
<td>1272552</td>
<td>1272552</td>
<td>1272552</td>
<td>1272552</td>
</tr>
<tr>
<td>18</td>
<td>35</td>
<td>1266</td>
<td>5253</td>
<td>29526</td>
<td>1272552</td>
<td>1272552</td>
<td>1272552</td>
<td>1272552</td>
<td>1272552</td>
</tr>
</tbody>
</table>

---


