Serial Concatenation of Interleaved Codes: Performance Analysis, Design, and Iterative Decoding

S. Benedetto,[a] D. Divsalar,[b] G. Montorsi,[a] and F. Pollara[b]

A serially concatenated code with an interleaver consists of the cascade of an outer code, an interleaver permuting the outer codewords' bits, and an inner code whose input words are the permuted outer codewords. The construction can be generalized to h cascaded codes separated by h − 1 interleavers. We obtain upper bounds to the average maximum-likelihood bit-error probability of serially concatenated block and convolutional coding schemes. Then, we derive some design guidelines for the outer and inner codes that maximize the interleaver gain and the asymptotic slope of the error probability curves. Finally, we propose a new, low-complexity iterative decoding algorithm that yields performance close to maximum-likelihood decoding. Throughout the article, extensive comparisons with parallel concatenated convolutional codes, known as "turbo codes," are performed, showing that the new schemes offer superior performance.

I. Introduction

In his goal to find a class of codes whose probability of error decreased exponentially at rates less than capacity, while decoding complexity increased only algebraically, David Forney [1] arrived at a solution consisting of the multilevel coding structure known as concatenated code. It consists of the cascade of an inner code and an outer code, which, in Forney's approach, would be a relatively short inner code (typically, a convolutional code) admitting simple maximum-likelihood decoding, and a long high-rate algebraic nonbinary Reed–Solomon outer code equipped with a powerful algebraic error-correction algorithm, possibly using reliability information from the inner decoder.

Initially motivated only by theoretical research interests, concatenated codes have since then evolved as a standard for those applications where very high coding gains are needed, such as space and deep-space applications among many others. Alternative solutions for concatenation have also been studied, such as using a trellis-coded modulation scheme as an inner code [2] or concatenating two convolutional codes [3]. In the latter case, the inner Viterbi decoder employs a soft-output decoding algorithm to provide soft-input decisions to the outer Viterbi decoder. An interleaver was also proposed between the two encoders to separate bursts of errors produced by the inner decoder.

---

[a] Politecnico di Torino, Torino, Italy.

[b] Communications Systems and Research Section.

We find then, in a "classical" concatenated coding scheme, the main ingredients that formed the basis for the invention of "turbo codes" [4], namely, two or more constituent codes (CCs) and an interleaver. The novelty of turbo codes, however, consists of the way they use the interleaver, which is embedded into the code structure to form an overall concatenated code with a very large block length, and in the proposal of a parallel concatenation to achieve a higher rate for given rates of CCs. The latter advantage is obtained by using systematic CCs and not transmitting the information bits entering the second encoder. In the following, we will refer to turbo codes as parallel concatenated convolutional codes (PCCCs). The so-obtained codes have been shown to yield very high coding gains at bit-error probabilities around 10^{-5} through 10^{-7}; in particular, low bit-error probabilities can be obtained at rates well beyond the channel cutoff rate, which had been regarded for a long time as the "practical" capacity. Quite remarkably, this performance can be achieved by a relatively simple iterative decoding technique whose computational complexity is comparable to that needed to decode the two CCs.

In this article, we consider the serial concatenation of interleaved codes or serially concatenated codes (SCCs), called serially concatenated block codes (SCBCs) or serially concatenated convolutional codes (SCCCs) according to the nature of the CCs. For this class of codes, we obtain analytical upper bounds to the performance of a maximum-likelihood (ML) decoder, propose design guidelines leading to the optimal choice of CCs for maximizing the interleaver gain and the asymptotic code performance, and present a new iterative decoding algorithm yielding results close to maximum-likelihood decoding with limited complexity. Extensive comparisons with turbo codes of the same complexity and decoding delay are performed.

With these results, we think that SCCC can be considered a valid, in some cases superior, alternative to turbo codes.

In Section II, we derive analytical upper bounds to the bit-error probability of both SCBCs and SCCCs, using the concept of a "uniform interleaver" that decouples the output of the outer encoder from the input of the inner encoder. In Section III, we propose design rules for SCCCs through an asymptotic approximation of the bit-error probability bound, assuming long interleavers or large signal-to-noise ratios. In Section IV, we compare serial and parallel concatenated codes. Section V is devoted to the presentation of a new iterative decoding algorithm and to its application to some significant codes. Examples of SCCCs for deep-space communications are presented in Section VI.

II. Analytical Bounds to the Performance of Serially Concatenated Codes

For simplicity of presentation, we begin by considering serially concatenated block codes (SCBCs).

A. Serially Concatenated Block Codes

The scheme of two serially concatenated block codes is shown in Fig. 1. It is composed of two cascaded CCs, the outer (N, k) code C_o with rate $R_c^o = k/N$ and the inner (n, N) code C_i with rate $R_c^i = N/n$, linked by an interleaver of length N. The overall SCBC is then an (n, k) code, and we will refer to it as the (n, k, N) code C_S, including also the interleaver length. In the following, we will derive an upper bound to the ML performance of the overall code C_S. We assume that the CCs are linear, so that the SCBC also is linear and the uniform error property applies, i.e., the bit-error probability can be evaluated assuming that the all-zero codeword has been transmitted.

As in [5,6], a crucial step in the analysis consists of replacing the actual interleaver that performs a permutation of the N input bits with an abstract interleaver called a uniform interleaver, defined as a probabilistic device that maps a given input word of weight l into all distinct $\binom{N}{l}$ permutations of it with equal probability $p = 1/\binom{N}{l}$ (see Fig. 2), so that the output word of the outer code and the input word of the inner code share the same weight. Use of the uniform interleaver permits the computation of the "average" performance of SCBCs, intended as the expectation of the performance of SCBCs using
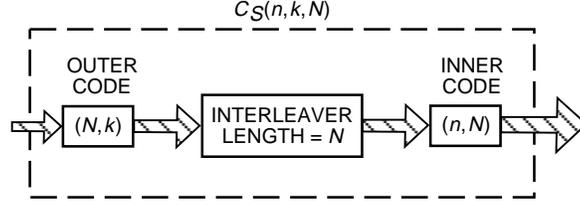
Fig. 1. Serially concatenated (n,k,N) block code.
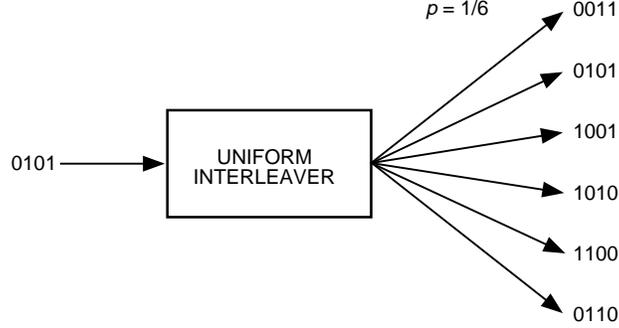
Fig. 2. The action of a uniform interleaver of length 4 on sequences of weight 2.

the same CCs, taken over the ensemble of all interleavers of a given length. A theorem proved in [6] guarantees the meaningfulness of the average performance, in the sense that there will always be, for each value of the signal-to-noise ratio, at least one particular interleaver yielding performance better than or equal to those of the uniform interleaver.

Let us define the input–output weight enumerating function (IOWEF) of the SCBC C_S as

$$A^{C_S}(W, H) = \sum_{w,h} A^{C_S}_{w,h} W^w H^h \tag{1}$$

where $A^{C_S}_{w,h}$ is the number of codewords of the SCBC with weight h associated with an input word of weight w. We also define the conditional weight enumerating function (CWEF) $A^{C_S}(w, H)$ of the SCBC as the weight distribution of codewords of the SCBC that have input word weight w. It is related to the IOWEF by

$$A^{C_S}(w, H) = \frac{1}{w!} \left. \frac{\partial^w A^{C_S}(W, H)}{\partial W^w} \right|_{W=0} \tag{2}$$

With knowledge of the CWEF, an upper bound to the bit-error probability of the SCBC can be obtained in the form [6]

$$P_b(e) \leq \sum_{w=1}^{k} \frac{w}{k} A^{C_S}(w, H) \big|_{H=e^{-R_c E_b/N_0}} \tag{3}$$

where $R_c = k/n$ is the rate of C_S, and E_b/N_0 is the signal-to-noise ratio per bit.

3

The problem thus consists in the evaluation of the CWEF of the SCBC from the knowledge of the CWEFs of the outer and inner codes, which we call $A^{C_o}(w, L)$ and $A^{C_i}(l, H)$. To do this, we exploit the properties of the uniform interleaver, which transforms a codeword of weight l at the output of the outer encoder into all its distinct $\binom{N}{l}$ permutations. As a consequence, each codeword of the outer code C_o of weight l, through the action of the uniform interleaver, enters the inner encoder generating $\binom{N}{l}$ codewords of the inner code C_i. Thus, the number $A^{C_S}_{w,h}$ of codewords of the SCBC of weight h associated with an input word of weight w is given by

$$A^{C_S}_{w,h} = \sum_{l=0}^{N} \frac{A^{C_o}_{w,l} \times A^{C_i}_{l,h}}{\binom{N}{l}} \tag{4}$$

From Eq. (4), we derive the expressions of the IOWEF and CWEF of the SCBC:

$$A^{C_S}(w, H) = \sum_{l=0}^{N} \frac{A^{C_o}_{w,l} \times A^{C_i}(l, H)}{\binom{N}{l}} \tag{5}$$

$$A^{C_S}(W, H) = \sum_{l=0}^{N} \frac{A^{C_o}(W, l) \times A^{C_i}(l, H)}{\binom{N}{l}} \tag{6}$$

where $A^{C_o}(W, l)$ is the conditional weight distribution of the input words that generate codewords of the outer code of weight l.

Example 1. Consider the (7,3) SCBC code obtained by concatenating the (4,3) parity check code to a (7,4) Hamming code through an interleaver of length $N = 4$. The IOWEF $A^{C_o}(W, L)$ and $A^{C_i}(L, H)$ of the inner and outer codes are

$$A^{C_o}(W, L) = 1 + W(3L^2) + W^2(3L^2) + W^3(L^4)$$

$$A^{C_i}(L, H) = 1 + L(3H^3 + H^4) + L^2(3H^3 + 3H^4) + L^3(H^3 + 3H^4) + L^4 H^7$$

so that

$$A^{C_o}(W, 0) = 1$$

$$A^{C_o}(W, 1) = 0$$

$$A^{C_o}(W, 2) = 3W + 3W^2$$

$$A^{C_o}(W, 3) = 0$$

$$A^{C_o}(W, 4) = W^3$$

and

$$A^{C_i}(0, H) = 1$$

$$A^{C_i}(1, H) = 3H^3 + H^4$$

$$A^{C_i}(2, H) = 3H^3 + 3H^4$$

$$A^{C_i}(3, H) = H^3 + 3H^4$$

$$A^{C_i}(4, H) = H^7$$

Through Eq. (6), we then obtain

$$A^{C_S}(W, H) = \sum_{l=0}^{4} \frac{A^{C_o}(W, l) \times A^{C_i}(l, H)}{\binom{N}{l}}$$

$$= \frac{1 \times 1}{1} + \frac{0 \times (3H^3 + H^4)}{4} + \frac{(3W + 3W^2) \times (3H^3 + 3H^4)}{6} + \frac{0 \times (H^3 + 3H^4)}{4} + \frac{W^3 \times H^7}{1}$$

$$= 1 + W(1.5H^3 + 1.5H^4) + W^2(1.5H^3 + 1.5H^4) + W^3 H^7$$

The previous results of Eqs. (6) and (5) can be easily generalized to the case of an interleaver of length N, which is an integer multiple of the length of the outer codewords. Denoting by $A^{C_o^m}(W, L)$ the IOWEF of the new (N, mk) outer code, and similarly by $A^{C_i^m}(L, H)$ the IOWEF of the new (mn, N) inner code, it is straightforward to obtain

$$\left. \begin{array}{l} A^{C_o^m}(W, L) = [A^{C_o}(W, L)]^m \\[2em] A^{C_i^m}(L, H) = [A^{C_i}(L, H)]^m \end{array} \right\} \tag{7}$$

From the IOWEFs of Eq. (7), and through Eq. (2), we obtain the CWEFs $A^{C_o^m}(W, l)$ and $A^{C_o^m}(l, H)$ of the new CCs and, finally, the IOWEF and CWEF of the new (n, k, mN) SCBC C_S^m:

$$A^{C_S^m}(w, H) = \sum_{l=0}^{N} \frac{A_{w,l}^{C_o^m} \times A^{C_i^m}(l, H)}{\binom{N}{l}} \tag{8}$$

$$A^{C_S^m}(W, H) = \sum_{l=0}^{N} \frac{A^{C_o^m}(W, l) \times A^{C_i^m}(l, H)}{\binom{N}{l}} \tag{9}$$

Example 2. Consider again the CCs of Example 1, linked by an interleaver of length $N = 4m$, and use Eqs. (8) and (3). The so-obtained upper bound to the bit-error probability is plotted in Fig. 3 for various values of the integer m. The curves show the interleaver gain, defined as the factor by which the bit-error probability is decreased with the interleaver length. Contrary to parallel concatenated block
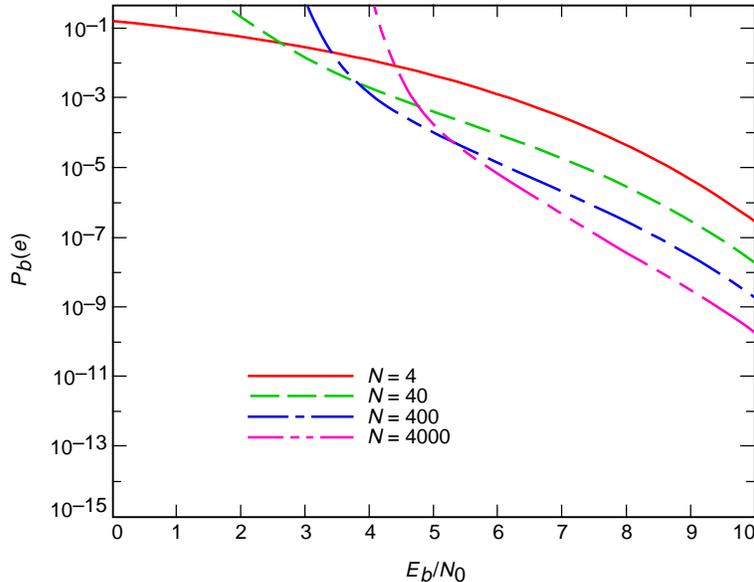
Fig. 3. Analytical bounds for the SCBC of Example 2 (SCBC1).

codes [6], the curves do not exhibit the interleaver gain saturation. Rather, the bit-error probability seems to decrease regularly with m as m^{-1}. We will explain this behavior in Section III.

B. Serially Concatenated Convolutional Codes

The structure of a serially concatenated convolutional code (SCCC) is shown in Fig. 4. It refers to the case of two convolutional CCs, the outer code C_o with rate $R_c^o = k/p$, and the inner code C_i with rate $R_c^i = p/n$, joined by an interleaver of length N bits, generating an SCCC C_S with rate $R_c = k/n$. Note that N must be an integer multiple of p.[1] We assume, as before, that the convolutional CCs are linear, so that the SCCC is linear as well, and the uniform error property applies.

The exact analysis of this scheme can be performed by appropriate modifications of the analysis described in [6] or [13] for PCCCs. It requires the use of a hypertrellis having as hyperstates pairs of states of outer and inner codes. The hyperstates S_{ij} and S_{lm} are joined by a hyperbranch that consists of all pairs of paths with length N/p that join states s_i and s_l of the inner code and states s_j and s_m of the outer code, respectively. Each hyperbranch is thus an equivalent SCBC labeled with an IOWEF
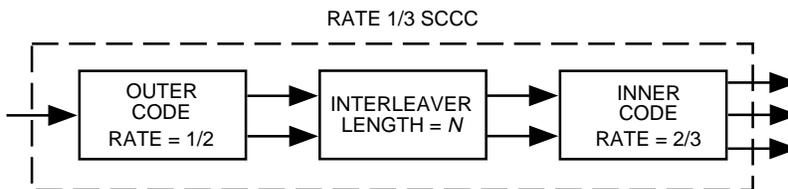
Fig. 4. Serially concatenated (n, k, N) convolutional code.

---

[1] Actually, this constraint is not necessary. We can choose in fact inner and outer codes of any rates $R_c^i = k_i/n_i$ and $R_c^o = k_o/n_o$, constraining the interleaver to be an integer multiple of the minimum common multiple of n_o and k_i, i.e., $N = K \cdot \mathrm{mcm}(n_o, k_i)$. This generalization, though, leads to more complicated expressions and is not considered in the following.

6

that can be evaluated as explained in the previous subsection. From the hypertrellis, the upper bound to the bit-error probability can be obtained through the standard transfer function technique employed for convolutional codes [7]. As proved in [6], when the length of the interleaver is significantly greater than the constraint length of the CCs, an accurate approximation of the exact upper bound consists in retaining only the branch of the hypertrellis joining the hyperstates S_{00} and S_{00}. In the following, we will always use this approximation.

Example 3. Consider a rate 1/3 SCCC formed by an outer four-state convolutional code with rate 1/2 and an inner four-state convolutional code with rate 2/3, joined by a uniform interleaver of length $N = 200, 400, 600, 800, 1000$, and 2000. Both codes are systematic and recursive, and the generator matrices are given in the first and third rows of Table 1. Using the previously outlined analysis, we have obtained the bit-error probability bounds shown in Fig. 5. The performance shows a very significant interleaver gain, i.e., lower values of the bit-error probability for higher values of N.

Table 1. Generating matrices for the constituent convolutional codes.

Code description	$G(D)$
Rate 1/2 recursive	$\left[1, \quad \frac{1+D^2}{1+D+D^2} \right]$
Rate 1/2 nonrecursive	$[\, 1 + D + D^2, \quad 1 + D^2 \,]$
Rate 2/3 recursive	$\begin{bmatrix} 1, & 0, & \frac{1+D^2}{1+D+D^2} \\ 0, & 1, & \frac{1+D}{1+D+D^2} \end{bmatrix}$
Rate 2/3 nonrecursive	$\begin{bmatrix} 1+D, & D, & 1 \\ 1+D, & 1, & 1+D \end{bmatrix}$
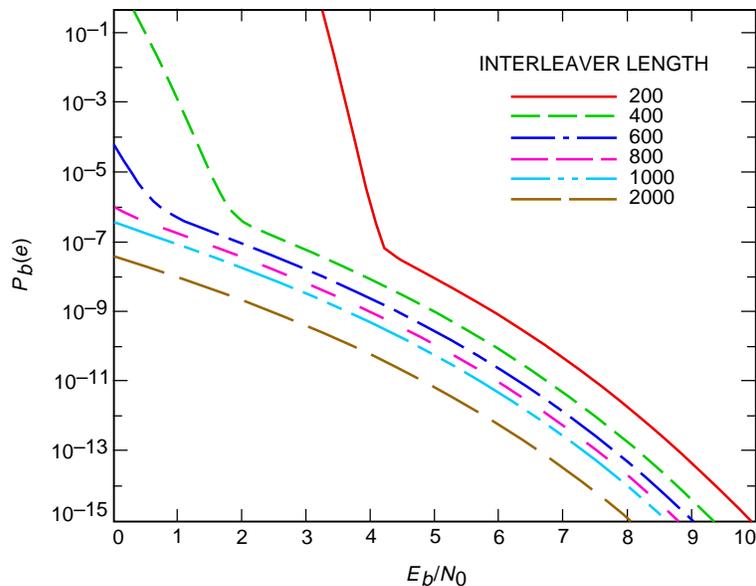
Fig. 5. Analytical bounds for the SCCC of Example 3 (SCCC1).

III. Design of Serially Concatenated Codes

In the previous section, we presented an analytical bounding technique to find the ML performance of SCBC and SCCC. For practical applications, SCCCs are to be preferred to SCBCs. One reason is that maximum a posteriori algorithms are less complex for convolutional than for block codes; a second is that the interleaver gain can be greater for convolutional CCs, provided they are suitably designed. Hence, we deal mainly with the design of SCCCs, extending our conclusions to SCBCs when appropriate.

Consider the SCCC depicted in Fig. 4. Its performance can be approximated by that of an equivalent block code whose IOWEF labels the branch of the hypertrellis joining the zero states of the outer and inner codes. Denoting by $A^{C_S}(w, H)$ the CWEF of this equivalent block code, we can rewrite the upper bound, Eq. (3), as[2]

$$P_b(e) \leq \sum_{w=w_m^o}^{NR_c^o} \frac{w}{NR_c^o} A^{C_S}(w, H)|_{H=e^{-R_c E_b/N_0}} = \sum_{h=h_m}^{N/R_c^i} \sum_{w=w_m^o}^{NR_c^o} \frac{w}{NR_c^o} A_{w,h}^{C_S} e^{-hR_c E_b/N_0} \qquad (10)$$

where w_m^o is the minimum weight of an input sequence generating an error event of the outer code, and h_m is the minimum weight[3] of the codewords of C_S. By "error event of a convolutional code" we mean a sequence diverging from the zero state at time zero and remerging into the zero state at some discrete time $j > 0$. For constituent block codes, an error event is simply a codeword.

The coefficients $A_{w,h}^{C_S}$ of the equivalent block code can be obtained from Eq. (4) once the quantities $A_{w,l}^{C_o}$ and $A_{l,h}^{C_i}$ of the CCs are known. To evaluate them, consider a rate $R = p/n$ convolutional code C with memory ν, and its equivalent $(N/R, N - p\nu)$ block code whose codewords are all sequences of length N/R bits of the convolutional code starting from and ending at the zero state. By definition, the codewords of the equivalent block code are concatenations of error events of the convolutional codes. Let

$$A(l, H, j) = \sum_h A_{l,h,j} H^h \qquad (11)$$

be the weight enumerating function of sequences of the convolutional code that concatenate j error events with total input weight l (see Fig. 6), where $A_{l,h,j}$ is the number of sequences of weight h, input weight l, and number of concatenated error events j. For N much larger than the memory of the convolutional code, the coefficient $A_{l,h}^C$ of the equivalent block code can be approximated by[4]

$$A_{l,h}^C \sim \sum_{j=1}^{n_M} \binom{N/p}{j} A_{l,h,j} \qquad (12)$$

---

[2] In the following, a subscript m will denote minimum, and a subscript M will denote maximum.

[3] Since the input sequences of the inner code are *not* unconstrained independent identically distributed (i.i.d.) binary sequences but, instead, codewords of the outer code, h_m can be greater than the inner code free distance, d_f^i.

[4] This assumption permits neglecting the length of error events compared to N, which also assumes that the number of ways j input sequences producing j error events can be arranged in a register of length N is $\binom{N/p}{j}$]. The ratio N/p derives from the fact that the code has rate p/n, and thus N bits corresponds to N/p input words or, equivalently, trellis steps.
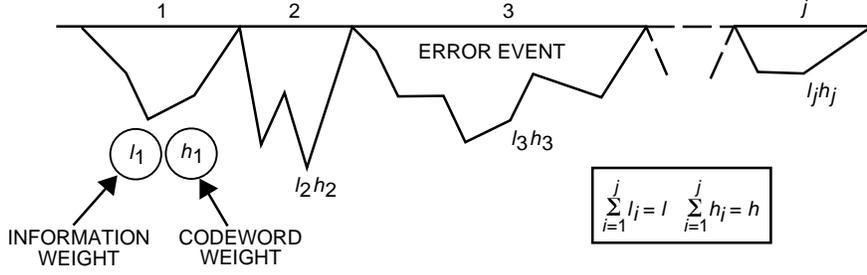
Fig. 6. A code sequence in $A_{l,h,j}$.

where n_M, the largest number of error events concatenated in a codeword of weight h and generated by a weight l input sequence, is a function of h and l that depends on the encoder, as we will see later.

Let us return now to the block code equivalent to the SCCC. Using the previous result of Eq. (12) with $j = n^i$ for the inner code, and the analogous one, $j = n^o$, for the outer code,[5]

$$A_{w,l}^{C_o} \sim \sum_{n^o=1}^{n_M^o} \binom{N/p}{n^o} A_{w,l,n^o}^o \tag{13}$$

and substituting them into Eq. (4), we obtain the coefficient $A_{w,h}^{C_S}$ of the serially concatenated block code equivalent to the SCCC in the form

$$A_{w,h}^{C_S} \sim \sum_{l=d_f^o}^{N} \sum_{n^o=1}^{n_M^o} \sum_{n^i=1}^{n_M^i} \frac{\binom{N/p}{n^o}\binom{N/p}{n^i}}{\binom{N}{l}} A_{w,l,n^o}^o A_{l,h,n^i}^i \tag{14}$$

where d_f^o is the free distance of the outer code. By free distance d_f we mean the minimum Hamming weight of error events for convolutional CCs and the minimum Hamming weight of codewords for block CCs.

We are interested in large interleaver lengths and thus use for the binomial coefficient the asymptotic approximation

$$\binom{N}{n} \sim \frac{N^n}{n!}$$

Substitution of this approximation in Eq. (14) yields

$$A_{w,h}^{C_S} \sim \sum_{l=d_f^o}^{N} \sum_{n^o=1}^{n_M^o} \sum_{n^i=1}^{n_M^i} N^{n^o+n^i-l} \frac{l!}{p^{n^o+n^i} n^o! n^i!} A_{w,l,n^o}^o A_{l,h,n^i}^i \tag{15}$$

Finally, substituting Eq. (15) into Eq. (10) gives the bit-error probability bound in the form

---

[5] In the following, superscripts o and i will refer to quantities pertaining to outer and inner code, respectively.

$$P_b(e) \overset{\sim}{\leq} \sum_{h=h_m}^{N/R_c^i} e^{-hR_cE_b/N_0} \sum_{w=w_m^o}^{NR_c^o} \sum_{l=d_f^o}^{N} \sum_{n^o=1}^{n_M^o} \sum_{n^i=1}^{n_M^i} N^{n^o+n^i-l-1} \frac{l!}{p^{n^o+n^i-1}n^o!n^i!} \frac{w}{k} A_{w,l,n^o}^o A_{l,h,n^i}^i \qquad (16)$$

Using Expression (16) as the the starting point, we will obtain some important design considerations. The bound, Expression (16), to the bit-error probability is obtained by adding terms of the first summation with respect to the SCCC weights h. The coefficients of the exponentials in h depend, among other parameters, on N. For large N, and for a given h, the dominant coefficient of the exponentials in h is the one for which the exponent of N is maximum. Define this maximum exponent as

$$\alpha(h) \triangleq \max_{w,l}\{n^o + n^i - l - 1\} \qquad (17)$$

Evaluating $\alpha(h)$ in general is not possible without specifying the CCs. Thus, we will consider two important cases for which general expressions can be found.

A. The Exponent of *N* for the Minimum Weight

For large values of E_b/N_0, the performance of the SCC is dominated by the first term of the summation in h, corresponding to the minimum value $h = h_m$. Remembering that, by definition, n_M^i and n_M^o are the maximum number of concatenated error events in codewords of the inner and outer code of weights h_m and l, respectively, the following inequalities hold true:

$$n_M^i \leq \left\lfloor \frac{h_m}{d_f^i} \right\rfloor \qquad (18)$$

$$n_M^o \leq \left\lfloor \frac{l}{d_f^o} \right\rfloor \qquad (19)$$

and

$$\alpha(h_m) \leq \max_l \left\{ \left\lfloor \frac{h_m}{d_f^i} \right\rfloor + \left\lfloor \frac{l}{d_f^o} \right\rfloor - l - 1 \right\} = \left\lfloor \frac{h_m}{d_f^i} \right\rfloor + \left\lfloor \frac{l_m(h_m)}{d_f^o} \right\rfloor - l_m(h_m) - 1 \qquad (20)$$

where $l_m(h_m)$ is the minimum weight l of codewords of the outer code yielding a codeword of weight h_m of the inner code, and $\lfloor x \rfloor$ means "integer part of x." In most cases, $l_m(h_m) < 2d_f^o$ and $h_m < 2d_f^i$, so that $n_M^i = n_M^o = 1$ and Eq. (20) becomes[6]

$$\alpha(h_m) = 1 - l_m(h_m) \leq 1 - d_f^o \qquad (21)$$

The result, Eq. (21), shows that the exponent of N corresponding to the minimum weight of SCCC codewords is always negative for $d_f^o \geq 2$, thus yielding an interleaver gain at high E_b/N_0. Substitution of the exponent $\alpha(h_m)$ into Expression (16) truncated to the first term of the summation in h yields

---

[6] This will be seen in the examples that follow and corresponds to the most favorable situation.

$$\lim_{\frac{E_b}{N_0} \to \infty} P_b(e) \stackrel{\sim}{\leq} B_m N^{1-d_f^o} \exp\left(\frac{-h_m R_c E_b}{N_0}\right) \qquad (22)$$

where the constant B_m is

$$B_m = \frac{A_{l_m(h_m),h_m,1}^i \, [l_m(h_m)]!}{kp} \sum_{w \in \mathcal{W}_m} w A_{w,l_m(h_m),1}^o$$

and $\mathcal{W}_m$ is the set of input weights w that generates codewords of the outer code with weight $l_m(h_m)$.

Expression (22) suggests the following conclusions:

(1) For the values of E_b/N_0 and N where the SCCC performance is dominated by its free distance $d_f^{Cs} = h_m$, increasing the interleaver length yields a gain in performance.

(2) To increase the interleaver gain, one should choose an outer code with a large d_f^o.

(3) To improve the performance with E_b/N_0, one should choose an inner and outer code combination such that h_m is large.

These conclusions do not depend on the structure of the CCs, and thus they apply for both recursive and nonrecursive encoders.

However, the curves of Fig. 3, showing the performance of the various SCBCs of Example 1 with increasing interleaver length, also show a different phenomenon: For a given E_b/N_0, there seems to be a minimum value of N that forces the bound to diverge. In other words, there seem to be coefficients of the exponents in h, for $h > h_m$, that increase with N. To investigate this phenomenon, we will evaluate the largest exponent of N, defined as

$$\alpha_M \stackrel{\triangle}{=} \max_h \{\alpha(h)\} = \max_{w,l,h} \{n^o + n^i - l - 1\} \qquad (23)$$

This exponent will permit one to find the dominant contribution to the bit-error probability for $N \to \infty$.

B. The Maximum Exponent of *N*

We need to treat the cases of nonrecursive and recursive inner encoders separately. As we will see, nonrecursive encoders and block encoders show the same behavior.

1. Block and Nonrecursive Convolutional Inner Encoders. Consider the inner code and its impact on the exponent of N in Eq. (23). For a nonrecursive inner encoder, we have $n_M^i = l$. In fact, every input sequence with weight 1 generates a finite-weight error event, so that an input sequence with weight l will generate, at most, l error events corresponding to the concatenation of l error events of input weight 1. Since the uniform interleaver generates all possible permutations of its input sequences, this event will certainly occur. Thus, from Eq. (23) we have

$$\alpha_M = n_M^o - 1 \geq 0$$

and interleaving gain is not allowed. This conclusion holds true for both SCCC employing a nonrecursive inner encoder and for all SCBCs, since block codes have codewords corresponding to input words with

weight equal to 1. For those SCCs, we *always* have, for some h, coefficients of the exponential in h of Expression (16) that increase with N, and this explains the divergence of the bound arising, for each E_b/N_0, when the coefficients increasing with N become dominant.

2. Recursive Inner Encoders. In [8], we proved that, for recursive convolutional encoders, the minimum weight of input sequences generating error events is 2. As a consequence, an input sequence of weight l can generate at most $\lfloor \frac{l}{2} \rfloor$ error events.

Assuming that the inner encoder of the SCCC is recursive, the maximum exponent of N in Eq. (23) becomes

$$\alpha_M = \max_{w,l} \left\{ n_M^o + \left\lfloor \frac{l}{2} \right\rfloor - l - 1 \right\} = \max_{w,l} \left\{ n_M^o - \left\lfloor \frac{l+1}{2} \right\rfloor - 1 \right\} \tag{24}$$

The maximization involves l and w, since n_M^o depends on both quantities. In fact, remembering the definition of n_M^o as the maximum number of concatenated error events of codewords of the outer code with weight l generated by input words of weight w, it is straightforward, as in Eq. (19), to obtain

$$n_M^o \le \left\lfloor \frac{l}{d_f^o} \right\rfloor \tag{25}$$

Substituting now the last inequality, Eq. (25), into Eq. (24) yields

$$\alpha_M \le \max_l \left\{ \left\lfloor \frac{l}{d_f^o} \right\rfloor - \left\lfloor \frac{l+1}{2} \right\rfloor - 1 \right\} \tag{26}$$

To perform the maximization of the right-hand side (RHS) of Expression (26), consider first the case of

$$l = q d_f^o$$

where q is an integer, so that

$$\alpha_M \le \max_q \left\{ q - \left\lfloor \frac{q d_f^o + 1}{2} \right\rfloor - 1 \right\} \tag{27}$$

The RHS of Expression (27) is maximized, for $d_f^o \ge 2$, by choosing $q = 1$. On the other hand, for

$$q d_f^o \le l < (q+1) d_f^o$$

the most favorable case is $l = q d_f^o$, which leads us again to the previously discussed situation. Thus, the maximization requires $l = d_f^o$. For this value, on the other hand, we have, from Eq. (25), $n_M^o \le 1$, and the inequality becomes an equality if $w \in \mathcal{W}_f$, where $\mathcal{W}_f$ is the set of input weights w that generates codewords of the outer code with weight $l = d_f^o$. In conclusion, the largest exponent of n is given by

$$\alpha_M = - \left\lfloor \frac{d_f^o + 1}{2} \right\rfloor \tag{28}$$

The value of α_M in Eq. (28) shows that the exponents of N in Expression (16) are always negative integers. Thus, for all h, the coefficients of the exponents in h decrease with N, and we always have an interleaver gain.

As in [6], denoting by $d^i_{f,eff}$ the minimum weight of codewords of the inner code generated by weight-2 input sequences, we obtain a different weight $h(\alpha_M)$ for even and odd values of d^o_f. For even d^o_f, the weight $h(\alpha_M)$ associated to the highest exponent of N is given by

$$h(\alpha_M) = \frac{d^o_f d^i_{f,eff}}{2}$$

since it is the weight of an inner codeword that concatenates $d^o_f/2$ error events with weight $d^i_{f,eff}$. Substituting the exponent α_M into Expression (16), approximated by only the term of the summation in h corresponding to $h = h(\alpha_M)$, yields

$$\lim_{N\to\infty} P_b(e) \overset{\sim}{\leq} B_{even} N^{d^o_f/2} \exp\left[-\frac{d^o_f d^i_{f,eff}}{2} R_c \frac{E_b}{N_0}\right] \tag{29a}$$

where

$$B_{even} = \frac{d^o_f!}{kp^{d^o_f/2}(d^o_f/2)!} \sum_{w\in\mathcal{W}_f} wA^o_{w,d^o_f,1} \leq w_{M,f} N^o_f \frac{d^o_f!}{kp^{d^o_f/2}(d^o_f/2)!} \tag{29b}$$

In Eq. (29b), $w_{M,f}$ is the maximum input weight yielding outer codewords with weight equal to d^o_f, and N^o_f is the number of such codewords.

For d^o_f odd, the value of $h(\alpha_M)$ is given by

$$h(\alpha_M) = \frac{(d^o_f - 3)d^i_{f,eff}}{2} + h^{(3)}_m \tag{30}$$

where $h^{(3)}_m$ is the minimum weight of sequences of the inner code generated by a weight-3 input sequence. In this case, in fact, we have

$$n^i_M = \frac{d^o_f - 1}{2}$$

concatenated error events, of which $n^i_M - 1$ are generated by weight-2 input sequences and one is generated by a weight-3 input sequence.

Thus, substituting the exponent α_M into Expression (16) approximated by keeping only the term of the summation in h corresponding to $h = h(\alpha_M)$ yields

$$\lim_{N\to\infty} P_b(e) \overset{\sim}{\leq} B_{odd} N^{-([d^o_f+1]/2)} \exp\left\{-\left[\frac{(d^o_f-3)d^i_{f,eff}}{2} + h^{(3)}_m\right] R_c \frac{E_b}{N_0}\right\} \tag{31}$$

13

where

$$B_{odd} = \frac{d_f^o!}{kp^{(d_f^o-1)/2}[(d_f^o-3)/2]!} \sum_{w \in \mathcal{W}_f} wA_{w,d_f^o,1}^o \leq w_{M,f} N_f^o \frac{d_f^o!}{kp^{(d_f^o-1)/2}[(d_f^o-3)/2]!} \qquad (32)$$

In cases of d_f^o both even and odd, we can draw from Expressions (29) and (31) a few important design considerations, as follows:

(1) In contrast with the case of block codes and nonrecursive convolutional inner encoders, the use of a recursive convolutional inner encoder always yields an interleaver gain. As a consequence, the first design rule states that *the inner encoder must be a convolutional recursive encoder.*

(2) The coefficient $h(\alpha_M)$ that multiplies the signal-to-noise ratio E_b/N_0 in Expression (16) increases for increasing values of $d_{f,eff}^i$. Thus, we deduce that *the effective free distance of the inner code must be maximized.* Both this and the previous design rule also had been stated for PCCCs [8].[7] As a consequence, the recursive convolutional encoders optimized for use in PCCCs (see the tables in [8] and [9]) can be employed altogether as inner CC in SCCCs. When d_f^o is odd, for special cases it is possible to increase $h(\alpha_M)$ and h_m further by choosing the feedback polynomial of the inner code to have a factor $(1+D)$, yielding $h_m^{(3)} = \infty$. This is discussed for a few examples in Section VI. Note that there are other feedback polynomials such as $(1+D+D^2+D^3+D^4)$ or $(1+D+D^2+D^3+D^4+D^5+D^6)$ yielding $h_m^{(3)} = \infty$.

(3) The interleaver gain is equal to $N^{-(d_f^o/2)}$ for even values of d_f^o and to $N^{-[(d_f^o+1)/2]}$ for odd values of d_f^o. As a consequence, *we should choose,* compatibly with the desired rate R_c of the SCCC, *an outer code with a large and, possibly, odd value of the free distance.*

(4) As to other outer code parameters, N_f^o and $w_{M,f}$ should be minimized. In other words, we should have the minimum number of input sequences generating free distance error events of the outer code, and their input weights should be minimized. Since nonrecursive encoders have error events with $w = 1$ and, in general, less input errors associated with error events at free distance [10], it can be convenient *to choose as an outer code a nonrecursive encoder* with minimum N_f^o and $w_{M,f}$. Conventional nonrecursive convolutional codes found in books (see, for example, [11]) are appropriate.

C. Examples Confirming the Design Rules

To confirm the design rules obtained asymptotically, i.e., for large signal-to-noise ratios and large interleaver lengths N, we evaluate the upper bound, Expression (16), to the bit-error probability for several block and convolutional SCCs with different interleaver lengths, and compare their performances with those predicted by the design guidelines.

1. Serially Concatenated Block Codes. We consider three different SCBCs obtained, as follows: The first is the $(7m, 3m, N)$ SCBC described in Example 2; the second is a $(15m, 4m, N)$ SCBC using as outer code a $(5, 4)$ parity-check code and as inner code a $(15, 5)$ Bose–Chaudhuri–Hocquenghem (BCH) code; and the third is a $(15m, 4m, N)$ SCBC using as outer code a $(7, 4)$ Hamming code and as inner code a $(15, 7)$ BCH code. Note that the second and third SCBCs have the same rate, 4/15. The outer, inner, and SCBC code parameters introduced in the design analysis of Section III are listed in Table 2. In Figs. 3, 7, and 8, we plot the bit-error probability bounds for SCBCs 1,2, and 3 of Table 2.

Code SCBC1 has $d_f^o = 2$; thus, from Eq. (21), we expect an interleaver gain going as N^{-1}. This is confirmed by the curves of Fig. 3, which, for a fixed and sufficiently large signal-to-noise ratio, show a

---

[7] For PCCCs, however, both CCs had to comply with those design rules.

**Table 2. Design parameters of CCs and SCBCs for three
serially concatenated block codes.**

Code	Outer code			Inner code				SCBC	
	Code type	w_m^o	d_f^o	Code type	w_m^i	d_f^i	$d_{f,eff}^i$	h_m	$\alpha(h_m)$
SCBC1	Parity check (4,3)	1	2	Hamming (7,4)	1	3	3	3	-1
SCBC2	Parity check (5,4)	1	2	BCH (15,5)	1	7	7	7	-1
SCBC3	Hamming (7,4)	1	3	BCH (15,7)	1	5	5	5	-2
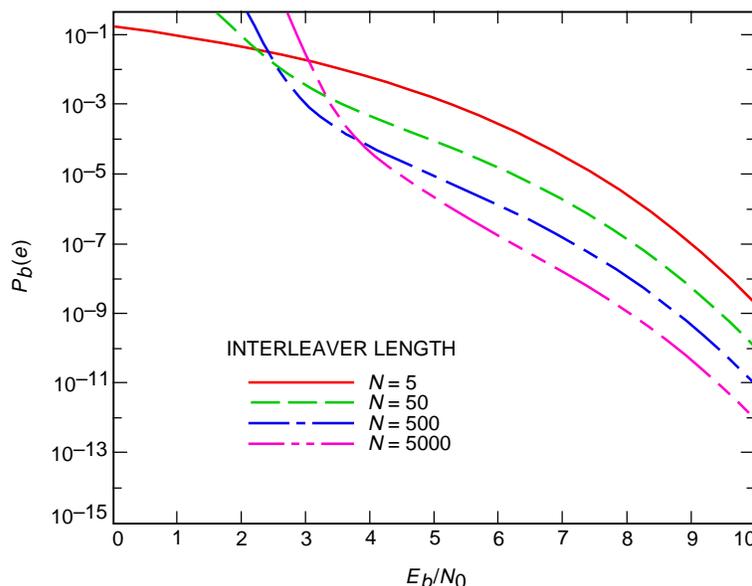
Fig. 7. Analytical bounds for SCBC2.

decrease in $P_b(e)$ of a factor of 10 when N passes from 4 to 40, from 40 to 400, and from 400 to 4000. Moreover, from Expression (22), we expect, in each curve for $\ln P_b(e)$, a slope with E_b/N_0 as $-h_m R_c$. From Table 2, we know that $R_c = 3/7, h_m = 3$, so that $P_b(e)$ should decrease by a factor of $e^{h_m R_c} = 3.6$ when the signal-to-noise ratio increases by 1 (not in dB). This behavior fully agrees with the curves of Fig. 3. Finally, the curves of Fig. 3 show a divergence of the bound at lower E_b/N_0 for increasing N. This is due to coefficients of terms with $h > h_m$ in Expression (16) that increase with N and whose influence becomes more important for larger N.

Code SCBC2 has $d_f^o = 2$; thus, from Eq. (21), we expect the same interleaver gain as for SCBC1, i.e., N^{-1}. This is confirmed by the curves of Fig. 7. This code, however, has a larger minimum distance $h_m = 7$, and a rate $R_c = 4/15$. Thus, we expect a steeper descent of $P_b(e)$ with E_b/N_0. More precisely, we expect a decrease by a factor of 6.5 when the signal-to-noise ratio increases by 1. This, too, is confirmed by the curves, which also show the bound divergence predicted in the analysis of Section III.

Code SCBC3 has $d_f^o = 3$; thus, from Eq. (21), we expect a larger interleaver gain than for SCBC1 and SCBC2, i.e., N^{-2}. This is confirmed by the curves of Fig. 8, which, for a fixed and sufficiently large signal-to-noise ratio, show a decrease in $P_b(e)$ of a factor of 100 when N passes from 7 to 70, from 70 to 700, and from 700 to 7000. This code has a minimum distance $h_m = 5$ and a rate $R_c = 4/15$, which
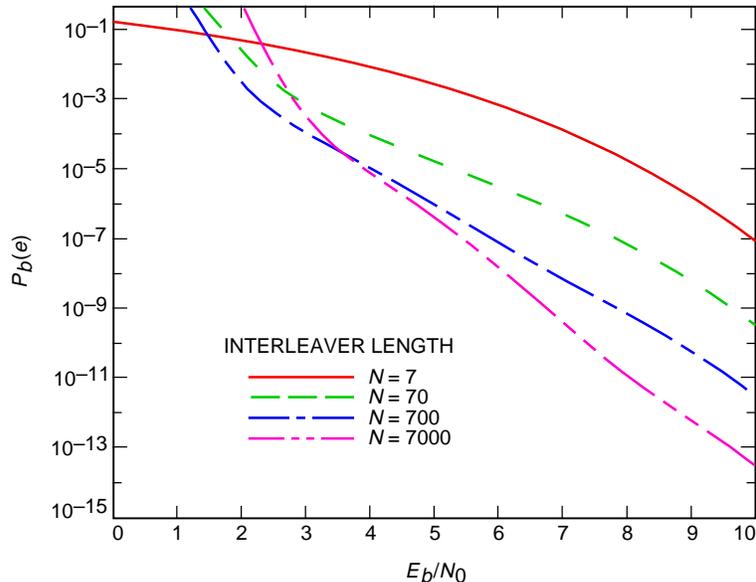
Fig. 8. Analytical bounds for SCBC3.

means a descent of $P_b(e)$ with E_b/N_0 by a factor of 3.8 when the signal-to-noise ratio increases by 1. This, too, is confirmed by the curves. As to the bound divergence, we notice a slightly different behavior with respect to previous cases. The curve with $N = 7000$, in fact, denotes a strong influence of coefficients increasing with N for E_b/N_0 lower than 7.

2. Serially Concatenated Convolutional Codes. We consider four different SCCCs obtained as follows: The first, SCCC1, is a $(3,1,N)$ SCCC (the same as Example 3), using as outer code a four-state $(2,1)$ recursive, systematic convolutional encoder and as inner code a four-state $(3,2)$ recursive, systematic convolutional encoder. The second, SCCC2, is a $(3,1,N)$ SCCC, using as outer code the same four-state $(2,1)$ recursive, systematic convolutional encoder as SCCC1, and as inner code a four-state $(3,2)$ nonrecursive convolutional encoder. The third, SCCC3, is a $(3,1,N)$ SCCC (the same as Example 3), using as outer code a four-state $(2,1)$ nonrecursive, convolutional encoder, and as inner code the same four-state $(3,2)$ recursive, systematic convolutional encoder as SCCC1. Finally, the fourth, SCCC4, is a $(6,2,N)$ SCCC using as outer code a four-state $(3,2)$ nonrecursive convolutional encoder, and as inner code a four-state $(6,3)$ recursive, systematic convolutional encoder obtained by using three times the four-state $(2,1)$ recursive, systematic convolutional encoders in Table 1.

The outer, inner, and SCCC code parameters introduced in the design analysis in Section III are listed in Table 3. In this table, the CCs are identified through the descriptions of Table 1. In Figs. 5, 9, 10, and 11, we plot the bit-error probability bounds for SCCCs 1,2,3, and 4 of Table 3, with input information block lengths $R_c^o N = 100, 200, 300, 400, 500$, and 1000.

Consider first the SCCCs employing as inner CCs recursive, convolutional encoders as suggested in Section III. They are SCCC1, SCCC3, and SCCC4. Code SCCC1 has $d_f^o = 5$; thus, from Expression (31), we expect an interleaver gain behaving as N^{-3}. This is fully confirmed by the curves of Fig. 5, which, for a fixed and sufficiently large signal-to-noise ratio, show a decrease in $P_b(e)$ of a factor of 1000 when N passes from 200 to 2000. For an even more accurate confirmation, one can compare the interleaver gain for every pair of curves in the figure. Moreover, from Expression (31), we expect in each curve for $\ln P_b(e)$ a slope with E_b/N_0 as $-h(\alpha_M)R_c$. From Table 3, we know that $R_c = 1/3$ and $h(\alpha_M) = 7$, so that $P_b(e)$ should decrease by a factor of 10.3 when the signal-to-noise ratio increases by 1. This behavior fully agrees with the curves of Fig. 5. Finally, the curves of Fig. 5 do not show a divergence of the bound at

Table 3. Design parameters of CCs and SCCCs for four SCCCs.

Code	Outer code			Inner code				SCCC			
	Code type	w_m^o	d_f^o	Code type	w_m^i	d_f^i	$d_{f,eff}^i$	h_m	$\alpha(h_m)$	$h(\alpha_M)$	α_M
SCCC1	Rate 1/2 recursive	2	5	Rate 2/3 recursive	2	3	4	5	−4	7	−3
SCCC2	Rate 1/2 recursive	2	5	Rate 2/3 nonrecursive	1	3	4	5	−4	—	—
SCCC3	Rate 1/2 nonrecursive	1	5	Rate 2/3 recursive	2	3	4	5	−4	7	−3
SCCC4	Rate 2/3 nonrecursive	1	3	Rate 1/2 recursive	2	5	6	5	−2	5	−2
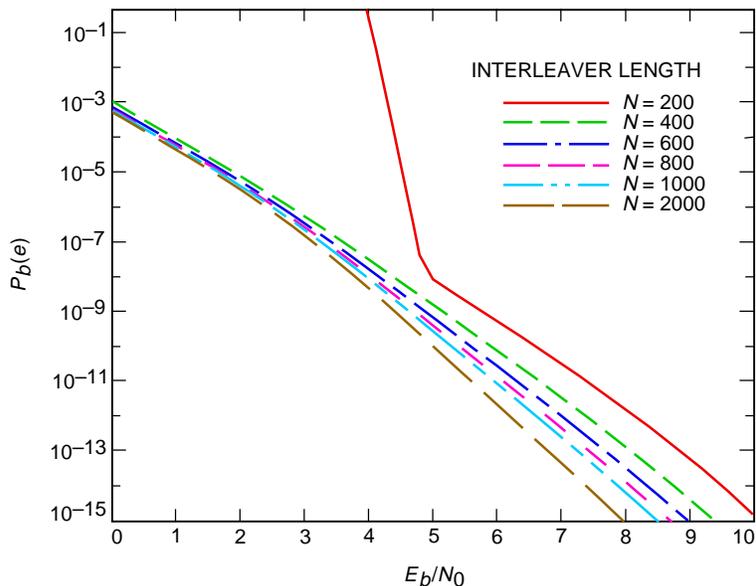
Fig. 9. Analytical bounds for SCCC2.

lower E_b/N_0 for increasing N. This is due to the choice of a recursive encoder for the inner code, which guarantees that all coefficients $\alpha(h)$ decrease with N.

Code SCCC3 differs from SCCC1 only in the choice of a nonrecursive outer encoder, which is a four-state encoder (see Tables 1 and 3) with the same d_f^o as for SCCC1, but with $w_m^o = 1$ instead of $w_m^o = 2$. From the design conclusions, we expect a slightly better behavior from this SCCC. This is confirmed by the performance curves of Fig. 10, which present the same interleaver gain and slope as those of SCCC1 but have a slightly lower $P_b(e)$ (the curves for SCCC3 are translated versions of those of SCCC1 by 0.1 dB).

Code SCCC4 employs the same CCs as SCCC2 but reverses their order. It uses as outer code a rate 2/3 nonrecursive convolutional encoder, and as inner code a rate 1/2 recursive convolutional encoder. As a consequence, it has a lower $d_f^o = 3$ and a higher $\alpha_M = -2$. Thus, from Expression (31), we expect a lower interleaver gain than for SCCC1 and SCCC3 as N^{-2}. This is confirmed by the curves of Fig. 11, which, for a fixed and sufficiently large signal-to-noise ratio, show a decrease in $P_b(e)$ of a factor of 100 when N passes from 150 to 1500. As to the slope with E_b/N_0, this code has the same $-h(\alpha_M)R_c$ as SCCC1 and SCCC3 and, thus, the same slope. On the whole, SCCC4 loses more than 2 dB in coding gain with respect to SCCC3. This result confirms the design rule suggesting the choice of an outer code with d_f^o as large as possible.
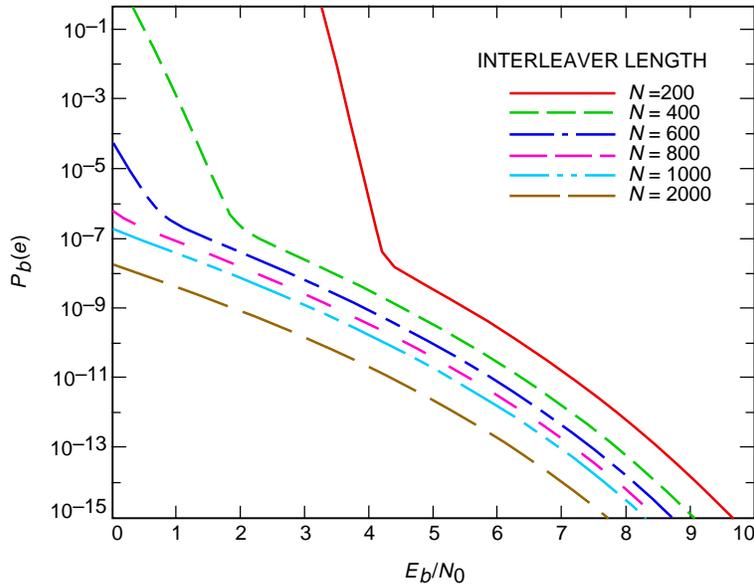
Fig. 10. Analytical bounds for SCCC3.
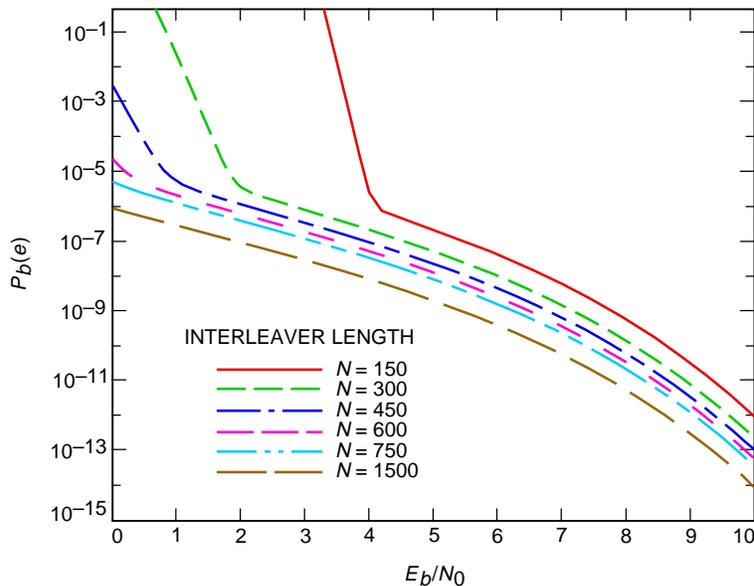
Fig. 11. Analytical bounds for SCCC4.

Finally, let us consider code SCCC2, which differs from SCCC1 in the choice of a nonrecursive inner encoder, with the same parameters but with the crucial difference of $w_m^i = 1$. Its bit-error probability curves are shown in Fig. 9. They confirm the predictions of Section III. We see, in fact, that for low signal-to-noise ratios, say below 3, no interleaver gain is obtained. This is because the performance is dominated by the exponent $h(\alpha_M)$, whose coefficient increases with N. On the other hand, for larger signal-to-noise ratios, where the dominant contribution to $P_b(e)$ is the exponent with the lowest value of h_m, the interleaver gain makes its appearance. From Expression (22), we foresee a gain as N^{-4}, meaning four orders of magnitude for N passing from 100 to 1000. Curves in Fig. 9 show a smaller gain (slightly higher than 1/1000), which is, on the other hand, rapidly increasing with E_b/N_0.

IV. Comparison Between Parallel and Serially Concatenated Codes

In this section, we will use the bit-error probability bounds previously derived to compare the performance of parallel (turbo codes [6]) and serially concatenated block and convolutional codes.

A. Parallel and Serially Concatenated Block Codes

To obtain a fair comparison, we have chosen the following PCBC and SCBC: The PCBC has parameters $(11m, 3m, N)$ and employs two equal $(7,3)$ systematic cyclic codes with generator $g(D) = (1 + D)(1 + D + D^3)$; the SCBC, instead, is a $(15m, 4m, N)$ SCCC obtained by the concatenation of the $(7, 4)$ Hamming code with a $(15, 7)$ BCH code.

They have almost the same rates ($R_{C_S} = 0.266$ and $R_{C_P} = 0.273$), and have been compared choosing the interleaver length in such a way that the decoding delay due to the interleaver, measured in terms of input information bits, is the same. As an example, to obtain a delay equal to 12 input bits, we must choose an interleaver length $N = 12$ for the PCBC and $N = 12/R_C^o = 21$ for the SCBC.

The results are shown in Fig. 12, where we plot the bit-error probability versus the signal-to-noise ratio E_b/N_0 for various input delays. The results show that, for low values of the delay, the performances are almost the same. On the other hand, increasing the delay (and thus the interleaver length N) yields a significant interleaver gain for the SCBC and almost no gain for the PCBC. The difference in performance is 3 dB at $P_b(e) = 10^{-6}$ in favor of the SCBC.
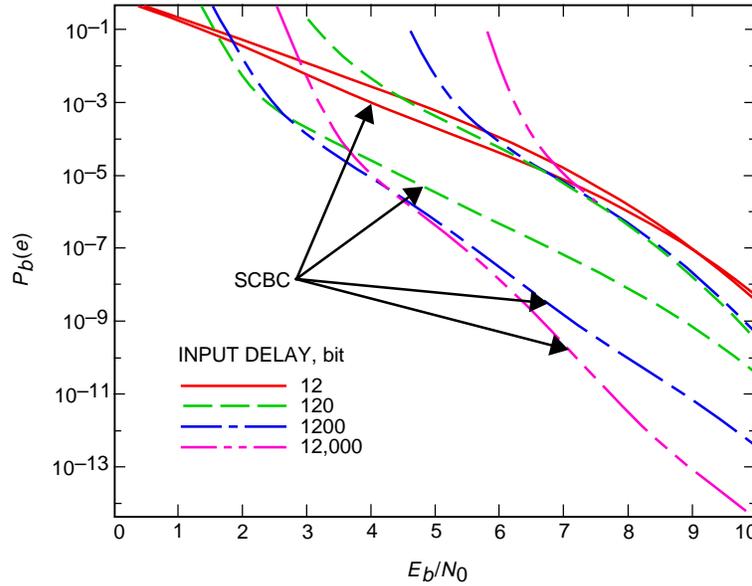
Fig. 12. Comparison of SCBC and PCBC with various interleaver lengths chosen so as to yield the same input decoding delay.

B. Parallel and Serially Concatenated Convolutional Codes

To obtain a fair comparison, we have chosen the following PCCC and SCCC: The PCCC is a rate $1/3$ code obtained concatenating two equal rate $1/2$, four-state systematic recursive convolutional codes with a generator matrix as in the first row of Table 1. The SCBC is a rate $1/3$ code using as an outer code the same rate $1/2$, four-state code as in the PCCC and, as an inner code, a rate $2/3$, four-state systematic recursive convolutional code with a generator matrix as in the third row of Table 1. Also, in this case, the interleaver lengths have been chosen so as to yield the same decoding delay, due to the interleaver,

in terms of input bits. The results are shown in Fig. 13, where we plot the bit-error probability versus the signal-to-noise ratio E_b/N_0 for various input delays.

The results show the great difference in the interleaver gain. In particular, the PCCC shows an interleaver gain going as N^{-1}, whereas the interleaver gain of the SCCC, as from Expression (31), goes as $N^{-(d_f^o+1)/2} = N^{-3}$, since the free distance of the outer code is equal to 5, which is odd. This means, for $P_b(e) = 10^{-11}$, a gain of more than 2 dB in favor of the SCCC.

Previous comparisons have shown that serial concatenation is advantageous with respect to parallel concatenation in terms of maximum-likelihood performance. For long interleaver lengths, this significant result remains a theoretical one, as maximum-likelihood decoding is an almost impossible achievement. For parallel concatenated codes (turbo codes), iterative decoding algorithms have been proposed that yield performance close to optimum, with limited complexity. In the following section, we show that this is also possible for serially concatenated codes and that the performance gain is maintained.
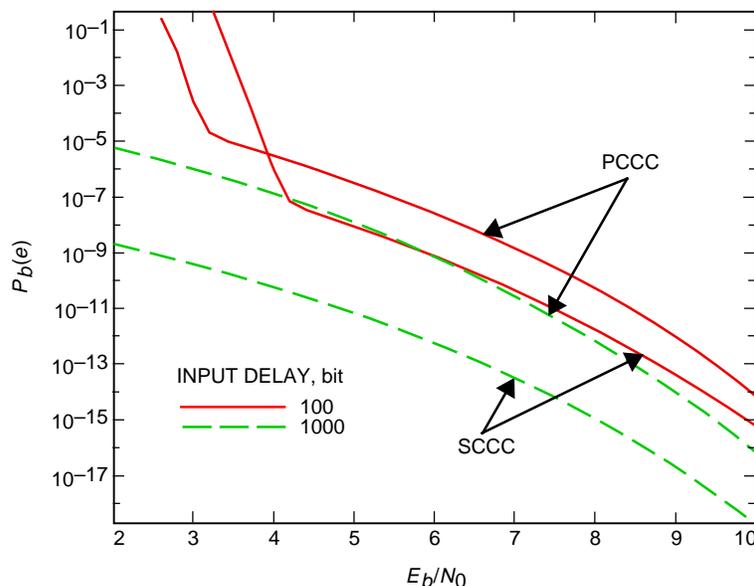
Fig. 13. Comparison of SCCC and PCCC with four-state CCs.

V. Iterative Decoding of Serially Concatenated Codes

In Sections II and III, we have shown by examples and analytical findings that SCCCs can outperform PCCCs when decoded using an ML algorithm. In practice, however, ML decoding of these codes with large N is an almost impossible achievement. Thus, to acquire a practical significance, this theoretical result needs the support of a decoding algorithm of the same order of complexity as turbo decoding, yet retaining the performance advantages. In this section, we will present a new iterative algorithm whose complexity is not significantly higher than that needed to separately decode the two CCs. Because of their importance in applications, all examples will refer to SCCCs, although the decoding algorithm can be applied to SCBCs as well. The core of the new decoding procedure consists of a maximum a posteriori (MAP) decoding algorithm applied to the CCs. The functionality of the MAP decoder to be used for SCCCs is sensibly different from those needed in the PCCC decoding algorithm, as we will show in the following. To permit a continuous decoding of the received sequence, we will use a modified version of the
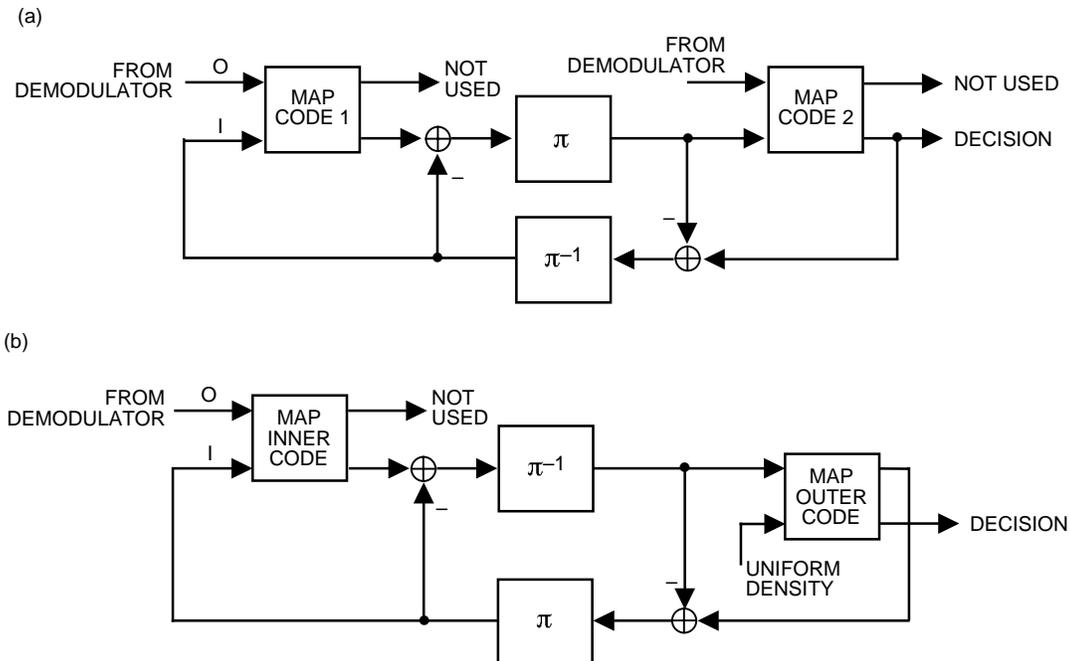
Fig. 14. Iterative decoding algorithm for serially concatenated convolutional codes: (a) PCCC and (b) SCCC.

sliding-window MAP algorithm described in [12].[8] A functional diagram of the new iterative decoding algorithm for SCCCs is presented in Fig. 14, where we also show the algorithm that decodes turbo codes in order to enlighten analogies as well as differences.

Let us explain how the algorithm works, according to the blocks of Fig. 14. The blocks labeled "MAP" are drawn with two inputs and two outputs. The input labeled O represents the logarithm of the probability density function (LPDF) of the unconstrained output symbols of the encoder, while that labeled I represents the LPDF of unconstrained input symbols. Similarly, the outputs represent the same quantities conditioned to the code constraint as they are evaluated by the MAP decoding algorithm. Differently from the iterative decoding algorithm employed for turbo decoding, in which the MAP algorithm only computes the LPDF of input symbols conditioned on the code constraint based on the unconstrained LPDF of input symbols, we fully exploit here the potential of the MAP algorithm. It can, in fact, update the LPDFs of both the input and output symbols based on the code constraints.

We assume that the pair (i, o) of symbols, labeling each branch of the code trellis, is independent at the input of the MAP decoder, so that their joint LPDF is given by

$$LPDF(i, o) = LPDF(i) + LPDF(o)$$

During the first iteration of the SCCC algorithm, the "MAP inner code" block is fed with the de-modulator soft output, consisting of the LPDF of symbols received from the channels, i.e., of the output symbols of the inner encoder. The LPDF is processed by the first MAP decoder that computes the LPDF relative to the input symbols conditioned on the inner code constraints. This information, from which we subtract the unconstrained input LPDF to obtain the "extrinsic" information as is done for turbo

---

[8] This has been generalized in S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "MAP Decoding Algorithms as Building Blocks for Iterative Decoding of Concatenated Codes," in preparation, 1996.

decoding, is passed through the inverse interleaver[9] (block labeled "π^{-1}"). As the input symbols of the inner code, after inverse interleaving, correspond to the output symbols of the outer code, they are sent to the "MAP outer code" block in the upper entry, which corresponds to output symbols. The outer MAP decoder, in turn, processes the LPDF of the unconstrained output symbols and computes the LPDF of both output and input symbols based on the code constraints. The LPDF of the input symbols (the MAP information) will be used in the final iteration to recover the information bits, whereas the LPDF of output symbols, after subtraction and interleaving, is fed back to the MAP inner decoder to start the second iteration.

A. Performance of the Decoding Algorithm

To show the performance of the SCCC decoded using the new algorithm, we have simulated a rate 1/3 SCCC employing two four-state recursive convolutional codes, the first (outer code) with rate 1/2 and the second (inner code) with rate 2/3, joined by an interleaver of length $N = 2048$. Since the interleaver operates on coded sequences produced by the outer rate 1/2 encoder, its length of 2048 bits corresponds to a delay of 1024 information bits. The simulation results are shown as dashed curves in Fig. 15 in terms of bit-error probability versus E_b/N_0 for a number of iterations ranging from 1 to 7. The nice convergence of the decoding algorithm is manifest.

In the figure, we also show (solid lines) the simulation results pertaining to a rate 1/3 PCCC formed by two equal four-state rate 1/2 recursive convolutional constituent codes joined by an interleaver of length $N = 1024$, which induces the same delay on the information bits, using the iterative turbo-decoding algorithm and the same numbers of iterations. The comparison between the two sets of curves is striking, as they show the large advantage of SCCC over PCCC in terms of interleaving gain. In fact, the change
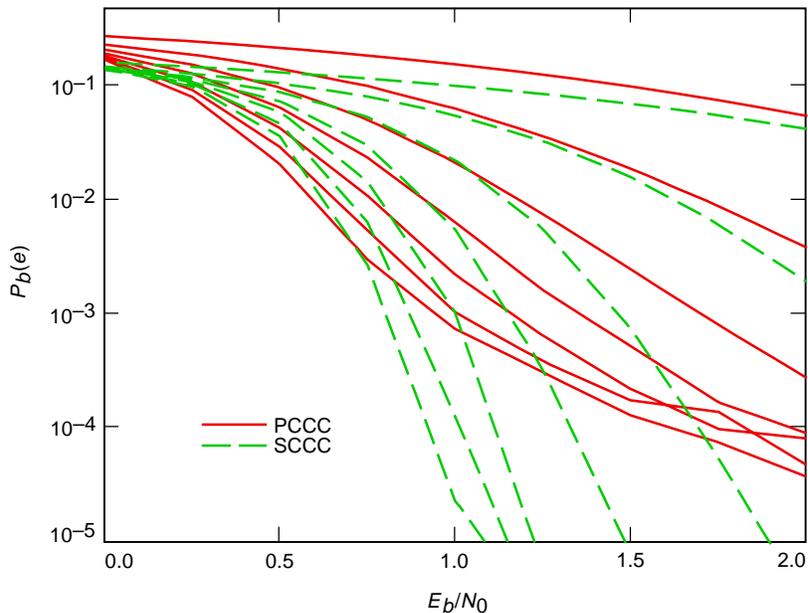
Fig. 15. Comparison of simulated performance between two rate 1/3 serial and parallel concatenated convolutional codes. The dashed-line curves refer to code SCCC1 of Table 3; the solid-line curves refer to a turbo code using two equal constituent codes (first row of Table 1).

---

[9] To simplify the description, we assume that the interleaver acts on symbols instead of bits. In the actual decoder, we deal with bit LPDF and the interleaver.

of slope that characterizes the error probability curves of turbo codes does not take place with SCCC, or, at least, it appears more than one order of magnitude later. The gain in E_b/N_0 is more than 0.65 dB at $P_b(e) = 10^{-4}$.

In Section III, we proved that, asymptotically, the best choice between outer and inner code rates consists of an outer code with a free distance as large as possible. As a consequence, one should choose as an outer code the more powerful code, and, consequently, the code with the lower rate.

This behavior does not hold for the iterative decoding algorithm working at low signal-to-noise ratios, where the bootstrap effect coming from the inner code, which is the first to be decoded in the decoding algorithm, requires one to reverse the choice of codes. This was already pointed out in the optimization of the serial concatenation of convolutional codes without an interleaver in [3]. To give heuristic evidence to this guess, we have simulated the decoding algorithm for two different rate 1/3 SCCCs. The first uses code SCCC1 of Table 3. The second SCCC instead reverses the inner and outer encoders, i.e., it uses as an outer code the four-state, rate 2/3 code of Table 1, third row, and as an inner code a rate 3/6 four-state code obtained by using three times the rate 1/2 code of Table 1, first row. The performances are shown in Fig. 16, where the bit-error probability for the two SCCCs are plotted versus E_b/N_0 for different values of iterations of the decoding algorithm. It can be seen that the second SCCC, whose inner code is more powerful, has better performance at a medium-high bit-error probability. However, when the signal-to-noise ratio increases, the curves change slope and the first SCCC overtakes the second one, as anticipated by the design considerations, because its outer code has a larger d_f^o.
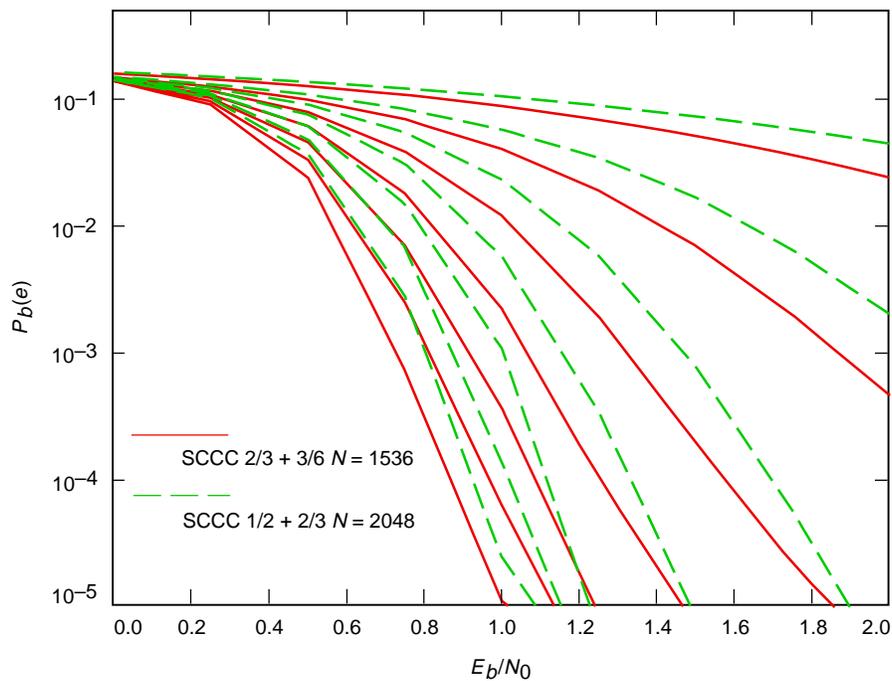
Fig. 16. Simulated performance of two SCCCs. The dashed-line curves refer to code SCCC1 of Table 3; the solid-line curves refer to an outer code with rate 2/3 (third row of Table 1) and an inner code of rate 3/6 (three times the code in the first row of Table 1).

VI. Examples of Serially Concatenated Codes for Deep-Space Communications

In this section, we propose simple examples of rate 1/3, 1/4, and 1/6 SCCCs that are constructed using a rate 1/2, four-state nonrecursive convolutional code, with generator $G_0(D) = (1+D+D^2, 1+D^2)$ having $d_f^o = 5$ as an outer code, and a rate 2/3, 1/2, or 1/3, four-state recursive convolutional code as an inner code. First we consider a rate 1/2 inner code. If we select the first code in Table 1, we obtain $h(\alpha_M) = 11$ using Eq. (30) and we find that $h_m = 7$ (minimum distance). However, for SCCCs with odd d_f^o and low-complexity inner code, it may be possible to obtain larger $h(\alpha_M)$ and h_m, if we choose an inner encoder whose feedback polynomial contains $(1 + D)$ as a factor. This choice produces a very large output weight for odd input-weight sequences, thus yielding $h_m^{(odd)} = \infty$ and, in particular, $h_m^{(3)} = \infty$. As a consequence, the design parameter $h(\alpha_M)$ cannot be obtained from Eq. (30) but must be computed using $h(\alpha_M) = [(d_f^o + 1)d_{f,eff}^i]/2$ since the input sequence to the inner code now has an effective minimum weight $d_f^o + 1$. This may produce a larger value for $h(\alpha_M)$ and h_m, still with an interleaver gain of $N^{-[(d_f^o+1)/2]}$.

Thus, we examine two such codes, $G_1(D) = (1, [(1 + D + D^2)/(1 + D)])$ and $G_2(D) = (1, [(1 + D + D^2)/(1 + D^2)])$, having $(1 + D)$ as a factor in the feedback polynomial. For $G_1(D)$, we obtained $h(\alpha_M) = 12$ and $h_m = 8$. For $G_2(D)$, we have $h(\alpha_M) = 15$ and $h_m = 8$. Both $G_1(D)$ and $G_2(D)$, in serial concatenation with $G_0(D)$, were simulated using iterative decoding with an information block of 16,384 bits and 12 iterations.. The code with $G_1(D)$ has shown slightly better performance around a bit-error rate (BER) of 10^{-7} or higher. For example, at BER $= 10^{-6}$, $G_1(D)$ is better than $G_2(D)$ by 0.07 dB. Using transfer function bounds $G_2(D)$ results in better performance than does $G_1(D)$ at higher signal-to-noise ratios (SNRs). At a very high SNR, the performances of both codes are almost the same since h_m is the same for both of them. Using the MAP decoder, $G_1(D)$ by itself performed better than all rate 1/2, four-state codes at low SNRs. It is worthwhile exploring other differences between $G_1(D)$ and $G_2(D)$. In [14], a parameter η was suggested as a quality criterion for a feedback convolutional encoder. This parameter represents the "rate of dependency" of an output parity bit on input bits. Let the output parity bit at time k be the modulo 2 sum of certain $n(k)$ input bits, depending on the code structure. Then $\eta = \lim_{k\to\infty}[n(k)/k]$. The parameter η for $G_1(D)$ is 1 (this is usually true if the feedback polynomial is $1+D$ and the weight of the feedforward polynomial is odd), where for $G_2(D)$ the parameter η is 0.5. Note that for the first code in Table 1 η is 2/3 and for nonrecursive convolutional codes η is 0.

The simulation performance of a rate 1/4 SCCC using $G_0(D)$ as an outer code and $G_1(D)$ as an inner code is shown in Fig. 17. The simulation performance of a rate 1/4 PCCC using two four-state codes, $(1, [(1+D^2)/(1+D+D^2)], [(1+D^2)/(1+D+D^2)]$ and $([1+D^2]/[1+D+D^2])$, is shown in the same figure for comparison. For a rate 1/6 SCCC, we selected a rate 1/3 inner code that is shown in Fig. 18 with the high values of $h(\alpha_M) = 21$ and $h_m = 13$ (the $G_2(D)$ code with parity repetition has $h(\alpha_M) = 24$ but $h_m = 10$ and has shown poor performance at low SNRs). This rate 1/6 SCCC with two four-state codes outperforms the rate 1/6 Cassini code (Reed–Solomon (255,223) as outer code and rate 1/6, 16,384-state convolutional code as inner code) by 0.8 dB at BER $= 10^{-6}$. For a rate 1/3 SCCC, we selected a rate 2/3 inner code shown in Fig. 18 with $h(\alpha_M) = 9$ and $h_m = 6$, which are the highest possible among all rate 2/3, four-state codes. Simulation performance results for selected SCCCs are shown in Fig. 17 using block log-MAP iterative decoding. In the simulations, the input block of 16,384 (delay in bits), S-random interleaver, and 12 iterations were used. Better performance can be obtained with a higher number of iterations.

For SCCCs with code rates higher than 1/3, the rate of the outer code should be increased. For example, if we use a rate 3/4, four-state nonrecursive outer code with $d_f^o = 3$ and the recursive rate 2/3, four-state code in Table 1 (third row) as an inner code, then a rate 1/2 SCCC can be obtained. This code with an input block of 16,383 bits and 12 iterations was simulated. A bit-error rate of 10^{-6} was achieved at $E_b/N_0 = 1.05$ dB.
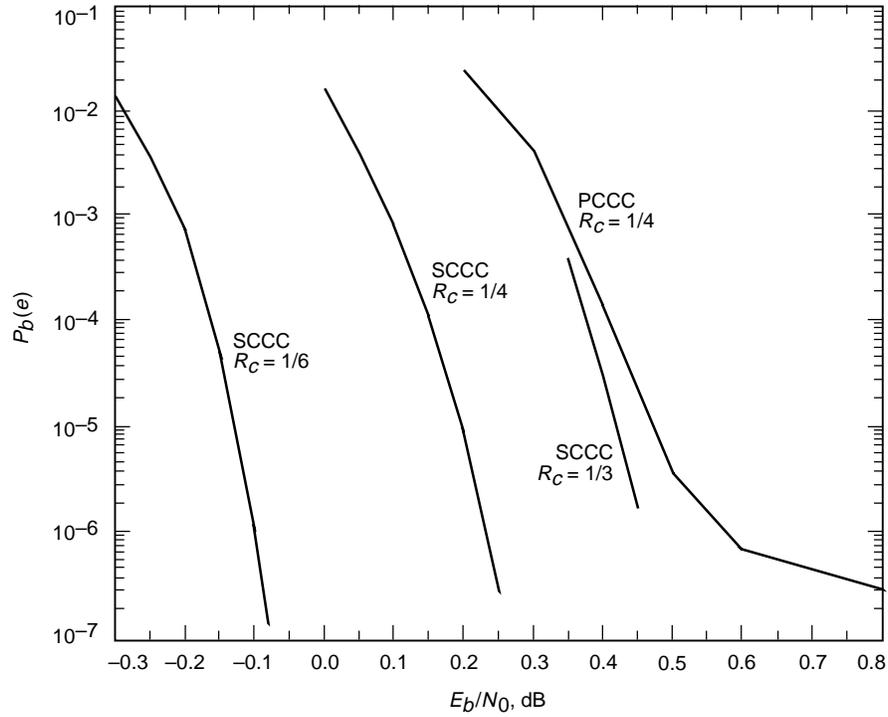
Fig. 17. The simulation performance of two four-state SCCCs. All concatenated codes are constructed from two four-state codes with the input information block = 16,384 bits and the number of iterations = 12.
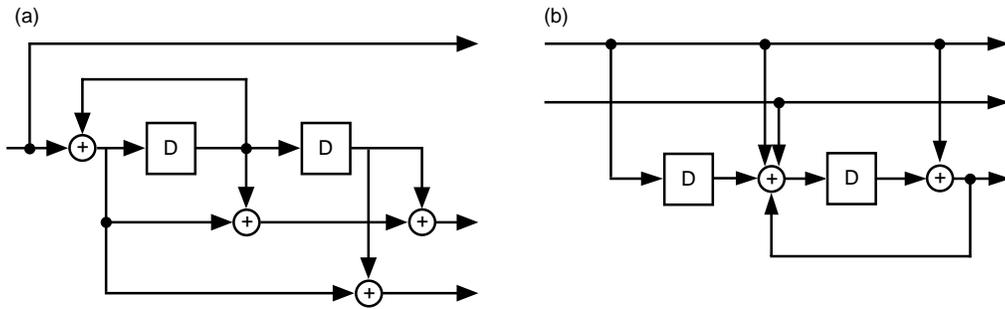
Fig. 18. Rate 1/3 and 2/3 four-state encoders with feedback (1 + *D*): (a) 1/3 code and (b) 2/3 code.

Acknowledgments

References

[1] G. D. Forney, Jr., *Concatenated Codes*, Cambridge, Massachusetts: Massachusetts Institute of Technology, 1966.

[2] R. H. Deng and D. J. Costello, "High Rate Concatenated Coding Systems Using Bandwidth Efficient Trellis Inner Codes," *IEEE Transactions on Communications*, vol. COM-37, no. 5, pp. 420–427, May 1989.

[3] J. Hagenauer and P. Hoeher, "Concatenated Viterbi Decoding," *Proceedings of the Fourth Joint Swedish–Soviet Int. Workshop on Information Theory*, Gotland, Sweden, pp. 29–33, August 1989.

[4] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes," *Proceedings of ICC'93*, Geneva, Switzerland, pp. 1064–1070, May 1993.

[5] S. Benedetto and G. Montorsi, "Average Performance of Parallel Concatenated Block Codes," *Electronics Letters*, vol. 31, no. 3, pp. 156–158, February 1995.

[6] S. Benedetto and G. Montorsi, "Unveiling Turbo-Codes: Some Results on Parallel Concatenated Coding Schemes," *IEEE Transactions on Information Theory*, vol. 43, no. 2, pp. 409–428, March 1996.

[7] A. J. Viterbi and J. K. Omura, *Principles of Digital Communication and Coding*, New York: McGraw-Hill, 1979.

[8] S. Benedetto and G. Montorsi, "Design of Parallel Concatenated Convolutional Codes," *IEEE Transactions on Communications*, vol. 44, no. 5, pp. 591–600, May 1996.

[9] D. Divsalar and R. J. McEliece, "The Effective Free Distance of Turbo Codes," *IEE Electronic Letters*, vol. 32, no. 5, pp. 445–446, February 29, 1996.

[10] P. Thitimajshima, *Systematic Recursive Convolutional Codes and Their Application to Parallel Concatenation*, Ph.D. Thesis, Université de Bretagne Occidentale, Brest, France, December 1993 (in French).

[11] S. Benedetto, E. Biglieri, and V. Castellani, *Digital Transmission Theory*, New York: Prentice-Hall, 1987.

[12] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Soft-Output Decoding Algorithms for Continuous Decoding of Parallel Concatenated Convolutional Codes," *Proceedings of ICC'96*, Dallas, Texas, June 23–27, 1996, June 1996.

[13] D. Divsalar, S. Dolinar, R. J. McEliece, and F. Pollara, "Transfer Function Bounds on the Performance of Turbo Codes," *The Telecommunications and Data Acquisition Progress Report 42-122, April–June 1995*, Jet Propulsion Laboratory, Pasadena, California, pp. 44–55, August 15, 1995.
http://tda.jpl.nasa.gov/tda/progress_report/42-122/122A.pdf

[14] Y. V. Svirid, "Weight Distributions and Bounds for Turbo Codes," *European Transactions on Telecommunications*, vol. 6, no. 5, September/October 1995.