Analysis, design and iterative decoding of double serially concatenated codes with interleavers

S. Benedetto, D. Divsalar, G. Montorsi, F. Pollara

S. Benedetto and G. Montorsi are with Dipartimento di Elettronica, Politecnico di Torino, C.so Duca degli Abruzzi n.24, 10129 Torino, Italy
D. Divsalar and F. Pollara are with Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California, USA

Abstract

A double serially concatenated code with two interleavers consists of the cascade of an *outer* encoder, an interleaver permuting the outer codeword bits, a *middle* encoder, another interleaver permuting the middle codeword bits and an *inner* encoder whose input words are the permuted middle codewords. The construction can be generalized to h cascaded encoders separated by $h-1$ interleavers, where $h > 3$. We obtain upper bounds to the *average* maximum-likelihood bit error probability of double serially concatenated block and convolutional coding schemes. Then, we derive design guidelines for the outer, middle, and inner codes that maximize the interleaver gain and the asymptotic slope of the error probability curves. Finally, we propose a low-complexity iterative decoding algorithm. Comparisons with parallel concatenated convolutional codes, known as "turbo codes", and with the recently proposed serially concatenated convolutional codes are also presented, showing that in some cases the new schemes offer better performance.

I. Acronyms

APP A-Posteriori Probability.

BCJR Bahl, Cocke, Jelinek, Raviv.

CC Constituent Code.

CWEF Conditional Weight Enumerating Function.

DPCCC Double Parallel Concatenated Convolutional Code.

DSCC Double Serially Concatenated Code.

DSCBC Double Serially Concatenated Block Code.

DSCCC Double Serially Concatenated Convolutional Code.

IOWEF Input Output Weight Enumerating Function.

LPDF Logarithm of the Probability Density Function.

ML Maximum Likelihood.

MPCCC Multiple Parallel Concatenated Convolutional Code.

SCC Serially Concatenated Code.

SCBC Serially Concatenated Block Code.

SCCC Serially Concatenated Convolutional Code.

PCC Parallel Concatenated Code.

PCCC Parallel Concatenated Convolutional Code.

II. 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, Dave 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 (deep-)space applications and many others. Alternative solutions for concatenation have also been studied, such as using a trellis-coded modulation scheme as 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.

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 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 using systematic CCs and not transmitting the information bits entering the second encoder. The idea of parallel concatenation of two codes was extended to multiple (> 2) codes, MPCCC, in [5] and [6]. The codes obtained in [6] have been shown to yield very high coding gains at low bit error probabilities; in particular, low bit error probabilities can be obtained at rates well beyond the channel cutoff rate, which had been regarded for long time as the "practical" capacity. As an example, a rate 1/4 MPCCC using three 8-state convolutional CCs, an interleaver with length 4096, and 20 iterations of the decoding algorithm yields a bit error probability of 10^{-5} at E_b/N_0 of 0.2 dB. 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 three CCs.

Recently, in [7], serially concatenated block and convolutional codes (SCBC and SCCC) with interleaver have been proposed. Their average maximum-likelihood performance, evaluated through an upper bound to the bit error probability, show an *interleaver gain*, i.e. the decrease of bit error probability with increasing interleaver length, significantly higher than for turbo codes. In [7], techniques for designing the CCs, and an iterative decoding algorithm were also illustrated.

In this paper, we extend the results of serial concatenation to the case of three interleaver codes, a scheme denoted by double serially concatenated code (DSCC), called double serially concatenated block code (DSCBC) or double serially concatenated convolutional code (DSCCC) according to the nature of 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 that maximize the *interleaver gain* and the asymptotic code performance, and present an iterative decoding algorithm that generalizes that presented in [7]. Comparisons with turbo codes and serially concatenated codes of the same complexity and decoding delay are also performed.

In Section III, we derive analytical upper bounds to the bit error probability of both DSCBCs and DSCCCs, using the concept of "uniform interleaver" that decouples the output of the outer encoder from the input of the middle encoder, and the output of the middle encoder from the input of the inner encoder. In Section IV, we propose design rules for DSCCCs through an asymptotic approximation of the bit error probability bound assuming long interleavers or large signal-to-noise ratios. In Section V we compare double and simple serial concatenations of block and convolutional codes in terms of maximum-likelihood analytical upper bounds. Section VI is devoted to the presentation of an iterative decoding algorithm, derived from the one introduced in [7] and to its application to some significant codes.

III. Analytical bounds to the performance of double serially concatenated codes

For simplicity of the presentation, we begin considering double serially concatenated block codes (DSCBCs).

A. Double serially concatenated block codes

The scheme of a double serially concatenated block code is shown in Fig. 1. It is composed of three cascaded CCs, the *outer* (N_1, k) code C_o with rate $R_c^o = k/N_1$, the *middle* (N_2, N_1) code C_m with rate $R_c^m = N_1/N_2$ and the *inner* (n, N_2) code C_i with rate $R_c^i = N_2/n$, linked by two interleavers of lengths N_1 and N_2. The overall DSCBC is then an (n, k) code, and we will refer to it as the (n, k, N_1, N_2) code C_S, including also the interleaver lengths. 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 also the DSCBC 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 [8], [9], [7], a crucial step in the analysis consists in replacing the actual interleaver that performs a permutation of the N input bits with an abstract interleaver called *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 input and output weight is preserved. Use of the uniform interleaver permits the computation

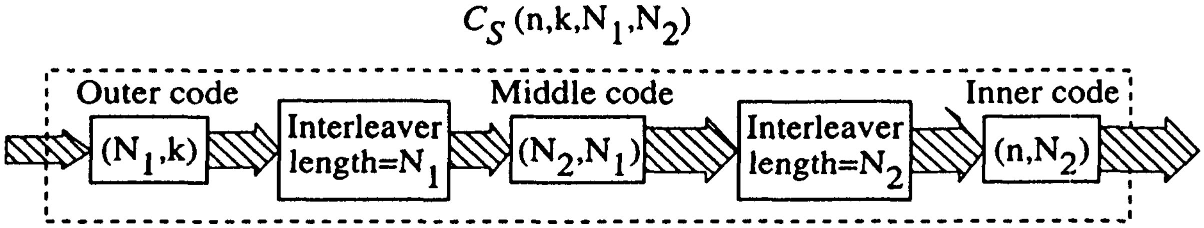

Fig. 1. Double serially concatenated (n, k, N_1, N_2) block code.

of the "average" performance of DSCBCs, intended as the expectation of the performance of DSCBCs using the same CCs, taken over the ensemble of all interleavers of given lengths. A theorem proved in [9] 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 a set of two particular interleavers yielding performance better than or equal to those of the two uniform interleavers.

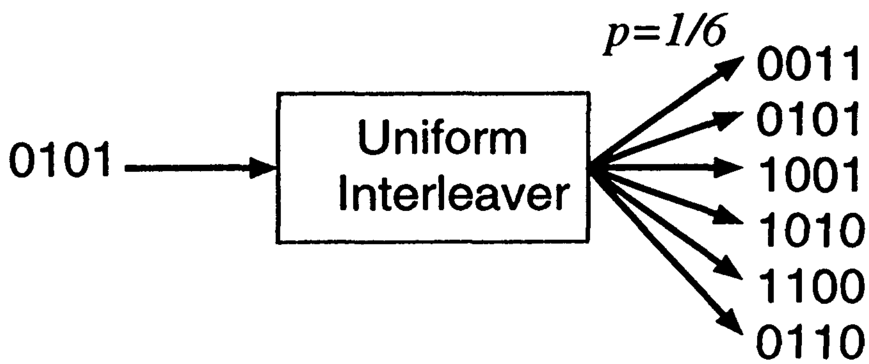

Fig. 2. The action of a uniform interleaver of length 4 on sequences of weight 2

Let us define the *Input-Output Weight Enumerating Function* (IOWEF) of the DSCBC 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 DSCBC with weight h associated to an input word of weight w.

We also define the *Conditional Weight Enumerating Function* (CWEF) $A^{C_S}(w,H)$ of the DSCBC as the weight distribution of codewords of the DSCBC which have input word weight w. It is related to the IOWEF by

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

With the knowledge of the CWEF, an upper bound to the bit error probability of the maximum-likelihood decoded DSCBC can be obtained in the form [9]

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

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

The problem thus consists in the evaluation of the CWEF of the DSCBC from the knowledge of the CWEFs of the outer, the middle, and the inner codes, which we call $A^{C_o}(w,L_1)$, $A^{C_m}(l_1,L_2)$ and $A^{C_i}(l_2,H)$. To do this, we exploit the properties of the uniform interleavers. The first interleaver transforms a codeword of weight l_1 at the output of the outer encoder into all its distinct $\binom{N_1}{l_1}$ permutations. Similarly, the second interleaver transforms a codeword of weight l_2 at the output of the middle encoder into all its distinct $\binom{N_2}{l_2}$ permutations. As a consequence, each codeword of the outer code C_o of weight l_1, through the action of the first uniform interleaver, enters the middle encoder generating $\binom{N_1}{l_1}$ codewords of the middle code C_m, and each codeword of the middle code C_m of weight l_2, through the action of the second uniform interleaver, enters the inner encoder generating $\binom{N_2}{l_2}$ codewords of the inner code C_i. Thus, the number $A^{C_S}_{w,h}$ of codewords of the DSCBC of weight h associated with an input word of weight w is given by

$$A^{C_S}_{w,h} = \sum_{l_1=0}^{N_1} \sum_{l_2=0}^{N_2} \frac{A^{C_o}_{w,l_1} \times A^{C_m}_{l_1,l_2} \times A^{C_i}_{l_2,h}}{\binom{N_1}{l_1}\binom{N_2}{l_2}} \ . \tag{4}$$

From (4) we derive the expressions of the IOWEF and CWEF of the DSCBC

$$A^{C_S}(w,H) = \sum_{l_1=0}^{N_1} \sum_{l_2=0}^{N_2} \frac{A^{C_o}_{w,l_1} \times A^{C_m}_{l_1,l_2} \times A^{C_i}(l_2,H)}{\binom{N_1}{l_1}\binom{N_2}{l_2}} \ , \tag{5}$$

$$A^{C_S}(W,H) = \sum_{l_1=0}^{N_1} \sum_{l_2=0}^{N_2} \frac{A^{C_o}(W,l_1) \times A^{C_m}_{l_1,l_2} \times A^{C_i}(l_2,H)}{\binom{N_1}{l_1}\binom{N_2}{l_2}} \ , \tag{6}$$

where $A^{C_o}(W,l_1)$ is the conditional weight distribution of the input words that generate codewords of the outer code of weight l_1.

Example 1

Consider the $(7,2)$ DSCBC code obtained by concatenating a $(3,2)$ parity check code to another $(4,3)$ parity check code, and, finally to a $(7,4)$ systematic Hamming code through two interleavers of lengths $N_1 = 3$ and $N_2 = 4$. The IOWEF $A^{C_o}(W,L_1)$, $A^{C_m}(L_1,L_2)$, and $A^{C_i}(L_2,H)$ of the outer, the middle and inner codes are

$$A^{C_o}(W,L_1) = 1 + W(2L_1^2) + W^2(L_1^2)$$

$$A^{C_m}(L_1,L_2) = 1 + L_1(3L_2^2) + L_1^2(3L_2^2) + L_1^3(L_2^4)$$

$$A^{C_i}(L_2,H) = 1 + L_2(3H^3 + H^4) + L_2^2(3H^3 + 3H^4) + L_2^3(H^3 + 3H^4) + L_2^4 H^7 \ ,$$

so that

$$\begin{aligned} A^{C_o}(W,0) &= 1 \\ A^{C_o}(W,1) &= 0 \\ A^{C_o}(W,2) &= 2W + W^2 \\ A^{C_o}(W,3) &= 0 \end{aligned}$$

$$\begin{aligned} A^{C_m}(L_1,0) &= 1 \\ A^{C_m}(L_1,1) &= 0 \\ A^{C_m}(L_1,2) &= 3L_1 + 3L_1^2 \\ A^{C_m}(L_1,3) &= 0 \\ A^{C_m}(L_1,4) &= L_1^3 \end{aligned}$$

This implies $A^{\overline{C_m}}_{0,\overline{0}} = 1$, $A^{\overline{C_m}}_{1,\overline{2}} = 3$, $A^{\overline{C_m}}_{2,\overline{2}} = 3$, $A^{\overline{C_m}}_{3,\overline{4}} = 1$, and $A^{\overline{C_m}}_{l_1,\overline{l_2}} = 0$ for all other l_1, l_2, so that

$$\begin{aligned} A^{C_i}(0,H) &= 1 \\ A^{C_i}(1,H) &= 3H^3 + H^4 \\ A^{C_i}(2,H) &= 3H^3 + 3H^4 \\ A^{C_i}(3,H) &= H^3 + 3H^4 \\ A^{C_i}(4,H) &= H^7 \ . \end{aligned}$$

Through (6), we then obtain

$$\begin{aligned} A^{C_s}(W,H) &= \sum_{l_1=0}^{3}\sum_{l_2=0}^{4} \frac{A^{C_o}(W,l_1) \times A^{\overline{C_m}}_{l_1,\overline{l_2}} \times A^{C_i}(l_2,H)}{\binom{N_1}{l_1}\binom{N_2}{l_2}} \\ &= 1 + W(H^3 + H^4) + W^2(0.5H^3 + 0.5H^4)\,. \end{aligned}$$

◇

Previous results (5) and (6) can be easily generalized to the case of two interleavers, the first with length N_1, which is an integer q multiple of the length of the outer codewords, and the second with length N_2, which is the same integer q multiple of the length of the middle codewords. Denoting by $A^{C_o^q}(W,L_1)$ the IOWEF of the new (N_1,qk) outer code, by $A^{C_m^q}(L_2,L_1)$ the IOWEF of the new (N_2,N_1) middle code and finally by $A^{C_i^q}(L_2,H)$ the IOWEF of the new (qn,N_2) inner code, it is straightforward to obtain

$$A^{C_o^q}(W,L_1) = [A^{C_o}(W,L_1)]^q$$

$$\begin{aligned} A^{C_m^q}(L_1, L_2) &= [A^{C_m}(L_1, L_2)]^q \\ A^{C_i^q}(L_2, H) &= [A^{C_i}(L_2, H)]^q. \end{aligned} \tag{7}$$

From the IOWEFs (7), through (2) we obtain the CWEFs $A^{C_o^q}(W, l_1)$, $A^{C_m^q}_{l_1,l_2}$, and $A^{C_i^q}(l_2, H)$ of the new CCs, and, finally, the IOWEF and CWEF of the new (qn, qk, N_1, N_2) DSCBC C_S^q

$$A^{C_S^q}(w, H) = \sum_{l_1=0}^{N_1} \sum_{l_2=0}^{N_2} \frac{A^{C_o^q}_{w,l_1} \times A^{C_m^q}_{l_1,l_2} \times A^{C_i^q}(l_2, H)}{\binom{N_1}{l_1}\binom{N_2}{l_2}}, \tag{8}$$

$$A^{C_S^q}(W, H) = \sum_{l_1=0}^{N_1} \sum_{l_2=0}^{N_2} \frac{A^{C_o^q}(W, l_1) \times A^{C_m^q}_{l_1,l_2} \times A^{C_i^q}(l_2, H)}{\binom{N_1}{l_1}\binom{N_2}{l_2}}. \tag{9}$$

Example 2

We consider a DSCBC composed by a (4,3) parity-check code as outer code, a (7,4) Hamming code as middle code, and a (15,7) BCH code as inner code. These CCs are linked by interleavers of length $N_1 = 4q$ and $N_2 = 7q$. Using equations (8) and (3), upper bounds to the bit error probability are obtained and plotted in Fig. 3 for various values of the integer q. The curves show the *interleaver gain*, defined as the factor by which the bit error probability is decreased with the interleavers length. Contrary to the case of parallel concatenated block codes [9], the curves do not exhibit the interleaver gain saturation, i.e. a phenomenon in which the interleaver gain progressively decreases while increasing the interleaver length, up to a point in which increasing the interleaver length does not yield any further gain (examples of gain saturation have been reported in [9]. Rather, for sufficiently high signal-to-noise ratios, the bit error probability seems to decrease regularly with q as q^{-3}. We will explain this behavior in Section V.

◇

B. Double Serially concatenated convolutional codes

The structure of a double serially concatenated convolutional code (DSCCC) is shown in Fig. 4. It refers to the case of three convolutional CCs, the outer code C_o with rate $R_c^o = k/p_1$, the middle code C_m with rate $R_c^m = p_1/p_2$, and the inner code code C_i with rate $R_c^i = p_2/n$, joined by two interleavers of length N_1, N_2 bits, generating a DSCCC C_S with rate $R_c = k/n$. Note that N_1 must be an integer multiple of p_1, and N_2 must be an integer multiple of p_2 [1]. In addition, the middle code rate imposes the constraint $N_1/p_1 = N_2/p_2 = N$, so that the input block length is kN. We assume, as before, that the convolutional CCs are linear, so that the DSCCC is linear as well, and the uniform error property applies.

[1] Actually, this constraint is not necessary. We can choose, in fact, inner, middle and outer codes of any rates $R_c^i = k_i/n_i$, $R_c^m = k_m/n_m$ and $R_c^o = k_o/n_o$, constraining the interleaver lengths to be an integer multiple of the minimum common multiple of n_o and k_m, and of the minimum common multiple of n_m and k_i i.e. $N_1 = K_1 \cdot \text{mcm}(n_o, k_m)$, and $N_2 = K_2 \cdot \text{mcm}(n_m, k_i)$ such that $N_1/N_2 = R_c^m$. This generalization, though, leads to more complicated expressions and is not considered in the following.

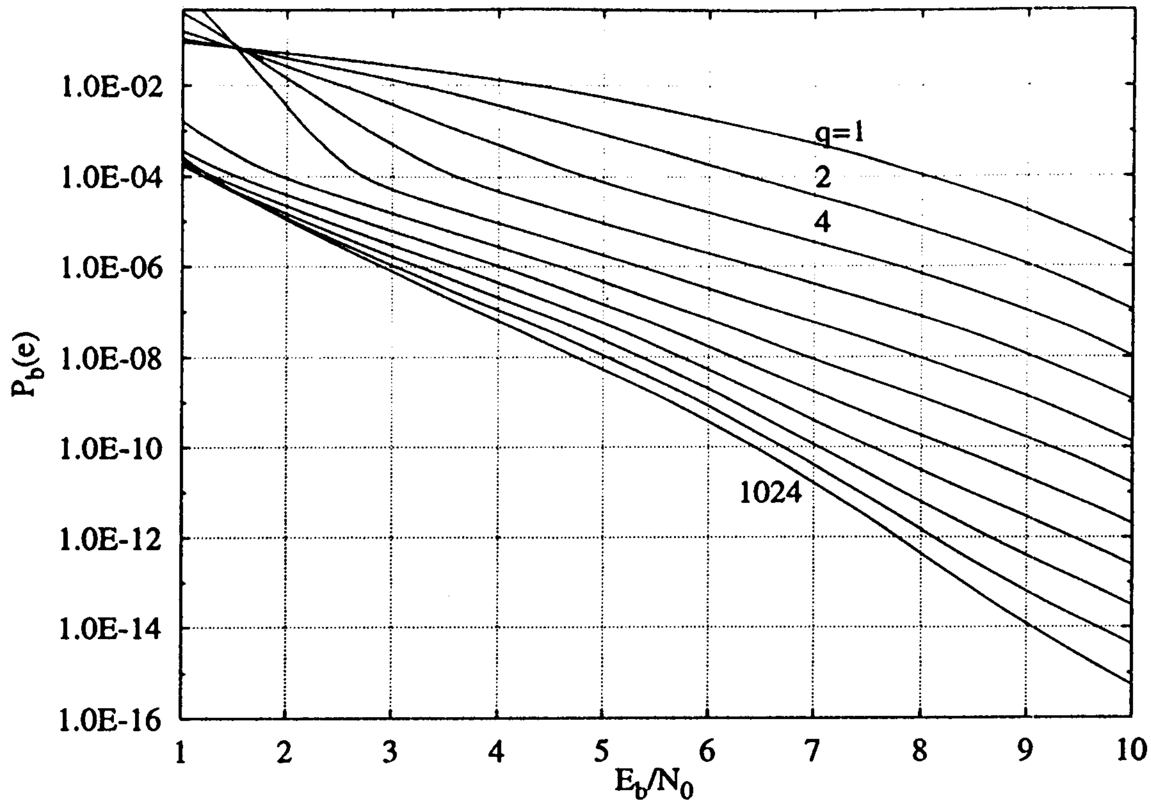

Fig. 3. Analytical bound to the ML bit error probability for the DSCBC of Example 2. The values of q are consecutive powers of 2, i.e. $q = 2^l$, $l = 0, \ldots, 10$

The exact analysis of this scheme can be performed by appropriate modifications of that described in [9] for PCCCs. It requires the use of a *hyper-trellis* having as hyper-states set of states of outer, middle and inner codes. The hyper-states S_{ijk} and S_{lmn} are joined by a *hyper-branch* that consists of all pairs of paths with length N that join states s_i, s_l of the inner code , states s_j, s_m of the middle code, states s_k, s_n of the outer code, respectively. Each hyper-branch is thus an equivalent DSCBC labeled with an IOWEF that can be evaluated as explained in previous subsection. From the hyper-trellis, the upper bound to the bit error probability can be obtained through the standard transfer function technique employed for convolutional codes [10]. As proved in [9] for the case of two parallel concatenated convolutional codes, 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 hyper-trellis joining the hyper-states S_{000}, S_{000}. In the following, we will always use this approximation.

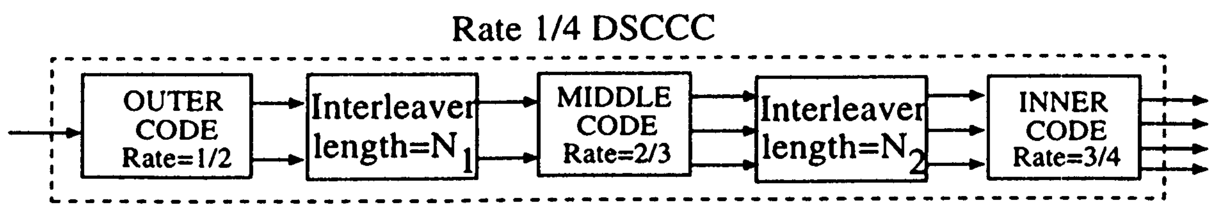

Fig. 4. Double serially concatenated (n, k, N_1, N_2) convolutional code.

IV. Design of double serially concatenated codes

In previous section, we have presented an analytical bounding technique to find the ML performance of DSCBC and DSCCC. For practical applications, DSCCCs are to be preferred to DSCBCs. One reason is that trellis-based maximum a-posteriori algorithms like the BCJR algorithm [11], [12] are less complex for convolutional than for block codes, since the trellis is time-invariant for convolutional codes and time-varying for block codes, the second is that the interleaver gain can be greater for convolutional CCs, provided they are suitably designed [9]. Hence, we deal mainly with the design of DSCCCs, extending our conclusions to DSCBCs when appropriate.

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

$$P_b(e) \leq \sum_{w=w_m^o}^{N_1 R_c^o} \frac{w}{N_1 R_c^o} A^{C_S}(w,H)|_{H=e^{-R_c E_b/N_0}} = \sum_{h=h_m}^{N_2/R_c^i} \sum_{w=w_m^o}^{N_1 R_c^o} \frac{w}{N_1 R_c^o} A_{w,h}^{C_S} e^{-hR_c E_b/N_0} \,, \tag{10}$$

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 merging 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 (4), once the quantities $A_{w,l_1}^{C_o}$, $A_{l_1,l_2}^{C_m}$ and $A_{l_2,h}^{C_i}$ 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

[2] In the following, a subscript "m" will denote "minimum", and a subscript "M" will denote "maximum". Note that superscript m is also used to denote the middle code.

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

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

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

be the weight enumerating function of sequences of the convolutional code that concatenate j error events with total input weight l (see Fig. 5), 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 $A^C_{l,h}$ of the equivalent block code can be approximated by[4]

$$A^C_{l,h} \sim \sum_{j=1}^{n_M} \binom{N/p}{j} A_{l,h,j} \tag{12}$$

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.

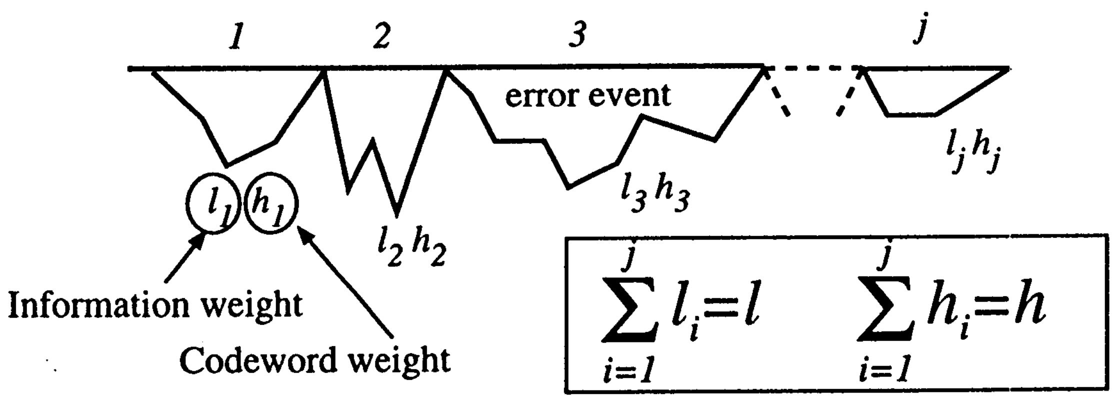

Fig. 5. A code sequence with parameters l, h, j.

Let us return now to the block code equivalent to the DSCCC. Using previous result (12) with $j = n^i$ for the inner code, $j = n^m$ for the middle code, and the analogous one $j = n^o$ for the

[4]This assumption permits neglecting the length of error events compared to N, and assuming 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.

outer code[5], and noting that $N_2/p_2 = N_1/p_1 = N$, we obtain for the outer code

$$A^{C_o}_{w,l} \sim \sum_{n^o=1}^{n^o_M} \binom{N}{n^o} A^o_{w,l,n^o} \tag{13}$$

For middle and inner codes, similar expressions can be obtained. Then, substituting them into (4), we obtain the coefficient $A^{C_S}_{w,h}$ of the double serially concatenated block code equivalent to the DSCCC in the form

$$A^{C_S}_{w,h} \sim \sum_{l_1=d^o_f}^{N_1} \sum_{l_2=l_{2,m}}^{N_2} \sum_{n^o=1}^{n^o_M} \sum_{n^m=1}^{n^m_M} \sum_{n^i=1}^{n^i_M} \frac{\binom{N}{n^o}\binom{N}{n^m}\binom{N}{n^i}}{\binom{N_1}{l_1}\binom{N_2}{l_2}} A^o_{w,l_1,n^o} A^m_{l_1,l_2,n^m} A^i_{l_2,h,n^i} \ , \tag{14}$$

where d^o_f is the free distance of the outer code, and $l_{2,m}$ is the minimum weight of the sequences of the middle code due to sequences of the outer code with weight d^o_f, $l_{2,m} \geq d^m_f$. 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 into (14) yields

$$A^{C_S}_{w,h} \sim \sum_{l_1=d^o_f}^{N_1} \sum_{l_2=l_{2,m}}^{N_2} \sum_{n^o=1}^{n^o_M} \sum_{n^m=1}^{n^m_M} \sum_{n^i=1}^{n^i_M} N^{n^o+n^m+n^i-l_1-l_2} \frac{l_1! l_2!}{n^o! n^m! n^i!} A^o_{w,l_1,n^o} A^m_{l_1,l_2,n^m} A^i_{l_2,h,n^i} \ . \tag{15}$$

Finally, substituting (15) into (10), yields the bit error probability bound in the form

$$P_b(e) \stackrel{\sim}{\leq} \sum_{h=h_m}^{N_2/R^i_c} A(h,N) e^{-hR_cE_b/N_0} \tag{16}$$

where the coefficients $A(h,N)$ have been defined as

$$\begin{aligned} A(h,N) &\triangleq \sum_{w=w^o_m}^{N_1R^o_c} \frac{w}{N_1R^o_C} A^{C_S}_{w,h} \\ &= \sum_{w=w^o_m}^{N_1R^o_c} \sum_{l_1=d^o_f}^{N_1} \sum_{l_2=l_{2,m}}^{N_2} \sum_{n^o=1}^{n^o_M} \sum_{n^m=1}^{n^m_M} \sum_{n^i=1}^{n^i_M} N^{n^o+n^m+n^i-l_1-l_2-1} \frac{l_1! l_2!}{n^o! n^m! n^i!} \frac{w}{k} A^o_{w,l_1,n^o} A^m_{l_1,l_2,n^m} A^i_{l_2,h,n^i} \ . \end{aligned} \tag{17}$$

having used the result (15) for the $A^{C_S}_{w,h}$.

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

Using expressions (16) and (17) as the starting point, we will obtain some important design considerations.

The bound (16) to the bit error probability is obtained by adding terms of the summation with respect to the DSCCC weights h. From (17), the coefficients $A(h, N)$ of the exponentials in h depend, among other parameter, 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

$$\alpha(h) \stackrel{\triangle}{=} \max_{w,l_1,l_2} \{n^o + n^m + n^i - l_1(w) - l_2(w) - 1\} . \tag{18}$$

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 DSCC are dominated by the first term of the summation in h, corresponding to the minimum value $h = h_m$. Remembering that, by definition, n^i_M, n^m_M and n^o_M are the maximum number of concatenated error events in codewords of the inner, middle, and outer code of weights h_m, l_2 and l_1, respectively, the following inequalities hold true:

$$n^i_M \le \left\lfloor \frac{h_m}{d^i_f} \right\rfloor , \tag{19}$$

$$n^m_M \le \left\lfloor \frac{l_2(h_m)}{l_{2,m}} \right\rfloor \le \left\lfloor \frac{l_2(h_m)}{d^m_f} \right\rfloor , \tag{20}$$

$$n^o_M \le \left\lfloor \frac{l_1(l_2(h_m))}{d^o_f} \right\rfloor , \tag{21}$$

and

$$\begin{aligned} \alpha(h_m) &\le \max_{l_1,l_2} \left\{ \left\lfloor \frac{h_m}{d^i_f} \right\rfloor + \left\lfloor \frac{l_2(h_m)}{d^m_f} \right\rfloor + \left\lfloor \frac{l_1(l_2(h_m))}{d^o_f} \right\rfloor - l_1(l_2(h_m)) - l_2(h_m) - 1 \right\} \\ &= \left\lfloor \frac{h_m}{d^i_f} \right\rfloor + \left\lfloor \frac{l_{2,m}(h_m)}{d^m_f} \right\rfloor + \left\lfloor \frac{l_{1,m}(l_{2,m}(h_m))}{d^o_f} \right\rfloor - l_{1,m}(l_{2,m}(h_m)) - l_{2,m}(h_m) - 1 , \end{aligned} \tag{22}$$

where $l_{2,m}(h_m)$ is the minimum weight l_2 of codewords of the middle code yielding a codeword of weight h_m of the inner code, $l_{1,m}(l_{2,m}(h_m))$ is the minimum weight l_1 of codewords of the outer code yielding a codeword of weight $l_{2,m}(h_m)$ of the inner code and $\lfloor x \rfloor$ means "integer part of x".

In most cases[6], $l_{1,m}(l_{2,m}(h_m)) < 2d^o_f$, $l_{2,m}(h_m) < 2d^m_f$, and $h_m < 2d^i_f$, so that $n^i_M = n^m_M = n^o_M =$

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

1, so that (22) becomes

$$\alpha(h_m) = 2 - l_{1,m}(l_{2,m}(h_m)) - l_{2,m}(h_m) \leq 2 - d_f^o - d_f^m \; . \tag{23}$$

The result (23) shows that the exponent of N corresponding to the minimum-weight of DSCCC codewords is always less than -2 for $d_f^o \geq 2$ and $d_f^m \geq 2$, thus yielding an interleaver gain at high E_b/N_0. Substitution of the exponent $\alpha(h_m)$ into (16) truncated to the first term of the summation in h yields

$$\lim_{\frac{E_b}{N_0} \to \infty} P_b(e) \overset{\sim}{\leq} B_m N^{2-d_f^o-d_f^m} \exp(-h_m R_c E_b/N_0) \tag{24}$$

where the constant B_m is

$$B_m = \frac{A^i_{l_{2,m}(h_m),h_m,1} A^m_{l_{1,m}(l_{2,m}(h_m)),l_{2,m}(h_m),1} [l_{1,m}(l_{2,m}(h_m))]![l_{2,m}(h_m)]!}{k} \sum_{w \in \mathcal{W}_m} w A^o_{w,l_{1,m}(l_{2,m}(h_m)),1} \; , \tag{25}$$

and $\mathcal{W}_m$ is the set of input weights w that generate codewords of the outer code with weight $l_{1,m}(l_{2,m}(h_m))$.

Expression (24) suggests the following conclusions:

- For the values of E_b/N_0 and N where the DSCCC performance is dominated by its free distance $d_f^{C_S} = h_m$, increasing the interleaver length yields a gain in performance.
- To increase the interleaver gain, one should choose an outer code, and a middle code with large d_f^o, and d_f^m, respectively.
- To improve the performance with E_b/N_0, one should choose an inner, middle and outer code combination such that h_m is large.

These conclusions do not depend on the structure of the CCs, and thus they yield for both recursive and non recursive encoder.

The curves of Fig. 3 showing the performance of the various DSCBCs of Example 2 with increasing interleaver length, however, 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

$$\alpha_M = \max_{w,l_1,l_2} \{ n_M^o - l_1(w) + n_M^m - l_2(l_1(w)) + n_M^i - 1 \} \; . \tag{26}$$

B. The maximum exponent of N

For any $w, l_1(w), l_2(l_1(w))$, the following inequality holds

$$n_M^i \leq \left\lfloor \frac{l_2(l_1(w))}{w_m^i} \right\rfloor \tag{27}$$

so that

$$n_M^o - l_1(w) + n_M^m - l_2(l_1(w)) + n_M^i - 1 \leq n_M^o - l_1(w) + n_M^m - l_2(l_1(w)) + \left\lfloor \frac{l_2(l_1(w))}{w_m^i} \right\rfloor - 1 \tag{28}$$

and the following upper bound to α_M in (26) holds:

$$\alpha_M \leq \max_{w,l_1,l_2} \left\{ n_M^o - l_1(w) + n_M^m - l_2(l_1(w)) + \left\lfloor \frac{l_2(l_1(w))}{w_m^i} \right\rfloor - 1 \right\} . \tag{29}$$

Since now

$$l_2(l_1(w)) \geq n_M^m d_f^m$$

we can write the inequality

$$-l_2(l_1(w)) + \left\lfloor \frac{l_2(l_1(w))}{w_m^i} \right\rfloor \leq -n_M^m d_f^m + \left\lfloor \frac{n_M^m d_f^m}{w_m^i} \right\rfloor$$

and finally

$$\alpha_M \leq \max_{w,l_1} \left\{ n_M^o - l_1(w) + n_M^m (1 - d_f^m) + \left\lfloor \frac{n_M^m d_f^m}{w_m^i} \right\rfloor - 1 \right\} \tag{30}$$

Starting from (30), we will evaluate the upper bound to α_M for all possible configurations.

B.1 Block encoders, and nonrecursive convolutional inner and middle encoders

For non recursive inner and middle encoders, we have $n_M^m = l_1$ and $w_m^i = 1$. In fact, every input sequence with weight one 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 one. Since the uniform interleavers generate all possible permutation of their input sequences, this event will certainly occur. Substituting these values into (30) we obtain

$$\alpha_M = n_M^o - 1 \geq 0 , \tag{31}$$

and interleaving gain is not allowed. This conclusion holds true for both DSCCC employing nonrecursive inner, and middle encoders and for all DSCBCs, since block codes have codewords corresponding to input words with weight equal to one.

For those DSCCs we *always* have, for some h, coefficients of the exponential in h of (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.

B.2 Nonrecursive inner, recursive middle encoders

Since the inner encoder is nonrecursive, we have $w_m^i = 1$, so that (30) becomes

$$\alpha_M \leq \max_{w,l_1}\{n_M^o - l_1(w) + n_M^m - 1\}$$

For any $w, l_1(w)$, owing to the recursiveness of the middle code, the following inequality holds:

$$n_M^m \leq \left\lfloor \frac{l_1(w)}{2} \right\rfloor$$

so that

$$\alpha_M \leq \max_{w,l_1}\left\{n_M^o - \left\lfloor \frac{l_1(w)+1}{2} \right\rfloor - 1\right\}$$

Finally, since for any $w, l_1(w)$ we can write $l_1(w) \geq n_M^o d_f^o$, we obtain

$$\alpha_M \leq \max_{n_M^o}\left\{n_M^o - \left\lfloor \frac{n_M^o d_f^o + 1}{2} \right\rfloor - 1\right\} = -\left\lfloor \frac{d_f^o + 1}{2} \right\rfloor . \tag{32}$$

For d_f^o even, the weight $h(\alpha_M)$ associated to the highest exponent of N, is given by

$$h(\alpha_M) = \frac{d_f^o d_{\text{f,eff}}^m d_1^i}{2} ,$$

where d_1^i is weight of sequences of the inner code generated by input sequences of weight. In fact, $h(\alpha_M)$ is the weight of an inner code sequence formed by l_2 error events, each generated by a weight 1 input sequence. On the other hand, l_2 is the weight of a middle code sequence that concatenates $d_f^o/2$ error events with weight $d_{\text{f,eff}}^m$.

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

$$h(\alpha_M) = \frac{(d_f^o - 3) d_{\text{f,eff}}^m d_1^i}{2} + d_3^m d_1^i , \tag{33}$$

where d_3^m is the minimum weight of the sequences of the middle code generated by a weight 3 input sequence. In this case, in fact, we have

$$n_M^m = \frac{d_f^o - 1}{2}$$

concatenated error events, of which $n_M^m - 1$ generated by weight 2 input sequences and one generated by a weight 3 input sequence. Note that the interleaving gain in this case is similar to the one obtainable with SCCCs in [7]; in the case of DSCCCs, however, $h(\alpha_M)$ can be made larger.

B.3 Recursive inner encoder

When the inner encoder is recursive, we obtain a value for α_M that holds for both recursive and nonrecursive middle encoders.

We start replacing $w_m^i = 2$ into (30), since the inner encoder is recursive, obtaining

$$\alpha_M \leq \max_{w,l_1} \left\{ n_M^o - l_1(w) + n_M^o(1-d_f^m) + \left\lfloor \frac{n_M^m d_f^m}{2} \right\rfloor - 1 \right\}$$

For any w, l_1, the following inequality holds

$$n_M^m(1-d_f^m) + \left\lfloor \frac{n_M^m d_f^m}{2} \right\rfloor \leq 1 - d_f^m + \left\lfloor \frac{d_f^m}{2} \right\rfloor = 1 - \left\lfloor \frac{d_f^m+1}{2} \right\rfloor$$

so that

$$\alpha_M \leq \max_{w,l_1} \left\{ n_M^o - l_1(w) - \left\lfloor \frac{d_f^m+1}{2} \right\rfloor \right\} . \tag{34}$$

Moreover, taking into account that, for any $w, l_1(w)$, we can write

$$l_1(w) \geq n_M^o d_f^o$$

equation (34) becomes

$$\alpha_M \leq \max_{n_M^o} \left\{ n_M^o - n_M^o d_f^o - \left\lfloor \frac{d_f^m+1}{2} \right\rfloor \right\} = 1 - d_f^o - \left\lfloor \frac{d_f^m+1}{2} \right\rfloor . \tag{35}$$

It is interesting to note that when $d_f^o = 1$ (35) simplifies to the same result obtained for the serial concatenation of two codes (SCCC) in [7].

The weight $h(\alpha_M)$ associated to the highest exponent of N, satisfies the following inequality

$$h(\alpha_M) \geq \frac{d_f^m d_{\mathrm{f,eff}}^i}{2},$$

for d_f^m even, and

$$h(\alpha_M) \geq \frac{(d_f^m - 3) d_{\mathrm{f,eff}}^i}{2} + d_3^m ,$$

for d_f^m odd.

The following design considerations can be drawn from (31),(32) and (35):

- The choice of a non recursive encoder for both middle and inner CCs should be avoided, as it leads (see (31)) to at least one coefficient of the exponential in h that increases with N, thus preventing from the possibility of obtaining an interleaver gain for large N.
- Since at least one between middle and inner encoders must be recursive, we can have three different cases, which all guarantee a certain interleaver gain. The worst is the one in which

the middle encoder is recursive and the inner is non recursive. In this case, in fact, the value of α_M given by (32) is the highest. The best choice is to have recursive the inner encoder, no matter how the middle is. In this case, in fact, the value of α_M is given by (35), which yield the lowest exponent of N, and thus the largest interleaver gain.

V. Comparison between simple and double serially concatenated codes

To confirm the design rules obtained asymptotically, i.e. for large signal-to-noise ratio and large interleaver lengths N, we analyze block and convolutional DSCCs, with different interleaver lengths, and compare their performance with those predicted by the design guidelines. Moreover, we compare the analytical upper bounds to the bit error probability for block and convolutional SCCs and DSCCs having the same code rate.

A. Serially concatenated block codes

Consider first the DSCBC of Example 2. The predicted value of $\alpha(h_m)$ is given by (23). In our case, the minimum distance of the outer code is 2 and that of the middle code 3. As a consequence, $\alpha(h_m) \leq -3$. Looking at the upper bounds to the bit error probability shown in Fig. 3, it is easily verified that the interleaver gain, for a fixed and sufficiently large signal-to-noise ratio, goes as N^{-3}, as predicted.

To compare simple and double serial block code concatenations, we have constructed two rate 3/15 codes. The first is the DSCBC of Example 2, and the second is an SCBC obtained by concatenating the (7,3) code whose 8 codewords are the even-weight codewords of the (7,4) Hamming code with the (15,7) BCH code. The interleavers for the two concatenated codes have been chosen so as to yield the same latency, expressed with the parameter q in terms of number of input words. The curves of the bit error probability bounds reported in Fig. 6 show the superior performance of the DSCBC for low-medium signal-to-noise ratios. In fact, for $q = 1$ the performance are the same, whereas for larger values of q the DSCBC behaves better, owing to the larger interleaving gain (at $P_b(e) = 10^{-6}$ the gain is 2 dB for $q = 1000$). For sufficiently large E_b/N_0, the curves corresponding to the same value of q merge, owing to the fact that the two codes have the same minimum distance.

B. Serially concatenated convolutional codes

We consider several rate 1/4 DSCCCs formed by an outer 4-state convolutional code with rate 1/2, a middle 4-state convolutional code with rate 2/3 and an inner 4-state convolutional code with rate 3/4, joined by two uniform interleavers of length $N_1 = 2N$, and $N_2 = 3N$.

The main parameters of the employed CCs are described in Table I. In building the DSCCCs, we keep as outer encoder a non recursive encoder, whereas for the middle and inner encoders we use three different combinations. For the first code, DSCCC1, the middle and the inner encoders

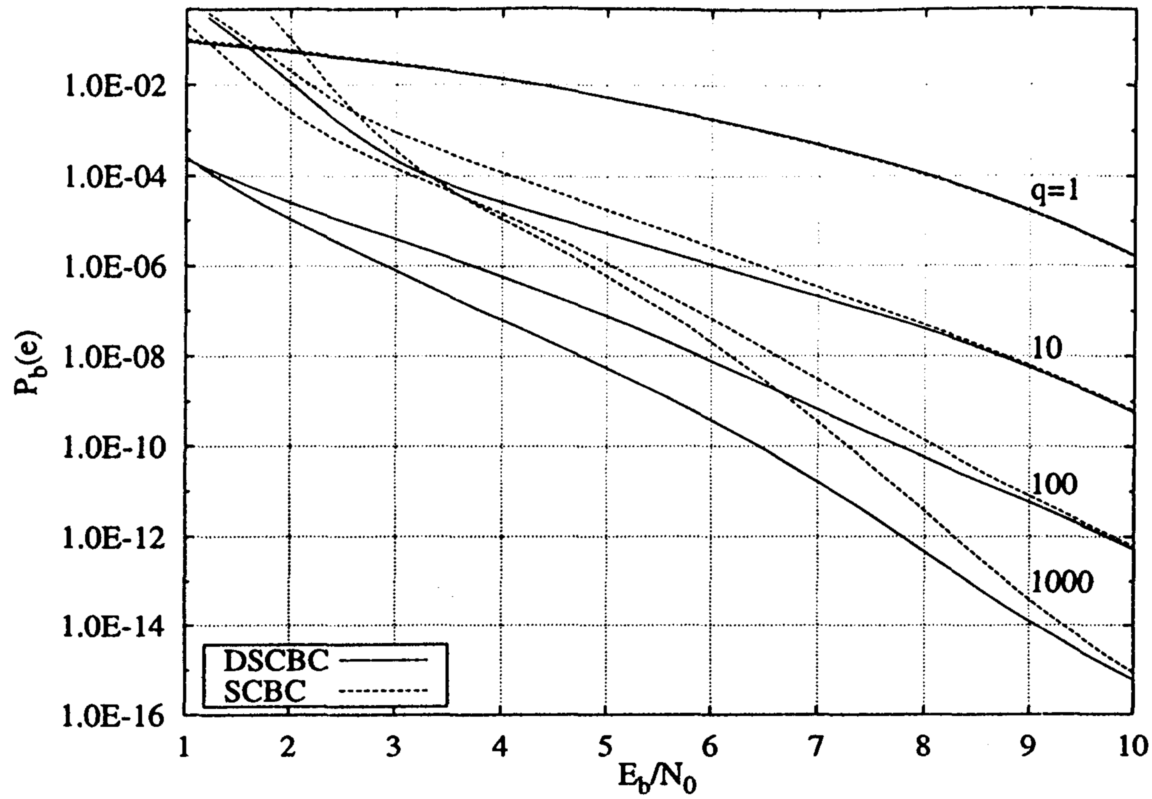

Fig. 6. Comparison between analytical upper bounds to the bit error probability for the DSCBC of Example 2 and for an SCBC obtained by concatenating a (7,3) code with the (15,7) BCH code. The parameter q represents, for both DSCBC and SCBC, the code latency expressed in terms of the number of input words

are both non recursive; for the second code, DSCCC2, the middle is a recursive encoder and the inner is non recursive; finally, for the third code, DSCCC3, both middle and inner are recursive encoders. In Table II the main design parameters of the three DSCCCs are reported.

To check the accuracy of the bounds on $\alpha(h_m)$ and α_M, we have evaluated the coefficients $A(h,N)$ defined in (17) for two large values of N, $N_1 = 10,000$ and $N_2 = 20,000$. Then, we have computed the coefficients $B(h)$ defined through

$$B(h) \triangleq \frac{\log\left[\frac{A(h,N_2)}{A(h,N_1)}\right]}{\log\left[\frac{N_2}{N_1}\right]} \qquad (36)$$

If the asymptotic (for large N) analysis of Section IV is true, then the coefficients $B(h)$, based on their definition (36) and on the definition (17) of $A(h,N)$, should provide a good estimate of the exponent $\alpha(h)$ of N defined in (18). To check this, we have reported the coefficients $B(h)$ versus h in Fig. 7 for the three codes DSCCC1, DSCCC2 and DSCCC3.

Encoder description	$G(D)$	d_1	$d_{f,\text{eff}}$	d_3
Rate 1/2 NR	$\begin{bmatrix} 1+D+D^2 & 1+D^2 \end{bmatrix}$	5	6	7
Rate 2/3 NR	$\begin{bmatrix} 1+D & D & 1 \\ 1+D & 1 & 1+D \end{bmatrix}$	4	3	4
Rate 2/3 R	$\begin{bmatrix} 1 & 0 & \frac{1+D^2}{1+D+D^2} \\ 0 & 1 & \frac{1+D}{1+D+D^2} \end{bmatrix}$	-	4	3
Rate 3/4 NR	$\begin{bmatrix} 1+D & 1+D & D & 0 \\ 1+D & D & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$	4	3	3
Rate 3/4 R	$\begin{bmatrix} 1 & 0 & 0 & \frac{D}{1+D} \\ 0 & 1 & 0 & \frac{1+D+D^2}{1+D^2} \\ 0 & 0 & 1 & \frac{1}{1+D^2} \end{bmatrix}$	-	3	3

TABLE I

CONSTITUENT CONVOLUTIONAL ENCODERS USED FOR CONSTRUCTING DOUBLE SERIAL CONCATENATED CODES.

Code description (Rate 1/4)	CCs 1/2	2/3	3/4	h_m	$\alpha(h_m)$	$h(\alpha_M)$	α_M
DSCCC1	NR	NR	NR	4	-8 (-6)	-	-
DSCCC2	NR	R	NR	4	-8 (-6)	28 (28)	-3 (-3)
DSCCC3	NR	R	R	5	-8 (-6)	6 (3)	-7 (-6)

TABLE II

THREE DOUBLE SERIAL RATE 1/4 CONCATENATED CONVOLUTIONAL CODES. THE NUMBER IN PARENTHESES ARE THE VALUES OF THE PARAMETERS OBTAINED USING THE BOUNDS OF SECTION IV.

Consider first the code DSCCC1 (continuous curve). The values of $B(h)$ keep on increasing with h, according to the value $\alpha_M = \infty$ for $N \to \infty$ predicted by (31). On the other hand, the value $B(h_m)$ is equal to -8, in agreement with the result (23) which stated $\alpha(h_m) \leq -6$. Passing to code DSCCC2 (dashed curve), we find the same value of $B(h_m)$, whereas the largest value $\max_h B(h) = -3$, yielding an interleaver gain. The result (32) predicted $\alpha_M \leq -3$, which is in perfect agreement with our finding. Finally, the dotted curve of code DSCCC3 yields $\max_h B(h) = -7$, also in agreement with (35) that stated $\alpha_M \leq -6$.

To compare simple and double serial concatenation of convolutional codes, we have constructed

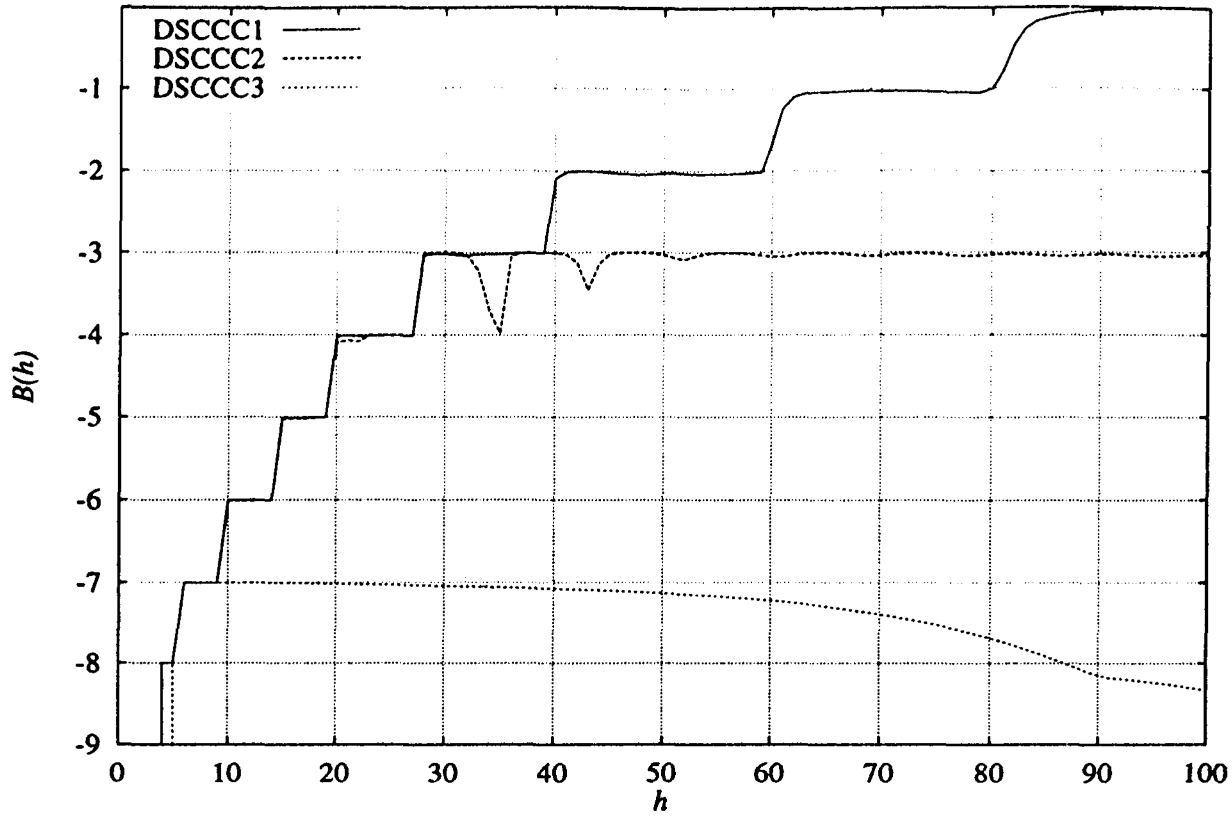

Fig. 7. The coefficients $B(h)$ versus h for the three DSCCCs of Table II.

two rate 1/4 DSCCC and SCCC. The DSCCC is simply DSCCC3, whereas the SCCC is obtained concatenating a 4-state rate 1/2 recursive convolutional encoder with a 4-state rate 2/4 recursive convolutional encoder. The interleaver lengths are chosen so as to yield the same latency for the two schemes. The results in terms of bit error probability are reported in Fig. 8, and show also in this case the clear superiority of the DSCCC.

VI. Iterative decoding of double serially concatenated codes

In Sections 2 and 3, we have shown by examples and analytical findings that DSCCCs can outperform SCCCs, 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 present an iterative algorithm, which is an extension of the one introduced in [7] to decode serially concatenated convolutional codes, with complexity not significantly higher than that needed to separately decode the three CCs, which approaches the maximum-likelihood performance.

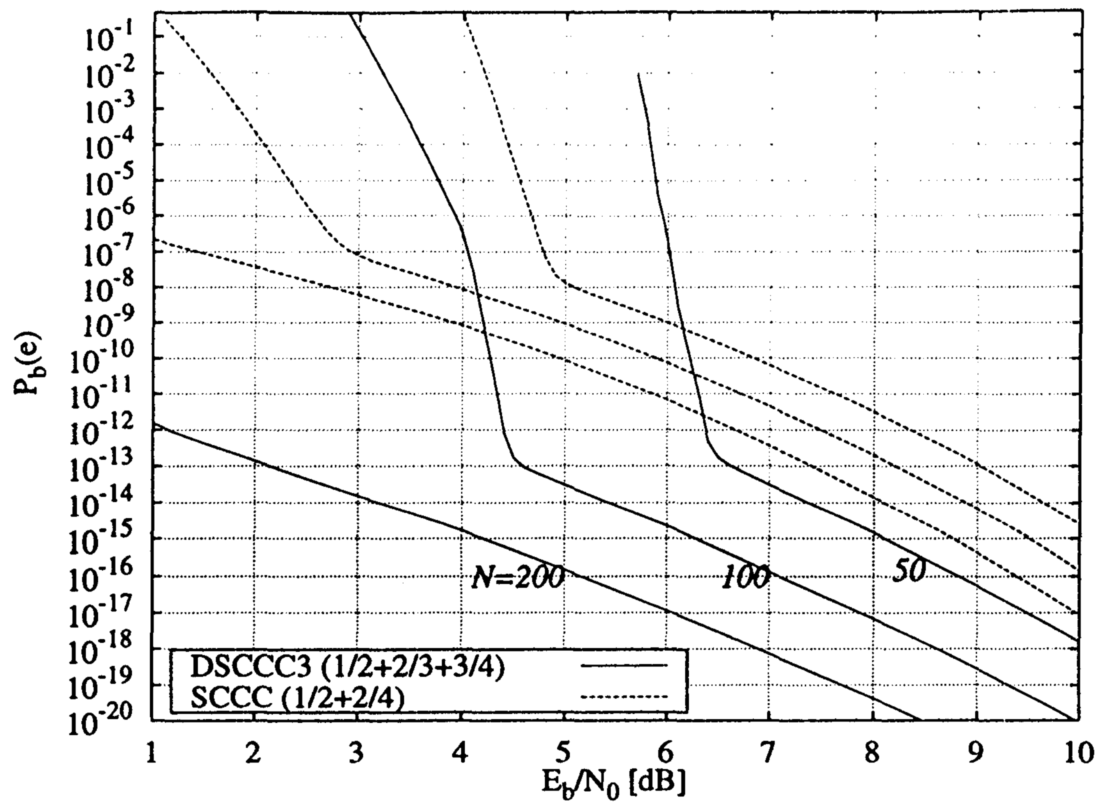

Fig. 8. Comparison between analytical upper bounds to the bit error probability for the code DSCCC3 of Table II and for an SCCC obtained by concatenating a 4-state outer recursive convolutional encoder with rate 1/2 and a 4-state inner recursive convolutional encoder with rate 2/4. The parameter N represents, for both DSCCC and SCCC, the code latency expressed in terms of the number of input bits

Because of the importance in applications, all examples will refer to DSCCCs, although the decoding algorithm can be applied to DSCBCs as well.

A. The iterative decoding algorithm for DSCCCs

The core of the new decoding procedure consists of an a-posteriori-probability (APP) decoding algorithm (it will be described in next Subsection) applied to the CCs. The functionality of the APP decoder to be used for DSCCCs are 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 sliding-window APP algorithm described in [13].

A functional diagram of the iterative decoding algorithm for DSCCCs is presented in Fig. 9, where we also show a double turbo encoder, using three CCs and two interleaver, and its iterative decoder to enlighten analogies as well as differences.

Let us explain how the algorithm works, according to the blocks of Fig. 9. The blocks labeled "APP" 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 APP decoding algorithm. Differently from the iterative decoding algorithm employed for turbo decoding (also shown in Fig. 9), in which the APP algorithm only computes the LPDF of input symbols conditioned on the code constraints based on the unconstrained LPDF of input symbols, the DSCC decoder must fully exploit the potential of the APP algorithm, which can, in fact, update both LPDF of input and output symbols based on the code constraints. Both outputs of APP directly generate the "extrinsic" informations required for iterative decoding. So there is no need to subtract the unconstrained input LPDF from the output.

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

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

During the first iteration of the DSCCC algorithm, the block "APP INNER" is fed with the demodulator soft output, consisting of the LPDF of symbols received from the channels, i.e. of output symbols of the inner encoder. The LPDF is processed by the first APP decoder that computes the LPDF relative to the input symbols conditioned on the inner code constraints. This information is passed through the second inverse interleaver (block labeled "π_2^{-1}"). As the input symbols of the inner code, after inverse interleaving, correspond to the output symbols of the middle code, they are sent to the "APP MIDDLE" block in the upper entry, which corresponds to output symbols. The middle APP 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 output symbols are fed back to the APP inner decoder in the second iteration. The LPDF of input symbols, instead, are passed through the first inverse interleaver (block labeled "π_1^{-1}"). Since the input symbols of the middle code, after inverse interleaving, correspond to the output symbols of the outer code, they are sent to the "APP OUTER" block in the upper entry, which corresponds to output symbols. The LPDF of input symbols will be used in the final iteration to recover the information bits, whereas the LPDF of output symbols are interleaved and fed back to the APP middle decoder.

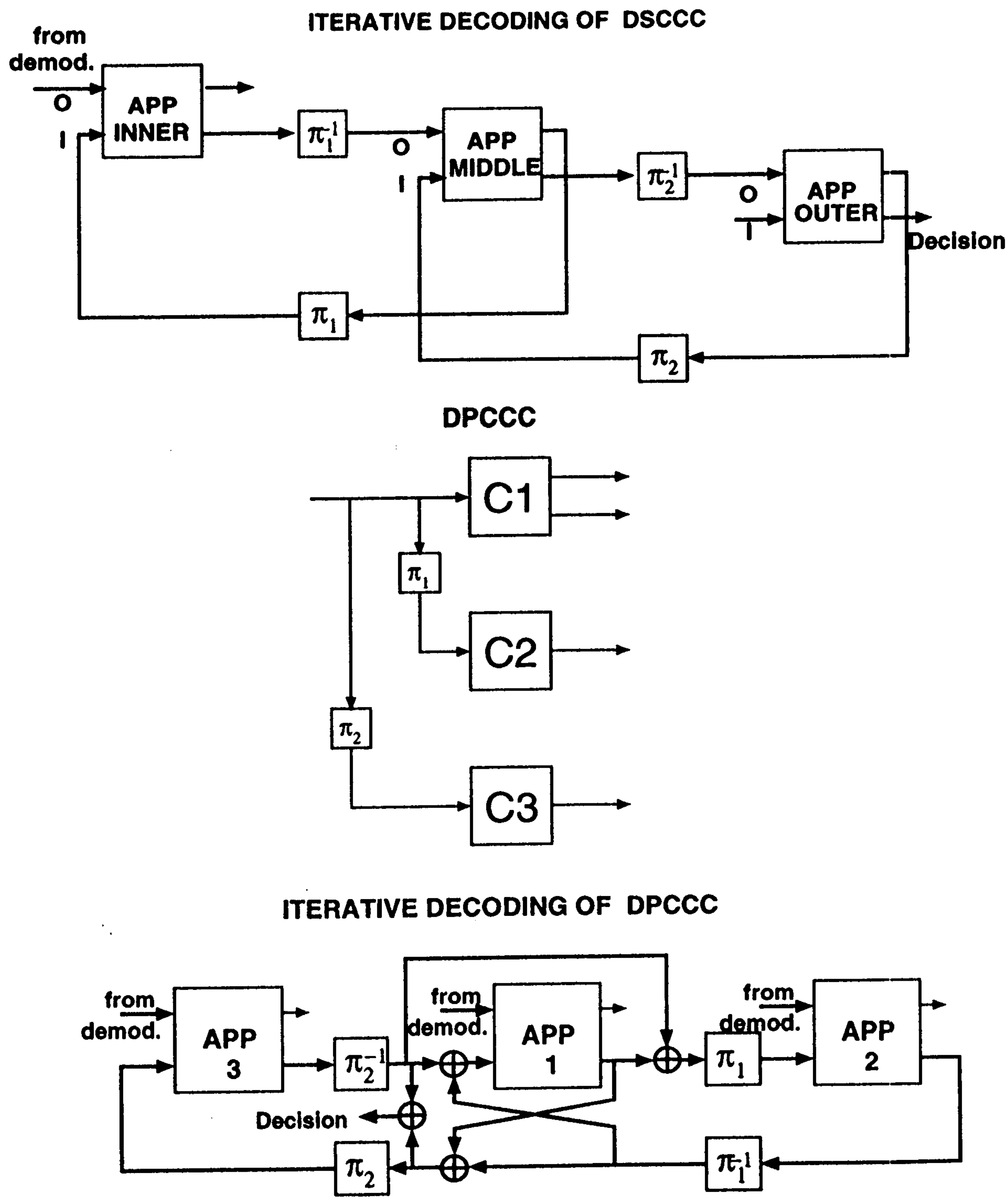

Fig. 9. Iterative decoding algorithm for double serially and parallel concatenated convolutional codes. The encoder for the double parallel concatenated code is also shown.

B. The additive algorithm for bit a-posteriori probability evaluation

As seen in Fig. 9, the main building block of the iterative decoder, for both DSCCCs and DPCCCs, is the one implementing the a-posteriori probability evaluation, denoted by APP. Its main characteristics have been illustrated in [14]. Here, we will briefly describe the input-output relationship needed to implement the APP modules in Fig. 9. The description will be based on the the trellis section shown in Fig 10.

Consider a code with p input bits and q output bits $\{0,1\}$, Let $\mathbf{U}_k(e)$ represent $U_{k,i}(e)$; $i = 1,2,\cdots,p$ the input bits on a trellis edge at time k and let $\mathbf{C}_k(e)$ represent $C_{k,i}(e)$; $i = 1,2,\cdots,q$ the output bits on the same trellis edge at time k.

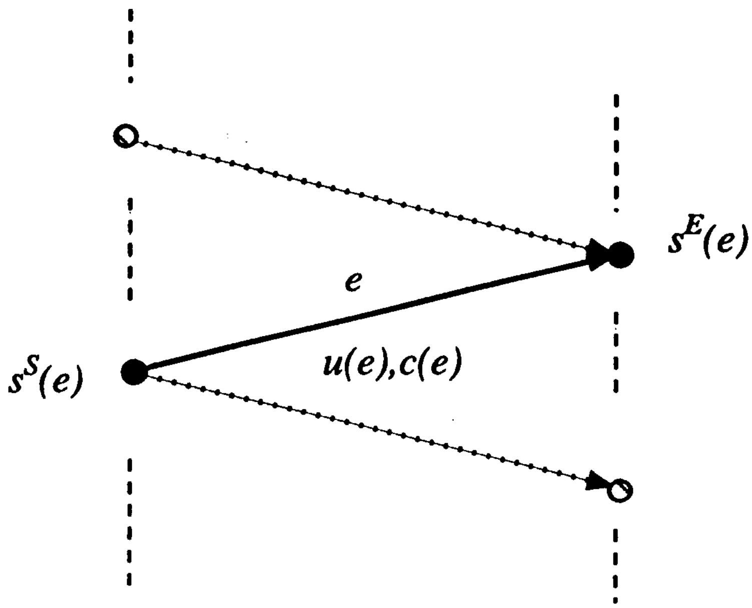

Fig. 10. Trellis Section.

Define the reliability of a bit Z taking values $\{0,1\}$ at time k as

$$\lambda_k[Z;\cdots] \triangleq \log \frac{P_k[Z=1;\cdot]}{P_k[Z=0;\cdot]}$$

It is in fact the difference of LPDFs for bit values 1, and 0. The second argument in the brackets, denoted by a dot, may represent I, the input to, or O, the output from the APP module.

We use the following identity

$$a = \log[\sum_{i=1}^{L} e^{a_i}] = \max_i\{a_i\} + \delta(a_1,\ldots,a_L) \triangleq \max_i{}^*\{a_i\}$$

where $\delta(a_1,\ldots,a_L)$ is the correction term which can be computed using a look-up table [15], thus defining the "max*" operation as a maximization (compare-select) plus a correction term (lookup table).

Assuming a binary modulation scheme (e.g. BPSK, or, also, QPSK if we consider QPSK as two independent parallel BPSK), and normalizing the received samples $\{y_{k,i}\}$ at the output of the receiver matched filter in such a way that the additive complex noise samples have unit variance per dimension, we can write

$$y_{k,i} = \sqrt{\frac{2E_s}{N_o}}(2c_{k,i} - 1) + n_{k,i} \ .$$

The relationships relating the input-output extrinsic reliabilities of information and code bits are given by the following expressions

$$\begin{aligned}\lambda_k(U_{k,j};O) = \\ &\max^*_{e:u_{k,j}(e)=1}\{\alpha_{k-1}[s^S(e)] + \sum_{\substack{i=1\\i\neq j}}^{p} u_{k,i}(e)\lambda_k[U_{k,i};I] + \sum_{i=1}^{q} c_{k,i}(e)\lambda_k[C_{k,i}(e);I] + \beta_k[s^E(e)]\} \\ &- \max^*_{e:u_{k,j}(e)=0}\{\alpha_{k-1}[s^S(e)] + \sum_{\substack{i=1\\i\neq j}}^{p} u_{k,i}(e)\lambda_k[U_{k,i};I] + \sum_{i=1}^{q} c_{k,i}(e)\lambda_k[C_{k,i}(e);I] + \beta_k[s^E(e)]\} \quad (37)\end{aligned}$$

$$\begin{aligned}\lambda_k(C_{k,j};O) = \\ &\max^*_{e:c_{k,j}(e)=1}\{\alpha_{k-1}[s^S(e)] + \sum_{i=1}^{p} u_{k,i}(e)\lambda_k[U_{k,i};I] + \sum_{\substack{i=1\\i\neq j}}^{q} c_{k,i}(e)\lambda_k[C_{k,i};I] + \beta_k[s^E(e)]\} \\ &- \max^*_{e:c_{k,j}(e)=0}\{\alpha_{k-1}[s^S(e)] + \sum_{i=1}^{p} u_{k,i}(e)\lambda_k[U_{k,i};I] + \sum_{\substack{i=1\\i\neq j}}^{q} c_{k,i}(e)\lambda_k[C_{k,i};I] + \beta_k[s^E(e)]\} \quad (38)\end{aligned}$$

where the quantities $\alpha_k(\cdot)$ and $\beta_k(\cdot)$ are obtained through the following **forward** and **backward** recursions:

$$\alpha_k(s) = \max^*_{e:s^E(e)=s}\{\alpha_{k-1}[s^S(e)] + \sum_{i=1}^{p} u_{k,i}(e)\lambda_k[U_{k,i};I] + \sum_{i=1}^{q} c_{k,i}(e)\lambda_k[C_{k,i};I] + h_{\alpha_k} \quad (39)$$

$$\beta_k(s) = \max^*_{e:s^S(e)=s}\{\beta_{k+1}[s^E(e)] + \sum_{i=1}^{p} u_{k+1,i}(e)\lambda_{k+1}[U_{k+1,i};I] + \sum_{i=1}^{q} c_{k+1,i}(e)\lambda_{k+1}[C_{k+1,i};I] + h_{\beta_k} \quad (40)$$

with initial values, $\alpha_0(s) = 0$, if $s = 0$ (initial zero state) and $\alpha_0(s) = -\infty$, otherwise, and $\beta_n(s) = 0$, if $s = 0$ (final zero state) and $\beta_n(s) = -\infty$, otherwise. To yield continuous decoding employing the *sliding window* algorithm described in [16], the initial condition for the β recursion is modified as $\beta_n(s) = 1/N_s$, $\forall s$, where N_s is the number of states in the trellis.

The h_{α_k} and h_{β_k} are normalization constants used to prevent form buffer overflows in a hardware implementation of the algorithm.

The APP algorithm has strong similarities with the Viterbi algorithm when used in the forward and backward directions, except for a correction term that is added when compare-select operations are performed.

When the code to which the APP algorithm refers to is either the inner code in DSCCCs (or any code in DPCCCs), then $\lambda_k[C_{k,i}; I] = 2\sqrt{\frac{2E_s}{N_o}}y_{k,i}$ in (37). For the outer code of DSCCCs (38) is used in the iterations with $\lambda_k[U_{k,i}; I] = 0$. To make the final decisions, the outer code uses (37), again with $\lambda_k[U_{k,i}; I] = 0$. The middle code uses both (37) and (38) during the iterations.

B.1 Simulation results

We have applied the previously described decoding algorithm to four concatenated codes. All use random interleavers yielding an input latency of 256 bits and have the same rate equal to 1/4. The first is the DSCCC3 of Table II, the second is the SCCC obtained by concatenating a 4-state rate 1/2 CC with a 4-state rate 2/4 CC; the third is a PCCC (turbo code) obtained concatenating two equal 4-state rate 1/2 CCs, and, finally, the last one is a double turbo code, employing two interleavers and three equal 4-state recursive convolutional codes with generating matrix

$$G(D) = \left[1, \frac{1+D^2}{1+D+D^2}\right] .$$

The simulation results in terms of bit error probability versus E_b/N_0 are reported in Fig. 11 for 5 and 10 iterations of the decoding algorithms. The curves show a clear behavior; we analyze it for 10 iterations. For very low signal-to-noise ratios, below 0.5 dB, the performance hierarchy points to the PCCC as the best, followed by the DPCCC, the SCCC and then DSCCC. Between 0.5 and 1.2 dB, the SCCC is the best, followed by the DPCCC, the PCCC and then the DSCCC. At 1.2 dB, the DSCCC starts outperforming the PCCC; at 1.4 dB it outperforms the the DPCCC, and at 1.75 dB it becomes the best code. The PCCC shows the well known phenomenon of "error floor", which is more precisely a change of slope, around 10^{-5}; the SCCC and DPCCC also have a sensible change of slope around 10^{-6}, whereas the DSCCC seems immune from this phenomenon, and would allow reaching very low bit error probabilities. The advantage is clear, in that very good performance can be obtained by the DSCCC even with small-medium interleavers, in those situations where simple and double turbo codes (and also SCCC) exhibit change of slope in their bit error probability curves.

Sometimes, system constraints require that the bit stream is organized in frames, which are then either accepted or rejected by the decoder. In those situations, the performance measure is

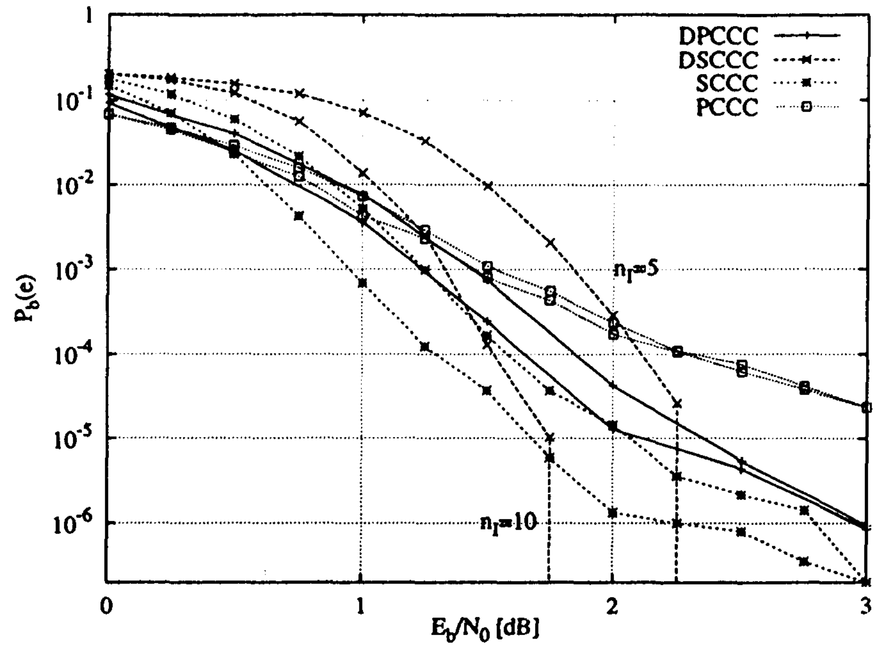

Fig. 11. Simulation results for the bit error probability of four rate 1/4 concatenated coding schemes. The parameter n_I is the number of iterations of the decoding algorithms.

the *frame error probability*, rather than the bit error probability, i.e. the probability that there is at least one error in a frame. We have evaluated the frame error probability[7] for the same four concatenated coding schemes previously examined with respect to the bit error probability, assuming a frame size of 256.

The simulation results are shown in Fig. 12 for 5 and 10 iterations of the decoding algorithms. With respect to the frame error rate, the PCCC is the worst of the four. In fact, the crossing between its performance and that of the DSCCC happens at 0.8 dB, instead of 1.2 dB for the previous curves, and the SCCC is now uniformly better than the PCCC. As to the DPCCC, it becomes worse than the DSCCC at 1.5 dB. The same behavior previously described concerns the frame error floors.

VII. Conclusions

We have proposed double serially concatenated codes with two interleavers: they consist of the cascade of an outer encoder, an interleaver permuting the outer codewords bits, a middle encoder, another interleaver permuting the middle codewords bits and an inner encoder whose input words are the permuted middle codewords. For these new coding schemes, we have obtained upper

[7]This is, to our knowledge, the first time that concatenated codes with interleavers are compared using this parameter.

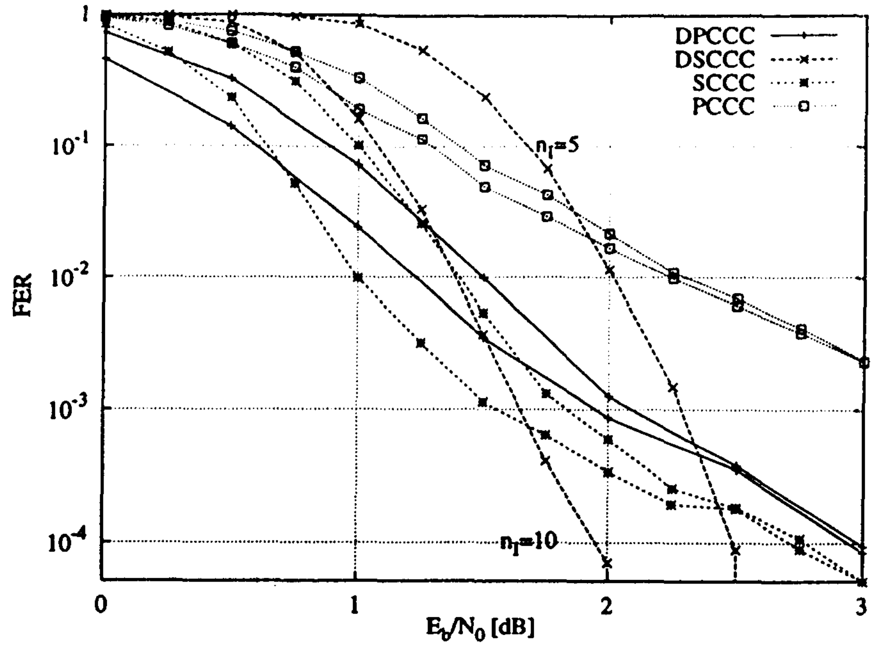

Fig. 12. Simulation results for the frame error probability of four rate 1/4 concatenated schemes. The parameter n_I is the number of iterations of the decoding algorithms.

bounds to the average maximum-likelihood bit error probability, and derived design guidelines for the outer, middle, and inner encoders that maximize the interleaver gain and the asymptotic slope of the error probability curves. Finally, we have proposed a low-complexity, iterative decoding algorithm. Both analytical and simulation results, in which the performance of the new scheme have been compared with parallel concatenated convolutional codes, known as "turbo codes", and with the recently proposed serially concatenated convolutional codes, have been presented. They show that the new scheme offer superior performance when maximum-likelihood decoded. Moreover, with the suboptimum decoding algorithm, the drawback of the error "floor" to the bit error probability typical of PCCC, and, to a lower extent, of the SCCC, is pushed down to very low bit error probabilities. Thus, there is no need for very large interleavers to obtain low bit error probability, as for turbo codes, and, as a consequence, the new scheme can be adopted when high performance are sought at not too low signal-to-noise ratios with a small decoding latency.

VIII. Acknowledgement

The research in this article has been partially supported by NATO under Research Grant CRG 951208, and by Qualcomm, Inc. Sergio Benedetto and Guido Montorsi also acknowledge the support of Italian Space Agency (ASI). The research described in this paper was partially carried out at the Jet Propulsion Laboratory, California Institute of Technology, under contract with National Aeronautics and Space Administration (NASA).

REFERENCES

[1] G.D. Forney Jr., *Concatenated Codes*, M.I.T., Cambridge, MA, 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] Joachim Hagenauer and Peter Hoeher, "Concatenated Viterbi Decoding", in *Proceedings of Fourth Joint Swedish-Soviet Int. Workshop on Information Theory*, Gotland, Sweden, Studenlitteratur, Lund, Aug. 1989, pp. 29–33.

[4] Claude Berrou, Alain Glavieux, and Punya Thitimajshima, "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes", in *Proceedings of ICC'93*, Geneve, Switzerland, May 1993, pp. 1064–1070.

[5] J.H. Lodge, R.J. Young, P. Hoeher, and J. Hagenauer, "Separable MAP "Filters" for the Decoding of Product Codes And Concatenated Codes", in *Proceedings of ICC'93*, Geneve, Switzerland, May 1993, pp. 1740–1745.

[6] D. Divsalar and F. Pollara, "Turbo Codes for PCS Applications", in *Proceedings of ICC'95*, Seattle, Washington, June 1995.

[7] Sergio Benedetto, Guido Montorsi, Dariush Divsalar, and Fabrizio Pollara, "Serial concatenation of interleaved codes: Performance analysis, design and iterative decoding", *JPL TDA Progress Report 42-126*, Aug. 1996.

[8] Sergio Benedetto and Guido Montorsi, "Average Performance of Parallel Concatenated Block Codes", *Electronics Letters*, vol. 31, no. 3, pp. 156–158, Feb. 1995.

[9] Sergio Benedetto and Guido Montorsi, "Unveiling turbo-codes: some results on parallel concatenated coding schemes", *IEEE Transactions on Information Theory*, vol. 42, no. 2, pp. 409–429, Mar. 1996.

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

[11] L.R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate", *IEEE Transactions on Information Theory*, pp. 284–287, Mar. 1974.

[12] R.J. McEliece, "On the BCJR trellis for linear block codes", *IEEE Transactions on Information Theory*, vol. vol. IT-42, pp. pp. 1072–1091, July 1996.

[13] Sergio Benedetto, Dariush Divsalar, Guido Montorsi, and Fabrizio Pollara, "Soft-output decoding algorithms for continuous decoding of parallel concatenated convolutional codes", in *Proceedings of ICC'96*, Dallas, Texas, June 1996.

[14] Sergio Benedetto, Dariush Divsalar, and Fabrizio Pollara, "A soft-input soft-output APP module to decode concatenated codes", *IEEE Communications Letters*, Jan. 1997.

[15] Patrick Robertson, Emmanuelle Villebrun, and Peter Hoeher, "A Comparison of Optimal and Sub-Optimal MAP Decoding Algorithms Operating in the Log Domain", in *Proceedings of ICC'95*, Seattle, Washington, June 1995, pp. 1009–1013.

[16] Sergio Benedetto, Guido Montorsi, Dariush Divsalar, and Fabrizio Pollara, "Soft-output decoding algorithms in iterative decoding of turbo codes", *JPL TDA Progress Report 42-124*, Feb. 1996.