Turbo codes with rate-m/(M+1) constituent convolutional codes

Catherine Douillard, Claude Berrou

Turbo Codes With Rate-$m/(m+1)$ Constituent Convolutional Codes

Catherine Douillard, *Member, IEEE,* and Claude Berrou, *Member, IEEE*

***Abstract*—The original turbo codes (TCs), presented in 1993 by Berrou *et al.*, consist of the parallel concatenation of two rate-1/2 binary recursive systematic convolutional (RSC) codes. This paper explains how replacing rate-1/2 binary component codes by rate-$m/(m+1)$ binary RSC codes can lead to better global performance. The encoding scheme can be designed so that decoding can be achieved closer to the theoretical limit, while showing better performance in the region of low error rates. These results are illustrated with some examples based on double-binary ($m=2$) 8-state and 16-state TCs, easily adaptable to a large range of data block sizes and coding rates. The double-binary 8-state code has already been adopted in several telecommunication standards.**

***Index Terms*—Iterative decoding, permutation, rate-$m/(m+1)$ recursive systematic convolutional (RSC) code, tailbiting code, turbo code (TC).**

I. Introduction

A CLASSICAL turbo code (TC) [1] is a parallel concatenation of two binary recursive systematic convolutional (RSC) codes based on single-input linear feedback shift registers (LFSRs).

The use of multiple-input LFSRs, which allows several information bits to be encoded or decoded at the same time, offers several advantages compared with classical TCs. In the past, the parallel concatenation of multiple-input LFSRs had mainly been investigated for the construction of turbo trellis-coded modulation schemes [2]–[4], based on Ungerboeck trellis codes. Actually, the combination of such codes, providing high natural coding rates with high-order modulations, leads to very powerful coded modulation schemes.

In this paper, we propose the construction of a family of TCs calling for RSC constituent codes based on m-input LFSRs, that outperforms classical TCs. We provide two examples of TCs with reasonable decoding complexity that allow decoding to be achieved very close to the theoretical limit, and at the same time, show good performance in the region of low error rates.

Paper approved by R. D. Wesel, the Editor for Coding and Communication Theory of the IEEE Communications Society. Manuscript received May 1, 2003; revised August 6, 2004.

The authors are with the Electronics Department, GET/École Nationale Supérieure des Télécommunications de Bretagne, CS 83818 – 29238 Brest Cedex 3, France (e-mail: catherine.douillard@enst-bretagne.fr; claude.berrou@enst-bretagne.fr).

Digital Object Identifier 10.1109/TCOMM.2005.857165

Section II describes the structure adopted for m-input RSC component encoders, along with some conditions that guarantee large free distances, regardless of the value of m.

In Section III, we describe the turbo-encoding scheme, and the advantages of this construction compared with classical TCs.

Section IV presents some practical examples of TCs with $m=2$ and their simulated performance. The 8-state family has already been adopted in the digital video broadcasting (DVB) standards for return channel via satellite (DVB-RCS) [5] and the terrestrial distribution system (DVB-RCT) [6], and also in the 802.16a standard for local and metropolitan area networks [7]. Combined with the powerful technique of circular trellises, this $m=2$ TC offers good performance and versatility for encoding blocks with various sizes and rates, while keeping reasonable decoding complexity. Replacing the 8-state component encoder by a 16-state encoder allows better performance at low error rates, at the price of a doubled decoding complexity. Minimum Hamming distances are increased by 30%–50%, with regard to 8-state TCs, and allow frame-error rate (FER) curves to decrease below 10^{-7} without any noticeable change in the slope (the so-called *flattening* effect).

Finally, conclusions and perspectives are summarized in Section V.

II. Rate-$m/(m+1)$ RSC Encoders Based on m-Input LFSRs

In this section, we will define the constituent RSC codes to be used in the design of the proposed TCs. Fig. 1 depicts the general structure of the RSC encoder under study. It involves a single ν-stage LFSR, whose ν-row and ν-column generator matrix is denoted $\mathbf{G}$. At time i, the m-component input vector $\mathbf{d}_i = (d_{i,1} \cdots d_{i,l} \cdots d_{i,m})^T$ is connected to the ν possible taps via a connection grid represented by a ν-row and m-column binary matrix denoted $\mathbf{C}$. The tap column vector at time i, $\mathbf{T}_i$, is then given by

$$\mathbf{T}_i = \mathbf{C}\mathbf{d}_i. \tag{1}$$

In order to avoid parallel transitions in the corresponding trellis, the condition $m \leq \nu$ has to be satisfied, and matrix $\mathbf{C}$ has to be full rank.

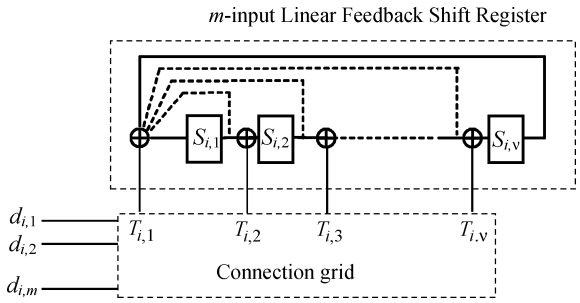

Fig. 1. General structure of a rate-$m/(m+1)$ RSC encoder with code memory ν.

Except for very particular cases, this encoder is not equivalent to a single-input encoder fed successively by $d_{i,1}, d_{i,2}, \ldots, d_{i,m}$, that is, the m-input encoder is not generally decomposable.

The redundant output of the machine, not represented in Fig. 1, is calculated at time i as

$$y_i = \sum_{j=1,\ldots,m} d_{i,j} + \mathbf{R}\mathbf{S}_i \tag{2}$$

where $\mathbf{S}_i$ denotes the ν-component column vector describing the encoder state at time i, and $\mathbf{R}$ is the ν-component row-redundancy vector. The pth component of $\mathbf{R}$ is equal to one if the pth component of $\mathbf{S}_i$ is present in the calculation of y_i, and zero, otherwise.

The code being linear, we assume that the "all zero" sequence is encoded. Let us define a *return to zero* (RTZ) sequence as an input sequence of a recursive encoder, that makes the encoder leave state $\mathbf{0}$ and return to it again. Calculating the minimum free distance of such a code involves finding the RTZ sequence path with minimum output Hamming weight.

Leaving the null path at time i implies that $\mathbf{S}_i \equiv \mathbf{0}$ and at least one component of $\mathbf{d}_i$ is equal to one. In this case, (2) ensures that the Hamming weight of $(d_{i,1}, d_{i,2}, \ldots, d_{i,m}, y_i)$ is at least two when leaving the reference path, since the inversion of one component of $\mathbf{d}_i$ implies the inversion of y_i.

Moreover, since

$$\mathbf{S}_{i+1} = \mathbf{G}\mathbf{S}_i + \mathbf{T}_i \tag{3}$$

it can be shown that y_i may also be written as

$$y_i = \sum_{j=1,\ldots,m} d_{i,j} + \mathbf{R}\mathbf{G}^{-1}\mathbf{S}_{i+1} \tag{4}$$

on the condition that

$$\mathbf{R}\mathbf{G}^{-1}\mathbf{C} \equiv \mathbf{0}. \tag{5}$$

Consequently, if the code is devised to verify condition (5), (4) ensures that if the RTZ sequence retrieves the all-zero reference path at time i ($\mathbf{S}_{i+1} \equiv \mathbf{0}$) because one of the information bits $d_{i,j}$ is equal to one, y_i is also equal to one.

Hence, relations (2) and (4) together guarantee that the minimum free distance of the unpunctured code, whose rate is $R = m/(m+1)$, is at least four, whatever m. The two codes proposed in this paper for $m = 2$ meet this requirement.

The minimum Hamming distance of a concatenated code being larger than that of its constituent codes, provided that the permutation function is carefully devised, we can imagine that large minimum distances may be obtained for TCs, for low as well as for high coding rates.

Choosing large values of m implies high decoding complexity, because ν also has to be large, and 2^m paths have to be processed for each trellis node. For this reason, only low values of m can be contemplated for practical applications for the time being (typically, $m = 2$, possibly 3).

Up to now, we have only investigated the case $m = 2$ in order to construct practical coding and decoding schemes in this new family of TCs. In the following, we call such $m = 2$ RSC or turbo encoders *double binary*. Double-binary RSC codes have a natural rate of 2/3. When higher coding rates are required, a simple regular or quasi-regular puncturing pattern is applied.

The decoding solution calling for the application of the *maximum a posteriori* (MAP) algorithm based on the dual code [8] can also be considered for large values of m, since it requires fewer edge computations than the classical MAP algorithm for high coding rates. However, for practical implementations, its application in the log-domain means the computation of transition metrics with a far greater number of terms, and the computation of the $\max^*$ (Jacobian) function requires great precision. Consequently, the relevance of this method for practical use is not so obvious when the rate of the mother code is not close to 1.

III. Block Turbo Coding With Rate-$m/(m+1)$ Constituent RSC Codes

The TC family proposed in this paper calls for the parallel concatenation of two identical rate-$m/(m+1)$ RSC encoders with m-bit word interleaving. Blocks of k bits, $k = mN$, are encoded twice by this bidimensional code, whose rate is $m/(m+2)$.

A. Circular RSC Codes

Among the different techniques aiming at transforming a convolutional code into a block code, the best way is to allow any state of the encoder as the initial state, and to encode the sequence so that the final state of the encoder is equal to the initial state. The code trellis can then be viewed as a circle, without any state discontinuity. This termination technique, called *tailbiting* [9], [10] or circular, presents three advantages in comparison with the classical trellis-termination technique using tail bits to drive the encoder to the all-zero state. First, no extra bits have to be added and transmitted; thus, there is no rate loss, and the spectral efficiency of the transmission is not reduced. Next, when classical trellis termination is applied for TCs, a few codewords with input Hamming weight one may appear at the end of the block (in both coding dimensions), and can be the cause of a marked decrease in the minimum Hamming distance of the composite code. With tailbiting RSC codes, only codewords with minimum input weight two have to be considered. In other words, tailbiting encoding avoids any side effects. Moreover, in a tailbiting or circular trellis, the past is also the future and vice versa. This means that a non-RTZ sequence produces effects on

the whole set of redundant symbols stemming from the encoder, around the whole circle. Consequently, the output weights associated with non-RTZ sequences are large, and do not contribute to the minimum Hamming distance of the code.

In practice, the circular encoding of a data block consists of a two-step process [9]. At the first step, the information sequence is encoded from state **0** and the final state is memorized. During this first step, the outputs bits are ignored. The second step is the actual encoding, whose initial state is a function of the final state previously memorized. The double encoding operation represents the main drawback of this method, but in most cases, it can be performed at a frequency much higher than the data rate.

The iterative decoding of such codes involves repeated and continuous loops around the circular trellis. The number of loops performed is equal to the required number of iterations. The state probabilities or metrics, according to the chosen decoding algorithm, computed at the end of each turn are used as initial values for the next turn. With this method, the initial, or final, state depends on the encoded information block and is *a priori* unknown to the decoder at the beginning of the first iteration. If all the states are assumed to be equiprobable at the beginning of the decoding process, some side errors may be produced by the decoders at the beginning of the first iteration. These errors are removed at the subsequent iterations, since final state probabilities or metrics computed at the end of the previous iteration are used as initial values.

B. Permutation

Among the numerous permutation models that have been suggested up to now, the apparently most promising ones, in terms of minimum Hamming distances, are based on regular permutation calling for circular shifting [11] or the co-prime [12] principle. After writing the data in a linear memory, with address i ($0 \leq i \leq N-1$), the information block is likened to a circle, both extremities of the block ($i = 0$ and $i = N-1$) then being contiguous. The data are read out such that the jth datum read was written at the position i, given by

$$i = \Pi(j) = Pj + i_0 \mod N \tag{6}$$

where the skip value P is an integer, relatively prime with N, and i_0 is the starting index. This permutation does not require the block to be seen as rectangular; that is, N may be any integer.

In [13] and [14], two very similar modifications of (6) were proposed, which generalize the permutation principle adopted in the DVB-RCS/RCT or IEEE802.16a TCs. In the following, we will consider the *almost regular permutation* (ARP) model detailed in [14], which changes relation (6) into

$$i = \Pi(j) = Pj + Q(j) + i_0 \mod N \tag{7}$$

where $Q(j)$ is an integer, whose value is taken in a limited set $\{0, Q_1, Q_2, \ldots, Q_{C-1}\}$, in a cyclic way. C, called the *cycle* of the permutation, must be a divider of N and has a typical value of four or eight. For instance, if $C = 4$, the permutation law is defined by

$$\begin{aligned}
&\text{if } j = 0 \bmod 4, \quad i = \Pi(j) = Pj + 0 + i_0 \mod N \\
&\text{if } j = 1 \bmod 4, \quad i = \Pi(j) = Pj + Q_1 + i_0 \mod N \\
&\text{if } j = 2 \bmod 4, \quad i = \Pi(j) = Pj + Q_2 + i_0 \mod N \\
&\text{if } j = 3 \bmod 4, \quad i = \Pi(j) = Pj + Q_3 + i_0 \mod N
\end{aligned} \tag{8}$$

and N must be a multiple of four, which is not a very restricting condition, with respect to flexibility.

In order to ensure the bijection property of Π, the Q values are not just any values. A straightforward way to satisfy the bijection condition is to choose all Q's as multiples of C.

The regular permutation law expressed by (6) is appropriate for error patterns which are simple RTZ sequences for both encoders; that is, RTZ sequences which are not decomposable as a sum of shorter RTZ sequences. A particular and important case of a simple RTZ sequence is the two-symbol RTZ sequence, which may dominate in the asymptotic characteristics of a TC (see [15] for TCs with $m = 1$). A two-symbol sequence is a sequence with two nonzero m-bit input symbols, which may contain more than one nonzero bit. Let us define the total spatial distance (or total span) $S(j_1, j_2)$ as the sum of the two spatial distances, before and after permutation, according to (6), for a given pair of positions j_1 and j_2

$$S(j_1, j_2) = \mathrm{f}(j_1, j_2) + \mathrm{f}(\Pi(j_1), \Pi(j_2)) \tag{9}$$

where

$$\mathrm{f}(u, v) = \min\{|u - v|, N - |u - v|\}. \tag{10}$$

Finally, we denote by $S_{\min}$ the minimum value of $S(j_1, j_2)$, for all possible pairs j_1 and j_2

$$S_{\min} = \min_{j_1, j_2} \{S(j_1, j_2)\}. \tag{11}$$

It was demonstrated in [16] that the maximum possible value for $S_{\min}$, when using regular interleaving, is

$$S = (S_{\min})_{\max} = \sqrt{2N} = \sqrt{\frac{2k}{m}}. \tag{12}$$

If any two-symbol RTZ sequence for one component encoder is transformed by Π or Π^{-1} into another two-symbol RTZ sequence for the other encoder, the upper bound given by (12) is amply sufficient to guarantee a large weight for parity bits, and thus, a large minimum binary Hamming distance. This is the same for any number of symbols, on the condition that both RTZ sequences, before and after permutation, are simple RTZ sequences.

On the other hand, ARP aims at combating error patterns which are not simple RTZ sequences, but are combinations of simple RTZ sequences for both encoders. Instilling some controlled disorder, through the $Q(j)$ values in (7), tends to break most of the composite RTZ sequences. Meanwhile, because the value of cycle C is small, the good property of regular permutation for simple RTZ sequences is not lost, and a total span close to $\sqrt{2N}$ can be achieved. [14] describes a procedure to obtain appropriate values for P and for the set of Q parameters.

The algorithmic permutation model described by (7) is simple to implement, does not require any ROM, and the parameters can be changed on-the-fly for adaptive encoding and decoding. Moreover, as explained in [14], massive parallelism, allowing several processors to run at the same time without increasing the memory size, can be exploited.

In addition to the ARP principle and the advantages developed above, the rate-$m/(m+1)$ component code adds one more

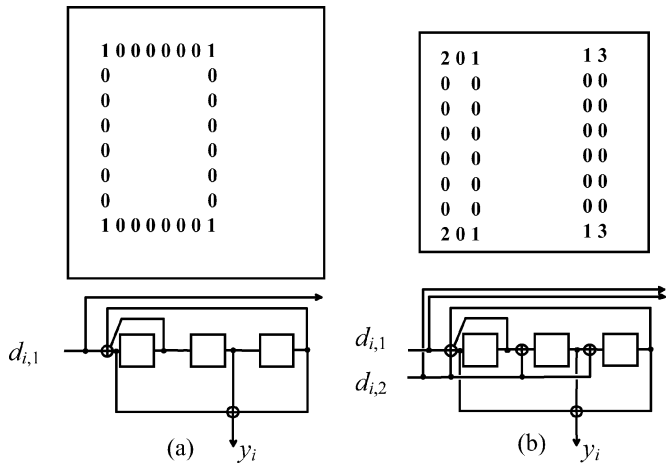

Fig. 2. Possible rectangular error patterns. For (a) binary and (b) double-binary TCs with regular permutations.

degree of freedom in the design of permutations: intrasymbol permutation, which enables some controlled disorder still to be added into the permutation without altering its global quasi-regularity. Intrasymbol permutation means modifying the contents of the m-bit symbols periodically, before the second encoding, in such a way that a large proportion of composite RTZ sequences for both codes can no longer subsist. Let us develop this idea in the simplest case of $m = 2$.

Fig. 2(a) depicts the minimal rectangular error pattern (input weight $w = 4$) for a parallel concatenation of two identical binary RSC encoders, involving a regular permutation (linewise writing, columnwise reading). This error pattern is a combination of two input weight-two RTZ sequences in each dimension, leading to a composite RTZ pattern with distance 16, for coding rate 1/2. If the component encoder is replaced by a double-binary encoder, as illustrated in Fig. 2(b), RTZ sequences and error patterns involve couples of bits, instead of binary values. Fig. 2(b) gives two examples of rectangular error patterns, corresponding to distance 18, still for coding rate 1/2 (i.e., no puncturing). Data couples are numbered from **0** to **3**, with the following notation: (0,0):**0**; (0,1):**1**; (1,0):**2**; (1,1):**3**. The periodicities of the double-binary RSC encoder, depicting all the combinations of pairs of input couples different from **0** that are RTZ sequences, are summarized in the diagram of Fig. 3. For instance, if the encoder, starting from state **0**, is fed up with successive couples **1** and **3**, it retrieves state **0**. The same behavior can be observed with sequences **201**, **2003**, **30002**, **3000001**, or **30000003**, for example.

The change from binary to double-binary code, though leading to a slight improvement in the distance (18 instead of 16), is not sufficient to ensure very good performance at low error rates. Let us suppose now that couples are inverted (**1** becomes **2** and vice versa) once every other time before second (vertical) encoding, as depicted in Fig. 4. In this way, the error patterns displayed in Fig. 2(b) no longer remain error patterns. For instance, **20000002** is still an RTZ sequence for the second (vertical) encoder, but **10000002** is no longer RTZ. Thus, many error patterns, especially short patterns, are eliminated, thanks to the disorder introduced inside input symbols. The right-hand side of Fig. 4 shows two examples of rectangle error patterns

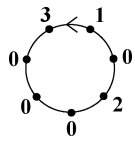

Fig. 3. Periodicities of the double-binary encoder of Fig. 2(b). Input couples (0,0), (0,1), (1,0), and (1,1) are denoted **0**, **1**, **2**, and **3**, respectively.

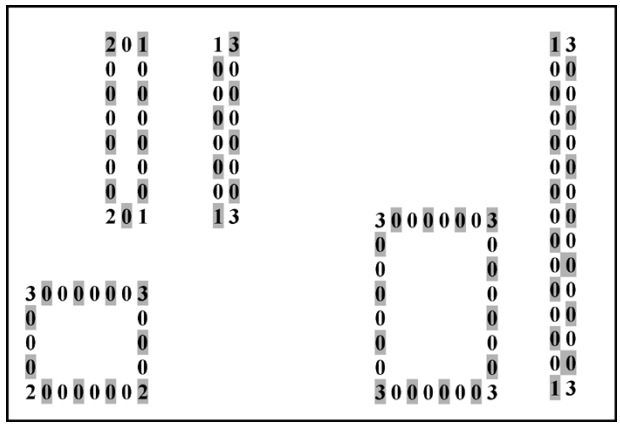

Fig. 4. Couples in gray spaces are inverted before second (vertical) encoding. **1** becomes **2**; **2** becomes **1**; **0** and **3** remain unaltered. The three patterns on the left-hand side are no longer error patterns. Those on the right-hand side remain possible error patterns, with distances 24 and 26 for coding rate 1/2.

that remain possible error patterns after the periodic inversion. The resulting minimal distances, 24 and 26, are large enough for the transmission of short data blocks [17]. For longer data blocks (a few thousand bits), combining this intrasymbol permutation with intersymbol ARP, as described above, can lead to even larger minimum distances, at least with respect to the rectangular error patterns with low input weights we gave as examples.

C. Advantages of TCs With Rate-$m/(m+1)$ RSC Constituent Codes

Parallel concatenation of m-input binary RSC codes offers several advantages in comparison with classical (one input) binary TCs, which have already been partly commented on in [18].

1) Better Convergence of the Iterative Process: This point was first observed in [19] and commented on in [20]. The better convergence of the bidimensional iterative process is explained by a lower error density in each code dimension, which leads to a decrease in the correlation effect between the component decoders.

Let us consider again (12), which gives the maximum total span achievable when using regular or quasi-regular permutation. For a given coding rate R, the number of parity bits involved all along the total span, and used by either one decoder or the other, is

$$n_{\text{parity}}(S) = \left(\frac{1-R}{R}\right)\frac{m}{2}S = \left(\frac{1-R}{R}\right)\sqrt{\frac{mk}{2}}. \qquad (13)$$

Thus, replacing a classical binary ($m = 1$) with a double-binary ($m = 2$) TC multiplies this number of parity bits by $\sqrt{2}$, though dividing the total span by the same value. Because the

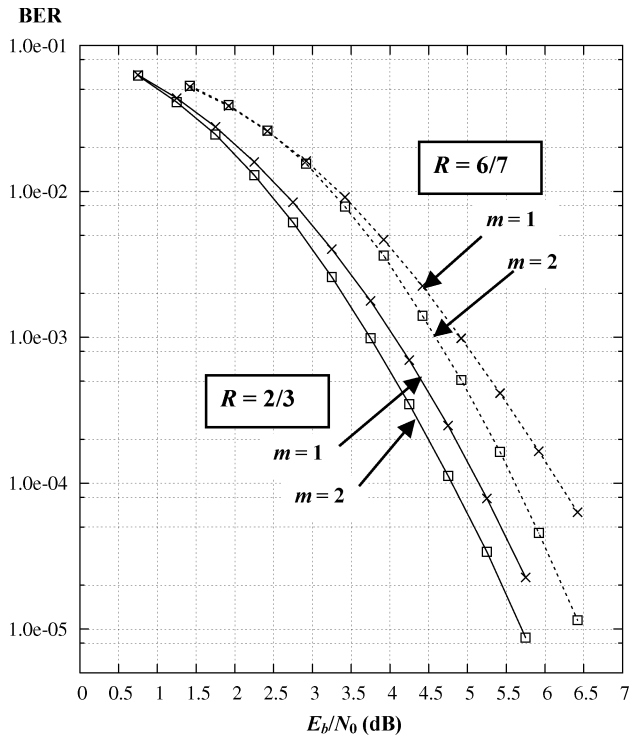

Fig. 5. Performance in BER of single 8-state RSC codes with $m = 1$ and $m = 2$. Encoder polynomials: 15 (feedback) and 13 (redundancy) in octal form (DVB-RCS constituent encoder for $m = 2$). Coding rates are 2/3 and 6/7 (regular puncturing). Binary/quaternary phase-shift keying (BPSK/QPSK) modulation, additive white Gaussian noise (AWGN) channel, and MAP decoding. No quantization.

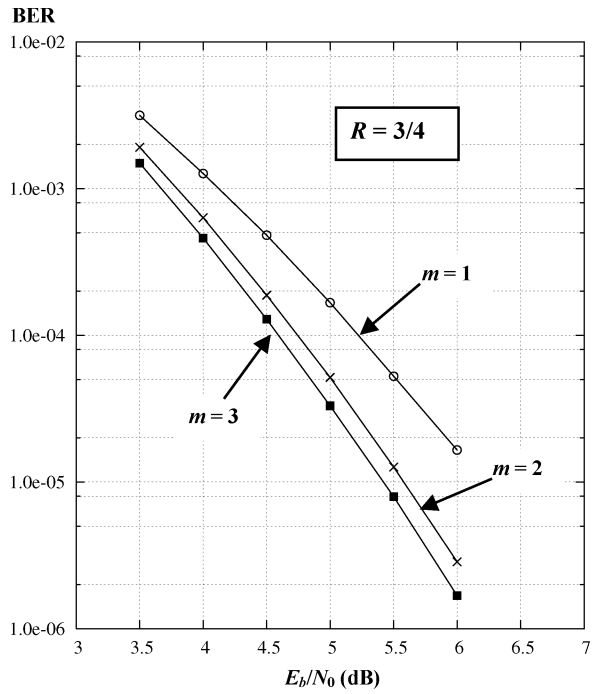

Fig. 6. Performance in BER of three single 16-state RSC codes with $m = 1$, $m = 2$, and $m = 3$. Encoder polynomials: 23 (feedback) and 35 (redundancy) in octal notation. Coding rate is 3/4, regular puncturing. BPSK/QPSK modulation, AWGN channel and MAP decoding. No quantization.

parity bits are not a matter of information exchange between the two decoders (they are just used locally), the more numerous they are, with respect to a given possible error pattern (here, the weight-two patterns), the fewer correlation effects between the component decoders.

Raising m beyond two still improves the turbo algorithm, regarding correlation, but the gains get smaller and smaller as m increases.

2) Larger Minimum Distances: As explained above, the number of parity bits involved in simple two-symbol RTZ sequences for both encoders is increased when using rate-$m/(m+1)$ component codes. The number of parity bits involved in any simple RTZ sequence, before and after permutation, is at least equal to $n_{\text{parity}}(S)$, regardless of the number of nonzero symbols in the sequence. The binary Hamming distances corresponding to all simple RTZ sequences are then high, and do not pose any problem with respect to the minimum Hamming distance of the TC. This comes from error patterns made up of several (typically, two or three) short simple RTZ sequences on both dimensions of the TC. Different techniques can be used to break most of these patterns, one of them (ARP) having been presented in Section III-B.

3) Less Puncturing for a Given Rate: In order to obtain coding rates higher than $m/(m+1)$, from the RSC encoder of Fig. 1, fewer redundant symbols have to be discarded, compared with an $m = 1$ binary encoder. Consequently, the correcting ability of the constituent code is less degraded. In order to illustrate this assertion, Fig. 5 compares the performance, in terms of bit-error rate (BER), of two 8-state RSC codes with the same generator polynomials (15, 13) in octal notation, for

$$m = 1 \quad (\mathbf{C} = [1 \quad 0 \quad 0]^T)$$
$$m = 2 \quad \left(\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}^T\right).$$

The two-input RSC code displays better performance than the one-input code, for both simulated coding rates.

Raising m to three and beyond for this code is not of interest, since there is no full-rank three-column matrix $\mathbf{C}$ that satisfies (5). The choice of a constituent code with parallel transitions in the trellis would lead to a TC with very low minimum Hamming distance.

In Fig. 6, an $m = 3$ curve has been introduced for the (23,35) 16-state RSC code with coding rate 3/4. The connection matrices are equal to

$$\mathbf{C} = [1 \quad 0 \quad 0 \quad 0]^T \text{ for } m = 1$$
$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}^T \text{ for } m = 2$$
$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}^T \text{ for } m = 3.$$

As expected, we observe that the performance gain between $m = 2$ and $m = 3$ is smaller than the gain between $m = 1$ and $m = 2$. Raising m to four and beyond for this 16-state code is not of interest, since there is no full-rank four-column matrix $\mathbf{C}$ that satisfies (5).

4) Higher Throughput and Reduced Latency: The decoder of an $m/(m+1)$ convolutional code provides m bits at each decoding step. Thus, once the data block is received, and for a

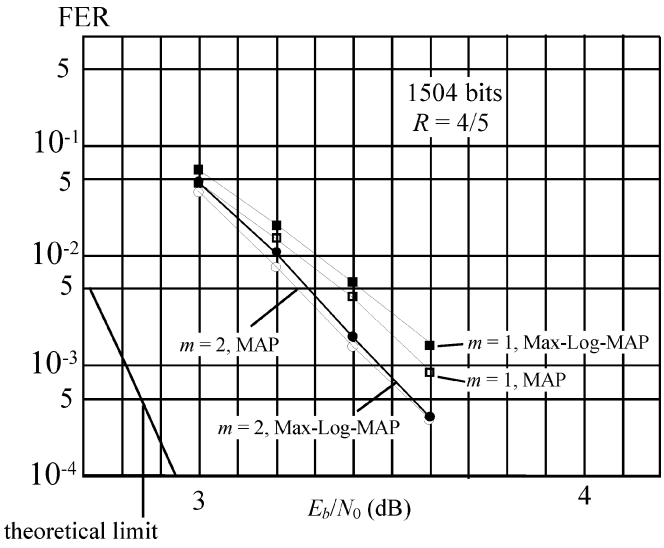

Fig. 7. Comparison of performance in FER of two TCs based on both 8-state RSC codes of Fig. 5, with $k = 1504$, $R = 4/5$, for MAP and Max-Log-MAP decoding. AWGN channel, QPSK modulation, eight iterations. Scaling factor for Max-Log-MAP decoding: 0.7 for iterations 1–7, 1.0 for iteration 8. No quantization.

given processing clock, the decoding throughput of a hardware decoder is multiplied by m. The latency, i.e., the number of clock periods required to decode a data block, is then divided by m, compared with the classical case ($m = 1$).

However, the critical path of the decoder being the Add–Compare–Select (ACS) unit, the decoder with $m > 1$ has a lower maximum clock frequency than with $m = 1$. For instance, for $m = 2$, the Compare–Select operation has to be done on four metrics instead of two, thus with an increased propagation delay. The use of specialized look-ahead operators and/or the introduction of parallelism, in particular, the multistreaming method [21], makes it possible to significantly increase the maximum frequency of the decoder, and even to reach that of the decoder with $m = 1$.

5) Robustness of the Decoder: Fig. 7 represents the simulated performance in FER, as a function of E_b/N_0, of four coding/decoding schemes dealing with blocks of 1504 information bits and coding rate 4/5: binary and double-binary 8-state TCs exhibited in Fig. 2, both with the full MAP decoding algorithm [22], [23], and with the simplified Max-Log-MAP version [24]. In the latter case, the extrinsic information is less reliable, especially at the beginning of the iterative process. To compensate for this, a scaling factor, lower than 1.0, is applied to extrinsic information [25]. The best observed performance was obtained when a scaling coefficient of 0.7 for all the iterations, except for the last one, was applied.

In Fig. 7, both codes have ARP internal permutation with optimized span. We can observe that the double-binary TC performs better, at both low and high E_b/N_0, and the steeper slope for the double-binary TC indicates a larger minimum binary Hamming distance. These characteristics were justified by 1) and 2) of this section.

What is also noteworthy is the very slight difference between the decoding performance of the double-binary TC when using the MAP or the Max-Log-MAP algorithms. This property of nonbinary turbo decoding actually makes the full MAP decoder unnecessary (or the Max-Log-MAP decoder with Jacobian logarithm correction [24]), which requires more operations than the Max-Log-MAP decoder. Also, the latter does not need the knowledge of the noise variance on Gaussian channels, which is a nice advantage. The rigorous explanation for this quasi-equivalence of the MAP and the Max-Log-MAP algorithms, when decoding m-input TCs, has still to be found.

IV. Performance of Double-Binary TCs

This section describes two examples of double-binary TCs, with memory 3 and 4, whose reasonable decoding complexity allows them to be implemented in actual hardware devices for practical applications. Simulation results for transmissions over an AWGN channel with QPSK modulation are provided.

A. Eight-State Double-Binary TC

The parameters of the component codes are

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \quad \mathbf{R}_1 = [1 \quad 1 \quad 0]$$
$$\mathbf{R}_2 = [1 \quad 0 \quad 0]. \tag{14}$$

The diagram of the encoder is described in Fig. 8. Redundancy vector $\mathbf{R}_2$ is only used for coding rates less than 1/2. For coding rates higher than 1/2, puncturing is performed on redundancy bits in a regular periodical way, following patterns that are described in [5]. These patterns are identical for both constituent encoders.

The permutation function $i = \Pi(j)$ is performed on two levels, as explained in Section III-B.

For $j = 0, \ldots, N-1$, we have the following.

- Level 1: inversion of $d_{j,1}$ and $d_{j,2}$ in the data couple, if $j \bmod 2 = 0$.
- Level 2: this permutation level is described by a particular form of (8)

$$\begin{aligned} i &= (P \times j + Q(j) + 1) \bmod N, \text{ with} \\ Q(j) &= 0 && \text{if } j \bmod 4 = 0 \\ Q(j) &= \frac{N}{2} + P_1 && \text{if } j \bmod 4 = 1 \\ Q(j) &= P_2 && \text{if } j \bmod 4 = 2 \\ Q(j) &= \frac{N}{2} + P_3 && \text{if } j \bmod 4 = 3. \end{aligned} \tag{15}$$

Value $i_0 = 1$ is added to the incremental relation in order to comply with the odd–even rule [26]. The disorder is instilled in the permutation function, according to the ARP principle, in two ways.

- A shift by $N/2$ is added for odd values of j. This is done because the lowest subperiod of the code generator is one (see Fig. 3). The role of this additional increment is thus to spread to the full the possible errors associated with the shortest error patterns.
- P_1, P_2, and P_3 act as local additional pseudorandom fluctuations.

Notice that the permutation equations and parameters do not depend on the coding rate considered. The parameters can be

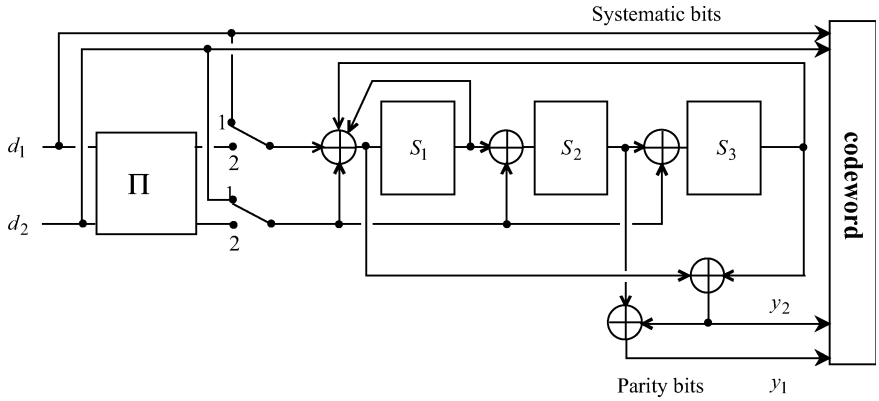

Fig. 8. Structure of the 8-state encoder. Redundancy y_2 is only used for turbo-coding rates less than 1/2.

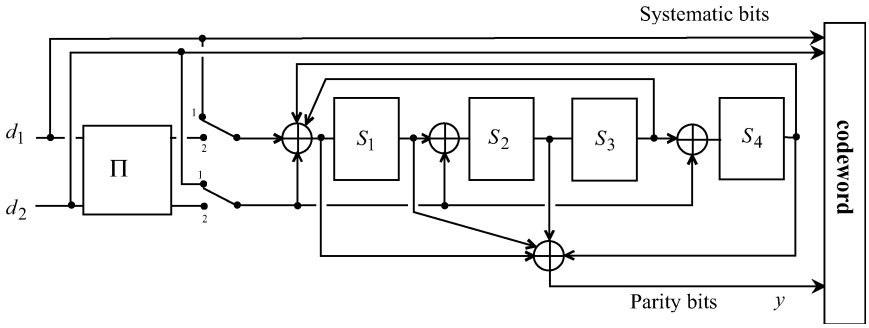

Fig. 9. Structure of the proposed 16-state double-binary turbo encoder.

optimized to provide good behavior, on average, at low error rates for all coding rates, but seeking parameters for a particular coding rate could lead to slightly better performance.

B. Sixteen-State Double-Binary TC

The parameters of the best component code we have found are

$$\mathbf{G}=\begin{bmatrix}0&0&1&1\\1&0&0&0\\0&1&0&0\\0&0&1&0\end{bmatrix}\quad \mathbf{C}=\begin{bmatrix}1&1\\0&1\\0&0\\0&1\end{bmatrix}\quad \mathbf{R}=[1\quad 1\quad 1\quad 0]. \tag{16}$$

The diagram of the encoder is described in Fig. 9. Puncturing is performed on redundancy in a periodic way, with identical patterns for both constituent encoders. It is usually regular, except when the puncturing period is a divisor of the LFSR period. For example, for coding rate 3/4, the puncturing period is chosen equal to six, with puncturing pattern [101000].

For this code, the permutation parameters have been carefully chosen, following the procedure described in [14], in order to guarantee a large minimum Hamming distance, even for high rates. The level-1 permutation is identical to the intrapermutation of the 8-state code. The level-2 intersymbol permutation is given by

$$\begin{aligned}&\text{For } j=0,\ldots,N-1\\&\quad i=(P\times j+Q(j)+3) \bmod N, \text{ with}\\&Q(j)=0 \qquad \text{if } j \bmod 4=0\\&Q(j)=Q_1 \qquad \text{if } j \bmod 4=1\\&Q(j)=4Q_0+Q_2 \quad \text{if } j \bmod 4=2\\&Q(j)=4Q_0+Q_3 \quad \text{if } j \bmod 4=3.\end{aligned} \tag{17}$$

TABLE I
ESTIMATED VALUES OF MINIMUM BINARY HAMMING DISTANCES $d_{\min}$ OF PROPOSED 8-STATE AND 16-STATE DOUBLE-BINARY TCS FOR 188-B DATA BLOCKS. DISTANCES WERE ESTIMATED WITH THE ALL-ZERO ITERATIVE DECODING ALGORITHM [27]

Coding rate	1/2	2/3	3/4
8-state double-binary turbo code ($P=19, P_1=376, P_2=224, P_3=600$)	19	12	9
16-state double-binary turbo code ($P=35, Q_0=1, Q_1=4, Q_2=4, Q_3=12$)	26	18	12

The spirit in which this permutation was designed is the same as that already explained for the 8-state TC. The only difference is that the lowest subperiod of the 16-state generator is two, instead of one. That is why the additional shift (by $4Q_0$) is applied consecutively, twice every four values of j.

Table I compares the minimum binary Hamming distances of the proposed 8-state and 16-state TCs, for 188-B data blocks and four different coding rates. The distance values were estimated with the so-called all-zero iterative algorithm, a fast computational method described in [27], which provides distance values with very high reliability for block sizes larger than a few hundred bits. We can observe a significant increase in the minimum distance when using 16-state component codes; the gain varies from 30%–50% depending on the case considered. With this code, we were also able to define permutation parameters leading to minimum distances as large as 33 for $R=1/2$, 22 for $R=2/3$, and 16 for $R=3/4$ for (10×188)-B blocks.

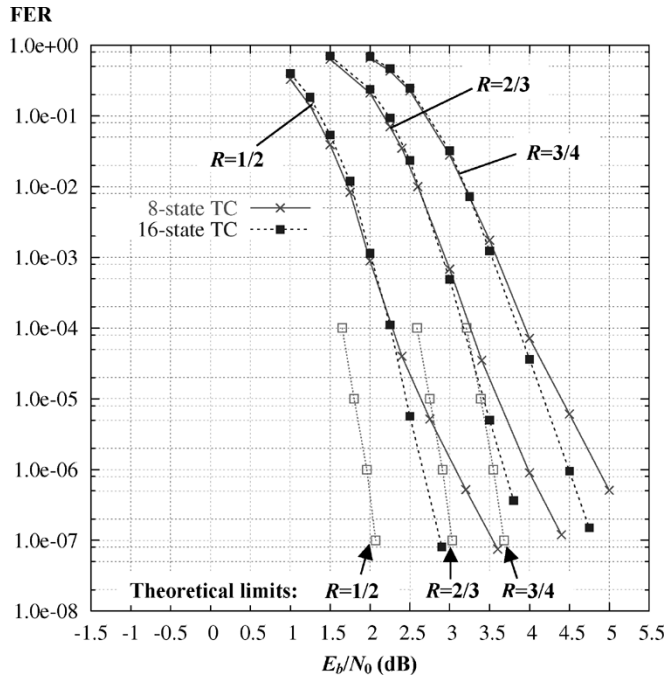

Fig. 10. Performance in FER of 8-state and 16-state double-binary TCs for ATM (53 B) blocks and rates 1/2, 2/3, and 3/4. QPSK modulation and AWGN channel. Max-Log-MAP decoding with 4-b input samples and 8 iterations. The theoretical limits on FER are derived from [28].

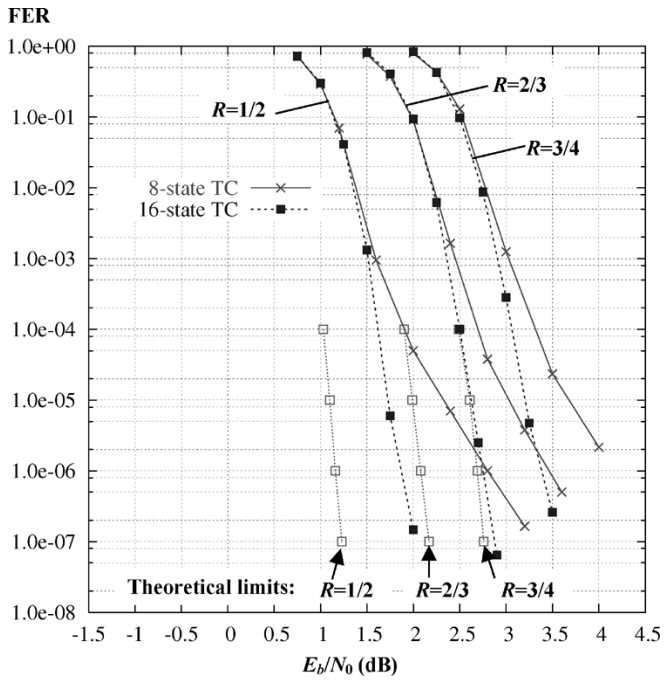

Fig. 11. Performance in FER of 8-state and 16-state double-binary TCs for MPEG (188 B) blocks and rates 1/2, 2/3, and 3/4. QPSK modulation and AWGN channel. Max-Log-MAP decoding with 4-b input samples and 8 iterations. The theoretical limits on FER are derived from [28].

From an implemention point of view, the complexity of the corresponding decoder is about twice the complexity of the 8-state decoder.

C. Simulation Results

We have simulated and compared these two codes for two block sizes and three coding rates for transmissions over an AWGN channel with QPSK modulation. The simulation results take actual implementation constraints into account. In particular, the decoder inputs are quantized for hardware complexity considerations. According to our experience, the performance degradation due to input quantization is not significant beyond 5 b. The observed loss is less than 0.15 dB for 4-b quantization, and about 0.4 dB for 3-b quantization. When quantization is applied, clipping extrinsic information at a threshold around twice the maximum range of the input samples does not degrade the performance, while limiting the amount of required memory.

Solid line curves in Figs. 10 and 11 show the FER as a function of E_b/N_0 for the transmission of ATM (53 B) and MPEG (188 B) packets for three different values of coding rate of the 8-state double-binary TC. The component decoders use the Max-Log-MAP algorithm, with input samples quantized on 4 b. Eight iterations were simulated and at least 100 erroneous frames were considered for each point indicated, except for the lowest points, where approximately 30 erroneous frames were simulated.

We can observe good average performance for this code whose decoding complexity is very reasonable. For a hardware implementation, less than 20 000 logical gates are necessary to implement one iteration of the decoding process when decoding is performed at the system clock frequency, plus the memory required for extrinsic information and input data. Its performance improves predictably with block size and coding rate in relation to the theoretical limit. The reported limits given in Figs. 10 and 11, as well as in Fig. 7, take the block size and the target FER into account. They are derived from the Gallager random coding bound on the error probability for binary-input channels, as described in [28]. At FER $= 10^{-4}$, the simulated curves lie within 0.6–0.8 dB from the limit, regardless of block size and coding rate. To improve the performance of this code family at FER below 10^{-5}, the more powerful 16-state component code has to be selected so as to increase the overall minimum Hamming distance of the composite code.

Dotted-line curves in Figs. 10 and 11 show the 16-state TC performance for the same simulation conditions as for the 8-state code. Similar to this code, the permutation parameters are related to the block size, not to the coding rate.

We can observe that the selected code does not lead to a convergence-threshold shift of the iterative decoding process in comparison with the previous 8-state code. For FERs above 10^{-4}, 16-state and 8-state codes behave similarly. For lower error rates, thanks to the increase in distance, there is no noticeable floor effect for the simulated signal-to-noise ratio (SNR) ranges, that is, down to a FER of 10^{-7}. Again, performance improves predictably with block size and coding rate in relation to the theoretical limit. At FER $= 10^{-6}$, the simulated curves lie within 0.7–1.0 dB from the limits, regardless of block size and coding rate, even with the simplified Max-Log-MAP algorithm.

V. Conclusion

Searching for perfect channel coding presents two challenges: encoding in such a way that large minimum distances can be reached; and achieving decoding as close to the theoretical limit as possible. In this paper, we have explained why

m-input binary TCs combined with a two-level permutation can represent a better answer to these challenges than classical one-input binary TCs.

In practice, with $m = 2$, we have been able to design coding schemes with moderate decoding complexity, and whose performance approaches the theoretical limit by less than 1 dB at FER $= 10^{-6}$. The 8-state TC with $m = 2$ has already found practical applications through several international standards.

Furthermore, the parallel concatenation of RSC circular codes leads to flexible composite codes, easily adaptable to a large range of data block sizes and coding rates. Consequently, as m-input binary codes are well suited for association with high-order modulations, in particular M-ary quadrature amplitude modulation and M-ary PSK, TCs based on these constituent codes appear to be good candidates for most future digital communication systems based on block transmission.

ACKNOWLEDGMENT

The authors are grateful to the Associate Editor and the reviewers for their valuable comments and suggestions for improving this paper.

REFERENCES

[1] C. Berrou and A. Glavieux, "Near-optimum error-correcting coding and decoding: Turbo codes," *IEEE Trans. Commun.*, vol. 44, no. 10, pp. 1261–1271, Oct. 1996.

[2] P. Robertson and T. Wörz, "Bandwidth-efficient turbo trellis-coded modulation using punctured component codes," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 2, pp. 206–218, Feb. 1998.

[3] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Parallel concatenated trellis-coded modulation," in *Proc. IEEE Int. Conf. Commun.*, vol. 2, Dallas, TX, Jun. 1996, pp. 974–978.

[4] C. Fragouli and R. D. Wesel, "Turbo-encoder design for symbol-interleaved parallel concatenated trellis-coded modulation," *IEEE Trans. Commun.*, vol. 49, no. 3, pp. 425–435, Mar. 2001.

[5] "Interaction Channel for Satellite Distribution Systems," DVB, ETSI EN 301 790, vol. 1.2.2, 2000.

[6] "Interaction Channel for Digital Terrestrial Television," DVB, ETSI EN 301 958, vol. 1.1.1, 2002.

[7] *IEEE Standard for Local and Metropolitan Area Networks*, IEEE 802.16a, 2003.

[8] S. Riedel, "Symbol-by-symbol MAP decoding algorithm for high-rate convolutional codes that use reciprocal dual codes," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 2, pp. 175–185, Feb. 1998.

[9] C. Weiss, C. Bettstetter, S. Riedel, and D. J. Costello, "Turbo decoding with tailbiting trellises," in *Proc. IEEE Int. Symp. Signals, Syst., Electron.*, Pisa, Italy, Oct. 1998, pp. 343–348.

[10] R. Johannesson and K. S. Zigangirov, *Fundamentals of Convolutional Coding*, ser. Digital, Mobile Commun.. New York: IEEE Press, 1999, ch. 4.

[11] S. Dolinar and D. Divsalar, "Weight distribution of turbo codes using random and nonrandom permutations," JPL, NASA, TDA Prog. Rep. 42–122, 1995.

[12] C. Heegard and S. B. Wicker, *Turbo Coding*. Norwell, MA: Kluwer, 1999, ch. 3.

[13] S. Crozier, J. Lodge, P. Guinand, and A. Hunt, "Performance of turbo codes with relatively prime and golden interleaving strategies," in *Proc. 6th Int. Mobile Satellite Conf.*, Ottawa, ON, Canada, Jun. 1999, pp. 268–275.

[14] C. Berrou, Y. Saouter, C. Douillard, S. Kerouédan, and M. Jézéquel, "Designing good permutations for turbo codes: Toward a single model," in *Proc. IEEE Int. Conf. Commun.*, Paris, France, Jun. 2004, pp. 341–345.

[15] S. Benedetto and G. Montorsi, "Design of parallel concatenated convolutional codes," *IEEE Trans. Commun.*, vol. 44, no. 5, pp. 591–600, May 1996.

[16] E. Boutillon and D. Gnaedig, "Maximum spread of D-dimensional multiple turbo codes," *IEEE Trans. Commun.*, vol. 53, no. 8, pp. 1237–1242, Aug. 2005.

[17] C. Berrou, E. A. Maury, and H. Gonzalez, "Which minimum Hamming distance do we really need?," in *Proc. 3rd Symp. Turbo Codes*, Brest, France, Sep. 2003, pp. 141–148.

[18] C. Berrou, M. Jézéquel, C. Douillard, and S. Kerouédan, "The advantages of nonbinary turbo codes," in *Proc. Inf. Theory Workshop*, Cairns, Australia, Sep. 2001, pp. 61–63.

[19] C. Berrou, "Some clinical aspects of turbo codes," in *Proc. Int. Symp. Turbo Codes Related Top.*, Brest, France, Sep. 1997, pp. 26–31.

[20] C. Berrou and M. Jézéquel, "Nonbinary convolutional codes for turbo coding," *Electron. Lett.*, vol. 35, no. 1, pp. 39–40, Jan. 1999.

[21] H. Lin and D. G. Messerschmitt, "Algorithms and architectures for concurrent Viterbi decoding," in *Proc. IEEE Int. Conf. Commun.*, Boston, MA, Jun. 1989, pp. 836–840.

[22] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol-error rate," *IEEE Trans. Inf. Theory*, vol. IT-20, no. 3, pp. 284–287, Mar. 1974.

[23] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "A soft-input soft-output APP module for iterative decoding of concatenated codes," *IEEE Commun. Lett.*, vol. 1, pp. 22–24, Jan. 1997.

[24] P. Robertson, P. Hoeher, and E. Villebrun, "Optimal and suboptimal maximum *a posteriori* algorithms suitable for turbo decoding," *Eur. Trans. Telecommun.*, vol. 8, pp. 119–125, Mar./Apr. 1997.

[25] J. Vogt and A. Finger, "Improving the max-log-MAP turbo decoder," *Electron. Lett.*, vol. 36, no. 23, pp. 1937–1939, Nov. 2000.

[26] A. S. Barbulescu, "Iterative decoding of turbo codes and other concatenated codes," Ph.D. dissertation, Univ. South Australia, 1996.

[27] R. Garello and A. V. Casado, "The all-zero iterative decoding algorithm for turbo code minimum distance computation," in *Proc. Int. Conf. Commun.*, Paris, France, Jun. 2004, pp. 361–364.

[28] R. G. Gallager, *Information Theory and Reliable Communication*. New York: Wiley, 1968, sec. 5.6.

Catherine Douillard (M'01) received the engineering degree in telecommunications from the École Nationale Supérieure des Télécommunications (ENST) de Bretagne, Brest, France, in 1988, and the Ph.D. degree in electrical engineering from the Université de Bretagne Occidentale, Brest, France, in 1992.

In 1991 she joined ENST Bretagne, where she is currently a Professor in the Electronics Department. Her main interests are turbo codes and iterative decoding, iterative detection, and the efficient combination of high spectral efficiency modulation and turbo-coding schemes.

Claude Berrou (M'87) was born in Penmarc'h, France, in 1951. In 1978, he joined the Ecole Nationale Supérieure des Télécommunications (ENST) de Bretagne, where he is currently a Professor in the Electronics Department. In the early 80's, he started up the training and research activities in VLSI technology and design, to meet the growing demand from industry for microelectronics engineers. Some years later, he took an active interest in the field of algorithm/silicon interaction for digital communications. In collaboration with Prof. A. Glavieux, he introduced the concept of probabilistic feedback into error-correcting decoders, and developed a new family of quasi-optimal error-correction codes that he named "turbo codes." He also pioneered the extension of the turbo principle to joint detection and decoding processing, known today as turbo detection and turbo equalization. His current research topics, besides algorithm/silicon interaction, are electronics and digital communications at large, error-correction codes, turbo codes and iterative processing, soft-in/soft-out (probabilistic) decoders, etc. He is the author or co-author of 8 registered patents and about 60 publications in the field of digital communications and electronics.

Dr. Berrou has received several distinctions (with Prof. Glavieux), amongst which are the 1997 SEE Médaille Ampère, one of the 1998 IEEE (Information Theory) Golden Jubilee Awards for Technological Innovation, the 2003 IEEE Richard W. Hamminag medal, the 2003 French Grand Prix France Télécom de l'Académie des sciences, and the 2005 Marconi Prize.