Transactions Papers

Turbo-Encoder Design for Symbol-Interleaved Parallel Concatenated Trellis-Coded Modulation

Christina Fragouli, *Member, IEEE*, and Richard D. Wesel, *Senior Member, IEEE*

***Abstract*—This paper addresses turbo-encoder design for coding with high spectral efficiency using parallel concatenated trellis-coded modulation and symbol interleaving. The turbo-encoder design involves the constituent encoder design and the interleaver design. The constituent encoders are optimized for symbol-wise effective free distance, and each has an infinite symbol-wise impulse response. We identify the canonical structures for the constituent encoder search space. In many cases of practical interest, the optimal structure for these constituent encoders connects the memory elements in a single row. This single row generally applies to turbo-code constituent encoders for parallel concatenation and is not restricted to symbol interleaving. To lower the error floor, a new semi-random interleaver design criteria and a construction method extends the spread-interleaver concept introduced by Divsalar and Pollara. Simulation results show that the proposed system employing symbol interleaving can converge at a lower signal-to-noise ratio than previously reported systems. We report simulation results between 0.5 and 0.6 dB from constrained capacity for rates of 2 and 4 bits/s/Hz.**

***Index Terms*—Concatenated coding, convolutional codes, interleaved coding, trellis-coded modulation, turbo codes.**

I. Introduction

THIS paper presents a method for parallel concatenated trellis-coded modulation (PCTCM) with constituent encoders of rate k/n, $k > 1$. The k binary inputs are one symbol over the extension field $GF(2^k)$. Two main approaches are proposed in the literature for the turbo-encoder structure, one employing bit interleaving by Benedetto *et al.* [1] and the other employing symbol interleaving by Robertson and Wörz [2], [3].

For bit interleaving, k bit interleavers are used to keep the bit streams separate. The first constituent encoder in [1], for k even, has half of the k input bits as systematic outputs and a parity output. The second constituent encoder is the same as the first, but the other half of the k input bits become systematic. Thus, the overall turbo encoder is systematic.

For symbol interleaving as described in [2] and [3] to have the overall turbo encoder systematic, the interleaver maps even symbol positions to even symbol positions and odd ones to odd. The output of the second encoder is deinterleaved and the output symbols from each encoder are punctured alternatively. The odd-to-odd and even-to-even interleaving was first described by Barbulescu and Pietrobon in [4], and is equivalent to using two separate symbol interleavers of half the length, one for the odd positions and another for the even ones. This additional structure of the symbol interleaver reduces the interleaving gain, as is also observed by Ogiwara and Yano in [5]. Moreover, puncturing complicates the design of the constituent encoders.

Our proposed approach combines the turbo-encoder approach of [1] with a symbol interleaver. Each k/n constituent encoder, for k even, has $k/2$ systematic outputs and $r \geq 1$ parity outputs. The $n = (k/2) + r$ total output bits of the encoder are mapped to one constellation point. The upper constituent encoder has as systematic outputs the $k/2$ most significant (MSB) input bits while the lower constituent encoder has as systematic outputs the $k/2$ least significant (LSB) input bits. Thus, the systematic bits are evenly divided between the constituent encoders without puncturing or interleaver constraints as in [2] (in [5] the interleaver constraints are removed but puncturing is still employed). Fig. 1 shows an example of the proposed parallel turbo-code structure that employs 16QAM modulation in connection with rate $4/4$ constituent encoders, each with $k/2 = 2$ systematic and $r = 2$ parity outputs. Fig. 2 shows another example that employs 8PSK (phase-shift keying) modulation in connection with rate $4/3$ constituent encoders, each with $k/2 = 2$ systematic and $r = 1$ parity outputs. The generalization to k/n encoders using 2^n-point constellations is straightforward when k is even.

Generally, using a symbol interleaver is equivalent to using k bit interleavers that implement the same interleaving pattern. In contrast, interleaving the k bits separately allows spreading of the components of one error event to k times more error events, typically accumulating more distance. Thus, a symbol

Paper approved by S. S. Pietrobon, the Editor for Coding Theory and Techniques of the IEEE Communications Society. Manuscript received June 23, 1999; revised May 15, 2000 and August 7, 2000. This work was supported by a National Science Foundation CAREER Award 9733089, Xetron Corporation, and Conexant under California MICRO Grant 99-123. This paper was presented in part at the International Conference on Communications (ICC), Vancouver, BC, Canada, June 1999, and the Communications and Control (ComCon), Athens, Greece, June 1999, and GLOBECOM, Rio de Janeiro, Brazil, December 1999.

C. Fragouli was with the Department of Electrical Engineering, University of California, Los Angeles, CA 90095-1595 USA. She is now with AT&T Research Laboratories, Florham Park, NJ 07932 USA (e-mail: fragouli@research.att.com).

R. D. Wesel is with the Department of Electrical Engineering, University of California, Los Angeles, CA 90095-1595 USA (e-mail: wesel@ee.ucla.edu).

Publisher Item Identifier S 0090-6778(01)02171-7.

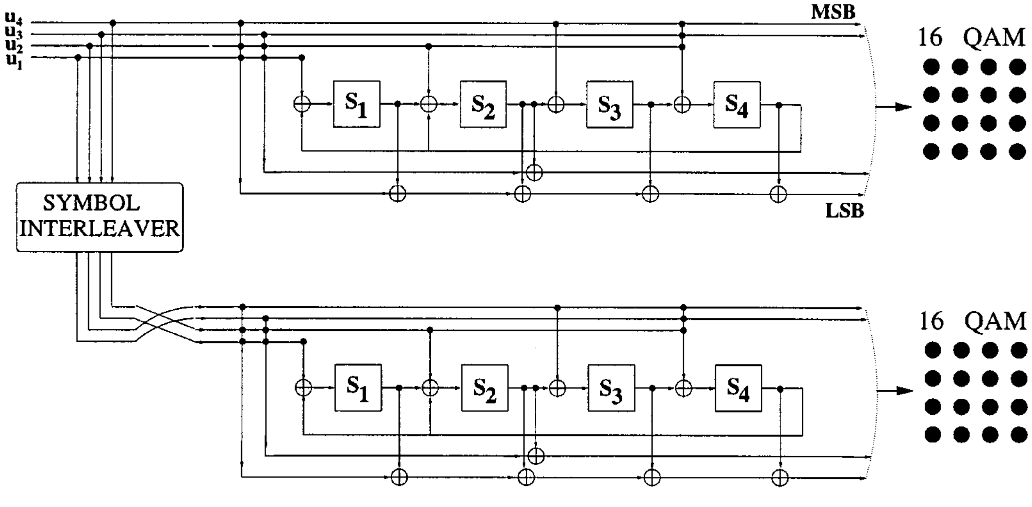


Fig. 1. Two-bits/s/Hz PCTCM turbo code with rate 4/4 constituent encoders.

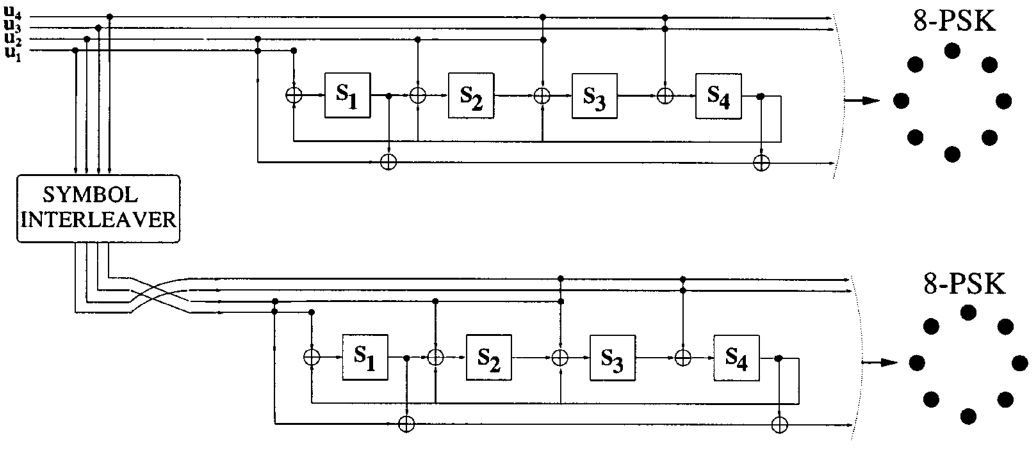


Fig. 2. Two-bits/s/Hz PCTCM turbo code with rate 4/3 constituent encoders.

interleaver imposes a structure that reduces the interleaver gain of a turbo encoder. Despite the loss in interleaver gain, we are motivated to use symbol interleaving because it imposes fewer assumptions on iterative decoding, as discussed below.

Our iterative decoder implements the soft-input soft-output (SISO) equations appearing in [6], with input bit probabilities substituted by input symbol probabilities. Let $\mathbf{Y_1}$ be the observed sequence at the SISO module corresponding to the upper constituent encoder, and $\mathbf{u} = \{\mathbf{u_t}\}$ the input symbols sequence we try to estimate. The iterative turbo decoder uses the assumption that the exchanged input symbol probabilities are independent. This is not true because they are conditioned on the observed output sequence $\mathbf{Y_1}$

$$P(\mathbf{u}|\mathbf{Y_1}) \neq \Pi_t P(\mathbf{u_t}|\mathbf{Y_1}). \tag{1}$$

Using bit interleaving leads to the additional assumption that the bits $\{u_{t,i}\}$ within each symbol $\mathbf{u_t}$ are also independent. Again, this is not true

$$P(\mathbf{u_t}|\mathbf{Y_1}) \neq \Pi_i P(u_{t,i}|\mathbf{Y_1}). \tag{2}$$

Symbol interleaving avoids this additional assumption of independence [the assumption of equality in (2)].

This paper addresses the design of our proposed system for high spectral efficiency, and investigates what benefits it offers as compared to the previously proposed approaches in the literature. The turbo-encoder design consists of two components, the constituent encoder design and the interleaver design, which are examined in Sections II and III, respectively. More specifically, Section II derives the optimization criteria for the PCTCM constituent encoders and extends the effective distance bounds to symbol-wise inputs. The appropriate encoder structure for turbo-code constituent encoders is identified, and applied to the special case of PCTCM. Tables of codes optimized for effective free distance are provided. Section III addresses the interleaver design. We propose new semi-random interleaver design criteria and a construction method that is an extension of the spread interleaver concept introduced by Divsalar and Pollara. The interleaver design applies to both bit and symbol interleaving. Section IV presents simulation results, and finally Section V concludes the paper.

II. Constituent Encoder Design

The use of a symbol interleaver implies that the constituent encoders should be optimized for "symbol-wise effective free distance." This term refers to the minimum output distance when the input symbol sequence has exactly two symbols different from zero. The usual notion of effective free distance refers to the minimum output distance for a binary input Hamming distance of two.

In the rest of this paper, we use several variations of effective free distance. The superscript refers to the output distance, Hamming (H) or Euclidean (E). The number in the subscript denotes the input weight, whether bit-wise (b) or symbol-wise (s). We always imply squared Euclidean distance. For example, d_{s2}^{E} stands for the output squared Euclidean distance when the symbol-wise input weight is two.

A. Desired Distance Properties

An analytical upper bound to the bit-error probability of turbo codes by Benedetto and Montorsi in [7] identified effective free distance as a key parameter. A similar analysis still holds when the input of the constituent encoders is over $GF(2^k)$, with the slight modification that the input Hamming weight now refers to Hamming weight in the extension Galois field $GF(2^k)$. Repeating the analysis for symbol-wise input along the lines of [7] (we do not repeat the exact derivation here), two main guidelines for the design of constituent encoders are derived as follows.

- For a given symbol-interleaver length, to achieve interleaver gain, the constituent convolutional encoders must have infinite output weight when the input symbol sequence contains only one symbol different than zero ($d_{s1}^{H} = \infty$).
- Among the encoders with $d_{s1}^{H} = \infty$, the ones with the best symbol effective distance (Hamming d_{s2}^{H} or Euclidean d_{s2}^{E} depending on the application) optimize the asymptotic turbo-code performance.

The first guideline equivalently states that there should be no parallel transitions in the trellis diagram, which was also presented in [2].

B. Distance Upper Bounds

Consider convolutional codes with k binary inputs, m memory elements, and r parity (not systematic) outputs. Assume that $d_{b1}^{H} = \infty$, i.e., the impulse response corresponding to every one of the k binary inputs is infinite. Divsalar *et al.* presented in [8] the following bound on the effective free distance d_{b2}^{H} that is a key design metric for constituent encoders used with bit interleaving:

$$d_{b2}^{H} \le \min\left(\left\lceil \frac{2^m}{k} \right\rceil r,\, 2r + \left\lfloor \frac{2^{m-1} r}{k} \right\rfloor\right) \tag{3}$$

where $\lfloor x \rfloor$ denotes the largest integer smaller than x, and $\lceil x \rceil$ denotes the smallest integer larger than x.

For symbol-interleaved PCTCM, it is interesting to examine the d_{s2}^{H} bound. An upper bound to d_{s2}^{H}, when $d_{s1}^{H} = \infty$, with r parity (not systematic) outputs and k binary inputs, is given by substituting k with $2^k - 1$ in (3)

$$d_{s2}^{H} \le \min\left(\left\lceil \frac{2^m}{2^k - 1} \right\rceil r,\, 2r + \left\lfloor \frac{2^{m-1} r}{2^k - 1} \right\rfloor\right). \tag{4}$$

The proof follows along the same lines as the proof of (3) given in [9]. The main point is as follows. If the feedback polynomial of a convolutional encoder is primitive, then the state diagram has one loop with zero inputs and nonzero outputs. This loop includes all the $2^m - 1$ nonzero states. An input sequence with two nonzero symbols causes the encoder to enter the loop (with the first nonzero input) and exit it (with the second nonzero input). The output weight of any output parity bit going around the whole loop is 2^{m-1}. If k binary inputs exist, there are k ways to enter and leave the loop via single input bits, and thus the minimum output weight of a single parity output, along the part of the loop that it travels before it exits, can be in the best case, $\lfloor 2^{m-1}/k \rfloor$. Considering symbol inputs, there are instead $2^k - 1$ ways to join/exit this loop, and thus the minimum output weight of a single parity bit can be, in the best case, $\lfloor 2^{m-1}/(2^k - 1) \rfloor$. This reasoning leads to the second argument of the minimization in (4). In general, $2^k - 1$ inputs should be taken into account instead of k, which can be similarly applied to the bit-wise proof in [8] for the first argument and for nonprimitive feedback polynomials.

The upper bound (4) indicates that there is less symbol-wise effective free distance available than bit-wise, as expected. Indeed, grouping any convolutional code's error events symbol-wise instead of bit-wise can only reduce the effective distance.

C. Range of Encoders to Search

Without concatenation, searching for good trellis codes that maximize free distance requires examining only one code within each group of range-equivalent encoders. Two encoders are called *range-equivalent* if they have the same set of possible output sequences [10] (Forney's notion of equivalence). So, it is sufficient to restrict attention within a set of canonical encoders, which are identified by Forney [11]. For turbo codes, the mapping from input to output sequences plays an important role. Range-equivalent codes can have quite different performance. For example, feedback encoders always have a range-equivalent feedforward encoder which would perform poorly with parallel concatenation.

Define as *input-Hamming-weight equivalent* encoders that map the same input weight error events to the same output distance. If two encoders are not input-Hamming-weight equivalent, we call them input-Hamming-weight distinct. When searching for constituent encoders that maximize effective distance, it is sufficient to examine all codes that are input-Hamming-weight distinct to each other.

We now examine the structural properties of encoders that should be included in this search. A general description of a convolutional encoder with k inputs, n outputs, and m memory elements is given by the state-space equations over $GF(2)$

$$\begin{aligned} s_{j+1} &= s_j \mathbf{A} + u_j \mathbf{B} \\ x_j &= s_j \mathbf{C} + u_j \mathbf{D} \end{aligned} \tag{5}$$

where s_j is the state vector of dimension $1 \times m$, x_j is the output vector of dimension $1 \times n$, u_j is the input vector of dimension $1 \times k$, matrix $\mathbf{B}$ has dimension $k \times m$, matrix $\mathbf{C}$ has dimension $m \times n$, and matrix $\mathbf{D}$ has dimension $k \times n$.

Matrix $\mathbf{A}$ determines the way the m memory elements are connected. For a feedback encoder, $\mathbf{A}$ is the companion matrix of the encoder's feedback polynomial. The companion matrix of a polynomial $f(D) = D^4 + f_3D^3 + f_2D^2 + f_1D + f_0$ is defined [12] as

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ f_0 & f_1 & f_2 & f_3 \end{bmatrix}. \tag{6}$$

For example, the upper constituent encoder in Fig. 1 has feedback polynomial $f(D) = D^4 + D + 1$ which corresponds to the matrix $\mathbf{A_1}$ in the encoders state-space description

$$\begin{aligned}[s_1\ s_2\ s_3\ s_4]_{j+1} =& [s_1\ s_2\ s_3\ s_4]_j \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}}_{\mathbf{A_1}} \\ &+ [u_1\ u_2\ u_3\ u_4]_j \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}}_{\mathbf{B_1}}\end{aligned} \tag{7}$$

$$\begin{aligned}[y_1\ y_2\ y_3\ y_4]_j =& [s_1\ s_2\ s_3\ s_4]_j \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}}_{\mathbf{C_1}} \\ &+ [u_1\ u_2\ u_3\ u_4]_j \underbrace{\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{D_1}}.\end{aligned} \tag{8}$$

Consider the state vector similarity transformation $s_j = \hat{s}_j\mathbf{S}$, where $\mathbf{S}$ is a nonsingular matrix. Under this transformation, the encoder described by the linear system

$$\begin{aligned}\hat{s}_{j+1} &= \hat{s}_j\mathbf{SAS}^{-1} + u_j\mathbf{BS}^{-1} \\ x_j &= \hat{s}_j\mathbf{SC} + u_j\mathbf{D}\end{aligned} \tag{9}$$

has the same generator matrix [13] $G(D)$ as (5)

$$\begin{aligned}G(D) &= \mathbf{D} + \mathbf{BS}^{-1}(D^{-1}\mathbf{I} - \mathbf{SAS}^{-1})^{-1}\mathbf{SC} \\ &= \mathbf{D} + \mathbf{B}(D^{-1}\mathbf{I} - \mathbf{A})^{-1}\mathbf{C}\end{aligned} \tag{10}$$

where $\mathbf{I}$ is the identity matrix. Moreover, an invertible transformation maps the zero state to the zero state, and thus the encoders have the same mapping from input error events to output error events. The matrix $\mathbf{SAS}^{-1}$, with $\mathbf{S}$ nonsingular, is called *similar* to the matrix $\mathbf{A}$. In an exhaustive search for encoders which are input-Hamming-weight distinct, it is redundant to examine similar matrices $\mathbf{A}$. For example, it is redundant to examine the matrix $\mathbf{A_2}$ that is similar to matrix $\mathbf{A_1}$

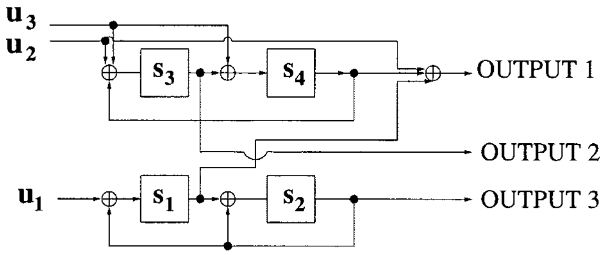


Fig. 3. An encoder with two rows of memory elements.

$$\underbrace{\begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{A_2}} = \underbrace{\begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{S}} \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}}_{\mathbf{A_1}} \cdot \underbrace{\begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{S^{-1}}}. \tag{11}$$

The range of matrices $\mathbf{A}$ to consider can be found from the rational form theorem presented in Horn and Johnson [12, p. 154] and the references therein. This theorem states that any matrix over a field $\mathcal{F}$ ($\mathcal{F} = GF(2)$ in our case) is similar over $\mathcal{F}$ to a matrix that may be written as the direct sum of R companion matrices, i.e., to a block diagonal matrix with R block elements each having the form of (6). This block diagonal matrix is the same as the matrix $\mathbf{A}$ of an encoder with R rows of memory elements.

In other words, this theorem states that for any convolutional encoder with m memory elements, no matter how these memory elements are connected, there exists an input-Hamming-weight equivalent encoder with the memory elements connected in R rows for some R. The following theorem helps to further refine the encoder structures of interest. The proof is provided in the Appendix.

Theorem 1: Consider a convolutional encoder with k inputs, r parity outputs, and m memory elements. For all (k, m, r) values such that

$$k < \min(2^{m-1} - 2, r(2^{m-2} - 1)) \tag{12}$$

the bound in (3) cannot be achieved if the m memory elements are connected in multiple "disconnected" memory rows, that is, multiple distinct memory rows with no common inputs.

Disconnected memory rows are described by block diagonal matrices $\mathbf{A}$ and $\mathbf{B}$, so the state equation in (5) can be decomposed into a separate state equation for each memory row. Fig. 3

shows an example of an encoder with two disconnected memory rows, described by the state equation

$$[s_1\ s_2\ s_3\ s_4]_{j+1} = [s_1\ s_2\ s_3\ s_4]_j \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\mathbf{A}}$$

$$+ [u_1\ u_2\ u_3]_j \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}}_{\mathbf{B}_1}. \qquad (13)$$

In an exhaustive search, if the memory elements connected in a single row give a code with d_{b2}^H that achieves the upper bound, there is no need to expand the search to multiple rows. This is very often the case in practice. The rational form theorem and Theorem 1 usually result in a small number of matrices $\mathbf{A}$ to examine. However, for each matrix $\mathbf{A}$ all matrices $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{D}$ still have to be examined.

The input vector transformation $\hat{u} = u\mathbf{T}$ for any $k \times k$ matrix $\mathbf{T}$ that performs row permutation also leads to input-Hamming-weight equivalent encoders. Indeed, the corresponding generator matrices $G(D)$ and $\mathbf{T}G(D)$ still map the same input weight error events to the same output distance. Thus, in an exhaustive search there is no need to examine both sets of matrices $(\mathbf{B}, \mathbf{D})$ and $(\mathbf{TB}, \mathbf{TD})$. For example, keep only the matrices $\mathbf{B}$ where each row (interpreted as a binary number) is greater than the previous row, coupled with all possible matrices $\mathbf{D}$. Similarly, output Hamming weight is not affected by permuting the r outputs, which in this case reduces the number of matrices $\mathbf{C}$ to examine. To describe an encoder we give in octal notation the feedback polynomial f, the k rows $\{\mathbf{b}_1 \cdots \mathbf{b}_k\}$ of matrix $\mathbf{B}$, the r columns $\{\mathbf{c}_1 \cdots \mathbf{c}_r\}$ of matrix $\mathbf{C}$, and the r columns $\{\mathbf{d}_1 \cdots \mathbf{d}_r\}$ of matrix $\mathbf{D}$ that correspond to the r parity outputs. For example, the upper constituent encoder in Fig. 1 with state-space equations (7), (8) is described as

$$\begin{aligned} &\{f,\, \mathbf{b}_1,\, \mathbf{b}_2,\, \mathbf{b}_3,\, \mathbf{b}_4,\, \mathbf{c}_1,\, \mathbf{c}_2,\, \mathbf{d}_1,\, \mathbf{d}_2\} \\ &\quad = \{023,\, 010,\, 05,\, 01,\, 03,\, 017,\, 04,\, 015,\, 016\}. \end{aligned}$$

Application to Bit Interleaving: Table I provides code fragments with k inputs, r parity outputs, and m memory elements optimized for d_{b2}^H and identified through exhaustive search using our proposed structure. Such tables are useful for bit-interleaving coupled with binary PSK or 4PSK modulation, and outperform in terms of d_{b2}^H similar code tables provided in [14]. Benedetto *et al.* [15] use a group theoretic approach to propose a different encoder structure and provide encoder tables that are equally good (but not better) in terms of d_{b2}^H than the codes identified in Table I.

Table I includes for each code fragment the upper bound on the effective Hamming distance d_{b2}^u, the d_{b2}^H distance, the d_{b3}^H distance, and the free distance denoted by d_{free}, with the number of nearest neighbors in parentheses. The code fragments can be made systematic by adding k systematic outputs. So, although their free distance might be zero, the free distance of the complete code will be positive because of the additional systematic outputs. Codes noted with an * have repeated outputs and do not perform well in simulations [14].

TABLE I
CODE FRAGMENTS OPTIMIZED FOR d_{b2}^H

(r=1, k=2) code fragments					
m	$\{f,\mathbf{b}_1,\mathbf{b}_2,\mathbf{c}_1,\mathbf{d}_1\}$	$\mathbf{d}_{\mathbf{b2}}^{\mathbf{u}}$	d_{b2}^H	d_{b3}^H	d_{free}
1	$\{03, 01, 01, 01, 0\}$	**1**	0(1)	∞	0(1)
2	$\{07, 02, 01, 03, 02\}$	**2**	**2**(3)	0(1)	0(17)
3	$\{011, 02, 05, 03, 03\}$	**4**	**4**(2)	2(4)	0(33)
4	$\{027, 010, 03, 07, 03\}$	**6**	**6**(2)	4(12)	0(65)
5	$\{053, 020, 03, 010, 03\}$	**10**	**10**(2)	5(12)	0(129)

(r=2, k=2) code fragments					
m	$\{f,\mathbf{b}_1,\mathbf{b}_2,\mathbf{c}_1,\mathbf{c}_2,\mathbf{d}_1,\mathbf{d}_2\}$	$\mathbf{d}_{\mathbf{b2}}^{\mathbf{u}}$	d_{b2}^H	d_{b3}^H	d_{free}
1	$\{03, 01, 01, 01, 0, 01, 02\}$	**2**	**2**(5)	∞	2(37)
2	$\{07, 02, 03, 02, 03, 01, 03\}$	**4**	**4**(2)	2(2)	2(34)
$*3$	$\{011, 02, 05, 03, 03, 03, 03\}$	**8**	**8**(2)	4(4)	0(33)
3	$\{015, 04, 05, 04, 07, 03, 02\}$	**8**	7(2)	3(1)	1(1)
4	$\{023, 010, 012, 02, 013, 03, 03\}$	**12**	**12**(2)	3(1)	1(2)

(r=1, k=3) code fragments					
m	$\{f,\mathbf{b}_1,\mathbf{b}_2,\mathbf{b}_3,\mathbf{c}_1,\mathbf{d}_1\}$	$\mathbf{d}_{\mathbf{b2}}^{\mathbf{u}}$	d_{b2}^H	d_{b3}^H	d_{free}
1	$\{03, 01, 01, 01, 01, 0\}$	**1**	0(3)	∞	0(3)
2	$\{05, 02, 02, 03, 03, 05\}$	**2**	1(2)	0(2)	0(∞)
3	$\{017, 04, 02, 07, 06, 05\}$	**3**	2(6)	∞	0(∞)
4	$\{033, 014, 02, 05, 013, 07\}$	**4**	**4**(2)	2(7)	0(∞)

Application to Symbol Interleaving: For PCTCM, we are interested in $(k,\, m,\, r + k/2)$ constituent encoders with r parity and $k/2$ systematic outputs optimized for d_{s2}^E. Theorem 1 can be extended to symbol-wise inputs by replacing k with $2^k - 1$. Theorem 1 and Section II-B refer to output Hamming distance. For the simulations, we want to maximize the output Euclidean distance d_{s2}^E. Although there is no monotone relation between Hamming and Euclidean distance, they are closely related. For example, for 16QAM and Gray labeling [16], it holds that

$$d_{s2}^H \leq d_{s2}^E \leq 2d_{s2}^H. \qquad (14)$$

Motivated by the previous arguments, and because completely exhaustive searches are beyond our computational capabilities, in Section IV, we restrict our attention to encoders with memory elements connected in a single row.

For symbol interleaving, to have $d_{s1} = \infty$, the k rows $\{\mathbf{b}_1 \cdots \mathbf{b}_k\}$ of matrix $\mathbf{B}$ of dimension $k \times m$ have to be linearly independent, i.e., matrix $\mathbf{B}$ has to be full-rank and $k \leq m$. Moreover, the input vector transformation $\hat{u} = u\mathbf{T}$ for any $k \times k$ nonsingular matrix $\mathbf{T}$ does not affect the *symbol-wise* input-Hamming-weight. So there is no need to examine both sets of matrices, $\{\mathbf{A},\, \mathbf{B},\, \mathbf{C},\, \mathbf{D}\}$ and $\{\mathbf{A},\, \mathbf{TB},\, \mathbf{C},\, \mathbf{TD}\}$, for any nonsingular $\mathbf{T}$.

In the special case where $k = m$, the rows/columns of matrix $\mathbf{B}$ form a basis. Any other basis is related with a linear transformation to it. So, in an exhaustive search, it is sufficient to use any one full-rank matrix $\mathbf{B}$, since any other full-rank matrix $\hat{\mathbf{B}}$ is related through a linear transformation $\mathbf{T}$ to it, coupled with all possible matrices $\mathbf{D}$.

III. Interleaver Design

The turbo-encoder performance depends upon both the constituent encoders and the interleaver it employs. This section addresses the interleaver design, which can be applied to both bit and symbol interleaving.

TABLE II
SQUARED EUCLIDEAN DISTANCE FOR ERROR EVENTS. ROW W: SYMBOL-WISE INPUT WEIGHT. COLUMN L: SYMBOL-WISE ERROR EVENT LENGTH

$W \backslash L$	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
2	1.17	1.17	1.76	2.34	2.34	2.34	2.34	2.93	2.93	4.10	4.10	4.10	4.10	4.69	4.69	5.87
3	–	0.59	0.59	0.59	1.17	1.17	1.75	1.75	2.34	2.34	2.34	2.92	2.92	2.92	2.92	3.51
4	–	–	0.59	0.59	0.59	0.59	0.59	0.59	1.17	1.17	1.76	1.76	1.76	2.34	2.34	2.93

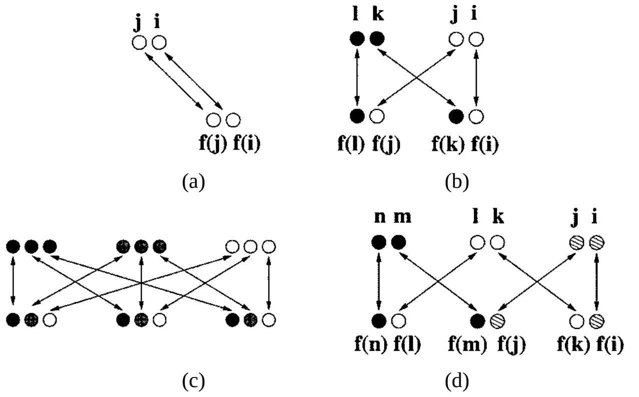


Fig. 4. Small input weight error events.

An interleaver of length N is completely described by a mutually exclusive and collectively exhaustive listing of the integers from 1 to N. Define $f(i)$ to be the integer in the ith position in the list. The input symbol in position i before interleaving is in position $f(i)$ after interleaving.

A. Spread Interleaver

The role of the interleaver is to interconnect the error events of the constituent encoders in such a way that the total output weight of a turbo-encoder codeword accumulates distance from as many distinct error events as possible from each constituent encoder.

A commonly used example is that when a constituent encoder has a single error event of input weight one, it unavoidably maps to a single error event of weight one in the second constituent encoder. This produces a very small total output weight, unless the constituent encoders have infinite impulse responses. A second way to have a small number of interconnected error events is depicted in Fig. 4(a) where component symbols of one error event in the upper encoder become part of the same error event in the second encoder. This case can be avoided by using the spread interleaver introduced by Divsalar and Pollara. The spread interleaver is described in [17] as a semi-random interleaver based on the random selection without replacement of N integers from 1 to N under the following constraint.

Constraint 1: The ith randomly selected integer $f(i)$ must be rejected if there exists $j < i$, such that

$$0 < i - j \le S_1 \qquad |f(i) - f(j)| \le S_2. \tag{15}$$

This constraint guarantees that if two symbols i, j are within distance S_1 in the upper constituent encoder, they cannot be mapped to distance less than S_2 in the lower constituent encoder.

B. Extended Spread Interleaver

An extension of the spread interleaver concept considers multiple error events in the upper encoder. As an example, Fig. 4(b) depicts two error events of the upper encoder that interchange their component symbols, and thus the weight accumulation stops in two steps. To avoid this situation, we define two more parameters T_1 and T_2 and impose on the construction of the spread interleaver an additional constraint. Again, randomly select without replacement integers from 1 to N, and if the ith selection $f(i)$ satisfies Constraint 1 described previously, check if the following condition is also satisfied.

Constraint 2: The ith randomly selected integer $f(i)$ must be rejected if there exist j, k, $l < i$, such that

$$\begin{aligned} &0 < i - j \le T_1 \qquad |f(i) - f(k)| \le T_2\\ &0 < |k - l| \le T_1 \qquad |f(j) - f(l)| \le T_2. \end{aligned} \tag{16}$$

This constraint guarantees that two relatively close component symbols i and j in the upper encoder do not have $f(k)$ near $f(i)$ and $f(l)$ near $f(j)$ in the lower encoder, with k and l near each other in the upper encoder. Fig. 4(b) and (c) illustrate error events that are avoided.

This procedure can be extended to three error events in the upper encoder. Define parameters X_1 and X_2 and impose on the semi-random interleaver the following additional condition.

Constraint 3: The ith randomly selected integer $f(i)$ must be rejected if there exist j, k, l, m, $n < i$, such that

$$\begin{aligned} &0 < i - j \le X_1 \qquad |f(i) - f(k)| \le X_2\\ &0 < |k - l| \le X_1 \qquad |f(j) - f(m)| \le X_2\\ &0 < |m - n| \le X_1 \qquad |f(n) - f(l)| \le X_2. \end{aligned} \tag{17}$$

Fig. 4(d) illustrates an example of an avoided error event. Extension to more than three error events is usually not of interest, because it leads to increased output weight that does not determine the free distance.

To motivate the introduction of Constraints 2 and 3, consider the following example for a symbol-interleaved system with constituent encoders of rate 4/3 employing 8PSK. The element (w, l) of Table II is the minimum squared Euclidean distance, associated with a constituent encoder error event with symbol-wise input Hamming weight w and symbol-wise length l. Observe that when Constraint 1 is satisfied with $(S_1, S_2) = (10, 10)$, the minimum squared Euclidean distance that can be associated with the error event depicted in Fig. 4(a) is $4.10 + 1.17 = 5.27$ (let the upper error event have length 11 and the lower 2). For the case depicted in Fig. 4(b) though, if the constituent error events have length 2 or 3, the associated squared Euclidean distance is $4 \times 1.17 = 4.68$. For the case depicted in Fig. 4(c), the minimum squared Euclidean distance is $6 \times 0.59 = 3.54$. Thus, these error events dominate the performance and should be mitigated before further increasing S_1 and S_2. Similarly, the minimum squared Euclidean distance associated with the error event depicted in Fig. 4(d) is $6 \times 1.17 = 7.02$, so this error event also determines the performance for S_1 and S_2 larger than 16.

The interleaver performance is closely related to the constituent encoders that are employed. A constituent encoder, to take advantage of the interleaver structure, must have the following important property: the output weight of the error events with small input weight must increase with the error event length.

C. Construction Procedure

A uniform interleaver of length N is created by randomly selecting without replacement integers from 1 to N with equal probability. For a semi-random interleaver, the randomly selected integers need to satisfy a set of imposed constraints. This section presents a technique for constructing such interleavers.

The generation of a length N interleaver consists of N steps, where each step selects an integer for the respective position. At the ith step of the interleaver generation, the interleaver contains $i-1$ assigned numbers and there exist $N-i+1$ unassigned numbers. Randomly select one of the $N-i+1$ unassigned numbers with equal probability, for example, number j. Check if placing number j at interleaver position i violates any of the imposed constraints. If it does not violate any constraints, then continue with the next step $i+1$. If it does violate a constraint, try to place j in between two other previously assigned indices. Uniformly choose one of the i candidate positions and check if placing j there violates any of the constraints. Continue until either all previously assigned indices have been examined or a suitable position is found. If there does not exist an appropriate position, repeat for a number selected among the unassigned and not already examined $N-i$ numbers k, $k \neq j$.

This procedure does not guarantee that it will identify a semi-random interleaver that meets the constraints, even if such an interleaver exists, but generally gives good results. Whether such an interleaver exists depends upon the interleaver length and the constraint parameter values.

IV. Simulation Results

This section provides simulation results for 2 bits/s/Hz employing 16QAM and 8PSK and 4 bits/s/Hz employing 64QAM $= 2 \times 8$ PAM.

Table III contains in octal notation codes identified through computer search, and optimized for normalized d_{s2}^E with the edge profile optimal [10], [16] constellation labelings illustrated in Fig. 5. Our search over all interesting constituent convolutional encoders mapped onto the chosen labeling, in fact, produces all interesting constituent codes that could be found with any other labeling related to the chosen one by a binary linear transformation. If two constellations are related with a linear transformation, an exhaustive search would lead to the same encoders with the linear transformation applied to their output. The set of labelings related to each other with a linear transformation is broad enough to include all common labelings. For example, the Gray, natural, and reordered 8PSK labeling reported in [1] are included in such a set.

Each code has the $k/2$ MSB inputs as systematic outputs and r parity outputs. To describe a code, we give in octal notation the feedback polynomial f, the k rows $\{\mathbf{b}_1 \cdots \mathbf{b}_k\}$ of matrix $\mathbf{B}$, the r columns $\{\mathbf{c}_1 \cdots \mathbf{c}_r\}$ of matrix $\mathbf{C}$, and the r columns $\{\mathbf{d}_1 \cdots \mathbf{d}_r\}$ of matrix $\mathbf{D}$ that correspond to the r parity outputs (for an example, see Section II). The simulated codes are shown in boldface. The search identified a large number of codes with the same value of d_{s2}^E.

TABLE III
Codes Optimized for d_{s2}^E

Codes optimized for d_{s2}^E for 16-QAM		
$\{f, \mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3, \mathbf{b}_4, \mathbf{c}_1, \mathbf{c}_2, \mathbf{d}_1, \mathbf{d}_2\}$	d_{s2}^E	d_{s3}^E
{**023,010,05,01,03, 04,017,016,015**}	3(4)	1(1)
{023,010,05,01,03,04,06,016,011}	3(4)	2(2)
{031,010,011,013,017, 014,06,015,04}	3(5)	1(1)
Codes optimized for d_{s2}^E for 64-QAM=8 × 8-PAM		
$\{f,\mathbf{b}_1,\mathbf{b}_2,\mathbf{b}_3,\mathbf{b}_4,\mathbf{c}_1,\mathbf{d}_1\}$	d_{s2}^E	d_{s3}^E
{**027,01,011,015,07,015,014**}	0.3810(3)	0.1905(5)
{035,010,011,013,07,016,010}	0.3810(3)	0.1905(5)
{025,010,014,01,07,05,011}	0.3810(7)	0.3810(66)
Codes optimized for d_{s2}^E for 8 PSK		
$\{f,\mathbf{b}_1,\mathbf{b}_2,\mathbf{b}_3,\mathbf{b}_4,\mathbf{c}_1,\mathbf{d}_1\}$	d_{s2}^E	d_{s3}^E
{**027,010,06,01,03,011,014**}	1.171573(3)	0.585786(5)
{**031,010,012,015,011,015,017**}	1.171573(5)	0.585786(5)
{035,012,016,01,015,011,05}	1.171573(3)	0.585786(5)
{027,015,014,010,07,03,016}	1.171573(3)	0.585786(5)

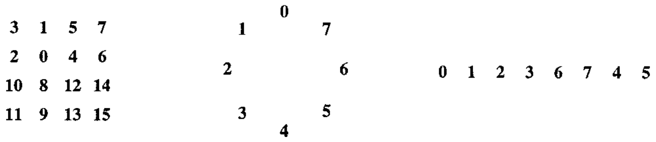


Fig. 5. Labeling for the constellations used in the simulations.

The constituent encoders of rate greater than one (4/3, for example) are catastrophic, but the overall turbo encoder is not. The upper constituent encoder has as systematic bits the $k/2$ MSB input bits, so in the catastrophic loops (i.e., nonzero-input, zero-output loops) only the $k/2$ LSB bits may be nonzero. Similarly the lower constituent encoder has as systematic bits the $k/2$ LSB input bits so in a catastrophic loop only the $k/2$ MSB input bits may be nonzero. Because the input symbols for the upper and lower constituent encoder catastrophic loops are different, the overall turbo encoder does not have an error event that involves catastrophic loops in both the constituent encoders, so the overall turbo encoder is not catastrophic. Each constituent encoder individually implies no coding gain; the coding gain of our system is provided by connecting two constituent encoders in the proposed turbo-encoder structure.

The interleavers used in the simulations are uniform random or semi-random, as specified in each case. To describe an interleaver we give the constraint parameters in the following order: (S, T, X), where $S_1 = S_2 = S$, $T_1 = T_2 = T$, and $X_1 = X_2 = X$. A uniform random interleaver can be described by the spread parameters (0, 0, 0).

The performance is compared against the constrained capacity, which is the mutual information between the channel's input drawn uniformly from a finite constellation and the channel's output [18].

A. Two-bits/s/Hz PCTCM with 16QAM

For 2-bits/s/Hz PCTCM with 16QAM, the constituent encoders implement 4/4 codes with $r = 2$ parity and $k/2 = 2$ systematic outputs, and have $m = 4$ memory elements. The

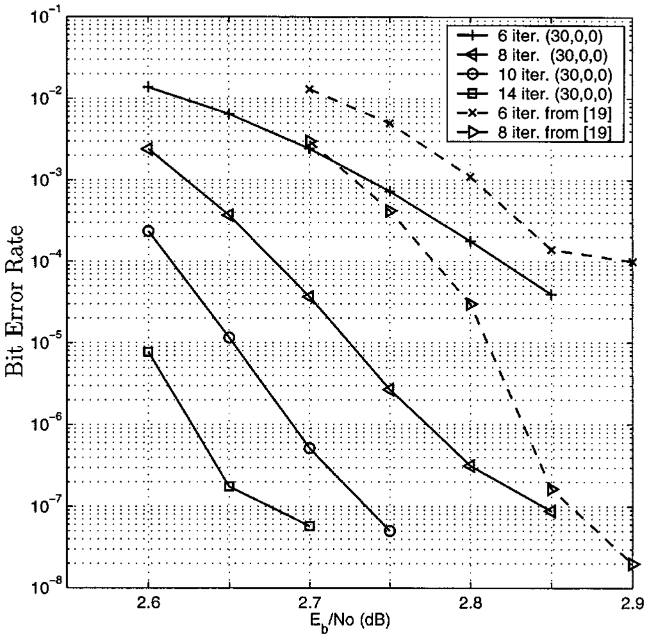


Fig. 6. Two-bits/s/Hz turbo code employing 16QAM. Capacity = 1.76 dB. Constrained capacity = 2.1 dB. Interleaver length = 8192 symbols. Input block size 8192×4 bits.

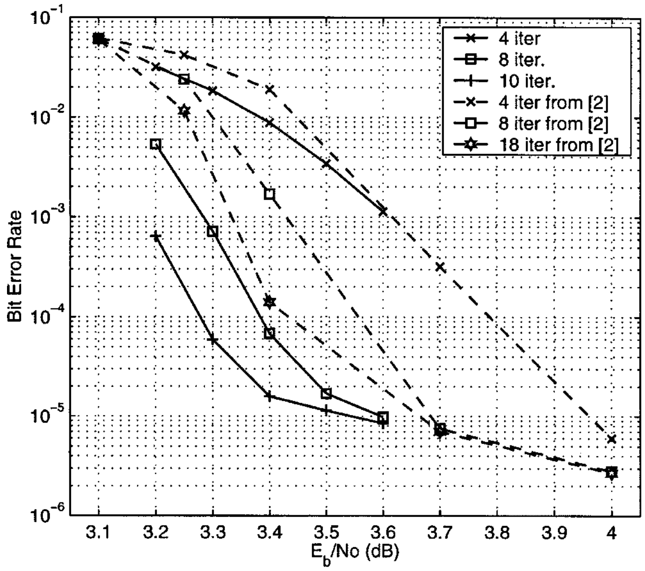


Fig. 7. Two-bits/s/Hz turbo code employing 8PSK. Capacity = 1.76 dB. Constrained capacity = 2.8 dB. Interleaver length = 2500 symbols. Input block size 2500 × 4 bits.

whole turbo encoder is depicted in Fig. 1. The simulated code is in the first row of Table III.

For interleaver length 8192 symbols (input block size in bits: 8192 × 4), the performance is within 0.5 dB of constrained capacity at BER 10^{-5} (Fig. 6) with a semi-random interleaver (30-0-0). Compared with the bit-interleaved performance in [19] for 2-bits/s/Hz PCTCM with 16QAM, constituent encoders with four memory elements, and interleaver length 16 384 (input block size in bits: 16 384 × 2), the proposed symbol-interleaved system can converge 0.1 dB earlier with the same number of decoder iterations, and 0.2 dB earlier with a few more decoder iterations. The proposed system has an error floor at around 5 × 10^{-8} which is higher than the error floor in [19]. This is because the smaller symbol-wise effective distance leads to a smaller $d_{\rm free}$ for the symbol-interleaved system and thus a higher error floor. Convergence at a lower signal-to-noise ratio (SNR) with increased number of iterations may be useful for applications such as deep-space communications.

B. Two-bits/s/Hz PCTCM with 8PSK

For 2-bits/s/Hz PCTCM with 8PSK, the constituent encoders implement a 4/3 code with $r = 1$ parity and $k/2 = 2$ systematic outputs and have $m = 4$ memory elements. The simulated code is at the first row of Table III. The turbo encoder is depicted in Fig. 2.

For interleaver length 2500 symbols (input block size in bits: 2500 × 4), the performance is within 0.6 dB of constrained capacity at $\mathrm{BER} = 10^{-5}$ (Fig. 7). Compared with the symbol-interleaved system in [2] for interleaver length 5000 symbols (input block size in bits: 5000 × 2), 2-bits/s/Hz PCTCM with 8PSK but constituent encoders with $m = 3$ memory elements (which leads to roughly half the decoder complexity), the proposed system can converge up to 0.25 dB earlier. The random interleaver causes a high error floor, which can be lowered by using a more elaborate interleaver. Fig. 8 shows that the interleaver (20, 4, 1) lowers the error floor compared to the random (0, 0, 0) interleaver.

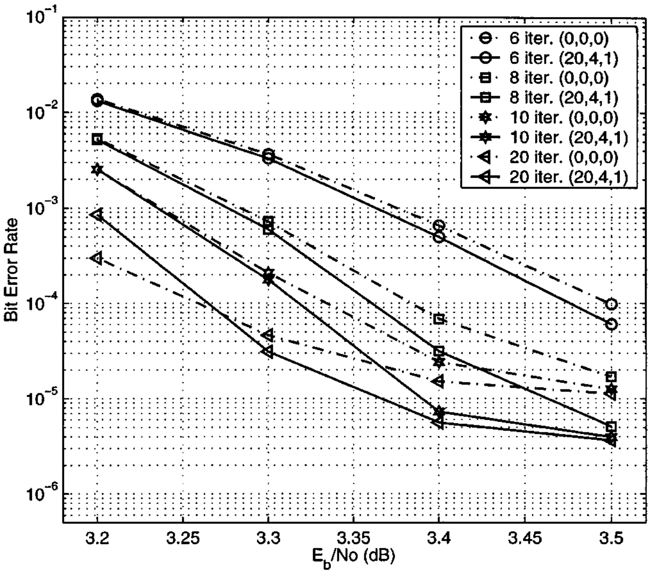


Fig. 8. Two-bits/s/Hz/ turbo code employing 8PSK. Performance with a (20, 4, 1) interleaver. Interleaver length = 2500 symbols. Input block size 2500× 4 bits.

Fig. 9 shows that by using different constituent encoders a designer has the option to trade-off convergence at a lower SNR with a lower error floor. The first encoder can converge 0.1 dB earlier, while the second encoder has an error floor more than an order of magnitude lower. The first encoder is the same as in Fig. 8 with the (20, 4, 1) interleaver. The second encoder is described by the encoder polynomials in the second row of Table III, and employs a (20, 5, 0) interleaver.

To identify the second encoder, we observed that in Fig. 8 the T type error events [Fig. 4(b)] determine the free distance. The

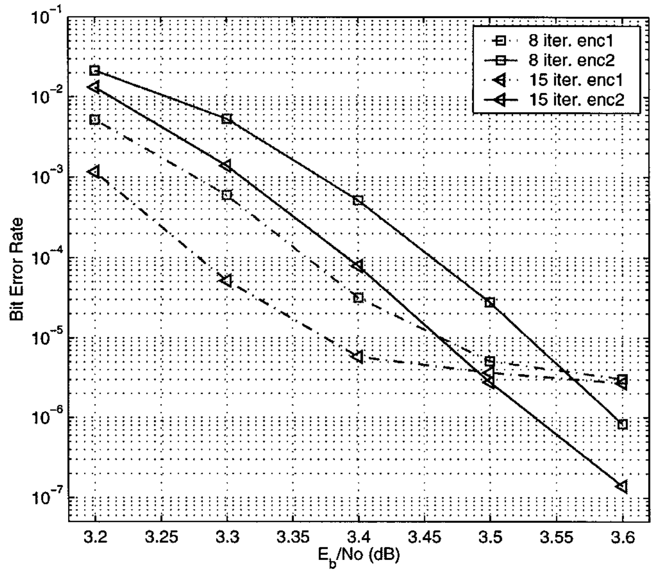


Fig. 9. Two-bits/s/Hz/ turbo code employing 8PSK. The first encoder employs a (20, 4, 1) interleaver and the second encoder employs a (20, 5, 0). Input block size 2500× 4 bits.

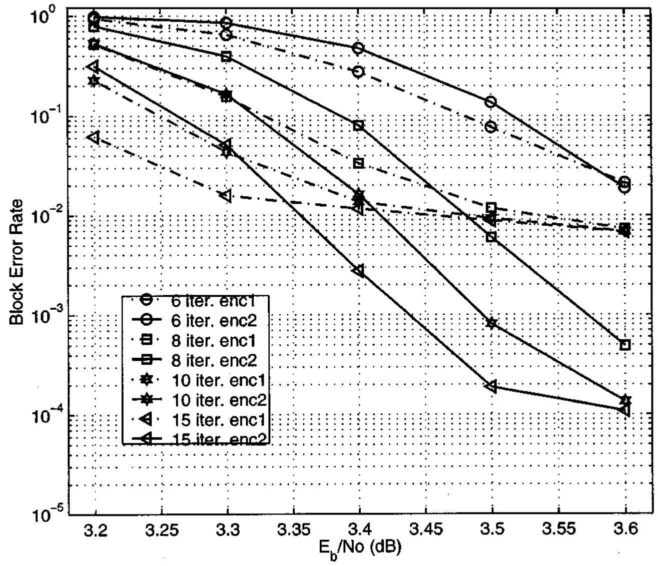


Fig. 10. Four bits/s/Hz turbo code employing 8PAM. Capacity 5.74 dB. Constrained Capacity 6.6 dB. Interleaver length 4096 symbols. Input block size 4096×4 bits.

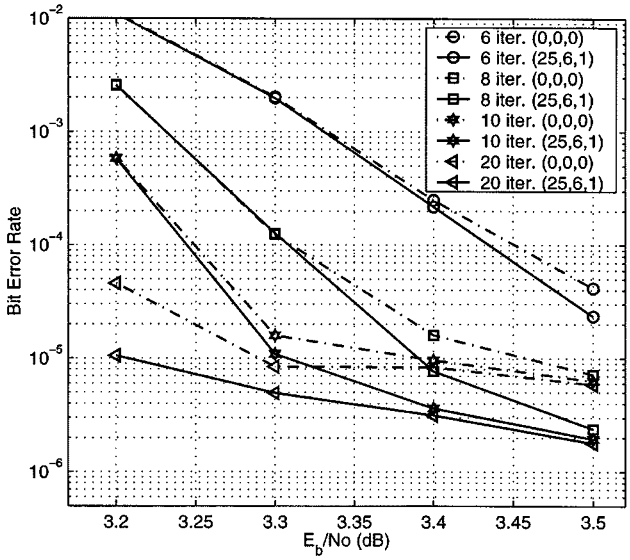


Fig. 11. Two-bits/s/Hz turbo code employing 8PSK. Capacity = 1.76 dB. Constrained capacity = 2.8 dB. Interleaver length = 5000 symbols. Input block size 5000 × 4 bits.

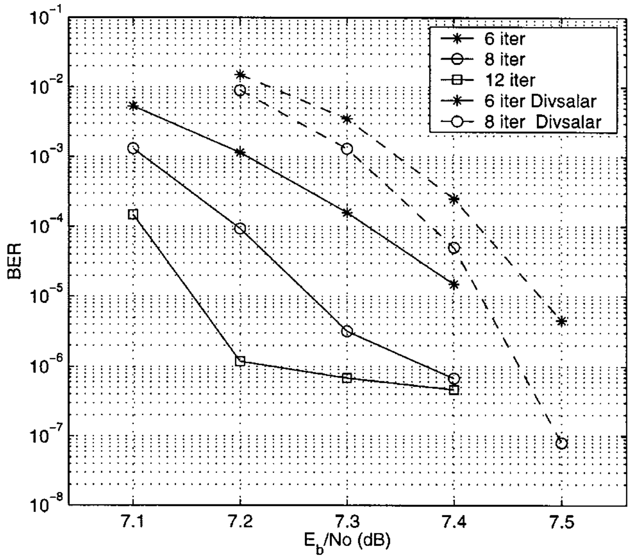


Fig. 12. Four bits/s/Hz/ turbo code employing 8PAM. Capacity 5.74 dB. Constrained Capacity 6.6 dB. Interleaver length 4096 symbols. Input block size 4096×4 bits.

highest T parameter value interleaver the construction procedure could create for interleaver length 2500 was the (20, 5, 0) interleaver. We performed a search among all encoders with the largest value of d_{s2}^{E} to find the encoder that achieves the largest free distance with this interleaver. We additionally required that the output weight of input weight two error events increases with the error event length. In other words, we specifically designed the second encoder to perform well with the (20, 5, 0) interleaver. Fig. 10 shows the block-error rate performance for the two encoders [20]

For interleaver length 5000 symbols (input block size in bits: 5000 × 4), the performance is within 0.5 dB of constrained capacity at BER = 10^{-5} (Fig. 11) for the first encoder. The interleaver (25, 6, 1) lowers the error floor compared to the random (0, 0, 0) interleaver.

C. Four-bits/s/Hz PCTCM with 8×8 PAM = 64QAM

For 4-bits/s/Hz PCTCM with 8×8 PAM = 64QAM (Fig. 12), the constituent encoders implement a 4/3 code with $r = 1$ parity and $k/2 = 2$ systematic outputs, and have $m = 4$ memory elements. For interleaver length 4096 symbols (input block size in bits: 4096 × 4), and (30, 0, 0) interleaver, the performance at BER 10^{-6} is within 0.6 dB of constrained capacity. Compared with the bit-interleaved system performance in [1] for 4-bits/s/Hz PCTCM with 64QAM, four memory elements, interleaver length 4096 symbols (input block size in bits: 4096 × 4), and (30, 0, 0) interleaver, the proposed symbol-interleaved system can converge earlier (e.g., approximately 0.2 dB for 8 iterations) but again has a higher error floor at around 5×10^{-7}.

V. Conclusions

This paper examined a method to achieve high spectral efficiency using symbol-interleaved PCTCM. The turbo-encoder design consists of two parts, constituent encoder design and interleaver design. The constituent encoder design determined the optimization criteria and extended the effective distance bound to symbol-wise inputs. The rational form theorem shows that in order to examine all strictly equivalent encoders, it is sufficient to consider only the canonical memory structures with R rows. In many cases, only encoders with the memory elements connected in a single row ($R = 1$) or with common inputs can achieve the maximum output effective distance. The interleaver design extended the spread interleaver design to take into account multiple error events and proposed a semi-random interleaver construction method. Simulation results for 2 bits/s/Hz with 16QAM and 4 bits/s/Hz with 64QAM show that the proposed symbol-interleaved system, as compared to bit interleaving reported in the literature, can converge at lower SNR but at the cost of a higher error floor, which is due to a lower free distance.

Appendix
Proof of Theorem 1

It suffices to show that the use of multiple rows of memories enforces an upper bound on the effective free distance lower than the bound in (3).

Assume that the m memory elements are connected in R rows with m_j memory elements in row j, $j = 1 \cdots R$, and $\sum_{j=1}^{R} m_i = m$. Let k_j be the number of inputs in row j, $k_j \le k$, $\sum_{j=1}^{R} k_j = k$, and r_j be the number of outputs from row j, $r_j \le r$. In the example of Fig. 3, $R = 2$, $m = 4$, $m_1 = m_2 = 2$, $k = 3$, $k_1 = 2$, $k_2 = 1$, $r = 3$, and $r_1 = r_2 = 2$.

For one memory chain j, d_{b2}^H is bounded by

$$d_{b2}^H \le \min\left(\left\lceil \frac{2^{m_j}}{k_j} \right\rceil r_j,\, 2r_j + \left\lfloor \frac{2^{m_j-1} r_j}{k_j} \right\rfloor\right). \tag{18}$$

So, for the total encoder, it holds that

$$\begin{aligned} d_{b2}^H &\le \min_{j=1\cdots R}\left(\min\left(\left\lceil \frac{2^{m_j}}{k_j} \right\rceil r_j, 2r_j + \left\lfloor \frac{2^{m_j-1} r_j}{k_j} \right\rfloor\right)\right) \\ &\le \min_{j=1\cdots R}\left(\min\left(\left\lceil \frac{2^{m_j}}{k_j} \right\rceil r, 2r + \left\lfloor \frac{2^{m_j-1} r}{k_j} \right\rfloor\right)\right). \end{aligned} \tag{19}$$

To show that the multiple-row upper bound is lower, it suffices to show that it is lower for both the two terms of the min function in (3).

1) For the second term of the bound, if the m memories are connected in one row

$$d_{b2}^H \le 2r + \left\lfloor \frac{2^{m-1} r}{k} \right\rfloor. \tag{20}$$

If the m memories are connected in R rows

$$\begin{aligned} d_{b2}^H &\le \min_{j=1\cdots R}\left(2r + \left\lfloor \frac{2^{m_j-1} r}{k_j} \right\rfloor\right) \qquad (21) \\ &\le 2r + \left\lfloor \frac{\sum_{j=1}^{R} 2^{m_j-1} r}{k} \right\rfloor. \qquad (22) \end{aligned}$$

2) Similarly, for the first term of the bound, if the m memories are connected in one row

$$d_{b2}^H \le \left\lceil \frac{2^m}{k} \right\rceil r. \tag{23}$$

If the m memories are connected in R rows

$$d_{b2}^H \le \left\lceil \frac{\sum_{j=1}^{R} 2^{m_j}}{k} \right\rceil r. \tag{24}$$

If it holds that $k < 2^m - \sum_{j=1}^{R} 2^{m_j}$ for all possible R and m_j partitions, then (23) and (24), are related by a strict inequality. However, for any integer q, $1 \le q \le m$, integers m_j, $0 < m_j < m$, and integer $R > 1$, it holds that

$$\sum_{j=1}^{R} 2^{m_j} \le 2^{m-q} + 2^q \le 2^{m-1} + 2 \tag{25}$$

so it suffices that $k < 2^{m-1} - 2$. Similarly, for strict inequality between (20) and (22), it suffices that $k < r(2^{m-2} - 1)$.

Acknowledgment

The authors would like to thank D. Divsalar and R. J. McEliece for kindly providing material on their work. They would also like to thank K. Narayan and M. Shane for helpful discussions.

References

[1] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, “Parallel concatenated trellis coded modulation,” in *Proc. IEEE Int. Conf. Communications*, vol. 2, Dallas, TX, June 1996, pp. 974–978.

[2] P. Robertson and T. Worz, “A novel bandwidth efficient coding scheme employing turbo codes,” in *Proc. IEEE Int. Conf. Communications*, vol. 2, Dallas, TX, June 1996, pp. 962–967.

[3] ——, “Bandwidth-efficient turbo trellis-coded modulation using punctured component codes,” *IEEE J. Select. Areas Commun.*, vol. 16, pp. 206–218, Feb. 1998.

[4] A. S. Barbulescu and S. S. Pietrobon, “Interleaver design for turbo codes,” *Electron. Lett.*, vol. 30, pp. 2107–2108, Dec. 8, 1994.

[5] H. Ogiwara and M. Yano, “Improvement of turbo trellis-coded modulation system,” *Trans. Inst. Electron. Inf. Commun. Eng. A*, vol. E81-A, pp. 2040–2046, Oct. 1998.

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

[7] S. Benedetto and G. Montorsi, “Unveiling turbo codes: Some results on parallel concatenated coding schemes,” *IEEE Trans. Inform. Theory*, vol. 42, pp. 409–429, Mar. 1996.

[8] D. Divsalar and R. J. McEliece, “On the design of generalized concatenated coding systems with interleavers,” TMO PR 42-134, Apr.–June 1998.

[9] D. Divsalar and F. Pollara, “On the design of turbo codes,” TMO PR 42-123, July–Sept. 1995.

[10] R. D. Wesel, X. Liu, J. M. Cioffi, and C. Komninakis, "Constellation labeling for linear encoders," IEEE Trans. Inform. Theory, to be published.

[11] G. D. Forney, "Convolutional codes I: Algebraic structure," *IEEE Trans. Inform. Theory*, vol. IT-26, pp. 720–738, Nov. 1970.

[12] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1985.

[13] C.-T. Chen, *Linear System Theory and Design*. Oxford, U.K.: Oxford Univ. Press, 1999.

[14] D. Divsalar and R. J. McEliece, "Effective free distance of turbo codes," *Electron. Lett.*, vol. 32, pp. 445–446, Feb. 29, 1996.

[15] S. Benedetto, R. Garello, and G. Montorsi, "A search for good convolutional codes to be used in the construction of turbo codes," *IEEE Trans. Commun.*, vol. 46, pp. 1101–1105, Sept. 1998.

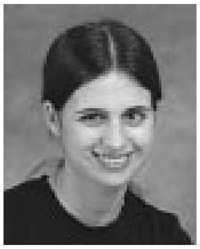
[16] R. Wesel and X. Liu, "Edge-profile optimal constellation labeling," in *Proc. ICC 2000*, New Orleans, LA, June 2000, pp. 1198–1202.

[17] D. Divsalar, S. Benedetto, F. Pollara, and G. Montorsi, "Turbo codes: principles and applications," UCLA EXTENSION, ser. Lecture Notes, Oct. 1997.

[18] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 55–67, Jan. 1982.

[19] D. Divsalar and F. Pollara, "Multiple turbo codes," in *Proc. MILCOM'95 Universal Communications*, vol. 1, San Diego, CA, Nov. 1995, pp. 279–285.

[20] C. Fragouli, "Turbo encoder design for high spectral efficiency," Ph.D. dissertation, Univ. California, Los Angeles, 2000.



Christina Fragouli (M'00) received the B. S. degree in electrical engineering from the National Technical University of Athens, Athens, Greece, in 1996, and the M.Sc. and Ph.D. degrees in electrical engineering from the University of California at Los Angeles in 1998 and 2000, respectively.

She is currently a consultant with AT&T Research Laboratories, Florham Park, NJ. Her research interests include channel coding, channel equalization, and signal processing for communications.

Richard D. Wesel (S'91–M'96–SM'01) received both the B.S. and M.S. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, in 1989, and the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, in 1996.

From 1989 to 1991, he was with AT&T Bell Laboratories where he worked on nonintrusive measurement and adaptive correction of analog impairments in AT&T's long distance network and the compression of facsimile transmissions in packet-switched networks. He holds patents resulting from his work in both these areas. Since 1996, he has been with the University of California at Los Angeles as an Assistant Professor in the Electrical Engineering Department. His research interests include coded modulation, communication theory, and data fusion in distributed sensor networks.

Dr. Wesel received a National Science Foundation CAREER Award to pursue research on robust and rate-compatible coded modulation. He also received an Okawa Foundation Award in 1999 for his research in information and telecommunications. In 1999, he was appointed to the editorial staff of the IEEE TRANSACTIONS ON COMMUNICATIONS in the area of coding and coded modulation. For additional information, please visit http://www.ee.ucla.edu/faculty/Wesel.html.