

# Low-Complexity Concatenated Two-State TCM Schemes With Near-Capacity Performance

Li Ping, *Member, IEEE*, Baoming Bai, *Associate Member, IEEE*, and Xinmei Wang, *Member, IEEE*

**Abstract**—This paper presents a family of concatenated two-state trellis-coded modulation (CT-TCM) schemes. Compared with the existing turbo-type bandwidth-efficient coded modulation schemes, the proposed codes have significantly reduced complexity without sacrificing performance. A joint design strategy for all component codes is established. This leads to so-called asymmetrical and time-varying trellis structures, which possess good Hamming and Euclidean distance distributions. The performance of the proposed codes is demonstrated by simulation results.

**Index Terms**—Error-correction codes, graph codes, iterative decoding, parallel concatenated codes, trellis-coded modulation (TCM), turbo codes.

## I. INTRODUCTION

SINCE the advent of turbo codes [1], various bandwidth-efficient coded modulation schemes based on turbo-like codes with iterative decoding have been investigated [2]–[10]. The bit–interleaver-based schemes considered in [2], [4], [6] involve converting between symbol and bit likelihood values. They generally demonstrate good error floor performance. The symbol–interleaver-based schemes presented in [3], [5] avoid symbol–bit conversions. They have lower decoding complexity and generally demonstrate good performance in the waterfall region. Alternative methods include the multilevel codes presented in [7] and the low-density parity-check (LDPC) code based schemes presented in [8]–[10]. An overview of bandwidth-efficient coded modulation schemes can be found in [11], [12].

The standard turbo-type code designs, including those for coded modulation [2]–[5], employ a symmetrical structure using two identical generator polynomials. Recently, it has been demonstrated that asymmetrical codes (i.e., codes with two different component generator polynomials) have some interesting advantages over symmetrical codes [13]. An asym-

metric turbo-coded modulation scheme was investigated in [14] using the EXIT chart technique [15].

For a trellis code with a spectral efficiency of  $n$  bits per symbol, using a signal constellation of size  $2^{n+1}$  and a state number of less than  $2^n$ , the trellis diagram would inevitably contain parallel branches that are likely to be detrimental to performance. Therefore, for higher spectral efficiency, more complex component codes have to be used (e.g., at least 16-state codes for 32-QAM). This implies that decoding complexity will increase rapidly with constellation size.

In this paper, we present a family of very-low-complexity concatenated two-state trellis-coded modulation (TCM) (CT-TCM) schemes with near-capacity performance. A CT-TCM code consists of multiple two-state component codes (typically more than two), concatenated in parallel by symbol-interleavers. A notable feature of the CT-TCM codes is that the design strategy is based on asymmetrical and time-varying trellises with parallel branches. Several useful parameters are introduced which characterize CT-TCM codes: namely, the minimum divergence degree, pairwise remerging probability, and diverging length. Compared with existing turbo-type coded-modulation schemes [2]–[6], CT-TCM codes offer a low-cost alternative with comparable performance. A complexity analysis is provided in the Appendix, which shows that the cost saving factor is substantial (for example, twelve times lower than the 16-state codes used in [4], [5]).

The paper is organized as follows. In Section II, we introduce the basic principles of binary two-state trellis codes and CT-TCM schemes. Sections III and IV are concerned with the design criteria for CT-TCM codes based on Hamming and Euclidean distances, respectively. In Section V, design examples are presented which demonstrate the performance of CT-TCM codes. Finally, Section VI presents conclusions.

## II. CONCATENATED TWO-STATE TCM SCHEMES

### A. Component Encoder

The component encoder of a CT-TCM code consists of a binary two-state trellis encoder followed by a multi-ary signal mapper, see Fig. 1. Let a binary  $n$ -tuple  $\mathbf{d}_k = (d_{k,0}, \dots, d_{k,n-1})$  be an information symbol. Let  $\mathbf{d} = \{\mathbf{d}_k, k \geq 0\}$  be an input sequence to the binary encoder, producing a coded symbol sequence  $\mathbf{c} = \{\mathbf{c}_k\}$ . Each  $\mathbf{c}_k$  contains a parity-check bit  $q_k$ , i.e.,  $\mathbf{c}_k = (\mathbf{d}_k, q_k)$ , and is mapped to a signal constellation of size  $2^{n+1}$ , producing a modulated symbol  $x_k$ . In the following,  $\mathbf{c}$  and  $\mathbf{x} = \{x_k\}$  are also referred to as unmodulated and modulated codewords, respectively.

Manuscript received April 27, 2001; revised July 23, 2003. This work was supported jointly by the Research Grants Council of Hong Kong SAR, China, under Grant N\_CityU N\_101/01 and by the National Natural Science Foundation of China, under Grant 60131160742. The material in this paper was presented in part at the IEEE International Symposium on Information Theory, Lausanne, Switzerland, June/July 2002.

L. Ping is with the Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong (e-mail: eeliping@cityu.edu.hk).

B. Bai was with the Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong. He is now with the State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an 710071, China (e-mail: bmbai@mail.xidian.edu.cn).

X. Wang is with the State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an 710071, China (e-mail: xmwang@xidian.edu.cn).

Communicated by R. Urbanke, Associate Editor for Coding Techniques.

Digital Object Identifier 10.1109/TIT.2003.820011

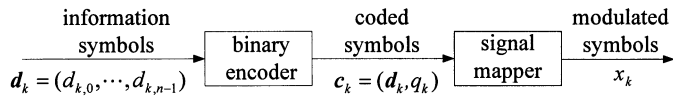


Fig. 1. The component encoder structure for the CT-TCM scheme.

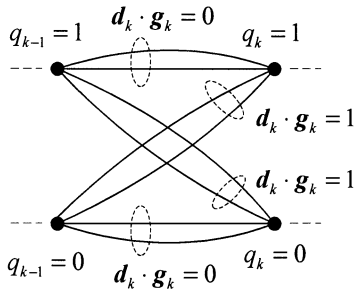


Fig. 2. A two-state trellis diagram with  $2^{n+1}$  branches in the section. (In this example,  $n = 2$ .)

In this paper, we assume that the binary encoder in Fig. 1 is characterized by the two-state trellis in Fig. 2 (similar to the tree encoder in [16]). The parity-check bit  $q_k$  is generated by

$$q_k = q_{k-1} + \mathbf{d}_k \cdot \mathbf{g}_k = \sum_{i=0}^k \mathbf{d}_i \cdot \mathbf{g}_i \pmod{2}, \quad k > 0 \quad (1)$$

with  $\mathbf{q}_0 = \mathbf{d}_0 \cdot \mathbf{g}_0$ . Here,  $\mathbf{g}_k = (g_{k,0}, g_{k,1}, \dots, g_{k,n-1})^T$  is an indicator vector defined by

$$g_{k,j} = \begin{cases} 1, & \text{if } d_{k,j} \text{ participates in parity check} \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

The code in Fig. 2 is completely specified by  $\{\mathbf{g}_k\}$ . For obvious reasons, we will refer to a trellis branch (i.e., a state transition) corresponding to  $\mathbf{d}_k \cdot \mathbf{g}_k = 0$  as a horizontal branch, and a branch corresponding to  $\mathbf{d}_k \cdot \mathbf{g}_k = 1$  as a cross branch.

### B. The Global Encoder of CT-TCM Scheme

Fig. 3 depicts a global CT-TCM scheme, where  $M$  component encoders are concatenated in parallel by  $M$  symbol-interleavers. Modulo- $M$  interleavers satisfying the following constraints are assumed in this paper:

$$\pi^{(m)}(k) \pmod{M} = k \pmod{M}, \quad \text{for } m = 0, 1, \dots, M-1. \quad (3)$$

In order to increase spectral efficiency, we puncture all the modulated symbols in the  $m$ th component code, except those at position  $\{k | k \pmod{M} = m\}$ . This, together with the constraint in (3), ensures that one and only one modulated symbol carrying the same  $\mathbf{d}_k$  is transmitted, and that the punctured symbols are uniformly distributed in each component code. When  $M = 2$ , the above modulo- $M$  interleaving-puncturing rule is equivalent to that used in [3]. We will always assume that a signal constellation of size  $2^{n+1}$  is used. This yields a spectral efficiency of  $n$  bits per symbol. Without confusion, we will still use  $\mathbf{c}$  and  $\mathbf{x}$  to denote unmodulated and modulated codewords, respectively, in the concatenated code.

### C. Iterative Decoder

The CT-TCM decoder structure is based on a multidimensional turbo decoder incorporating the Bahl-Cocke-Jelinek-Raviv (BCJR) algorithm, as detailed in [16]. A brief discussion on iterative decoding and its complexity is given in the Appendix.

### III. DESIGN OF THE UNDERLYING BINARY CODES

Let  $\mathbf{c}$  and  $\tilde{\mathbf{c}}$  be two unmodulated codewords generated by the information words  $\mathbf{d}$  and  $\tilde{\mathbf{d}}$ , respectively. Denote by  $\phi$  the all-zero codeword. The *symbol Hamming distance* between  $\mathbf{c}$  and  $\tilde{\mathbf{c}}$ , denoted by  $D_H(\mathbf{c}, \tilde{\mathbf{c}})$ , is the number of symbols by which they differ. The *symbol Hamming weight* of  $\mathbf{c}$  is defined as  $W_H(\mathbf{c}) \equiv D_H(\mathbf{c}, \phi)$ . The *information Hamming distance* between  $\mathbf{c}$  and  $\tilde{\mathbf{c}}$  is defined as  $ID(\mathbf{c}, \tilde{\mathbf{c}}) \equiv D_H(\mathbf{d}, \tilde{\mathbf{d}})$ . The *information Hamming weight* of  $\mathbf{c}$  is defined as  $IW(\mathbf{c}) \equiv W_H(\mathbf{d})$ .

A common design rule for TCM codes is to optimize the distribution of Euclidean distances [17], which is a complicated task. We now consider a suboptimal procedure that uses the symbol Hamming distance as the design criterion. The rationale is as follows. We assume that a one-to-one mapping is established between the  $2^{n+1}$  branches in a trellis section (see Fig. 2) and the  $2^{n+1}$  signal points in the constellation. Consider  $\mathbf{c}$  and  $\tilde{\mathbf{c}}$  again. A nonzero contribution to their Euclidean distance will be made if their encoding paths are different in a section. Thus, a large symbol Hamming distance between the unmodulated codewords will be very likely to result in a large Euclidean distance between the modulated codewords.

It is convenient to adopt the symbol Hamming weight as a design criterion, since it is a linear metric for the underlying binary code. According to the analysis in [18], the performance of turbo-type codes is dominated by the codewords with  $IW(\mathbf{c}) \leq 2$ . It is reasonable to expect that the performance of CT-TCM codes will behave similarly. Motivated by this, we will concentrate on codewords with  $IW(\mathbf{c}) \leq 2$  for code optimization. It turns out that some very good codes can be designed in this way.

#### A. State Equations in Matrix Form

For convenience, we first assume trivial interleavers  $\{\pi^{(m)}(k) = k, \forall k\}$  for all  $m$ . We will consider the impact of interleavers in Section III-E. Refer to Fig. 3. Define

$$\mathbf{G}_k \equiv [\mathbf{g}_k^{(0)} \quad \dots \quad \mathbf{g}_k^{(m)} \quad \dots \quad \mathbf{g}_k^{(M-1)}] \quad (4a)$$

$$\mathbf{q}_k \equiv [q_k^{(0)} \quad \dots \quad q_k^{(m)} \quad \dots \quad q_k^{(M-1)}] \quad (4b)$$

where  $\mathbf{g}_k^{(m)}$  and  $q_k^{(m)}$  are the  $k$ th indicator vector and the  $k$ th state variable for the  $m$ th component code, respectively. From (1) and (4), we have a matrix-form state equation

$$\mathbf{q}_k = \mathbf{q}_{k-1} + \mathbf{d}_k \mathbf{G}_k = \sum_{i=0}^k \mathbf{d}_i \mathbf{G}_i, \quad \text{with } \mathbf{q}_0 = \mathbf{d}_0 \mathbf{G}_0. \quad (5)$$

#### B. Divergence Degree

*Definition 1:* The number of 1's in the binary vector  $\mathbf{d}_k \mathbf{G}_k$  is called the *divergence degree* of the pair  $(\mathbf{d}_k, \mathbf{G}_k)$  and will be denoted by  $\text{Div}(\mathbf{d}_k, \mathbf{G}_k)$ .

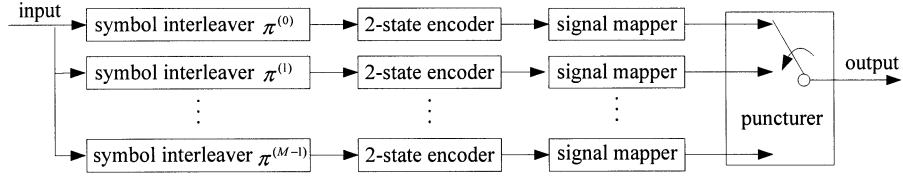


Fig. 3. A global CT-TCM encoder structure.

Comparing Definition 1 with the discussion in Section II-A,  $\text{Div}(\mathbf{d}_k, \mathbf{G}_k)$  represents the number of cross transitions (out of  $M$  component codes) caused by a nonzero  $\mathbf{d}_k$  at the  $k$ th trellis section. The following analogy provides a convenient way to design  $\mathbf{G}_k$  with a required divergence degree distribution.

*Remark 1:* Supposing that  $\mathbf{G}_k$  is used as the generator matrix of a linear block code  $\mathbf{C}$ , then  $\text{Div}(\mathbf{d}_k, \mathbf{G}_k)$  is the weight of the codeword in  $\mathbf{C}$  generated by an information word  $\mathbf{d}_k$ . Let  $\lambda_k = \min_{\mathbf{d}_k \neq \mathbf{0}} \text{Div}(\mathbf{d}_k, \mathbf{G}_k)$ . Then  $\lambda_k$  is the minimum Hamming distance of  $\mathbf{C}$ .

We will use  $\lambda_{\min}$  to denote the minimum  $\lambda_k$  considering all  $k$  and all possible  $\mathbf{d}_k$ , which is a useful parameter for CT-TCM codes. For example, using the generator matrix of a single parity-check (SPC) code for every  $\mathbf{G}_k$ , we have  $\lambda_{\min} = 2$ , so any  $\mathbf{c} (\neq \phi)$  will diverge from  $\phi$  in at least two component codes.

We now consider the codewords where  $IW(\mathbf{c}) = 1$ . Let  $\mathbf{d}_{k'}$  be the only nonzero symbol in an information word  $\mathbf{d}$ . Then

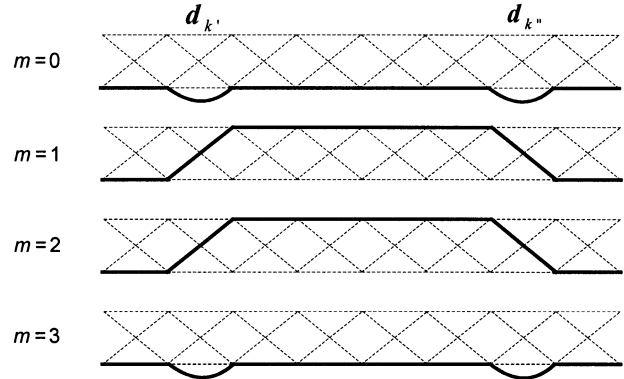
$$\mathbf{q}_k = \mathbf{q}_{k'} = \mathbf{d}_{k'} \mathbf{G}_{k'}, \quad \text{for } k \geq k' \quad (6)$$

(assuming an all-zero initial state for all component codes). Equation (6) indicates that there are exactly  $\text{Div}(\mathbf{d}_{k'}, \mathbf{G}_{k'})$  component encoding paths (i.e., the trellis paths of  $\mathbf{c}$  in the individual component codes) diverging from  $\phi$  at the  $k'$ th trellis section, and they will remain separate from  $\phi$  afterwards (since  $IW(\mathbf{c}) = 1$ ). This is likely to result in a large Hamming weight. It is thus desirable to have large divergence degrees for all  $(\mathbf{d}_k, \mathbf{G}_k)$  pairs and, in particular, to avoid  $\lambda_{\min} = 0$ . It can be verified that  $\lambda_k > 0$  if and only if  $\mathbf{G}_k$  is row-linearly-independent, and this implies that the columns in  $\mathbf{G}_k$  are not all identical. It also implies that  $n \leq M$ , since  $\mathbf{G}_k$  is an  $n \times M$  matrix. Recall that each column in  $\mathbf{G}_k$  contains the encoding information in a component code. Thus, nonidentical  $\mathbf{G}_k$  columns imply different encoding methods in different component codes, and such a code is said to be *asymmetrical*.

Some other general observations can also be made. The *relative divergence degree* between  $(\mathbf{d}_{k_1}, \mathbf{G}_{k_1})$  and  $(\mathbf{d}_{k_2}, \mathbf{G}_{k_2})$  is defined as the number of 1's in  $(\mathbf{d}_{k_1} - \mathbf{d}_{k_2}) \cdot \mathbf{G}_{k_1}$ , where  $\mathbf{d}_{k_1}$  and  $\mathbf{d}_{k_2}$  are two information symbols. Since the Hamming distance is a linear metric, the distribution of relative divergence degrees is completely determined by the distribution of  $\text{Div}(\mathbf{d}_k, \mathbf{G}_k)$ . For example, the encoding paths of  $\mathbf{d}_k$  and  $\mathbf{d}_{k'}$  will diverge from each other in at least  $\lambda_{\min}$  component codes after the first symbol at which  $\mathbf{d}_k$  and  $\mathbf{d}_{k'}$  differ.

### C. Remerging Probability

We now proceed to consider codewords with  $IW(\mathbf{c}) = 2$ . Again we only consider trivial interleavers. Let  $\mathbf{d}_{k'}$  and  $\mathbf{d}_{k''}$  be

Fig. 4. An illustration of a remerging event for  $M = 4$ . The bold lines are the encoding paths corresponding to  $\mathbf{d}$ . The encoding path of  $\phi$  is the lower horizontal line.

the only two nonzero symbols in  $\mathbf{d}$ . From (5), the state variables at time  $k \geq \hat{k} = \max(k', k'')$  will always be

$$\mathbf{q}_k = \mathbf{q}_{\hat{k}} = \mathbf{d}_{k'} \mathbf{G}_{k'} + \mathbf{d}_{k''} \mathbf{G}_{k''}, \quad \text{for } k \geq \max(k', k''). \quad (7)$$

If  $\mathbf{q}_{\hat{k}} \neq \mathbf{0}$ , (7) will result in a large Hamming weight with a high probability, which is the preferred situation. However, it is usually impossible to ensure  $\mathbf{q}_{\hat{k}} \neq \mathbf{0}$  for arbitrary (nonzero)  $\mathbf{d}_{k'}$  and  $\mathbf{d}_{k''}$ ,  $\mathbf{G}_{k'}$  and  $\mathbf{G}_{k''}$ . We thus treat this as a probability event below.

*Definition 2:* The *pairwise remerging probability*, denoted by  $\gamma_{pw}$ , is the occurrence probability of the following event:

$$\mathbf{d}_{k'} \mathbf{G}_{k'} + \mathbf{d}_{k''} \mathbf{G}_{k''} = \mathbf{0} \quad (8)$$

over all possible values of  $\mathbf{d}_{k'} (\neq \mathbf{0})$ ,  $\mathbf{d}_{k''} (\neq \mathbf{0})$ ,  $\mathbf{G}_{k'}$ , and  $\mathbf{G}_{k''}$ .

For example, Fig. 4 shows a remerging event for a code with four component codes.

Consider the calculation of  $\gamma_{pw}$ . Assume that every  $\mathbf{d}_k$  takes values independently and with equal probability from an input alphabet of size  $2^n$ . Let  $S(\mathbf{G}_k)$  be the vector space over  $\text{GF}(2)$  spanned by  $\mathbf{G}_k$ . The size of  $S(\mathbf{G}_k)$  is  $2^{\dim S(\mathbf{G}_k)}$ . Denote  $V = S(\mathbf{G}_{k'}) \cap S(\mathbf{G}_{k''})$ . Then

$$\Pr\{\mathbf{d}_{k'} \mathbf{G}_{k'} \in V\} = \frac{2^{\dim V}}{2^{\dim S(\mathbf{G}_{k'})}}$$

and

$$\Pr\{\mathbf{d}_{k''} \mathbf{G}_{k''} \in V\} = \frac{2^{\dim V}}{2^{\dim S(\mathbf{G}_{k''})}}.$$

In  $\text{GF}(2)$ , (8) is equivalent to  $\mathbf{d}_{k'} \mathbf{G}_{k'} = \mathbf{d}_{k''} \mathbf{G}_{k''}$ . Note that

$$\Pr\{\mathbf{d}_{k'} \mathbf{G}_{k'} = \mathbf{d}_{k''} \mathbf{G}_{k''} \mid \mathbf{d}_{k'} \mathbf{G}_{k'} \in V, \mathbf{d}_{k''} \mathbf{G}_{k''} \in V\} = \frac{1}{2^{\dim V}}.$$

With the early assumption that  $\mathbf{d}_{k'}$  and  $\mathbf{d}_{k''}$  are independent of each other, and using Bayes' rule, we have

$$\begin{aligned} & \Pr\{\mathbf{d}_{k'}\mathbf{G}_{k'} = \mathbf{d}_{k''}\mathbf{G}_{k''}\} \\ &= \Pr\{\mathbf{d}_{k'}\mathbf{G}_{k'} = \mathbf{d}_{k''}\mathbf{G}_{k''} | \mathbf{d}_{k'}\mathbf{G}_{k'} \in V, \mathbf{d}_{k''}\mathbf{G}_{k''} \in V\} \\ & \quad \cdot \Pr\{\mathbf{d}_{k'}\mathbf{G}_{k'} \in V\} \Pr\{\mathbf{d}_{k''}\mathbf{G}_{k''} \in V\} \\ &= \frac{2^{\dim V}}{2^{\dim S(\mathbf{G}_{k'})} \times 2^{\dim S(\mathbf{G}_{k''})}}. \end{aligned}$$

This, together with the condition in (8) that  $\mathbf{d}_{k'} \neq \mathbf{0}$  and  $\mathbf{d}_{k''} \neq \mathbf{0}$ , leads to the following remark.

*Remark 2:* Given  $\mathbf{G}_{k'}$  and  $\mathbf{G}_{k''}$ , for randomly selected  $\mathbf{d}_{k'} \neq \mathbf{0}$  and  $\mathbf{d}_{k''} \neq \mathbf{0}$

$$\begin{aligned} & \Pr\{\mathbf{d}_{k'}\mathbf{G}_{k'} + \mathbf{d}_{k''}\mathbf{G}_{k''} = \mathbf{0}\} \\ &= \frac{2^{\dim(S(\mathbf{G}_{k'}) \cap S(\mathbf{G}_{k''}))} - 1}{(2^{\dim S(\mathbf{G}_{k'})} - 1)(2^{\dim S(\mathbf{G}_{k''})} - 1)}. \quad (9) \end{aligned}$$

In general,  $\gamma_{pw}$  can be calculated by averaging the conditional probability in (9) over all possible  $(\mathbf{G}_{k'}, \mathbf{G}_{k''})$  pairs in the code.

#### D. Time-Varying Encoder Structure

Following the preceding discussions, we have a useful design rule.

**Rule 1:**  $\lambda_{\min}$  should be maximized, and  $\gamma_{pw}$  should be minimized.

From Remark 2,  $\gamma_{pw}$  can be made small by reducing the intersection between the vector spaces spanned by different  $\mathbf{G}_k$ . This implies that a good CT-TCM code should be *time varying* (i.e., it should have different  $\mathbf{G}_k$  for different  $k$ ). However, except for very short codes, it is inevitable that one must use repeated  $\mathbf{G}_k$  due to limited choices.

In this paper, we consider the following *periodically time-varying* code structure. We select an initial subset<sup>1</sup>

$$\mathbf{A} = \{\mathbf{G}_k, k = 0, 1, \dots, M-1\}$$

from all possible  $\{\mathbf{G}_k\}$ . Repeating the subset  $\mathbf{A}$  for  $k \geq M$  generates the overall code. The selections of  $\mathbf{G}_k$  in  $\mathbf{A}$  are based on Rule 1. When the two requirements in Rule 1 cannot be satisfied simultaneously, we make a tradeoff between them by considering the diverging length (to be defined in Section III-E). A useful property is as follows.

*Remark 3:* Using the periodically time-varying structure and modulo- $M$  interleavers defined in (3), we always have  $\mathbf{g}_k^{(m)} = \mathbf{g}_{\pi^{(m)}(k)}^{(m)}$ ,  $\forall k$ . Thus, the encoding method for every  $\mathbf{d}_k$  is the same with and without interleaving, and so  $\lambda_{\min}$  and  $\gamma_{pw}$  are not affected by modulo- $M$  interleaving.

When all  $\{\mathbf{G}_k\}$  contain linearly independent rows, we have  $\dim S(\mathbf{G}_k) = n$ ,  $\forall k$ . With the above periodically time-varying structure, the average pairwise remerging probability can be calculated from (9) as follows. (*Note:*  $\mathbf{G}_{k'}$  and  $\mathbf{G}_{k''}$  can take any values in the initial set  $\mathbf{A}$ .)

$$\gamma_{pw} = \frac{1}{M \times M} \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} \frac{2^{\dim(S(\mathbf{G}_i) \cap S(\mathbf{G}_j))} - 1}{(2^n - 1)(2^n - 1)}. \quad (10)$$

<sup>1</sup>The size of the initial set  $\mathbf{A}$  can be different from  $M$ , but we will not consider it in this paper.

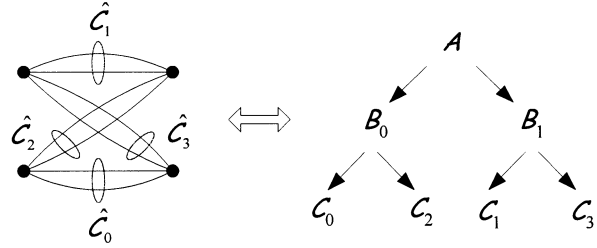


Fig. 5. Assignment of signal point sets to parallel branch sets.

#### E. The Impact of Interleavers

From Remark 3, we can see that the modulo- $M$  interleavers  $\{\pi^{(m)}(k), 0 \leq m < M\}$  defined in (3) will not affect  $\lambda_{\min}$  and  $\gamma_{pw}$ . However, they will have an important impact on the diverging length, as discussed below. Consider again a codeword  $\mathbf{c}$  with  $IW(\mathbf{c}) = 2$  generated by  $\mathbf{d}$ . Again let  $\mathbf{d}_{k'} \neq \mathbf{0}$  and  $\mathbf{d}_{k''} \neq \mathbf{0}$  in  $\mathbf{d}$ , and  $\mathbf{d}_{k'}\mathbf{G}_{k'} + \mathbf{d}_{k''}\mathbf{G}_{k''} = \mathbf{0}$ . After interleaving, at least  $\lambda_{\min}$  component encoding paths of  $\mathbf{c}$  diverge from  $\phi$  between sections  $\pi^{(m)}(k')$  and  $\pi^{(m)}(k'')$ . If the  $m$ th component encoding path diverges from  $\phi$ , the diverging part covers  $\rho^{(m)} = |\pi^{(m)}(k'') - \pi^{(m)}(k')| + 1$  sections for which  $q_k^{(m)} = 1$ , contributing a Hamming weight of  $\rho^{(m)}$ . Hence,

$$W_H(\mathbf{c}) = \sum_{\substack{m=0 \\ p^{(m)}=1}}^{M-1} \rho^{(m)} \quad (11)$$

where  $p^{(m)} = \mathbf{d}_{k'}\mathbf{g}_{\pi^{(m)}(k')}^{(m)}$ . We refer to  $\rho^{(m)}$  as the *diverging length* of the  $m$ th component code. When interleaving is trivial, i.e.,  $\{\pi^{(m)}(k) = k, \forall k\}$  for all  $m$ , (11) reduces to

$$W_H(\mathbf{c}) = (|k'' - k'| + 1) \cdot \text{Div}(\mathbf{d}_{k'}, \mathbf{G}_{k'}). \quad (12)$$

In this case, the minimum value of the diverging length is the dominant factor.

On the other hand, with random interleavers, the  $\text{Div}(\mathbf{d}_k, \mathbf{G}_k)$  nonzero values in  $\{\rho^{(m)}, 0 \leq m < M\}$  can be regarded as independent random variables uniformly distributed in  $[2, L]$ , where  $L$  is the interleaver size. The probability that  $W_H(\mathbf{c})$  is very small reduces rapidly as  $\text{Div}(\mathbf{d}_{k'}, \mathbf{G}_{k'})$  or  $L$  increases. (We see here again the advantage of maximizing  $\lambda_{\min}$ .) This effect is similar to the interleaving gain discussed for turbo codes in [18] and it also applies to codewords with  $IW(\mathbf{c}) > 2$ .

#### IV. MAPPING RULES

We now consider the design of the signal mapper in Fig. 1. Following Ungerboeck's principle of mapping by set partitioning [19], we partition the original constellation  $\mathbf{A}$  of size  $2^{n+1}$  into four subsets:  $(\mathbf{C}_0, \mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3)$ . These subsets are assigned to the four sets of parallel branches in a two-state trellis section, as shown in Fig. 5 (i.e., the signal points in  $\mathbf{C}_i$  are assigned to the branches in  $\hat{\mathbf{C}}_i$ ). The Euclidean distance between the parallel transitions is maximized in this way.

To further partition  $\mathbf{C}_i$ , we consider the relative Euclidean distance between each pair of codewords. At the starting and ending positions of a diverging span, the two codewords always have different input symbols. Large Euclidean distances should be assigned to the coded symbol pairs corresponding to small

relative divergence degrees at these two positions since the interleaving gain is relatively weak in this case. A good balance between Euclidean distance (related to signal points) and interleaving gain (related to relative divergence degree) can be achieved using the following Rule 2. The application of Rule 2 is explained in the 16-QAM design example in Section V.

**Rule 2:** A larger Euclidean distance should be assigned to a coded symbol pair in  $\hat{\mathcal{C}}_i$  possessing a smaller relative divergence degree.

Specifically, we continue to partition  $\mathcal{C}_i$  into  $\mathcal{D}_{2i}$  and  $\mathcal{D}_{2i+1}$ . Both  $\mathcal{D}_{2i}$  and  $\mathcal{D}_{2i+1}$  have larger minimum intrasubset Euclidean distance than  $\mathcal{C}_i$ . We partition  $\hat{\mathcal{C}}_i$  into  $\hat{\mathcal{D}}_{2i}$  and  $\hat{\mathcal{D}}_{2i+1}$ , having smaller intrasubset relative divergence degree. We assign  $\mathcal{D}_{2i}$  and  $\mathcal{D}_{2i+1}$  to  $\hat{\mathcal{D}}_{2i}$  and  $\hat{\mathcal{D}}_{2i+1}$ , respectively. In this way, Rule 2 is satisfied. This process can be continued if necessary.

## V. EXAMPLES AND NUMERICAL RESULTS

In this section, some design examples of CT-TCM codes are provided, which involve handcrafted tradeoff among divergence degrees, pairwise remerging probabilities, diverging lengths, and constellation mapping. Good performances will be demonstrated. In all simulations, we assume additive white Gaussian noise (AWGN) channels with zero mean and variance  $N_0/2$  per dimension. Pseudorandom modulo- $M$  interleaving and puncturing are always used.

### A. CT-TCM Codes for 8-PSK

Consider a CT-TCM code for 8-PSK modulation with a spectral efficiency of 2 bits/symbol. In this case,  $n = 2$ . Let  $M = 4$ . We adopt the initial subset  $\{\mathbf{G}_k, k = 0, 1, 2, 3\}$  as follows:

$$\mathbf{A} = \left\{ \underbrace{\begin{pmatrix} \boxed{1} & 0 & 1 & 1 \\ \boxed{1} & 1 & 1 & 0 \end{pmatrix}}_{\mathbf{G}_0}, \underbrace{\begin{pmatrix} 0 & \boxed{1} & 1 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix}}_{\mathbf{G}_1}, \right. \\ \left. \underbrace{\begin{pmatrix} 1 & 1 & \boxed{1} & 0 \\ 1 & 1 & \boxed{0} & 1 \end{pmatrix}}_{\mathbf{G}_2}, \underbrace{\begin{pmatrix} 1 & 1 & 0 & \boxed{1} \\ 0 & 1 & 1 & \boxed{1} \end{pmatrix}}_{\mathbf{G}_3} \right\}. \quad (13)$$

With periodic repetition of  $\mathbf{A}$ , an asymmetric time-varying (ATV) CT-TCM code is obtained. This code has  $\lambda_{\min} = 2$  and

$$\gamma_{pw} = \frac{2^n - 1}{(2^n - 1)^2} \times \frac{4}{16} + \frac{1}{(2^n - 1)^2} \times \frac{8}{16} = \frac{5}{36}.$$

When the reference codeword is  $\phi$ ,  $\lambda_{\min} = 2$  is related to only one type of codeword with  $IW(\mathbf{c}) = 2$ , i.e., the two nonzero information symbols are  $\mathbf{d}_k = \mathbf{d}_{k+4j} = (11)$ , with  $j$  any nonzero integer. The diverging lengths of such codewords are at least 4, so they have relatively large Hamming weights. All other codewords have a divergence degree of 3. Only two types of sections (the circled ones in (13)) are unpunctured. The mapping rules for these unpunctured sections are shown in Fig. 6.

The performance of this code is shown in Fig. 7 (labeled by CT-TCM ATV). The results of [3]–[5] are also plotted in Fig. 7 for comparison. It can be seen that the proposed CT-TCM code

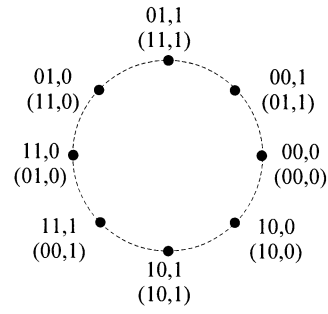


Fig. 6. Mapping of the CT-TCM code for 8-PSK. The signal points are labeled by the values of  $(\mathbf{d}_k, \mathbf{q}_k)$ . The labels on top are for  $\mathbf{g}_k = [1 \ 1]^T$  and those in brackets are for  $\mathbf{g}_k = [1 \ 0]^T$ .

has comparable performance to the codes in [3]–[5].<sup>2</sup> However, as analyzed in the Appendix, the complexity of this CT-TCM code is about six times lower than the code in [3] and about twelve times lower than those in [4], [5].

To illustrate the impact of the asymmetric and time-varying principles, the bit-error rate (BER) curves for different 8-PSK CT-TCM codes (with  $M = 4$ ) using best effort searching are also compared in Fig. 7, and it is seen that the ATV structure yields a noticeable improvement. In addition to the ATV code defined earlier, other codes are subsequently defined. Since  $n = 2$ , the nonzero columns in  $\mathbf{G}_k$  are selected from

$$\mathbf{a} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \text{and} \quad \mathbf{c} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

- For the symmetric time-invariant (STI) code

$$\mathbf{G}_k = [\mathbf{a} \ \mathbf{a} \ \mathbf{a} \ \mathbf{a}], \quad \text{for all } k$$

$$\lambda_{\min} = 0, \text{ and } \gamma_{pw} = 1/3.$$

- For the symmetric time-varying (STV) code,  $\lambda_{\min} = 0$  and  $\gamma_{pw} = \frac{1}{6}$ , and

$$\mathbf{A} = \left\{ \mathbf{G}_0 = [\mathbf{a} \ \mathbf{a} \ \mathbf{a} \ \mathbf{a}], \mathbf{G}_1 = [\mathbf{c} \ \mathbf{c} \ \mathbf{c} \ \mathbf{c}], \right.$$

$$\left. \mathbf{G}_2 = [\mathbf{a} \ \mathbf{a} \ \mathbf{a} \ \mathbf{a}], \mathbf{G}_3 = [\mathbf{b} \ \mathbf{b} \ \mathbf{b} \ \mathbf{b}] \right\}.$$

- For the asymmetric time-invariant (ATI) code

$$\mathbf{G}_k = [\mathbf{a} \ \mathbf{c} \ \mathbf{a} \ \mathbf{b}], \quad \text{for all } k$$

$$\lambda_{\min} = 2, \text{ and } \gamma_{pw} = 1/3.$$

### B. CT-TCM Codes for 16-QAM

Consider 16-QAM CT-TCM schemes with  $n = 3$  and  $M = 4$ . Each  $\mathbf{G}_k$  is a  $3 \times 4$  matrix over GF(2). The maximum achievable  $\lambda_{\min}$  is 2 using the generator matrix of an SPC code, e.g.,

$$\mathbf{G}_{\text{SPC}} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix}.$$

For a fixed length, the SPC code is unique. If we want to make  $\lambda_{\min} = 2$ , the corresponding  $\gamma_{pw} = 1/7$  is relatively small since  $S(\mathbf{G}_k) = S(\mathbf{G}_{\text{SPC}})$ ,  $\forall k$ . Such a code can be realized using an ATI structure. To increase  $\gamma_{pw}$ , we can construct an ATV structure using the  $\mathbf{G}_0$  in (14), which is obtained by

<sup>2</sup>A reviewer pointed out that the performance of the scheme in [3] might be improved with an optimized interleaver, and a bit-interleaved coded modulation scheme using LDPC codes can achieve a similar performance as that shown in Fig. 7 with a similar complexity.

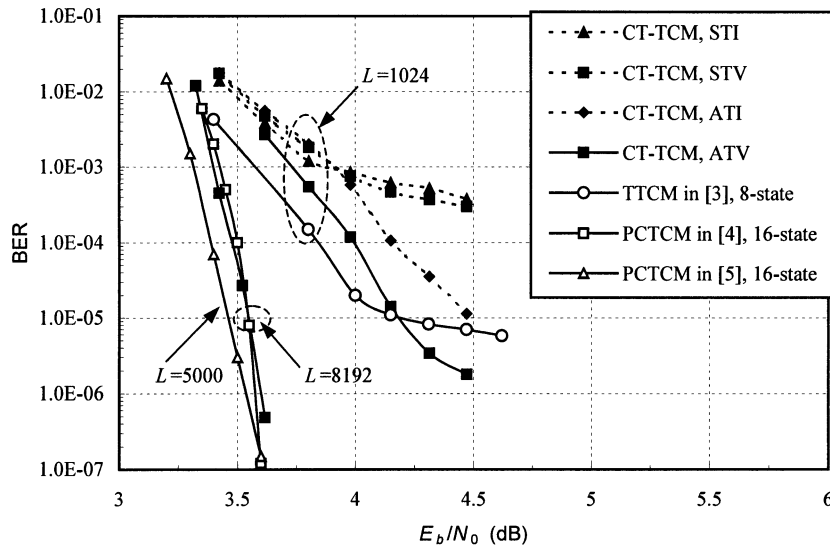


Fig. 7. BER performance of different 8-PSK CT-TCM schemes. “ $L$ ” represents interleaver size (in symbols). Spectral efficiency = 2 bits/symbol. Constrained-capacity = 2.9 dB [20]. All the results are produced using 18 iterations except those cited from [4] (8 iterations) and [5] (15 iterations).

reversing a column in  $\mathbf{G}_{\text{SPC}}$ . Then  $\lambda_0 = 1$ . The initial set  $\{\mathbf{G}_k, k = 0, 1, 2, 3\}$  is given below, where  $\mathbf{G}_1$ ,  $\mathbf{G}_2$ , and  $\mathbf{G}_3$  are obtained by cyclic shifting of  $\mathbf{G}_0$  so that  $\lambda_{\min} = \lambda_k = 1$ ,  $0 \leq k \leq 3$

$$\mathbf{A} = \left\{ \underbrace{\begin{pmatrix} \boxed{1} & 1 & 1 & 0 \\ \boxed{1} & 0 & 1 & 1 \\ \boxed{0} & 1 & 1 & 1 \end{pmatrix}}_{\mathbf{G}_0}, \underbrace{\begin{pmatrix} 0 & \boxed{1} & 1 & 1 \\ 1 & \boxed{1} & 0 & 1 \\ 1 & \boxed{0} & 1 & 1 \end{pmatrix}}_{\mathbf{G}_1}, \right. \\ \left. \underbrace{\begin{pmatrix} 1 & 0 & \boxed{1} & 1 \\ 1 & 1 & \boxed{1} & 0 \\ 1 & 1 & \boxed{0} & 1 \end{pmatrix}}_{\mathbf{G}_2}, \underbrace{\begin{pmatrix} 1 & 1 & 0 & \boxed{1} \\ 0 & 1 & 1 & \boxed{1} \\ 1 & 1 & 1 & \boxed{0} \end{pmatrix}}_{\mathbf{G}_3} \right\}. \quad (14)$$

For this code,

$$\gamma_{pw} = \frac{3}{4} \times \frac{3}{7^2} + \frac{1}{4} \times \frac{7}{7^2} = 4/49.$$

We consider the application of Rule 2 in Section IV. It can be verified that the only input symbol with  $\lambda_{\min} = 1$  (relative to  $\phi$ ) is  $\mathbf{d}_k = (111)$ , which produces  $\mathbf{c}_k = (111, 0)$ . When  $IW(\mathbf{c}) = 2$  and  $\mathbf{d}_k = (111)$ , a remerging event takes place only when  $\mathbf{d}_{k+4j} = (111)$ , with  $j$  any nonzero integer. The related diverging length is  $\rho^{(m)} \geq 4$ ,  $0 \leq m < M$ . Other codewords have divergence degrees of at least 2. We adopt the mapping in Fig. 8 for the unpunctured section with  $\mathbf{g}_k = [1 \ 1 \ 0]^T$  (the circled ones in (14)). A large Euclidean distance is assigned to every pair of coded symbols  $\mathbf{c}_k$  and  $\tilde{\mathbf{c}}_k$  with a relative divergence degree of 1, i.e.,  $\mathbf{c}_k - \tilde{\mathbf{c}}_k = (111, 0)$  (such as  $(100, 1)$  and  $(011, 1)$ , etc.).

This ATV CT-TCM code is compared in Fig. 9 (marked by “A”) with the ATI CT-TCM code (based on  $\mathbf{G}_{\text{SPC}}$  and marked by “B”) and the eight-state 16-QAM scheme from [3] and the four-state (for both inner and outer codes) 16-QAM scheme from [6]. The ATV code shows improved performance over the ATI one. The ATV CT-TCM scheme has a complexity similar

$$\begin{array}{cccc} \blacksquare & \bullet & \blacksquare & \bullet \\ 101,0 & 001,0 & 011,0 & 111,0 \\ \circ & \square & \circ & \square \\ 011,1 & 111,1 & 101,1 & 001,1 \\ \blacksquare & \bullet & \blacksquare & \bullet \\ 100,0 & 000,0 & 010,0 & 110,0 \\ \circ & \square & \circ & \square \\ 010,1 & 110,1 & 100,1 & 000,1 \end{array}$$

Fig. 8. Mapping of the CT-TCM code for 16-QAM. Four subsets of signal points are distinguished by different marks. They form  $\{\mathbf{C}_0, \mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3\}$  as discussed in Section IV.

to the scheme in [6] (lower than that of [3]). Overall, the ATV CT-TCM code represents a good compromise between complexity and performance.

### C. A CT-TCM Code for 32-QAM

Next we consider a 32-QAM ATV CT-TCM scheme with  $n = 4$  and  $M = 5$ . The initial set  $\{\mathbf{G}_k, k = 0, 1, 2, 3, 4\}$  is

$$\mathbf{A} = \left\{ \underbrace{\begin{pmatrix} \boxed{1} & 1 & 1 & 1 & 1 \\ \boxed{1} & 1 & 1 & 0 & 0 \\ \boxed{1} & 1 & 0 & 1 & 0 \\ \boxed{0} & 1 & 1 & 1 & 0 \end{pmatrix}}_{\mathbf{G}_0}, \underbrace{\begin{pmatrix} 1 & \boxed{1} & 1 & 1 & 1 \\ 0 & \boxed{1} & 1 & 1 & 0 \\ 0 & \boxed{1} & 1 & 0 & 1 \\ 0 & \boxed{0} & 1 & 1 & 1 \end{pmatrix}}_{\mathbf{G}_1}, \right. \\ \underbrace{\begin{pmatrix} 1 & 1 & \boxed{1} & 1 & 1 \\ 0 & 0 & \boxed{1} & 1 & 1 \\ 1 & 0 & \boxed{1} & 1 & 0 \\ 1 & 0 & \boxed{0} & 1 & 1 \end{pmatrix}}_{\mathbf{G}_2}, \underbrace{\begin{pmatrix} 1 & 1 & 1 & \boxed{1} & 1 \\ 1 & 0 & 0 & \boxed{1} & 1 \\ 0 & 1 & 0 & \boxed{1} & 1 \\ 1 & 1 & 0 & \boxed{0} & 1 \end{pmatrix}}_{\mathbf{G}_3}, \\ \left. \underbrace{\begin{pmatrix} 1 & 1 & 1 & 1 & \boxed{1} \\ 1 & 1 & 0 & 0 & \boxed{1} \\ 1 & 0 & 1 & 0 & \boxed{1} \\ 1 & 1 & 1 & 0 & \boxed{0} \end{pmatrix}}_{\mathbf{G}_4} \right\}. \quad (15)$$

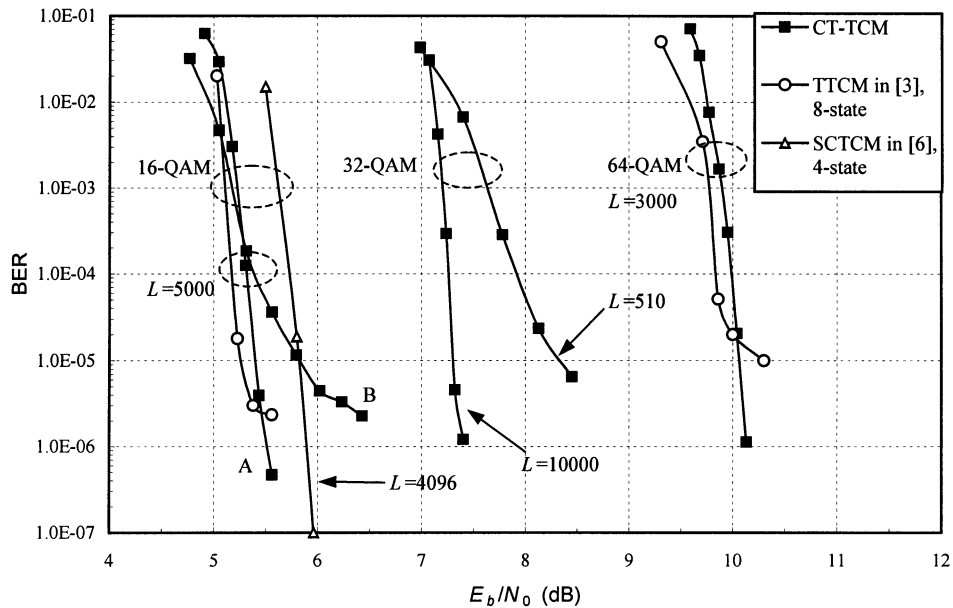


Fig. 9. BER performance of CT-TCM schemes for different modulations. “ $L$ ” represents interleaver size (in symbols). The spectral efficiencies for 16-QAM, 32-QAM, and 64-QAM are 3, 4, and 5 bits/symbol, respectively. The corresponding constrained-capacity limits are 4.5, 6.8, and 9.2 dB, respectively [20]. All the results are produced using 18 iterations except those cited from [6] (8 iterations).

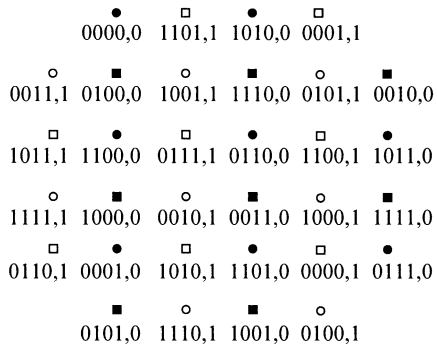


Fig. 10. Mapping of the CT-TCM code for 32-QAM.

which is obtained, in a similar fashion to the 16-QAM design, by reversing a column in the generator matrix of a length-5 SPC code. It has  $\lambda_{\min} = 1$  and

$$\gamma_{pw} = \frac{4}{5} \times \frac{3}{15^2} + \frac{1}{5} \times \frac{15}{15^2} = 27/1125.$$

Using  $\phi$  as the reference,  $\lambda_{\min} = 1$  only for one type of input pattern with  $\mathbf{d}_k = \mathbf{d}_{k+5j} = (0111)$ , and all other codewords have a divergence degree of at least 2. The mapping is given in Fig. 10 for the unpunctured section with  $\mathbf{g}_k = [1 \ 1 \ 1 \ 0]^T$  (the circled ones in (15)). See Fig. 9 for performance.

#### D. CT-TCM Codes for Higher Order Constellations

Following [3], for a larger signal constellation, the operating signal-to-noise ratio (SNR) is usually very high. At a certain point of the set-partitioning chain, the intrasubset Euclidean distances may be sufficient to guarantee a very small error rate. In this situation, given a received symbol  $y_k$ , the probability that  $x_k$  is a constellation point outside a preset distance threshold from  $y_k$  is very small and we will simply ignore this possibility.

Equivalently, we expurgate the corresponding branches in the decoding trellis; thus, the decoding cost is greatly reduced.

For a code with a constellation of size  $2^{n+1}$ , a simple implementation of the above principle is to encode only  $\tilde{n}$  bits using an appropriate two-state trellis code and leave  $(n - \tilde{n})$  bits uncoded for each information symbol. The coded bits are used to define the signal subsets and the uncoded bits are used to select signal points from a subset. The value of  $\tilde{n}$  is determined according to the operating SNR and the Euclidean distances between  $2^{n-\tilde{n}}$  signal points within a subset.

Consider a CT-TCM code for 64-QAM with 5 bits/symbol. In this case, the channel capacity is  $E_s/N_0 = 16.2$  dB. According to [3], the intrasubset error probability is  $P_b \leq 10^{-6}$  after three levels of Ungerboeck-type set partitioning. As a result, we adopt  $\tilde{n} = 3$  and employ the previous design for 16-QAM in (14). The performance of this code is shown in Fig. 9.

## VI. CONCLUSION

In this paper, we proposed a family of concatenated two-state TCM codes using symbol interleavers. The proposed codes are characterized by asymmetrical and time-varying trellis structures. A joint design strategy considering all component codes is established. Compared with existing turbo-type coded modulation schemes, the proposed codes have significantly reduced complexity without compromising performance.

For future work, a general analysis of the CT-TCM schemes is necessary, but it is a complicated task involving specific puncturing patterns, mapping rules, and interleaver design. Therefore, the existing methods (such as [21]) for performance analysis of turbo-TCM codes are not directly applicable here. Research in this direction offers interesting prospects. The codes given in Section V are mostly handcrafted and are not optimized. A systematic design strategy or an optimization procedure may offer another interesting avenue for future work.

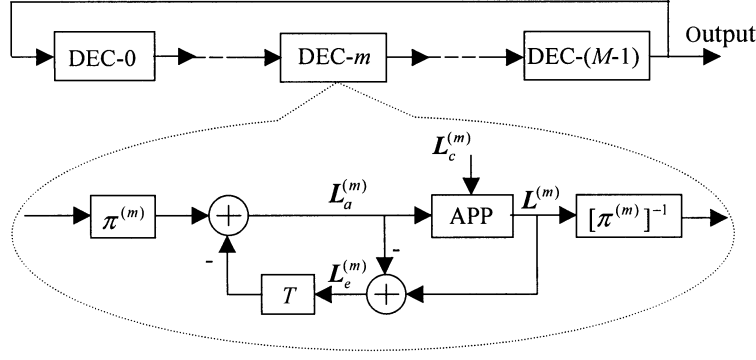


Fig. 11. The global decoder: “ $T$ ” for delay of one iteration and “ $\pi$ ” for interleaving.

## APPENDIX ITERATIVE DECODER AND ITS COMPLEXITY

### A. Local Decoder Based on the BCJR Algorithm

Consider a general trellis code. Let  $\mathbf{d} = \{\mathbf{d}_k\}$  be the information symbol sequence. It is encoded into a modulated symbol sequence  $\mathbf{x} = \{x_k\}$ . Let  $\mathbf{y} = \{y_k\}$  be the received sequence, where  $y_k$  is the noisy observation of  $x_k$  if  $x_k$  is not punctured; otherwise, set  $y_k = *$  (implying no observation obtained).

Let  $s_k$  be the encoder state at time  $k$ . (Note: For CT-TCM,  $s_k = q_k$ , see Fig. 2.) A branch at the  $k$ th section in the corresponding trellis diagram can be specified by  $b \equiv (s_{k-1}, \mathbf{d}_k, x_k, s_k)$ , where  $\mathbf{d}_k$  and  $x_k$  are the information and modulated symbols associated with the state transition  $s_{k-1} \rightarrow s_k$ . The branch metric of  $b$  is defined and calculated as

$$\begin{aligned} \delta(b) &\equiv p(\mathbf{d}_k, y_k, s_k | s_{k-1}) \\ &= \begin{cases} P(\mathbf{d}_k), & \text{if } y_k = * \\ p(y_k | x_k) \cdot P(\mathbf{d}_k), & \text{otherwise.} \end{cases} \end{aligned}$$

Denote by  $B(s_{k-1}, s_k)$  the set of all the parallel branches connecting  $s_{k-1}$  and  $s_k$ . The BCJR algorithm is summarized as follows [22]–[24]:

#### Combining branch metrics :

$$\gamma(s_{k-1}, s_k) = \sum_{b \in B(s_{k-1}, s_k)} \delta(b). \quad (\text{A1})$$

#### Forward recursion :

$$\alpha(s_k) = \sum_{s_{k-1}} \gamma(s_{k-1}, s_k) \alpha(s_{k-1}). \quad (\text{A2})$$

#### Backward recursion :

$$\beta(s_{k-1}) = \sum_{s_k} \gamma(s_{k-1}, s_k) \beta(s_k). \quad (\text{A3})$$

#### Output :

$$P(\mathbf{d}_k = i | \mathbf{y}) \propto \sum_{b: \mathbf{d}_k = i} \delta(b) \alpha(s_{k-1}) \beta(s_k). \quad (\text{A4})$$

### B. Global Decoder

The global decoder operating in the log domain for a CT-TCM code is shown in Fig. 11. It consists of  $M$  local *a posteriori* probability (APP) decoders, one for each component code. The variables involved in Fig. 11 are the log-likelihood (LL) values, as detailed as follows.

$\mathbf{L}^{(m)}$  The *a posteriori* LL values for all information symbols after decoding the  $m$ th component code.

$\mathbf{L}_c^{(m)}$  The LL values  $\{\log p(y_k | x_k)\}$  based on individual channel observations of the  $m$ th component code. Its elements  $\{L_{c,k}(m)\}$  are calculated as

$$L_{c,k}(m) = \begin{cases} \log p(y_k | x_k), & \text{for unpunctured symbols} \\ 0, & \text{for punctured symbols.} \end{cases} \quad (\text{A5})$$

$\mathbf{L}_a^{(m)}$  The *a priori* LL values for all information symbols for the  $m$ th component code. It is initialized to zeroes, implying no *a priori* information.

$\mathbf{L}_e^{(m)}$  The extrinsic information produced by the  $m$ th component code, defined by

$$\mathbf{L}_e^{(m)} = \mathbf{L}^{(m)} - \mathbf{L}_a^{(m)}.$$

The  $M$  local decoders operate successively.  $\mathbf{L}_a^{(m)}$  contains the accumulated extrinsic information generated by all the local decoders except DEC- $m$ ,

$$\begin{aligned} \mathbf{L}_a^{(m)} &= \underbrace{\mathbf{L}_a^{(m+1)} + \mathbf{L}_a^{(m+2)} + \dots + \mathbf{L}_a^{(M-1)}}_{\text{from the previous iteration}} \\ &\quad + \underbrace{\mathbf{L}_e^{(0)} + \mathbf{L}_e^{(1)} + \dots + \mathbf{L}_e^{(m-1)}}_{\text{from the current iteration}}. \end{aligned} \quad (\text{A6})$$

It is used together with channel observations  $\mathbf{L}_c^{(m)}$  in the next APP decoding of the  $m$ th component code. A discussion on this global decoder can also be found in [16].

### C. Complexity Analysis

For convenience, we count costs in the probability domain. Let  $S$  be the number of trellis states and  $n$  the number of information bits in an input symbol. For a BCJR decoder, normalizing one of  $S$  state metrics to unity can reduce the decoding cost. This is particularly beneficial when  $S = 2$ . Let us normalize  $\alpha(s_k = 0)$  and  $\beta(s_k = 0)$  to 1 for every  $k$ . The multiplications associated with  $\alpha(s_{k-1} = 0)$  and  $\beta(s_k = 0)$  in (A2)–(A4) can be eliminated. For the output stage, we first find  $\alpha(s_{k-1} = 1)\beta(s_k = 1)$  and then multiply it by every associated branch metric. In this way, the total number of multiplications required in (A4) is  $2^{n+1} \times \frac{3}{4} + 1$ .

For (A1), additions are needed for the decoder of a two-state code to combine individual branch metrics. The addition

TABLE I  
THE COMPUTATIONAL COST OF THE BCJR ALGORITHM FOR TWO-STATE AND  $S$ -STATE TRELLIS CODES (UNIT: OPERATIONS PER TRELLIS SECTION PER COMPONENT CODE PER ITERATION)

Operation	Case-I $S$ -state without parallel branches		Case-II 2-state with parallel branches	
	additions	multiplications	additions	multiplications
Branch metric computation	0	$\frac{1}{M} \times 2^{n+1}$	$(2^n - 2) \times (1 + \frac{1}{M})$	$\frac{1}{M} \times 2^{n+1}$
Forward recursion	$(2^n - 1)S + S$	$2^n S + S$	2	3
Backward recursion	$(2^n - 1)S + S$	$2^n S + S$	2	3
Output stage	$2^n S$	$2^n S$	$2^n$	$2^{n+1} \times \frac{3}{4} + 1$
Total (for large $M$ )	$3 \times 2^n S$	$\approx 3 \times 2^n S$	$\approx 2 \times 2^n$	$\approx \left(\frac{3}{2}\right) \times 2^n$

number is  $(2^{n+1} - 4)$  for an unpunctured section, and  $(2^n - 2)$  for a punctured one. (For a punctured section,  $\gamma(0, 0) = \gamma(1, 1)$  and  $\gamma(0, 1) = \gamma(1, 0)$ .) Recall that for the code in Fig. 3, every  $(M - 1)$  trellis sections out of  $M$  trellis sections are punctured. The average addition number is

$$\frac{1}{M} \times (2^{n+1} - 4) + \frac{M - 1}{M} \times (2^n - 2) = \frac{M + 1}{M} \times (2^n - 2).$$

The average multiplication number required in (A1) is  $\frac{1}{M} \times 2^{n+1}$  including the generation of  $\delta(b)$ .

Table I summarizes the decoding costs involved in (A1)–(A4) for a two-state and an  $S$ -state ( $S \geq 4$ ) code, respectively, with normalization costs included. Note that the cost saving due to normalization also applies to trellises with  $S \geq 4$ , but the benefit becomes marginal as  $S$  increases. For simplicity, such saving is only considered for Case-II in Table I. (*Note*: normalization is always necessary to prevent overflow.) From Table I, the decoding cost of a two-state trellis code is about  $1.2S$  times lower than that of an  $S$ -state ( $S \geq 4$ ) one without parallel branches. This ratio should be adjusted considering  $M$ . For example, the complexity of the CT-TCM code defined in (13) with  $M = 4$  is about six times lower than that of the eight-state,  $M = 2$  code in [3], and twelve times lower than that of the 16-state,  $M = 2$  codes in [4], [5].

The preceding discussion is in the probability domain. In practice, all of the operations can be carried out in the log domain. We store all variables using their log values. For  $a \times b$ , we actually evaluate  $\log(a) + \log(b)$  and for  $a + b$ , we actually evaluate  $\max(\log(a), \log(b)) + \exp(1 + |\log(a) - \log(b)|)$ . In this way, there is no conversion between log and probability values since  $\exp(1 + |\log(a) - \log(b)|)$  can be implemented using a lookup table. With some modifications, Table I can still be used for comparison purposes.

#### ACKNOWLEDGMENT

The authors wish to thank Dr. G. D. Forney for helpful suggestions and encouraging comments. They would also like to acknowledge the reviewers for their constructive advices and valuable comments. Help from Dr. X. Ma, W. K. Leung, and K. Y. Wu is gratefully appreciated.

#### REFERENCES

- [1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit error-correcting coding and decoding: Turbo-codes," in *Proc. 1993 IEEE Int. Conf. Commun.*, Geneva, Switzerland, May 1993, pp. 1064–1070.
- [2] S. L. Goff, A. Glavieux, and C. Berrou, "Turbo-codes and high spectral efficiency modulation," in *Proc. IEEE Int. Communications Conf. (ICC'94)*, 1994, pp. 645–649.
- [3] P. Robertson and T. Worz, "Bandwidth-efficient turbo trellis-coded modulation using punctured component codes," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 206–218, Feb. 1998.
- [4] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Parallel concatenated trellis coded modulation," in *Proc. IEEE Int. Communications Conf. (ICC'96)*, 1996, pp. 974–978.
- [5] C. Fragouli and R. D. Wesel, "Turbo-encoder design for symbol-interleaved parallel concatenated trellis-coded modulation," *IEEE Trans. Commun.*, vol. 49, pp. 425–435, Mar. 2001.
- [6] D. Divsalar, S. Dolinar, and F. Pollara, "Serial concatenated trellis-coded modulation with rate-1 inner code," in *Proc. 2000 IEEE GLOBECOM*, 2000, pp. 777–782.
- [7] U. Wachsmann, R. F. H. Fischer, and J. B. Huber, "Multilevel codes: Theoretical concepts and practical design rules," *IEEE Trans. Inform. Theory*, vol. 45, pp. 1361–1391, July 1999.
- [8] J. Hou, P. H. Siegel, L. B. Milstein, and H. D. Pfister, "Design of low-density parity-check codes for bandwidth efficient modulation," in *Proc. 2001 IEEE Inform. Theory Workshop*, Cairns, Australia, Sept. 2001, pp. 24–26.
- [9] D. Sridhara and T. E. Fuja, "Bandwidth efficient modulation based on algebraic low density parity check codes," in *Proc. 2001 IEEE Int. Symp. Information Theory*, Washington, DC, June 2001, p. 165.
- [10] K. R. Narayanan and J. Li, "Bandwidth efficient low density parity check coding using multilevel coding," in *Proc. 2nd Int. Symp. Turbo Codes and Related Topics*, Brest, France, Sept. 2000, pp. 165–169.
- [11] D. J. Costello Jr, A. Banerjee, T. E. Fuja, and P. C. Massey, "Some reflections on the design of bandwidth efficient turbo codes," in *Proc. 4th Int. ITG Conf. Source & Channel Coding*, Berlin, Germany, Jan. 2002, pp. 357–363.
- [12] G. D. Forney, Jr and G. Ungerboeck, "Modulation and coding for linear gaussian channels," *IEEE Trans. Inform. Theory*, vol. 44, pp. 2384–2415, Oct. 1998.
- [13] P. C. Massey and D. J. Costello, "New developments in asymmetric turbo codes," in *Proc. 2nd Int. Symp. Turbo Codes & Related Topics*, Brest, France, Sept. 2000, pp. 93–99.
- [14] A. Banerjee, D. J. Costello, Jr, T. E. Fuja, and P. C. Massey, "Asymmetric turbo codes for bit-interleaved coded modulation," in *Proc. 39th Annu. Allerton Conf. Communication, Control, and Computing*, Monticello, IL, Oct. 2001.
- [15] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Trans. Commun.*, vol. 49, pp. 1727–1737, Oct. 2001.
- [16] L. Ping and K. Y. Wu, "Concatenated tree codes: A low complexity, high performance approach," *IEEE Trans. Inform. Theory*, vol. 47, pp. 791–799, Feb. 2001.

- [17] E. Biglieri, D. Divsalar, P. J. McLane, and M. K. Simon, *Introduction to Trellis-Coded Modulation With Applications*. New York: MacMillan, 1991.
- [18] S. Benedetto and G. Montorsi, "Unveiling turbo codes: some results on parallel concatenated decoding schemes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 409–428, Mar. 1996.
- [19] G. Ungerboeck, "Channel coding with multilevel/phase signaling," *IEEE Trans. Inform. Theory*, vol. IT-25, pp. 55–67, Jan. 1982.
- [20] D. J. Costello, J. Hagenauer, H. Imai, and S. B. Wicker, "Applications of error-control coding," *IEEE Trans. Inform. Theory*, vol. 44, pp. 2531–2560, Oct. 1998.
- [21] T. M. Duman and M. Salehi, "Performance bounds for turbo-coded modulation systems," *IEEE Trans. Commun.*, vol. 47, pp. 511–521, Apr. 1999.
- [22] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 284–287, Mar. 1974.
- [23] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 429–445, Mar. 1996.
- [24] X. Ma and A. Kavčić, "Path partitions and forward-only trellis algorithm," *IEEE Trans. Inform. Theory*, vol. 49, pp. 38–52, Jan. 2003.