

Unifying Views of Tail-Biting Trellis Constructions for Linear Block Codes

Aditya V. Nori and Priti Shankar

Abstract—In this paper, we present new ways of describing and constructing linear tail-biting trellises for block codes. We extend the well-known Bahl–Cocke–Jelinek–Raviv (BCJR) construction for conventional trellises to tail-biting trellises. The BCJR-like labeling scheme yields a simple specification for the tail-biting trellis for the dual code, with the dual trellis having the same state-complexity profile as that of the primal code. Finally, we show that the algebraic specification of Forney for state spaces of conventional trellises has a natural extension to tail-biting trellises.

Index Terms—Block codes, displacement matrix, duality, linear trellises, nonmergeable trellises, tail-biting trellises.

I. INTRODUCTION

TAIL-BITING trellises for linear block codes represent the simplest form of *factor graphs* with cycles. These regularly structured graphs are viewed as trellises on a circular time axis and are known to attain a state-complexity that could be considerably lower than that of a conventional trellis for the same code. Conventional trellises for block codes have a well-understood underlying theory [1]–[7] and an excellent survey appears in [7]. The theory of tail-biting trellises appears to be somewhat more involved, though several advances in the understanding of the structure and properties of such trellises have been made in recent years [8]–[14].

Given a block code, it is known that there exists a unique minimal conventional trellis representing the code [6], and there are several different algorithms for the construction of such a trellis [1]–[3], [5]. The trellis simultaneously minimizes all measures of minimality. However, it is known that tail-biting trellises do not have such a property. Koetter and Vardy [9], [10] have made a detailed study of the structure of linear tail-biting trellises and have also defined several measures of minimality. They have shown that different measures of minimality correspond to different partial orders on the set of linear tail-biting trellises for a block code.

An interesting property that is known for conventional trellises is that the minimal conventional trellis for a linear block

code and its dual have the same state-complexity profile [2]. Koetter and Vardy have suggested a dual trellis construction using an intersection product [10] and the state-complexity profiles of the primal and dual trellises are equal if the primal trellis is \prec_θ -minimal [10].

An algebraic characterization of conventional trellises is provided by Forney [2] where state spaces at each time index correspond to quotient groups with respect to a normal subgroup. The subgroup is constructed by introducing the notions of *past* and *future* subcodes.

In this paper, we generalize the Bahl–Cocke–Jelinek–Raviv (BCJR) [1], [15] and Forney [2] constructions to obtain tail-biting trellises. We show that a linear tail-biting trellis for an (n, k) linear block code over \mathbb{F}_q can be constructed from an arbitrary parity check matrix \mathbf{H} for the code along with a $(n - k) \times k$ matrix \mathcal{D} with coefficients from \mathbb{F}_q which we term a *displacement matrix*. The set of BCJR labels is constructed from \mathbf{H} and \mathcal{D} . This description facilitates a simple and direct dual construction algorithm yielding dual trellises with exactly the same state-complexity profiles as those for the primal trellises for the class of nonmergeable trellises [4], [7], [16]. This class properly includes the class of \prec_θ -minimal trellises [10].

We further show that pasts and futures for conventional trellises can be generalized to circular pasts and futures for tail-biting trellises, thereby obtaining a natural extension of the algebraic specification of Forney to tail-biting trellises.

Section II contains the notation and definitions used in the subsequent sections. Section III describes our scheme for a BCJR-like labeling of tail-biting trellises. Section IV shows how the BCJR scheme can be used to directly construct dual trellises whose state-complexity profiles are identical to those of the primal ones. Section V gives a natural generalization of the Forney specification for conventional trellises to tail-biting trellises. Section VI concludes the paper.

II. PRELIMINARIES

We refer to an (n, k) linear block code over \mathbb{F}_q as an $(n, k)_q$ code. Every block code has a *combinatorial* description in the form of a *trellis*. Trellises for block codes were introduced in 1974 by Bahl *et al.* [1], and further important results were reported in [2], [3], [5], [6], [15], [17]. We give a few definitions below.

Definition 1: A conventional trellis $T = (V, E, \Sigma)$ of depth n is an edge-labeled directed graph with the property that the set V can be partitioned into $n + 1$ vertex classes

$$V = V_0 \cup V_1 \cup \dots \cup V_n \quad (1)$$

Manuscript received January 7, 2004; revised May 13, 2006. The material in this paper was presented in part in the IEEE International Symposium on Information Theory, Yokohama, Japan, June/July 2003, in part in the 41st Annual Allerton Conference on Communication, Control and Computing, Allerton, IL, October 2003, and in part in the International Symposium on Information Theory and Its Applications (ISITA), Parma, Italy, October 2004.

A. V. Nori was with the Department of Computer Science and Automation, Indian Institute of Science, Bangalore 560 012, India. He is now with Microsoft Research India, Bangalore 560 080, India (e-mail: adityan@microsoft.com).

P. Shankar is with the Department of Computer Science and Automation, Indian Institute of Science, Bangalore 560 012, India (e-mail: priti@csa.iisc.ernet.in).

Communicated by R. J. McEliece, Associate Editor for Coding Theory.

Digital Object Identifier 10.1109/TIT.2006.881733

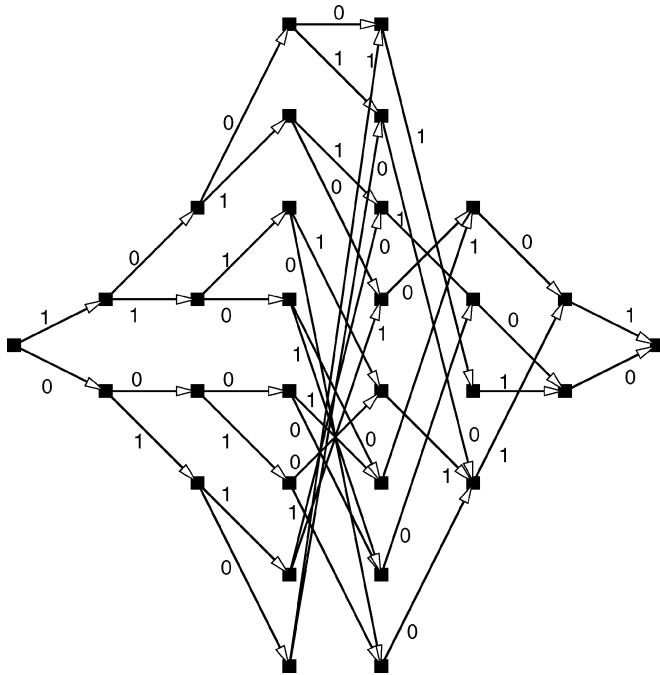


Fig. 1. The minimal conventional trellis for a $(7, 4)_2$ Hamming code.

such that every edge in T is labeled with a symbol from the alphabet Σ , and begins at a vertex of V_i and ends at a vertex of V_{i+1} , for some $i \in \{0, 1, \dots, n-1\}$.

The length of a path (in edges) from the root to any vertex is unique and the set of indices $\mathcal{I} = \{0, 1, \dots, n\}$ for the partition in (1) are the *time indices*. The quantity $\log_{|\Sigma|} |V_i|$ is the *state-complexity* of the trellis at time index i and the sequence $\{\log_{|\Sigma|} |V_i|, 0 \leq i \leq n\}$ defines the *state-complexity profile* (SCP) of the trellis.

A measure of trellis complexity commonly used by coding theorists has to do with the SCP. A trellis T is said to be *minimal* if the maximum state-complexity over all time indices denoted by $s_{\max}(T)$ is minimized over all possible coordinate permutations of the code [6]. Other measures are the total number of states, the total number of edges, the maximum number of edges at any time index and the product of the state cardinalities over all time indices. It is well known that minimal trellises for linear block codes are unique [6] and simultaneously satisfy all definitions of minimality. They are also *biproper* (that is, any pair of edges directed towards a vertex has distinct labels, and so also any pair of edges leaving a vertex). The trellis in Fig. 1 is the minimal trellis for a $(7, 4)_2$ Hamming code. We will now review concepts related to tail-biting trellises [10].

Definition 2: A tail-biting trellis $T = (V, E, \Sigma)$ of depth n is an edge-labeled directed graph with the property that the set V can be partitioned into n vertex classes

$$V = V_0 \cup V_1 \cup \dots \cup V_{n-1} \quad (2)$$

such that every edge in T is labeled with a symbol from the alphabet Σ , and begins at a vertex of V_i and ends at a vertex of $V_{i+1(\text{mod } n)}$, for some $i \in \{0, 1, \dots, n-1\}$.

As for conventional trellises, the set of indices $\mathcal{I} = \{0, 1, \dots, n-1\}$ for the partition in (2) are the *time indices*. We

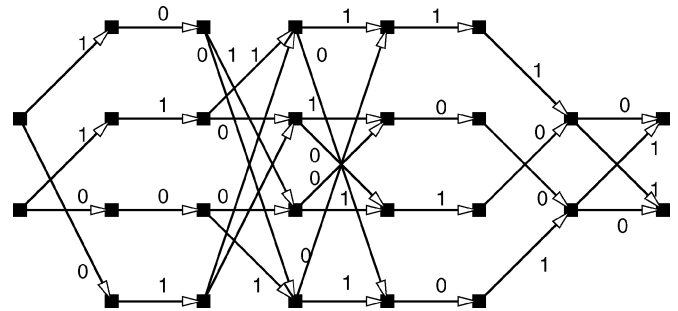


Fig. 2. A tail-biting trellis for the $(7, 4)_2$ Hamming code of Fig. 1.

identify \mathcal{I} with \mathbb{Z}_n , the residue classes of integers modulo n . An interval of indices $[i, j]$ represents the sequence $\{i, i+1, \dots, j\}$ if $i < j$, and the sequence $\{i, i+1, \dots, n-1, 0, \dots, j\}$ if $i > j$. Every cycle of length n in T starting at a vertex of V_0 defines a vector $(a_1, a_2, \dots, a_n) \in \Sigma^n$ which is an *edge-label sequence*. We assume that every vertex and every edge in the tail-biting trellis lies on some cycle. The trellis T is said to *represent* a block code \mathcal{C} over Σ if the set of all edge-label sequences in T is equal to \mathcal{C} . Let $\mathcal{C}(T)$ denote the code represented by the trellis T . The trellis in Fig. 2 is a tail-biting trellis for the $(7, 4)_2$ Hamming code of Fig. 1.

In addition to the labeling of edges, each vertex in the set V_i can be labeled by a sequence of length $l_i \geq \lceil \log_{|\Sigma|} |V_i| \rceil$ of elements in Σ , all vertex labels at a given depth being distinct. Thus every cycle in this labeled trellis defines a sequence of length $n+l$ (where $l = l_1 + l_2 + \dots + l_n$) over Σ , consisting of alternating labels of vertices and edges in T . This sequence is called the *label sequence* of a cycle in T . The set of all label sequences in a labeled tail-biting trellis is called the *label code* represented by T and is denoted by $\mathcal{S}(T)$. Fig. 11 illustrates a labeled tail-biting trellis. A trellis is said to be *one-to-one* if there is a one-to-one correspondence between the cycles in T and the codewords in $\mathcal{C}(T)$, and it is *reduced* if every vertex and every edge in T belongs to at least one cycle.

Definition 3: A trellis T is said to be linear if there exists a vertex labeling of T such that $\mathcal{S}(T)$ is a vector space.

The notion of mergeability [4], [7], [16] is also useful here.

Definition 4: A trellis T is mergeable if there exist vertices in the same vertex class of T that can be replaced by a single vertex, while retaining the edges incident on the original vertices, without modifying $\mathcal{C}(T)$.

If a trellis contains no vertices that can be merged, it is said to be nonmergeable.

Example 1: Consider a $(3, 2)_2$ code with generator matrix \mathbf{G} defined as follows:

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}.$$

A mergeable linear tail-biting trellis for this code is shown in Fig. 3—the mergeable vertices are enclosed by dashed ellipses. In contrast, another tail-biting trellis for this code shown in Fig. 4 is nonmergeable and a non one-to-one linear trellis.

Koetter and Vardy [9] have shown that if a linear trellis is nonmergeable, then it is also biproper. However, though the converse is true for conventional trellises, it is not true in general

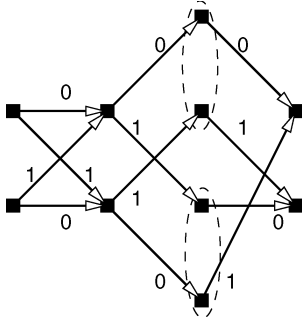


Fig. 3. A mergeable tail-biting trellis for the $(3, 2)_2$ code in Example 1.

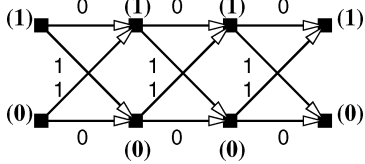


Fig. 4. A nonmergeable tail-biting trellis for the $(3, 2)_2$ code in Example 1.

for tail-biting trellises as illustrated by Fig. 3. They show that for tail-biting trellises the following relation holds:

$$\begin{aligned} & \{ \text{linear trellises} \} \\ & \cup \\ & \{ \text{biproper linear trellises} \} \\ & \cup \\ & \{ \text{nonmergeable linear trellises} \}. \end{aligned}$$

In the discussion that follows, we restrict ourselves to trellises representing linear block codes over the alphabet $\Sigma = \mathbb{F}_q$. Any linear trellis, conventional or tail-biting, for an $(n, k)_q$ linear code \mathcal{C} can be constructed as a *trellis product* [3] of the representation of the individual trellises corresponding to the k rows of the generator matrix \mathbf{G} for \mathcal{C} [10]. The trellis product T of a pair of trellises T_1 and T_2 will have at index i a set of vertices which is the Cartesian product of vertices of T_1 and T_2 at that time index, with an edge labeled $a + a'$ (where $+$ denotes addition in the field \mathbb{F}_q), from (v_1, v_2) to (v'_1, v'_2) , at time indices i and $i + 1$, respectively, whenever (v_1, a, v_2) is an edge between vertices at time indices i and $i + 1$ in T_1 , and (v'_1, a', v'_2) is an edge between vertices at time indices i and $i + 1$ in T_2 . Let $\{\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_k\}$ be the rows of a generator matrix \mathbf{G} for the linear code \mathcal{C} . Each vector \mathbf{g}_i generates a one-dimensional (1-D) subcode of \mathcal{C} , which we denote by $\langle \mathbf{g}_i \rangle$. Therefore $\mathcal{C} = \langle \mathbf{G} \rangle \stackrel{\text{def}}{=} \langle \mathbf{g}_1 \rangle + \langle \mathbf{g}_2 \rangle + \dots + \langle \mathbf{g}_k \rangle$, and the trellis representing \mathcal{C} is given by $T = T_1 \times T_2 \times \dots \times T_k$, where T_i is the trellis representing $\langle \mathbf{g}_i \rangle$, $1 \leq i \leq k$.

To specify the component trellises in the trellis product above, we will need the notions of linear and circular spans and elementary trellises [10]. Given a codeword $\mathbf{c} \in \mathcal{C}$, the *linear span* of \mathbf{c} is the smallest interval $[i, j]$, such that $i, j \in \{1, 2, \dots, n\}$, $j > i$, which contains all the nonzero positions of \mathbf{c} . A *circular span* has exactly the same definition with $i > j$. Note that for a given vector, the linear span is unique, but circular spans are not—they depend on the runs of consecutive zeros chosen for the complement of the span with respect to the index set \mathcal{I} . For example, the vector (100101) can be chosen to have circular span $[4, 1]$

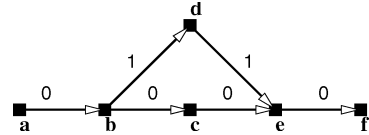


Fig. 5. Elementary trellis for (0110) with span $[2, 3]$.

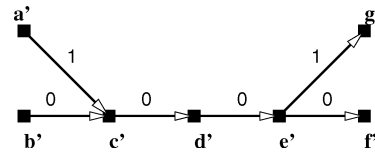


Fig. 6. Elementary trellis for (1001) with span $[4, 1]$.

or $[6, 4]$. For a vector $\mathbf{x} = (x_1, \dots, x_n)$ over the field \mathbb{F}_q with span $[a, b]$ (either linear or circular), there is a unique *elementary trellis* representing $\langle \mathbf{x} \rangle$ [10]. This trellis has q vertices at those positions that belong to $[a, b]$, and a single vertex at other positions. Consequently, T_i in the trellis product mentioned earlier, is the elementary trellis representing $\langle \mathbf{g}_i \rangle$ for some choice of span (either linear or circular).

Definition 5: If $[i, j]$ is the span of a vector $\mathbf{v} = (v_1, v_2, \dots, v_n)$, then $[j, (i - 1) \bmod n]$ is its zero-run. The zero-run of \mathbf{v} is a (possibly circular) interval of time-indices at which the path corresponding to \mathbf{v} merges with the all-zero path.

Note that in case the span of a vector is chosen to be the whole interval \mathcal{I} , the zero run is the empty sequence.

Koetter and Vardy [10] have shown that any linear trellis, conventional or tail-biting can be constructed from a generator matrix whose rows can be partitioned into two sets, those which have linear span, and those taken to have circular span. The trellis is formed as a product of the elementary trellises corresponding to these rows. We will represent such a generator matrix as

$$\mathbf{G}_{\text{KV}} = \begin{bmatrix} \mathbf{G}_l \\ \mathbf{G}_c \end{bmatrix}$$

where \mathbf{G}_l is the submatrix consisting of rows with linear span, and \mathbf{G}_c the submatrix of rows with circular span, and refer to it as a Koetter–Vardy (KV) product matrix and the corresponding trellis as the KV trellis.

Example 2: Consider a $(4, 2)_2$ linear block code whose KV generator matrix is

$$\mathbf{G}_{\text{KV}} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \begin{matrix} [2, 3] \\ [4, 1] \end{matrix}.$$

The spans and elementary trellises for the rows 0110 and 1001 are shown in Figs. 5 and 6, respectively. The resulting KV product trellis is shown in Fig. 7.

III. THE TAIL-BITING BCJR TRELLIS

The original BCJR algorithm [1] constructs the minimal conventional trellis for a linear block code in the following way. Let \mathbf{H} be an arbitrary $(n - k) \times n$ parity check matrix for an

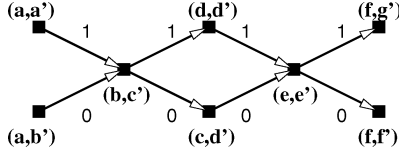


Fig. 7. The KV product trellis for the (4, 2) linear code.

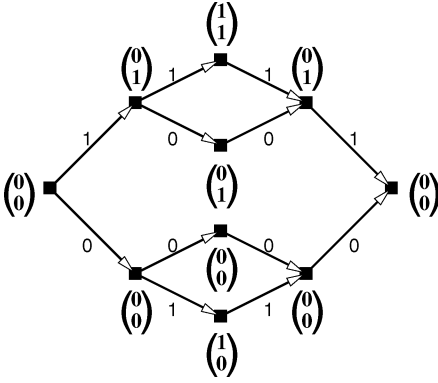


Fig. 8. The minimal conventional BCJR trellis for the (4, 2)₂ code.

(n, k) linear block code \mathcal{C} over \mathbb{F}_q , and let $\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_n, \mathbf{h}_i \in \mathbb{F}_q^{(n-k) \times n}$, $1 \leq i \leq n$ be the columns of \mathbf{H} . Every codeword $\mathbf{c} = (c_1, \dots, c_n) \in \mathcal{C}$ induces a sequence of states $\{\mathbf{s}_i\}_{i=0}^n$, each state being labeled by a vector in $\mathbb{F}_q^{(n-k) \times 1}$ as follows:

$$\mathbf{s}_i = \begin{cases} \mathbf{0} & \text{if } i = 0 \\ \mathbf{s}_{i-1} + c_i \mathbf{h}_i & \text{otherwise} \end{cases} \quad (3)$$

Clearly, there will be a single state at time index n as $\mathbf{H}\mathbf{c}^T = \mathbf{0}$ for all codewords $\mathbf{c} \in \mathcal{C}$. There is an edge labeled $a \in \mathbb{F}_q$ from state \mathbf{s}_{i-1} to state \mathbf{s}_i , $1 \leq i \leq n - 1$, if and only if

$$\mathbf{s}_i = \mathbf{s}_{i-1} + a\mathbf{h}_i.$$

We refer to such a labeling as a BCJR labeling of the trellis in the following section. We will also refer to the labeled sequences in the BCJR label code as the BCJR labeled words. It is well known that the set of vectors that are labels at each time index form a vector space whose dimension is the state-complexity at that time index [15]. Any generator matrix \mathbf{G} for this code can then be represented as a set of k BCJR labeled codewords. This labeled matrix is henceforth referred to as the \mathbf{G}_{BCJR} matrix and its labels depend on the parity check matrix chosen.

Example 3: Consider a self dual (4,2)₂ code with parity check matrix \mathbf{H} defined as

$$\mathbf{H} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}.$$

The BCJR label generator matrix \mathbf{G}_{BCJR} is given below

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

The minimal conventional trellis resulting from the BCJR construction for this code is shown in Fig. 8.

The BCJR algorithm for conventional trellises is now generalized to construct labeled tail-biting trellises for any linear code with the resultant trellis satisfying the following two properties:

- i) the trellis formed is biproper and linear;
- ii) the state labels at each time index form a vector space.

Let \mathcal{C} be an $(n, k)_q$ linear block code with generator matrix $\mathbf{G} = \{\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_k\}$, where $\mathbf{g}_i \in \mathbb{F}_q^{1 \times n}$, $1 \leq i \leq k$, are the k rows of \mathbf{G} , and parity check matrix $\mathbf{H} = [\mathbf{h}_1 \mathbf{h}_2 \dots \mathbf{h}_n]$, with columns $\mathbf{h}_i \in \mathbb{F}_q^{(n-k) \times 1}$, $1 \leq i \leq n$.

The tail-biting BCJR specification includes an $(n-k) \times 1$ displacement vector \mathbf{d}_g with every generator row $\mathbf{g} \in \mathbf{G}$. Specifically, the set of displacement vectors for an $k \times n$ generator matrix \mathbf{G} is an arbitrary set of k vectors from $\mathbb{F}_q^{(n-k) \times k}$, which could be linearly independent or dependent, with repetitions allowed. The displacement vector \mathbf{d}_c for any codeword $\mathbf{c} \in \mathcal{C}$ is defined as follows:

$$\mathbf{d}_c = \sum_{i=1}^k \alpha_i \mathbf{d}_{g_i}, \text{ where } \mathbf{c} = \sum_{i=1}^k \alpha_i \mathbf{g}_i, \alpha_i \in \mathbb{F}_q, \mathbf{g}_i \in \mathbf{G}. \quad (4)$$

Definition 6: Given an $(n, k)_q$ linear code \mathcal{C} with generator matrix \mathbf{G} and displacement vectors $\{\mathbf{d}_g\}_{g \in \mathbf{G}}$, the $(n-k) \times k$ matrix \mathcal{D} whose i^{th} column is equal to \mathbf{d}_{g_i} , where \mathbf{g}_i is the i^{th} row of \mathbf{G} , $1 \leq i \leq k$, is called a displacement matrix for \mathcal{C} .

By definition the all-zero codeword is associated with the $\mathbf{0}$ displacement vector.

The following lemma defines the decomposition of the code induced by the displacement matrix.

Lemma 1: Let \mathcal{C} be an $(n, k)_q$ linear code with generator matrix \mathbf{G} and displacement matrix \mathcal{D} . Then \mathcal{D} specifies a coset decomposition $\mathcal{C}/\mathcal{C}_0$ of the code \mathcal{C} , where $\mathcal{C}_0 = \{\mathbf{c} \in \mathcal{C} : \mathbf{d}_c = \mathbf{0}\}$, with every coset being associated with a unique displacement vector.

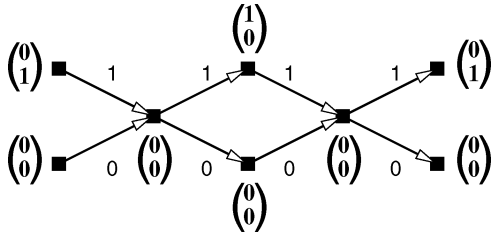
Proof: It is easily seen that $\mathcal{C}_0 = \{\mathbf{c} \in \mathcal{C} : \mathbf{d}_c = \mathbf{0}\}$ is a linear subcode of \mathcal{C} . Consider a coset $\mathcal{C}_i = \mathcal{C}_0 + \mathbf{v}_i, \mathbf{v}_i \in \mathcal{C} \setminus \mathcal{C}_0$. Since every codeword $\mathbf{x} \in \mathcal{C}_i$ takes the form $\mathbf{x} = \mathbf{c} + \mathbf{v}_i$ for some $\mathbf{c} \in \mathcal{C}_0$, it follows that $\mathbf{d}_x = \mathbf{d}_{v_i}$ (since $\mathbf{d}_c = \mathbf{0}$). Therefore, every coset is associated with a unique displacement vector and the lemma follows. \square

It should be noted that there are in general many displacement matrices that induce the same coset decomposition of a code \mathcal{C} , as the coset decomposition depends only on the choice of \mathcal{C}_0 . This subcode is generated by the largest set of linearly independent vectors, each element of which has an associated displacement vector corresponding to the all-zero vector.

Example 4: Let \mathcal{C} be a self dual (4,2)₂ code with generator matrix \mathbf{G} as follows:

$$\mathbf{G} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}.$$

Let the displacement matrix $\mathcal{D} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$. Therefore $\mathbf{d}_{0110} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ and $\mathbf{d}_{1001} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, and from (4), we have $\mathbf{d}_{0000} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ and $\mathbf{d}_{1111} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$. The matrix $\mathcal{D}' = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$ also induces the same coset decomposition as the matrix \mathcal{D} where the subgroup \mathcal{C}_0 is 0000, 0110. However the displacement vectors for the rows 1001 and 1111 are the vector $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$.


 Fig. 9. A T-BCJR trellis for the $(4, 2)_2$ code with displacement matrix \mathcal{D} .

Definition 7 (T-BCJR Labeling): Every codeword $\mathbf{c} = (c_1, c_2, \dots, c_n) \in \mathcal{C}$ induces a sequence of states $\{\mathbf{s}_i\}_{i=0}^{n-1}$, each state being labeled by a vector in $\mathbb{F}_q^{(n-k) \times 1}$ as follows:

$$\mathbf{s}_i = \begin{cases} \mathbf{d}\mathbf{c} & \text{if } i = 0 \\ \mathbf{s}_{i-1} + c_i \mathbf{h}_i & \text{otherwise} \end{cases}. \quad (5)$$

There is an edge labeled $a \in \mathbb{F}_q$ from state \mathbf{s}_{i-1} to state \mathbf{s}_i , $1 \leq i \leq n-2$, if and only if

$$\mathbf{s}_i = \mathbf{s}_{i-1} + a\mathbf{h}_i. \quad (6)$$

We refer to such a labeling as a Tail-biting BCJR (T-BCJR) trellis, and the vector \mathbf{s}_i as the i th partial syndrome of \mathbf{c} .

Definition 8: Let \mathcal{C} be an $(n, k)_q$ code with parity check matrix \mathbf{H} , generator matrix \mathbf{G} and a displacement matrix \mathcal{D} . The label generator matrix $\mathbf{G}_{\mathbf{H}, \mathcal{D}}$ is defined to be the matrix consisting of BCJR label rows of \mathbf{G} formed with respect to \mathbf{H} and \mathcal{D} .

Let \mathcal{C} be an $(n, k)_q$ code with generator matrix \mathbf{G} and parity check matrix \mathbf{H} . The T-BCJR trellis T constructed with respect to \mathbf{H} , \mathbf{G} and some arbitrary $\mathcal{D} \in \mathbb{F}_q^{(n-k) \times k}$ is thus given by

$$T = \langle \mathbf{G}_{\mathbf{H}, \mathcal{D}} \rangle \quad (7)$$

where we recall that $\langle \mathbf{G}_{\mathbf{H}, \mathcal{D}} \rangle$ is the vector space generated by the rows of $\mathbf{G}_{\mathbf{H}, \mathcal{D}}$.

Example 5: For the $(4, 2)_2$ code from Example 4, with $\mathcal{D} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$

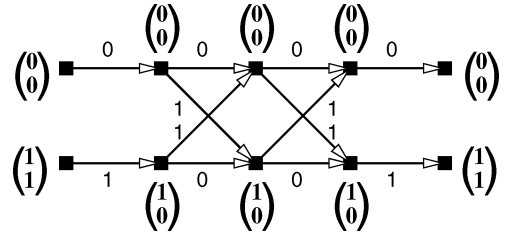
$$\mathbf{G}_{\mathbf{H}, \mathcal{D}} = \begin{bmatrix} \begin{pmatrix} 0 \\ 0 \end{pmatrix} & 0 & \begin{pmatrix} 0 \\ 0 \end{pmatrix} & 1 & \begin{pmatrix} 1 \\ 0 \end{pmatrix} & 1 & \begin{pmatrix} 0 \\ 0 \end{pmatrix} & 0 & \begin{pmatrix} 0 \\ 0 \end{pmatrix} \\ \begin{pmatrix} 0 \\ 1 \end{pmatrix} & 1 & \begin{pmatrix} 0 \\ 0 \end{pmatrix} & 0 & \begin{pmatrix} 0 \\ 0 \end{pmatrix} & 0 & \begin{pmatrix} 0 \\ 0 \end{pmatrix} & 1 & \begin{pmatrix} 0 \\ 1 \end{pmatrix} \end{bmatrix}$$

and $\langle \mathbf{G}_{\mathbf{H}, \mathcal{D}} \rangle$ is the T-BCJR trellis shown in Fig. 9. On the other hand, choosing $\mathcal{D}' = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$, results in a completely different label generator matrix

$$\mathbf{G}_{\mathbf{H}, \mathcal{D}'} = \begin{bmatrix} \begin{pmatrix} 0 \\ 0 \end{pmatrix} & 0 & \begin{pmatrix} 0 \\ 0 \end{pmatrix} & 1 & \begin{pmatrix} 1 \\ 0 \end{pmatrix} & 1 & \begin{pmatrix} 0 \\ 0 \end{pmatrix} & 0 & \begin{pmatrix} 0 \\ 0 \end{pmatrix} \\ \begin{pmatrix} 1 \\ 1 \end{pmatrix} & 1 & \begin{pmatrix} 1 \\ 0 \end{pmatrix} & 0 & \begin{pmatrix} 1 \\ 0 \end{pmatrix} & 0 & \begin{pmatrix} 1 \\ 0 \end{pmatrix} & 1 & \begin{pmatrix} 1 \\ 1 \end{pmatrix} \end{bmatrix}$$

and $\langle \mathbf{G}_{\mathbf{H}, \mathcal{D}'} \rangle$ is the T-BCJR trellis shown in Fig. 10.

As illustrated by Example 5, the structure of the T-BCJR trellis not only depends on the coset decomposition of the code,


 Fig. 10. A T-BCJR trellis for the $(4, 2)_2$ code with displacement matrix \mathcal{D}' .

but also on the label generator matrix structure imposed by the displacement matrix \mathcal{D} .

Let T be a T-BCJR trellis constructed for \mathcal{C} . The following lemmas define the properties of T .

Lemma 2: The trellis T is linear and represents \mathcal{C} .

Proof: We first prove that $\mathcal{C}(T) \subseteq \mathcal{C}$. Assume to the contrary that $\exists \mathbf{x} = (x_1, \dots, x_n) \in \mathcal{C}(T)$ such that $\mathbf{x} \notin \mathcal{C}$. Let $\mathbf{d}\mathbf{x}$ be the start vertex of the cycle representing the word \mathbf{x} . The invariant maintained by the algorithm for every edge $\mathbf{e} = (\mathbf{v}_{i-1}, a, \mathbf{v}_i) \in V_{i-1} \times \mathbb{F}_q \times V_i$ is

$$\mathbf{v}_{i-1} + a\mathbf{h}_i = \mathbf{v}_i.$$

Therefore, $\mathbf{d}\mathbf{x} + \sum_{i=1}^n x_i \mathbf{h}_i = \mathbf{d}\mathbf{x} \Rightarrow \mathbf{H}\mathbf{x}^t = \mathbf{0} \Rightarrow \mathbf{x} \in \mathcal{C}$, thus contradicting our original assumption. That $\mathcal{C} \subseteq \mathcal{C}(T)$ follows from the construction. We next show that the trellis is linear.

Let $\mathbf{x}, \mathbf{y} \in \mathcal{C}(T)$ and let $\mathbf{x}', \mathbf{y}' \in \mathcal{S}(T)$ respectively, be their label codewords. Since $\mathcal{C}(T) = \mathcal{C}$, we have $\mathbf{z} = \mathbf{x} + \mathbf{y} \in \mathcal{C}(T)$. In order to prove linearity of T , we need to show that $\mathbf{z}' = \mathbf{x}' + \mathbf{y}'$ also belongs to $\mathcal{S}(T)$. We have $\mathbf{x}' = \langle \mathbf{d}\mathbf{x}, x_1, \mathbf{u}_1, \dots, x_n, \mathbf{d}\mathbf{x} \rangle$, such that $\mathbf{u}_i \mid_{1 \leq i \leq n} = \mathbf{d}\mathbf{x} + \sum_{j=1}^i x_j \mathbf{h}_j$, and $\mathbf{y}' = \langle \mathbf{d}\mathbf{y}, y_1, \mathbf{v}_1, \dots, y_n, \mathbf{d}\mathbf{y} \rangle$, such that $\mathbf{v}_i \mid_{1 \leq i \leq n} = \mathbf{d}\mathbf{y} + \sum_{j=1}^i y_j \mathbf{h}_j$. Therefore

$$\begin{aligned} \mathbf{z}' &= \langle \mathbf{d}\mathbf{x} + \mathbf{d}\mathbf{y}, x_1 + y_1, \mathbf{u}_1 + \mathbf{v}_1, \dots, x_n + y_n, \mathbf{d}\mathbf{x} + \mathbf{d}\mathbf{y} \rangle \Rightarrow \\ \mathbf{z}' &= \langle \mathbf{d}\mathbf{z}, z_1, \mathbf{d}\mathbf{z} + z_1 \mathbf{h}_1, \dots, z_n, \mathbf{d}\mathbf{z} \rangle \end{aligned}$$

which shows that \mathbf{z}' is the label codeword in $\mathcal{S}(T)$ representing the codeword \mathbf{z} , thus proving that T is indeed a linear trellis representing \mathcal{C} . \square

Lemma 3: The trellis T is biproper.

Proof: Assume there exists a vertex \mathbf{v} at some time index i , with two outgoing edges $(\mathbf{v}, a, \mathbf{v}_1)$ and $(\mathbf{v}, a, \mathbf{v}_2)$, $\mathbf{v}_1 \neq \mathbf{v}_2$ (that is, T is not proper). From the BCJR construction we know that $\mathbf{v} + a\mathbf{h}_{i+1} = \mathbf{v}_1$ and $\mathbf{v} + a\mathbf{h}_{i+1} = \mathbf{v}_2 \Rightarrow \mathbf{v}_1 = \mathbf{v}_2$, which contradicts our assumption that \mathbf{v}_1 and \mathbf{v}_2 are distinct. Therefore T must be proper. A similar argument shows that T must also be co-proper (that is, T with its edges reversed is also proper), thus proving that T is a biproper trellis. \square

Let \mathbf{G}_i and \mathbf{H}_i respectively, denote the submatrices consisting of the first i columns of \mathbf{G} and \mathbf{H} . For every $i \in \{0, 1, \dots, n-1\}$, define a matrix M_i as follows:

$$M_i = [\mathbf{H}_i \mathbf{G}_i^t + \mathcal{D}]. \quad (8)$$

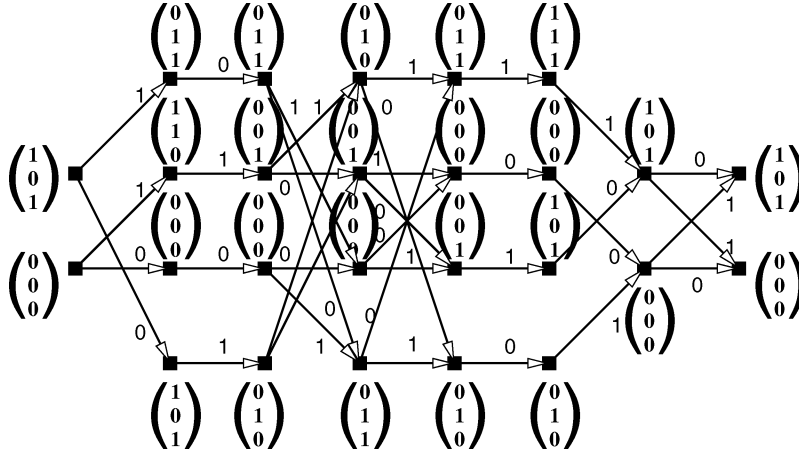


Fig. 11. A minimal T-BCJR trellis for the $(7, 4)_2$ Hamming code.

Lemma 4 (State-Space Lemma): For all time indices $i \in \mathcal{I}$, V_i the state space of the trellis T at time index i equals the column-space of M_i .

Proof: For every generator $\mathbf{g} = (g_1, g_2, \dots, g_n) \in \mathbf{G}$, the state \mathbf{v}_i at time index i is given by $\mathbf{v}_i = \mathbf{d}\mathbf{g} + \mathbf{H}_i(g_1, g_2, \dots, g_i)^t$. Therefore the set of states at time index i for generators in \mathbf{G} is defined by the columns of $M_i = [\mathbf{H}_i \mathbf{G}_i^t + \mathfrak{D}]$, and the lemma follows. \square

Example 6: For the $(4, 2)_2$ code defined in Example 4, the T-BCJR trellis shown in Fig. 9.

Example 7: Consider the $(7, 4)_2$ Hamming code \mathcal{C} with parity check and generator matrices defined as follows:

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{G} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Choosing

$$\mathfrak{D} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

we obtain a minimal T-BCJR trellis T for \mathcal{C} with $s_{\max}(T) = 2$, as illustrated in Fig. 11.

We will now define a displacement matrix derived directly from a KV product specification of a nonmergeable linear trellis.

Definition 9: Let \mathbf{G} be a KV product matrix for a nonmergeable linear trellis T representing an $(n, k)_q$ code \mathcal{C} . Let the parity check matrix for \mathcal{C} be $\mathbf{H} = [\mathbf{h}_1 \mathbf{h}_2 \dots \mathbf{h}_n]$, with columns $\mathbf{h}_i \in \mathbb{F}_q^{(n-k) \times 1}$, $1 \leq i \leq n$. A T-BCJR labeling for T is defined by the displacement matrix \mathfrak{D} with respect to \mathbf{H} and \mathbf{G} as follows.

The i th column of \mathfrak{D} is equal to $\mathbf{0}$ if the i th row of \mathbf{G} has linear span, else it is equal to $\mathbf{d}\mathbf{g} = \sum_{j=a}^n g_j \mathbf{h}_j$, where $\mathbf{g} = (g_1, g_2, \dots, g_n)$ is the i th row of \mathbf{G} with circular span $[a, b]$.

The KV product construction of tail-biting trellises naturally induces a k -tuple labeling of each vertex of the final trellis for

the code formed as the product of the k elementary trellises for the basis vectors in the generator matrix. Each element of the k -tuple is a label of a vertex of an elementary trellis. The following lemma follows directly from the product construction.

Lemma 5: Let $\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_k$ be prefixes of codewords in elementary trellises leading to vertices v_1, v_2, \dots, v_k respectively in the trellises. Then the codeword prefix $\mathbf{c}_1 + \mathbf{c}_2 + \dots + \mathbf{c}_k$ leads to a vertex labeled (v_1, v_2, \dots, v_k) , that is, the Cartesian product of the k vertices in the elementary trellis.

The following lemma follows directly from the T-BCJR construction.

Lemma 6: Let $\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_k$ be prefixes of codewords in elementary trellises leading to vertices with T-BCJR labels $\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_k$ respectively in the trellises. Then the codeword prefix $\mathbf{c}_1 + \mathbf{c}_2 + \dots + \mathbf{c}_k$ leads to vertex with T-BCJR label $(\mathbf{b}_1 + \mathbf{b}_2 \dots + \mathbf{b}_j)$, that is, the sum of the k T-BCJR labels.

Definition 10: A subtrellis of a tail-biting trellis is defined by the set of all paths that begin and end at the same vertex at time index 0.

The number of subtrellises of a tail-biting trellis is thus the number of vertices at time index 0.

The following lemmas state some properties of paths in the T-BCJR trellis.

Lemma 7: Let $\mathbf{c} = (\mathbf{c}_1, \dots, \mathbf{c}_i)$ and $\mathbf{c}' = (\mathbf{c}'_1, \mathbf{c}'_2, \dots, \mathbf{c}'_i)$ be prefixes of codewords \mathbf{c} and \mathbf{c}' of length i such that they induce the same T-BCJR label \mathbf{b} . Then if they are paths in the same T-BCJR subtrellis they share all continuations. If they are paths in distinct subtrellises they share no continuations.

Proof: If $\mathbf{c} = (\mathbf{c}_1, \dots, \mathbf{c}_i)$ and $\mathbf{c}' = (\mathbf{c}'_1, \mathbf{c}'_2, \dots, \mathbf{c}'_i)$ correspond to paths in the same subtrellis and if \mathbf{d} is the displacement vector associated with that subtrellis, we have $\mathbf{d} + \mathbf{c}_1 \mathbf{h}_1 + \mathbf{c}_2 \mathbf{h}_2 + \dots + \mathbf{c}_i \mathbf{h}_i = \mathbf{d} + \mathbf{c}'_1 \mathbf{h}_1 + \mathbf{c}'_2 \mathbf{h}_2 + \dots + \mathbf{c}'_i \mathbf{h}_i$ which implies that $(\mathbf{c}_1, \dots, \mathbf{c}_i) \mathbf{H}_i^t = (\mathbf{c}'_1, \dots, \mathbf{c}'_i) \mathbf{H}_i^t$ where \mathbf{H}_i is the submatrix consisting of the first i columns of \mathbf{H} . Thus they obviously share all continuations. Assume they correspond to paths in different subtrellises. If they share a continuation, say \mathbf{c}'' then there must be two paths labelled \mathbf{c}'' originating at vertex labeled \mathbf{b} to the final nodes of the two subtrellises. This violates the property of biproperness guaranteed by Lemma 3. Hence the lemma. \square

A similar statement can be proved replacing continuations by pasts and codeword prefixes by suffixes.

Lemma 8: Let codeword prefixes \mathbf{c} and \mathbf{c}' lead to distinct states s and s' in a KV trellis T but lead to the same state in the T-BCJR trellis constructed according to Definition 9. Then merging s and s' does not change the code.

Proof: Assume \mathbf{c} and \mathbf{c}' induce the same T-BCJR label, but lead to different states s and s' in the KV trellis. Merging states s and s' will change the code only if a cycle with a new edge-label sequence is introduced as a result of the merging. By Lemma 7, prefixes originating at the same vertex inducing the same T-BCJR label share all continuations, and therefore no new edge-label sequences corresponding to cycles are introduced as a result of the merging. Hence the code defined by the trellis does not change. \square

Lemma 9: Codewords $\mathbf{c} = (c_1, \dots, c_n)$ and $\mathbf{c}' = (c'_1, c'_2, \dots, c'_n)$ lead to distinct vertices at level i in a nonmergeable KV trellis if and only if \mathbf{c} and \mathbf{c}' induce distinct T-BCJR labels (as given in Definition 9) at level i .

Proof: Assume that \mathbf{c} and \mathbf{c}' lead to distinct vertices at level i in a KV trellis. Assume they lead to the same T-BCJR label. This implies that merging the corresponding vertices in the KV trellis will not change the code by Lemma 8, contradicting the assumption that T is nonmergeable.

Conversely assume that \mathbf{c} and \mathbf{c}' induce distinct T-BCJR labels at level i , but lead to the same KV trellis vertex, say (v_1, v_2, \dots, v_k) , at level i . Then the projections of this vertex into the elementary trellis vertices are v_1, v_2, \dots, v_k . Let vertices v_1, v_2, \dots, v_k have T-BCJR labels $\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_k$ respectively. Therefore, by Lemma 6, the corresponding T-BCJR node is $\mathbf{b}_1 + \mathbf{b}_2 + \dots + \mathbf{b}_k$, contradicting the assumption that \mathbf{c} and \mathbf{c}' induce distinct T-BCJR nodes. \square

Lemma 10: The KV nonmergeable elementary trellis for a row vector \mathbf{g} of \mathbf{G} and the T-BCJR trellis for \mathbf{g} constructed according to Definition 9 are isomorphic to each other.

Proof: From Lemma 9 there is a one-to-one correspondence between the vertices of the KV and T-BCJR nonmergeable elementary trellises. That there is a one-to-one correspondence between the edges follows directly from the definitions of the elementary trellises. \square

Theorem 3.1: Let \mathbf{G} be a KV product matrix for a nonmergeable linear trellis T representing an $(n, k)_q$ linear block code \mathcal{C} . Let the T-BCJR trellis T' be defined as in Definition 9. Then T is isomorphic to T' .

Proof: Assume that $T = (V, E, \mathbb{F}_q)$ is a nonmergeable linear KV trellis. Let $L_i = \{v_1, v_2, \dots, v_k\}$ be a set of vertices at time index i one in each elementary trellis with T-BCJR labels $\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_k$ respectively. By Lemma 5, we have $\mathbf{v} = (v_1, v_2, \dots, v_k) \in V_i$ as the corresponding node in the product trellis T . For the T-BCJR trellis $T' = \langle \mathbf{G}, \mathbf{H}, \mathcal{D} \rangle = (V', E', \mathbb{F}_q)$, by Lemma 6, the corresponding T-BCJR vertex $\mathbf{b} \in V'_i$ is given by

$$\mathbf{b} = \mathbf{b}_1 + \mathbf{b}_2 + \dots + \mathbf{b}_k.$$

Define a map $\phi: V \rightarrow V'$, from the labeled vertices in T to the labeled vertices in T' as follows:

$$(v_1, v_2, \dots, v_k) \mapsto \mathbf{b}_1 + \mathbf{b}_2 + \dots + \mathbf{b}_k.$$

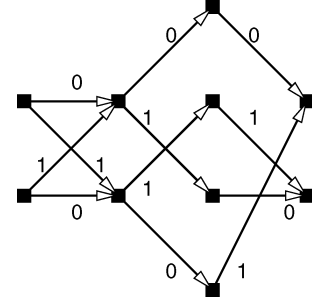


Fig. 12. A trellis for the $(3, 2)_2$ code not computable by a T-BCJR construction.

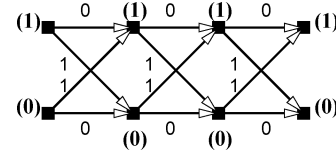


Fig. 13. A non-one-to-one nonmergeable T-BCJR trellis for the $(3, 2)_2$ code.

From Lemma 9, we see that ϕ is a bijection. Next, from the isomorphism of the elementary KV and T-BCJR trellises, and linearity of the trellises it follows that $(\mathbf{x}_1, \alpha, \mathbf{x}_2) \in E$ is an edge in the KV trellis T if and only if the T-BCJR trellis has an edge $(\phi(\mathbf{x}_1), \alpha, \phi(\mathbf{x}_2)) \in E'$. Therefore, the trellises T and T' are isomorphic. \square

Corollary 3.2: Every nonmergeable linear tail-biting trellis can be constructed by a T-BCJR construction.

It is worth noting that every biproper linear trellis cannot be constructed by T-BCJR construction. The biproper linear trellis for the $(3, 2)_2$ code in Fig. 12 is one such example. Note that this trellis is computable by the KV product construction with the product matrix defined as

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1, 3 \\ 2, 1 \end{bmatrix}$$

The following is an example of a T-BCJR trellis which is biproper but not one-to-one.

Example 8: Consider a $(3, 2)_2$ code \mathcal{C} with generator matrix \mathbf{G} , parity check matrix \mathbf{H} and displacement matrix \mathcal{D} defined as follows:

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \quad \mathbf{H} = [1 \quad 1 \quad 1] \quad \mathcal{D} = [0 \quad 1].$$

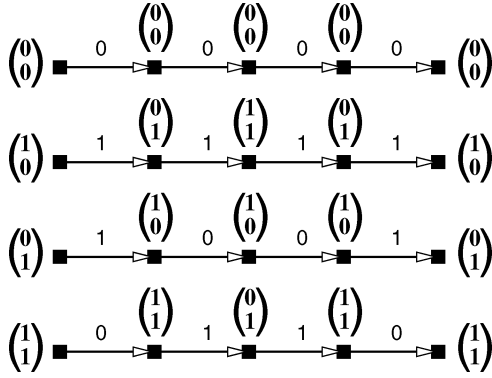
The T-BCJR trellis shown in Fig. 13 is nonmergeable but not one-to-one.

A question that naturally arises here is whether the T-BCJR trellises exactly represent the class of nonmergeable trellises. Below is an example to show they do not.

Example 9: Consider the self dual $(4, 2)_2$ linear code specified by

$$\mathbf{H} = \mathbf{G} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad \mathcal{D} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

The T-BCJR trellis (for the above choice of \mathcal{D}) for \mathcal{C} is shown in Fig. 14 and is mergeable. It is easy to see that to see that this

Fig. 14. A mergeable T-BCJR trellis for the $(4, 2)_2$ code.

trellis corresponds to a coset decomposition of the code with respect to the subcode $\mathcal{C}_0 = \{0000\}$.

Therefore, we have the following lemma.

Lemma 11: The class of trellises computed by the T-BCJR construction lies in between the class of nonmergeable linear trellises and the class of biproper linear trellises.

We conclude this section by observing that the s_{\max} -minimality problem for an $(n, k)_q$ code \mathcal{C} with parity check matrix \mathbf{H} and generator matrix \mathbf{G} may be stated as follows.

1) *Problem:* Find a displacement matrix \mathcal{D} that minimizes $s_{\max}(T)$ for a T-BCJR trellis $T = \langle \mathbf{G}_{\mathbf{H}, \mathcal{D}} \rangle$ representing \mathcal{C}

$$\mu(\mathcal{C}) \stackrel{\text{def}}{=} \min_{\mathcal{D} \in \mathbb{F}_q^{(n-k) \times k}} s_{\max}(\langle \mathbf{G}_{\mathbf{H}, \mathcal{D}} \rangle). \quad (9)$$

Let \mathcal{C} be an $(n, k)_q$ code with generator matrix \mathbf{G} and parity check matrix \mathbf{H} . Then from the BCJR construction, we know that a conventional trellis for \mathcal{C} is completely specified by the sequence

$$\Phi_{\mathcal{C}} = \{\mathbf{S}_i\}_{i=0}^n, \quad \text{where } \mathbf{S}_i = \mathbf{H}_i \mathbf{G}_i^t \quad (10)$$

where we recall that the column-space of each \mathbf{S}_i represents the set of states at level i of the trellis. Every nonmergeable linear tail-biting trellis for \mathcal{C} can be specified as

$$\Phi_{\mathcal{C}} + \mathcal{D} \stackrel{\text{def}}{=} \{\mathbf{S}_i + \mathcal{D}\}_{i=0}^n, \quad \text{where } \mathcal{D} \in \mathbb{F}_q^{(n-k) \times k}. \quad (11)$$

Define the rank of the sequence $\Phi_{\mathcal{C}} + \mathcal{D}$ as follows:

$$\text{rank}(\Phi_{\mathcal{C}} + \mathcal{D}) \stackrel{\text{def}}{=} \max_{0 \leq i \leq n} \text{rank}(\mathbf{S}_i + \mathcal{D}). \quad (12)$$

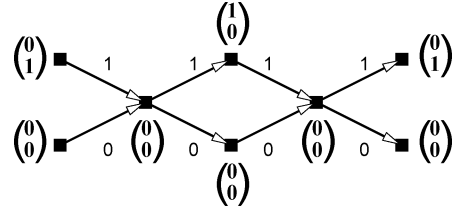
Thus the s_{\max} -minimal tail-biting trellis computation problem may thus be formulated as follows.

2) *Problem:* Find a matrix \mathcal{D} such that $\text{rank}(\Phi_{\mathcal{C}} + \mathcal{D})$ is minimized

$$\mu(\mathcal{C}) \stackrel{\text{def}}{=} \min_{\mathcal{D} \in \mathbb{F}_q^{(n-k) \times k}} \text{rank}(\Phi_{\mathcal{C}} + \mathcal{D}). \quad (13)$$

IV. THE TAIL-BITING DUAL TRELLIS

There are a number of interesting connections between the minimal conventional trellis for a linear code and the minimal conventional trellis for its dual code. It is an interesting fact that

Fig. 15. A T-BCJR[⊥] trellis for the $(4, 2)_2$ code.

algebraic and combinatorial duality are related, and this was first established by Forney [2]. Koetter and Vardy [10] have defined a special product operation called the *intersection product* to construct a dual linear tail-biting trellis directly from a generator matrix for the primal code. This results in a linear tail-biting trellis T^\perp for the dual code that has the same SCP if the primal trellis T is \prec_θ -minimal (that is, minimal under component-wise ordering), otherwise T^\perp has a smaller SCP than T . Our construction that is based on the T-BCJR specification of linear tail-biting trellises extends the Koetter–Vardy result to a larger class of trellises.

Let \mathbf{G} , \mathbf{H} and \mathcal{D} respectively, be the generator, parity and displacement matrices for a primal code \mathcal{C} . Given a primal trellis specification $T = \langle \mathbf{G}_{\mathbf{H}, \mathcal{D}} \rangle$, the dual BCJR construction (T-BCJR[⊥]) computes a biproper linear tail-biting trellis T^\perp for the dual code \mathcal{C}^\perp , with the property that the SCP of T^\perp is equal to the SCP of T . In other words, given a *minimal* (under any definition of minimality) trellis T for the primal code, T-BCJR[⊥] computes a *minimal* linear tail-biting trellis T^\perp for the dual code such that the SCP of T is equal to the SCP of T^\perp . We define the trellis T^\perp as follows.

Definition 11 (T-BCJR[⊥] Construction): Let \mathcal{C} be an $(n, k)_q$ code with generator and parity check matrices \mathbf{G} and \mathbf{H} respectively. Let $T = \langle \mathbf{G}_{\mathbf{H}, \mathcal{D}} \rangle$ be a biproper linear trellis representing \mathcal{C} for some $\mathcal{D} \in \mathbb{F}_q^{(n-k) \times k}$. Then the T-BCJR[⊥] trellis $T^\perp = (V^\perp, E^\perp, \mathbb{F}_q)$ representing a $(n, n-k)_q$ code is defined as

$$T^\perp = \langle \mathbf{H}_{\mathbf{G}, \mathcal{D}^t} \rangle. \quad (14)$$

Example 10: For the $(4, 2)_2$ code defined in Example 6, we have

$$\mathbf{H}_{\mathbf{G}, \mathcal{D}^t} = \begin{bmatrix} \begin{pmatrix} 0 \\ 0 \end{pmatrix} & 0 & \begin{pmatrix} 0 \\ 0 \end{pmatrix} & 1 & \begin{pmatrix} 1 \\ 0 \end{pmatrix} & 1 & \begin{pmatrix} 0 \\ 0 \end{pmatrix} & 0 & \begin{pmatrix} 0 \\ 0 \end{pmatrix} \\ \begin{pmatrix} 0 \\ 1 \end{pmatrix} & 1 & \begin{pmatrix} 0 \\ 0 \end{pmatrix} & 0 & \begin{pmatrix} 0 \\ 0 \end{pmatrix} & 0 & \begin{pmatrix} 0 \\ 0 \end{pmatrix} & 1 & \begin{pmatrix} 0 \\ 1 \end{pmatrix} \end{bmatrix}$$

and $\langle \mathbf{H}_{\mathbf{G}, \mathcal{D}^t} \rangle$ is the trellis shown in Fig. 15.

Example 11: For the $(7, 4)_2$ Hamming code given in Example 7, we have the equation shown at the bottom of the following page and $\langle \mathbf{H}_{\mathbf{G}, \mathcal{D}^t} \rangle$ is the trellis shown in Fig. 16.

The following lemma states the properties of the T-BCJR[⊥] trellis T^\perp .

Lemma 12: The trellis T^\perp is a biproper linear trellis that represents \mathcal{C}^\perp the dual code.

Proof: Since T^\perp is a T-BCJR trellis, the lemma follows from Lemmas 2 and 3. \square

Lemma 13: For all time indices $i \in \mathcal{I}$, the state space of T^\perp at time index i equals the column-space of M_i^t .

Proof: Let M_i be the matrix defined in (8). Recall that

$$M_i = [\mathbf{H}_i \mathbf{G}_i^t + \mathcal{D}].$$

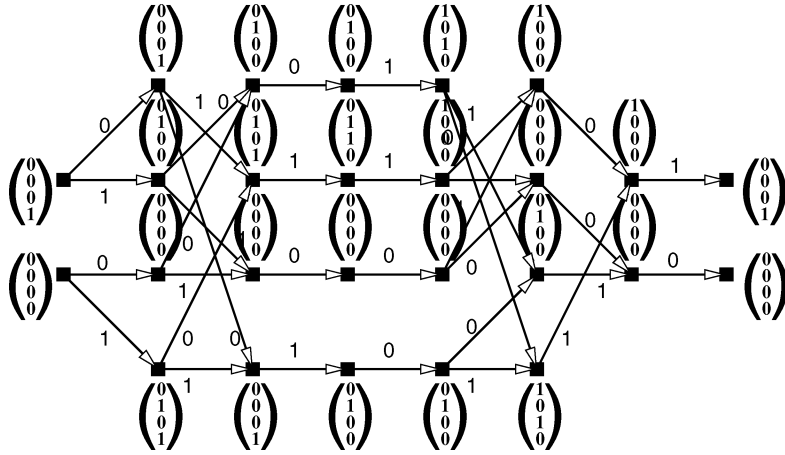


Fig. 16. A T-BCJR[⊥] trellis for the (7, 4)₂ Hamming code.

Therefore

$$M_i^t = [\mathbf{G}_i \mathbf{H}_i^t + \mathcal{D}^t].$$

Hence the column space of M_i^t is the state space of the dual trellis from Definition 11. \square

Example 12: The T-BCJR[⊥] trellises for the (4, 2)₂ self dual code and the (7, 4)₂ Hamming code from Examples 10 and 11, are shown in Figs. 15 and 16 respectively. These trellises have the same SCPs as their primal counterparts (Figs. 9 and 11).

Lemma 14: Let T and T^\perp be the trellises computed by the T-BCJR and T-BCJR[⊥] constructions, respectively. Then for all all-time indices $i \in \mathcal{I}$, the state-cardinality of T at level i equals the state-cardinality of T^\perp at level i . In other words, $|V_i| = |V_i^\perp|$.

Proof: From Lemmas 4 and 13 we know that V_i is equal to the column-space of M_i , and V_i^\perp is equal to the column-space of M_i^t . Therefore, by the “row rank = column rank” theorem of linear algebra [18], $|V_i| = |V_i^\perp|$. \square

Finally, we have the following theorem.

Theorem 4.1: Let T be a minimal linear trellis, either conventional or tail-biting, for a linear code \mathcal{C} . Then there exists a minimal linear dual trellis T^\perp for the dual code \mathcal{C}^\perp such that the SCP of T^\perp is identical to the SCP of T .

Proof: Follows from Corollary 3.2, Lemma 12 and Lemma 14. \square

V. THE TAIL-BITING FORNEY TRELLIS

In this section we generalize the Forney construction [2] for conventional trellises to obtain tail-biting trellises. Specifically, we show that there is a generalization of “pasts” and “futures” which enables the construction of a linear tail-biting trellis for a linear block code from a coset decomposition of the code, with respect to a subcode of the code.

The Forney construction [2] is an elegant algebraic characterization of conventional trellises in terms of a coset decomposition of the linear block code. We briefly review this characterization here.

Let \mathcal{C} be an $(n, k)_q$ linear code with generator matrix \mathbf{G} and parity check matrix $\mathbf{H} = [\mathbf{h}_1 \mathbf{h}_2 \dots \mathbf{h}_n]$. Let $\pi_i : \mathcal{C} \rightarrow \pi_i(\mathcal{C})$ be a map defined by $\mathbf{c} = (c_1, \dots, c_n) \mapsto c_1 \mathbf{h}_1 + \dots + c_i \mathbf{h}_i$. The map π_i thus effectively maps a codeword to its i^{th} partial syndrome. Define the past projection at time index i as follows:

$$\mathcal{P}_i = \{ (c_1, \dots, c_i) : \mathbf{c} = (c_1, \dots, c_i, c_{i+1}, \dots, c_n) \in \mathcal{C}, \pi_i(\mathbf{c}) = \mathbf{0} \}. \quad (15)$$

$$\mathbf{H}_{\mathbf{G}, \mathcal{D}^t} = \begin{bmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} & 1 & \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} & 1 & \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} & 0 & \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} & 0 & \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} & 1 & \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} & 0 & \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} & 1 & \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \\ \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} & 1 & \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} & 1 & \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} & 1 & \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} & 0 & \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} & 0 & \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} & 1 & \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} & 0 & \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \\ \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} & 0 & \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} & 1 & \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} & 1 & \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \end{pmatrix} & 1 & \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} & 0 & \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} & 0 & \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} & 1 & \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \end{bmatrix}$$

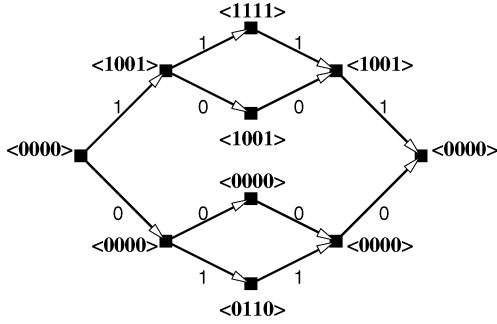


Fig. 17. The minimal conventional Forney trellis for the $(4, 2)_2$ code.

These are projections of codewords that share an all-zero suffix with the all-zero codeword, from index $i + 1$ to index n . Define the future projection at time index i as follows:

$$\mathcal{F}_i = \{(c_{i+1}, \dots, c_n) : \mathbf{c} = (c_1, \dots, c_i, c_{i+1}, \dots, c_n) \in \mathcal{C}, \pi_i(\mathbf{c}) = \mathbf{0}\}. \quad (16)$$

These are projections of codewords that share an all-zero prefix with the all-zero codeword, from index 1 to index i .

Definition 12: Let \mathcal{C}_1 and \mathcal{C}_2 be past and future projections of \mathbb{F}_q^n at some time index i . Then the product of \mathcal{C}_1 and \mathcal{C}_2 , denoted by $\mathcal{C}_1 \times \mathcal{C}_2$, is defined as

$$\mathcal{C}_1 \times \mathcal{C}_2 \stackrel{\text{def}}{=} \{(\mathbf{c}_1, \mathbf{c}_2) : \mathbf{c}_1 \in \mathcal{C}_1, \mathbf{c}_2 \in \mathcal{C}_2\}. \quad (17)$$

It is easy to see that the product $(\mathcal{P}_i \times \mathcal{F}_i)$ is a linear subcode of \mathcal{C} . Therefore $\mathcal{C}/(\mathcal{P}_i \times \mathcal{F}_i)$ forms a quotient space. The Forney trellis $T = (V, E, \mathbb{F}_q)$ for \mathcal{C} is constructed by identifying vertices in V_i with the quotient group corresponding to cosets of \mathcal{C} modulo $(\mathcal{P}_i \times \mathcal{F}_i)$, that is

$$V_i \stackrel{\text{def}}{=} \mathcal{C}/(\mathcal{P}_i \times \mathcal{F}_i) \text{ for } i \in \{1, \dots, n\}. \quad (18)$$

There is an edge from $e \in E_i$ labeled c_i from a vertex $\mathbf{u} \in V_{i-1}$ to a vertex $\mathbf{v} \in V_i$, if and only if there exists a codeword $\mathbf{c} = (c_1, \dots, c_n) \in \mathcal{C}$ such that $\mathbf{c} \in \mathbf{u} \cap \mathbf{v}$. The resulting trellis is isomorphic to the minimal conventional trellis [7].

Example 13: Consider again the self dual $(4, 2)_2$ code \mathcal{C} from Example 3. Let ϵ denote the empty sequence of length 0. The past projections are $\mathcal{P}_0 = \{\epsilon\}$, $\mathcal{P}_1 = \{0\}$, $\mathcal{P}_2 = \{00\}$, $\mathcal{P}_3 = \{000, 011\}$, $\mathcal{P}_4 = \mathcal{C}$. The future projections are $\mathcal{F}_0 = \mathcal{C}$, $\mathcal{F}_1 = \{000, 110\}$, $\mathcal{F}_2 = \{00\}$, $\mathcal{F}_3 = \{0\}$, $\mathcal{F}_4 = \{\epsilon\}$. The subcodes $\mathcal{P}_i \times \mathcal{F}_i$ are thus given by

$$\begin{aligned} \mathcal{P}_0 \times \mathcal{F}_0 &= \{0000, 0110, 1001, 1111\} \\ \mathcal{P}_1 \times \mathcal{F}_1 &= \{0000, 0110\} \\ \mathcal{P}_2 \times \mathcal{F}_2 &= \{0000\} \\ \mathcal{P}_3 \times \mathcal{F}_3 &= \{0000, 0110\} \\ \mathcal{P}_4 \times \mathcal{F}_4 &= \{0000, 0110, 1001, 1111\}. \end{aligned}$$

The minimal conventional trellis for \mathcal{C} resulting from the Forney construction is shown in Fig. 17. The vertices at level i in Fig. 17 are labeled by coset representatives in $\mathcal{C}/(\mathcal{P}_i \times \mathcal{F}_i)$. For example, the vertex $\{1001, 1111\} \in V_1$ is labeled by $\{1001\}$.

This determines the quotient space $\mathcal{C}/(\mathcal{P}_i \times \mathcal{F}_i)$ at all time indices. Therefore we have the vertex sets below (with coset representatives underlined)

$$\begin{aligned} V_0 &= \{\{\underline{0000}\}\} \\ V_1 &= \{\{\underline{0000}, 0110\}, \{\underline{1001}, 1111\}\} \\ V_2 &= \{\{\underline{0000}\}, \{\underline{0110}\}, \{\underline{1001}\}, \{\underline{1111}\}\} \\ V_3 &= \{\{\underline{0000}, 0110\}, \{\underline{1001}, 1111\}\} \\ V_4 &= \{\{\underline{0000}\}\}. \end{aligned}$$

We now generalize the Forney construction to tail-biting trellises. We first state an observation that has been made by several researchers in the past [13], [14], [19].

1) Observation 1: Every linear tail-biting trellis for an $(n, k)_q$ linear code \mathcal{C} can be viewed as a superposition of subtrellises each of which represents a coset of the the code \mathcal{C} with respect to a subcode \mathcal{C}_0 . The subtrellises are all structurally identical to the the subtrellis for \mathcal{C}_0 .

Let

$$\mathbf{G}_{\text{KV}} = \begin{bmatrix} \mathbf{G}_l \\ \mathbf{G}_c \end{bmatrix}$$

be a KV product matrix for a linear block code. Then if \mathcal{C}_0 is the subcode generated by \mathbf{G}_l the coset decomposition is $\mathcal{C}/\mathcal{C}_0$. Each vector g_c in $\langle \mathbf{G}_c \rangle$ is taken to be a coset leader and the path associated with g_c in the trellis shares all states with that corresponding to the all-zero codeword in the zero run of g_c . If $\mathcal{C}_0 + g_l$ denotes the coset \mathcal{C}_l then the subtrellis associated with \mathcal{C}_l shares all states with that associated with \mathcal{C}_0 in the zero run of g_l . The zero run of g_l is called the *merging interval* of subtrellises T_0 and T_l associated with cosets \mathcal{C}_0 and \mathcal{C}_l and is also henceforth referred to as the merging interval of g_l . Note that if the zero run of a coset leader is the empty interval then the subtrellis defined by that coset leader shares no states with that associated with \mathcal{C}_0 .

Our adaptation of Forney's coset construction computes a trellis (henceforth referred to as the T-Forney trellis) that uses a coset decomposition of the code and the associated merging interval for each coset. This as we have seen, depends on the coset leader chosen for each subtrellis. Let \mathcal{C}_0 be a linear subcode of an $(n, k)_q$ code \mathcal{C} . Denote by $\{\mathcal{C}_0, \mathcal{C}_1, \dots, \mathcal{C}_m\}$, the coset decomposition of \mathcal{C} over \mathcal{C}_0 , and choose coset leaders $\mathbf{v}_i, i = 1, 2, \dots, m$, where coset leader \mathbf{v}_i has merging interval $[a_i, b_i]$. For all vectors in coset $\mathcal{C}_l, l \neq 0$, with coset leader \mathbf{v}_l , having merging interval $[a_l, b_l]$ with the all-zero codeword, the past (perhaps more appropriately, the *circular* past) at i begins at time index b_l , wraps around mod n and ends at i . Similarly, a *future* of a vector in \mathcal{C}_l at time index i begins at $i + 1$ and ends at b_l . If the merging interval is empty then pasts and futures are undefined.

Define partial maps $\pi_{il}, 1 \leq i \leq n, 0 \leq l \leq m$, as follows:

$$\pi_{il} : \mathcal{C}_l \rightarrow \pi_{il}(\mathcal{C}_l) \quad (19)$$

where

$$\mathbf{c} = (c_1, \dots, c_n) \mapsto c_{b_l} \mathbf{h}_{b_l} + \dots + c_n \mathbf{h}_n + c_1 \mathbf{h}_1 + \dots + c_i \mathbf{h}_i.$$

The maps π_{il} map codewords into partial syndromes computed with respect to an initial time index b_l instead of 0, and are partial as they are undefined for a coset whose merging interval is empty.

Example 14: Let \mathcal{C} be a $(7, 4)_2$ Hamming code specified by

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}.$$

Define the linear subcode \mathcal{C}_0 as

$$\mathcal{C}_0 = \{0000000, 1000110, 0010111, 1010001\}.$$

Then the rest of the cosets in $\mathcal{C}/\mathcal{C}_0$ are

$$\mathcal{C}_1 = \{0100011, 1100101, 0110100, 1110010\}$$

$$\mathcal{C}_2 = \{0111001, 1111111, 0101110, 1101000\}$$

$$\mathcal{C}_3 = \{0011010, 1011100, 0001101, 1001011\}.$$

Define coset leaders and corresponding merging intervals and spans as follows:

$$\mathbf{v}_1 = 0100011 \quad [2, 5] \quad [6, 2]$$

$$\mathbf{v}_2 = 0111001 \quad [4, 6] \quad [7, 4]$$

$$\mathbf{v}_3 = 0011010 \quad [4, 5] \quad [6, 4]$$

where the first interval alongside the vector is the merging interval, and the second interval is the span. For $i = 3$ and $l = 2$, the map π_{il} is given by

$$\pi_{32}(0111001) = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

$$\pi_{32}(1111111) = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

$$\pi_{32}(0101110) = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

$$\pi_{32}(1101000) = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

For every time index i , we define the past $\mathcal{P}_i(\mathcal{C}_l)$ and future $\mathcal{F}_i(\mathcal{C}_l)$ for vectors in coset \mathcal{C}_l as follows:

$$\mathcal{P}_i(\mathcal{C}_l) = \{(c_1, \dots, c_i) : \mathbf{c} = (c_1, \dots, c_i, c_{i+1}, \dots, c_n) \in \mathcal{C}_l, \pi_{il}(\mathbf{c}) = \mathbf{0}\} \quad (20)$$

$$\mathcal{F}_i(\mathcal{C}_l) = \{(c_{i+1}, \dots, c_n) : \mathbf{c} = (c_1, \dots, c_i, c_{i+1}, \dots, c_n) \in \mathcal{C}_l, \pi_{il}(\mathbf{c}) = \mathbf{0}\}. \quad (21)$$

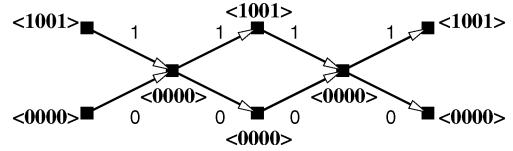


Fig. 18. A T-Forney trellis for the $(4, 2)_2$ code.

If $\pi_{il}(\mathbf{c})$ is undefined then $\mathcal{P}_i(\mathcal{C}_l)$ is the empty set and so is $\mathcal{F}_i(\mathcal{C}_l)$.

Lemma 15: The set $\bigcup_{0 \leq l \leq m} (\mathcal{P}_i(\mathcal{C}_l) \times \mathcal{F}_i(\mathcal{C}_l))$ is a linear subcode of \mathcal{C} .

Proof: Fix a time index i . For cosets \mathcal{C}_r and \mathcal{C}_s , let $\mathbf{c}_1 \in \mathcal{P}_i(\mathcal{C}_r) \times \mathcal{F}_i(\mathcal{C}_r)$ and $\mathbf{c}_2 \in \mathcal{P}_i(\mathcal{C}_s) \times \mathcal{F}_i(\mathcal{C}_s)$. Let $\mathbf{z} = \alpha\mathbf{c}_1 + \beta\mathbf{c}_2$, $\alpha, \beta \in F_q$, belong to coset \mathcal{C}_t . It can easily be verified that $\pi_{it}(\mathbf{z}) = \alpha\pi_{ir}(\mathbf{c}_1) + \beta\pi_{is}(\mathbf{c}_2) = \mathbf{0}$, and therefore $\mathbf{z} \in \mathcal{P}_i(\mathcal{C}_t) \times \mathcal{F}_i(\mathcal{C}_t)$ whenever $\mathcal{P}_i(\mathcal{C}_r) \times \mathcal{F}_i(\mathcal{C}_r)$ and $\mathcal{P}_i(\mathcal{C}_s) \times \mathcal{F}_i(\mathcal{C}_s)$ are nonempty. If either of them is empty the result follows trivially. \square

Definition 13 (T-Forney Trellis): The T-Forney trellis $T = (V, E, F_q)$ for \mathcal{C} is constructed by identifying vertices in V_i with the quotient group corresponding to cosets of \mathcal{C} modulo $\bigcup_{0 \leq l \leq m} (\mathcal{P}_i(\mathcal{C}_l) \times \mathcal{F}_i(\mathcal{C}_l))$, that is

$$V_i = \mathcal{C} / \bigcup_{0 \leq l \leq m} (\mathcal{P}_i(\mathcal{C}_l) \times \mathcal{F}_i(\mathcal{C}_l)). \quad (22)$$

There is an edge from $e \in E_i$ labeled c_i from a vertex $\mathbf{u} \in V_{i-1}$ to a vertex $\mathbf{v} \in V_i$, if and only if there exists a codeword $\mathbf{c} = (c_1, \dots, c_i, \dots, c_n) \in \mathcal{C}$ such that $\mathbf{c} \in \mathbf{u} \cap \mathbf{v}$.

Example 15: Let \mathcal{C} be a self dual $(4, 2)_2$ code with parity check matrix \mathbf{H} and generator matrix \mathbf{G} defined as follows.

$$\mathbf{H} = \mathbf{G} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}.$$

Let the linear subcode of \mathcal{C} be defined as

$$\mathcal{C}_0 = \{0000, 0110\}.$$

Let the coset leader for the coset \mathcal{C}_1 be $\mathbf{v}_1 = 1001$ (with span $[4, 1]$ and merging interval $[1, 3]$). The subcodes $\bigcup_{0 \leq l \leq 1} \mathcal{P}_i(\mathcal{C}_l) \times \mathcal{F}_i(\mathcal{C}_l)$, $0 \leq i \leq 3$, are given by

$$\bigcup_{0 \leq l \leq 1} \mathcal{P}_0(\mathcal{C}_l) \times \mathcal{F}_0(\mathcal{C}_l) = \{0000, 0110\}$$

$$\bigcup_{0 \leq l \leq 1} \mathcal{P}_1(\mathcal{C}_l) \times \mathcal{F}_1(\mathcal{C}_l) = \{0000, 0110, 1001, 1111\}$$

$$\bigcup_{0 \leq l \leq 1} \mathcal{P}_2(\mathcal{C}_l) \times \mathcal{F}_2(\mathcal{C}_l) = \{0000, 0110\}$$

$$\bigcup_{0 \leq l \leq 1} \mathcal{P}_3(\mathcal{C}_l) \times \mathcal{F}_3(\mathcal{C}_l) = \{0000, 0110, 1001, 1111\}.$$

The T-Forney trellis for \mathcal{C} is shown in Fig. 18. The vertices in Fig. 18 are labeled by the coset representatives in V_i , $0 \leq i \leq 3$. For example, the vertex $\{1001, 1111\} \in V_0$ is labeled by $\langle 1001 \rangle$. We observe that since the coset with respect to which

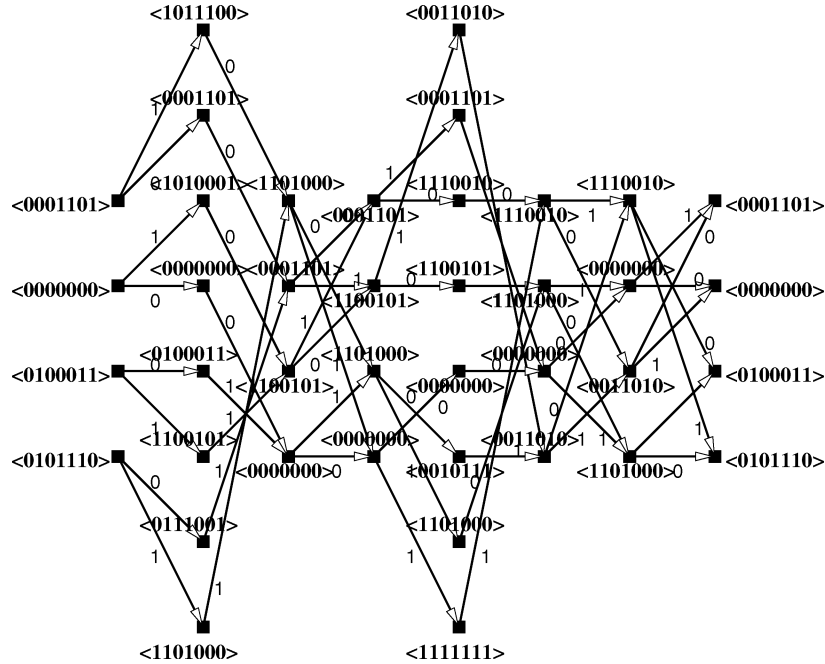


Fig. 19. A T-Forney trellis for the $(7, 4)_2$ Hamming code.

the decomposition is computed is the whole code at time indices 1 and 3, there is a single state at those time indices.

Example 16: Let \mathcal{C} be the $(7, 4)_2$ Hamming code from Example 14. The subcodes $\bigcup_{0 \leq l \leq 2} \mathcal{P}_i(\mathcal{C}_l) \times \mathcal{F}_i(\mathcal{C}_l)$, $0 \leq i \leq 6$, are given by

$$\begin{aligned}
 & \bigcup_{0 \leq l \leq 2} \mathcal{P}_0(\mathcal{C}_l) \times \mathcal{F}_0(\mathcal{C}_l) \\
 &= \{0000000, 0010111, 1010001, 1000110\} \\
 & \bigcup_{0 \leq l \leq 2} \mathcal{P}_1(\mathcal{C}_l) \times \mathcal{F}_1(\mathcal{C}_l) \\
 &= \{0000000, 0010111\} \\
 & \bigcup_{0 \leq l \leq 2} \mathcal{P}_2(\mathcal{C}_l) \times \mathcal{F}_2(\mathcal{C}_l) \\
 &= \{0000000, 0010111, 0110100, 0100011\} \\
 & \bigcup_{0 \leq l \leq 2} \mathcal{P}_3(\mathcal{C}_l) \times \mathcal{F}_3(\mathcal{C}_l) \\
 &= \{0000000, 1111111, 1011100, 0100011\} \\
 & \bigcup_{0 \leq l \leq 2} \mathcal{P}_4(\mathcal{C}_l) \times \mathcal{F}_4(\mathcal{C}_l) \\
 &= \{0000000, 0100011\} \\
 & \bigcup_{0 \leq l \leq 2} \mathcal{P}_5(\mathcal{C}_l) \times \mathcal{F}_5(\mathcal{C}_l) \\
 &= \{0000000, 0001101, 0100011, 0101110\} \\
 & \bigcup_{0 \leq l \leq 2} \mathcal{P}_6(\mathcal{C}_l) \times \mathcal{F}_6(\mathcal{C}_l) \\
 &= \{0000000, 0001101, 1001011, 1000110\}.
 \end{aligned}$$

The T-Forney trellis for \mathcal{C} is shown in Fig. 19. The vertices in Fig. 19 are labeled by the coset representatives in V_i , $0 \leq i \leq 6$. For example, the vertex $\{1101000, 1111111\} \in V_1$ is labeled by $\langle 1101000 \rangle$.

Note that this trellis is isomorphic to the KV trellis generated by the matrix

$$\mathbf{G}_{\text{KV}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{matrix} [1, 6] \\ [2, 7] \\ [6, 2] \\ [7, 4] \end{matrix}$$

where the intervals on the right represent the spans selected.

The corresponding T-BCJR representation has the same generator matrix, with the parity check matrix of Example 14 and the \mathfrak{D} matrix below

$$\mathfrak{D} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}.$$

The following theorem relates nonmergeable KV trellises, T-BCJR and T-Forney trellises.

Theorem 5.1: Given an $(n, k)_q$ code \mathcal{C} with generator matrix

$$\mathbf{G}_{\text{KV}} = \begin{bmatrix} \mathbf{G}_l \\ \mathbf{G}_c \end{bmatrix}$$

generating a nonmergeable trellis, parity check matrix \mathbf{H} and displacement matrix \mathfrak{D} computed according to Definition 9, the T-BCJR the T-Forney, and the KV trellises are isomorphic to one another if coset leaders are chosen to be elements of $\langle \mathbf{G}_c \rangle$.

Proof: As we saw in Definition 9,

$$\mathbf{G}_{\text{KV}} = \begin{bmatrix} \mathbf{G}_l \\ \mathbf{G}_c \end{bmatrix}$$

determines \mathfrak{D} which in turn defines a coset decomposition of the code \mathcal{C} . We refer to this decomposition as the \mathfrak{D} -coset

decomposition to distinguish it from the coset decomposition induced by the Forney definition. Let $\mathcal{C}_0, \mathcal{C}_1, \dots, \mathcal{C}_m$ be the cosets of the \mathfrak{D} -coset decomposition. We refer to the T-BCJR label induced by a codeword prefix c as T-BCJR(c). Let $G_i = \bigcup_{0 \leq l \leq m} (\mathcal{P}_i(\mathcal{C}_l) \times \mathcal{F}_i(\mathcal{C}_l))$, where $\mathcal{P}_i(\mathcal{C}_l)$ and $\mathcal{F}_i(\mathcal{C}_l)$ are as defined in (20) and (21) choosing coset leaders as specified in the statement of the theorem. Denote the coset decomposition \mathcal{C}/G_i as the i -coset decomposition of \mathcal{C} . Denote the coset containing \mathbf{g} in the i -coset decomposition by $\mathbf{g} + G_i$. Let $\mathbf{g} = a_1 \mathbf{g}_1 + a_2 \mathbf{g}_2 \dots + a_k \mathbf{g}_k$ where $\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_k$ are the k rows of the generator matrix. Let $\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_k$ be the displacement vectors for $\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_k$ respectively. Denote the T-BCJR label for vector \mathbf{g} at level i by \mathbf{b} . Define the map

$$\mathbf{g} + G_i \mapsto \mathbf{b}.$$

We show that the mapping is an isomorphism. Assume that codewords $\mathbf{c} = (c_1, \dots, c_n)$ and $\mathbf{c}' = (c'_1, c'_2, \dots, c'_n)$ induce the same T-BCJR label at time index i . Then their difference $\mathbf{c}'' = \mathbf{c} - \mathbf{c}'$ induces the all-zero label at time index i . Hence, $\mathbf{c}'' \in G_i$. Therefore \mathbf{c} and \mathbf{c}' belong to the same coset in the i -coset decomposition and therefore lead to the same state in the T-Forney trellis.

Conversely assume that \mathbf{c} and \mathbf{c}' belong to the same coset in the i -coset decomposition of \mathcal{C} . Hence their difference \mathbf{c}'' belongs to G_i in the i -coset decomposition and therefore $\mathbf{c}'' + G_i$ corresponds to the $\mathbf{0}$ state and has partial syndrome $\mathbf{0}$ at level i . Hence the T-BCJR label of \mathbf{c}'' at level i is $\mathbf{0}$, which in turn implies that the T-BCJR labels of prefixes of \mathbf{c} and \mathbf{c}' at level i are the same.

Next, let $(\mathbf{b}, a, \mathbf{b}')$ be an edge in the T-BCJR trellis from index i to index $i + 1$. Then there exists a codeword $\mathbf{c} = (c_1, c_2, \dots, c_i, \dots, c_n)$ such that T-BCJR(c_1, \dots, c_i) = \mathbf{b} , T-BCJR(c_1, \dots, c_{i+1}) = \mathbf{b}' , and $c_{i+1} = a$. The codeword \mathbf{c} is in coset $\mathbf{c} + G_i$ of the i -coset decomposition as well as in $\mathbf{c} + G_{i+1}$ of the $i + 1$ -coset decomposition. Hence by definition there is an edge labeled c_{i+1} from the state represented by $\mathbf{c} + G_i$ to that represented by $\mathbf{c} + G_{i+1}$ in the T-Forney trellis. Conversely, assume that there is an edge in the T-Forney trellis from vertex associated with coset $\mathbf{c} + G_i$ in the i -coset decomposition to that associated with coset $\mathbf{c}' + G_{i+1}$ in the $i + 1$ -coset decomposition with label a . Then by Definition 13, there exists a codeword \mathbf{c}'' such that $\mathbf{c} + G_i = \mathbf{c}'' + G_i$ and $\mathbf{c}' + G_{i+1} = \mathbf{c}'' + G_{i+1}$ and the i^{th} symbol of the codeword is a . Let $\mathbf{c}'' + G_i$ map to \mathbf{b} and $\mathbf{c}'' + G_{i+1}$ map to \mathbf{b}' . By definition there is an edge from \mathbf{b} to \mathbf{b}' on label a establishing the isomorphism between the T-Forney and the T-BCJR trellis. Since the nonmergeable KV trellis is isomorphic to the T-BCJR trellis by Theorem 3.1, the theorem follows. \square

VI. CONCLUSION

We have given a specification for tail-biting trellises which is a generalization of the BCJR construction for conventional trellises for linear block codes. The generalized BCJR construc-

tion begins with an arbitrary parity check matrix \mathbf{H} and a matrix \mathfrak{D} which we term the displacement matrix, and proceeds to produce a labeled tail-biting trellis whose state spaces are vector spaces. It is easy to get dual trellises using this specification; these have state-complexity profiles which are identical to those of the primal trellises. Adaptation of the algebraic Forney construction which views trellis state spaces as quotient spaces with respect to a coset decomposition is also shown to be possible with a redefinition of “pasts” and “futures.” The most interesting problem which to our knowledge is as yet open, is that of finding a polynomial time algorithm for obtaining a minimal trellis which minimizes the maximum state-complexity over all time indices.

ACKNOWLEDGMENT

The authors would like to thank the anonymous referees for their valuable comments.

REFERENCES

- [1] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, “Optimal decoding of linear codes for minimizing symbol error rate,” *IEEE Trans. Inf. Theory*, vol. IT-20, no. 2, pp. 284–287, Mar. 1974.
- [2] G. D. Forney Jr., “Coset codes II: Binary lattices and related codes,” *IEEE Trans. Inf. Theory*, vol. 34, no. 6, pp. 1152–1187, Sep. 1988.
- [3] F. Kschischang and V. Sorokine, “On the trellis structure of block codes,” *IEEE Trans. Inf. Theory*, vol. 41, pp. 1924–1937, Nov. 1995.
- [4] F. Kschischang, “The trellis structure of maximal fixed-cost codes,” *IEEE Trans. Inf. Theory*, vol. 42, pp. 1828–1838, Nov. 1996.
- [5] J. Massey, “Foundations and methods of channel encoding,” in *Proc. Int. Conf. Theory Syst.*, Berlin, 1978, pp. 148–157.
- [6] D. J. Muder, “Minimal trellises for block codes,” *IEEE Trans. Inf. Theory*, vol. 34, no. 6, pp. 1049–1053, Sep. 1988.
- [7] A. Vardy, *Trellis Structure of Codes*, V. Pless and W. Huffman, Eds. Amsterdam, The Netherlands: Elsevier, 1998.
- [8] A. Calderbank, G. D. Forney Jr., and A. Vardy, “Optimal decoding of linear codes for minimizing symbol error rate,” *IEEE Trans. Inf. Theory*, vol. 45, no. 5, pp. 1435–1455, Jul. 1999.
- [9] R. Koetter and A. Vardy, *On the Theory of Linear Trellises*, M. Blaum, P. Farrel, and H. van Tilborg, Eds. Boston, MA: Kluwer, 2002.
- [10] —, “The structure of tail-biting trellises: Minimality and basic principles,” *IEEE Trans. Inf. Theory*, vol. 49, no. 9, pp. 1877–1901, Sep. 2003.
- [11] I. Reuven and Y. Be’ery, “Tail-biting trellises of block codes: Trellis complexity and viterbi decoding complexity,” *IEICE Trans. Fundamentals*, vol. E82-A, no. 10, Oct. 1999.
- [12] P. Shankar, P. N. A. Kumar, H. Singh, and B. S. Rajan, “Minimal tail-biting trellises for certain cyclic block codes are easy to construct,” *ICALP*, pp. 627–638, 2001.
- [13] P. Shankar, A. Dasgupta, K. Deshmukh, and B. Rajan, “On viewing block codes as finite automata,” *Theor. Comput. Sci.*, vol. 290, no. 3, pp. 1775–1795, 2003.
- [14] Y. Shany and Y. Be’ery, “Linear tail-biting trellises, the square-root bound and applications for Reed-Muller codes,” *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1514–1523, Jul. 2000.
- [15] R. McEliece, “On the bcjr trellis for linear block codes,” *IEEE Trans. Inf. Theory*, vol. 42, no. 4, pp. 1072–1092, Jul. 1996.
- [16] A. Vardy and F. Kschischang, “Proof of a conjecture of mceliece regarding the expansion index of the minimal trellis,” *IEEE Trans. Inf. Theory*, vol. 42, no. 6, pp. 2027–2034, Nov. 1996.
- [17] J. Wolf, “Efficient maximum-likelihood decoding of linear block codes using a trellis,” *IEEE Trans. Inf. Theory*, vol. 24, pp. 76–80, 1978.
- [18] K. Hoffman and R. Kunze, *Linear Algebra*. N.J.: Prentice, 1961.
- [19] S. Lin and R. Y. Shao, “General structure and construction of tail-biting trellises for linear block codes,” in *Proc. IEEE Int. Symp. Information Theory*, Sorrento, Italy, Jun./Jul. 2000, p. 117.