International Symposium on Information Theory and its Applications, ISITA2004
Parma, Italy, October 10–13, 2004

A Coset Construction for Tail-Biting Trellises

Aditya Nori and Priti Shankar

Department of Computer Science and Automation
Indian Institute of Science Bangalore, India 560012
E-mail: {aditya, priti}@csa.iisc.ernet.in

Abstract

A coset construction is presented for tail-biting trellises which for a given linear block code $\mathcal{C}$ computes a non-mergeable linear trellis representing $\mathcal{C}$. This construction is analogous to the Forney construction for minimal conventional trellises. Unlike the case of conventional trellises where the Forney construction and the Kschischang-Sorokine (KS) product construction give isomorphic trellises, the generalization of the Forney construction presented here and the Koetter-Vardy adaptation of the KS construction do not give isomorphic trellises in general.

1. INTRODUCTION

Tail-biting trellises are compact combinatorial descriptions of block codes. Though conventional trellises for block codes have an elegant underlying theory [9], the theory of tail-biting trellises appears to be fairly complex. Several advances in the understanding of the structure and properties of such trellises have been made in recent years [2, 4, 7, 8]. More recently, Koetter and Vardy [4] have made a detailed study of the structure of linear tail-biting trellis and have also shown how the product construction of Kschischang and Sorokine [5] can be profitably exploited for the construction of tail-biting trellises. In this paper, we show that we can use a suitable modification of the Forney construction [3] of the minimal conventional trellis, for producing non-mergeable tail-biting trellises as well. However, this gives a trellis that is non-isomorphic to that obtained from the Koetter-Vardy (KV) product construction [4] if the latter gives a mergeable trellis.

2. PRELIMINARIES

In this section, we review some concepts related to tail-biting trellises borrowing definitions from [4].

Priti Shankar acknowledges support from the Scientific Analysis Group, DRDO, India.

Definition 2.1 *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*

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

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 $V_{i+1(mod\ n)}$, for some $i \in \{0, 1, \ldots, n-1\}$.

The set of indices $\mathcal{I} = \{0, 1, \ldots, n-1\}$ for the partition in (1) are the *time indices*. We will refer to $\log_{|\Sigma|}|V_i|$ as the *state-complexity* of the trellis at time index i and the sequence $\left\{\log_{|\Sigma|}|V_i|,\ 0 \le i \le n\right\}$ as the *state-complexity profile* of the trellis. We identify $\mathcal{I}$ with $\mathbb{Z}_n$, the residue classes of integers modulo n. An interval of indices $[i, j]$ represents the sequence $\{i, i+1, \ldots j\}$ if $i < j$, and the sequence $\{i, i+1, \ldots n-1, 0, \ldots j\}$ if $i > j$. Every cycle in T starting at a vertex of V_0 defines a vector $(a_1, a_2, \ldots, a_n) \in \Sigma^n$ 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 *represents* 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. In addition to the labeling of edges, each vertex in the set V_i is labeled by a sequence of length $l_i \ge \lceil \log_{|\Sigma|}|V_i| \rceil$ of elements in Σ , all vertex labels at a given depth being distinct. Thus every cycle in this labeled trellis defines sequences of length $n + l$ (where $l = l_1 + l_2 + \cdots + l_n$) over Σ, consisting of alternating labels of vertices and edges in T. This sequence is called the *label sequence* of 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 $\mathcal{S}(T)$. 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. T is *non-mergeable* [6] if there do not 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 T is non-mergeable, then it is also biproper – though the converse is true for conventional trellises, it is not true in general for tail-biting trellises [4]. In the discussion that follows, we restrict ourselves to trellises representing linear block codes over the alphabet $\Sigma = \mathbb{F}_q$. We will also be required to define the notion of a span. Given a codeword $\mathbf{c} \in \mathcal{C}$, the *linear span* of $\mathbf{c}$, is the interval $[i, j] \in \mathcal{I}$, $[i, j]$, $j > i$, which contains all the non-zero 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 $\mathcal{I}$.

3. The COSET CONSTRUCTION

The Forney construction [3] for a conventional trellis produces a minimal trellis, whereas the Coset Construction that we will describe (henceforth referred to as the CC tail-biting trellis) in this section computes a non-mergeable linear trellis. The construction is intimately tied up with the generator matrix with which we begin. We modify the concepts of past and future subcodes defined in [3] to take into account vectors of circular span. Let $\mathcal{C}$ be an (n, k) linear code over $\mathbb{F}_q$ with generator matrix G and parity check matrix $H = [\mathbf{h}_1 \dots \mathbf{h}_n]$. Define a map $\pi_i : \mathcal{C} \to \pi_i(\mathcal{C})$, defined by $\mathbf{c} = (c_1, \dots, c_n) \mapsto c_1\mathbf{h}_1 + \cdots + c_i\mathbf{h}_i$, and also a map $\tau_i : \mathcal{C} \to \tau_i(\mathcal{C})$, defined by
$\mathbf{c} = (c_1, \dots, c_n) \mapsto c_{i+1}\mathbf{h}_{i+1} + \cdots + c_n\mathbf{h}_n$.
Define the *past subcode* $\mathcal{P}_i =$
$\{(c_1, \dots, c_i) : \mathbf{c} = (c_1, \dots, c_i, c_{i+1}, \dots, c_n) \in \mathcal{C},\ \pi_i(\mathbf{c}) = \mathbf{0}\}$
and the *future subcode* $\mathcal{F}_i =$
$\{(c_{i+1}, \dots, c_n) : \mathbf{c} = (c_1, \dots, c_i, c_{i+1}, \dots, c_n) \in \mathcal{C},\ \tau_i(\mathbf{c}) = \mathbf{0}\}$.
The Forney conventional trellis $T = (V, E, \mathbb{F}_q)$ for $\mathcal{C}$ is constructed by identifying vertices in V_i with cosets of $\mathcal{C}$ modulo $(\mathcal{P}_i \times \mathcal{F}_i)$, that is, $V_i = \mathcal{C}/(\mathcal{P}_i \times \mathcal{F}_i)$ for $i \in \{1, \dots, n\}$. There is an edge from $e \in E_i$ labeled c_i from a vertex $\mathbf{u} \in V_{i-1}$ to a vertex $\mathbf{v} \in V_i$, iff there exists a codeword $\mathbf{c} = (c_1, \dots, c_n) \in \mathcal{C}$ such that $\mathbf{c} \in \mathbf{u} \cap \mathbf{v}$. In order to define the CC tail-biting trellis, we will be required to slightly modify the definitions of the maps π_i and τ_i as follows. $\pi_i : \mathcal{C} \to \pi_i(\mathcal{C})$, defined by $\mathbf{c} = (c_1, \dots, c_n) \mapsto c_1\mathbf{h}_1 + \cdots + c_i\mathbf{h}_i + \mathbf{d}_\mathbf{c}$ $\tau_i : \mathcal{C} \to \tau_i(\mathcal{C})$,
defined by $\mathbf{c} = (c_1, \dots, c_n) \mapsto c_{i+1}\mathbf{h}_{i+1} + \cdots + c_n\mathbf{h}_n - \mathbf{d}_\mathbf{c}$, where

$$\mathbf{d}_\mathbf{c} = \begin{cases} \mathbf{0} & \text{if } \mathbf{c} \text{ is a row of } G \text{ with linear span} \\ \sum_{j=a}^{n} c_i\mathbf{h}_i & \text{if } \mathbf{c} \text{ is row of } G \text{ with circular span } [a,b] \\ \sum_{i=1}^{k} \alpha_i \mathbf{d}_{\mathbf{g}_i} & \text{otherwise, where } \mathbf{c} = \sum_{i=1}^{k} \alpha_i \mathbf{g}_i, \\ & \alpha_i \in \mathbb{F}_q,\ \mathbf{g}_i \in G,\ 1 \le i \le k. \end{cases}$$

Let D be the subspace consisting of all possible $\mathbf{d}$ vectors and define past subcodes $\mathcal{P}_i(\mathbf{d}) =$
$\{(c_1, \dots, c_i) : \mathbf{c} = (c_1, \dots, c_i, c_{i+1}, \dots, c_n) \in \mathcal{C},\ \mathbf{d} = \mathbf{d}_\mathbf{c},\ \pi_i(\mathbf{c}) = \mathbf{0}\}$
and future subcodes $\mathcal{F}_i(\mathbf{d}) =$
$\{(c_1, \dots, c_i) : \mathbf{c} = (c_1, \dots, c_i, c_{i+1}, \dots, c_n) \in \mathcal{C},\ \mathbf{d} = \mathbf{d}_\mathbf{c},\ \tau_i(\mathbf{c}) = \mathbf{0}\}$,
$\forall \mathbf{d} \in D$. The set of vertices V_i, $0 \le i \le n$, at level i is given by $V_i = \mathcal{C}/\left(\bigcup_{\mathbf{d} \in D} \mathcal{P}_i(\mathbf{d}) \times \mathcal{F}_i(\mathbf{d})\right)$. The condition for edge placement is the same as that for the Forney conventional trellis.

Example 1 *Consider the $(7, 4)$ Hamming code defined by the parity check matrix H and generator matrix G (annotated with spans).*

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} \begin{matrix} [1,6] \\ [6,2] \\ [3,7] \\ [7,5] \end{matrix}$$

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

The past and future subcode of $\mathcal{C}$ are given as follows: $\mathcal{P}_0 = \phi$, $\mathcal{P}_1 = \{0\}$, $\mathcal{P}_2 = \{00, 01\}$, $\mathcal{P}_3 = \{000, 111, 101, 010\}$, $\mathcal{P}_4 = \{0000, 0100\}$, $\mathcal{P}_5 = \{00000, 00011, 01000, 01011\}$, $\mathcal{P}_6 = \{000000, 000110, 100101, 100011\}$, $\mathcal{P}_7 = \{0000000, 0010111, 1010001, 1000110\}$, $\mathcal{F}_0 = \{0000000, 0010111, 1010001, 1000110\}$, $\mathcal{F}_1 = \{000000, 010111\}$, $\mathcal{F}_2 = \{00000, 10100, 00011, 10111\}$, $\mathcal{F}_3 = \{0000, 1111, 1100, 0011\}$, $\mathcal{F}_4 = \{000, 011\}$, $\mathcal{F}_5 = \{00, 01, 11, 10\}$, $\mathcal{F}_6 = \{0, 1\}$, $\mathcal{F}_7 = \phi$.

The subcodes $\bigcup_{\mathbf{d} \in D} \mathcal{P}_i(\mathbf{d}) \times \mathcal{F}_i(\mathbf{d})$, $0 \le i \le 7$ are thus given by:

$\bigcup_{\mathbf{d} \in D} \mathcal{P}_0(\mathbf{d}) \times \mathcal{F}_0(\mathbf{d}) =$
$\{0000000, 0010111, 1010001, 1000110\}$,
$\bigcup_{\mathbf{d} \in D} \mathcal{P}_1(\mathbf{d}) \times \mathcal{F}_1(\mathbf{d}) = \{0000000, 0010111\}$,
$\bigcup_{\mathbf{d} \in D} \mathcal{P}_2(\mathbf{d}) \times \mathcal{F}_2(\mathbf{d}) =$
$\{0000000, 0010111, 0110100, 0100011\}$,
$\bigcup_{\mathbf{d} \in D} \mathcal{P}_3(\mathbf{d}) \times \mathcal{F}_3(\mathbf{d}) =$
$\{0000000, 1111111, 1011100, 0100011\}$,
$\bigcup_{\mathbf{d} \in D} \mathcal{P}_4(\mathbf{d}) \times \mathcal{F}_4(\mathbf{d}) = \{0000000, 0100011\}$,
$\bigcup_{\mathbf{d} \in D} \mathcal{P}_5(\mathbf{d}) \times \mathcal{F}_5(\mathbf{d}) =$
$\{0000000, 0001101, 0100011, 0101110\}$,
$\bigcup_{\mathbf{d} \in D} \mathcal{P}_6(\mathbf{d}) \times \mathcal{F}_6(\mathbf{d}) =$
$\{0000000, 0001101, 1001011, 1000110\}$,
$\bigcup_{\mathbf{d} \in D} \mathcal{P}_7(\mathbf{d}) \times \mathcal{F}_7(\mathbf{d}) =$
$\{0000000, 0010111, 1010001, 1000110\}$.

We can now determine the coset structure $\bigcup_{\mathbf{d} \in D} \mathcal{C}/(\mathcal{P}_i(\mathbf{d}) \times \mathcal{F}_i(\mathbf{d}))$, $0 \le i \le 7$, *and therefore we* *have*

$$V_0 = \Big\{\{\underline{0000000}, 0010111, 1010001, 1000110\}, \{\underline{0100011}, 1100101, 0110100, 1110010\}, \{\underline{0001101}, 1001011, 0011010, 1011100\}, \{\underline{0101110}, 1101000, 0111001, 1111111\}\Big\},$$

$$V_1 = \Big\{\{\underline{0000000}, 0010111\}, \{\underline{0001101}, 0011010\}, \{\underline{1101000}, 1111111\}, \{\underline{1100101}, 1110010\}, \{\underline{1010001}, 1000110\}, \{\underline{1011100}, 1001011\}, \{\underline{0111001}, 0101110\}, \{\underline{0100011}, 0110100\}\Big\},$$

$$V_2 = \Big\{\{\underline{0000000}, 0010111, 0110100, 0100011\}, \{\underline{0001101}, 0011010, 0111001, 0101110\}, \{\underline{1101000}, 1111111, 1011100, 1001011\}, \{\underline{1100101}, 1110010, 1010001, 1000110\}\Big\},$$

$$V_3 = \Big\{\{\underline{0000000}, 1111111, 1011100, 0100011\}, \{\underline{0001101}, 1110010, 1010001, 0101110\}, \{\underline{1101000}, 0010111, 0110100, 1001011\}, \{\underline{1100101}, 0011010, 0111001, 1000110\}\Big\},$$

$$V_4 = \Big\{\{\underline{0000000}, 0100011\}, \{\underline{0001101}, 0101110\}, \{\underline{1101000}, 1001011\}, \{\underline{1100101}, 1000110\}, \{\underline{0011010}, 0111001\}, \{\underline{0010111}, 0110100\}, \{\underline{1110010}, 1010001\}, \{\underline{1111111}, 1011100\}\Big\},$$

$$V_5 = \Big\{\{\underline{0000000}, 0001101, 0100011, 0101110\}, \{\underline{1101000}, 1100101, 1001011, 1000110\}, \{\underline{0011010}, 0010111, 0111001, 0110100\}, \{\underline{1110010}, 1111111, 1010001, 1011100\}\Big\},$$

$$V_6 = \Big\{\{\underline{0000000}, 0001101, 1001011, 1000110\}, \{\underline{1101000}, 1100101, 0100011, 0101110\}, \{\underline{0011010}, 0010111, 1010001, 1011100\}, \{\underline{1110010}, 1111111, 0111001, 0110100\}\Big\},$$

$$V_7 = \Big\{\{\underline{0000000}, 0010111, 1010001, 1000110\}, \{\underline{0100011}, 1100101, 0110100, 1110010\}, \{\underline{0001101}, 1001011, 0011010, 1011100\}, \{\underline{0101110}, 1101000, 0111001, 1111111\}\Big\}.$$

The resulting trellis for the code is shown in Figure 1. The vertices of the trellis are labeled by the representatives (which are underlined above) of the corresponding cosets in $\bigcup_{\mathbf{d}\in D} \mathcal{C}/(\mathcal{P}_i(\mathbf{d}) \times \mathcal{F}_i(\mathbf{d}))$, $0 \le i \le n$.

We will now state without proof our main results.

Lemma 3.1 *Given an (n,k) code $\mathcal{C}$ with generator matrix G and parity check matrix H, the* KV *and* CC *tail-biting trellises are isomorphic to each other, iff the* KV *trellis is non-mergeable.*

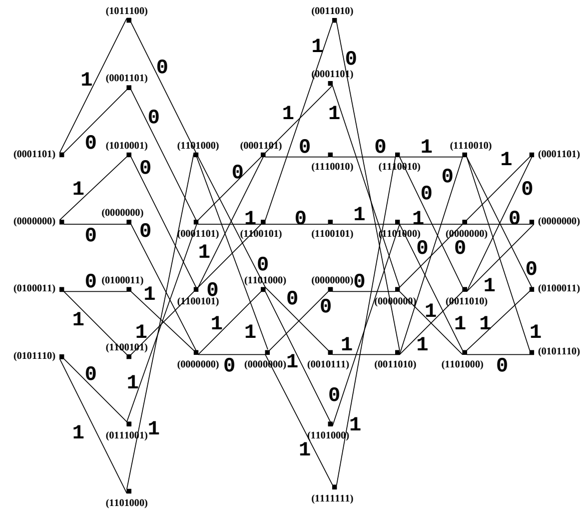
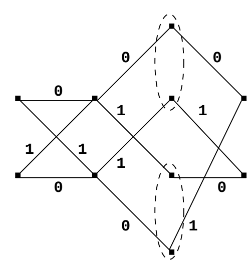


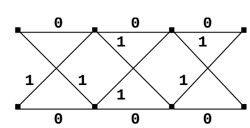
Figure 1: A CC tail-biting trellis for the $(7,4)$ Hamming code with generator matrix G

Lemma 3.2 *The* CC *tail-biting trellis is a non-mergeable linear trellis.*

The following example illustrates the difference between the CC construction and the Koetter-Vardy (KV) construction [4].



(a)



(b)

Figure 2: The KV and the CC trellises for the $(3,2)$ code in Example 2

Example 2 *Consider a $(3,2)$ code with generator matrix G' and parity check matrix H' defined as follows:*

$$G' = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \begin{matrix} [1,3] \\ [2,1] \end{matrix} \qquad H' = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$$

The KV *tail-biting trellis for this code is a mergeable linear trellis and is shown in Figure 3 (a) – the mergeable vertices are indicated by dotted lines. In contrast, the* CC *tail-biting trellis shown in Figure 3 (b) is non-mergeable.*

References

[1] L.R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, Optimal decoding of linear codes for minimizing symbol error rate, *IEEE Trans. Inform. Theory*, **20**(2), March 1974, pp. 284–287.

[2] A.R. Calderbank, G.D. Forney, Jr., and A. Vardy, Minimal Tail-Biting Trellises: The Golay Code and More, *IEEE Trans. Inform. Theory*, **45**(5), July 1999, pp. 1435–1455.

[3] G. D. Forney, Jr., Coset codes II: Binary lattices and related codes, *IEEE Trans. Inform. Theory*, **34**, September 1988, pp. 1152–1187.

[4] R. Koetter and A. Vardy, The Structure of Tail-Biting Trellises: Minimality and Basic Principles, *IEEE Trans. Inform. Theory*, **49**, September 2003, pp. 1877–1901.

[5] F.R. Kschischang and V. Sorokine, On the trellis structure of block codes, *IEEE Trans. Inform. Theory*, **41**(6), November 1995, pp. 1924–1937.

[6] F.R. Kschischang, The trellis structure of maximal fixed-cost codes, *IEEE Trans. Inform. Theory*, **42**, 1996, pp. 1828–1838.

[7] A. Nori and P. Shankar, A BCJR-like Labeling Algorithm for Tail-Biting Trellises, *Proceedings of the IEEE International Symposium on Information Theory*, Yokohama, Japan, July 2003.

[8] A. Nori and P. Shankar, Tail-Biting Trellises for Linear Codes and their Duals, *Forty-First Annual Allerton Conference on Communication, Control and Computing, Allerton*, IL, USA, October 2003.

[9] A. Vardy, Trellis structure of codes, *Handbook of Coding Theory*, V.S. Pless and W.C. Huffman, Elsevier, 1998.