

via time slices which involve only nine information symbols, viz. $u_t, u_{t-1}, \dots, u_{t-8}$. From the time slices for the phases that are multiples of 4 it follows that information symbol u_{t-8} does not contribute to the corresponding output. This information symbol does not contribute to the outputs in the following time instants either; hence, we do not have to store it! Thus the Kötter–Vardy convolutional encoder can be realized by only eight memory elements [11], although not in controller canonical form. (When we use time-varying convolutional codes to prove ensemble properties we consider realizations in controller canonical form with feedforward shift registers and time-varying connections [3]. For example, the Golay convolutional code can be encoded by a rate $R = 1/2$ time-varying encoder with four delay elements in controller canonical form.)

We showed that the Kötter–Vardy convolutional code has $d_{\text{free}} = 12$ and its first spectral components are

phase	n_{12}	n_{16}	n_{20}	n_{24}
1	34	1194	38966	1311243
2	23	678	22724	763371
3	30	834	28438	952966
4	12	381	12882	431568
5	54	1700	56924	1908782
6	22	712	23692	792423
7	21	570	19682	656667
8	13	328	11434	382063
$\sum_{i=1}^8 n_d$	209	6397	214742	7199083
$\frac{1}{8} \sum_{i=1}^8 n_d$	26.125	799.625	26842.75	899885.375

(28)

Apart from the Golay convolutional code, the Kötter–Vardy convolutional code is the only Type II, rate $R = 1/2$, convolutional code known to us. Calderbank, Forney, and Vardy proved that a Type II, binary, *time-invariant* convolutional code of rate $R = 1/2$ does not exist ([1, Lemma 4]).

The best rate $R = 1/2$, time-invariant convolutional code of memory $m = 8$ has [12]

$$T(W) = 10W^{12} + 9W^{13} + 30W^{14} + 51W^{15} + 156W^{16} + \dots \quad (29)$$

which is better than $T_{KV}(W)$ for high signal-to-noise ratios but worse for low signal-to-noise ratios, since $n_{13} = n_{14} = n_{15} = 0$ in $T_{KV}(W)$.

ACKNOWLEDGMENT

The authors wish to thank G. D. Forney. Not only did he suggest the problem and act as a clearing house for the “Type II News Group,” but also, perhaps most importantly, he provided constant encouragement during their search for a needle in a (huge) haystack. They are also grateful to N. J. A. Sloane for identifying \mathcal{B}_2 and to R. Kötter for an illuminating discussion on the Kötter–Vardy convolutional code.

REFERENCES

- [1] A. R. Calderbank, G. D. Forney, Jr., and A. Vardy, “Minimal tail-biting trellises: The Golay code and more,” *IEEE Trans. Inform. Theory*, vol. 45, pp. 1435–1455, July 1999.
- [2] G. D. Forney, Jr., “The forward-backward algorithm,” in *Proc. 34th Allerton Conf. Communication, Control, and Computing*, Monticello, IL, Oct. 1996, pp. 432–446.
- [3] R. Johannesson and K. Sh. Zigangirov, *Fundamentals of Convolutional Coding*. Piscataway, NJ: IEEE Press, 1999.
- [4] E. M. Rains and N. J. A. Sloane, “Self-Dual Codes,” in *Handbook of Coding Theory*, V. S. Pless and W. C. Huffman, Eds. Amsterdam, The Netherlands: Elsevier Science, 1998.
- [5] R. Kötter and A. Vardy, private communication, Jan. 16, 1998.
- [6] ———, Tail-biting trellises, Part II: Bounds and applications, preprint.
- [7] G. D. Forney, Jr., “Dimension/length profiles and trellis complexity of linear block codes,” *IEEE Trans. Inform. Theory*, vol. 40, pp. 1741–1752, Nov. 1994.
- [8] S. Lin, T. Kasami, T. Fujiwara, and M. Fossorier, *Trellises and Trellis-based Decoding Algorithms for Linear Block Codes*. Boston, MA: Kluwer Academic, 1998.
- [9] N. J. A. Sloane, private communication, Dec. 2, 1997.
- [10] J. H. Conway, V. Pless, and N. J. A. Sloane, “The binary self-dual codes of length up to 32: A revised enumeration,” *J. Comb. Theory*, ser. A, vol. 60, pp. 183–195, 1992.
- [11] R. Kötter, private communication, Feb. 9, 1999.
- [12] R. Johannesson and P. Ståhl, “New rate 1/2, 1/3, and 1/4 binary convolutional encoders with an optimum distance profile,” *IEEE Trans. Inform. Theory*, vol. 45, pp. 1653–1658, July 1999.

Linear Tail-Biting Trellises, the Square-Root Bound, and Applications for Reed–Muller Codes

Yaron Shany and Yair Be’ery, *Senior Member, IEEE*

Abstract—Linear tail-biting trellises for block codes are considered. By introducing the notions of subtrellis, merging interval, and sub-tail-biting trellis, some structural properties of linear tail-biting trellises are proved. It is shown that a linear tail-biting trellis always has a certain simple structure, the parallel-merged-cosets structure. A necessary condition required from a linear code in order to have a linear tail-biting trellis representation that achieves the square-root bound is presented. Finally, the above condition is used to show that for $r \geq 2$ and $m \geq 4r - 1$ or $r \geq 4$ and

$$r + 3 \leq m \leq \lfloor (4r + 5)/3 \rfloor$$

the Reed–Muller code $RM(r, m)$ under any bit order cannot be represented by a linear tail-biting trellis whose state complexity is half of that of the minimal (conventional) trellis for the code under the standard bit order.

Index Terms—Linear tail-biting trellis, Reed–Muller codes, square-root bound.

I. INTRODUCTION

Following the success of the turbo decoding algorithm, the subject of decoding a code using a *Tanner graph* [16] became very popular (see

Manuscript received February 12, 1999; revised November 3, 1999.
 The authors are with the Department of Electrical Engineering-Systems, Tel-Aviv University, Ramat-Aviv 69978, Israel (e-mail: shany@eng.tau.ac.il; ybeery@eng.tau.ac.il).
 Communicated by F. R. Kschischang, Associate Editor for Coding Theory.
 Publisher Item Identifier S 0018-9448(00)04655-1.

[20] and references therein). A relatively simple Tanner graph, containing a single cycle, is related to the *tail-biting trellis* representation of a code. Tail-biting trellises were originally introduced by Solomon and van Tilborg [15]. Recently, there is interest in the efficient representation of block codes by tail-biting trellises [3], [8].

Currently, much is known about the *conventional trellis* representation of linear block codes (see [12], [17] for a summary of this subject). It is known that under a fixed coordinate ordering there exists a unique trellis representation of a linear block code, the *minimal trellis*, which minimizes the *state complexity profile* at all indices simultaneously [13]. It is also known that the same trellis minimizes the *branch complexity profile* at all indices simultaneously [12] and the total number of addition-equivalent operations required to implement the Viterbi decoding algorithm [18], [14]. Moreover, both the original construction of a trellis for a linear block code [1] and the construction of Forney [4] which renewed the interest in the subject are isomorphic to the minimal trellis.

Much less is known about tail-biting trellises of linear block codes. The additional degree of freedom (in comparison to conventional trellises) makes the theory of tail-biting trellises more complex, even with the restriction to linear codes and a fixed coordinate ordering. In [3], Calderbank *et al.* investigated the tail-biting trellis representation of several short codes. It was shown that using a length-2 *sectionalization* [10], there exists a tail-biting trellis for the Golay code which meets the *square-root bound* [21], [3]. The formal definition of a *linear trellis* was first given by Kötter and Vardy [8]. In that paper, the authors also gave a simple algorithm for finding a linear tail-biting trellis for which the product of the cardinalities of the state spaces at all indices is minimum. It is known [3] that it is sometimes possible to have a nonlinear tail-biting trellis for a linear code with a smaller state complexity than that of any linear tail-biting trellis for the code. However, performing the product of the trellises corresponding to a set of codewords that generate a given linear code as a space over \mathbb{F}_q (which results in a linear trellis) is a systematic way for constructing a tail-biting trellis for the code.

In this correspondence, linear tail-biting trellises are considered. Using the notions of *subtrellis*, *merging interval*, and *sub-tail-biting trellis* we prove some structural properties of linear tail-biting trellises. Specifically, it is shown that a linear tail-biting trellis has a certain simple structure, the *parallel-merged-cosets structure*. We present a necessary condition that a linear code must satisfy in order to have a linear tail-biting trellis representation that meets the square-root bound. The condition involves the code's *generalized Hamming weight hierarchy* [19] and the distance between the first and last index at which the state complexity profile of the minimal (conventional) trellis for the code gets its maximum. It is then shown that for $r \geq 2$ and $m \geq 4r - 1$ or $r \geq 4$ and $r + 3 \leq m \leq \lfloor (4r + 5)/3 \rfloor$ the Reed-Muller code $\text{RM}(r, m)$ under any bit order cannot be represented by a linear tail-biting trellis whose state complexity is half of that of the minimal (conventional) trellis for the code under the standard bit order.

The correspondence is organized as follows. In the following section we give several definitions that will be required throughout the correspondence. Section III is devoted to the structure of linear tail-biting trellises. In Section IV, we develop a necessary condition for the square-root bound. Finally, in Section V, we apply the previous results in order to show that for $r \geq 2$ and $m \geq 4r - 1$ or $r \geq 4$ and $r + 3 \leq m \leq \lfloor (4r + 5)/3 \rfloor$, $\text{RM}(r, m)$ cannot be represented by a linear tail-biting trellis that meets the square-root bound.

II. PRELIMINARIES

An *edge-labeled directed graph* is the triple (V, E, A) , where V is a set of *vertices*, A is a finite set (the *alphabet*), and E is a set of *edges*, i.e., ordered triples of the form (v, v', α) , where $v, v' \in V$, and

$\alpha \in A$. A trellis diagram $T = (V, E, A)$ is an edge-labeled directed graph with the property that V can be divided into $n + 1$ disjoint subsets $V = V_0 \cup V_1 \cup \dots \cup V_n$ in a way that if $(v, v', \alpha) \in E$ then $v \in V_{i-1}$ and $v' \in V_i$ for some $i \in \{1, 2, \dots, n\}$. The parameter n is said to be the length of T . When an edge-labeled directed graph is a trellis diagram we shall sometimes refer to vertices as *states* and to edges as *branches*. The set V_i is called the vertex class at *index* (or *depth*) i . Also, the set of edges connecting the vertices of V_{i-1} with the vertices of V_i , $i \in \{1, 2, \dots, n\}$, is denoted by E_i . A conventional trellis diagram is a trellis diagram with a single initial vertex and a single final vertex, i.e., $|V_0| = |V_n| = 1$. A tail-biting trellis diagram is a trellis diagram with the property that $|V_0| = |V_n|$. Moreover, it is assumed that the vertices of V_0 and V_n are identically labeled with labels from some set. Let $T = (V, E, A)$ be a trellis diagram of length n . Every path from V_0 to V_n defines a length- n vector with entries from A : the i th entry of the vector, $i \in \{1, 2, \dots, n\}$, is the label on the edge connecting a vertex from V_{i-1} with a vertex from V_i . A *valid path* in a tail-biting trellis (also called a *reduction* in [8]) is a path from V_0 to V_n that has the same initial and final vertex labels. In a conventional trellis, every path from the initial vertex to the final vertex is valid. We confine ourselves to trellis diagrams (either conventional or tail-biting) in which every edge and every vertex lie on some valid path. Such trellises are referred to as *reduced* in [8]. In addition, parallel edges with identical labels are not allowed. A trellis (either conventional or tail-biting) for which no two distinct valid paths represent the same vector will be referred to as a *one-to-one trellis*. In the sequel, the alphabet A is always a finite field \mathbb{F}_q . The state complexity profile of the trellis $T = (V, E, \mathbb{F}_q)$ is the sequence

$$\mathbf{s}(T) = (s_0(T), s_1(T), \dots, s_n(T))$$

whose i th entry, $i \in \{0, 1, \dots, n\}$, is defined as

$$s_i(T) \triangleq \log_q |V_i|.$$

We refer to $s_i(T)$ as the state complexity of T at index i . The maximum of the above sequence is defined as $s(T)$, the state complexity of T .

A length- n code over \mathbb{F}_q is a subset of \mathbb{F}_q^n . A length- n code is said to be linear when it is a subspace of \mathbb{F}_q^n . We denote a linear code of length n , dimension k , and minimum Hamming distance d as an $[n, k, d]$ code. Let T be a trellis diagram (either conventional or tail-biting). The set of vectors associated with all valid paths is referred to as $C(T)$, the code related to T . T is said to be the trellis representation of the code C (or a trellis for C) if and only if $C = C(T)$.

Throughout this correspondence, the term "square-root bound" refers to a *fixed* coordinate ordering of the discussed code. That is, if C is an $[n, k, d]$ code, and T^{conv} is the minimal (conventional) trellis for C , then there is no tail-biting trellis for C whose state complexity is less than $s(T^{\text{conv}})/2$. The validity of this version of the square-root bound can be easily verified.

Let C be an $[n, k, d]$ code over \mathbb{F}_q , and let $\mathbf{c} = (c_1, c_2, \dots, c_n)$ be a codeword of C . For

$$I = \{i_1, i_2, \dots, i_{|I|}\} \subseteq \{1, 2, \dots, n\}$$

let $P_I(\mathbf{c}) = (c_{i_1}, c_{i_2}, \dots, c_{i_{|I|}})$ be the projection of \mathbf{c} onto I . We also define the projection of the code C onto I as

$$P_I(C) = \bigcup_{\mathbf{c} \in C} P_I(\mathbf{c}).$$

The linear subcode C_I is defined as the set of all codewords in C that have zero entries at the indices in $\{1, 2, \dots, n\} - I$, where $A - B$ stands for the complementary set of B in A . By convention,

$\mathcal{C}_{\{1,2,\dots,n\}} = \mathcal{C}$. For a fixed i , $1 \leq i \leq n-1$, there are two subsets of $\{1, 2, \dots, n\}$ that are of special interest

$$i^- \triangleq \{1, 2, \dots, i\}$$

and

$$i^+ \triangleq \{i+1, i+2, \dots, n\}.$$

The subcode \mathcal{C}_{i^-} is referred to as the *past subcode* of \mathcal{C} at index i , and the subcode \mathcal{C}_{i^+} is referred to as the *future subcode* of \mathcal{C} at index i . We denote the *dimension/length profile* (DLP) [5] of \mathcal{C} as $(k_0(\mathcal{C}), k_1(\mathcal{C}), \dots, k_n(\mathcal{C}))$. Also, the set $\{d_0(\mathcal{C}), d_1(\mathcal{C}), \dots, d_k(\mathcal{C})\}$ stands for the generalized Hamming weight hierarchy of \mathcal{C} . It was shown in [5] that

$$d_u(\mathcal{C}) = [\text{first index } i \text{ for which } k_i(\mathcal{C}) = u].$$

Let

$$\mathbf{a} = (a_1, a_2, \dots, a_{l_a}), \quad l_a \geq 1$$

and

$$\mathbf{b} = (b_1, b_2, \dots, b_{l_b}), \quad l_b \geq 1$$

be two vectors with entries from \mathbb{F}_q . We denote the *concatenation* of the vectors as

$$\mathbf{a} \cdot \mathbf{b} = (a_1, a_2, \dots, a_{l_a}, b_1, b_2, \dots, b_{l_b}).$$

Two length- n trellis diagrams $T = (V, E, A)$ and $T' = (V', E', A)$ are said to be *isomorphic* if there is a one-to-one mapping $f: V \rightarrow V'$ such that $f(V_i) = V'_i$ for all $i \in \{0, 1, \dots, n\}$ and (v, u, α) is an edge in T if and only if $(f(v), f(u), \alpha)$ is an edge in T' . We say that T and T' are *structurally isomorphic* if the two trellises obtained by setting all the edge labels on T and T' to a certain constant $\alpha \in A$ are isomorphic. For $0 \leq i < j \leq n$, let $T_{[i,j]}$ and $T'_{[i,j]}$ be the trellises obtained from T and T' (respectively) after removing all vertices at indices $0, 1, \dots, i-1, j+1, j+2, \dots, n$, and deleting all edges incident with one or more of these vertices. We say that T and T' are isomorphic between indices i and j if and only if $T_{[i,j]}$ and $T'_{[i,j]}$ are isomorphic.

Let $T = (V, E, \mathbb{F}_q)$ and $T' = (V', E', \mathbb{F}_q)$ be two trellis diagrams (either conventional or tail-biting) of length n , and let $\mathcal{C} = C(T)$ and $\mathcal{C}' = C(T')$ be the related codes. The *product trellis* [9], denoted by

$$T^\pi = (V^\pi, E^\pi, \mathbb{F}_q) = T \times T'$$

is defined as follows. The set of vertices at index i in T^π is the Cartesian product of the sets of vertices at index i in T and T' : $V_i^\pi = V_i \times V'_i$, $i \in \{0, 1, \dots, n\}$. The edge

$$((v_{i-1}, v'_{i-1}), (v_i, v'_i), \alpha + \alpha'), \quad i \in \{1, 2, \dots, n\}$$

is in E_i^π if and only if $(v_{i-1}, v_i, \alpha) \in E_i$ and $(v'_{i-1}, v'_i, \alpha') \in E'_i$. As mentioned in [9], [8] it could be easily verified that $C(T^\pi) = \mathcal{C} + \mathcal{C}'$, where

$$\mathcal{C} + \mathcal{C}' \triangleq \{\mathbf{c} + \mathbf{c}' : \mathbf{c} \in \mathcal{C}, \mathbf{c}' \in \mathcal{C}'\}.$$

Let \mathcal{C} be an $[n, k, d]$ code over \mathbb{F}_q , and let G be a generator matrix for \mathcal{C} , i.e., a full-rank matrix whose row space is \mathcal{C} . In [9], the *span* of each row $\mathbf{g} = (g_1, g_2, \dots, g_n)$ of G is defined as $\{j_1, j_1 + 1, \dots, j_2\}$, where j_1 is the first index j and j_2 is the last index j for which $g_j \neq 0$. To each row \mathbf{g} of G we assign the minimal trellis $T_{\mathbf{g}}$ for $\langle \mathbf{g} \rangle$, where $\langle \mathbf{g} \rangle$ is the space generated by \mathbf{g} over \mathbb{F}_q . The trellis $T_{\mathbf{g}}$ has a single vertex at each depth i , $i \in \{0, 1, 2, \dots, j_1 - 1, j_2, j_2 + 1, \dots, n\}$, and q vertices at each depth i , $i \in \{j_1, j_1 + 1, \dots, j_2 - 1\}$.

The second set is referred to as the *active interval* of \mathbf{g} . The row \mathbf{g} is said to be *active* at i if and only if i is in the active interval of \mathbf{g} . Also, the indices j_1 and j_2 mentioned above are referred to as the *left index* of \mathbf{g} , $\text{left}(\mathbf{g})$, and the *right index* of \mathbf{g} , $\text{right}(\mathbf{g})$, respectively. Note that if $j_1 = j_2$ then the active interval of \mathbf{g} is empty. The trellis $T_{\mathbf{g}}$ consists of a single path from the initial vertex to the vertex at index $j_1 - 1$, a single path from the vertex at index j_2 to the final vertex, and q parallel paths from the vertex at index $j_1 - 1$ to the vertex at index j_2 . Clearly, the multiplication of the trellises associated with all rows of G gives a (conventional) trellis for \mathcal{C} . The state complexity of the resulting trellis at each index i is the number of rows in G that are active at that index. Moreover, when G is a *trellis-oriented generator matrix* of \mathcal{C} (i.e, $\text{left}(\mathbf{g}_1) \neq \text{left}(\mathbf{g}_2)$, $\text{right}(\mathbf{g}_1) \neq \text{right}(\mathbf{g}_2)$ for any pair of distinct rows, \mathbf{g}_1 and \mathbf{g}_2 , in G [4], [9]) the resultant trellis is the unique minimal trellis representation of \mathcal{C} [9]. In [8], there is a more general definition of a generator's span and of the trellis associated with a generator, which will be adopted in this correspondence. Define

$$[i, j]_n \triangleq \begin{cases} \{i, i+1, \dots, j\}, & i \leq j \\ \{i, i+1, \dots, n, 1, \dots, j\}, & i > j. \end{cases}$$

In the definition of [8], the span of $\mathbf{g} = (g_1, g_2, \dots, g_n)$ can be taken as $[i, j]_n$ provided that $P_{\{1,2,\dots,n\}-[i,j]_n}(\mathbf{g}) = \mathbf{0}$, where $\mathbf{0} \triangleq (0, 0, \dots, 0)$, and both g_i and g_j are not zero. The active interval of \mathbf{g} is now $[i, j-1]_n$. When $i \leq j$ the trellis associated with \mathbf{g} is the same as described above (that is, a conventional trellis). If $i > j$, then let \mathbf{g}' be the vector obtained by cyclicly shifting \mathbf{g} to the left j times. Let $T_{\mathbf{g}'}$ be the minimal (conventional) trellis for $\langle \mathbf{g}' \rangle$. Then the (tail-biting) trellis associated with \mathbf{g} , $T_{\mathbf{g}}$, is defined as the cyclic shift to the right j times of $T_{\mathbf{g}'}$ (c.f. [8]). As before, the multiplication of the trellises associated with all the rows of G gives a (tail-biting) trellis for \mathcal{C} . We refer to a span in the form $[i, j]_n$ as a *conventional span* if $i \leq j$. The span $[i, j]_n$ is called a *cyclic span* if $i > j$. Notice that there are many choices of a span for \mathbf{g} , each resulting in a different tail-biting trellis representation of $\langle \mathbf{g} \rangle$, $T_{\mathbf{g}}$.

Let $T = (V, E, \mathbb{F}_q)$ be a trellis diagram (either conventional or tail-biting). Following Kötter and Vardy [8], we assume that in addition to the usual labeling of edges in T , there is also a labeling of the vertices of T : Each vertex class V_i , $i \in \{0, 1, \dots, n\}$, is labeled by a vector from $\mathbb{F}_q^{[s_i(T)]}$. We require that all the vertex labels within the same vertex class are distinct. To each valid path in T

$$v_0 \xrightarrow{c_1} v_1 \xrightarrow{c_2} \dots \xrightarrow{c_{n-1}} v_{n-1} \xrightarrow{c_n} v_n$$

$v_i \in V_i$, $i \in \{0, 1, \dots, n\}$, $c_j \in \mathbb{F}_q$, $j \in \{1, 2, \dots, n\}$, we associate the *pathword*

$$\mathbf{v}_0 \cdot c_1 \cdot \mathbf{v}_1 \cdot \dots \cdot c_n \cdot \mathbf{v}_0$$

where \mathbf{v}_i is the length- $[s_i(T)]$ vector related to $v_i \in V_i$. Note that we have used the fact that in a valid path the initial and final vertex labels are identical. As in [8], we refer to the code obtained by all pathwords of T as the *edge-vertex label code* of T , $S(T)$. We can now give the definition of a linear trellis. Note that from the requirement that there are no parallel edges with the same label in T , it follows that there is a one-to-one correspondence between the set of paths in T and $S(T)$.

Definition 1 [8]: A trellis $T = (V, E, \mathbb{F}_q)$ is said to be \mathbb{F}_q -linear if there exists a labeling of the vertices of T such that $S(T)$ is a linear code over \mathbb{F}_q .

When T is a linear tail-biting trellis over \mathbb{F}_q , we will always assume that the labeling of the vertices is such that $S(T)$ is linear over \mathbb{F}_q . In [8, Theorem 2] it is stated that every linear trellis representation of a linear code \mathcal{C} can be constructed as $\prod_{\mathbf{g} \in \mathcal{G}} T_{\mathbf{g}}$, where \mathcal{G} is a set that generates \mathcal{C}

over \mathbb{F}_q , and T_g is the trellis related to g for some legal choice of span. On the contrary, it could be easily verified that $\prod_{g \in \mathcal{G}} T_g$ will always give a linear (either conventional or tail-biting) trellis for \mathcal{C} , so we can conclude that a trellis for \mathcal{C} is linear if and only if it is in the form of the above trellis product.

It is interesting to note that if there are two members in \mathcal{G} that have the same left index or the same right index then it is always possible to find a linear tail-biting trellis for \mathcal{C} with a state complexity profile that is strictly smaller for some indices than that of $\prod_{g \in \mathcal{G}} T_g$, and never larger than that of $\prod_{g \in \mathcal{G}} T_g$. This could be easily established in the following way. Suppose that two members of \mathcal{G} , g_1 and g_2 have the same left index or the same right index, and assume without loss of generality that the active interval of g_1 is not smaller than that of g_2 . Then it is always possible to assign a span to $g_\Delta \triangleq g_1 - g_2$ so that the active interval of g_Δ is a proper subset of the active interval of g_1 . It is therefore clear that every "interesting" linear tail-biting trellis over \mathbb{F}_q has either q edges or a single edge entering or leaving each state at all indices.

III. THE STRUCTURE OF LINEAR TAIL-BITING TRELLISES

In this section we examine the structure of linear tail-biting trellises. The main idea is to regard a tail-biting trellis as a collection of conventional trellises, each corresponding to a different initial vertex. Note that whereas Kötter and Vardy [8] considered subgraphs of a linear tail-biting trellis consisting of all valid paths passing through a $\mathbf{0}$ -labeled vertex at a given depth, we examine the subgraphs corresponding to *all* vertices at a given depth (for simplicity, this depth is chosen to be 0, but the results apply to any other choice of depth). It will be shown that for a linear tail-biting trellis these subgraphs have some useful properties. In particular, it will be shown that a linear tail-biting trellis always has the parallel-merged-cosets structure.

Definition 2: Let $T = (V, E, \mathbb{F}_q)$ be a tail-biting trellis. The subgraph of T consisting of all valid paths with a certain fixed vertex from V_0 will be called a *subtrellis* of T .

A subtrellis is clearly a conventional trellis. Of course, if \mathcal{T} is the set of all subtrellises of the tail-biting trellis T then

$$C(T) = \bigcup_{T' \in \mathcal{T}} C(T').$$

We shall now see that if T is a linear trellis over \mathbb{F}_q , then all subtrellises of T are very similar.

Proposition 1: Let $T = (V, E, \mathbb{F}_q)$ be a linear tail-biting trellis for \mathcal{C} . Then every subtrellis of T is a trellis for a coset of a fixed linear subcode of \mathcal{C} . Moreover, all subtrellises of T are structurally isomorphic.

Proof: From the assumption that the vertex labels within the same vertex class are distinct, it is clear that there can be only one subtrellis in T that has an initial $\mathbf{0}$ -labeled vertex. Moreover, from the assumption that T is linear it is clear that such a subtrellis must exist, since the all-zero pathword must be in $S(T)$. Let us denote the subtrellis with a $\mathbf{0}$ -labeled initial vertex as T_0 . Clearly, $S(T_0)$ is a linear code, and $C(T_0)$, a projection of $S(T_0)$, is also a linear code [8]. Let T_i be any subtrellis of T which is not T_0 . Adding any pathword from $S(T_0)$ to any pathword from $S(T_i)$ will clearly give a pathword from $S(T_i)$. Hence $S(T_i)$ contains one or more complete cosets of $S(T_0)$. On the other hand, all paths in T_i have the same initial vertex, so the difference between any two pathwords in $S(T_i)$ is a pathword from $S(T_0)$, and $S(T_i)$ is contained in a single coset of $S(T_0)$. Thus $S(T_i)$ is a coset of $S(T_0)$ and the proposition is established. \square

Note that Proposition 1 does not contradict the possibility of having two distinct subtrellises of T that represent the same coset of $C(T_0)$,

where T is defined in Proposition 1 and T_0 is defined in its proof. However, recall that T can be expressed as $\prod_{g \in \mathcal{G}} T_g$, where \mathcal{G} is a set that generates $C(T)$ over \mathbb{F}_q . If there are two distinct valid paths in T that represent the same codeword, then essentially \mathcal{G} contains a basis for $C(T)$ as a proper subset. In such a case, it is always possible to find a linear tail-biting trellis for $C(T)$ with a componentwise smaller state complexity than that of T . We shall therefore assume that the linear trellises discussed from this point on are one-to-one, and hence the subcodes represented by any two distinct subtrellises are disjoint. Under the above assumption, it is clear that the code related to a subtrellis is linear if and only if the subtrellis is linear.

Definition 3: Let $T = (V, E, \mathbb{F}_q)$ be a length- n tail-biting trellis, and let $T_1 = (V^1, E^1, \mathbb{F}_q)$ and $T_2 = (V^2, E^2, \mathbb{F}_q)$ be two distinct subtrellises of T . Let I be the interval

$$[i_1, i_2] \triangleq \{i_1, i_1 + 1, \dots, i_2\}, \quad 1 < i_1 \leq i_2 < n.$$

We say that T_1 and T_2 are merged in I if the set of all paths in T_1 beginning at vertices of $V_{i_1-1}^1$ and ending at vertices of $V_{i_2}^1$ is identical to the set of all paths in T_2 beginning at vertices of $V_{i_1-1}^2$ and ending at vertices of $V_{i_2}^2$.

Informally speaking, T_1 and T_2 are merged in I if they share the sections specified in I . Notice that the above definition is not restricted to linear tail-biting trellises. However, from the proof of Proposition 1 we know that $S(T_i)$ is a coset of $S(T_0)$ (T_0 and T_i are defined in the proof of Proposition 1). Hence the projections of $S(T_i)$ and $S(T_0)$ onto the same set are either identical or disjoint. We conclude that for the special case of a linear tail-biting trellis, two subtrellises either share all their paths or do not share any paths on any given interval. An example of two subtrellises that are merged on an interval is given in Fig. 1.

Of course, two subtrellises of length n can be merged in the interval $[i_1, i_2]$, $1 < i_1 \leq i_2 < n$, if and only if they are isomorphic between indices $i_1 - 1$ and i_2 . The following proposition is therefore of interest.

Proposition 2: Let T be a length- n linear tail-biting trellis over \mathbb{F}_q . Then two subtrellises of T , T_1 , and T_2 , are isomorphic between indices $i_1 - 1$ and i_2 , $1 < i_1 \leq i_2 < n$, if and only if

$$P_{[i_1, i_2]}[C(T_1)] = P_{[i_1, i_2]}[C(T_2)].$$

Proof: Let \mathbf{a} be a codeword of $C(T_1)$, and let \mathbf{b} be a codeword of $C(T_2)$. We define $\mathbf{c} = (c_1, c_2, \dots, c_n) = \mathbf{a} - \mathbf{b}$. From Proposition 1 it is clear that a trellis isomorphic to T_1 can be obtained from $T_2 = (V, E, \mathbb{F}_q)$ by adding c_i to the labels on all edges in E_i , $i \in \{1, 2, \dots, n\}$. Now, if

$$P_{[i_1, i_2]}[C(T_1)] = P_{[i_1, i_2]}[C(T_2)]$$

there is a choice of \mathbf{c} satisfying $P_{[i_1, i_2]}(\mathbf{c}) = \mathbf{0}$ and the first part of the proposition is established.

Clearly, if

$$P_{[i_1, i_2]}[C(T_1)] \neq P_{[i_1, i_2]}[C(T_2)]$$

it is not possible that T_1 and T_2 are isomorphic between indices $i_1 - 1$ and i_2 . \square

Definition 4: Let T be a linear tail-biting trellis of length n , and let T_0 be the linear subtrellis of T . The interval $I = [i_1, i_2]$, $1 < i_1 \leq i_2 < n$, will be referred to as a *merging interval* of T if any subtrellis that is merged with T_0 in $J \subset I$ is also merged with T_0 in the entire interval I .

We shall now see that although the definition of the merging interval refers only to T_0 , it is actually relevant to all other subtrellises.

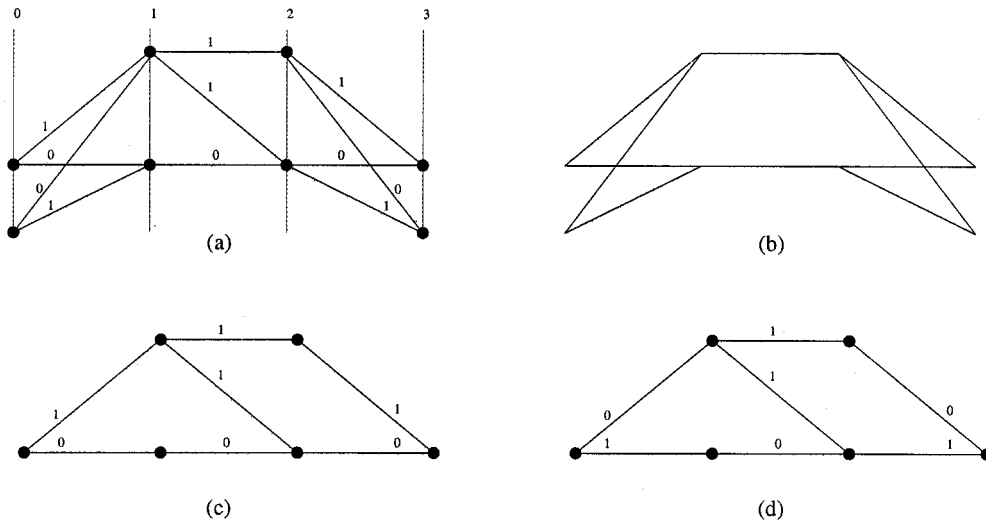


Fig. 1. An illustration of the merging concept. (a) A tail-biting trellis, T , for the code $\{(0, 0, 0), (1, 1, 1), (1, 1, 0), (0, 1, 0), (0, 1, 1), (1, 0, 1)\}$, consisting of two subtrellises that are merged in the interval $\{2\}$. The numbers above the vertex classes are their depths. (b) A schematic representation. (c) The subtrellis of T corresponding to $\{(0, 0, 0), (1, 1, 1), (1, 1, 0)\}$. (d) The subtrellis of T corresponding to $\{(0, 1, 0), (0, 1, 1), (1, 0, 1)\}$. *Remark:* The code $C(T)$ is nonlinear.

Proposition 3: Let T be a linear tail-biting trellis over \mathbb{F}_q , and let I be a merging interval of T . Suppose that two subtrellises of T , T_1 and T_2 , are merged in $J \subset I$. Then T_1 and T_2 are merged in I .

Proof: Let us assume that T_1 and T_2 are not merged in I . Let S^1 and S^2 be the edge-vertex label codes of T_1 and T_2 , respectively, and let s^1 be a codeword of S^1 . From Proposition 1 it is clear that $\{s - s^1 : s \in S^1\}$ is $S(T_0)$, where T_0 is the linear subtrellis of T . Now, $\{s - s^1 : s \in S^2\}$ is $S(T_I)$, where T_I is a subtrellis that is merged with T_0 in the same interval in which T_1 is merged with T_2 . So we found a subtrellis that is merged with T_0 in $J \subset I$ but not in I , contradicting our assumption that I is a merging interval. \square

From Proposition 3 we know that two subtrellises of a linear tail-biting trellis are either merged or parallel in the *entire* merging interval I . It is therefore of interest to give the following definition.

Definition 5: Let T be a linear tail-biting trellis, and let I be a merging interval of T . Let $T_I = \{T_1, T_2, \dots, T_l\}$ be a set of subtrellises of T that are pairwise merged in I , with the property that there is no other subtrellis of T that is merged with T_1 (say) in I . Then the subgraph of T consisting of all subtrellises in T_I will be called a *sub-tail-biting trellis (sub-TBT)* of T in I .

Clearly, every sub-TBT of T in I is a tail-biting trellis. If T_I is the set of all sub-TBT's in I , then

$$C(T) = \bigcup_{\hat{T} \in T_I} C(\hat{T}).$$

We refer to a merging interval I as a *maximal merging interval* if there is no merging interval I' such that $I \subset I'$. Fig. 2 clarifies the definition of a sub-TBT. Note that the partition of a linear tail-biting trellis into sub-TBT's depends on the choice of a merging interval.

Proposition 4 (Parallel-merged-cosets structure): Let $T = (V, E, \mathbb{F}_q)$ be a linear tail-biting trellis over \mathbb{F}_q for the $[n, k, d]$ code C , and let $I = [i_1, i_2]$, $1 < i_1 \leq i_2 < n$, be a merging interval of T . Then every sub-TBT of T in I is a tail-biting trellis for a coset of a fixed linear subcode of C . Moreover, all the sub-TBT's of T in I are structurally isomorphic.

Proof: Let us observe the sub-TBT of T in I which contains T_0 , the linear subtrellis of T , as a subtrellis. We refer to this sub-TBT as \hat{T}_0 . Since at any index the $\mathbf{0}$ -labeled vertex appears only once, it is clear that at any index the $\mathbf{0}$ -labeled vertex appears only in \hat{T}_0 . Let $T_I^0 = \{T_0, T_1, \dots, T_{l-1}\}$ be the set of subtrellises of \hat{T}_0 . Since any member of T_I^0 is merged with T_0 in I , it is clear that for any choice of T_j , $0 \leq j \leq l-1$ there exists a valid path in T_j with $\mathbf{0}$ -labeled vertices at all indices $i_1 - 1, i_1, \dots, i_2$. Let $\mathbf{a} \in S(T_{j_1})$ and $\mathbf{b} \in S(T_{j_2})$, $0 \leq j_1 < j_2 \leq l-1$, be the pathwords related to such two paths. From the linearity assumption, any linear combination of \mathbf{a} and \mathbf{b} over \mathbb{F}_q is the pathword related to a valid path in T . Since this pathword has $\mathbf{0}$ -labeled vertices, it is a pathword of $S(\hat{T}_0)$. Therefore, the label on the initial vertex related to a linear combination of \mathbf{a} and \mathbf{b} is the label of the initial vertex of some subtrellis in \hat{T}_0 . It follows that the set of labels related to the initial vertices in \hat{T}_0 is a space over \mathbb{F}_q , and, therefore, $S(\hat{T}_0)$ is a space over \mathbb{F}_q .

Using similar arguments, it is possible to show that $S(\hat{T}_m)$, where \hat{T}_m is a sub-TBT of T in I which is not \hat{T}_0 , is contained in a single coset of $S(\hat{T}_0)$. It remains to show that $|S(\hat{T}_m)| = |S(\hat{T}_0)|$. Let \mathcal{P} be a set that contains a single pathword from the edge-vertex label code of each subtrellis in \hat{T}_0 . Moreover, each pathword in \mathcal{P} is related to a path passing through $\mathbf{0}$ -labeled vertices at indices $i_1 - 1, i_1, \dots, i_2$. Let \mathbf{s} be a pathword from $S(\hat{T}_m)$. We examine the set $\mathcal{J} \triangleq \{\mathbf{s} + \mathbf{p} : \mathbf{p} \in \mathcal{P}\}$. From the linearity assumption, each pathword of \mathcal{J} is related to a valid path in T . In addition, each pathword in \mathcal{J} is related to a path whose vertex at index i , $i \in \{i_1 - 1, i_1, \dots, i_2\}$, belongs to the vertex class at index i in \hat{T}_m . It follows that $\mathcal{J} \subset S(\hat{T}_m)$, and hence the number of initial vertices in \hat{T}_m is not smaller than the number of initial vertices in \hat{T}_0 . Combining this with Proposition 1 the proof is completed. \square

Observe that a slightly modified version of Proposition 4 applies also for the case where T is a *group trellis* [3] for a group code C . The only difference is that in the group case every sub-TBT of T in I is a tail-biting trellis for a fixed *subgroup* of C (it can be verified that Proposition 1, Definition 4, Proposition 3, and Definition 5 can be suited to the group case by defining T_0 as the unique *group* subtrellis of C).

It follows from Proposition 4 that the state complexity of T at each index $i \in \{i_1 - 1, i_1, \dots, i_2\}$ can be totally determined from the state

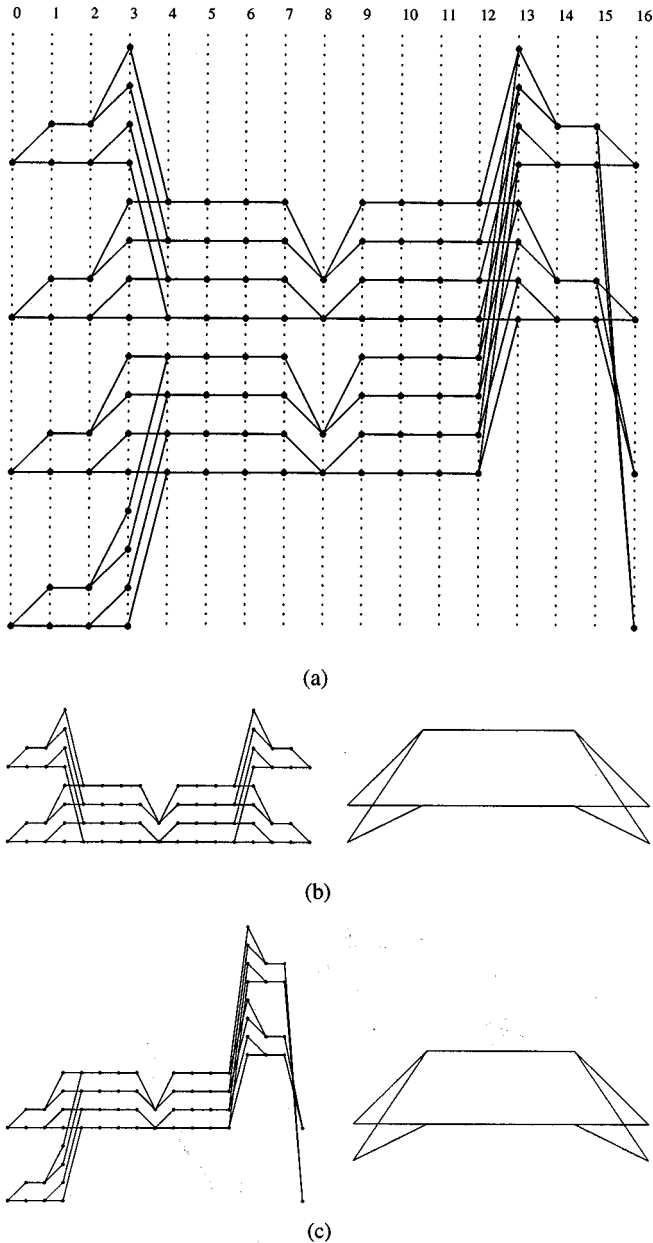


Fig. 2. (a) A linear tail-biting trellis for RM(1, 4), T , with two maximal merging intervals: $[5, 12]$ and $[14, 15]$. (b) The upper sub-TBT of T in $[5, 12]$, and a schematic representation. (c) The lower sub-TBT of T in $[5, 12]$, and a schematic representation. *Remark:* edge labels are suppressed in order to simplify the figure.

complexity of T_0 , and from the number of subtrellises in \hat{T}_0 (i_1, i_2, T, T_0 , and \hat{T}_0 are defined in Proposition 4 and in its proof). Let k and k^0 be the dimensions of $C(T)$ and $C(T_0)$, respectively, and let q^u be the number of subtrellises in \hat{T}_0 . From Proposition 4 and Proposition 1 it is clear that u is an integer, and that the number of distinct sub-TBT's of T in I is q^{k-k^0-u} . Hence, for each $i \in \{i_1 - 1, i_1, \dots, i_2\}$, we have

$$s_i(T) = s_i(T_0) + k - k^0 - u. \quad (1)$$

It is now of interest to see the connection between the foregoing discussion and the presentation of a linear tail-biting trellis T as $\prod_{\mathbf{g} \in \mathcal{G}} T_{\mathbf{g}}$ [3], where \mathcal{G} is a set that generates $C = C(T)$ over \mathbb{F}_q , and $T_{\mathbf{g}}$ is the trellis related to \mathbf{g} for some legal choice of span for \mathbf{g} . We can divide \mathcal{G} into two disjoint subsets, the first consisting of generators with conventional span, and the second consisting of generators with cyclic span.

These subsets are referred to as \mathcal{G}_0 and \mathcal{G}_1 , respectively. If we define $T_0 = \prod_{\mathbf{g} \in \mathcal{G}_0} T_{\mathbf{g}}$, and $\hat{T} = \prod_{\mathbf{g} \in \mathcal{G}_1} T_{\mathbf{g}}$, then it is clear that $T = T_0 \times \hat{T}$. It is also evident that T_0 is indeed the linear subtrellis of T . Let q^{u_i} be the number of subtrellises in the linear sub-TBT of T in a merging interval I which includes i or $i + 1$, $i \in \{1, 2, \dots, n - 1\}$. Then it follows that the state complexity of \hat{T} at index i is the base- q logarithm of the number of sub-TBT's of T in I , that is, $s_i(\hat{T}) = k - k^0 - u_i$, where k^0 is the dimension of $C(T_0)$, as before.

At the end of the previous section we mentioned that if two generators from \mathcal{G} have the same left index or the same right index (or both), then there exists a linear tail-biting trellis for C whose state complexity profile is componentwise smaller than that of $\prod_{\mathbf{g} \in \mathcal{G}} T_{\mathbf{g}}$. It is, therefore, clear that for any "interesting" tail-biting trellis for C , the linear subtrellis is the minimal trellis for the code it represents. This fact also rises as a private case of [8, Theorem 3]. However, from Proposition 1 it also follows that each subtrellis of an "interesting" tail-biting trellis is the minimum trellis for its code.

IV. A NECESSARY CONDITION FOR ACHIEVING THE SQUARE-ROOT BOUND

In this section we use the structural properties of linear tail-biting trellises from the previous section in order to develop a necessary condition required for achieving the square-root bound. The necessary condition will be derived in two steps. First, we will see that the maximum possible length of a merging interval decreases with the number of subtrellises that are merged with T_0 (the linear subtrellis) in this interval. Then we shall see that achieving the square-root bound requires that all the subtrellises are merged with T_0 in a merging interval of a certain minimal length.

We begin with the following simple observation, which does not require a proof.

Proposition 5: Let T be a linear tail-biting trellis over \mathbb{F}_q . Suppose that T_1 and T_2 are two subtrellises that are merged in $[i_1, i_2]$ and in $[i_3, i_4]$, $1 < i_1 \leq i_2 < i_3 \leq i_4 < n$. Then T_1 and T_2 are merged in $[i_1, i_4]$.

Let us now find a restriction on the number of subtrellises merged with the linear subtrellis of a linear tail-biting trellis in a given merging interval.

Proposition 6: Let T be a linear tail-biting trellis for the $[n, k, d]$ code C , and let T_0 be the linear subtrellis of T . Let $I = [i_1, i_2]$, $1 < i_1 \leq i_2 < n$ be a merging interval of T , and let \hat{T}_0 be the sub-TBT of T in I that contains T_0 as a subtrellis. Suppose that there are q^u subtrellises in \hat{T}_0 . Then

$$u \leq z_I(C) - z_I(C^0) \quad (2)$$

where $C^0 \triangleq C(T_0)$, and $z_I(C)$ is the dimension of $\mathcal{C}_{\{1, 2, \dots, n\} - I}$. Equality holds in (2) if and only if there is no sub-TBT of T in I, \hat{T}_I , with $0 \in P_I[C(\hat{T}_I)]$ except for \hat{T}_0 .

Proof: The number of paths referring to codewords of $C(\hat{T}_0)$ with a zero projection onto I is exactly $q^u \cdot q^{z_I(C^0)}$. Since these codewords form a subset of all codewords of C with zero projection onto I , the proposition follows. \square

An immediate result of Proposition 6 is that

$$u \leq z_I(C) \leq k_{n-|I|}(C).$$

The following corollary readily follows.

Corollary 7: Let T be a linear tail-biting trellis for the $[n, k, d]$ code C , and let T_0 be the linear subtrellis of T . Let $I = [i_1, i_2]$, $1 <$

$i_1 \leq i_2 < n$, be a merging interval of T , and let \hat{T}_0 be the sub-TBT of T in I that contains T_0 as a subtrellis. Suppose that there are q^u subtrellises in \hat{T}_0 . Then

$$|I| \leq n - d_u(\mathcal{C}).$$

In order to proceed, we have to find a connection between the minimal trellis of a linear code and a tail-biting trellis for the same code that achieves the square-root bound. Let \mathcal{C} be an $[n, k, d]$ code under a given bit-order, and let T^{conv} be its minimal trellis representation. As before, \mathcal{G} is a set that generates \mathcal{C} over F_q , and $T = \prod_{g \in \mathcal{G}} T_g$ is a linear tail-biting trellis for \mathcal{C} . Let k^0 and $k^1 = k - k^0$ be the number of generators from \mathcal{G} with conventional span and cyclic span, respectively. As usual, we denote the linear subtrellis of T as T_0 . Now, let i' , $i' \in \{0, 1, \dots, n\}$, be an index for which $s_{i'}(T^{\text{conv}}) = s(T^{\text{conv}})$. Clearly,

$$s(T) \geq s_{i'}(T) \geq s_{i'}(T_0) \geq s(T^{\text{conv}}) - k^1 \quad (3)$$

since a generator matrix for $\mathcal{C}(T_0)$, whose rows are the members of \mathcal{G} with conventional span, can be obtained by deleting k^1 rows from a generator matrix for \mathcal{C} , not all of them necessarily active at i' . It follows that T achieves the square-root bound only if $k^1 \geq s(T^{\text{conv}})/2$. On the other hand, k^1 is the state complexity of T at index 0, and, therefore, k^1 must not be larger than $s(T^{\text{conv}})/2$ if T achieves the square-root bound. We conclude that T achieves the square-root bound only if $k^1 = s(T^{\text{conv}})/2$. From (3) it follows that in such a case $s_{i'}(T_0) \geq s(T^{\text{conv}})/2$. Therefore, T achieves the square-root bound only if all q^{k^1} subtrellises of T are merged in an interval that contains i' or $i' + 1$. Combining this result with Proposition 5 and Corollary 7 the following proposition is established.

Proposition 8: Let \mathcal{C} be an $[n, k, d]$ code, and let T^{conv} be the minimal trellis for \mathcal{C} . Let i_1 and i_2 be the smallest and largest index (respectively) at which the state complexity of T^{conv} is $s(T^{\text{conv}})$. Then there exists a linear tail-biting trellis for \mathcal{C} that achieves the square-root bound only if

$$|I| \leq n - d_{s(T^{\text{conv}})/2}(\mathcal{C}) \quad (4)$$

where $I = [i_1 + 1, i_2]$.

In the next section we will use this proposition to prove that in many cases the Reed–Muller codes do not have a linear tail-biting trellis that achieves the square-root bound. Note that if the square-root bound is not achievable under a certain bit order, then it is also not achievable under any cyclic shift of this bit order. The results of Proposition 8 can be easily extended for the case of sectionalization. We say that a sectionalized tail-biting trellis meets the square-root bound if its state complexity is half of the state complexity of the identically sectionalized minimal trellis. The necessary condition from Proposition 8 is made suitable for a sectionalized tail-biting trellis with a constant section length by making two changes in (4). First, the state complexity of the identically sectionalized minimal trellis should be substituted as $s(T^{\text{conv}})$ in (4). Second, if the section length is l (l divides the length of the nonsectionalized trellis) then the interval I in (4) should be replaced by $[l i'_1 + 1, l i'_2]$, where i'_1 and i'_2 are the first and last index at which the state complexity of the identically sectionalized minimal trellis gets its maximum, respectively. Note that we did not provide a necessary condition required from a code in order to have a sectionalized tail-biting trellis with half of the state complexity of the nonsectionalized minimal trellis. However, if the section length is short enough so that the sectionalized state complexity is equal to the nonsectionalized state complexity, the condition given above coincides with the condition for half the state complexity of the nonsectionalized trellis.

V. REED–MULLER CODES AND THE SQUARE-ROOT BOUND

In this section we employ Proposition 8 in order to show that for $r \geq 2$ and $m \geq 4r - 1$ or $r \geq 4$ and $r + 3 \leq m \leq \lfloor (4r + 5)/3 \rfloor$ the Reed–Muller code $\text{RM}(r, m)$ under any bit order cannot have a linear tail-biting trellis whose state complexity is half of the state complexity of the minimal trellis for the code under the standard bit order.

Let $s(r, m)$ be the state complexity of the minimal trellis for $\text{RM}(r, m)$ under the standard bit order. Obviously, when $\text{RM}(r, m)$ is under any bit order for which the state complexity of its minimal trellis is larger than $s(r, m)$, there is no linear tail-biting trellis for the code whose state complexity is $s(r, m)/2$. However, there are many bit orders for which the state complexity is equal to $s(r, m)$.

Proposition 9: If $\text{RM}(r, m)$ does not satisfy the necessary condition from Proposition 8 under the standard bit order, then it cannot meet the square-root bound under any other bit order for which the state complexity of its minimal trellis is $s(r, m)$.

Proof: Suppose that $\text{RM}(r, m)$ is under one of the bit orders for which the state complexity of its minimal trellis is $s(r, m)$, and let i_1 and i_2 be the first and last index at which the state complexity of the minimal trellis gets its maximum, respectively. Let $I = [i_1 + 1, i_2]$ be the merging interval defined in Proposition 8. Now recall that the standard bit order is known to minimize all entries of the state complexity profile of the minimal trellis for $\text{RM}(r, m)$ simultaneously. It is therefore clear that the length of the interval I when $\text{RM}(r, m)$ is not under the standard bit order cannot be smaller than its length when the code is under the standard bit order. \square

The weight hierarchy of $\text{RM}(1, m)$ is given in [19]. Also, a simple formula for the bit-level state complexity of the minimal trellis for this code under the standard bit order is given in [11]. Using these results along with Proposition 8 it is straightforward to show that for $m \geq 4$ there is no linear tail-biting trellis for $\text{RM}(1, m)$ under the standard bit order that achieves the square-root bound. From Proposition 9 it also follows that for $m \geq 4$ there is no linear tail-biting trellis for $\text{RM}(1, m)$ under any bit order whose state complexity is $s(1, m)/2$. From Forney's recent duality theorem [6, Theorem 5.3] it also follows that for $m \geq 4$ there is no linear tail-biting trellis for $\text{RM}(m - 2, m)$ under any bit order whose state complexity is $s(m - 2, m)/2$. We remark that for this simple case ($r = 1$) exactly the same result can be obtained by using [3, Corollary 7]. However, for larger values of r this is *not* the case: At least by inspecting some specific values of r and m that are covered in the following development, one finds that the lower bound of [3, Corollary 7] falls below the square-root bound.

We shall now prove that for a high enough value of m , the second-order Reed–Muller code $\text{RM}(2, m)$ cannot have a linear tail-biting trellis representation that achieves the square-root bound. This result will be employed later as the basis for the induction used to prove the main result concerning Reed–Muller codes.

Proposition 10: For $m \geq 7$ there is no linear tail-biting trellis for $\text{RM}(2, m)$ under any bit order whose state complexity is $s(2, m)/2$.

Proof: From Proposition 9, it is sufficient to consider the standard bit order. Let T^{conv} be the minimal trellis for $\text{RM}(2, m)$ under the standard bit order. Let i_1 and i_2 be the first and last index, respectively, at which the state complexity of T^{conv} is $s(2, m)$. From the symmetry of the Reed–Muller codes, it is clear that $i_2 = 2^m - i_1$. Hence, in order to show that there is no linear tail-biting trellis for $\text{RM}(2, m)$ under the standard bit order that achieves the square-root bound, it is sufficient to show that

$$2i_1 < d_{\lfloor s(2, m)/2 \rfloor}[\text{RM}(2, m)].$$

From [11, Theorem 1] it is clear that $i_1 = 2^{m-2} + 2^{m-4} + 1$. Let us find the first index u' for which

$$d_{u'}[\text{RM}(2, m)] > 2^{m-1} + 2^{m-3} + 2.$$

According to [19], if u is expressed as

$$u = m + (m-1) + \cdots + (m-s+1) + \tilde{q}, \quad \tilde{q} < m-s \quad (5)$$

then

$$d_u[\text{RM}(2, m)] = \sum_{i=1}^s 2^{m-i} + \sum_{i=1}^{\tilde{q}} 2^{m-s-1-i}.$$

It is, therefore, clear that u' is equal to the value obtained from (5) when $s = 1$ and $\tilde{q} = 2$, i.e., $u' = m+2$. Hence, from the monotonicity of the weight hierarchy [19], it remains to show that for $m \geq 7$, $s(2, m)/2 \geq m+2$. Using the well-known state complexity of $\text{RM}(2, m)$ [2]

$$s(2, m) = \sum_{i=0}^2 \binom{m-2i-1}{2-i} = \frac{1}{2}m^2 - \frac{1}{2}m - 1, \quad m \geq 5$$

it is evident that for $m \geq 7$ there is no linear tail-biting trellis for $\text{RM}(2, m)$ under the standard bit order that achieves the square-root bound. \square

In order to proceed to higher order Reed–Muller codes, we will need a recursive formula for their weight hierarchy. We remark that the representation of the weight hierarchy of these codes that appears in [19] does not seem to be suitable for our purposes. Let $\{d_0^{r,m}, d_1^{r,m}, \dots, d_{k^{r,m}}^{r,m}\}$ be the weight hierarchy of $\text{RM}(r, m)$, where

$$k^{r,m} = \sum_{i=0}^r \binom{m}{i}.$$

We denote the dimension of the past subcode of $\text{RM}(r, m)$ at index i , $1 \leq i \leq 2^m$, by $p_i^{r,m}$. It is known [7] that under the standard bit order

$$d_u^{r,m} = [\text{first index } i \text{ for which } p_i^{r,m} = u].$$

From this point on (unless otherwise mentioned) all Reed–Muller codes are under the standard bit order. Let G be a generator matrix for the linear code \mathcal{C} . If no two rows of G have the same right index, then G is called a *past-oriented* generator matrix. When G is in this form, the dimension of the past subcode of \mathcal{C} at index i is the number of rows of G whose right index is not larger than i . As in [11], let $G_{r,m-1}$ and $G_{r-1,m-1}$ be past-oriented generator matrices of $\text{RM}(r, m-1)$ and $\text{RM}(r-1, m-1)$, respectively. Then

$$G_{r,m} = \begin{pmatrix} G_{r-1,m-1} & 0 \\ G_{r,m-1} & G_{r,m-1} \end{pmatrix} \quad (6)$$

is a past-oriented generator matrix for $\text{RM}(r, m)$ [2], [11]. It follows that for $1 \leq i \leq 2^{m-1}$, $p_i^{r,m} = p_i^{r-1,m-1}$. Hence for $d_u^{r,m} \leq 2^{m-1}$ we can write

$$\begin{aligned} d_u^{r,m} &= [\text{first index } i \text{ for which } p_i^{r,m} = u] \\ &= [\text{first index } i \text{ for which } p_i^{r-1,m-1} = u] \\ &= d_u^{r-1,m-1}. \end{aligned}$$

From (6) it is apparent that $d_u^{r,m} \leq 2^{m-1}$ for $u \leq k^{r-1,m-1}$. For $2^{m-1} + 1 \leq i \leq 2^m$ it follows from (6) that

$$p_i^{r,m} = k^{r-1,m-1} + p_{i-2^{m-1}}^{r,m-1}.$$

Hence, for $d_u^{r,m} \geq 2^{m-1} + 1$ we have

$$\begin{aligned} d_u^{r,m} &= [\text{first index } i \text{ for which } p_i^{r,m} = u] \\ &= 2^{m-1} + [\text{first index } i \text{ for which } p_i^{r,m-1} = u - k^{r-1,m-1}] \\ &= 2^{m-1} + d_{u-k^{r-1,m-1}}^{r,m-1}. \end{aligned}$$

We conclude that the recursive formula for the weight hierarchy of $\text{RM}(r, m)$ is

$$d_u^{r,m} = \begin{cases} d_u^{r-1,m-1}, & 0 \leq u \leq k^{r-1,m-1}, \\ 2^{m-1} + d_{u-k^{r-1,m-1}}^{r,m-1}, & k^{r-1,m-1} + 1 \leq u \leq k^{r,m}. \end{cases} \quad (7)$$

We shall now present another useful recursive relation. Let T^{conv} be the minimal trellis for $\text{RM}(r, m)$. Let $i_1^{r,m}$ be the first index at which the state complexity of T^{conv} is $s(r, m)$. Using [11, Theorem 1] it can be verified that

$$i_1^{r,m} = 2^{m-2} + i_1^{r-1,m-2}. \quad (8)$$

Besides (7) and (8), two lemmas will be required for the proof of the main theorem concerning Reed–Muller codes and the square-root bound.

Lemma 11: For any integers $r \geq 1$ and $m \geq 4r - 2$

$$\binom{m}{r} \geq 2 \sum_{i=0}^{r-1} \binom{m}{i}. \quad (9)$$

The proof of this lemma is given in the Appendix.

Lemma 12: For $r \geq 2$ and $m - 1 \geq 4r - 2$

$$\frac{1}{2}[s(r, m) - s(r-1, m-2)] \geq k^{r-1,m-1}.$$

Proof: From the results of [2] it follows that for $m - 1 \geq 4r - 2$

$$\begin{aligned} & s(r, m) - s(r-1, m-2) \\ &= \sum_{i=0}^r \binom{m-2i-1}{r-i} - \sum_{i=0}^{r-1} \binom{m-2i-3}{r-i-1} \\ &= \binom{m-1}{r}. \end{aligned} \quad (10)$$

Using Lemma 11 the proof is established. \square

Theorem 13: For $r \geq 2$ and $m \geq 4r - 1$ or $r \geq 4$ and $r + 3 \leq m \leq \lfloor (4r + 5)/3 \rfloor$, there is no linear tail-biting trellis for $\text{RM}(r, m)$ under any bit order whose state complexity is $s(r, m)/2$.

Proof: We begin by proving for $r \geq 2$ and $m \geq 4r - 1$. Similarly to the proof of Proposition 10, it is sufficient to show that

$$2i_1^{r,m} < d_{\lfloor s(r,m)/2 \rfloor}^{r,m}. \quad (11)$$

We define

$$\Delta(r, m) \triangleq d_{\lfloor s(r,m)/2 \rfloor}^{r,m} - 2i_1^{r,m}$$

and prove by induction that $\Delta(r, m) > 0$ for $r \geq 2$ and $m \geq 4r - 1$. From (8) it follows that

$$\Delta(r, m) = d_{\lfloor s(r,m)/2 \rfloor}^{r,m} - 2^{m-1} - 2i_1^{r-1,m-2}. \quad (12)$$

From Lemma 12 it follows that for $r \geq 2$ and $m \geq 4r - 1$, we have $\lfloor s(r, m)/2 \rfloor > k^{r-1,m-1}$. Using this result in conjunction with (7), we obtain that for $r \geq 2$ and $m \geq 4r - 1$

$$d_{\lfloor s(r,m)/2 \rfloor}^{r,m} = 2^{m-1} + d_{\lfloor s(r,m)/2 \rfloor - k^{r-1,m-1}}^{r,m-1}. \quad (13)$$

Substituting (13) in (12) we get that for $r \geq 2$ and $m \geq 4r - 1$

$$\Delta(r, m) = \Delta(r-1, m-2) + \delta(r, m), \quad (14)$$

where

$$\delta(r, m) \triangleq d_{\lfloor s(r, m)/2 \rfloor - kr - 1, m - 1}^{r, m - 1} - d_{\lfloor s(r - 1, m - 2)/2 \rfloor}^{r - 1, m - 2}.$$

Assume that $\Delta(r', m') > 0$ for $2 \leq r' < r$ and $m' \geq 4r' - 1$. Since $m \geq 4r - 1$ then $m - 2 > 4(r - 1) - 1$, and it follows from the above assumption that $\Delta(r - 1, m - 2) > 0$. It remains to show that $\delta(r, m) \geq 0$. Using (7) we obtain that $\delta(r, m) \geq 0$ if and only if

$$\lfloor s(r, m)/2 \rfloor - k^{r-1, m-1} \geq \lfloor s(r - 1, m - 1)/2 \rfloor. \quad (15)$$

Clearly, if

$$s(r, m)/2 - k^{r-1, m-1} \geq s(r - 1, m - 1)/2 \quad (16)$$

then (15) is valid. However, it follows from Lemma 12 that (16) holds for $r \geq 2$ and $m \geq 4r - 1$. The proof for $r \geq 2$ and $m \geq 4r - 1$ is completed by using Proposition 10 as the basis of the induction. The part concerning $r \geq 4$ and $r + 3 \leq m \leq \lfloor (4r + 5)/3 \rfloor$ follows from [6, Theorem 5.3] and the fact that the dual of $\text{RM}(r, m)$ is $\text{RM}(m - r - 1, m)$. \square

From [11, Theorem 1] it is easy to verify that the number of successive indices at which the state complexity profile of $\text{RM}(r, m)$ gets its maximum is at least as large as the number of successive indices at which the state complexity of $\text{RM}(1, m - 2r + 2)$ gets its maximum. The latter number is $2^{m-2r} - 1$, and hence a sectionalized minimal trellis for $\text{RM}(r, m)$ with a constant section length of $l = 2^v$, $v \leq m - 2r - 1$, will have a state complexity of $s(r, m)$. Yet, if i'_1 and $i'_2 = 2^m/l - i'_1$ are the first and last index at which the state complexity of the sectionalized minimal trellis gets its maximum, respectively, then it is possible that

$$\lfloor li'_1 + 1, 2^m - li'_1 \rfloor \subset \lfloor i_1^{r, m} + 1, 2^m - i_1^{r, m} \rfloor.$$

In order to show that a sectionalized linear tail-biting trellis for $\text{RM}(r, m)$ cannot achieve the square-root bound, we have to show that $2li'_1 < d_{\lfloor s(r, m)/2 \rfloor}^{r, m}$. Now, li'_1 is at most $i_1^{r, m} + l - 1$, so in order to show that a sectionalized linear tail-biting trellis whose section length is l cannot achieve the square-root bound it is sufficient to show that $\Delta(r, m) > 2l - 2$, where $\Delta(r, m)$ is defined in the proof of Theorem 13. From the proof of Proposition 10 it can be verified that for $m \geq 7$, $\Delta(2, m) > 2^{m-4} - 2$. We can also see from the proof of Theorem 13 that for $r \geq 2$ and $m \geq 4r - 1$

$$\Delta(r, m) \geq \Delta(2, m - 2r + 4) > 2^{m-2r} - 2 \geq 2l - 2$$

for $l = 2^v$, $v \leq m - 2r - 1$. So we proved the following corollary.

Corollary 14: For $r \geq 2$ and $m \geq 4r - 1$ or $r \geq 4$ and $r + 3 \leq m \leq \lfloor (4r + 5)/3 \rfloor$, there is no sectionalized linear tail-biting trellis with a constant section length of $l = 2^v$, $v \leq m - 2r - 1$, for $\text{RM}(r, m)$ under any bit order whose state complexity is $s(r, m)/2$.

We conclude with the following remark. We have shown in this section that, for some Reed–Muller codes, there is a lower bound on the minimal state complexity of a linear tail-biting trellis representation which is larger by 1 than the square-root bound. Proving this was rather involved, but it appears that trying to employ the *total span bound* [3] would be even harder, as it seems to be difficult to find the minimum possible *total active length*. Of course, one would like to find tighter bounds on the state complexity of a linear tail-biting representation of a linear code. We hope that the tools developed in this correspondence (especially Proposition 4 and Corollary 7) will be the basis for future research in this direction.

APPENDIX

Proof of Lemma 11: Define

$$g(m, r) \triangleq \binom{m}{r} - 2 \sum_{i=0}^{r-1} \binom{m}{i}.$$

We rewrite $g(m, r)$ as

$$g(m, r) = \binom{m}{r} [1 - S]$$

where

$$S \triangleq 2 \sum_{i=0}^{r-1} \binom{m}{i} / \binom{m}{r}.$$

Obviously, $g(m, r) \geq 0$ if and only if $S \leq 1$. Now, S can be expressed as $S = \sum_{i=0}^{r-1} a_i(m)$, where

$$a_{r-1}(m) = 2r/(m - r + 1)$$

and

$$a_{i-1}(m) = a_i(m)[i/(m - i + 1)], \quad \text{for } 1 \leq i \leq r - 1.$$

The ratio $a_{i-1}(m)/a_i(m)$, $1 \leq i \leq r - 1$, is denoted by $q_i(m)$. At this point it is clear that if $g(m, r) \geq 0$ then also $g(m_1, r) > 0$ for any integer m_1 , $m_1 > m$. This follows from the fact that $a_{r-1}(m_1) < a_{r-1}(m)$ and $q_i(m_1) < q_i(m)$, $1 \leq i \leq r - 1$. Therefore, in order to prove the lemma, it is sufficient to show that $g(4r - 2, r) \geq 0$. In order to do that, we first find an upper bound on S . Since

$$q_r(m) > q_{r-1}(m) > \cdots > q_1(m)$$

it follows that

$$\begin{aligned} S &= a_{r-1}(m) + a_{r-2}(m) + \cdots + a_0(m) \\ &< a_{r-1}(m) + q_{r-1}(m)a_{r-1}(m) + \cdots + q_{r-1}(m)^{r-1}a_{r-1}(m) \\ &= \frac{1 - q_{r-1}(m)^r}{1 - q_{r-1}(m)} a_{r-1}(m). \end{aligned} \quad (17)$$

Substituting the expressions for $a_{r-1}(m)$ and $q_{r-1}(m)$ when $m = 4r - 2$ in (17) we get

$$S < \frac{2r}{3r - 1} \cdot \frac{1 - (1/3 - 1/3r)^r}{2/3 + 1/3r}. \quad (18)$$

The right-hand side of (18) is denoted by $f(r)$. It can be verified that $f(1) = 1$, $f(2) < 1$, and $\lim_{r \rightarrow \infty} f(r) = 1$. Using a lengthy development it is also possible to verify that $df(r)/dr \geq 0$ for $r \geq 2$, completing the proof. \square

ACKNOWLEDGMENT

The authors wish to thank Dr. I. Reuven for many helpful discussions. The authors also wish to thank Prof. S. Encheva for her help with Lemma 11, and Prof. G. D. Forney, Jr. for providing a preprint of [6]. Finally, the authors wish to thank the anonymous referees for some helpful comments.

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*, vol. IT-20, pp. 284–287, Mar. 1974.
- [2] Y. Berger and Y. Be'ery, "Bounds on the trellis size of linear block codes," *IEEE Trans. Inform. Theory*, vol. 39, pp. 203–209, Jan. 1993.
- [3] A. R. Calderbank, G. D. Forney, Jr., and A. Vardy, "Minimal tail-biting trellises: The Golay code and more," *IEEE Trans. Inform. Theory*, vol. 45, pp. 1435–1455, July 1999.
- [4] G. D. Forney, Jr., "Coset codes—Part II: Binary lattices and related codes," *IEEE Trans. Inform. Theory*, vol. 34, pp. 1152–1187, Sept. 1988.
- [5] —, "Dimension/length profiles and trellis complexity of linear block codes," *IEEE Trans. Inform. Theory*, vol. 40, pp. 1741–1752, Nov. 1994.

- [6] —, “Codes on graphs: Generalized state realizations,” *IEEE Trans. Inform. Theory*, submitted for publication.
- [7] T. Kasami, T. Takata, T. Fujiwara, and S. Lin, “On the optimum bit orders with respect to the state complexity of trellis diagrams for binary linear codes,” *IEEE Trans. Inform. Theory*, vol. 39, pp. 242–245, Jan. 1993.
- [8] R. Kötter and A. Vardy, “Construction of minimal tail-biting trellises,” in *Proc. IEEE Information Theory Workshop*, Killarney, Ireland, June 1998, pp. 72–74.
- [9] F. R. Kschischang and V. Sorokine, “On the trellis structure of block codes,” *IEEE Trans. Inform. Theory*, vol. 41, pp. 1924–1937, Nov. 1995.
- [10] A. Lafourcade and A. Vardy, “Optimal sectionalization of a trellis,” *IEEE Trans. Inform. Theory*, vol. 42, pp. 689–703, May 1996.
- [11] C.-C. Lu and S.-H. Huang, “On bit-level trellis complexity of Reed–Muller codes,” *IEEE Trans. Inform. Theory*, vol. 41, pp. 2061–2064, Nov. 1995.
- [12] R. J. McEliece, “On the BCJR trellis for linear block codes,” *IEEE Trans. Inform. Theory*, vol. 42, pp. 1072–1092, July 1996.
- [13] D. J. Muder, “Minimal trellises for block codes,” *IEEE Trans. Inform. Theory*, vol. 34, pp. 1049–1053, Sept. 1988.
- [14] V. R. Sidorenko, “The Euler characteristic of the minimal code trellis is maximum,” *Probl. Inform. Transm.*, vol. 33, no. 1, pp. 87–93, Mar. 1997.
- [15] G. Solomon and H. C. A. van Tilborg, “A connection between block and convolutional codes,” *SIAM J. Appl. Math.*, vol. 37, pp. 358–369, Oct. 1979.
- [16] R. M. Tanner, “A recursive approach to low complexity codes,” *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 533–547, Sept. 1981.
- [17] A. Vardy, “Trellis structure of codes,” in *Handbook of Coding Theory*, V. S. Pless and W. C. Huffman, Eds. Amsterdam, The Netherlands: Elsevier, Dec. 1998.
- [18] A. Vardy and F. R. Kschischang, “Proof of a conjecture of McEliece regarding the expansion index of the minimal trellis,” *IEEE Trans. Inform. Theory*, vol. 42, pp. 2027–2034, Nov. 1996.
- [19] V. K. Wei, “Generalized Hamming weights for linear codes,” *IEEE Trans. Inform. Theory*, vol. 37, pp. 1412–1418, Sept. 1991.
- [20] N. Wiberg, “Codes and decoding on general graphs,” Ph.D. dissertation, Dept. Elec. Eng., Univ. Linköping, Sweden, Apr. 1996.
- [21] N. Wiberg, H.-A. Loeliger, and R. Kötter, “Codes and iterative decoding on general graphs,” *Euro. Trans. Telecommun.*, vol. 6, pp. 513–526, Sept. 1995.

Bounds on the State Complexity of Codes from the Hermitian Function Field and its Subfields

Yaron Shany and Yair Be’ery, *Senior Member, IEEE*

Abstract—An upper bound on the minimal state complexity of codes from the Hermitian function field and some of its subfields is derived. Coordinate orderings under which the state complexity of the codes is not above the bound are specified. For the self-dual Hermitian code it is proved that the bound coincides with the minimal state complexity of the code. Finally, it is shown that Hermitian codes over fields of characteristic 2 admit a recursive twisted squaring construction.

Index Terms—Geometric Goppa codes, Hermitian codes, minimal state complexity, trellises, twisted squaring construction.

I. INTRODUCTION

A trellis diagram can be regarded as an efficient representation of a code for the purpose of soft-decision decoding. Formally, a trellis

$T = (V, E)$ of rank n is a finite-directed graph, with vertex set V and edge set E , in which every vertex is assigned a “depth” in the range $\{0, 1, \dots, n\}$, and each edge connects a vertex at depth $i - 1$ to one at depth i , $1 \leq i \leq n$. The class of vertices at depth i , $0 \leq i \leq n$, is denoted by V_i . We assume that each edge of T is labeled with an element of \mathbb{F}_q , the finite field of q elements. In addition, we only consider trellises for which $|V_0| = |V_n| = 1$. Let \mathcal{C} be an $[n, k, d]$ linear code over \mathbb{F}_q (i.e., \mathcal{C} is a k -dimensional subspace of \mathbb{F}_q^n with minimum Hamming distance d). We say that the rank- n trellis T represents the code \mathcal{C} if \mathcal{C} is identical to the set of n -tuples read from all paths of T connecting the vertex in V_0 to the one in V_n . It is well known that any length- n linear code under a fixed coordinate ordering has a unique trellis representation (up to isomorphism), $T = (V, E)$, that minimizes $|V_i|$ simultaneously for all i , $0 \leq i \leq n$ (see [7], [15], and references therein for a summary of the subject). This trellis is called the *minimal trellis* of the code. For the code \mathcal{C} we define $s_i := \log_q |V_i|$, $0 \leq i \leq n$, where V_i corresponds to the minimal trellis of \mathcal{C} . The *state complexity profile* of \mathcal{C} is the sequence s_0, s_1, \dots, s_n . The *state complexity* of \mathcal{C} is defined as $s := \max_{0 \leq i \leq n} s_i$. Forney [3] demonstrated that the state complexity of \mathcal{C} may vary with respect to different ordering of coordinates. The *minimal state complexity* of \mathcal{C} is the minimal state complexity attainable by any ordering of the coordinates.

For a fixed coordinate ordering of the $[n, k]$ linear code \mathcal{C} , the entire state complexity profile can be calculated from $s_i = k - k_{i-} - k_{i+}$, $1 \leq i \leq n - 1$, where k_{i-} is the dimension of the *past subcode* at i , i.e., the subcode consisting of all codewords $(c_1, c_2, \dots, c_n) \in \mathcal{C}$ with $(c_{i+1}, c_{i+2}, \dots, c_n) = (0, 0, \dots, 0)$, and k_{i+} is the dimension of the *future subcode* at i , i.e., the subcode consisting of all codewords $(c_1, c_2, \dots, c_n) \in \mathcal{C}$ with $(c_1, c_2, \dots, c_i) = (0, 0, \dots, 0)$ [3].

The determination of the minimal state complexity and the attempt to find “good” coordinate orderings with respect to the trellis complexity of some important classes of codes were considered in several papers, e.g., [10], [11], [1], [16], [2]. These papers addressed the trellises of Reed–Muller, Bose–Chaudhuri–Hocquenghem (BCH), and quadratic-residue codes, and it seems only natural to investigate the trellis of *geometric Goppa codes*. Among geometric Goppa codes, the class of *Hermitian codes* (geometric Goppa codes arising from the *Hermitian function field*) was extensively studied. A simple presentation of the Hermitian function field and the related codes was given by Stichtenoth in [12]. The results of [12] were used in [19] to give a description of Hermitian codes which will be useful for our purposes. The *generalized Hamming weights (GHW) hierarchy* [17] of Hermitian codes and of geometric Goppa codes arising from some subfields of the Hermitian function field was studied in [20], [8], and [9].

In this correspondence, we give an upper bound on the state complexity of codes associated with the Hermitian function field and some of its subfields. For self-dual Hermitian codes, the minimal state complexity is determined, and coordinate orderings under which the state complexity coincides with the minimal state complexity are specified. The correspondence is organized as follows. In the following section we give some background on geometric Goppa codes. A lower bound on the minimal state complexity of geometric Goppa codes is presented. Then, in Section III, we give an upper bound on the state complexity profile of Hermitian codes and codes from certain subfields of the Hermitian function field, and specify coordinate orderings for which the state complexity profile is actually not above the bound. A simple formula for an upper bound on the minimal state complexity of self-dual codes from the Hermitian function field and some of its subfields is then derived. Finally, it is proved that the bound on the minimal state complexity of self-dual Hermitian codes is indeed the minimal state complexity itself. We conclude in Section IV by showing

Manuscript received September 29, 1999; revised March 9, 2000.

The authors are with the Department of Electrical Engineering–Systems, Tel-Aviv University, Ramat-Aviv 69978, Israel (e-mail: shany@eng.tau.ac.il; ybeery@eng.tau.ac.il).

Communicated by I. F. Blake, Associate Editor for Coding Theory.

Publisher Item Identifier S 0018-9448(00)05294-9.