A note on Type II covolutional codes

Johannesson, Rolf; Ståhl, Per; Wittenmark, Emma

Published in:
IEEE Transactions on Information Theory

DOI:
10.1109/18.850684

2000

Correspondence

A Note on Type II Convolutional Codes

Rolf Johannesson, *Fellow, IEEE*, Per Ståhl, *Student Member, IEEE*, and Emma Wittenmark, *Member, IEEE*

***Abstract*—The result of a search for the world's second Type II (doubly-even and self-dual) convolutional code is reported. A rate $R = 4/8$, 16-state, time-invariant, convolutional code with free distance $d_{\text{free}} = 8$ was found to be Type II. The initial part of its weight spectrum is better than that of the Golay convolutional code (GCC). Generator matrices and path weight enumerators for some other Type II convolutional codes are given. By the "wrap-around" technique tail-biting versions of $(32, 16, 8)$ Type II block codes are constructed.**

***Index Terms*—Convolutional codes, Golay code, tail-biting, Type II codes.**

I. Introduction

Recently, the existence of a 16-state tail-biting trellis representation for the $(24, 12, 8)$ binary Golay code was shown by Calderbank, Forney, and Vardy [1] confirming a conjecture by Forney [2]. By "unwrapping" the tail-biting generator they obtained the world's first Type II, i.e., doubly-even and self-dual (DESD), binary convolutional code, viz., the rate $R = 1/2$, time-varying, 16-state, Golay convolutional code (GCC). Stimulated by these remarkable results, Forney sent an e-mail (Aug. 14, 1997) with "Open questions and conjectures" to some colleagues. Among the open questions was: "Open question 2. Find additional doubly-even self-dual binary convolutional codes."

In this correspondence we report on a successful search for the world's second DESD; a rate $R = 4/8$, time-invariant, 16-state convolutional code was found to be Type II.

Some preliminaries are given in Section II (see also [3]). We give a brief description of the search procedure in Section III. In Section IV a few Type II convolutional codes are presented and in Section V we use a "wrap-around" technique to construct $(32, 16, 8)$ Type II, tail-biting block codes from our rate $R = 4/8$ convolutional codes. In Section VI we give the initial part of the weight spectrum for a Type II, rate $R = 1/2$, time-varying, 256-state convolutional code that Kötter and Vardy constructed by "unwrapping" the $(48, 24, 12)$ Type II, tail-biting quadratic residue (QR) block code.

II. Preliminaries

In a rate $R = b/c$ binary, convolutional encoder the information sequence $\boldsymbol{u} = \cdots \boldsymbol{u}_{-1}\boldsymbol{u}_0\boldsymbol{u}_1\boldsymbol{u}_2\cdots$, where $\boldsymbol{u}_i = (u_i^{(1)}u_i^{(2)}\cdots u_i^{(b)})$, is encoded as the code sequence $\boldsymbol{v} = \cdots \boldsymbol{v}_{-1}\boldsymbol{v}_0\boldsymbol{v}_1\boldsymbol{v}_2\cdots$, where $\boldsymbol{v}_i = (v_i^{(1)}v_i^{(2)}\cdots v_i^{(c)})$. The sequences must start at some finite time (positive or negative) and may or may not end. We write

$$\boldsymbol{v}_t = \boldsymbol{u}_t G_0 + \boldsymbol{u}_{t-1}G_1 + \cdots + \boldsymbol{u}_{t-m}G_m \tag{1}$$

where the parameter m is called the memory of the code and G_i, $0 \le i \le m$, is a binary $b \times c$ matrix.

Using (1) we can write the expression for the code sequence as

$$\cdots \boldsymbol{v}_{-1}\boldsymbol{v}_0\boldsymbol{v}_1\boldsymbol{v}_2\cdots = (\cdots \boldsymbol{u}_{-1}\boldsymbol{u}_0\boldsymbol{u}_1\boldsymbol{u}_2\cdots)\boldsymbol{G} \tag{2}$$

or, in shorter notation, as

$$\boldsymbol{v} = \boldsymbol{u}\boldsymbol{G} \tag{3}$$

where

$$\boldsymbol{G} = \begin{pmatrix} G_0 & G_1 & \cdots & G_m & & \\ & G_0 & G_1 & \cdots & G_m & \\ & & \ddots & \ddots & & \ddots \end{pmatrix}. \tag{4}$$

It is often convenient to express the information and code sequences in terms of the delay operator D

$$\boldsymbol{u}(D) = \cdots + \boldsymbol{u}_{-1}D^{-1} + \boldsymbol{u}_0 + \boldsymbol{u}_1 D + \boldsymbol{u}_2 D^2 + \cdots \tag{5}$$

$$\boldsymbol{v}(D) = \cdots + \boldsymbol{v}_{-1}D^{-1} + \boldsymbol{v}_0 + \boldsymbol{v}_1 D + \boldsymbol{v}_2 D^2 + \cdots. \tag{6}$$

Then we have

$$\boldsymbol{v}(D) = \boldsymbol{u}(D)G(D) \tag{7}$$

where the generator matrix $G(D)$ is a $b \times c$ full-rank matrix with entries in the field of realizable rational functions [3].

When an information sequence is encoded using a tail-biting trellis we impose the *tail-biting condition*, i.e., the starting state of the encoder is equal to its terminating state. A tail-biting trellis is therefore sometimes defined on a circular time axis. The generating matrix G for a rate $R = K/N$ tail-biting block code can be obtained from the semi-infinite generator matrix $\boldsymbol{G}$ of memory m, given in (4), for a rate $R = b/c = K/N$ convolutional code as follows. First we truncate $\boldsymbol{G}$ after $L = K/b$ rows (K is assumed to be a multiple of b). Then we "wrap-around" the ith columns, where $L < i \le L + m$ and obtain

$$\boldsymbol{G}_{tb} = \begin{pmatrix} G_0 & G_1 & \cdots & & G_m & & & \\ & G_0 & G_1 & \cdots & & G_m & & \\ & & \ddots & \ddots & & & \ddots & \\ & & & G_0 & G_1 & \cdots & & G_m \\ G_m & & & & G_0 & G_1 & \cdots & G_{m-1} \\ G_{m-1} & G_m & & & & \ddots & \ddots & \vdots \\ \vdots & & \ddots & & & & \ddots & G_1 \\ G_1 & G_2 & \cdots & G_m & & & & G_0 \end{pmatrix} \tag{8}$$

which is an $L \times L$ generator matrix for the tail-biting representation of a block code [3].

Manuscript received March 22, 1999; revised January 12, 2000. This work was supported in part by the Swedish Research Council for Engineering Sciences under Grant 94-77 and in part by the Foundation for Strategic Research—Personal Computing and Communication under Grant PCC-9706-09.

R. Johannesson and P. Ståhl are with the Department of Information Technology, Information Theory Group, Lund University, SE-221 00 Lund, Sweden (e-mail: rolf@it.lth.se; per.stahl@it.lth.se).

E. Wittenmark was with the Department of Information Technology, Information Theory Group, Lund University, SE-221 00 Lund, Sweden. She is now with Research, Cellular Systems, Ericsson Mobile Communications AB, SE-221 83 Lund, Sweden (e-mail: emma.wittenmark@ecs.ericsson.se).

Communicated by F. R. Kschischang, Associate Editor for Coding Theory.

Publisher Item Identifier S 0018-9448(00)04652-6.

The *dual code* $\mathcal{C}^{\perp}$ to a block or convolutional code $\mathcal{C}$ is the set of all codewords or c-tuples of sequences $\boldsymbol{v}^{\perp}$ such that the inner product $(\boldsymbol{v},\ \boldsymbol{v}^{\perp}) \stackrel{\text{def}}{=} \boldsymbol{v}(\boldsymbol{v}^{\perp})^T$ is zero, i.e., $\boldsymbol{v}$ and $\boldsymbol{v}^{\perp}$ are orthogonal, for all finite $\boldsymbol{v}$ in $\mathcal{C}$. The dual code $\mathcal{C}^{\perp}$ to a rate $R = b/c$ code is a rate $R = (c-b)/c$ code. It may be generated by any generator matrix $\boldsymbol{G}^{\perp}$ such that

$$\boldsymbol{G}\left(\boldsymbol{G}^{\perp}\right)^T = \boldsymbol{0}. \tag{9}$$

A linear code is said to be *self-orthogonal* if it is contained in its dual, and *self-dual* if it is equal to its dual. A linear code is said to be *even* if all codeword weights are even, and *doubly-even* if all codeword weights are multiples of four. Doubly-even self-dual codes are called Type II [4].

III. Search Strategy

Our search for Type II, rate $R = 4/8$, 16-state convolutional codes goes as follows:

First we obtain by computer search a (huge) set of rate $R = 1/8$ doubly-even, two-state convolutional codes. Their generators serve as building blocks that are pairwise combined to rate $R = 2/8$, four-state convolutional codes that are tested for doubly-evenness. The (huge) set of doubly-even, rate $R = 2/8$, four-state convolutional codes contains the final building blocks that are again pairwise combined. Thus we obtain rate $R = 4/8$, 16-state convolutional codes that are in turn tested for doubly-evenness. Finally, we exploit [1, Lemma 3], i.e., a block or convolutional binary, linear code is doubly-even if and only if it is self-orthogonal and has a set of noncatastrophic doubly-even generators. Hence, our doubly-even $R = 4/8$ convolutional codes, if any, are also self-orthogonal, or equivalently, self-dual.

IV. Doubly-Even Self-Dual Convolutional Codes

Using the search strategy described in Section III we found the world's second Type II, i.e., doubly-even and self-dual, convolutional code. It is a rate $R = 4/8$, time-invariant, unit-memory (16-state), binary convolutional code encoded by generator matrix as shown in (10) at the bottom of this page. It has free distance $d_{\text{free}} = 8$ and its path weight enumerator is

$$\begin{aligned} T_2(W) &= W^8\left(33 - 6W^4 + 8W^8 - 138W^{12} + 260W^{16}\right. \\ &\quad \left. -226W^{20} + 112W^{24} - 32W^{28} + 4W^{32}\right)/ \\ &\quad \left(1 - 30W^4 - W^8 + 20W^{12} + 32W^{16}\right. \\ &\quad \left. -74W^{20} + 56W^{24} - 22W^{28} + 4W^{32}\right) \\ &= 33W^8 + 984W^{12} + 29561W^{16} + 887016W^{20} + \cdots \end{aligned} \tag{11}$$

which is better than that of the GCC when considered as a rate $R = 4/8$, time-invariant convolutional code, viz.

$$T_{\text{GCC}}(W) = 49W^8 + 1352W^{12} + 38521W^{16} + 1096224W^{20} + \cdots. \tag{12}$$

When we consider the GCC as a time-varying rate $R = 1/2$ convolutional code it is reasonable to average the spectra for the four different phases. With this convention we have

$$T_{\text{GCC}}(W) = 12.25W^8 + 338W^{12} + 9455.25W^{16} + 264376W^{20} + \cdots. \tag{13}$$

If we multiply the values n_8 and n_{12} in (13) by four, then we obtain the numbers given by (12) but

$$4n_{16} = 4 \cdot 9455.25 = 37821 \tag{14}$$

which is 700 less than the corresponding number for the rate $R = 4/8$ GCC. This discrepancy is explained in [1].

The extended path enumerator, which counts the paths not only according to their weights but also according to the number of 1's in the information sequences, for the generator matrix $G_2(D)$ is

$$\begin{aligned} T_2(W, I) &= (4I + 6I^2 + 8I^3 + 4I^4 + 5I^5 + 4I^6 + I^7 + I^9)W^8 \\ &\quad + (15I^2 + 43I^3 + 94I^4 + 120I^5 + 145I^6 + 143I^7 \\ &\quad + 136I^8 + 108I^9 + 73I^{10} + 52I^{11} + 35I^{12} \\ &\quad + 10I^{13} + 7I^{14} + 3I^{15})W^{12} + \cdots. \end{aligned} \tag{15}$$

For the rate $R = 4/8$ GCC with the generator matrix, as shown in (16) at the bottom of this page, we have

$$\begin{aligned} T_{\text{GCC}}(W, I) &= (4I + 9I^2 + 12I^3 + 9I^4 + 8I^5 + 6I^6 + I^8)W^8 \\ &\quad + (7I^2 + 40I^3 + 113I^4 + 188I^5 + 233I^6 + 260I^7 \\ &\quad + 212I^8 + 148I^9 + 91I^{10} + 40I^{11} + 15I^{12} \\ &\quad + 4I^{13} + I^{14})W^{12} + \cdots \end{aligned} \tag{17}$$

We also found rate $R = 4/8$, time-invariant, unit-memory (16-state), binary convolutional codes encoded by the generator matrices shown in (18)–(20) (see the top of the following page), respectively.

$$G_2(D) = \begin{pmatrix} 0 & 0 & 1 & D & 1+D & 1+D & 1 & D \\ D & 0 & 0 & 1 & 1 & 1+D & 1+D & 1 \\ 1+D & D & 1+D & D & D & 0 & 1 & 0 \\ 1+D & 1+D & 0 & 0 & D & 1 & D & 1 \end{pmatrix} \tag{10}$$

$$G_{\text{GCC}} = \begin{pmatrix} 1+D & 0 & 1 & 0 & 1+D & 1 & 1 & 1 \\ 0 & 1+D & 1 & 1 & D & 1+D & 1 & 0 \\ D & D & 1+D & 0 & 0 & D & 1+D & 1 \\ 0 & D & 0 & 1+D & D & D & D & 1+D \end{pmatrix} \tag{16}$$

$$G_3(D) = \begin{pmatrix} 1+D & D & 1+D & 1 & 1 & 0 & 0 & 1 \\ 0 & 1+D & 1 & D & 1+D & 0 & 1 & 1 \\ D & D & 0 & 1+D & 1 & D & 1+D & 0 \\ D & 0 & 0 & D & D & 1+D & 1 & 1+D \end{pmatrix} \tag{18}$$

$$G_4(D) = \begin{pmatrix} 1+D & 1+D & D & 1 & 0 & 1 & 1 & 0 \\ D & 1 & 1+D & 1+D & 0 & D & D & 0 \\ D & 0 & D & 1+D & 1+D & 1 & 0 & 1 \\ 0 & D & D & 0 & D & 1 & 1+D & 1+D \end{pmatrix} \tag{19}$$

and

$$G_5(D) = \begin{pmatrix} 1+D & 1+D & 1 & 0 & 1 & 0 & 1 & 1 \\ D & 0 & 1+D & 1+D & 1 & 1 & 1 & 0 \\ 0 & D & D & 0 & 1 & 1 & 1+D & 1+D \\ D & D & D & D & D & D & 1 & 1 \end{pmatrix} \tag{20}$$

They are all Type II. The generator matrices $G_3(D)$ and $G_4(D)$ have the same path weight enumerator as $G_2(D)$ but they are not equivalent, and the path weight enumerator of $G_5(D)$ is given by

$$\begin{aligned} T_5(W) &= W^8\left(49 - 20W^4 - 168W^8 + 434W^{12} - 560W^{16}\right. \\ &\quad \left. + 448W^{20} - 224W^{24} + 64W^{28} - 8W^{32}\right)\Big/ \\ &\quad \left(1 - 28W^4 - 21W^8 + 118W^{12} - 160W^{16} + 80W^{20}\right. \\ &\quad \left. + 52W^{24} - 112W^{28} + 80W^{32} - 28W^{36} + 4W^{40}\right) \\ &= 49W^8 + 1352W^{12} + 38717W^{16} + 1107120W^{20} + \cdots. \end{aligned} \tag{21}$$

Furthermore, the generator matrices obtained by permuting the columns of $G_{\mathrm{GCC}}(D)$, $G_2(D)$, $G_3(D)$, $G_4(D)$, and $G_5(D)$ are also nonequivalent.

The extended path enumerators of $G_3(D)$, $G_4(D)$, and $G_5(D)$ are

$$\begin{aligned} T_3(W, I) &= \left(4I + 6I^2 + 8I^3 + 8I^4 + 4I^5 + I^6 + 2I^8\right)W^8 \\ &\quad + \left(13I^2 + 40I^3 + 87I^4 + 146I^5 + 180I^6 + 178I^7\right. \\ &\quad + 121I^8 + 94I^9 + 60I^{10} + 30I^{11} + 25I^{12} \\ &\quad \left. + 8I^{13} + I^{14} + I^{16}\right)W^{12} + \cdots \end{aligned} \tag{22}$$

$$\begin{aligned} T_4(W, I) &= \left(4I + 6I^2 + 8I^3 + 7I^4 + 4I^5 + 2I^6 + I^7 + I^8\right)W^8 \\ &\quad + \left(13I^2 + 42I^3 + 90I^4 + 143I^5 + 174I^6 + 165I^7\right. \\ &\quad + 137I^8 + 100I^9 + 57I^{10} + 35I^{11} + 18I^{12} \\ &\quad \left. + 5I^{13} + 2I^{14} + 2I^{15} + I^{16}\right)W^{12} + \cdots \end{aligned} \tag{23}$$

and

$$\begin{aligned} T_5(W, I) &= \left(4I + 9I^2 + 13I^3 + 12I^4 + 8I^5 + I^6 + 2I^7\right)W^8 \\ &\quad + \left(7I^2 + 38I^3 + 108I^4 + 200I^5 + 278I^6 + 268I^7\right. \\ &\quad + 211I^8 + 130I^9 + 68I^{10} + 27I^{11} + 14I^{12} \\ &\quad \left. + I^{13} + 2I^{14}\right)W^{12} + \cdots \end{aligned} \tag{24}$$

respectively.

V. Tailbiting Representations of Wrap-Around Block Codes

By the wrap-around technique described in [1] we obtain from the generator matrix $G_3(D)$ given in (18) a tail-biting representation of the $(24, 12, 8)$ binary Golay code. Applying the same technique to the generator matrices $G_2(D)$ and $G_4(D)$ results in $(24, 12, 4)$ block codes. If we "wrap-around" $G_5(D)$ we obtain a tail-biting representation of the Golay code which is a cyclic rotation of the columns of the representation of the Golay code given by Kötter and Vardy [5], [6]. Different column permutations of the Golay block code correspond to convolutional codes with different path weight enumerators. A permutation of the columns of the generator matrix for a block code does not always correspond to a permutation of columns of the ("unwrapped") generator matrix for the corresponding convolutional code. However, a permutation of the columns of a generator matrix for a convolutional code always corresponds to a permutation of columns of the corresponding "wrap-around" generator matrix for the block code.

By applying the wrap-around technique to $G_2(D)$ we also obtain a tail-biting representation of the $(32, 16, 8)$ Type II block code with generator matrix

$$G_2 = \begin{pmatrix} 00101110 & 00011101 & 00000000 & 00000000 \\ 00011111 & 10000110 & 00000000 & 00000000 \\ 10100010 & 11111000 & 00000000 & 00000000 \\ 11000101 & 11001010 & 00000000 & 00000000 \\ 00000000 & 00101110 & 00011101 & 00000000 \\ 00000000 & 00011111 & 10000110 & 00000000 \\ 00000000 & 10100010 & 11111000 & 00000000 \\ 00000000 & 11000101 & 11001010 & 00000000 \\ 00000000 & 00000000 & 00101110 & 00011101 \\ 00000000 & 00000000 & 00011111 & 10000110 \\ 00000000 & 00000000 & 10100010 & 11111000 \\ 00000000 & 00000000 & 11000101 & 11001010 \\ 00011101 & 00000000 & 00000000 & 00101110 \\ 10000110 & 00000000 & 00000000 & 00011111 \\ 11111000 & 00000000 & 00000000 & 10100010 \\ 11001010 & 00000000 & 00000000 & 11000101 \end{pmatrix}. \tag{25}$$

For the block code $\mathcal{B}_2$ with generator matrix G_2 the state complexity profile [7], [8] within an 8-tuple block is $\{16, 32, 64, 128, 256, 128,$

$64, 32, 16\}$, which is as bad as it could be. By permuting coordinates within 8-tuples and adding rows we obtain

$$G_2' = \begin{pmatrix} 10101101 & 11100000 & 00000000 & 00000000 \\ 01111010 & 00001110 & 00000000 & 00000000 \\ 00011011 & 01111000 & 00000000 & 00000000 \\ 00000111 & 10110101 & 00000000 & 00000000 \\ 00000000 & 10101101 & 11100000 & 00000000 \\ 00000000 & 01111010 & 00001110 & 00000000 \\ 00000000 & 00011011 & 01111000 & 00000000 \\ 00000000 & 00000111 & 10110101 & 00000000 \\ 00000000 & 00000000 & 10101101 & 11100000 \\ 00000000 & 00000000 & 01111010 & 00001110 \\ 00000000 & 00000000 & 00011011 & 01111000 \\ 00000000 & 00000000 & 00000111 & 10110101 \\ 11100000 & 00000000 & 00000000 & 10101101 \\ 00001110 & 00000000 & 00000000 & 01111010 \\ 01111000 & 00000000 & 00000000 & 00011011 \\ 10110101 & 00000000 & 00000000 & 00000111 \end{pmatrix} \tag{26}$$

which has the state complexity profile

$$\{16, 32, 64, 32, 64, 32, 64, 32, 16\}.$$

By the "wrap-around" technique we obtain rate $R = 16/32$, Type II, tail-biting block codes with generator matrices G_3, G_4, and G_5 from the generator matrices $G_3(D)$, $G_4(D)$, and $G_5(D)$, respectively. The generators G_3 and G_4 have the same state complexity profile as G_2' while the state complexity profile for G_5, $\{16, 32, 16, 32, 16, 32, 16, 32, 16\}$, is the same as that of the Golay code.

Let A_i denote the number of codewords of weight i in the block code $\mathcal{B}$. Then the block codes $\mathcal{B}_j$ with generator matrices G_j, where $j = \mathrm{GCC}, 2, 3, 4, 5$, all have $A_0 = A_{32} = 1$, $A_8 = A_{24} = 620$, $A_{12} = A_{20} = 13888$, and $A_{16} = 36518$. Sloane [9] identified the block code $\mathcal{B}_2$ as number $C83$ in the list of self-dual codes [4], [10]. Its name is g_{16}^{2+} and its group order $10321920 = 2^{15} \cdot 3^2 \cdot 5 \cdot 7$. (Sloane had earlier identified the $(32, 16, 8)$ code obtained by the "wrap-around" of the Golay convolutional code as $C85$, that is, f_2^{16+} with group order $23040 = 2^9 \cdot 3^2 \cdot 5$ [1], [4].)

VI. Convolutional Codes from Unwrapped Block Codes

Kötter and Vardy constructed the following generator matrix for the $(48, 24, 12)$ Type II, tail-biting QR code [5], [6] as shown in (27) at the bottom of this page. It has a period 16 tail-biting trellis with 256 states and its state complexity profile within a 16-tuple block is

$$\{256, 512, 256, 512, 256, 128, 256, 512, 256, 512, 256, 512, 256, 512, 256, 512, 256\}.$$

By first swapping rows 7 and 8, 15 and 16, and 23 and 24, and then unwrapping G_{KV}, Kötter and Vardy obtained a Type II, rate $R = 1/2$, time-varying, period 16, convolutional code having 256 states. This Kötter–Vardy convolutional code is quite remarkable. It is time-varying and is encoded by eight different generators. The outputs at phases 4, 12, and 20 and at phases 8, 16, and 24 are obtained via the time slices

$$(11, 10, 11, 01, 11, 11, 01, 10, 00, 11)^T$$

and

$$(11, 01, 01, 01, 11, 00, 00, 10, 00, 11)^T$$

respectively. These time slices involve 10 information symbols $u_t, u_{t-1}, \cdots, u_{t-9}$, but the outputs at all other phases are obtained

$$G_{\mathrm{KV}} = \begin{pmatrix} 11110001 & 11110010 & 11000000 & 00000000 & 00000000 & 00000000 \\ 00111111 & 10000100 & 11110000 & 00000000 & 00000000 & 00000000 \\ 00000110 & 11110000 & 11100111 & 00000000 & 00000000 & 00000000 \\ 00000011 & 10111111 & 01101000 & 00000000 & 00000000 & 00000000 \\ 00000000 & 11101101 & 01010110 & 11000000 & 00000000 & 00000000 \\ 00000000 & 00111001 & 11001101 & 10110000 & 00000000 & 00000000 \\ 00000000 & 00000011 & 10101011 & 11011100 & 00000000 & 00000000 \\ 00000000 & 00001101 & 11010011 & 00010111 & 00000000 & 00000000 \\ 00000000 & 00000000 & 11110001 & 11110010 & 11000000 & 00000000 \\ 00000000 & 00000000 & 00111111 & 10000100 & 11110000 & 00000000 \\ 00000000 & 00000000 & 00000110 & 11110000 & 11100111 & 00000000 \\ 00000000 & 00000000 & 00000011 & 10111111 & 01101000 & 00000000 \\ 00000000 & 00000000 & 00000000 & 11101101 & 01010110 & 11000000 \\ 00000000 & 00000000 & 00000000 & 00111001 & 11001101 & 10110000 \\ 00000000 & 00000000 & 00000000 & 00000011 & 10101011 & 11011100 \\ 00000000 & 00000000 & 00000000 & 00001101 & 11010011 & 00010111 \\ 11000000 & 00000000 & 00000000 & 00000000 & 11110001 & 11110010 \\ 11110000 & 00000000 & 00000000 & 00000000 & 00111111 & 10000100 \\ 11100111 & 00000000 & 00000000 & 00000000 & 00000110 & 11110000 \\ 01101000 & 00000000 & 00000000 & 00000000 & 00000011 & 10111111 \\ 01010110 & 11000000 & 00000000 & 00000000 & 00000000 & 11101101 \\ 11001101 & 10110000 & 00000000 & 00000000 & 00000000 & 00111001 \\ 10101011 & 11011100 & 00000000 & 00000000 & 00000000 & 00000011 \\ 11010011 & 00010111 & 00000000 & 00000000 & 00000000 & 00001101 \end{pmatrix} \tag{27}$$

via time slices which involve only nine information symbols, viz. $u_t, u_{t-1}, \cdots, u_{t-8}$. 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]

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

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

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.