ISSN 0333–3590

Constructions and proofs of nonexistence of some convolutional codes

Øyvind Ytrehus

REPORT NO. 60 April 1992

Abstract — An $(n, n-r, \nu, d_{\text{free}})$ code is a convolutional code of block length n, dimension $n-r$, constraint length ν, and free distance d_{free}. In [1], $N(r, \nu, d_{\text{free}})$ is defined as the largest n such that an $(n, n-r, \nu, d_{\text{free}})$ code exists, and bounds on this function are developed. This report demonstrates that for some sets of parameters $(n, n-r, \nu, d_{\text{free}})$, a detailed analysis of some associated block codes can lead to either a construction of an $(n, n-r, \nu, d_{\text{free}})$ code, or a proof that such a code does not exist. In particular, a $(9, 5, 4, 8)$ code and a $(6, 2, 6, 16)$ code are constructed, while it is proven that codes of parameters $(9, 7, 5, 6)$ do not exist.

**University of Bergen,
Dept. of Informatics,
HiB, Thormøhlensgt. 55,
N-5020 Bergen, Norway.**

This research was sponsored by the Norwegian Research Council for Science and the Humanities (NAVF).

1. Introduction

An $(n, n-r, \nu, d_{\text{free}})$ code is a convolutional code of block length n, dimension $n-r$, constraint length ν, and free distance d_{free}. In [1], $N(r, \nu, d_{\text{free}})$ is defined as the largest n such that an $(n, n-r, \nu, d_{\text{free}})$ code exists, and bounds on this function are developed. A particularly useful upper bound on $N(r, \nu, d_{\text{free}})$, originally due to Heller [2], arises from the fact that a set of convolutional code words of bounded length constitutes a linear block code, called a *terminated* code [3]. See Lemma 2.2.1.

In many cases the methods in [1] are insufficient to determine $N(r, \nu, d_{\text{free}})$ exactly. The purpose of this report is to demonstrate that a detailed analysis of the terminated block codes sometimes provides constructions or proofs of nonexistence of the associated convolutional codes. This analysis employs standard techniques from block code analysis (see, for instance, [4]), described in Section 2.1.

The notation in this report is different from that in [1], since the encoder matrices rather than the parity check matrices are considered.

The next section introduces some basic notation and techniques necessary for the subsequent analysis. In Section 3, it is shown that convolutional codes of parameters $(9, 7, 5, 6)$ do not exist. Section 4 describes the construction of a $(9, 5, 4, 8)$ code and a $(6, 2, 6, 16)$ code. It was previously not known whether codes of these sets of parameters existed, thus the results determine $N(2, 5, 6)$, $N(4, 4, 8)$, and $N(4, 6, 16)$.

The proofs employ different generalizable arguments.

2. Preliminaries

2.1 Linear Block Codes

An $[n, k]$ code $\mathcal{C}$ is a binary linear block code with block length n and dimension k. The minimum Hamming distance $d_H(\mathcal{C})$ is the minimum number of nonzero entries in a nonzero vector (or *codeword*) in $\mathcal{C}$. If the minimum Hamming distance $d = d_H(\mathcal{C})$ is known, $\mathcal{C}$ is called an $[n, k, d]$ code. Two $[n, k, d]$ codes $\mathcal{C}_a$ and $\mathcal{C}_b$ are equivalent if and only if a generating matrix $\mathbf{G}_a$ for $\mathcal{C}_a$ can be obtained by performing row operations and/or column permutations on a generating matrix $\mathbf{G}_b$ for $\mathcal{C}_b$.

Assume that $\mathcal{C}$ is an $[n, k, d]$ code containing a codeword $\mathbf{c}$ of weight w. Then the *residual code* $\mathcal{C}_1 = Res(\mathcal{C}, \mathbf{c})$ is obtained from $\mathcal{C}$ by deleting the coordinates where $\mathbf{c}$ is nonzero. If only the weight w of $\mathbf{c}$ is important, $\mathcal{C}_1$ is denoted by $Res(\mathcal{C}, w)$.

Lemmas 2.1.1–3 below are well known results, see e. g. Ytrehus [4].

Lemma 2.1.1. *If $w < 2d$, then $Res(\mathcal{C}, w)$ is an $[n - w, k - 1, d_1]$ code where $d_1 \geq d - \lfloor w/2 \rfloor$.*

Definition. *Let* $W(\mathcal{C}) = \{w(\mathbf{c}) \mid \mathbf{c} \in \mathcal{C} \setminus \{\mathbf{0}\}\}$*, where* $w(\mathbf{c})$ *is the Hamming weight of* $\mathbf{c}$*.*

Lemma 2.1.2. *Let* $\mathcal{C}$ *be an* $[n, k, d]$ *code, and let* $\mathcal{C}^\perp$ *be the dual code of* $\mathcal{C}$*, i. e. the vector space of dimension* $n - k$ *consisting of the binary vectors of length* n *that are orthogonal to all the codewords of* $\mathcal{C}$*. Let* $d^\perp = d_H(\mathcal{C}^\perp)$*. Then there exists an* $[n - d^\perp, k - d^\perp + 1, d]$ *code.*

Lemma 2.1.3 (The MacWilliams identities, [5], ch.5, eq. (13)). *Let* $\mathcal{C}$ *be an* $[n, k]$ *code, and let* $\mathcal{C}^\perp$ *be the dual code of* $\mathcal{C}$*. Also, in the rest of this report, let* A_i *and* B_i*,* $i = 0, \ldots, n$*, be the number of codewords of weight* i *in* $\mathcal{C}$ *and* $\mathcal{C}^\perp$*, respectively. Then*

$$B_j = \frac{1}{2^k} \sum_{i=0}^{n} A_i P_j(n, i), \qquad \forall j : 0 \leq j \leq n,$$

where

$$P_j(n, x) = \sum_{l=0}^{j} (-1)^l \binom{x}{l} \binom{n-x}{j-l}$$

is known as a Krawtchouk *polynomial.*

2.2 Convolutional Codes

A binary sequence $(\ldots, \mathbf{v}_p, \mathbf{v}_{p+1}, \ldots,)$, where each $\mathbf{v}_i \in GF(2)^n$, is conveniently represented by its *D-transform*, $\sum_{i=p}^{\infty} \mathbf{v}_i D^i$, where p is the smallest index such that $\mathbf{v}_p$ is nonzero. The set of all sequences (with a finite number of negative indices i such that $\mathbf{v}_i$ is nonzero) is called the set (or field, or vector space) of Laurent series, $\mathcal{F}^n$, over $GF(2)^n$ (see Piret [6]).

A binary convolutional code is the set of sequences generated by a *convolutional encoder*, which will be given two alternate descriptions: The D-transform encoder and the binary matrix encoder.

1. *The D-transform convolutional encoder.* Let $\mathbf{G}(D)$ be a $k \times n$ matrix of polynomials in D with binary coefficients. The polynomial in the ith row and jth column is denoted $g_{i,j}(D) = g_{i,j,0} + g_{i,j,1}D + \ldots + g_{i,j,q}D^q$. The convolutional code consists of all sequences whose D-transforms can be represented in the form $\mathbf{U}(D)\mathbf{G}(D)$, where $\mathbf{U}(D) \in \mathcal{F}^k$.
   The memory order m of $\mathbf{G}(D)$ is the largest degree of any polynomial $g_{i,j}(D)$. The constraint length ν is the sum

$$\sum_{i=1}^{k} \nu_i \triangleq \sum_{i=1}^{k} \max_{j=1,\ldots,n} \{\deg g_{i,j}(D)\}.$$

   The vector $(\nu_1, \ldots, \nu_k)$ will here be called the constraint length vector. A convolutional encoder is *minimal* if no other encoder for the same code has smaller constraint length [7]. Every convolutional code has at least one

minimal encoder. All minimal encoders of a given code have the same constraint length vector.

2. *The binary matrix convolutional encoder.* Let $\mathbf{G}$ be the semi-infinite matrix

$$\mathbf{G} = \begin{pmatrix} \mathbf{G}_0 & \mathbf{G}_1 & \dots & \mathbf{G}_m & & & \\ & \mathbf{G}_0 & \mathbf{G}_1 & \dots & \mathbf{G}_m & & \\ & & \mathbf{G}_0 & \mathbf{G}_1 & \dots & \mathbf{G}_m & \dots \\ & & & \ddots & \ddots & \ddots & \ddots \end{pmatrix}$$

where each $\mathbf{G}_l, l = 0, \dots, m$ is the binary $k \times n$ matrix in which the entry in the ith row and jth column is $g_{i,j,l}$. The convolutional code is the set of binary linear combinations of the rows of $\mathbf{G}$. There is a one-two-one correspondence between the D-transform description and the binary matrix description. We shall use these descriptions interchangeably.

In the following, let $\mathcal{C}$ be a convolutional code generated by the minimal binary matrix encoder $\mathbf{G}$ of constraint length ν. The *terminated* code $\mathcal{T}_j, j \geq 1$, is the block code of length jn that is generated by the rows of $\mathbf{G}$ whose nonzero entries are confined to the first j n-bit blocks. See examples in the next sections. The *free distance* of $\mathcal{C}$, $d_{\text{free}}(\mathcal{C})$, is defined as $\lim_{j \to \infty} d_H(\mathcal{T}_j)$. In [1], $\mathcal{C}$ is called an $(n, k, \nu, d_{\text{free}}(\mathcal{C}))$ convolutional code.

Lemma 2.2.1 ([2]). *Let $\mathcal{C}$ be a (n, k, ν, d) convolutional code with constraint length vector $(\nu_1, \dots, \nu_k)$. Then for all $j \geq 1$, $\mathcal{T}_j$ is an $[jn, \kappa(j), d]$ code, where $\kappa(j) = \sum_{i: j > \nu_i} (j - \nu_i)$.*

Lemma 2.2.1 can be used in conjunction with the tables of block codes in [8] to upper bound the free distance of convolutional codes. However, this approach does not utilize all the power of Lemma 2.2.1. In the next two sections, examples will be given where Lemma 2.2.1 is used together with simple block code analysis.

3. Nonexistence Proofs

We shall need the following lemma.

Lemma 3.1 *Suppose $\mathcal{C}$ is an $[18, 9, 6]$ code. Then the weight enumerator of $\mathcal{C}$ is*

$$A_0 = A_{18} = 1, A_6 = A_{12} = 102, A_8 = A_{10} = 153. \qquad (1)$$

.

Proof. By Lemma 2.1.2 and [8], $d^{\perp} = 6$, and by Lemma 2.1.1 and [8], $7 \notin W(\mathcal{C})$ (or $W(\mathcal{C}^{\perp})$). Then (1) is the only positive integer solution to the MacWilliams identities introduced in Lemma 2.1.3.

□

Theorem 3.2. *There is no (9,7,5,6) convolutional code.*

Proof. Suppose $\mathcal{C}$ is a (9,7,5,6) convolutional code, and that $\mathbf{G}(D)$ is a minimal encoder for $\mathcal{C}$. We can always assumed that the constraint length vector is ordered. Hence, the only possible constraint length vector is $(0, 0, 1, 1, 1, 1, 1)$, otherwise the parameters of $\mathcal{T}_1$ are $[9, k(\geq 3), 6]$, but this is impossible [8]. This implies that $\mathcal{T}_1$ is a $[9, 2, 6]$ code, and it follows from Lemma 2.1.1 that the upper left corner 2×9 submatrix of the binary encoder $\mathbf{G}(D)$ is equivalent (except for column permutations) to

$$\mathbf{T} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}.$$

Further, $\mathcal{T}_2$ is a $[18, 9, 6]$ code. From Lemma 3.1, it follows that $\mathcal{T}_2$ contains the all-one codeword, so a sequence of suitable row operations will transform $\mathbf{G}(D)$ into another encoder $\mathbf{G}^*(D)$ which (i) has the same or smaller constraint length as $\mathbf{G}(D)$, and (ii) contains a row of the form $(1 + D)(1, 1, 1, 1, 1, 1, 1, 1, 1)$. It follows (see, for instance, [6]) that $\mathbf{G}^*(D)$ is nonminimal; a contradiction.

□

4. Constructions of New Codes

Lemma 4.1 (Dodunekov and Encheva, [9]). *There exist exactly two nonequivalent [18,6,8] codes. Their weight distributions are*

$$\begin{array}{lllll} \text{(i)} & A_0 = 1, & A_8 = 46, & A_{12} = 16, & A_{16} = 1; \\ \text{(ii)} & A_0 = 1, & A_8 = 45, & A_{12} = 18. & \end{array}$$

Theorem 4.2. *There is a (9,5,4,8) convolutional code.*

Proof. Suppose $\mathcal{C}$ is a (9,5,4,8) convolutional code with binary encoder matrix $\mathbf{G}$. Since the constraint length is 4, it follows that $\mathbf{G}_1$ has rank at most 4. However, if $\mathbf{G}_1$ has rank less than 4, then the block code generated by $\mathbf{G}_0$ contains a subbcode of parameters $[9, 2, 8]$ which is impossible. Hence, without loss of generality, the first row of $\mathbf{G}_0$ is a vector of weight 8 or 9, while the first row of $\mathbf{G}_1$ is the zero vector. Thus, by Lemma 2.2.1, $\mathcal{T}_2$ is an [18,6,8] block code that contains one codeword (the linear combination of the first and sixth row) of weight at least 16. The only possible weight distribution is then (i) in Lemma 4.1. This block code is equivalent to the one generated by the matrix

$$\mathbf{G}_{\text{(i)}} = \begin{pmatrix} \mathbf{G}_{\text{RM}} & \mathbf{0^5} & \mathbf{0}^5 \\ \mathbf{b}_{\text{f}} & 1 & 1 \end{pmatrix},$$

where $\mathbf{G}_{\text{RM}}$ is the generating matrix for the [16,5,8] first-order Reed-Muller code [5,ch. 13–14], $\mathbf{0}^5$ is an all-zero column vector of length 5, and $\mathbf{b}_{\text{f}}$ is a binary row vector of length 16 associated with a bent function [5,pp. 426–430]. For our purposes we choose

$$\mathbf{b}_{\text{f}} = (0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1 \quad 1 \quad 0\,).$$

One binary representation of $\mathbf{G}_{(\mathrm{i})}$ is

$$\mathbf{G}_{(\mathrm{i})} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{pmatrix},$$

where the linear combination of the first and last row has weight 16. If we interpret $\mathbf{G}_{(\mathrm{i})}$ as the initial part of a binary convolutional encoder, then the encoder is not minimal and hence the free distance of the corresponding code is less than 8. However, performing suitable column permutations on $\mathbf{G}_{(\mathrm{i})}$, we obtain

$$\mathbf{G}^*_{(\mathrm{i})} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & & & & & & & & & \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \\ & & & & & & & & & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}$$

(where blanks denote zeros) which corresponds to the convolutional encoder

$$\mathbf{G}(D) = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1+D & 1+D & 1+D & 1 & 0 & 0 & 0 & D & 0 \\ 1+D & 1 & 0 & D & 1+D & 1 & 0 & D & 0 \\ 1 & D & 1 & D & 1 & 0 & 1+D & D & 0 \\ 0 & 0 & D & 1+D & D & 0 & D & 1 & 1+D \end{pmatrix}.$$

This encoder is minimal and it is possible to show that the code that it generates indeed has free distance 8.

□

Theorem 4.2. *There is a (6,2,6,16) convolutional code.*

Proof. Suppose $\mathcal{C}$ is a (6,2,6,16) convolutional code with binary encoder matrix $\mathbf{G}$, with constraint length vector (3,3). Then $\mathcal{T}_4$ and $\mathcal{T}_5$ are block codes of parameters $[24, 2, 16]$ and $[30, 4, 16]$, respectively. It is easy to show that $\mathcal{T}_4$ is equivalent to the code generated by

$$\mathbf{G}_{(4)} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix},$$

while $\mathcal{T}_5$ is equivalent to the code generated by

$$\mathbf{G}_{(5)} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \end{pmatrix}.$$

The task is now to modify, by use of column permutations, the matrix $\mathbf{G}_{(5)}$ such that (i) its upper left corner 2×24 submatrix $\mathbf{G}^*_{(4)}$ is a column permutation of $\mathbf{G}_{(4)}$, (ii) its lower right corner 2×24 submatrix is equal to $\mathbf{G}^*_{(4)}$, and (iii) its lower left and upper right 2×6 submatrices are zero. One solution is

$$\mathbf{G}^*_{(5)} = \begin{pmatrix} 1&1&1&1&0&0&1&0&1&1&0&1&0&1&1&1&1&0&1&1&1&1&0&0 & & & & & & \\ 1&1&0&0&1&1&0&1&1&0&1&1&1&0&1&1&0&1&1&0&1&0&1&1 & & & & & & \\ & & & & & & 1&1&1&1&0&0&1&0&1&1&0&1&0&1&1&1&1&0&1&1&1&1&0&0 \\ & & & & & & 1&1&0&0&1&1&0&1&1&0&1&1&1&0&1&1&0&1&1&0&1&0&1&1 \end{pmatrix},$$

(where blanks denote zeros) which correponds to the convolutional encoder

$$\mathbf{G}(D) = \begin{pmatrix} 1+D+D^3 & 1+D^2+D^3 & 1+D+D^2+D^3 & 1+D+D^2+D^3 & D^2 & D \\ 1+D^2+D^3 & 1+D & D+D^2+D^3 & D^2 & 1+D+D^3 & 1+D+D^2+D^3 \end{pmatrix}.$$

It can be verified that $\mathbf{G}(D)$ is minimal and that it generates a convolutional code with free distance 16.

□

References

[1] Ø. Ytrehus, "Long high-rate binary convolutional codes." Submitted for publication.

[2] J. A. Heller, "Sequential decoding: Short constraint length convolutional codes," space programs summary 37-54, Jet Propul. Lab., Calif. Inst. Tech., Pasadena, Dec. 1968.

[3] E. Paaske, "Short convolutional codes with maximal free distances for rates 2/3 and 3/4," *IEEE Trans. on Information Theory*, vol. IT-20, pp. 683–689, Sept. 1974.

[4] Ø. Ytrehus, "Code-buster: A software tool for characterizing abstract codes," Tech. Rep. 26, Dept. of Informatics, University of Bergen, Feb 1987.

[5] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. North-Holland, 1977.

[6] P. Piret, *Convolutional Codes - An Algebraic Approach*. The MIT Press, 1988.

[7] G. D. Forney, Jr., "Convolutional codes I: Algebraic structure," *IEEE Trans. on Information Theory*, vol. IT-16, pp. 720–738, Nov. 1970.

[8] A. E. Brouwer and T. Verhoeff, "An updated table of minimum-distance bounds for binary linear codes," *IEEE Trans. on Information Theory*, vol. IT-39, pp. 662–677, March 1993.

[9] S. Dodunekov and S. Encheva, "On the uniqueness of some linear subcodes of the binary extended Golay code," in *Proceedings of the International Workshop on Algebraic and Combinatorial Coding Theory, Varna, September 18-24*, pp. 38–40, 1988.