Binary Convolutional Codes of High Rate

Øyvind Ytrehus

Abstract — The function $N(r, \nu, d_{\text{free}})$, defined as the maximum n such that there exists a binary convolutional code of block length n, dimension $n - r$, constraint length ν, and free distance d_{free}, is studied. For $d_{\text{free}} \in \{3, 4\}$, codes are obtained from simple constructions that determine $N(r, \nu, d_{\text{free}})$ for all positive integers ν and r. Codes with minimum Hamming distance $d_{\text{free}} \geq 5$ are found by computer search. A number of new codes, with specified constraint lengths, redundancies, and free distances, and higher code rates than previously reported codes, are presented.

Keywords — Convolutional Codes, High Rate Codes.

University of Bergen,
Department of Informatics,
Høyteknologisenteret,
N-5020 Bergen, Norway.

Supported by the Norwegian Research Council for Science and the Humanities (NAVF).

1. Introduction

While good convolutional codes of low code rates are known, less is known about high rate codes. However, for some applications the bandwidth expansion associated with a low code rate is intolerable.

Definition. $N(r, \nu, d_{\text{free}})$ is the largest n such that there exists a convolutional code of block length n, dimension $n - r$, constraint length ν, and free distance d_{free}. We assume that $d_{\text{free}} \geq 3$; then $N(r, \nu, d_{\text{free}})$ is finite.

This correspondence paper is devoted to the problem of determining $N(r, \nu, d)$, for modest values of n, ν, and d.

(***Remark.*** A more common approach is to consider $d_{\text{MAX}}(n, k, \nu)$, the maximum attainable free Hamming distance of any code with rate k/n and constraint length ν. However, the tables of $N(r, \nu, d_{\text{free}})$ turn out to be more compact for high rate codes.)

Section 2 provides a brief review of the relevant notation. In Section 3 upper bounds on $N(r, \nu, d_{\text{free}})$ are considered. Section 4 presents lower bounds from explicit codes; these codes are found by general constructions as well as from computer search.

2. Preliminaries

The notation introduced in this section is kept at a minimum. In particular, we shall neither need nor define the concepts of *encoder*, *basic encoder*, *minimal encoder*, or the procedure for obtaining a minimal encoder from the parity check matrix. For a comprehensive treatment of the theory of convolutional codes the reader is referred to [1], or to the pioneering work in [2, 3, 4].

Let $\mathbf{h} = (h_1, \ldots, h_t)^T$ (for some $t > 0$) be a binary (column) vector. The *l-th shift* of $\mathbf{h}$ is the sequence $\mathbf{h}^{\stackrel{l}{\rightarrow}} = \left(\overbrace{0, \ldots, 0}^{l}, h_1, h_2, \ldots, h_t, 0, \ldots\right)^T$, starting at position one and extending to infinity, in which $\mathbf{h}$ can be found as the subvector starting in the $(l + 1)$-th position and all other entries are zero.

Let similarly $\mathbf{H}$ be a binary $(\nu + 1) \times n$ matrix. The l-th shift of $\mathbf{H}$ is the matrix $\mathbf{H}^{\stackrel{l}{\rightarrow}}$ of n columns and an infinite number of rows, in which $\mathbf{H}$ can be found as the submatrix starting in the $(l + 1)$-th row and all other entries are zero. Let further

$$\begin{aligned}\mathbf{H}^* =&\left(\mathbf{H}^{\stackrel{0}{\rightarrow}} \quad \mathbf{H}^{\stackrel{1}{\rightarrow}} \quad \mathbf{H}^{\stackrel{2}{\rightarrow}} \quad \cdots\right)\\ =&(\mathbf{h}_1 \quad \mathbf{h}_2 \quad \cdots \quad \mathbf{h}_n \quad \mathbf{h}_{n+1} \quad \cdots),\end{aligned}$$

where for $i \geq 1$, the column $\mathbf{h}_i = \mathbf{h}_{((i-1) \bmod n)+1}^{\stackrel{\lfloor (i-1)/n \rfloor}{\rightarrow}}$ is the $\lfloor (i-1)/n \rfloor$-th shift of the $(((i-1) \bmod n) + 1)$-th column of $\mathbf{H}$.

Let $\mathcal{F}$ be the set of infinite binary sequences $\mathbf{u} = (u_1, u_2, \ldots)$ starting at time 1. $\mathbf{H}$ is a parity-check matrix for a convolutional code $\mathcal{C} = \{\mathbf{u} \in \mathcal{F} | \mathbf{H}^* \mathbf{u}^T = \mathbf{h}_1 u_1 + \mathbf{h}_2 u_2 + \ldots + \mathbf{h}_i u_i + \ldots = \mathbf{0}\}$. The elements of $\mathcal{C}$ are called code words.

(***Remark.*** Strictly speaking, $\mathcal{C}$ should be defined as a vector space over the field of Laurent series over GF(2) (see e. g. [1]). The current definition was chosen to enhance simplicity and since in practice nonzero code words will start at or after some fixed time.)

$\mathcal{C}$ has block length n, dimension $n-1$ and constraint length at most ν. The constraint length is equal to ν if there exists no other parity check matrix $\mathbf{H}'$ for the same code $\mathcal{C}$ with smaller row dimension; in this case $\mathbf{H}$ is a minimal parity check matrix. In the following we will assume that parity check matrices are minimal.

Let $\mathcal{H}$, referred subsequently to as a *combined* matrix, be a collection of r matrices $\mathbf{H}^{(i)}, i = 1, \ldots, r$, over F where $\mathbf{H}^{(i)}$ has $\nu_i + 1$ rows and n columns.

The n columns of $\mathbf{H}^{(i)}$ are denoted $\mathbf{h}_j^{(i)} = (h_{j,0}^{(i)}, h_{j,1}^{(i)}, \ldots, h_{j,\nu_i}^{(i)})^T$, $j = 1, \ldots, n$, where the last row of $\mathbf{H}^{(i)}$ has at least one nonzero entry. The binary column vector $\mathbf{h}_j = (h_{j,0}^{(1)}, \ldots, h_{j,\nu_1}^{(1)}, h_{j,0}^{(2)}, \ldots, h_{j,0}^{(r)}, \ldots, h_{j,\nu_r}^{(r)})^T \in F^{\nu+r}$ will be called the j-th *combined column* of $\mathcal{H}$. Thus $\mathcal{H}$ can be regarded as a binary $\left(\sum_{i=1}^{r}(\nu_i + 1)\right) \times n$ matrix.

Each matrix $\mathbf{H}^{(r)}$ is a parity-check matrix for some $(n-1)$-dimensional convolutional code $\mathcal{C}^{(i)}$. We call $\mathcal{H}$ the parity-check matrix for the convolutional code $\mathcal{C} = \cap_{i=1}^{r}\mathcal{C}^{(i)}$. $\mathcal{C}$ has block length n, dimension at least $n - r$, and constraint length $\nu = \sum_{i=1}^{r}\nu_i$. The *free distance* $d_{\text{free}} = d_{\text{free}}(\mathcal{C})$ is the minimum Hamming weight of any nonzero code word in $\mathcal{C}$. $\mathcal{C}$ is said to be a $(n, n-r, \nu, d_{\text{free}})$ (or $(n, n-r, \underline{\nu}, d_{\text{free}})$, where the constraint length vector $\underline{\nu} = (\nu_1, \ldots, \nu_r)$, if we want to emphasize the distribution of the ν_i's) binary convolutional code.

3. Upper Bounds

The best known upper bounds on $N(r, \nu, d_{\text{free}})$ were established by considering the block codes obtained by truncation of the convolutional code. The following theorem is a slight generalization of a result appearing in [5].

Theorem 3.1 ([5]). *Let $\mathcal{C}$ be an $(n, n-r, \nu, d_{\text{free}})$ binary convolutional code defined by the combined parity check matrix $\mathcal{H}$ with combined columns $\mathbf{h}_j, j = 1, \ldots, n$. Let $\varepsilon_j, 1 \le j \le n$, be defined by $\varepsilon_j = max\{s | h_{j,\nu_i-l}^{(i)} = 0$ for $1 \le i \le r, 0 \le l \le s\}$. Then*

$$d_{\text{free}} \le \min\left\{ d(N, K) \mid N = mn + \sum_{j=1}^{n}\varepsilon_j,\ 1 \le K = N - (\nu + rm),\ m \ge 0 \right\}, \qquad (1)$$

where $d(N, K)$ is the largest minimum Hamming distance of any linear binary block code of block length* N and dimension* K. (Note: A table of bounds on $d(N, K)$ can be found in [6]).*

Proof. Let $\mathcal{C}_m^b$, for any integer $m > 0$, be the block code whose parity check matrix consists of all combined columns of $\mathcal{H}$ and those of their shifts that have zeros in all

but the first $(\nu + rm)$ positions. This block code has block length N and dimension K as indicated in (1). Since the code words of $\mathcal{C}_m^b$ after a suitable coordinate permutation can be regarded as code words of $\mathcal{C}$, $d_{\text{free}}(\mathcal{C})$ is not larger than the minimum Hamming distance of $\mathcal{C}_m^b$.

□

Theorem 3.1 appears to be strong for convolutional codes of short block length. However, the bound is much weaker for longer codes. For instance, the best bound obtainable from Theorem 3.1 and [6] on $N(1, 12, 5)$ is 84. (cf. Tables I,II). We shall consider a different approach to deriving upper bounds on $N(r, \nu, d_{\text{free}})$. In order to keep the notation simple (?), we will restrict ourselves in the remainder of this section to the case $r = 1$ (except for Lemma 3.3, where this notation is not used).

Definition 3.1. Let the *length* of a binary (column) vector (or sequence) $\mathbf{v} = (v_0, \ldots, v_m)$ be the largest l such that $v_i = v_{i+l-1} = 1$ for some $i, i \leq m - l$.

Definition 3.2. Let $\mathbf{h}_1, \mathbf{h}_2$ be two vectors. If $\mathbf{h}_1 = \mathbf{h}_2 \overrightarrow{\ }^{i}$, for some i, then $\mathbf{h}_1, \mathbf{h}_2$ are *shift-equivalent.*.

Definition 3.3. With respect to shift-equivalence, let V_l be the set of equivalence classes of vectors $\mathbf{v} = (v_0, \ldots, v_m)$ of length at most l. For convenience we will say that a vector belongs to V_l if it belongs to one of the equivalence classes in V_l. Note that $|V_l| = 2^{l-1}$.

Definition 3.4. Let $\mathbf{H}$ be a parity check matrix for an $(n, n-1, \nu, d_{\text{free}})$ binary convolutional code, in which the columns are $\mathbf{h}_j = (h_{j,0}, \ldots, h_{j,\nu})$, $1 \leq j \leq n$. We shall assume that $h_{j,0} = 1$, $1 \leq j \leq n$. For nonnegative integers a, b, $0 \leq a + b \leq \nu$, let $(\lambda_1, \lambda_2, \ldots, \lambda_a)$ and $(\tau_1, \tau_2, \ldots, \tau_b)$ be a binary a-tuple and b-tuple, respectively. Associated to $\mathbf{H}$ we have the number $n_{\lambda_1, \lambda_2, \ldots, \lambda_a}^{\tau_1, \tau_2, \ldots, \tau_b}$ of columns $\mathbf{h}_j$ of $\mathbf{H}$ such that $h_{j,s} = \tau_s$, $1 \leq s \leq b$, and $h_{j,\nu+1-s} = \lambda_s$, $1 \leq s \leq a$. Thus, for examples, $n_0 = n - n_1$ is the number of columns of $\mathbf{H}$ that belong to V_ν, $n_{00} = n_0 - n_{01}$ is the number of columns of $\mathbf{H}$ that belong to $V_{\nu-1}$, and $n_{00}^1 = n_{00} - n_{00}^0$ is the number of columns $\mathbf{h}_j$ that belong to $V_{\nu-1}$ and satisfy $h_{j,1} = 1$.

Lemma 3.2. $N(1, \nu, 4) \leq 2^{\nu-1}$.

Proof. Suppose $\mathbf{H}$ defines a code of constraint length ν, free distance 4 and rate $(n-1)/n$ for $n = 2^{\nu-1} + 1$.
If $n_0 = 0$, then all the $2^{\nu-1} + 1$ columns in $\mathbf{H}$ belong to $V_{\nu+1} \setminus V_\nu$, so they cannot be all distinct since $|V_{\nu+1} \setminus V_\nu| = 2^{\nu-1}$.
If $n_0 > 0$, then without loss of generality (w.l.o.g.), $\mathbf{h}_1, \ldots, \mathbf{h}_{n_0}$ belong to V_ν, while $\mathbf{h}_{n_0+1}, \ldots, \mathbf{h}_n$ belong to $V_{\nu+1} \setminus V_\nu$. Then the $2^{\nu-1} + 1 - n_0$ distinct linear combinations $\mathbf{h}_1 + \mathbf{h}_i$, $i = n_0 + 1, \ldots, n$, all belong to V_ν. Hence at least one sum $\mathbf{h}_1 + \mathbf{h}_i$ must

be shift-equivalent to one of $\mathbf{h}_1, \ldots, \mathbf{h}_{n_0}$, and the corresponding code will have free distance at most three.

□

Similarly, we have the following lemma which applies to codes with $r > 1$:

Lemma 3.3. *For $r \geq 2$, $N(r, \nu, 4) \leq 2^{\nu+r-1}$.*

Proof. Let $\mathcal{H}$ define an $(n, n-r, \nu, 4)$ binary convolutional code. Let $X_i, i = 1, \ldots, 2^r - 1$ be the set of combined columns $\mathbf{h}_j$ in $\mathcal{H}$ such that $(h_{j,0}^{(1)}, \ldots, h_0^{(r)})$ is the binary expansion of i, and let $x_i = |X_i|$. W. l. o. g., we can assume that $2^\nu \geq x_1 \geq x_i, 2 \leq i \leq 2^r - 1$. We first show that

$$\forall i : 1 \leq i \leq 2^{r-1} - 1 : \; x_{2i} + x_{2i+1} \leq 2^\nu. \tag{2}$$

If $x_{2i} = 0$, there is nothing to prove, so assume that $x_{2i} > 0$. Since the free distance is more than three, it follows that if $\mathbf{a} \in X_1$ and $\mathbf{b} \in X_{2i}$, then $(\mathbf{a} + \mathbf{b}) \notin X_{2i+1}$. There are x_1 choices for $\mathbf{a}$, hence $x_{2i+1} \leq 2^\nu - x_1$ implying (2). So

$$n = \sum_{j=1}^{2^r-1} x_j = x_1 + \sum_{i=1}^{2^{r-1}-1} (x_{2i} + x_{2i+1}) \leq x_1 + \left(2^{r-1} - 1\right)2^\nu \leq 2^{\nu+r-1}.$$

□

In Section 4 we shall see that the bounds of Lemmas 3.2 and 3.3 are tight.

Theorem 3.4. *Let $\mathcal{C}$ be an $(n, n-1, \nu, 5)$-code with parity check matrix $\mathbf{H}$. Then it holds that*

a)

$$n_0 + \binom{n}{2} + n_1 n_0 + n_0 n_{00} \leq 2^{\nu-1},$$

b)

$$n_{00} + \binom{n_0}{2} + \binom{n_1}{2} + \sum_{i,j \in \{0,1\}} n_{ij} n_{0ij} + \sum_{i \in \{0,1\}} n_0^i n_1^i \leq 2^{\nu-2},$$

c)

$$n_{000} + \sum_{i,j \in \{0,1\}} \binom{n_{ij}}{2} + \sum_{i,j \in \{0,1\}} n_{0ij} \cdot \min\left(n_{ij0}, n_{ij1}\right) + \sum_{i,j \in \{0,1\}} n_{i0}^j n_{i1}^j \leq 2^{\nu-3}.$$

Proof. a) The LHS is a lower bound on the number of sums of one or two columns of $\mathbf{H}^*$ that belong to V_ν. (If for convenience we regard $\mathbf{H}$ as a *set* of columns), such sums can be obtained by selecting one column in $\mathbf{H} \cap V_\nu$, the sum of two columns in $\mathbf{H}$, or the sum of a column in $\mathbf{H}$ and a column in $\mathbf{H}^{\overrightarrow{l}}$ for some $l > 0$. The inequality follows since all such sums must be distinct (or else the free distance would be four or less), and since $|V_\nu| = 2^{\nu-1}$.
b) and c) are similar, where we consider $V_{\nu-1}$ and $V_{\nu-2}$, respectively.

□

To apply Theorem 3.4, we can check all possible partitions of n into $\{n_{\lambda_1,\lambda_2,\lambda_3}\}$ and $\{n^{\tau}_{\lambda_1,\lambda_2}\}$ against the conditions given in the theorem. The largest values of n for which there exist such partitions that satisfy the conditions of Theorem 3.4 are shown in the right column of Table I.

Note that Theorem 3.4 represents a special case of a class of bounds. Checking increasingly more detailed partitions $\{n^{\tau_1,\tau_2,\ldots,\tau_b}_{\lambda_1,\lambda_2,\ldots,\lambda_a}\}$ we can obtain stronger bounds (up to the point where we actually perform a complete exhaustive search) but the computational effort will increase also. The detail level in Theorem 3.4 is chosen so as to provide a clear improvement over Theorem 3.1 while remaining of manageable complexity.

Table I illustrates the difference between Theorems 3.1 and 3.4 for $d_{\text{free}} = 5$. Even

Table I Upper bounds on $N(1, \nu, 5)$ for $5 \leq \nu \leq 12$.

ν	Theorem 3.1	Theorem 3.4
5	5	4
6	7	6
7	11	9
8	18	13
9	29	18
10	42	26
11	60	36
12	84	52

though Theorem 3.4 appears to be significantly stronger than Theorem 3.1 for these sets of parameters, a glance at Table II suggests the existence of even stronger bounds.

4. Codes

In this section, explicit constructions that give codes that determine $N(r, \nu, d_{free})$ for all positive integers n, ν and $d_{free} \in \{3, 4\}$, will be presented, as will also be tables of codes of larger free distance obtained from computer search.

4.1. Constructions for small minimum distances

We observe that if a combined matrix contains a combined column $\mathbf{h}$ as well as some shift $\mathbf{h}\overset{l}{\rightarrow}$ of $\mathbf{h}$, then the corresponding convolutional code will have minimum distance less than two. We will therefore avoid the use of combined columns $\mathbf{h} = (h_0^{(1)}, \ldots, h_{\nu_1}^{(1)}, \ldots, h_0^{(r)}, \ldots, h_{\nu_r}^{(r)})$ such that

$$h_0^{(i)} = 0, \quad \forall i \in \{1, \ldots, r\}. \tag{3}$$

Theorem 4.1. *For $\nu \geq 1$ and $r \geq 1$, $N(r, \nu, 3) = (2^r - 1)2^\nu$.*

Proof. Wyner and Ash [7] noted that an $(n, n-1, \nu, 3)$ code can be constructed by selecting the parity check matrix $\mathbf{H}$ as the matrix consisting of all distinct $(\nu+1)$-dimensional vectors $\mathrm{h} = (h_0, \ldots, h_\nu)$ that have first entry $h_0 = 1$.
For $r > 1$, we can similarly choose $\mathcal{H}$ as the matrix in which the set of combined columns consists of all distinct $(\nu + r)$-dimensional vectors $\mathrm{h} = (h_0^{(1)}, \ldots, h_{\nu_1}^{(1)}, \ldots, h_0^{(r)}, \ldots, h_{\nu_r}^{(r)})$ except those on the form (3). The number of such vectors is $2^{\nu+r} - 2^\nu$.

□

Theorem 4.2. *For $\nu \geq 2$, $N(1, \nu, 4) = 2^{\nu-1}$.*

Proof. "$\leq$" was shown in Lemma 3.2.
"$\geq$:" There are at least three (and probably more) nonequivalent constructions: (i) Select the parity check matrix $\mathbf{H}$ such that it consists of all distinct columns h of *odd* weight except those that satisfy (3). (ii) Select the parity check matrix $\mathbf{H}$ such that it consists of all distinct columns h except those that satisfy (3) or

$$h_\nu^{(1)} = 0.$$

(iii) A column $\mathrm{h} = (h_0, \ldots, h_\nu)$ can be expressed in terms of a polynomial $h(x) = \sum_{i=0}^{\nu} h_i x^i$. Replacing, in construction (ii), exactly one of the even weight columns $h(x) = (1+x)h^*(x)$ with $h^*(x)$, we obtain a new parity check matrix which can be shown to define another code with free distance 4.

□

Theorem 4.3. *For $\nu \geq 1$ and $r > 1$, $N(r, \nu, 4) = 2^{\nu+r-1}$.*

Proof. "$\leq$" was shown in Lemma 3.3.
"$\geq$:" A $(n, n-r, \nu, 4)$ code can be constructed as follows. Let the first element ν_1 of the constraint length vector be zero, the remainder of the constraint length vector can be chosen arbitrarily subject to $\sum_{j=2}^{r} \nu_j = \nu$. Select the combined parity check matrix $\mathcal{H}$ such that it consists of all distinct combined columns h that have "1" as their first entry. Since the columns are distinct the free distance is at least 3, and since the first row of $\mathcal{H}$ is the all-one vector and $\nu_1 = 0$, each code word has even Hamming weight in each block.

□

We note that the codes described here can be thought of as the convolutional code counterpart to the Hamming codes and its even weight subcodes.

4.2. Computer search results

A computer search (described in [8]) has been used to find lower bounds (non-exhaustive search) and exact values (exhaustive search) on $N(r, \nu, d_{\mathrm{free}})$ for $r \leq 4$ and moderate values of ν and d_{free}. The results of this search are presented in Tables II-V (values of $N(r, \nu, d_{\mathrm{free}})$) and Tables VI-IX (parity check matrices of codes).

The bounds in Tables II-V are printed in the format "lower bound"-"upper bound"; if the bounds coincide, the exact value of $N(r, \nu, d_{\text{free}})$ is printed. Super- and subscripts indicate remarks to the upper and lower bounds, respectively.

The upper bounds in Tables II-V are established by Theorem 3.1, with the exception of the upper bounds superscripted by S, T, and Y, which stem from complete computer search, Theorem 3.4, and [9], respectively.

The lower bounds in Tables II-V are established by codes that can be found in [10, ch. 11], or by simple modifications to stronger codes, with the following exceptions: Lower bounds subscripted by D, L, J, Y, and A are established by codes that are described in [11], [12], [13], [9], and provided by one referee, respectively. Lower bounds in **boldface** are established by new codes found by computer search. Parity check matrices for these codes are provided in Tables VI-IX.

Table II $N(1, \nu, d_{\text{free}})$.

d_{free}	$\nu \rightarrow$										
$\downarrow$	2	3	4	5	6	7	8	9	10	11	12
5	2	2	3	4^{T}	**6**$^{\text{T}}$	**8**$^{\text{S}}$	**10**-13^{T}	**13**-18^{T}	**16**-26^{T}	**20**-36^{T}	**24**-52^{T}
6		2	2^{S}	3^{S}	4^{S}	**6**$^{\text{S}}$	**8**$^{\text{S}}$	**10**-	**12**-	**16**-	**20**-
7			2	2	3	3^{S}	4^{P}	4^{S}	5-11	**6**-14	**7**-18
8				2	2^{S}	3^{P}	3^{S}	4^{P}	**5**$^{\text{S}}$	5-11	**6**-14
9					2	2	2	3	3	3-4	**4**-5
10					2	2	2	2^{S}	3	3-4	3-4
11							2	2	2	2^{S}	**3**
12							2	2	2	2^{P}	2^{S}

Table III $N(2, \nu, d_{\text{free}})$.

d_{free}	$\nu \rightarrow$										
$\downarrow$	1	2	3	4	5	6	7	8	9	10	11
5	3	4	**5**	**8**	**10**-11	**13**-17	**17**-23	**22**-37	26-62	32-89	40-126
6		3	4	**6**	**8**$^{\text{Y}}$	**10**-12	**13**-18	**17**-24	**21**-38	24-63	32-90
7		3	3	4	**5**	6-7	**7**-11	8-12	**9**-17	10-22	12-28
8		3	3	**4**	4^{S}	**5**-6	6-8	**7**-11	8-12	10-17	**11**-22
9			3	3	3	4	4-5	5	6	6-7	**7**-10
10			3	3	3	4	4	**5**	5-6	6-7	6-8

Table III (Continued) $N(2, \nu, d_{\rm free})$.

$d_{\rm free}$	$\nu \rightarrow$										
$\downarrow$	1	2	3	4	5	6	7	8	9	10	11
11				3	3	3	**4**	4	4-5	4-5	5-6
12				3	3	3	3	4	4	4-5	**5**-6

Table IV $N(3, \nu, d_{\rm free})$.

$d_{\rm free}$	$\nu \rightarrow$									
$\downarrow$	1	2	3	4	5	6	7	8	9	10
5	$4^{\rm S}$	6	**8**	**11**	**16**-17	**21**-23	24-37	30-62	39-89	48-126
6	4	$\mathbf{5}^{\rm S}$	**7**	**9**	**12**	**17**-18	**21**-24	24-38	30-63	36-90
7	**4**	4	5	**7**	**8**-9	9-11	**11**-15	12-22	**14**-24	**16**-34
8		4	**5**	**6**	**7**-8	$9_{\rm A}$-10	$10_{\rm A}$-12	**11**-16	12-22	15-25
9		4	4	5	5	6	**7**	**8**	9-10	9-11
10		4	4	**5**	5	6	6-7	**7**-8	**8**-9	9-10
11			4	4	**5**	5	**6**	6	**7**	7-8
12			4	4	4	**5**	5-6	6	6-7	**7**-8
13			4	4	4	4	5	5	5-6	6
14				4	4	4	**5**	5	5	6

Table V $N(4, \nu, d_{\rm free})$.

$d_{\rm free}$	$\nu \rightarrow$						
$\downarrow$	1	2	3	4	5	6	7
5	6	8	**11**	**17**	**22**-23	26-37	34-62
6	$6_{\rm L}$	**7**	**9**	$12_{\rm J}$	**18**	**24**	26-38
7	5	$6^{\rm S}$	8	9	**11**	12-15	**15**-23
8	5	6	$8_{\rm L}$	$\mathbf{9}_{\rm Y}$	**10**	**11**-12	**13**-16
9	**5**	5	6	**7**	7	8	9-10
10		5	6	6	**7**	8	**9**
11-12		5	5	6	6	**7**	7-8
13		$5_{\rm D}$	5	5	6	6	6-7
14			5	5	5	6	6-7
15			5	5	5	6	6
16			$5_{\rm D}$	5	5	$\mathbf{6}_{\rm Y}$	6

Tables VI-IX contain (combined) parity check matrices of new codes. For $r = 1$, the "parameters" printed are (length,dimension,constraint length,free distance). For $r > 1$, the "parameters" are (length,dimension,(constraint length vector),free distance), and the combined parity check matrices are printed in a compressed format. The matrices can be expanded as in the following example:

Example. Let $r = 3$. A combined parity check matrix for a $(5, 2, (1, 1, 0), 6)$ code is listed in compressed format as (6,20,23,24,37). Write these octal numbers as binary column vectors, least significant bit on top, and insert boundaries as prescribed by the constraint length vector. Thus

$$\mathcal{H} = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ - & - & - & - & - \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ - & - & - & - & - \\ 0 & 1 & 1 & 1 & 1 \end{pmatrix},$$

where $\mathbf{H}^{(1)}$ consists of the first two rows, $\mathbf{H}^{(2)}$ consists of the third and fourth rows, an $\mathbf{H}^{(3)}$ consists of the last row.

Table VI Parity check matrices in octal format of some codes with $r = 1$.

Parameters	Parity check matrix (octal)
(6,5,6,5)	(77, 105, 131, 135, 147, 163)
(8,7,7,5)	(127, 155, 217, 237, 273, 311, 325, 345)
(6,5,7,6)	(153, 155, 211, 235, 263, 367)
(10,9,8,5)	(265, 267, 313, 375, 413, 423, 471, 657, 727, 755)
(8,7,8,6)	(103, 373, 421, 501, 531, 637, 675, 705)
(13,12,9,5)	(1257, 1261, 1305, 1341, 1433, 1465, 1631, 427, 1713, 651, 1403, 467, 677)
(10,9,9,6)	(1261, 1341, 1077, 1407, 1767, 1555, 1533, 417, 765, 1445)
(16,15,10,5)	(3501, 2733, 3051, 3727, 2325, 1253, 2477, 2143, 3367, 1617, 3515, 1563, 2271, 1755, 2131, 2337)
(12,11,10,6)	(3075, 1731, 2333, 3271, 2563, 3361, 3433, 1751, 1055, 3537, 1137, 3147)
(5,4,10,8)	(2027, 2431, 2725, 3323, 3637)
(20,19,11,5)	(2007, 4771, 1731, 2571, 5021, 2557, 3655, 7357, 2063, 6527, 6025, 4413, 4751, 7367, 3271, 5735, 7457, 3003, 7741, 7063)
(16,15,11,6)	(4157, 4627, 3711, 2653, 3441, 5221, 6455, 6575, 7245, 2143, 5545, 5337, 6051, 7223, 7571, 5405)
(6,5,11,7)	(4035, 4457, 5005, 4167, 6113, 5443)

Table VI (Continued) Parity check matrices in octal format of some codes with $r = 1$.

Parameters	Parity check matrix (octal)
(24,23,12,5)	(13223, 10521, 12477, 12617, 16747, 16643, 6653, 11271, 7161, 13645, 14731, 15671, 14105, 15065, 11101, 11525, 17327, 17243, 17035, 12441, 6237, 14515, 6567, 12333)
(20,19,12,6)	(10503, 5263, 13371, 15311, 10175, 11405, 14345, 13761, 7261, 13731, 7407, 14747, 12651, 12623, 3755, 16617, 13027, 14673, 16077, 11667)
(7,6,12,7)	(15517, 10641, 15761, 16655, 10073, 15323, 14231)
(6,5,12,8)	(11067, 17657, 10553, 14525, 12205, 15115)
(4,3,12,9)	(7747, 12545, 14713, 11071)
(3,2,12,11)	(10417, 14475, 16173)

Table VII Parity check matrices in octal compressed format of some codes with $r = 2$.

Parameters	Parity check matrix (octal compressed)
(5,3,(3,0),5)	(11, 17, 20, 27, 32)
(8,6,(2,2),5)	(14, 16, 35, 43, 53, 61, 67, 74)
(6,4,(2,2),6)	(7, 14, 51, 67, 71, 72)
(4,2,(2,2),8)	(7, 25, 73, 76)
(10,8,(3,2),5)	(171, 157, 55, 123, 121, 126, 113, 166, 72, 115)
(8,6,(3,2),6)	(37, 135, 147, 122, 160, 171, 34, 155)
(5,3,(5,0),7)	(47, 100, 124, 141, 173)
(13,11,(3,3),5)	(134, 237, 362, 315, 351, 174, 325, 367, 137, 57, 111, 226, 353)
(10,8,(3,3),6)	(277, 230, 337, 265, 174, 345, 57, 301, 136, 331)
(5,3,(6,0),8)	(111, 200, 255, 313, 376)
(17,15,(4,3),5)	(663, 231, 75, 753, 515, 370, 627, 640, 625, 467, 746, 223, 675, 654, 160, 775, 137)
(13,11,(4,3),6)	(121, 466, 75, 551, 711, 260, 564, 272, 403, 477, 507, 567, 631)
(7,5,(4,3),7)	(515, 615, 165, 273, 474, 540, 576)
(4,2,(4,3),11)	(133, 31, 776, 657)
(22,20,(4,4,),5)	(1750, 621, 1415, 1044, 1121, 1667, 766, 1641, 264, 333, 237, 471, 1131, 437, 1576, 1151, 1775, 1646, 52, 1235, 1735, 1747)
(17,15,(4,4),6)	(1072, 625, 223, 1053, 1313, 366, 174, 425, 1363, 1513, 671, 1555, 1165, 1047, 1572, 1666, 1301)
(7,5,(4,4),8)	(1474, 137, 667, 1755, 1763, 1746, 1607)
(5,3,(4,4),10)	(1754, 1021, 661, 1113, 1066)

Table VII (Continued) Parity check matrices in octal compressed format of some codes with $r = 2$.

Parameters	Parity check matrix (octal compressed)
(21,19,(5,4),6)	(3457, 1544, 3716, 2527, 3133, 1346, 3245, 451, 3704, 3166, 3425, 2327, 3734, 2553, 174, 1375, 3161, 1053, 3241, 1657, 2546)
(9,7,(5,4),7)	(465, 562, 3743, 1745, 2305, 2515, 2237, 3433, 3667)
(11,9,(6,5),8)	(13604, 553, 1737, 13173, 12257, 10732, 4155, 3377, 16637, 15631, 11515)
(7,5,(6,5),9)	(16427, 15343, 6155, 17660, 12376, 3501, 6735)
(5,3,(6,5),12)	(1363, 717, 12431, 14155, 14610)

Table VIII Parity check matrices in octal compressed format of some codes with $r = 3$.

Parameters	Parity check matrix (octal compressed)
(4,1,(1,0,0),7)	(3, 4, 10, 15)
(5,2,(1,1,0),6)	(6, 20, 23, 24, 37)
(8,5,(1,1,1),5)	(47, 16, 66, 26, 11, 23, 65, 71)
(7,4,(1,1,1),6)	(36, 56, 7, 62, 27, 41, 35)
(5,2,(1,1,1),8)	(46, 34, 41, 23, 37)
(11,8,(2,1,1),5)	(143, 155, 33, 125, 35, 34, 60, 162, 172, 132, 46)
(9,6,(2,1,1),6)	(76, 177, 141, 31, 171, 103, 60, 116, 65)
(7,4,(2,1,1),7)	(136, 31, 147, 151, 170, 125, 153)
(6,3,(2,1,1),8)	(177, 144, 31, 45, 172, 112)
(5,2,(2,1,1),10)	(74, 44, 147, 32, 127)
(16,13,(2,2,1),5)	(5, 162, 256, 154, 327, 275, 330, 342, 117, 303, 325, 263, 300, 353, 57, 125)
(12,9,(2,2,1),6)	(252, 147, 175, 211, 114, 277, 134, 225, 142, 227, 232, 153)
(8,5,(2,2,1),7)	(333, 37, 173, 374, 273, 147, 210, 205)
(7,4,(2,2,1),8)	(57, 114, 256, 326, 301, 373, 61)
(5,2,(2,2,1),11)	(313, 251, 304, 163, 47)
(21,18,(2,2,2),5)	(175, 106, 516, 126, 336, 256, 527, 347, 710, 41, 722, 653, 745, 360, 453, 477, 305, 773, 54, 712, 627)
(17,14,(2,2,2),6)	(517, 7, 471, 142, 674, 542, 74, 737, 667, 167, 611, 314, 577, 171, 732, 27, 352)
(9,6,(1,2,3),8)	(156, 255, 277, 363, 574, 216, 772, 413, 407)(156, 255, 277, 363, 574, 216, 772, 413, 407)
(5,2,(2,2,2),12)	(156, 477, 562, 636, 117)
(21,18,(3,2,2),6)	(1571, 703, 31, 1367, 1047, 1711, 255, 570, 314, 1420, 1230, 1374, 636, 107, 656, 1572, 1363, 706, 1453, 172, 1767)

Table VIII (Continued) Parity check matrices in octal compressed format of some codes with $r = 3$.

Parameters	Parity check matrix (octal compressed)
(11,8,(1,3,3),7)	(1447, 306, 246, 1341, 1302, 1574, 1127, 471, 446, 522, 1257)
(10,7,(1,3,3),8)	(1166, 237, 1513, 1221, 6, 1607, 1172, 163, 1544, 1527)
(7,4,(3,2,2),9)	(1173, 57, 737, 574, 1120, 1614, 1247)
(6,3,(3,2,2),11)	(1427, 377, 1551, 630, 316, 1032)
(5,2,(3,2,2),14)	(1714, 155, 476, 1565, 1625)
(11,8,(4,4,0),8)	(1250, 3471, 1760, 2074, 2706, 3140, 2733, 421, 3037, 1175, 2216)
(8,5,(3,3,2),9)	(2036, 1726, 3041, 1230, 2073, 3600, 1477, 741)
(7,4,(3,3,2),10)	(3524, 1362, 3653, 3721, 1115, 416, 2163)
(14,11,(5,4,0),7)	(7252, 6776, 4645, 5104, 4036, 7215, 3211, 6747, 2453, 6220, 744, 5017, 1152, 5764)
(8,5,(3,3,3),10)	(3641, 7524, 7653, 330, 4227, 7427, 1550, 6602)
(7,4,(3,3,3),11)	(4320, 1223, 1471, 6147, 237, 4415, 7617)
(16,13,(4,3,3),7)	(13475, 14440, 11500, 11523, 540, 3563, 11615, 6657, 10025, 15001, 2377, 1030, 11770, 12354, 11172, 603)
(7,4,(4,3,3),12)	(14427, 13615, 17705, 15242, 17120, 5077, 6365)

Table IX Parity check matrices in octal compressed format of some codes with $r = 4$.

Parameters	Parity check matrix (octal compressed)
(5,1,(1,0,0,0),9)	(3, 4, 10, 20, 35)
(7,3,(1,1,0,0),6)	(24, 37, 61, 41, 55, 70, 77)
(11,7,(1,1,1,0),5)	(17, 34, 154, 27, 62, 157, 130, 165, 150, 51, 44)
(9,5,(1,1,1,0),6)	(105, 135, 102, 166, 144, 170, 172, 154, 141)
(17,13,(1,1,1,1),5)	(255, 75, 375, 62, 6, 146, 112, 172, 236, 3, 343, 141, 365, 106, 364, 334, 213)
(9,5,(0,0,2,2),8)	(342, 73, 333, 132, 326, 217, 57, 56, 174)
(7,3,(1,1,1,1),9)	(126, 46, 112, 232, 201, 373, 346)
(22,18,(2,1,1,1),5)	(260, 326, 437, 656, 545, 162, 30, 265, 423, 602, 121, 116, 703, 712, 611, 76, 650, 355, 66, 272, 472, 473)
(18,14,(2,1,1,1),6)	(510, 164, 60, 610, 722, 655, 326, 711, 663, 150, 453, 350, 413, 614, 527, 515, 660, 321)
(11,7,(2,1,1,1),7)	(704, 762, 261, 612, 552, 605, 671, 735, 274, 445, 425)
(10,6,(2,1,1,1),8)	(353, 265, 430, 223, 320, 560, 674, 565, 71, 127)
(7,3,(2,1,1,1),10)	(340, 265, 460, 527, 614, 435, 570)

Table IX (Continued) Parity check matrices in octal compressed format of some codes with $r = 4$.

Parameters	Parity check matrix (octal compressed)
(24,20,(2,1,1,1),6)	(555, 506, 536, 577, 1100, 1150, 1156, 1500, 1551, 210, 215, 677, 570, 1252, 1601, 1145, 144, 1261, 1664, 1137, 1674, 635, 507, 661)
(11,7,(2,2,1,1),8)	(600, 614, 1631, 1140, 1540, 1517, 127, 1607, 155, 1156, 561)
(7,3,(2,2,1,1),12)	(255, 1713, 1116, 250, 1410, 626, 1377)
(6,2,(0,2,2,2),16)	(727, 1425, 1703, 405, 775, 1537)
(15,11,(2,2,2,1),7)	(147, 1140, 1541, 1677, 3777, 1415, 3003, 2773, 2336, 2346, 3060, 3720, 1412, 2423, 2436)
(13,9,(2,2,2,1),8)	(1172, 1775, 2311, 2430, 1524, 2257, 1471, 1414, 1400, 1227, 2543, 2645, 2702)
(9,5,(3,2,2,0),10)	(3102, 1113, 2174, 2505, 2631, 3642, 430, 2426, 775)

5. Conclusion

We have introduced and investigated the function $N(r, \nu, d_{\text{free}})$, defined as the maximum n such that there exists a binary convolutional code of block length n, dimension $n - r$, constraint length ν, and free distance d_{free}. For $d_{\text{free}} \in \{3, 4\}$, codes were obtained from simple constructions that determine $N(r, \nu, d_{\text{free}})$ for all positive integers ν and r. Codes with minimum Hamming distance $d_{\text{free}} \geq 5$ were found by computer search. New efficient algorithms were derived to perform exhaustive and non-exhaustive computer search. A number of new codes, with specified constraint lengths, redundancies, and free distances, and higher code rates than previously reported codes, were presented. Not surprisingly, when the dimension is at least as large as the constraint length, the new codes turn out to be unit memory codes ([14], [12], [13]).

References

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

[2] J. L. Massey and M. K. Sain, "Inverses of linear sequential circuits," *IEEE Trans. Comput.*, vol. C-17, pp. 330–337, April 1968.

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

[4] G. D. Forney, Jr., "Structural analysis of convolutional codes via dual codes," *IEEE Trans. on Information Theory*, vol. IT-19, pp. 512–518, July 1973.

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

[6] 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.

[7] A. D. Wyner and R. B. Ash, "Analysis of recurrent codes," *IEEE Trans. on Information Theory*, vol. IT-9, pp. 143–156, 1963.

[8] Ø. Ytrehus, "A note on high rate convolutional codes," Department report 68, Department of Informatics, University of Bergen, August 1992.

[9] Ø. Ytrehus, "Constructions and proofs of nonexistence of some convolutional codes," Department report 60, Department of Informatics, University of Bergen, April 1992.

[10] S. Lin and D. J. Costello, Jr., *Error Control Coding: Fundamentals and Applications*. Prentice-Hall, 1983.

[11] D. G. Daut, J. W. Modestino, and L. D. Wismer, "New short constraint length convolutional code constructions for selected rational rates," *IEEE Trans. on Information Theory*, vol. IT-28, pp. 794–800, Sept. 1982.

[12] G. S. Lauer, "Some optimal partial-unit-memory codes," *IEEE Trans. on Information Theory*, vol. IT-25, pp. 240–243, March 1979.

[13] J. Justesen, E. Paaske, and M. Ballan, "Quasi-cyclic unit memory convolutional codes," *IEEE Trans. on Information Theory*, vol. IT-36, pp. 540–547, May 1990.

[14] L. N. Lee, "Short, unit-memory, byte-oriented, binary convolutional codes having maximal free distance," *IEEE Trans. on Information Theory*, vol. IT-22, pp. 349–352, May 1976.