Construction and decoding of matrix-product codes from nested codes*

Fernando Hernando† Kristine Lally‡ Diego Ruano§

Abstract

We consider matrix-product codes $[C_1 \cdots C_s] \cdot A$, where $C_1, \ldots, C_s$ are nested linear codes and matrix A has full rank. We compute their minimum distance and provide a decoding algorithm when A is a non-singular by columns matrix. The decoding algorithm decodes up to half of the minimum distance.

1 Introduction

We consider the matrix-product construction, $[C_1 \cdots C_s] \cdot A$, introduced by Blackmore and Norton in [2] which may also be seen as a generalization of the Plotkin $(u \mid u+v)$-construction. In [2] a lower bound for the minimum distance when the matrix A has a certain property, called non-singular by columns, is obtained. Moreover, this bound is sharp if the non-singular by columns matrix is triangular. The same construction and a similar lower bound for the minimum distance are also given by Özbudak and Stichtenoth in [9], but for an arbitrary matrix A. In the particular case of non-singular by columns matrices the bounds coincide. The construction in [9] is a generalization of the one in [8]. See also [1] and [7] for other generalizations.

In this paper we prove that the lower bound given in [9] is sharp when the codes $C_1, \ldots, C_s$ are nested. This key property also enables an efficient decoding procedure for a class of matrix-product codes. In section 3 we provide a decoding algorithm for matrix-product codes where the codes $C_1, \ldots, C_s$ are nested and the matrix A is non-singular by columns. This decoding algorithm decodes up to half of the minimum distance.

In section 4 we consider the matrix-product construction when $C_1, \ldots, C_s$ are cyclic codes. The resulting codes are quasi-cyclic codes. The quasi-cyclic class of codes is the natural generalization of the cyclic class, and acquired a special interest after it was shown that some codes in this class meet a modified

*The work of F. Hernando is supported in part by the Claude Shannon Institute, Science Foundation Ireland Grant 06/MI/006 (Ireland) and by MEC MTM2007-64704 and Junta de CyL VA025A07 (Spain). The work of D. Ruano is supported in part by DTU, H.C. Oersted post doc. grant (Denmark) and by MEC MTM2007-64704 and Junta de CyL VA065A07 (Spain)

†Department of Mathematics, University College Cork, Cork, Ireland. F.Hernando@ucc.ie

‡Department of Mathematics and Statistics, RMIT University, Melbourne, Australia. kristine.lally@rmit.edu.au

§Department of Mathematics, Technical University of Denmark, Matematiktorvet, Building 303, DK-2800, Kgs. Lyngby, Denmark. D.Ruano@mat.dtu.dk

Gilbert-Varshamov bound [3]. Applying our previous results to this case, we provide the minimum distance and a decoding algorithm for a class of quasi-cyclic codes.

2 Matrix-Product Codes

We follow the Blackmore and Norton approach in [2] to introduce matrix-product codes. With this setting, one can define large codes from small ones.

Definition 2.1. Let $C_1, \ldots, C_s \subset \mathbb{F}_q^m$ be linear codes of length m and a matrix $A = (a_{i,j}) \in \mathcal{M}(\mathbb{F}_q, s \times l)$, with $s \leq l$. The **matrix-product code** $C = [C_1 \cdots C_s] \cdot A$ is the set of all matrix-products $[c_1 \cdots c_s] \cdot A$ where $c_i \in C_i$ is an $m \times 1$ column vector $c_i = (c_{1,i}, \ldots, c_{m,i})^T$ for $i = 1, \ldots, s$. Therefore, a typical codeword $\mathbf{c}$ is

$$\mathbf{c} = \begin{pmatrix} c_{1,1}a_{1,1} + \cdots + c_{1,s}a_{s,1} & \cdots & c_{1,1}a_{1,l} + \cdots + c_{1,s}a_{s,l} \\ \vdots & \ddots & \vdots \\ c_{m,1}a_{1,1} + \cdots + c_{m,s}a_{s,1} & \cdots & c_{m,1}a_{1,l} + \cdots + c_{m,s}a_{s,l} \end{pmatrix}. \quad (1)$$

The matrix-product construction is also presented by Özbudak and Stichtenoth in [9]. Clearly the i-th column of any codeword is an element of the form $\sum_{j=1}^{s} a_{j,i}c_j \in \mathbb{F}_q^m$, therefore reading the entries of the $m \times l$-matrix above in column-major order, the codewords can be viewed as vectors of length ml,

$$\mathbf{c} = \left(\sum_{j=1}^{s} a_{j,1}c_j, \ldots, \sum_{j=1}^{s} a_{j,l}c_j \right) \in \mathbb{F}_q^{ml}. \quad (2)$$

Henceforth we will use either notation (1) or (2) as required without further comment.

From the above construction it follows that a generator matrix of C is of the form:

$$G = \begin{pmatrix} a_{1,1}G_1 & a_{1,2}G_1 & \cdots & a_{1,s}G_1 & \cdots & a_{1,l}G_1 \\ a_{2,1}G_2 & a_{2,2}G_2 & \cdots & a_{2,s}G_2 & \cdots & a_{2,l}G_2 \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ a_{s,1}G_s & a_{s,2}G_s & \cdots & a_{s,s}G_s & \cdots & a_{s,l}G_s \end{pmatrix},$$

where G_i is a generator matrix of C_i, $i = 1, \ldots, s$. Moreover, if C_i is a $[m, k_i, d_i]$ code then one has that $[C_1 \cdots C_s] \cdot A$ is a linear code over $\mathbb{F}_q$ with length lm and dimension $k = k_1 + \cdots + k_s$ if the matrix A has full rank and $k < k_1 + \cdots + k_s$ otherwise. In the following we assume A has full rank.

Let us denote by $R_i = (a_{i,1}, \ldots, a_{i,l})$ the element of $\mathbb{F}_q^l$ consisting of the i-th row of A, for $i = 1, \ldots, s$. We denote by D_i the minimum distance of the code C_{R_i} generated by $\langle R_1, \ldots, R_i \rangle$ in $\mathbb{F}_q^l$. In [9] the following lower bound for the minimum distance of the matrix-product code C is obtained,

$$d(C) \geq \min\{d_1D_1, d_2D_2, \ldots, d_sD_s\}, \quad (3)$$

where d_i is the minimum distance of C_i.

Considering the lower bound (3), since $D_1 \geq \cdots \geq D_s$, it is clear that, to obtain a matrix-product code with good parameters, it is advisable to choose C_j with minimum distance greater than or equal to that of C_{j-1}. To this aim it is therefore also advisable to choose C_j with dimension less than or equal to that of C_{j-1}. In particular it is wise to choose C_1 with large dimension and C_s with large minimum distance. For this reason, henceforth we will assume that $C_1, \ldots, C_s$ are nested codes, $C_1 \supset C_2 \supset \cdots \supset C_s$, and this condition is not particularly restrictive. It follows that $d_1 \leq \cdots \leq d_s$. Introducing this property allows us to obtain several results: the following theorem for computing the minimum distance and a decoding algorithm in the next section.

Theorem 2.2. *Let C be the matrix-product code $[C_1 \cdots C_s] \cdot A$ where $C_1 \supset C_2 \supset \cdots \supset C_s$ are linear codes and matrix $A \in \mathcal{M}(\mathbb{F}_q, s \times l)$ has full rank. Then, the minimum distance of C is*

$$d(C) = \min\{d_1 D_1, d_2 D_2, \ldots, d_s D_s\}, \tag{4}$$

where $d_i = d(C_i)$, $D_i = d(C_{R_i})$ and C_{R_i} is as described above.

Proof. From [9] we have $d(C) \geq \min\{d_1 D_1, d_2 D_2, \ldots, d_s D_s\}$ (this result is also valid for $C_1, \ldots, C_s$ non-nested). We include a rewriting of the proof here for completeness. Any codeword of C is of the form $\mathbf{c} = [c_1 \cdots c_s] \cdot A$. Let us suppose that $c_r \neq 0$ and $c_i = 0$ for all $i > r$. It follows that $[c_{j,1} \cdots c_{j,s}] \cdot A \in C_{R_r}$ for $j = 1, \ldots, m$, where $c_{j,i}$ is the j-th component of the word c_i. Since $c_r \neq 0$ it has at least d_r non-zero components. Suppose $c_{i_v,r} \neq 0$, for $v = 1, \ldots, d_r$. For each $v = 1, \ldots, d_r$, the product $[c_{i_v,1} \cdots c_{i_v,s}] \cdot A$ is a non-zero codeword in C_{R_r}, since A has full rank. Therefore the weight of $[c_{i_v,1} \cdots c_{i_v,s}] \cdot A$ is greater than or equal to D_r and the weight of $\mathbf{c}$ is greater than or equal to $d_r D_r$.

In order to see that the above bound is sharp, we construct a codeword with weight $d_r D_r$, for each $r = 1, \ldots, s$. Choose $c_1, \ldots, c_r$ such that $c_1 = \cdots = c_r$, (this is possible since the codes are nested), with $wt(c_1) = d_r$, and $c_{r+1} = \ldots = c_s = 0$. Let $f = \sum_{i=1}^{r} f_i R_i$, with $f_i \in \mathbb{F}_q$, be a word in C_{R_r} of weight D_r. If $c'_i = f_i c_i$ then

$$\left(\sum_{j=1}^{s} a_{j,1} c'_j, \ldots, \sum_{j=1}^{s} a_{j,l} c'_j \right) = c_1 \left(\sum_{j=1}^{r} a_{j,1} f_j, \ldots, \sum_{j=1}^{r} a_{j,l} f_j \right) = c_1 f,$$

is a codeword in C with weight $d_r D_r$, and the result holds. □ □

Example 2.3. Consider the ternary linear codes C_1, C_2, C_3 with generator matrices

$$G_1 = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 2 & 1 \\ 0 & 0 & 1 \end{pmatrix}, G_2 = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 2 & 1 \end{pmatrix}, G_3 = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix},$$

respectively. The parameters of these codes are $[3, 3, 1]$, $[3, 2, 2]$ and $[3, 1, 3]$, respectively, and the codes are nested, $C_1 \supset C_2 \supset C_3$. Here $d_1 = 1$, $d_2 = 2$ and $d_3 = 3$.

We consider the matrix-product code $C = [C_1C_2C_3] \cdot A$ where

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 2 & 1 \\ 0 & 0 & 1 \end{pmatrix}. \tag{5}$$

The minimum distances of the codes $C_{R_i}, i = 1, 2, 3$ obtained from the matrix A are $D_1 = 3$, $D_2 = 2$ and $D_3 = 1$.

On construction we find that C is a $[9, 6]$ code. Applying Theorem 2.2 we find that the minimum distance of C is $\min\{d_1D_1, d_2D_2, d_3D_3\} = 3$. We note that 3 is the largest possible minimum distance for a $[9, 6]$ linear code over $\mathbb{F}_3$.

Remark 2.4. If the codes $C_1, \ldots, C_s$ are not nested then the lower bound (3) for the minimum distance is not necessarily sharp, that is, Theorem 2.2 does not hold, as the following example shows.

Example 2.5. Consider the ternary cyclic codes C_1, C_2, C_3, C_4 with generator polynomials $f_1 = (x + 2)(x + 1)$, $f_2 = (x^2 + 1)(x + 2)$, $f_3 = (x^2 + 1)$ and $f_4 = (x^2 + 1)(x + 1)$ respectively. Notice that since $f_1 \nmid f_2$, the codes are not nested. The parameters of these codes are $[4, 2, 2]$, $[4, 1, 4]$, $[4, 2, 2]$ and $[4, 1, 4]$ respectively. So here $d_1 = 2$, $d_2 = 4$, $d_3 = 2$ and $d_4 = 4$.

Consider $C = [C_1C_2C_3C_4] \cdot A$, where

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{6}$$

The minimum distances of the codes $C_{R_i}, i = 1, \ldots, 4$ obtained from the matrix A are $D_1 = 4$, $D_2 = 1$, $D_3 = 1$ and $D_4 = 1$.

On construction we find that C is a $[16, 6, 4]$ code. However, the lower bound for the minimum distance is only $\min\{d_1D_1, d_2D_2, d_3D_3, d_4D_4\} = 2$.

In [2], the following condition for the matrix A is introduced.

Definition 2.6. [2] Let A be a $s \times l$ matrix and A_t be the matrix consisting of the first t rows of A. For $1 \leq j_1 < \cdots < j_t \leq l$, we denote by $A(j_1, \ldots, j_t)$ the $t \times t$ matrix consisting of the columns $j_1, \ldots, j_t$ of A_t.

A matrix A is **non-singular by columns** if $A(j_1, \ldots, j_t)$ is non-singular for each $1 \leq t \leq s$ and $1 \leq j_1 < \cdots < j_t \leq l$. In particular, a non-singular by columns matrix A has full rank.

By [2] a non-singular by columns matrix A defines MDS codes C_{R_i}, for $i = 1, \ldots, s$. Also in [2] the following lower bound for the minimum distance of a matrix-product code with A non-singular by columns is obtained,

$$d(C) \geq \min\{ld_1, (l - 1)d_2, \ldots, (l - s + 1)d_s\}. \tag{7}$$

This bound is the particular case of (3) when $D_i = l - i + 1$. Moreover, it was shown in [2] that if A is non-singular by columns and triangular, (i.e. it is a column permutation of an upper triangular matrix), then the bound (7) for the minimum distance is sharp. Here we can deduce, by applying Theorem 2.2, that if A is non-singular by columns and the codes $C_1, \ldots, C_s$ are nested, then this bound (7) is also sharp. It is known from [2] that, for $s \geq 2$ there exists an $s \times l$ non-singular by columns matrix over $\mathbb{F}_q$ if and only if $s \leq l \leq q$.

3 Decoding Algorithm

In this section, we present a decoding algorithm for a class of matrix-product codes: namely, we consider s nested linear codes $C_1, \ldots, C_s \subset \mathbb{F}_q^m$ and a non-singular by columns matrix $A \in \mathcal{M}(\mathbb{F}_q, s \times l)$, where $s \leq l$ (see definition 2.6). We provide a decoding algorithm for the matrix-product code $C = [C_1 \cdots C_s] \cdot A \subset \mathbb{F}_q^{ml}$, assuming that we have a decoding algorithm DC_i for C_i which decodes up to $t_i = \lfloor (d_i - 1)/2 \rfloor$ errors, for $i = 1, \ldots, s$. We assume that each DC_i answers *"failure"* if there is no codeword in C_i within distance $\lfloor (d_i - 1)/2 \rfloor$ of the received word.

A codeword in C has the form $\mathbf{c} = (\sum_{j=1}^{s} a_{j,1} c_j, \ldots, \sum_{j=1}^{s} a_{j,l} c_j)$, where $c_j \in C_j$, for all j. Since we consider $C_1, \ldots, C_s$ to be nested codes, $C_1 \supset \cdots \supset C_s$, each block $\sum_{j=1}^{s} a_{j,i} c_j$ of $\mathbf{c}$ is a codeword in C_1. A first approach to decoding might be to decode each block of a received word by decoder DC_1. However, with this approach, we may not achieve the error correcting capability of the code. A more sophisticated approach takes place here where we use the decoders DC_i, for $i = 1, \ldots, s$, in a such a way that we can always decode up to the full error correcting capability of the code, that is, our decoding algorithm for C decodes up to half of the minimum distance

$$d(C) = \min\{l d_1, (l-1) d_2, \ldots, (l-s+1) d_s\}. \tag{8}$$

We first describe the main steps in our decoding algorithm. The algorithm is outlined as a whole in procedural form in Algorithm 1.

Consider the codeword $\mathbf{c} = (\sum_{j=1}^{s} a_{j,1} c_j, \ldots, \sum_{j=1}^{s} a_{j,l} c_j)$, for some $c_j \in C_j$, for all j. Suppose that $\mathbf{c}$ is sent and that we receive $\mathbf{p} = \mathbf{c} + \mathbf{e}$, where $\mathbf{e} = (e_1, e_2, \ldots, e_l) \in \mathbb{F}_q^{ml}$ is an error vector. Let $\{i_1, \ldots, i_s\} \subset \{1, \ldots, l\}$ be an ordered subset of indices. We now also suppose that $\mathbf{e}$ satisfies the extra property that

$$wt(e_{i_j}) \leq t_j \text{ for all } j \in \{1, \ldots, s\}. \tag{9}$$

We denote by $p_i = \sum_{j=1}^{s} a_{j,i} c_j + e_i \in \mathbb{F}_q^m$ the i-th block of p, for $i = 1, \ldots, l$.

Since $C_1 \supset \cdots \supset C_s$, each block $\sum_{j=1}^{s} a_{j,i} c_j$ of $\mathbf{c}$ is a codeword of C_1. Therefore, we decode the i_1-th block p_{i_1} of $\mathbf{p}$ using DC_1. Since $wt(e_{i_1}) \leq t_1$ we obtain e_{i_1} and $\sum_{j=1}^{s} a_{j,i_1} c_j$.

We do not know c_1, but we can eliminate it in every other block in the following way: we consider a new vector $\mathbf{p}^{2)} \in \mathbb{F}_q^{ml}$ with components

$$p_i^{2)} = p_i - \frac{a_{1,i}}{a_{1,i_1}} (p_{i_1} - e_{i_1}) = \sum_{j=2}^{s} a_{j,i}^{2)} c_j + e_i, \text{ for } i \neq i_1,$$

where $a_{j,i}^{2)} = a_{j,i} - \frac{a_{1,i}}{a_{1,i_1}} a_{j,i_1}$, and $p_{i_1}^{2)} = p_{i_1} - e_{i_1}$. Since A is a non-singular by columns matrix, the elements of the first row of A are non-zero, and so the denominator a_{1,i_1} is non-zero.

Since $C_2 \supset \cdots \supset C_s$, we notice that the i-th block of $\mathbf{p}^{2)}$ is a codeword of C_2 plus the error block e_i, for $i \in \{1, \ldots, s\} \setminus \{i_1\}$. We now decode the i_2-th block $p_{i_2}^{2)} = \sum_{j=2}^{s} a_{j,i_2}^{2)} c_j + e_{i_2}$ of $\mathbf{p}^{2)}$ using DC_2. Since $w(e_{i_2}) \leq t_2$ we obtain e_{i_2} and $\sum_{j=2}^{s} a_{j,i_2}^{2)} c_j$.

As before, we do not know c_2, but we can eliminate it in every other block as follows: we consider a new vector $\mathbf{p}^{3)} \in \mathbb{F}_q^{ml}$ with components

$$p_i^{3)} = p_i^{2)} - \frac{a_{2,i}^{2)}}{a_{2,i_2}^{2)}}(p_{i_2}^{2)} - e_{i_2}) = \sum_{j=3}^{s} a_{j,i}^{3)} c_j + e_i, \text{ for } i \neq i_1, i_2,$$

where $a_{j,i}^{3)} = a_{j,i}^{2)} - \frac{a_{2,i}^{2)}}{a_{2,i_2}^{2)}} a_{j,i_2}^{2)}$, $p_{i_1}^{3)} = p_{i_1}^{2)}$ and $p_{i_2}^{3)} = p_{i_2}^{2)} - e_{i_2}$.

Notice that the i-th block of $\mathbf{p}^{3)}$ is a codeword of C_3 plus the error block e_i, for $i \in \{1, \ldots, s\} \setminus \{i_1, i_2\}$.

Then we iterate this process, defining $\mathbf{p}^{k)}$ for $k = 3, \ldots, s$, and decoding the i_k-th block using DC_k. In this way, we obtain the error blocks e_i, and the corresponding codeword blocks $\sum_{j=1}^{s} a_{j,i} c_j$, for $i \in \{i_1, \ldots, i_s\}$. The vector $(\sum_{j=1}^{s} a_{j,i_1} c_j, \ldots, \sum_{j=1}^{s} a_{j,i_s} c_j)$ formed from these s decoded blocks is equal to the product $[c_1 \cdots c_s] \cdot A(i_1, \ldots, i_s)$, where $A(i_1, \ldots, i_s)$ is the $s \times s$-submatrix of A consisting of the columns $i_1, \ldots, i_s$. Since this matrix is full rank, we can now easily compute $c_1, \ldots, c_s$ by inverting $A(i_1, \ldots, i_s)$ or solving the corresponding linear system. Finally we recover the remaining $l - s$ codeword blocks "for free" (i.e. no decoding procedure is involved for these blocks) by simply recomputing the entire codeword $\mathbf{c} = [c_1 \cdots c_s] \cdot A = (\sum_{j=1}^{s} a_{j,1} c_j, \ldots, \sum_{j=1}^{s} a_{j,l} c_j)$, since we know the c_j's and the matrix A.

For each elimination step in the above procedure, it is necessary that $a_{k,i_k}^{k)} \neq 0$, for each $k = 2, \ldots, s$, to avoid zero division. We now prove that this follows from the non-singular by columns property of A. Let $A^{1)} = A$. The matrix $A^{k)} = (a_{i,j}^{k)}) \in \mathcal{M}(\mathbb{F}_q, s \times l), k = 2, \ldots, s$, is obtained recursively from $A^{k-1)}$ by performing the following $l - (k-1)$ elementary column operations:

$$\text{column}_i(A^{k)}) = \text{column}_i(A^{k-1)}) - \frac{a_{k-1,i}^{k-1)}}{a_{k-1,i_{k-1}}^{k-1)}} \text{column}_{i_{k-1}}(A^{k-1)}),$$

for each $i \notin \{i_1, \ldots, i_{k-1}\}$. These operations introduce $l-(k-1)$ additional zero elements in the $k-1$-th row of $A^{k)}$ at each iteration. Hence the minor of $A^{k)}$ given by the first k rows and the $i_1, \ldots, i_k$ columns, is a triangular matrix (in this case, a column permutation of a lower triangular matrix) whose determinant is $a_{1,i_1}^{k)} \cdots a_{k,i_k}^{k)}$. Since A is non-singular by columns, this minor is non-singular. It follows that the determinant is non-zero, and therefore $a_{k,i_k}^{k)} \neq 0$.

The procedure described above will successfully decode the received word, if $wt(\mathbf{e}) \leq \lfloor (d(C)-1)/2 \rfloor$, and for a given choice of indices $\{i_1, \ldots, i_s\} \subset \{1, \ldots, l\}$, each error block satisfies $wt(e_{i_j}) \leq t_j = \lfloor (d_j - 1)/2 \rfloor$, for all $j = 1, \ldots, s$. The procedure may fail if $wt(e_{i_j}) > t_j$ for some j.

If a decoder DC_j answers *"failure"*, that is, it cannot decode $p_{i_j}^{j)}$, since there is no codeword in C_j within distance t_j to the received block $p_{i_j}^{j)}$, then we consider another ordered subset of indices, and start the procedure again.

Similarly if $wt(e_{i_j}) > t_j$ and the decoder DC_j incorrectly decodes the block $p_{i_j}^{j)}$, then we can detect this at the end by checking whether the entire decoded word is a codeword in C or not (overall failure) and by checking that we have not corrected more than $\lfloor (d(C) - 1)/2 \rfloor$ errors in total (incorrect overall decoding).

In either case we again consider another ordered subset of indices, and restart the procedure.

We now prove that for every error vector $\mathbf{e}$ with $wt(\mathbf{e}) \leq \lfloor (d(C)-1)/2 \rfloor$ there exists a "good" set of indices $\{i_1, \ldots, i_s\} \subset \{1, \ldots, l\}$ satisfying condition (9). We can therefore repeat the procedure described above, with various ordered subsets of indices, until a "good" set of indices is chosen and decoding is successful. It follows that our algorithm decodes all error patterns of weight up to half the minimum distance of the code.

Theorem 3.1. *Let C be the matrix-product code $[C_1 \cdots C_s] \cdot A$, where $C_1 \supset \cdots \supset C_s$ and A is a non-singular by columns matrix. Let $\mathbf{e} = (e_1, e_2, \ldots, e_l) \in \mathbb{F}_q^{ml}$ be an error vector with $wt(\mathbf{e}) \leq \lfloor (d(C)-1)/2 \rfloor$. Then there exists an ordered subset $\{i_1, \ldots, i_s\} \subset \{1, \ldots, l\}$ satisfying $wt(e_{i_j}) \leq t_j = \lfloor (d_j - 1)/2 \rfloor$, for all $j \in \{1, \ldots, s\}$.*

Proof. We claim that there exists i_1 such that $wt(e_{i_1}) \leq t_1$. Suppose that there is no $i_1 \in \{1, \ldots, l\}$ with $wt(e_{i_1}) \leq t_1$, that is, $wt(e_i) > \lfloor (d_1 - 1)/2 \rfloor$, for all $i = 1, \ldots, l$. This implies that $wt(e_i) \geq d_1/2$, for all i. Using (8) we obtain,

$$wt(\mathbf{e}) \geq \frac{ld_1}{2} > \frac{ld_1 - 1}{2} \geq \left\lfloor \frac{ld_1 - 1}{2} \right\rfloor \geq \left\lfloor \frac{d(C) - 1}{2} \right\rfloor$$

which contradicts our assumption.

Let us assume that the property holds for a subset $\{i_1, \ldots, i_{j-1}\} \subset \{1, \ldots, l\}$ of size $j - 1 < s$. We now prove it holds for a subset of size j. Suppose that there is no i_j with $wt(e_{i_j}) \leq t_j$, that is, $wt(e_i) > \lfloor (d_j - 1)/2 \rfloor$, for all $i \in \{1, \ldots, l\} \setminus \{i_1, \ldots, i_{j-1}\}$. This implies that $wt(e_i) \geq d_j/2$, for all $i \in \{1, \ldots, l\} \setminus \{i_1, \ldots, i_{j-1}\}$. Using (8) we obtain,

$$\begin{aligned} wt(\mathbf{e}) \geq \sum_{k=1}^{j-1} wt(e_{i_k}) + \frac{(l-j+1)d_j}{2} &> \sum_{k=1}^{j-1} wt(e_{i_k}) + \frac{(l-j+1)d_j - 1}{2} \geq \\ &\geq \left\lfloor \frac{(l-j+1)d_j - 1}{2} \right\rfloor \geq \left\lfloor \frac{d(C) - 1}{2} \right\rfloor \end{aligned}$$

which contradicts our assumption and the result holds. □ □

Summarizing, we can now formulate our decoding algorithm for $C = [C_1 \cdots C_s] \cdot A \subset \mathbb{F}_q^{ml}$, where $C_1 \supset \cdots \supset C_s$ and A is a non-singular by columns matrix, in procedural form in Algorithm 1.

Remark 3.2. We note that the algorithm becomes more computationally intensive as the number of blocks s that we may need to decode at each iteration, and the total number of blocks l, increase. So, it is worthwhile to say that for $s = l = 2$ (respectively 3) the algorithm only needs, at most, 2 (respectively 6) iterations to find the right ordered subset of indices. In this case we need to decode at most 4 (respectively 18) blocks of length m to achieve a successful decoding of the full-length received word of length ml.

An advantage of using nested codes in our construction, from the point of view of implementation in an electronic circuit, is that much of the circuitry

Algorithm 1 DECODING ALGORITHM FOR $C = [C_1 \cdots C_s] \cdot A$

Input: Received word $\mathbf{p} = \mathbf{c} + \mathbf{e}$ with $\mathbf{c} \in C$ and $wt(\mathbf{e}) \leq \lfloor (d(C) - 1)/2 \rfloor$. $C_1 \supset \cdots \supset C_s$ nested codes and A a non-singular by columns matrix. Decoder DC_i for code C_i, $i = 1, \ldots, s$.

Output: The codeword $\mathbf{c}$.

1: $\mathbf{p}' = \mathbf{p}$; $A' = A$;
2: **for** $\{i_1, \ldots, i_s\} \subset \{1, \ldots, l\}$ **do**
3: $\quad \mathbf{p} = \mathbf{p}'$; $A = A'$;
4: $\quad$ **for** $j = 1, \ldots, s$ **do**
5: $\quad\quad p_{i_j} = DC_j(p_{i_j})$;
6: $\quad\quad$ **if** p_{i_j} = *"failure"* **then**
7: $\quad\quad\quad$ *Break* the loop and consider another $i_1, \ldots, i_s$ in line 1;
8: $\quad\quad$ **end if**
9: $\quad\quad$ **for** $k = j + 1, \ldots, s$ **do**
10: $\quad\quad\quad p_{i_k} = p_{i_k} - \frac{a_{j,i_k}}{a_{j,i_j}} p_{i_j}$;
11: $\quad\quad\quad \text{column}_{i_k}(A) = \text{column}_{i_k}(A) - \frac{a_{j,i_k}}{a_{j,i_j}} \text{column}_{i_j}(A)$;
12: $\quad\quad$ **end for**
13: $\quad$ **end for**
14: $\quad$ Obtain $(c_1, \ldots, c_s)$ from $p_{i_1}, \ldots, p_{i_s}$;
15: $\quad \mathbf{p} = [C_1 \cdots C_s] \cdot A$; (see (1) and (2))
16: $\quad$ **if** $\mathbf{p} \in C$ and $wt(\mathbf{p} - \mathbf{p}') \leq \lfloor (d(C) - 1)/2 \rfloor$ **then**
17: $\quad\quad$ RETURN: $\mathbf{p}$;
18: $\quad$ **end if**
19: **end for**

used for implementing decoder DC_1 can also be used to implement decoders $DC_2, \ldots, DC_s$ of the subset subcodes $C_2, \ldots, C_s$, respectively.

Generalized Reed-Muller codes are iterative matrix-product codes defined using a non-singular by columns matrix (see [2]). Therefore, Algorithm 1 can be used to decode this family of codes. Furthermore, Algorithm 1 can be considered as a generalization of the decoding algorithm for Reed-Muller codes in [6, Chapter 13].

4 A class of Quasi-Cyclic codes from nested cyclic codes

Finally we consider the matrix-product code, $C = [C_1 \cdots C_s] \cdot A$, where the component codes $C_1, \ldots, C_s$ are cyclic codes of the same length. In this case, a generator matrix of C can be constructed as an $s \times l$ array of truncated circulant submatrices, and defines a quasi-cyclic code with s generators and index l, [4, 5].

Determining the minimum distance of an arbitrary quasi-cyclic code is not easy, and a good general decoding algorithm has not yet been developed for this class of codes. Here we note that when the cyclic codes $C_1, \ldots, C_s$ are nested, and A is a non-singular by columns matrix, then Theorem 2.2 provides the minimum distance, and Algorithm 1 provides a decoding algorithm for a restricted class of quasi-cyclic codes, obtained by matrix-product construction.

We now provide an example of how Algorithm 1 proceeds in this case.

Example 4.1. Consider the following linear codes over $\mathbb{F}_3$,

- C_1 the $[13, 10, 3]$ cyclic code generated by $f_1 = x^3 + x^2 + x + 2$.
- C_2 the $[13, 7, 5]$ cyclic code generated by $f_2 = (x^3 + x^2 + x + 2)(x^3 + 2x^2 + 2x + 2)$.
- C_3 the $[13, 3, 9]$ cyclic code generated by $f_3 = (x + 2)(x^3 + x + 2)(x^3 + x^2 + x + 2)(x^3 + 2x^2 + 2x + 2)$.

Let $C = [C_1 C_2 C_3] \cdot A$, where A is the non-singular by columns matrix

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 2 & 1 \\ 0 & 0 & 1 \end{pmatrix}.$$

We use decoder DC_i for C_i, which decodes up to half the minimum distance, i.e., DC_1, DC_2, DC_3 decode up to $t_1 = 1$, $t_2 = 2$ and $t_3 = 4$ errors, respectively.

One may easily check that $D_1 = 3$, $D_2 = 2$ and $D_3 = 1$. Therefore, $d(C) = \min\{d_1 D_1, d_2 D_2, d_3 D_3\} = 9$ by Theorem 2.2, thus we may correct up to $t = 4$ errors in a codeword of C.

We consider polynomial notation for codewords of C_i, for all i, that is the codewords of length 13 in C_i are polynomials in $\mathbb{F}_q[x]/(x^{13} - 1)$ and words in C are elements in $(\mathbb{F}_q[x]/(x^{13} - 1))^3$. Let $\mathbf{p} = \mathbf{c} + \mathbf{e}$ be the received word, with codeword $\mathbf{c} = (0, 0, 0)$ and the error vector of weight $t = 4$

$$\mathbf{e} = (e_1, e_2, e_3) = (1 + x, 2x^2, 2x^{11}).$$

- Consider, $i_1 = 1, i_2 = 2, i_3 = 3$.

  We decode the first block $p_1 = 1 + x$ of $\mathbf{p}$, using DC_1. There is only one word, $1 + x + x^4$, in C_1 within distance one to p_1, so DC_1 decodes p_1 to $p_1^{2)} = 1 + x + x^4$. Notice that it is incorrectly decoded, but we cannot detect it at this moment. Proceeding, we compute

  $$p_2^{2)} = p_2 - p_1^{2)} = 2 + 2x + 2x^2 + 2x^4,$$
  $$p_3^{2)} = p_3 - p_1^{2)} = 2 + 2x + 2x^4 + 2x^{11}.$$

  The new matrix of coefficient is

  $$A^{2)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 1 \\ 0 & 0 & 1 \end{pmatrix}.$$

  Then, we decode $p_2^{2)}$ using DC_2. There is only one word, $2 + 2x + 2x^2 + x^4 + x^{11}$, in C_2 within distance two to $p_2^{2)}$. Thus, DC_2 decodes $p_2^{2)}$ to $p_2^{3)} = 2 + 2x + 2x^2 + x^4 + x^{11}$. We compute

  $$p_3^{3)} = p_3^{2)} - 2p_2^{3)} = 1 + x + 2x^2.$$

Finally, we decode $p_3^{3)}$ using DC_3. Since there is only one word, 0, in C_3 within distance four to $p_3^{3)}$, DC_3 decodes $p_3^{3)}$ to $p_3^{4)} = 0$.

Notice that the distance of the decoded word $(p_1^{2)}, p_2^{3)}, p_3^{4)})$ to $\mathbf{p}$ is $6 > t = 4$, thus we have corrected more errors than the error correction capability of the code which implies that we should consider another set of indices.

- Second attempt, consider $i_1 = 2, i_2 = 1, i_3 = 3$.

  We decode the second block $p_2 = 1 + x$ of $\mathbf{p}$, using DC_1. There is only one word, 0, in C_1 within distance one to p_1. Thus, DC_1 decodes p_2 to $p_2^{2)} = 0$. Proceeding, we compute

$$p_1^{2)} = p_1 - p_2^{2)} = p_1 = 1 + x,$$
$$p_3^{2)} = p_3 - p_2^{2)} = p_3 = 2x^{11}.$$

  The new matrix of coefficient is

$$A^{2)} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 2 & 2 \\ 0 & 0 & 1 \end{pmatrix}.$$

  Then, we decode $p_1^{2)}$ using DC_2. There is only one word, 0, in C_2 within distance two to $p_1^{2)}$. Thus, DC_2 decodes $p_1^{2)}$ to $p_1^{3)} = 0$. We compute

$$p_3^{3)} = p_3^{2)} - p_1^{3)} = p_3 = 2x^{11}.$$

  We decode $p_3^{3)}$ using DC_3. Since there is only one word, 0, in C_3 within distance four to $p_3^{3)}$, DC_3 decodes $p_3^{3)}$ to $p_3^{4)} = 0$. And since the weight of $\mathbf{p} - \mathbf{p}^{3)}$ is equal to 4, we have decoded the received word successfully.

Acknowledgements

The authors would like to thank M. Greferath for his course at Claude Shannon Institute and P. Beelen and T. Høholdt for helpful comments on this paper.

References

[1] van Asch, B.; "Matrix-product codes over finite chain rings"; *Appl. Algebra Engrg. Comm. Comput.* 19 (2008), no. 1, 39–49

[2] Blackmore, T.; Norton, G.H.; "Matrix-product codes over $\mathbb{F}_q$"; *Appl. Algebra Engrg. Comm. Comput.* 12 (2001), no. 6, 477–500.

[3] Kasami, T.; "A Gilbert-Varshamov bound for quasi-cyclic codes of rate 1/2", *IEEE Trans. Inform. Theory*, 20, pp. 679, 1974.

[4] Lally, K.; Fitzpatrick, P.; "Algebraic Structure of quasicyclic codes"; *Discrete Appl. Math.,* 111, no. 1-2, pp. 157–175, 2001.

[5] Ling, S.; Solé. P.; "On the algebraic structure of quasi-cyclic codes I: finite fields", *IEEE Trans. Inform. Theory*, vol. IT-47, 2751–2759, Nov. 2001.

[6] Macwilliams, F.J.; Sloane N.J.A.; *The Theory of Error-Correcting Codes*, ser. North-Holland mathematical library. North-Holland, 1977, vol. 16.

[7] Martínez-Moro, E.; "A generalization of Niederreiter-Xing's propagation rule and its commutativity with duality"; *IEEE Trans. Inform. Theory* 50 (2004), no. 4, 701–702.

[8] Niederreiter, H.; Xing, C.; "A propagation rule for linear codes"; *Appl. Algebra Engrg. Comm. Comput.* 10 (2000), no. 6, 425–432.

[9] Özbudak, F.; Stichtenoth, H.; "Note on Niederreiter-Xing's propagation rule for linear codes"; *Appl. Algebra Engrg. Comm. Comput.* 13 (2002), no. 1, 53–56.