Error-locating pairs for cyclic codes

Iwan M. Duursma * and Ralf Kötter †

March 8, 1993

Abstract

A general decoding method for linear codes is investigated for cyclic codes. The decoding consists of solving two systems of linear equations. All but four binary cyclic codes of length less than 63 can so be decoded up to their actual distance. A new family of codes is given for which the decoding needs only $O(n^2)$ operations.

Index Terms — Algebraic decoding, cyclic code, error-locating pair, MDS-code, QR-code.

*Eindhoven University of Technology, Department of Mathematics and Computing Science, P.O. Box 513, 5600 MB Eindhoven, The Netherlands. Supported by NWO, through Stichting Mathematisch Centrum.

†Linköping University, Department of Electrical Engineering, S-581 83 Linköping, Sweden.

I Introduction

The most successful methods for decoding of linear codes separate the decoding into the location of the error positions and the determination of the error values. Particular examples are the decoding of cyclic codes up to the BCH-bound and the basic algorithm for the decoding of algebraic-geometric codes. The methods allow a unified description that applies to any linear code. This was noticed by Pellikaan [15], who used it to describe the decoding of AG-codes. Independently but later, Kötter [9] gave a similar description. The location of the error positions is done with the help of an *error-locating pair* of vector spaces. To decode a particular linear code one has to assign such a pair to the code. We investigate how this can be done for cyclic codes. For a given error-locating pair, the decoding itself can be performed by solving two systems of linear equations.

We first recall the unified description. It applies to any linear code. Thus, it is presented with a minimum of assumptions and notation and the proofs can remain short. Section III investigates the determination of the error values. The later sections restrict to cyclic codes. In Sections IV and V we suggest general formats for error-locating pairs. Section VI treats complexity aspects of the decoding procedure. Section VII shows how in certain cases it is possible to extend the error-correction capability of error-correcting pairs. Section VIII gives pairs to decode all but four binary cyclic codes of length less than 63 up to their actual distance. In the last section, some sequences of codes are given with corresponding error-locating pairs.

II Error-locating pairs

A An error-locating procedure

The n-tuples defined over a field $\mathbb{F}$ form a vector space denoted by $\mathbb{F}^n$. For two vectors $\mathbf{u} = (u_0, u_1, \ldots, u_{n-1})$ and $\mathbf{v} = (v_0, v_1, \ldots, v_{n-1})$, we define a product $\mathbf{u} * \mathbf{v} = (u_0 v_0, u_1 v_1, \ldots, u_{n-1} v_{n-1})$. For two subspaces $U, V \subset \mathbb{F}^n$, let $U * V$ denote the set of vectors $\{\mathbf{u} * \mathbf{v} : \mathbf{u} \in U, \mathbf{v} \in V\}$. For a linear code C we denote the dimension by $k(C)$ and the Hamming distance by $d(C)$, or by k and d respectively when no confusion arises.

Definition 1 (t-error-locating pair) Let U, V and C be linear codes of length n over the field $\mathbb{F}$. We call (U, V) a t-error-locating pair for C if the following conditions hold

$$U * V \subseteq C^{\perp}, \tag{1}$$
$$k(U) > t, \tag{2}$$
$$d(V^{\perp}) > t. \tag{3}$$

Using this definition we will derive a t-error-locating procedure based on the following central observation.

Theorem 1 *Let (U, V) be a t-error-locating pair for the code C. Let $\mathbf{y} = \mathbf{c} + \mathbf{e}$ be a word in $\mathbb{F}^n$ with $\mathbf{c} \in C$ and $\mathbf{e}$ a vector of weight at most t. There exists a non-zero vector $\mathbf{u} \in U$ such that*

$$\sum_{i=0}^{n-1} y_i u_i v_i \;=\; 0, \qquad \textit{for all } \mathbf{v} \in V. \tag{4}$$

Moreover, any solution $\mathbf{u} \in U$ of (4) satisfies

$$\mathbf{e} * \mathbf{u} = \mathbf{0} \tag{5}$$

Proof. In (4) we may replace $\mathbf{y}$ by $\mathbf{e}$ by condition (1). Thus any vector $\mathbf{u} \in U$ with property (5) is a solution to (4). Condition (2) guarantees the existence of a non-zero vector. This is because we impose at most t linear conditions on U. To prove (5), we note that (4) has the equivalent formulation

$$\mathbf{y} * \mathbf{u} \in V^{\perp}.$$

Again replacing $\mathbf{y}$ by $\mathbf{e}$ and using $\text{weight}(\mathbf{e}) \leq t$ and condition (3) we find (5). □

Assume we are given an error-locating pair (U, V) and a received word $\mathbf{y}$. We have to find a solution $\mathbf{u} \in U$ to the homogeneous system of linear equations (4), which then by property (5) locates possible error positions with zeros. We will give the matrix defining this system. Let $\text{diag}(\mathbf{y})$ denote the $n \times n$ matrix which has the elements of $\mathbf{y}$ on its main diagonal and which is zero everywhere else. Equation (4) can thus be written as:

$$\mathbf{v} \cdot \text{diag}(\mathbf{y}) \cdot \mathbf{u}^T = 0, \quad \textit{for all } \mathbf{v} \in V.$$

Obviously, it is enough to consider a set of basis vectors in V, forming a generator matrix G_V for V and we obtain

$$G_V \cdot \text{diag}(\mathbf{y}) \cdot \mathbf{u}^T = \mathbf{0}.$$

To make this equation solvable with methods of linear algebra we replace $\mathbf{u}$ by $\mathbf{u} = \sigma G_U$, where G_U is a generator matrix for U and σ is an element of $\mathbb{F}^{k(U)}$. Thus the key equation (4) can be rephrased as

$$S(\mathbf{y}) \cdot \sigma^T = \mathbf{0}, \tag{6}$$

where

$$S(\mathbf{y}) = G_V \cdot \text{diag}(\mathbf{y}) \cdot G_U^T.$$

Any solution σ for (6) gives a solution $\mathbf{u} = \sigma G_U$ for (4) which now locates possible error positions with zeros. Thus we have described a t-error-locating procedure provided we have a t-error-locating pair. The problem of error-location is now to find the spaces U and V that satisfy conditions (1)-(3) for a maximal value of t.

Remark 1 We are completely free in choosing bases for U and V, i.e. in choosing the matrices G_U and G_V, without affecting the space of solutions to the key equation. Nevertheless the choice of G_U and G_V determines the structure of the matrix $S(\mathbf{y})$. We will point out how this affects the computational complexity of solving (6) at a later stage.

Remark 2 Given a particular error vector $\mathbf{e}$, it is clear from the proof of Theorem 1 that the following conditions are sufficient to obtain $\mathbf{u} \in U \backslash \mathbf{0}$ with property (5):

$$C * U \subseteq V^{\perp}, \tag{7}$$

$$\exists \mathbf{u} \in U \backslash \mathbf{0} \ : \ \mathbf{e} * \mathbf{u} = \mathbf{0}, \tag{8}$$

$$\forall \mathbf{u} \in U \backslash \mathbf{0} \ : \ \mathbf{e} * \mathbf{u} \in V^{\perp} \ \Rightarrow \ \mathbf{e} * \mathbf{u} = \mathbf{0}. \tag{9}$$

The first condition is equivalent to (1). Conditions (8) and (9) are weaker than conditions (2) and (3) respectively. We will have to refer to them in some cases where the conditions in Definition 1 are too strong.

B Error-locating functions

Theorem 1 in the previous subsection gives a possibility to determine the error positions as zeros of a word $\mathbf{u} \in U$. This describes the general case. In some known algorithms, in particular for BCH-codes and AG-codes, an error-locating word $\mathbf{u}$ is associated in a natural way with an error-locating function. We will need this connection to make some properties of $\mathbf{u}$ and the corresponding error-locating function more transparent. Also the relation with functions is helpful in actually finding pairs (U, V). The rest of the section is devoted to this relation.
We have derived two sets of sufficient conditions for an error-locating pair. A pair with (1)-(3) locates all error patterns of a given weight. Such a pair is hard to find in general. Conditions (7)-(9) are weaker. They are formulated for a particular error pattern however and the verification for a large class of error patterns becomes cumbersome. We formulate a set of conditions that can be seen as a compromise. The conditions depend on the positions of the errors but not on the particular error values.

Lemma 1 *For an error vector* $\mathbf{e}$*, let* E $(\overline{E})$ *be the subspace of* $\mathbb{F}^n$ *consisting of all vectors that have zero components in the error (non-error) positions. The following conditions are sufficient to locate the error positions with the pair* (U, V)*:*

$$C * U \subseteq V^{\perp}$$
$$U \cap E \neq \mathbf{0}$$
$$V^{\perp} \cap \overline{E} = \mathbf{0}.$$

Proof. The conditions imply (7)-(9). □

The conditions of the lemma can be expressed in terms of functions. We need the following.

Notation 1 For a field $\mathbb{F}$ let S be the the $\mathbb{F}$-algebra of n-tuples defined over $\mathbb{F}$ with component-wise multiplication and addition. Let R be a $\mathbb{F}$-algebra with no zero-divisors such that there exists a surjective homomorphism $Ev : R \longrightarrow S$, with kernel I.
For a code $C \subset S$, let $L(C) \subset R$ denote a $\mathbb{F}$-vector space such that the restriction of Ev to $L(C)$ is a $\mathbb{F}$-vector space isomorphism from $L(C)$ to C. In particular $L(C) \cap I = (0)$.

Remark 3 R will be identified with a ring of functions. Ev is then the evaluation mapping, that means the evaluation of $f \in R$ in a set of points. Ev induces naturally an $\mathbb{F}$-algebra isomorphism between R/I and S.

Example 1 For cyclic codes we take $R = \mathbb{F}[x]$. Let $\alpha \in \mathbb{F}$ be a primitive n-th root of unity. Ev is the evaluation map that evaluates polynomials in points $\{1, \alpha, \alpha^2, \ldots, \alpha^{n-1}\}$, i.e.

$$Ev(x) = (1, \alpha, \alpha^2, \ldots, \alpha^{n-1}).$$

The ideal $I \subset R$ is generated by $x^n - 1$. We obtain the algebra isomorphism $Ev : \mathbb{F}[x]/(x^n - 1) \xrightarrow{\sim} S$. For cyclic codes this differs from the well-known vector space isomorphism $Id : \mathbb{F}[x]/(x^n - 1) \xrightarrow{\sim} S$, that identifies ideals in $\mathbb{F}[x]/(x^n - 1)$ with cyclic codes in S. For example:

$$\begin{aligned} Ev(1 + x) &= (1 + 1, 1 + \alpha, 1 + \alpha^2, \ldots, 1 + \alpha^{n-1}), \\ Id(1 + x) &= (1, 1, 0, 0, \ldots, 0, 0). \end{aligned}$$

Example 2 For AG-codes an evaluation map Ev occurs in their definition [7].

Now, let an algebra-homomorphism $Ev : R \longrightarrow S$ be given as in Notation 1. The reformulation of Lemma 1 becomes

Lemma 2 *Let $L(U), L(V^\perp)$ and $L(C)$ map to the codes $U, V^\perp$ and C after evaluation. Let the maps be bijective as in Notation 1. For an error vector $\mathbf{e}$, let J ($\overline{J}$) be the ideal in R consisting of all elements that evaluate to zero at the error (non-error) positions. The following conditions are sufficient to locate the error positions with the pair (U, V):*

$$\begin{aligned} &L(C) * L(U) \subseteq L(V^\perp) + I, \\ &L(U) \cap J \neq (0), \\ &L(V^\perp) \cap \overline{J} = (0). \end{aligned}$$

Proof. Immediate from Lemma 1. □

The question arises whether an error-locating procedure can be formulated in terms of error-locating functions. This is indeed the case. The decoding procedures for BCH-codes [2, p.248] or AG-codes [8, 19] use this approach. $L(C), L(U)$ and $L(V^\perp)$ have here a natural interpretation.

III Error-correcting pairs

The previous section shows how an error-locating pair (U, V) can be used to locate the error positions in a received word. This is the most important part of the decoding. Therefore error-locating pairs will play a major role in what follows. The key idea is that for a received word $\mathbf{y}$, a vector $\mathbf{u}$ can be obtained such that $\mathbf{u}$ has zeros at the error positions.

Remark 4 The error vector $\mathbf{e}$ satisfies the conditions

$$\mathbf{y} - \mathbf{e} \in C,$$
$$\mathbf{e} * \mathbf{u} = \mathbf{0}.$$

Thus $\mathbf{e}$ can be obtained by solving a system of linear equations.

In general the vector $\mathbf{u}$ may have zeros at other positions too. For the determination of the error values it is important that the set of zeros is not too large.

Lemma 3 *For a code C with error-locating pair (U, V), let $\mathbf{u} \in U \backslash \mathbf{0}$ locate the error positions of the error vector $\mathbf{e}$, that is $\mathbf{e} * \mathbf{u} = \mathbf{0}$. The error values are uniquely determined by $\mathbf{u}$ if and only if*

$$\forall \mathbf{c} \in C \ : \quad \mathbf{c} * \mathbf{u} = \mathbf{0} \ \Rightarrow \ \mathbf{c} = \mathbf{0}. \tag{10}$$

Proof. Assume we can write $\mathbf{y}$ in two different ways as $\mathbf{y} = \mathbf{e}_1 + \mathbf{c}_1 = \mathbf{e}_2 + \mathbf{c}_2$, where $\mathbf{c}_1, \mathbf{c}_2 \in C$ and $\mathbf{e}_1 * \mathbf{u} = \mathbf{e}_2 * \mathbf{u} = \mathbf{0}$. It follows that

$$(\mathbf{e}_1 - \mathbf{e}_2) * \mathbf{u} = \mathbf{0}, \qquad \text{with} \quad \mathbf{e}_1 - \mathbf{e}_2 = \mathbf{c}_2 - \mathbf{c}_1 \in C.$$

Condition (10) implies $\mathbf{e}_1 = \mathbf{e}_2$. If this condition fails, say $\mathbf{c} * \mathbf{u} = \mathbf{0}$ for $\mathbf{c} \neq \mathbf{0}$, we find the two different solutions $\mathbf{e}, \mathbf{e} - \mathbf{c}$. □

We follow the definition of a t-error-correcting pair in [15]. See also [9].

Definition 2 (t-error-*correcting* pair) Let (U, V) be a t-error-locating pair for the code C as in Definition 1. We call (U, V) a t-error-correcting pair for the code C if in addition to the conditions (1),(2) and (3) the following is satisfied

$$d(C) + d(U) > n, \tag{11}$$

where n denotes the code length of C.

Remark 5 The definition is justified by the lemma since condition (11) implies

$$\forall \mathbf{c} \in C, \forall \mathbf{u} \in U \ : \quad \mathbf{c} * \mathbf{u} = \mathbf{0} \ \Rightarrow \ \mathbf{c} = \mathbf{0} \ \vee \ \mathbf{u} = \mathbf{0}. \tag{12}$$

In some cases we will prefer to use the weaker condition (12).

Remark 6 Recall that a pair (U, V) needs to satisfy $C * U \subseteq V^{\perp}$ to be error-locating for a code C. By the lemma, an error-locating pair will be error-correcting if it satisfies

$$C^* * U^* \subseteq (V^{\perp})^*.$$

Remark 7 In terms of functions a pair (U, V) needs to satisfy $L(C)*L(U) \subseteq L(V^{\perp})+I$ to be error-locating. By the lemma it will be error-correcting if it satisfies

$$L(C) * L(U) \subseteq L(V^{\perp}).$$

Here we use the fact that R has no zero-divisors and that $L(V^{\perp}) \cap I = (0)$. Let $\langle L(C)*L(U)\rangle$ denote the linear space spanned by all functions in $L(C) * L(U)$. The following conditions are then sufficient to guarantee error-correction with a pair (U, V):

$$L(U) \cap J \neq (0), \tag{13}$$
$$\langle L(C) * L(U)\rangle \cap \overline{J} = (0). \tag{14}$$

The dilemma in algebraic decoding is obvious. For (13) we want $L(U)$ to be big and for (14) we want $\langle L(C) * L(U)\rangle$ that means $L(U)$ to be small.

IV Cyclic Codes

A Notation

From now on we restrict our attention to the class of cyclic codes. A cyclic code $C \subset I\!F^n$ is usually identified with an ideal in the ring $I\!F[x]/(x^n - 1)$ generated by a polynomial $g(x)$, which divides $x^n - 1$. A codeword $\mathbf{c} = (c_0, c_1, \ldots, c_{n-1}) \in C$ is interpreted as a polynomial by the relation

$$c(x) = c_0 + c_1 x^1 + \ldots + c_{n-1} x^{n-1}, \qquad \text{with } g(x) | c(x).$$

The code is determined by the zeros of $g(x)$. We assume $(\mathrm{char}\, I\!F, n) = 1$, so that $x^n - 1$ has n different zeros. Let the extension $\overline{I\!F}$ of $I\!F$ contain the n-th roots of unity and let $\alpha \in \overline{I\!F}$ be a primitive n-th root of unity. Let $m_i(x)$ be the minimal polynomial of α^i over $I\!F$. If $g(x)$ equals $\mathrm{lcm}\{m_i(x) : \alpha^i \in R\}$ then we call R a *defining set* for C. If R is the maximal defining set for C we call R complete. By abuse of standard notation we will describe the defining set by the exponents occurring in R. With $R = \{i_1, i_2, \ldots, i_l\}$ the matrix

$$M(R) = \begin{pmatrix} (\alpha^{i_1})^0 & (\alpha^{i_1})^1 & \ldots & (\alpha^{i_1})^{n-1} \\ (\alpha^{i_2})^0 & (\alpha^{i_2})^1 & \ldots & (\alpha^{i_2})^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ (\alpha^{i_l})^0 & (\alpha^{i_l})^1 & \ldots & (\alpha^{i_l})^{n-1} \end{pmatrix}$$

is a parity-check matrix for a code $\overline{C} \subset \overline{I\!F}^n$. The code C is obtained as the subfield subcode of $\overline{C}$, that means $C = \overline{C} \cap I\!F^n$.

Definition 3 Let R be a *defining set* of a cyclic code $C/I\!F$. C is then defined as

$$C = \{\mathbf{c} \in I\!F^n : M(R)\mathbf{c}^T = \mathbf{0}\}.$$

We also like to refer to a matrix $M(R)$ as a generator matrix to describe the codes U and V in an error-locating pair (U, V). To distinguish between the use of $M(R)$ as a parity-check matrix or as a generator matrix, we call R a defining set in the former case and a *generating set* in the latter case. As it will be seen the notation of generating sets is very convenient to describe error-locating pairs for cyclic codes.

Definition 4 Let I be a *generating set* of a cyclic code $U/\overline{\mathbb{F}}$. U is then defined as

$$U = \{\mathbf{u} \in \overline{\mathbb{F}}^n : \mathbf{u} = \sigma M(I), \sigma \in \overline{\mathbb{F}}^{|I|}\}.$$

In the following, generating sets for the codes U and V will be denoted by I and J respectively. We stress that both codes are defined over the large field $\overline{\mathbb{F}}$. Thus, their dimensions follow immediately as $k(U) = |I|$ and $k(V) = |J|$. Let $I = \{i_1, \ldots, i_l\}$, where $i_1 < \ldots < i_l$. We define

$$\overline{I} = \{i_1, i_1 + 1, \ldots, i_l - 1, i_l\}.$$

The following reformulation of the BCH-bound is obvious.

Lemma 4 *The minimum distance of a cyclic code of length n with generating set I is bounded below by*

$$d \geq n - |\overline{I}| + 1.$$

We will freely use the following observations. Let

$$\mathbf{a}(i) = (1, \alpha^i, \alpha^{2i}, \ldots, \alpha^{(n-1)i}).$$

We have

$$\mathbf{a}(i) \perp \mathbf{a}(j) \quad \Leftrightarrow \quad i + j \not\equiv 0 \pmod{n}.$$

Let $b + cR = \{b + c\,i \pmod{n} : i \in R\}$. The codes with defining sets R and $b + cR$ are equivalent, for $(c, n) = 1$. Also let $I + J = \{i + j \pmod{n} : i \in I, j \in J\}$. Let U, V and W be cyclic codes with generating sets I, J and $I + J$ respectively. Then $U * V$ is a subset of W.

B Decoding BCH-codes

Theorem 1 provides a way to construct a decoding algorithm for a particular linear code. The bottle-neck in the construction is the search for an error-correcting pair (U, V). As we shall see, there exists an obvious choice which enables us to decode up to the BCH-bound. We can do better however by using a correspondence between the pair of codes (U, V) and the pair of defining sets (A, B) in the lemma below.

Lemma 5 (Roos-bound) *(Theorem 3 [10])* *If A is a defining set for a cyclic code with minimum distance d_A and if the set B is such that $|\overline{B}| \leq |B| + d_A - 2$, then the code with defining set $A + B$ has minimum distance $d \geq |B| + d_A - 1$.*

Proof. After replacing A and B by sets of zeros see [10]. □

Corollary 1 *Let c_1 and c_2 satisfy $(c_1, n) = (c_2, n) = 1$. In the lemma, the same bound on the distance holds for a code with defining set $c_1A + c_2B$.*

Proof. First it is immediate from the proof in [10] that the constants play no essential role and can be taken equal to one. Also, we may restrict to the case $c_2 = 1$ by passing to an equivalent code. The lemma can now be applied with the sets c_1A and B. □

Theorem 2 *Let $s < t$. Let the generating sets I, J and K satisfy*

$$\begin{aligned} |I| &= t+1, \\ |J| &= t-s, \qquad |\overline{J}| = t-s, \\ |K| &= s+1, \qquad |\overline{K}| \leq t. \end{aligned}$$

Let $(c_1, n) = (c_2, n) = (c_3, n) = 1$. Then the code $C/\mathbb{F}$ with defining set $R = b + c_1I + c_2J + c_3K$ has a t-error-locating pair (U, V), where $U/\overline{\mathbb{F}}$ is defined by the generating set $b + c_1I$ and $V/\overline{\mathbb{F}}$ by the generating set $c_2J + c_3K$. For the distance of the code C we have

$$|\overline{I}| \leq 2t \;\Rightarrow\; d(C) \geq 2t+1.$$

The pair (U, V) is t-error-correcting whenever

$$|\overline{I}| \leq d(C).$$

Proof. The verification of conditions (1) and (2) is straightforward. The distance $d(V^\perp)$ can be estimated with the lemma. We use it with

$$\begin{aligned} &A = J, \quad d_A = t-s+1 \quad \text{and} \\ &B = K, \quad |\overline{B}| \leq (s+1) + (t-s+1) - 2, \end{aligned}$$

and apply the corollary. It follows that $d(V^\perp) \geq (s+1) + (t-s+1) - 1 = t+1$ and (3) holds. The distance $d(C)$ follows with another application of the lemma, this time with

$$\begin{aligned} &A = c_2J + c_3K, \quad d_A \geq t+1 \quad \text{and} \\ &B = bc_1^{-1} + I, \quad |\overline{B}| \leq (t+1) + (t+1) - 2. \end{aligned}$$

Using the corollary, $d(C) \geq (t+1)+(t+1)-1 = 2t+1$. For the last statement, combination of $d(U) \geq n - |\overline{I}| + 1$ (Lemma 4) and $d(C) > |\overline{I}| - 1$ yields condition (11). □

Example 3 *(Example 1 [6])* Let C be the binary cyclic code of type $[39, 15]$ defined by $R \supset \{1, 3\}$. In particular

$$R \supset \{1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12\}.$$

The BCH-bound yields $d \geq 7$ (the Hartmann-Tzeng bound or the Roos bound do not improve on this). The actual distance equals 10 ([17],[10]). In the theorem we may choose $I = \{1, 2, 3, 8, 9\}, J = \{0, 1, 2, 3\}, K = \{0\}$ with $t = 4$ and $s = 0$. Since $|\overline{I}| > 2t$, the theorem does not yield a better estimate for the distance $d(C)$. Using the knowledge that the distance is equal to 10, it yields that the 4-error-locating pair (U, V) is actually 4-error-correcting. The code C corresponds to entry 45 in Table 1.

A procedure to decode up to the Hartmann-Tzeng bound and in some cases up to the Roos bound is presented in [5]. We recall the two bounds, following [5], and show that the procedure is a special case of Theorem 2. Let the defining set for a cyclic code C contain $b + c_1 I^* + c_2 J^*$, with $I^* = \{1, 2, \ldots, d_0 - 1\}$ and $J^* = \{j_1, j_2, \ldots, j_{s+1}\}$, for $j_1 < j_2 < \ldots < j_{s+1}$ and $j_{s+1} - j_1 - s < d_0 - 1$. Also, let $(c_1, n) = (c_2, n) = 1$. Then

$$d(C) \geq d_{Roos} = d_0 + s.$$

The bound is a special case of Lemma 5. With the further restrictions

$$s + 1 \leq d_0 - 1, \tag{15}$$

$$j_{s+1} - j_1 < (d_0 + s - 1)/2. \tag{16}$$

the procedure in [5] decodes up to d_{Roos}. Note that the Hartmann-Tzeng bound corresponds to $j_h = h, h = 1, 2, \ldots, s + 1$ and in this case the restrictions can always be fulfilled.

Corollary 2 *(Theorem 4 and Theorem 5 [5]) In decoding up to the Roos-bound, we may assume that s is such that d_{Roos} is odd and we write $d_0 + s = 2t + 1$. Let the generating sets I, J, K be defined as*

$$I = \{0, 1, 2, \ldots, t\},$$
$$J = \{1, 2, \ldots, t - s\},$$
$$K = \{j_1, j_2, \ldots, j_{s+1}\}.$$

The code C with defining set $R = b + c_1 I + c_1 J + c_2 K$ has distance $d(C) \geq 2t + 1$. A t-error-correcting pair is given by (U, V), where U has generating set $b + c_1 I$ and V has generating set $c_1 J + c_2 K$.

Proof. The restriction (15) can be written as $s < t$. In particular the set J is well-defined. The restriction (16) yields $|\overline{K}| - 1 < t$. Thus all conditions of the theorem are fulfilled. □

C Recurrences

Some error-locating pairs in Theorem 2 do not satisfy the condition $|\overline{I}| \leq d(C)$. In that case an error-locating word $\mathbf{u} \in U$ is obtained but it is not immediately clear whether the error values are uniquely determined or not. To investigate this we use

Lemma 6 *Let the vector $\mathbf{e}$ have support in a set A and let R contain $|A|$ consecutive integers. Let the syndromes*

$$S_i = \langle \mathbf{e}, \mathbf{a}(i) \rangle, \quad i \in R, \tag{17}$$

be known. Then the set of equations (17) determines $\mathbf{e}$ uniquely.

Proof. By the BCH-bound the difference of two solutions for $\mathbf{e}$ has weight at least $|A| + 1$ or zero. Hence, two solutions with support in A are identical. An efficient way to solve for $\mathbf{e}$ is given by Forney's algorithm [17, p.297]. □

Recall from the computational scheme that in solving (4) for $\mathbf{u} \in U \setminus \mathbf{0}$ we actually find a vector σ solving the key equation (6). With $I = \{i_1, \ldots, i_l\}$ as generating set for U the generator matrix of U is given by $G_U = M(I)$. We are in the situation of Example 1 and for $\sigma = (\sigma_{i_1}, \ldots, \sigma_{i_l})$ we have $\mathbf{u} = \sigma G_U = Ev(\sigma)$ for the polynomial

$$\sigma = \sigma_{i_l} X^{i_l} + \ldots + \sigma_{i_2} X^{i_2} + \sigma_{i_1} X^{i_1}. \tag{18}$$

Having found an error-locating polynomial σ, the zeros of $\mathbf{u}$ are obtained as the zeros of σ by e.g. a Chien search. We denote this set with A. The following lemma provides a method to obtain a consecutive syndrome set of size $|A|$.

Lemma 7 *Let the polynomial σ (18) have the support of $\mathbf{e}$ among its zeros. Then*

$$\sigma_{i_l} S_{i_l+j} + \ldots + \sigma_{i_2} S_{i_2+j} + \sigma_{i_1} S_{i_1+j} = 0, \tag{19}$$

for all integers j.

Proof.

$$\begin{aligned} & \sigma_{i_l} S_{i_l+j} + \ldots + \sigma_{i_1} S_{i_1+j} \\ = \; & \langle\, \mathbf{e}\,,\, \sigma_{i_l}\mathbf{a}(i_l+j) + \ldots + \sigma_{i_1}\mathbf{a}(i_1+j)\,\rangle \\ = \; & \langle\, \mathbf{e}\,,\, (\sigma_{i_l}\mathbf{a}(i_l) + \ldots + \sigma_{i_1}\mathbf{a}(i_1)) * \mathbf{a}(j)\,\rangle \\ = \; & \langle\, \mathbf{e}\,,\, \mathbf{u} * \mathbf{a}(j)\,\rangle = \langle\, \mathbf{e} * \mathbf{u}\,,\, \mathbf{a}(j)\,\rangle = 0. \end{aligned}$$

□

Example 4 We consider the binary cyclic code C of type $[45, 15]$ with $R \supset \{1, 3, 7, 15\}$. It corresponds to entry 84 in Table 1. We have

$$R \supset \{1, 2, 3, 4, 11, 12, 13, 14, 15, 16, 17, 28, 29, 30, 31\}.$$

The BCH-bound gives $d \geq 8$, while the actual distance $d = 9$ and four errors can be corrected. The choice

$$I = \{1, 11, 12, 13, 14\}, \qquad J = \{0, 1, 2, 3\},$$

defines an error-locating pair (U, V) by Theorem 2. We can therefore compute

$$\sigma = \sigma_{14} X^{14} + \sigma_{13} X^{13} + \sigma_{12} X^{12} + \sigma_{11} X^{11} + \sigma_1 X,$$

such that the error positions are zeros of σ. We may assume that there are four errors and therefore that $\sigma_1 \neq 0$. Using (19) with $j = 17$ we find S_{18}. The syndrome S_5 is then obtained with $j = 4$. By then $S_1, S_2, \ldots, S_{20}$ are all known and Lemma 6 applies.

Remark 8 When using recurrences of type (19) we need to know which syndromes have a nonzero coefficient. In general there can be zero coefficients and these cases have to be treated separately. As in the example, one may be able to show that some coefficients cannot be zero. The procedure is described in [6], but there zero coefficients are not considered. Thus, the procedure as described in [6] may fail for the entries 17,84 and 121 in Table 1.

Example 5 We consider the code C of Example 4. The procedure in [6] corresponds to the choices

$$I = \{11, 12, 13, 14, 28\}, \qquad J = \{0, 1, 2, 3\}.$$

This defines an error-locating pair. In the case of four errors there will be a unique error-locating polynomial. The polynomial $X^{28} - X^{13}$ has fifteen zeros among the 45-th roots of unity. The zeros support a two-dimensional subcode of C and the error values are not uniquely determined from the error positions. In fact, the unknown syndromes $\{S_5, S_{10}, S_{20}, S_{40}, S_{35}, S_{25}\}$ cannot be obtained from the known syndromes with a recurrence $S_{15+j} = S_j$.

D Correcting more errors

Theorem 2 gives an error-correcting pair (U, V) to correct errors up to the BCH-bound and in some cases beyond. To achieve the error-correction capability of some cyclic codes we recall a well-known 'trick'. Considering binary cyclic codes, S_0 has value either 0 or 1.

Remark 9 Let the binary cyclic code C have distance $d(C) \geq 2t + 1$. A t-error-correcting algorithm for the even weight subcode becomes a t-error-correcting algorithm for the code itself when used twice with two different values of S_0.

Example 6 We consider the cyclic code of length 33 with defining set $R = \{1, 3\}$. The complete defining set contains the set $\{-4, -3, -2, -1, 1, 2,3,4\}$. The actual distance is equal to 10 and the even weight subcode can be decoded up to this distance with the pair (U, V) defined with generating sets $I = \{0, 1, 2, 3, 4\}$ and $J = \{-3, -2, -1, 0\}$.

Feng and Tzeng [5] showed how the trick can be applied with reduced complexity. We recall briefly their argument applied to error-correcting pairs. In many cases we find error-correcting pairs, such that S_0 occurs just once in the key matrix $S(\mathbf{y})$ (6). Without loss of generality we can assume that S_0 occurs in the last column. Let t be the maximal number of errors that we want to decode. If less than t errors have occurred we can find an error-locating polynomial σ from the leftmost columns, i.e. with vanishing coefficient at the last column. We only need to know S_0 if we cannot find such a solution. But then by assumption precisely t errors have occurred which means S_0 is equal to $t \pmod 2$.

Example 7 We consider the $[31, 16, 7]$ binary cyclic code with defining set $R = \{1, 5, 7\}$. An error-correcting pair is described by generator matrices $G_V = M(\{1, 2, 0\})$

and $G_U = M(\{7, 8, 18, 0\})$. The key equation is then given by

$$\begin{pmatrix} S_8 & S_9 & S_{19} & S_1 \\ S_9 & S_{10} & S_{20} & S_2 \\ S_7 & S_8 & S_{18} & S_0 \end{pmatrix} \begin{pmatrix} \sigma_7 \\ \sigma_8 \\ \sigma_{18} \\ \sigma_0 \end{pmatrix} = 0.$$

If less than 3 errors occurred, we will find a vector σ with $\sigma_0 = 0$ which locates the error positions. If we cannot find such a vector we assume three errors to have occurred and this means S_0 is equal to 1. So whenever S_0 is needed in order to calculate the error-locator polynomial, we know its value.

V Pairs from MDS-codes

A A class of MDS-codes

In this section we assume that the field $\mathbb{F}$ is finite of order q. Also let $(n, q) = 1$. As in the previous section let $\overline{\mathbb{F}} \supset \mathbb{F}$ contain the n-th roots of unity.

Theorem 3 *Let C and A denote two cyclic codes over $\overline{\mathbb{F}}$ of length n. Let their defining sets be given by*

$$R_C = \{1, q^l, q^{2l}, \ldots, q^{rl}\}, \quad \textit{and} \quad R_A = \{1, q^l, q^{2l}, \ldots, q^{sl}\},$$

for $l, r, s > 0$ with $r < s$. Then

$$d(C) \;=\; \min\{\, |R_C| + 1,\; d(A)\,\}.$$

In particular the code C is MDS for $|R_C| < d(A)$.

Proof. Clearly $d(C) \leq |R_C| + 1$ by the Singleton bound. It suffices to prove for a word $\mathbf{c} \in C$ of weight $\mathrm{wt}(\mathbf{c}) \leq |R_C|$ that $\mathbf{c} \in A$. The columns in $M(R_C)$ corresponding to the support of $\mathbf{c}$ are dependent. Thus the submatrix of $M(R_C)$ formed by these columns has row-rank less than $|R_C|$. The submatrix of $M(R_A)$ formed by columns at the support of $\mathbf{c}$ has a linear relation among its top $|R_C|$ rows. Taking coefficients and rows to the power q^l yields a relation on lower rows and it is seen that the two submatrices have the same row-space. □

Remark 10 The conclusion and the proof of the theorem remain the same when zero is added to the defining sets R_C and R_A

Example 8 The code C over $\overline{\mathbb{F}} = GF(2^{11})$ of length $n = 23$ with defining set $R_C = \{0, 1, 4, \ldots, 4^r\}$ is MDS for $r \leq 5$.

B Construction of pairs

For a t-error-correcting BCH-code C with defining set $R \supset \{1, 2, \ldots, 2t\}$ we have by Theorem 2 a t-error-correcting pair (U, V). The codes U and V have generating sets $I = \{0, 1, \ldots, t\}$ and $J = \{1, 2, \ldots, t\}$ respectively. More general, we have

Proposition 1 *Let the codes U and V be MDS of dimension $k(U) = t + 1$ and $k(V) = t$ respectively. A code C with $C \perp U * V$ has distance $d(C) \geq 2t + 1$. Moreover it has the t-error-correcting pair (U, V).*

Proof. Let $\mathbf{c} \in C$ have support of weight w. For $t + 1 < w < 2t + 1$, Theorem 5 in [10] yields $w \geq (t + 1) + t$, a contradiction. For $0 < w \leq t + 1$, it yields $w \geq w + 1$, again a contradiction. Thus $d(C) \geq 2t + 1$. The conditions (1)–(3) and (11) for an error-correcting pair follow immediately. □

Remark 11 In case the code U is not MDS, but otherwise the conditions on U and V are satisfied the pair (U, V) in the lemma is still t-error-locating for a code C with $C \perp U * V$.

We give two applications of MDS codes obtained with Theorem 3. In both cases, U and V are chosen such that the condition $C \perp U * V$ leads to a small defining set for C. Furthermore the key-equation (6) can be solved with complexity $\mathcal{O}(t^2)$ in both cases. Here the complexity is estimated by the number of required multiplications in the field $\overline{\mathbb{F}}$.

Theorem 4 *(first conjugacy format) Let the codes $U/\overline{\mathbb{F}}$ and $V/\overline{\mathbb{F}}$ have generating sets $I = \{1, q^l, q^{2l}, \ldots, q^{tl}\}$ and $J = \{0, q^l, q^{2l}, \ldots, q^{(t-1)l}\}$, for $t \geq 2$. Let $t = t_l$ be maximal such that both U and V are MDS of dimension $k(U) = t + 1$ and $k(V) = t$ respectively. For $2 \leq t \leq t_l$, a code $C/\mathbb{F}$ with*

$$R_C \supset \{1, 2, q^l + 1, q^{2l} + 1, \ldots, q^{(t-1)l} + 1\}$$

has the t-error-correcting pair (U, V). The key equation (6) can be solved with complexity $\mathcal{O}(t^2)$.

Proof. The value of t_l can be obtained with Theorem 3. The complete defining set R for C satisfies $R \supset I + J$ and thus $C \perp U * V$. We may use Proposition 1. For the solving of the key equation see Section VI. □

Theorem 5 *(second conjugacy format) Let the codes $U/\overline{\mathbb{F}}$ and $V/\overline{\mathbb{F}}$ have generating sets $I = \{0, 1, q^{2l}, q^{4l}, \ldots, q^{(2t-2)l}\}$ and $J = \{0, q^l, q^{3l}, \ldots, q^{(2t-3)l}\}$, for $t \geq 2$. Let $t = t_l$ be maximal such that both U and V are MDS of dimension $k(U) = t + 1$ and $k(V) = t$ respectively. For $2 \leq t \leq t_l$, a code $C/\mathbb{F}$ with*

$$R_C \supset \{0, 1, q^l + 1, q^{3l} + 1, \ldots, q^{(2t-3)l} + 1\}$$

has the t-error-correcting pair (U, V). The key equation (6) can be solved with complexity $\mathcal{O}(t^2)$.

Proof. As in Theorem 4. □

Example 9 We consider codes of length $n = 23$ over $GF(2^{11})$. Let U and V be as in Theorem 5 with $q = 2, l = 1, t = 3$, or $I = \{0, 1, 4, 16\}$ and $J = \{0, 2, 8\}$. By Example 8 both U and V are MDS and we have found a 3-error-correcting pair for the even weight subcode C of the binary Golay code, since $R_C \supset \{0, 1, 3, 9\}$.

Lemma 8 (recurrences) *If a pair (U, V) is t-error-locating and the generating sets I and J are of a conjugacy format, then the syndromes $S_i = \langle \mathbf{e}, \mathbf{a}(i) \rangle$ can be determined for*

$$\begin{array}{ll} i = q^{sl} + 1, \quad \textit{for } s \geq 1, & \textit{first format (Theorem 4).} \\ i = q^{(2s-1)l} + 1 \quad \textit{for } s \geq 1, & \textit{second format (Theorem 5).} \end{array}$$

Proof. The case $s < t$ is obvious. For $s \geq t$ we use induction. For both formats we may assume that the error-locating word $\mathbf{u} \in U$ has non-zero coordinate σ_1 at $\mathbf{a}(1)$. We have, for the first format,

$$\begin{aligned} 0 &= \langle \mathbf{e} * \mathbf{u}, \mathbf{a}(q^{sl}) \rangle \\ &= \langle \mathbf{e}, \mathbf{u} * \mathbf{a}(q^{sl}) \rangle \\ &= \sigma_1 \langle \mathbf{e}, \mathbf{a}(q^{sl} + 1) \rangle + \text{known terms.} \end{aligned}$$

Similar for the second format. □

Example 10 We consider codes of length $n = 39$ over $GF(2^{12})$. Let U and V be as in Theorem 4 with $q = 2, l = 1, t = 4$, or $I = \{1, 2, 4, 8, 16\}$ and $J = \{0, 2, 4, 8\}$. The code V is MDS and we have found a 4-error-locating pair for the binary code C with $R_C \supset \{1, 3, 5, 9\}$. It is of type $[39, 15, 10]$. By the lemma we can determine syndromes corresponding to the checks 17 and $65 \equiv 26 \pmod{39}$ and the error values can be determined by Lemma 6.

VI Complexity

A Short description of the Fundamaental Iterative Algorithm

In their paper [5], Feng and Tzeng proposed a fundamental iterative algorithm (FIA). For a matrix A, it gives the minimal set of dependent leading columns. It basically solves an arbitrary homogeneous system of linear equations and contains the Berlekamp-Massey algorithm as a special case. The algorithm is not required to make our decoding procedure work but it seems to be a key algorithm in treating complexity aspects. We recall it in a form that allows us to complete the proof of Theorem 4 and Theorem 5.

Whenever it is necessary for reasons of dimension, we extend a vector with a suitable number of zeros. Let the matrix $A^{(a,b)}$ be the submatrix of A consisting of the elements in the first a rows and the first b columns of A. For a fixed b, we consider column vectors σ with non-zero coordinate at position b that solve the equation

$$A^{(a,b)}\sigma = \mathbf{0}.$$

Let $a = a^{(b)}$ be maximal such that a solution exists and let $\sigma = \sigma^{(b)}$ be such a solution. To assure that a solution exists we use the convention $A^{(0,b)} = \mathbf{0}^T$. For these a and σ, let $\Delta^{(b)}$ be defined as

$$\Delta^{(b)} = \sum_{k=1}^{b} A_{a+1,k}\sigma_k,$$

or as $\Delta^{(b)} = 0$ when $A\sigma^{(b)} = \mathbf{0}$. For given $\{\,(\sigma^{(b)}, \Delta^{(b)}, a^{(b)})\,\}_{b<i}$ the idea of the FIA is now to calculate $\sigma^{(i)}$ with help of the $\sigma^{(b)}$, $b < i$. Starting with any vector σ of length i and σ_i unequal to zero, this is achieved by subtracting suitable scalar multiples of the known $\sigma^{(b)}$ from σ thereby obtaining a new σ. More precisely, whenever σ solves

$$A^{(j,i)}\sigma = \mathbf{0},$$

$$d = \sum_{k=1}^{i} A_{j+1,k}\sigma_k \neq 0,$$

and there exists a triple $(\sigma^{(b)}, \Delta^{(b)}, j)$, we construct

$$\sigma \leftarrow \sigma - \frac{d}{\Delta^{(b)}}\sigma^{(b)},$$

which now solves

$$A^{(j+1,i)}\sigma = \mathbf{0}.$$

Finally we will obtain the triple $(\sigma^{(i)}, \Delta^{(i)}, a^{(i)})$. For details and proofs see [5].

Let us assume that A is a Hankel matrix. By starting the calculation of $\sigma^{(i)}$ with a particular choice for σ, namely a shifted version of $\sigma^{(i-1)}$ with zero in the lowest position, we get the well-known Berlekamp-Massey algorithm. The calculations of most d are not necessary — they are zero or already known by the structure of Hankel matrices. This is the crucial point in saving complexity. In the next section we will show how to apply the FIA to matrices $S(\mathbf{y})$ obtained with Theorem 4 and Theorem 5. It turns out that solving the linear systems described by these matrices is achieved with basically the same complexity as used for the Berlekamp-Massey algorithm.

B Reducing complexity

In general the complexity of the procedure described in Section II equals $\mathcal{O}(n^3)$. This is due to the fact that only matrix inversions and multiplications are involved. We found

two possibilities to improve on the number of computations. One approach uses regular structures of the matrix $S(\mathbf{y})$ and the other approach reduces the size of the field that contains the entries of $S(\mathbf{y})$.

In the case of the conjugacy format, the matrix $S(\mathbf{y})$ has a highly regular structure. We explicitly treat the first conjugacy format. For the second conjugacy format similar considerations hold. We write the generating sets defining U and V for the first conjugacy format in the following ordered form

$$J = \{q^{l(t-1)}, q^{l(t-2)}, q^{l(t-3)}, \ldots, q^l\}, \quad I = \{1, q^l, q^{2l}, \ldots, q^{tl}\},$$

excluding the zero in J. We recall the definition of S_i given in Lemma 6.

$$S_i = \langle \mathbf{e}, \mathbf{a}(i) \rangle, \quad i \in R.$$

Obviously $S_i = \langle \mathbf{y}, \mathbf{a}(i) \rangle$ holds for all $i \in R$.

Lemma 9 *With generator matrices for U and V corresponding to the above ordering, the entries in $S(\mathbf{y})$ satisfy*

$$S(\mathbf{y})_{j,i} = (S(\mathbf{y})_{j+1,i-1})^{q^l}, \quad j = 1, ..., t-2 \ , i = 2, ..., t+1.$$

Proof. The entry $S(\mathbf{y})_{j,i}$ is equal to the syndrome $S_{h(j,i)}$ with

$$h(j,i) = q^{l(t-j)} + q^{l(i-1)}.$$

Obviously $h(j,i)$ is equal to $q^l h(j+1, i-1)$. The vector $\mathbf{y}$ was assumed to be defined over a field $\mathbb{F}$ of cardinality q. Hence $S_{q^l h} = {S_h}^{q^l}$ and the lemma follows. □

We see from Lemma 9 that the format of $S(\mathbf{y})$ is very similar to a Hankel matrix. Thus to find $S(\mathbf{y})$ we only have to calculate the entries in the first row and the last column. The other entries are found using the lemma. This structure is now used in finding the space of solutions to the key equation in the same way as in the Berlekamp-Massey algorithm. Recall that the complexity gain in using the Berlekamp-Massey algorithm was due to the fact that given a vector $\sigma^{(b)}$, which solves equation

$$A^{(a,b)}\sigma = 0,$$

we find a vector σ that solves equation

$$A^{(a-1,b+1)}\sigma = 0$$

as a shifted version of $\sigma^{(b)}$ with zero in the lowest position.

Proposition 2 *Let $S(\mathbf{y})$ be the $(t-1)\times(t+1)$-matrix of Lemma 9. Solving*

$$S(\mathbf{y})\sigma = \mathbf{0}$$

for σ can be done with complexity $\mathcal{O}(t^2)$.

Proof. Consider a typical step in the FIA. Given a solution $\sigma^{(i-1)}$ to the equation $S(\mathbf{y})^{(a,i-1)}\sigma = \mathbf{0}$, we also have a solution to the equation $S(\mathbf{y})^{(a-1,i)}\sigma = \mathbf{0}$. The latter solution is obtained by taking all elements in $\sigma^{(i-1)}$ to the q^l-power and shifting them by one position. Using normal bases for the field, raising a number to the q^l-th power can be performed by a cyclic shift, not requiring computational complexity. Whenever possible, we now perform an update of σ. This is done by performing the following operation

$$\sigma \leftarrow \sigma - \frac{d}{\Delta^{(b)}}\sigma^{(b)} \qquad \text{with } d = (\Delta^{(i-1)})^{q^l},$$

and the calculation of a new d. The whole step requires at most $\mathcal{O}(t)$ operations and we find a solution to $S(\mathbf{y})^{(a,i)}$. Thus in every complexity demanding step, starting from a solution to the system $S(\mathbf{y})^{(a,b)}$ we find a solution to the system $S(\mathbf{y})^{(a',b')}$ such that $a'+b'$ is equal to $a+b+1$. On the other hand $a'+b'$ is bounded by $2t$ which is the sum of the number of rows and the number of columns in $S(\mathbf{y})$. So we have to perform at most $2t$ times a calculation requiring $\mathcal{O}(t)$ operations. The proposition follows. □

Remark 12 (on the proof of Theorem 4 and Theorem 5) To complete the proof of Theorem 4, we have to add a row to $S(\mathbf{y})$ of Lemma 9. This row caused by the zero in J does not fit into the quasi Hankel format. This causes one additional step in the FIA with complexity $\mathcal{O}(t)$.Theorem 5 requires not only an additional row but also an additional column. We add this column as the rightmost column of $S(\mathbf{y})$. The FIA needs at most $\mathcal{O}(t)$ operations for every position in this column. In both cases the overall complexity is still ruled by $\mathcal{O}(t^2)$.

Example 11 The quadratic residue code of length 41 is a good example for the second conjugacy format. We find $I = \{1, 23, 37, 31, 0\}$ and $J = \{8, 20, 9, 0\}$ with $q^l = 2^3$. By the results of section 3 we can set $S_0 = 0$ whenever it is needed. An error-locating polynomial is found by solving the system

$$\begin{pmatrix} S_{8^5+1} & S_{8^2(8^3+1)} & S_{8^4(8+1)} & S_{8^5(8+1)} & S_{8^5} \\ S_{8^3+1} & S_{8^2(8+1)} & S_{8^3(8+1)} & S_{8^3(8^3+1)} & S_{8^3} \\ S_{8+1} & S_{8(8+1)} & S_{8(8^3+1)} & S_{8(8^5+1)} & S_8 \\ S_1 & S_{8^2} & S_{8^4} & S_{8^6} & S_0 \end{pmatrix} \sigma = 0.$$

We see that the submatrix $S(\mathbf{y})^{(3,4)}$ has a quasi Hankel structure which can be utilized to solve the system.

In considering cyclic codes of length n, in most cases codes U and V will be defined over the smallest field $\overline{I\!F}$ containing an n-th root of unity. In some cases however U and V can be taken to be codes defined over $\widehat{I\!F} \subset \overline{I\!F}$. This implies that $S(\mathbf{y})$ has entries from $\widehat{I\!F}$ rather than from $\overline{I\!F}$ which allows us to perform these operations faster.

Example 12 Let $n = 15$. Let C be the double error-correcting BCH-code with $R_C = \{1, 2, 3, 4, 6, 8, 9, 12\}$. To show how the choice of I and J influences the decoding, we notice two possible choices. First we see that we can choose $J = \{1, 2\}$ and $I = \{0, 1, 2\}$ and this would correspond to the usual decoding as a subcode of a RS-code. A different choice is $I = \{2, 8, 0\}$ and $J = \{1, 4, 0\}$. This choice corresponds to cyclotomic cosets with respect to $GF(4)$. S_0 is only needed if two errors occurred which gives $S_0 = 0$. $S(\mathbf{y})$ will be a matrix over $GF(4)$ and all calculations will only involve computations over $GF(4)$.

VII More on pairs

A Error-location by hyperplanes

In some cases we have a pair that does not satisfy condition (3). If the pair does satisfy the weaker condition (9), all solutions to the key equation are still error-locating. Example 16 treats such a situation. The situation becomes quite different when also condition (9) fails.

Proposition 3 *For a given code C, let the pair (U, V) satisfy conditions (1),(2) and let $W \neq \mathbf{0}$, with*

$$W = (\mathbf{e} * U) \cap V^{\perp}.$$

Then, for $\mathbf{y} \in \mathbf{e} + C$, the key equation (4)

$$\sum_{i=0}^{n-1} y_i u_i v_i = 0, \qquad \textit{for all } \mathbf{v} \in V,$$

has at least $m = k(W) + 1$ independent solutions $\mathbf{u}_1, \ldots, \mathbf{u}_m \in U$. Also, there exist $\lambda_1, \ldots, \lambda_m$ such that

$$\mathbf{e} * (\lambda_1 \mathbf{u}_1 + \ldots + \lambda_m \mathbf{u}_m) = \mathbf{0}. \tag{20}$$

Proof. We may replace $\mathbf{y}$ in (4) with $\mathbf{e}$. Clearly $\mathbf{u} \in U$ is a solution whenever $\mathbf{e} * \mathbf{u} \in V^{\perp}$. In other words the space of solutions is the inverse image of W under the linear map $U \longrightarrow \mathbf{e} * U$, $\mathbf{u} \longmapsto \mathbf{e} * \mathbf{u}$. The map has non-trivial kernel by condition (2). The vectors $\mathbf{e} * \mathbf{u}_1, \ldots, \mathbf{e} * \mathbf{u}_m$ are all in W and hence are dependent. □

With the vectors $\{\mathbf{u}_1, \ldots, \mathbf{u}_m\}$ we associate the n points $(u_{1i}, \ldots, u_{mi})$, $i = 1, \ldots, n$ in affine m-space. By the proposition all points corresponding to error positions are contained in a hyperplane through the origin

$$H : \lambda_1 X_1 + \ldots + \lambda_m X_m = 0.$$

Proposition 4 *For a given code C, let the pair (U, V) satisfy conditions (1),(2) and let $d(V^\perp)$ be equal to t. If the key equation (4) has a one-dimensional solution space spanned by $\mathbf{u}_1 \in U$ then*

$$\mathbf{e} * \mathbf{u}_1 = \mathbf{0}$$

is satisfied. If (4) has at least two independent solutions $\mathbf{u}_1, \mathbf{u}_2 \in U$ then the error points either lie in affine 2-space on a line through the origin excluding the origin or they coincide with the origin.

Proof. If the solution space is one-dimensional then by Proposition 3 $k(W)$ is equal to zero and condition (9) is satisfied. Otherwise $k(W)$ is not greater than one because the support of W coincides with the support of $\mathbf{e}$ and $d(W)$ is equal to t. Let $U(\mathbf{y})$ denote the space spanned by $\mathbf{u}_1$ and $\mathbf{u}_2$. If W is contained in $\mathbf{e} * U(\mathbf{y})$ then by Proposition 3 the error points lie on a line through the origin. At least one of $\mathbf{u}_1$ and $\mathbf{u}_2$ is unequal to zero at all error positions so the origin can be excluded. If W is not contained in $\mathbf{e} * U(\mathbf{y})$ both vectors $\mathbf{u}_1$ and $\mathbf{u}_2$ satisfy condition (9) and the error points coincide with the origin. □

Remark 13 In the above proposition we are looking for lines in affine space containing at least t points. There can be no more than n/t such lines. Thus solving for error values can be done in parallel, not affecting the time complexity. The computational complexity in this case is affected by a factor n/t.

Example 13 We consider the binary cyclic code C of type $[31, 11, 11]$ with $R = \{1, 3, 5, 11\}$. The complete defining set contains

$$\{1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13\}.$$

We choose $I = \{1, 2, 3, 8, 9, 10\}$ and $J = \{0, 1, 2, 3\}$. Conditions (1),(2) and (11) are satisfied with $t = 5$, but $d(V^\perp) = 5$ and condition (3) is not satisfied. We only have $k(W) \leq 1$. We may obtain two independent solutions $\mathbf{u}_1$ and $\mathbf{u}_2$ to the key equation. Following the remark this gives 31 points in the affine plane. The linear combination of $\mathbf{u}_1$ and $\mathbf{u}_2$ that locates the error positions corresponds to a line passing through the origin and additional five of the 31 points or $\mathbf{u}_1$ and $\mathbf{u}_2$ are both error-locating. The code corresponds to entry 19 in Table 1 of Section VIII. Entries 47,119,124,137,140,144,146 proceed likewise.

B Generalized Hamming weight

Assume now that conditions (1)-(3) are satisfied for a pair (U, V). Let $U(\mathbf{y})$ denote the solutions of the key equation for a received word $\mathbf{y}$. Any $\mathbf{u} \in U(\mathbf{y})$ locates the error positions. Combination of several solutions reduces the possible error patterns. We use the concept of generalized Hamming weight [21].

Proposition 5 *Let (U, V) be a t-error-locating pair for the code C as in Definition 1. Combination of $k(U) - t$ independent error-locating vectors $\mathbf{u} \in U(\mathbf{y})$ determines the error values uniquely if the following is satisfied:*

$$d(U, k(U) - t) + d(C) > n,$$

where $d(U,i)$ denotes the i-th generalized Hamming weight of U and n denotes the code length of C.

Proof. Immediate from the definition of generalized Hamming weight. □

Example 14 We consider binary Reed-Muller codes of length $n = 2^m$, for $m > 3$. See [12] for the definition and the main properties. The code $C = \mathcal{R}(m-3, m)$ has distance $d(C) = 8$. The dual code is the second order Reed-Muller code, $C^\perp = \mathcal{R}(2, m)$. For the decoding of C we set $U = V = \mathcal{R}(1, m)$. In particular $U * V \subset C^\perp$. Furthermore $k(U) = m + 1 > 3$ and $d(V^\perp) = d(\mathcal{R}(m-2, m)) = 4 > 3$. Thus the pair (U, V) satisfies Definition 1 and is 3-error-locating for the code C.For a codeword $\mathbf{u}$

$$(\mathbf{u} + \mathbf{1}) * \mathcal{R}(m-4, m) \subset C$$

holds. The zero set of $\mathbf{u}$ supports a subcode of C and the pair (U, V) is not 3-error-correcting. However we can reduce the possible error positions by combining $k(U) - 3 = m - 2$ independent solutions to the key equation. We have $d(U, i) = 2^m - 2^{m-i}$ and the combination reduces the number of possible error positions to four.
After puncturing the code C is cyclic and the pair (U, V) is defined with $I = J = \{0, 1, 2, 4, \ldots, 2^{m-1}\}$. In this case there may appear only three possible error positions, the fourth coinciding with the punctured position.

VIII Cyclic codes of length less than 63

Table 1 gives error-correcting pairs for binary cyclic codes which have error-correction capability exceeding the error-correction capability given by the BCH bound. We use the same numbering for codes as in [10]. Equivalent codes and subcodes with the same error correction capability are included in the table as remark. In four cases (no. 92,123,132,146) we stay one short of the actual error-correction capability. All other pairs allow decoding up to half the actual minimum distance of the code.
To check conditions (1) and (2) of Definition 1 is straightforward. In all but four cases (no. 85,106,107,137), the code V is MDS. This follows either immediately using the BCH-bound or with Theorem 3. Thus, also condition (3) is easily verified in these cases. In the cases 85 and 137, the distance $d(V^\perp)$ is obtained with the Hartmann-Tzeng bound and Theorem 3 respectively. Cases 106 and 107 are treated in Example 16 and Example 17 respectively. Usage of hyperplanes (Proposition 4) or usage of the unknown syndrome S_0 is indicated as remark. The remark FT indicates that the same error-correcting pair is given by Feng and Tzeng in [6].
To show that the pairs are error-correcting we use either the BCH-bound to show that condition (11) is satisfied or we use recurrences to determine unknown syndromes until we can apply Lemma 6. Whenever we use the conjugacy format, Lemma 8 provides us with a possibility to determine some unknown syndromes.

Cases that require other recurrences are listed in Table 2. For brevity we introduce the following notation. $\sigma_i \neq 0 : R(j) \rightarrow S_{i+j}$ means that we use equation (19) in Lemma 7,

$$\sigma_{i_l} S_{i_l+j} + \ldots + \sigma_{i_2} S_{i_2+j} + \sigma_{i_1} S_{i_1+j} = 0,$$

with the indicated j and that S_{i+j} will be the only unknown in the equation. Thus it can be obtained provided $\sigma_i \neq 0$. Recurrences separated by a comma can be computed in parallel.

Example 15 (26) The codes defined with $R = \{1, 3, 11\}$ and $R = \{3, 5, 11\}$ are equivalent. The given pairs are also equivalent. The codes U and V in the latter pair however have generator matrices G_U and G_V respectively that are defined over $GF(32)$. With this choice of G_U and G_V the key equation (6) can be solved over the field $GF(32)$.

Example 16 (106) The code C is of type $[51, 27, 8]$ with $R \supset \{1, 3, 9\}$. The pair (U, V) is defined with $I = \{2, 8, 12, 0\}$ and $J = \{1, 4, 0\}$. We need S_0 only when three errors occurred and may then set $S_0 = 1$ (see section 3). For the error-location all conditions except (3) are obviously satisfied. In fact $d(V^\perp) = 3$ and condition (3) does not hold for $t = 3$. We prove the weaker condition (9): $(\mathbf{e} * U) \cap V^\perp = \mathbf{0}$. Words of weight three in $V^\perp/GF(256)$ are in the code with defining set $R = \{1, 4, 16, 13, 0\}$ by Lemma 3. But $\{1, 4\} + \{0, 12\} \subset R$ and the support of a word of weight three must be of the form

$$\{\alpha, \rho\alpha, \rho^2\alpha\},$$

for α a 51-th root of unity and ρ a primitive third root of unity. Up to multiplication with a scalar the values at these positions are $(1, \rho, \rho^2)$. But the values of $\mathbf{u} \in U$ at these positions are a linear combination of $(1, 1, 1)$ and $(1, \rho^2, \rho)$ and (9) is satisfied.In [6] the pair $I = \{0, 2, 8, 12\}$ and $J = \{0, 1, 4\}$ is given without the above verification.

Example 17 (107) The code C is of type $[51, 27, 9]$ with $R \supset \{1, 5, 9\}$. The pair (U, V) is defined with $I = J = \{8, 13, 2, 7, 0\}$. We need S_0 only when four errors occurred and may then set $S0 = 0$ (see section 3). For the error-location we prove (3): $d(V^\perp) > 4$. A word $\mathbf{c} \in V^\perp$ satisfies the checks $\{8, 13, 2, 0\}$ and by theorem 3 also $R = \{8, 13, 2, 16, 26, 4, 32, 1, 0\}$ if it is of weight four or less. Thus $d(V^\perp) \geq 4$. Let $\mathbf{c}$ have non-zero values (c_1, c_2, c_3, c_4). At the same support we have codewords with values $(c_1c_1, c_1c_2, c_1c_3, c_1c_4)$ and $(c_1^2, c_2^2, c_3^2, c_4^2)$. Thus $c_1 = c_2 = c_3 = c_4$. Example 32 in [10] shows that a binary code with $R \supset \{1, 7\}$ has no words of weight four, using $R \supset \{0, 3\} + \{1, 2, 4\}$. The error-correction follows with the recurrences in Table 2.

Example 18 (123) There is an extended choice for U and V, namely $I = \{0, 1, 2, 3, 4, 19, 36\}$ and $J = \{13, 14, 15, 16, 17, 32, 49\}$. Assume six errors have occurred and we find a space of polynomials with σ_{36} equal to zero. Then we can apply the hyperplane method and the given recurrences. If there is no such solution then the error positions do not support a codeword in $V^\perp$ and the solution with $\sigma_{36} \neq 0$ is error-locating. An error-locator

of the form $\sigma_2 X^2 + \sigma_{19} X^{19} + \sigma_{36} X^{36}$ can occur and then the zero set supports an eight dimensional subcode of C. Solutions of another form determine the error values uniquely:

$$\begin{aligned}
\sigma_0 \neq 0: &\quad R(31) \to S_{31}, R(48) \to S_{48}. \\
\sigma_1 \neq 0: &\quad R(30) \to S_{31}, R(47) \to S_{48}. \\
\sigma_3 \neq 0: &\quad R(34) \to S_{37}, R(0) \to S_3. \\
\sigma_4 \neq 0: &\quad R(33) \to S_{37}, R(50) \to S_3.
\end{aligned}$$

IX Sequences of codes

In a certain range of the minimum distance, the conjugacy formats of Section V allow the construction of sequences of cyclic codes with the same designed distance and redundancy as BCH-codes.

Proposition 6 *Let n be equal to $2^m - 1$, $m = 2l + 1$. For a binary cyclic code C with defining set $R_C \supset \{1, 2^l + 1, 2^{l-1} + 1\}$ we have a 3-error-correcting pair (U, V) with generating sets $I = \{0, 1, 2^{2l}, 2^{4l}\}$ and $J = \{0, 2^l, 2^{3l}\}$.*

Proof. The code is defined in [12, Ch.9 §11]. There it is also proved that the distance $d = 7$. For the even weight subcode we may write $R \supset \{0, 1, 2^l+1, 2^{3l}+1\}$. Thus we find the formats of Theorem 5, except that the code U is not MDS. In fact the codes U and V are equivalent to codes with generating sets $I' = 4 \cdot I = \{0, 4, 2, 1\}$ and $J' = 4 \cdot 2^l \cdot J = \{0, 2, 1\}$. The conditions (1)-(3) and (11) are fulfilled. Since S_0 occurs only once in the key matrix $S(\mathbf{y})$, by the results of Section IV we can assume that $S_0 \equiv 3 \pmod 2$. □

Remark 14 (QR-codes) It is clear from Table 1, that the second conjugacy format yields good pairs for the smaller binary QR-codes. The conjugacy formats require defining sets of size t, while the BCH-format requires sets of size $2t$ (the largest set of consecutive quadratic residues is in general not formed by the residues $\{1, 2, \ldots, 2t\}$ and the usual argument that only $\{1, 3, \ldots, 2t-1\}$ need to be in the defining set does not apply). With a uniform distribution of the quadratic residues, the conjugacy formats should correct about twice the number of errors of the BCH-format. Calculations for codes of length less than 1024 agree with this. For example, for the codes of length $n = 863$ and $n = 887$, we apply Theorems 4 and 5 with $q = 2$. They yield pairs to correct $t = 10\,(l = 57)$ and $t = 7\,(l = 62)$ errors for $n = 863$ and $t = 8\,(l = 182)$ and $t = 11\,(l = 206)$ errors for $n = 887$. For both values of n, the BCH-format corrects $t = 4$ errors. Also, it is clear that both the BCH-format and the conjugacy formats have a capability that is of order $\log(n)$ for large codelength n. This is way below the square root bound.

In addition to these we give the following sequences.

Proposition 7 *Let C be a binary cyclic code of length n where 3 does not divide n. Let R_C contain the set $\{-1, 1\}$. Then a 2-error-correcting pair (U, V) is given through*

generating sets $I = \{-3, 0, 3\}$ *and* $J = \{-1, 1\}$.

Proof. The complete defining set R for C satisfies $R \supset I + J$. 3 does not divide the length and it follows that U and V are both MDS. The proof follows from Lemma 1. □

Example 19 (Zetterberg codes [22]) Let n be equal to $2^{2m} + 1$. The Zetterberg code C with defining set $R_C = \{1\}$ has the 2-error-correcting pair (U, V) given in Proposition 7.

Example 20 (Melas codes [14]) Let n be equal to $2^m - 1$ and let m be odd. The Melas code C with defining set $R_C = \{1, -1\}$ has the 2-error-correcting pair (U, V) given in Proposition 7.

The following sequence of reversible codes contains as members binary codes of type $[73, 37, \geq 11]$ and $[85, 45, \geq 11]$.

Proposition 8 *Let C be a binary cyclic code of length n where 3 does not divide n. Let R_C contain the set $\{-7, -5, -1, 1, 5, 7\}$. Then a 5-error-correcting pair (U, V) is given through generating sets $I = \{-4, -2, -1, 1, 2, 4\}$ and $J = \{-6, -3, -0, 3, 6\}$.*

Proof. Use Theorem 2. □

X Conclusions

In this paper we have presented algebraic decoding algorithms for linear cyclic codes. For binary codes of length less than 63, the full error-correction capability of the code is achieved in all but four cases. The decoding procedures depend heavily on the idea of *error-correcting pairs* which was first developed by Pellikaan in [15]. This idea unifies to a certain extent algebraic decoding techniques and as is shown in Section IV, former results obtained by Feng and Tzeng in [5] have a clear interpretation. As is shown in Section V, error-correcting pairs from MDS-codes are very often the key to the decoding of certain classes of cyclic codes. So the decoding of classical BCH-codes employs RS-codes in the corresponding error-locating pairs. We have derived a new class of MDS-codes which are used to describe a new family of codes. This new family resembles in many ways the classical BCH-codes and contains members like the $[23, 12, 7]$ Golay code and the $[41, 21, 9]$ QR-code. In correspondence with BCH-codes an efficient decoding procedure for the new family is derived which uses a Berlekamp Massey type algorithm. Thus the decoding of codes in this family needs only $\mathcal{O}(n^2)$ operations.

References

[1] E.R. Berlekamp, *Algebraic coding theory.* New York: McGraw-Hill, 1968.

[2] R.E. Blahut, *Theory and practice of error control codes.* Reading, Ma.: Addison-Wesley, 1983.

[3] P. Bours, J.C.M. Janssen, M. van Asperdt, and H.C.A. van Tilborg, "Algebraic decoding beyond e_{BCH} of some binary cyclic codes, when $e > e_{BCH}$," *IEEE Trans. Inform. Theory*, vol.IT-36, pp.214-222, 1990.

[4] M. Elia, "Algebraic decoding of the (23,12,7) Golay code," *IEEE Trans. Inform. Theory*, vol.IT-33, pp.150-151, 1987.

[5] G.L. Feng and K.K. Tzeng, "A generalization of the Berlekamp-Massey algorithm for multisequence shift-register synthesis with applications to decoding cyclic codes," *IEEE Trans. Inform. Theory*, vol.IT-37, pp.1274-1287, 1991.

[6] G.L. Feng and K.K. Tzeng, "Decoding Cyclic and BCH codes up to Actual Minimum Distance Using Nonrecurrent Syndrome Dependence Relations," *IEEE Trans. Inform. Theory*, vol.IT-37, pp.1716-1723, 1991.

[7] V.D. Goppa, "Codes on algebraic curves," *Soviet Math. Dokl.*, vol. 24, pp.170-172, 1981.

[8] J. Justesen, K.J. Larsen, H.E. Jensen, A. Havemose and T. Høholdt, "Construction and decoding of a class of algebraic geometric codes," *IEEE Trans. Inform. Theory*, vol.IT-35, pp.811-821, 1989.

[9] R. Kötter, "A unified description of an error locating procedure for linear codes", Proceedings of the international workshop on Algebraic and Combinatorial Coding Theory, Voneshta Voda, Bulgaria 1992.

[10] J.H. van Lint and R.M. Wilson, "On the minimum distance of cyclic codes," *IEEE Trans. Inform. Theory*, vol.IT-32, pp.23-40, 1986.

[11] J.H. van Lint, *Introduction to Coding Theory.* New York: Springer-Verlag, 1982.

[12] F.J. MacWilliams and N.J.A. Slone, *The Theory of Error-Correcting Codes.* Amsterdam: North-Holland, 1977.

[13] J.L. Massey, "Shift-register synthesis and BCH decoding," *IEEE Trans. Infor. Theory*, vol.IT-15, pp.122-127, 1969.

[14] C. M. Melas, "A cyclic code for double error correction," *IBM J. Res. Devel.,* 4, pp.364-366, 1960.

[15] R. Pellikaan, "On decoding linear codes by error correcting pairs," *Eindhoven University of Technology*, preprint, 1988.

[16] R. Pellikaan, "On decoding by error location and dependent sets of error positions," *Discrete Mathematics*, vol.106-107, pp.369-381, 1992.

[17] W.W. Peterson and E.J. Weldon Jr., *Error-correcting codes.* Cambridge, MA: M.I.T. Press, 1971.

[18] I. S. Reed, T. K. Truong, X. Chen, and X. Yin, "The algebraic decoding of the (41,21,9) quadratic residue code," *IEEE Trans. Inform. Theory*, vol.IT-38, pp.974-986, 1992.

[19] A.N. Skorobogatov and S.G. Vlăduţ, "On the decoding of algebraic-geometric codes," *IEEE Trans. Inform. Theory*, vol.IT-36, pp.1051-1060, 1990.

[20] P. Stevens, "Extension of the BCH decoding algorithm to decode binary cyclic codes up to their maximum error-correction capacity," *IEEE Trans. Inform. Theory*, vol.IT-34, pp.1332-1340, 1988.

[21] V. K. Wei, "Generalized Hamming weights for linear codes," *IEEE Trans. Inform. Theory*, vol.IT-37, pp.1412-1418, 1991.

[22] L. H. Zetterberg, "Cyclic codes from irreducible polynomials for correction of multiple errors," *IEEE Trans. Inform. Theory*, vol.IT-8, pp.13-20, 1962.

no.	n	k	d	R	J	I	$d(V^{\perp})$	Remark
3	17	9	5	{1}	{-1,+1}	{-3,0,+3}	3	$GF(16)$
					{0,8}	{1,8,13}	3	
8	21	9	8	{0,1,3,7}	{0,1,2}	{0,1,2,6}	4	FT
9		7	8	{1,3,7,9}	{0,1,2}	{1,2,6,7}	4	no.10;FT
11	23	12	7	{1}	{2,8,0}	{1,4,16,0}	4	no.12,S0=1
13	31	21	5	{1,5}	{1,4}	{0,1,4}	3	no.15
17		16	7	{1,5,7}	{0,1,2}	{0,7,8,18}	4	no.18,FT,S0=1
19		11	11	{1,3,5,11}	{0,...,3}	{1,2,3,8,9,10}	5	no.21;H,Ex.13
25	33	13	10	{1,3}	{-2,...,+2}	{-2,...,+2}	5	$GF(32)$,S0=0,1
26		11	11	{1,3,11}	{0,10,20,30,40}	{1,2,11,15,24,25}	6	no.27;FT
					{-2,...,2}	{-13,-12,-11,+11,+12,+13}	6	$GF(32)$,no.27,Ex.15
36	35	16	7	{1,5,7}	{0,3,6}	{1,2,4,5}	4	no.37,38,39;FT
40		7	14	{0,1,3,5}	{0,...,5}	{31,...,34,0,1,8}	7	FT
41	39	26	6	{0,1}	{0,1}	{0,1,4}	3	no.44;FT,S0=0
45		15	10	{1,3}	{0,2,4,8}	{1,2,4,8,16}	5	no.46,Ex.10
					{0,...,3}	{1,2,3,8,9}	5	FT,Ex.3
47		13	12	{1,3,13}	{0,...,3}	{1,2,3,8,9,10}	5	no.48;H
49	41	21	9	{1}	{8,20,9,0}	{1,23,37,31,0}	5	no.50
51	43	29	6	{1}	{1,2}	{0,20,40}	3	FT
52		15	13	{1,3}	{-6,...,-1}	{0,...,6}	7	S0=0,1
73	45	23	7	{1,5,21}	{0,1,2}	{31,32,33,38}	4	no.74;FT
83		16	10	{0,1,3,7}	{0,...,3}	{-2,...,1,11}	5	no.87,88;FT
84		15	9	{1,3,7,15}	{0,...,3}	{1,11,...,14}	5	no.87,89;Ex.4
85		15	10	{1,7,9,15}	{0,1,2,13,14,15}	{13,...,17}	5	no.86,FT
90		9	12	{1,5,7,9,15}	{13,...,17}	{0,1,2,13,14,15}	6	no.91;
92	47	24	11	{1}	{3,27,8,0}	{1,9,34,24,0}	5	no.93,S0=0
96	51	35	5	{1,9}	{0,1}	{1,8,15}	3	no.97,101,105;FT
					{0,8}	{1,8,13}	3	no.97,101,105;
98		34	6	{0,1,5}	{0,1}	{0,1,4}	3	no.104;FT
106		27	8	{1,3,9}	{0,2,8}	{0,1,4,16}	3	no.109,111,115;S0=1,Ex.16
107		27	9	{1,5,9}	{0,2,7,8,13}	{0,2,7,8,13}	5	no.110,113,116;S0=0,Ex.17
108		27	9	{1,3,19}	{-4,...,-1}	{0,...,4}	5	no.114,117;S0=0,1
119		19	14	{1,3,5,9}	{0,...,4}	{0,...,4,12,6}	6	no.120,S0=0,H
121		17	12	{1,3,9,17,19}	{-4,...,0}	{0,...,4,19}	6	no.125,S0,FT
122		17	14	{1,3,5,17,19}	{-4,...,1}	{0,...,6}	7	S0=0,1
123		17	14	{1,5,9,17,19}	{13,...,17}	{0,...,4,19}	6	no.126;Ex.18
124		17	16	{1,3,5,9,17}	{0,...,5}	{0,...,5,12,13}	7	no.127,H,S0=1
128		11	15	{1,3,5,11,19}	{-7,...,-1}	{0,...,7}	8	no.130,S0=0,1
132		8	24	{0,1,3,5,9,11,17}	{0,...,8}	{0,...,10}	10	H
135	55	35	5	{1}	{0,9}	{7,8,9}	3	FT
					{7,13}	{1,49,36}	3	
136		34	8	{0,1}	{0,7,13}	{0,1,49,36}	4	
137		30	10	{0,1,11}	{0,7,13,32}	{0,1,49,36,4}	4	H
138		25	11	{1,5}	{0,7,16,13,14}	{1,18,49,2,36,43}	6	no.139
140		21	15	{1,5,11}	{0,7,16,13,14,32}	{0,1,18,49,2,36,43,4}	7	no.141;H,S0=1
144	57	21	14	{1,3}	{0,2,4,8,16}	{0,-1,-2,-4,-8,-16,-32}	6	no.145;H,S0=0,1
146		19	16	{1,3,19}	{0,2,4,8,16}	{0,-1,-2,-4,-8,-16,-32}	6	no.147;H,S0=0,1

no.	condition	recurrence	syndromes
17	$\sigma_{18} \neq 0$	$R(25) \to S_{12}$, $R(28) \to S_{15}$; $R(24) \to S_{11}$	S_3, S_{15}; S_{11}
	$\sigma_{18} = 0 \wedge \sigma_8 \neq 0$	$R(7) \to S_{15}$, $R(9) \to S_{17}$,$R(18) \to S_{26}$	S_{15}, S_3, S_{11}
26	$\sigma_{24} \neq 0$	$R(16) \to S_7$	S_7
	$\sigma_1 \neq 0$	$R(6) \to S_7$	S_7
	$\sigma_{25} \neq 0$	$R(1) \to S_{26}$	S_7
	$\sigma_2 \neq 0$	$R(24) \to S_{26}$	S_7
73	$\sigma_{38} \neq 0$	$R(33) \to S_{26}$	S_7
83	$\sigma_0 \neq 0$	$R(5) \to S_5$,$R(18) \to S_{18}$	S_5, S_9
	$\sigma_0 = 0 \wedge \sigma_{11} \neq 0$	$R(25) \to S_{36}$; $R(10) \to S_{21}$; $R(44) \to S_{10}$	S_9, S_{21}, S_5
90	$\sigma_{15} \neq 0$	$R(27) \to S_{42}$	S_{21}
	$\sigma_{15} = 0 \wedge \sigma_{14} \neq 0$	$R(28) \to S_{42}$	S_{21}
	$\sigma_{15} = 0 \wedge \sigma_{14} = 0 \wedge \sigma_{13} \neq 0$	$R(29) \to S_{42}$	S_{21}
96	$\sigma_1 \neq 0$	$R(13) \to S_{14}$; $R(2) \to S_3$; $R(40) \to S_{41}$	$S_5; S_3; S_{11}$
106	$\sigma_1 \neq 0$	$R(48) \to S_{49}$; $R(9) \to S_{10}$, $R(43) \to S_{44}$	S_{19};S_5,S_{11}
107	$\sigma_7 \neq 0$	$R(5) \to S_{12}$;$R(16) \to S_{23}$	S_3;S_{11}
	$\sigma_7 = 0 \wedge \sigma_8 \neq 0$	$R(16) \to S_{24}$; $R(3) \to S_{11}$	S_3;S_{11}
121	$\sigma_0 \neq 0$	$R(14) \to S_{14}$,$R(23) \to S_{23}$	S_5,S_{11}
	$\sigma_0 = 0 \wedge \sigma_{19} \neq 0$	$R(46) \to S_{14}$;$R(4) \to S_{23}$	S_5;S_{11}
123	$\sigma_0 \neq 0$	$R(31) \to S_{31}$,$R(48) \to S_{48}$	S_{11},S_3
	$\sigma_0 = 0 \wedge \sigma_{19} \neq 0$	$R(12) \to S_{31}$;$R(29) \to S_{48}$	S_{11};S_3
136	$\sigma_1 \neq 0$	$R(32) \to S_{33}$;$R(28) \to S_{29}$; $R(19) \to S_{20}$	S_{11};S_3;S_5
137	$\sigma_1 \neq 0$	$R(28) \to S_{29}$; $R(19) \to S_{20}$	S_3;S_5