Decoding codes from curves and cyclic codes

Decoding Codes from Curves and Cyclic Codes

Iwan M. Duursma

Decoding Codes from Curves and Cyclic Codes

Decoding Codes from Curves and Cyclic Codes

Proefschrift

ter verkrijging van de graad van doctor aan de Technische Universiteit Eindhoven, op gezag van de Rector Magnificus, prof. dr. J.H. van Lint, voor een commissie aangewezen door het College van Dekanen in het openbaar te verdedigen op maandag 13 september 1993 om 16.00 uur

door

Iwan Maynard Duursma

geboren te Bussum

druk: wibro dissertatiedrukkerij, helmond.

Dit proefschrift is goedgekeurd door de promotoren
prof. dr. J. H. van Lint
en
prof. dr. H. Stichtenoth

copromotor: dr. G. R. Pellikaan

CIP-GEGEVENS KONINKLIJKE BIBLIOTHEEK, DEN HAAG

Duursma, Iwan Maynard

Decoding codes from curves and cyclic codes / Iwan Maynard
Duursma. – Eindhoven : Technische Universiteit Eindhoven
Proefschrift Eindhoven. – Met lit. opg.
ISBN 90-386-0212-X
Trefw. : coderingstheorie / algebraïsche meetkunde.

Preface

The thesis describes results of my research on decoding linear codes. This research has been carried out at the Eindhoven University of Technology in the period September 1990 – March 1993. It was supported by the Netherlands Organization for Scientific Research (NWO), through the foundation Stichting Mathematisch Centrum. Some of the results have found their way to international journals. In the order in which they have appeared as preprints they are:

1. "Algebraic decoding using special divisors," IEEE Transactions on Information Theory, volume 39, March 1993.

2. "On the decoding procedure of Feng and Rao," Proceedings Algebraic and Combinatorial Coding Theory III Conference, Voneshta Voda, Bulgaria, June 1992.

3. "Majority coset decoding," IEEE Transactions on Information Theory, volume 39, May 1993.

4. with R. Kötter, "Error-locating pairs for cyclic codes," preprint Eindhoven-Linköping, submitted for publication, March 1993.

The papers [1] and [3] each work towards a single theorem, given on page 39 and page 52 respectively of this thesis. The paper [4] has four theorems, on pages 71, 77, 78 and 78. In addition, the tables at pages 86 and 87 give explicit decoding procedures. The introduction further explains the topic and the contributions of the thesis.

I would like to thank the Discrete Mathematics group for the stimulating working environment, professor J.H. van Lint for his support and his comments on the preprints and R. Pellikaan for numerous discussions and managing the NWO project. Thanks also to professor G. van der Geer for introducing me to codes from algebraic curves and to professor H. Stichtenoth and professor R. Schoof for further discussions on algebraic curves. To all, I express my gratitude for judging the final manuscript. For the joint work on cyclic codes, I am grateful to the coauthor R. Kötter.

I thank the NWO for its financial support. For further support and for hospitality I wish to thank the University of Trento and professor R. Schoof, and the University of Linköping and professor T. Ericson.

May, 1993. Iwan Duursma

Contents

Introduction

The decoding problem has its roots in communication theory. Two electronic devices exchange information and due to noise or otherwise it will happen that the information received differs from the information sent. It is then assumed that the difference is in general small. In any case it is intended to have small differences by a proper design of the devices and if possible of the channel that connects them. The decoding problem is to attach the proper interpretation to the received information. If the possibilities for the received information are few, the interpretation can be attached to these once and for all and stored in a table. We consider situations where such tables are not feasible. Without a table, i.e. without deciding about interpretations beforehand, one needs a set of rules that can be applied any time information is received. Obviously, one prefers the set of rules to be small and the rules to be such that they can be carried out quickly. These two characteristics determine to a certain extent the size and the speed of a decoding device.

Decoding is not only a problem of the receiver. The sender and receiver together determine how information is to be sent. In air-traffic control it is common use to avoid "yes" and "no" and to say "affirm" and "negative" instead. Sender and receiver have agreed upon this and it greatly enhances the reliability of communication. Similarly, with two electronic devices, the information will be encoded before it is transmitted. The encoding should improve the reliability of the communication and allow easy decoding by the receiver.

The problem has a fruitful translation into mathematics: message, encoded message and received message are associated with suitable strings of letters. The strings of letters are easily transferred into the language of the electronic devices (i.e. strings of zeros and ones) and the rules for encoding and decoding can be formulated in terms of operations that can be carried out by a microprocessor. A small example is obtained with messages of length two that use three different letters A, B, and C. The nine possible messages are:

AA,	AB,	AC,
BA,	BB,	BC,
CA,	CB,	CC.

The messages are very much alike and by changing one letter a message is transferred into a different message. To enable the receiver to recognize that a letter was changed, and thus to improve the reliability of the communication, the messages are encoded as follows:

AAAA,	ABBB,	ACCC,
BABC,	BBCA,	BCAB,
CACB,	CBAC,	CCBA.

The set of possible encoded messages is called a code C, the encoded messages are called codewords. In the example, any two different codewords differ in precisely three positions and agree in the remaining position. If the encoded message reaches the receiver with one letter changed, the receiver can conclude by comparing with all codewords that something went wrong during transmission of the message. Moreover, he can find the message sent as the unique codeword that best resembles the received message, i.e. that differs from the received message in one position. For a given code C, the decoding problem can be formulated as:

- Find the codewords that agree with the received message in a maximum number of positions. If there is only one such codeword, take this codeword for the message sent. Otherwise, leave the interpretation undecided or make a choice.

The bottle-neck in the problem is how to find the codewords that resemble the received message. The most straightforward solution is to compare the received message with all codewords. This is far from efficient and we mention two other strategies. The first strategy uses combinatorial properties of the code and is known as *permutation decoding*. It basically consists of two steps. In general, the steps have to be executed several times:

- Guess which letters are correct.
- Find the codewords that match these letters.

Recall, that in the example any two different codewords agree in precisely one position. This combinatorial property tells us how to make suitable guesses: a codeword is determined by any two of its letters and it suffices to guess a combination of two letters correctly. Suitable guesses are that either the first two received letters or the last two received letters are correct. If one error occurred, one of the guesses is true and yields the codeword. For the received message BBCC, the codewords that match BB-- and --CC are BBCA and ACCC respectively. Thus, the fourth letter was changed and the message sent was BBCA.

Permutation decoding is fast, but only few examples are known where the strategy works well. The second strategy applies to *linear codes*, that have the structure of a *vector space*. It uses algebraic properties of the code and is known as *algebraic decoding*. It consists of two steps, that are executed only once, but that take more time than the steps in the previous strategy:

- Compute which letters are correct.
- Compute the codeword that matches these letters.

Note that we assume that sufficiently many correct letters are computed, such that only one codeword will match the correct letters. The code in the example is linear. After replacing the letters A, B and C by the numbers 0, 1 and 2 respectively:

0000,	0111,	0222,
1012,	1120,	1201,
2021,	2102,	2210,

we can formulate an algebraic property. Let $c_1c_2c_3c_4$ denote a codeword. For any codeword $u_1u_2u_3u_4$, the number $N = c_1u_1 + c_2u_2 + c_3u_3 + c_4u_4$ is divisible by three.

Again, let BBCA, or 1120, be the message sent and let BBCC, or 1122, be the received message. To verify if the received message is a codeword, we compute the number N, for all codewords $u_1u_2u_3u_4$. For comparison, we also compute the numbers N for the message sent:

$N = u_1 + u_2 + 2u_3 + 2u_4$			$N = u_1 + u_2 + 2u_3$		
0,	5,	10,	0,	3,	6,
7,	6,	5,	3,	6,	3,
8,	7,	6.	6,	3,	6.

Since the numbers in the left table are not all divisible by three, we conclude that the received message is not a codeword and that an error has occurred. On the other hand, the numbers in the right table illustrate the algebraic property and are all divisible by three. It is clear that the difference between the two tables is caused by the addition of $2u_4$ in the left table. The number N in this table is divisible by three only if the codeword $u_1u_2u_3u_4$ has u_4 equal to zero. Thus, to find the position of the error, it suffices to find the codewords for which N is divisible by three, namely 0000, 1120 and 2210. The error occurred at the position where these codewords have a zero. The codeword that matches the first three letters of the received message 1122 is 1120.

We refer to the first chapter for further details. In particular for the claim that it is possible to do the computations in a fairly straightforward way. In general, the codewords $u_1u_2u_3u_4$ are taken from a code U different from the code C. Also, for the computations, a third code V is required. The choice of codes U and V that make the procedure work is not obvious. For given codes U and V, the procedure is fairly straightforward.

In this thesis, I give results for the algebraic decoding of two families of linear codes. For each family, methods are given for the construction of decoding procedures. These can only be obtained by using additional features of a particular family. On the other hand, many questions arise with both families that can be answered for arbitrary linear codes without the restriction to a particular family. The results that are valid for arbitrary linear codes are presented in the first part. The first section gives the formulation of a general algebraic decoding procedure. In the next two sections, conditions are given that ensure that the procedure works in particular cases. The theory is then applied to an example that is not contained in one of the two families. The last section gives modifications of the general procedure. They apply to some cases where the conditions for the general procedure are not met.

The family of *codes from curves*, also called *algebraic geometry codes* or simply *AG-codes*, is in many ways remarkable. The codewords can still be identified with strings of letters, but they have a more natural interpretation as rational functions or rational differentials on an algebraic curve. It is immediately clear from the last interpretation and by using well-known results from algebraic-geometry that AG-codes have very good properties. For their application in practice, efficient decoding procedures are required. That algebraic decoding can be applied to AG-codes was noticed in 1988. The procedure as it was then formulated is called the basic algorithm. AG-codes have an obvious lower bound on the number of errors that can be corrected, called the designed capability. The basic algorithm does not correct up to this bound. For a particular class of AG-codes, a modified algorithm was formulated that corrects more errors but in general still not up to the designed capability. In Theorem 3.13, I give a formulation of the modified algorithm that applies to all AG-codes, rather than to a particular class. Several other improvements of the basic algorithm have been suggested. The idea of Feng and Rao is to use different but related procedures, such that if not the codeword itself at least some more information about the codeword will be obtained. Their idea is worked out in Chapter 4. Theorem 4.13 shows that all AG-codes can be decoded up to the designed capability without any further restrictions. The most time consuming calculations in the procedure involve solving systems of linear equations. This is not yet fast enough for applications.

For the family of *cyclic codes*, algebraic decoding procedures have been known since the 1960's. Among these procedures, several are fast enough for applications and have been implemented in chips. Similar to AG-codes, cyclic codes have an obvious designed capability. The general decoding procedure for cyclic codes does not correct beyond the designed capability. On the other hand, many of the best cyclic codes have an actual capabil-

ity that is better than their designed capability. More recent papers give procedures to decode some of these codes. A simpler and more general formulation of these procedures with a shorter proof is given in Theorem 5.6. Two other theorems give decoding procedures for particular classes of cyclic codes. The amount of computation in the procedures compares precisely with the general procedure, while the performance is much better for the classes considered. The binary quadratic residue codes of length 23 and 41 can be decoded in this manner. The binary cyclic codes of length less than 63 that have an actual capability exceeding the designed capability have been classified. The theorems in this part yield decoding procedures for all of these but four.

Part II and Part III are independent and both follow after Part I. Within Part II, the Chapters 3 and 4 are independent. Both follow after Chapter 2. Within Chapter 4, Section 4.5 is independent of the previous sections. One common bibliography is included at the end of the thesis. The results of Part I and Part III were obtained in co-operation with R. Kötter.

The main results appeared in separate preprints and articles and are included in their original form. They are divided over the thesis as follows: Chapter 3 [9], Chapter 4 [11], Section 4.5 [10] and Part I and Part III [12]. Additions in this thesis concern remarks, examples and cross-references. By abuse, we use the phrase decoding up to the minimum distance, where up to half the minimum distance is meant.

Part I

Algebraic decoding

Chapter 1

A unified description

The most successful methods for decoding 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 [36], who used it to describe the decoding of AG-codes. Independently but later, Kötter [26] 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. 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.

1.1 Error-locating pairs

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 minimum distance by $d(C)$, or by k and d respectively when no confusion arises.

Definition 1.1 (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

$$U * V \subseteq C^{\perp}, \tag{1.1}$$

$$k(U) > t, \tag{1.2}$$

$$d(V^{\perp}) > t. \tag{1.3}$$

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

Theorem 1.2 *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*

$$\sum_{i=0}^{n-1} y_i u_i v_i = 0, \qquad \textit{for all } \mathbf{v} \in V. \tag{1.4}$$

Moreover, any solution $\mathbf{u} \in U$ of (1.4) satisfies

$$\mathbf{e} * \mathbf{u} = \mathbf{0} \tag{1.5}$$

Proof. In (1.4) we may replace $\mathbf{y}$ by $\mathbf{e}$ by condition (1.1). Thus any vector $\mathbf{u} \in U$ with property (1.5) is a solution to (1.4). Condition (1.2) guarantees the existence of a non-zero vector. This is because we impose at most t linear conditions on U. To prove (1.5), we note that (1.4) 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 (1.3) we find (1.5). □

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 (1.4). By property (1.5), the coordinates of the vector $\mathbf{u}$ take the value zero at the error positions. 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 elsewhere. Equation (1.4) 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 (1.4) can be rephrased as

$$S(\mathbf{y}) \cdot \sigma^T = \mathbf{0}, \tag{1.6}$$

where

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

Any solution σ for (1.6) gives a solution $\mathbf{u} = \sigma G_U$ for (1.4). The vector $\mathbf{u}$ satisfies (1.5). 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 (1.1)-(1.3) for a maximal value of t.

Remark 1.3 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 (1.6) at a later stage.

Remark 1.4 Given a particular error vector $\mathbf{e}$, it is clear from the proof of Theorem 1.2 that the following conditions are sufficient to obtain $\mathbf{u} \in U \backslash \mathbf{0}$ with property (1.5):

$$C * U \subseteq V^{\perp}, \tag{1.7}$$

$$\exists \mathbf{u} \in U \backslash \mathbf{0} \ : \ \mathbf{e} * \mathbf{u} = \mathbf{0}, \tag{1.8}$$

$$\forall \mathbf{u} \in U \backslash \mathbf{0} \ : \ \mathbf{e} * \mathbf{u} \in V^{\perp} \ \Rightarrow \ \mathbf{e} * \mathbf{u} = \mathbf{0}. \tag{1.9}$$

The first condition is equivalent to (1.1). Conditions (1.8) and (1.9) are weaker than conditions (1.2) and (1.3) respectively. We will have to refer to them in some cases where the conditions in Definition 1.1 are too strong.

1.2 Error-locating functions

Theorem 1.2 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 (1.1)-(1.3) locates all error patterns of a given weight. Such a pair is hard to find in general. Conditions (1.7)-(1.9) 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 1.5 *For an error vector* $\mathbf{e}$, *let* E *(resp.* $\overline{E}$*) be the subspace of* $\mathbb{F}^n$ *consisting of all vectors that have zero components in the error (resp. 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 (1.7)-(1.9). □

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

Notation 1.6 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 without 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 1.7 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 naturally induces an $\mathbb{F}$-algebra isomorphism between R/I and S.

Example 1.8 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 $x^n - 1$. Cyclic codes are the subject of Part III.

Example 1.9 For algebraic-geometric codes, an evaluation map Ev occurs in their definition [21],[52]. AG–codes are the subject of Part II. In this chapter we take the conditions (1.1)–(1.3) as starting point to study decoding, since they are general and apply to any linear code. In Sections 2.2 and 2.3, we point out the particulars of AG–codes and their decoding.

Now, let an algebra-homomorphism $Ev : R \longrightarrow S$ be given as in Notation 1.6. The reformulation of Lemma 1.5 becomes

Lemma 1.10 *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 1.6. For an error vector* **e**, *let J (resp. $\overline{J}$) be the ideal in R consisting of all elements that evaluate to zero at the error (resp. 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 1.5. □

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 [4, p.248] or AG-codes [24, 48] use this approach. $L(C), L(U)$ and $L(V^\perp)$ have here a natural interpretation.

1.3 Error-correcting pairs

The previous sections show 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 1.11 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 1.12 *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*

$$\forall \mathbf{c} \in C \; : \quad \mathbf{c} * \mathbf{u} = \mathbf{0} \;\Rightarrow\; \mathbf{c} = \mathbf{0}. \tag{1.10}$$

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 (1.10) 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 [36]. See also [26].

Definition 1.13 (t-error-*correcting* pair) Let (U, V) be a t-error-locating pair for the code C as in Definition 1.1. We call (U, V) a t-error-correcting pair for the code C if in addition to the conditions (1.1),(1.2) and (1.3) the following is satisfied

$$d(C) + d(U) > n, \tag{1.11}$$

where n denotes the code length of C.

Remark 1.14 The definition is justified by the lemma since condition (1.11) implies

$$\forall \mathbf{c} \in C, \forall \mathbf{u} \in U \; : \quad \mathbf{c} * \mathbf{u} = \mathbf{0} \;\Rightarrow\; \mathbf{c} = \mathbf{0} \;\vee\; \mathbf{u} = \mathbf{0}. \tag{1.12}$$

In some cases we will prefer to use the weaker condition (1.12).

Remark 1.15 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 1.16 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):

$$L(U) \cap J \neq (0), \tag{1.13}$$

$$\langle L(C) * L(U) \rangle \cap \overline{J} = (0). \tag{1.14}$$

The dilemma in algebraic decoding is obvious. For (1.13), we want $L(U)$ to be large and for (1.14) we want $\langle L(C) * L(U) \rangle$ to be small, that means $L(U)$ should be small.

1.4 Example: projective RM-code

The general properties of projective Reed-Muller codes are described in [28], [49]. For a finite field $\mathbb{F}$ of order q, let R denote the graded ring in three variables $R = \mathbb{F}[X, Y, Z] = \sum_{d=0}^{\infty} R_d$, where R_d consists of the homogeneous polynomials in X, Y, Z of degree d. We assume $0 \in R_d$, $d \geq 0$. Let $\mathcal{P}$ denote the set of rational points in $PG(2, \mathbb{F})$. For $d \geq 0$, the image of R_d in $\mathbb{F}^{q^2+q+1}$, obtained by mapping a polynomial to its evaluation in the rational points, defines a linear code C_d. For $d \geq q + 1$, the mapping has non-trivial kernel and we deviate from Notation 1.6.

Lemma 1.17

$$\forall f \in R_{2(q-1)} : \sum_{P \in \mathcal{P}} f(P) = 0.$$

Proof. The sum is well-defined and in particular does not depend on the representation of the rational points. It suffices to prove the equality for a summation over the set of affine points $\mathcal{A}$ in $AG(3, \mathbb{F})$. Also, it suffices to prove the claim for monomials $f = X^a Y^b Z^c$, for $a + b + c = 2(q - 1)$. We may assume $c < q - 1$ and

$$\sum_{X,Y \in \mathbb{F}} X^a Y^b \sum_{Z \in \mathbb{F}} Z^c = 0.$$

□

Let the codes $U = V$ be defined by $L(U) = L(V) = R_2$ and the code C by $L(C) = R_{2q-6}$, for $q \geq 3$. By the lemma, condition (1.1) is satisfied. To see how many errors can be located with a pair (U, V) we note $k(U) = 6$ and $d(V^\perp) = 4$. The words of minimum weight in $V^\perp$ have as support four collinear points. The pair is 3-error-locating, but error patterns of weight five with no four positions collinear will be located. The distance of C is $d = 13\,(q = 3), d = 12\,(q = 4), d = 10\,(q = 5), d = 6\,(q \geq 7)$. The supports of words of minimum weight are given by a triple of four collinear points $(q = 4)$, a pair of five collinear points $(q = 5)$ and six collinear points $(q \geq 7)$.

The only non-trivial five-error-correcting code is obtained with $q = 4$. The code is of type $[21, 6, 12]$. The pair (U, V) as defined above may fail when four of the error positions are collinear. For some error patterns of weight five, we give the matrix that defines the key equation (1.6). Of course, in practise, the entries are computed with the received word. The obtained matrix, however, only depends on the error pattern. We interprete the solutions to $S(\mathbf{y})\sigma = \mathbf{0}$ as functions in $L(U) = R_2$.

	P_1	P_2	P_3	P_4	P_5
X	1	a	a^2	1	0
Y	1	0	0	0	1
Z	0	1	1	1	0
e	1	1	1	1	1

	X^2	XY	Y^2	XZ	YZ	Z^2
X^2	1	1	1	1	0	0
XY	1	1	1	0	0	0
Y^2	1	1	0	0	0	0
XZ	1	0	0	0	0	0
YZ	0	0	0	0	0	0
Z^2	0	0	0	0	0	1

The solution YZ cancels at the error positions. It has 9 zeros and the 12 remaining positions are correct and determine the codeword.

	P_1	P_2	P_3	P_4	P_6
X	1	a	a^2	1	0
Y	1	0	0	0	0
Z	0	1	1	1	1
e	1	1	1	1	1

	X^2	XY	Y^2	XZ	YZ	Z^2
X^2	1	1	1	1	0	0
XY	1	1	1	0	0	0
Y^2	1	1	1	0	0	0
XZ	1	0	0	0	0	0
YZ	0	0	0	0	0	0
Z^2	0	0	0	0	0	0

With one error position replaced, the rank of the matrix has decreased by two. The solutions are spanned by $Y(X + Y), YZ$ and Z^2. The functions that cancel at the error positions are spanned by $Y(X + Y)$ and YZ, and the procedure fails. The error pattern has distance seven to the word $Ev(Y^2 + YZ + Z^2)$. The seven error positions with respect to this word are among the zeros of the function $(Y + Z)Z = 0$. This explains the solution Z^2 and the failure.

	P_1	P_2	P_3	P_4	P_6
X	1	a	a^2	1	0
Y	1	0	0	0	0
Z	0	1	1	1	1
e	1	a^2	a	1	1

	X^2	XY	Y^2	XZ	YZ	Z^2
X^2	0	1	1	0	0	0
XY	1	1	1	0	0	0
Y^2	1	1	1	0	0	0
XZ	0	0	0	0	0	1
YZ	0	0	0	0	0	0
Z^2	0	0	0	1	0	1

With two error values replaced, the error pattern now has distance nine to the word $Ev(Y^2+YZ+Z^2)$. The nine error positions with respect to this word are no longer divided over two lines and the solution $(Y+Z)Z$ no longer holds.

	P_1	P_2	P_3	P_7	P_8
X	1	a	a^2	1	0
Y	1	0	0	1	1
Z	0	1	1	1	1
e	1	a^2	a	1	1

	X^2	XY	Y^2	XZ	YZ	Z^2
X^2	0	0	0	0	1	0
XY	0	0	0	1	1	1
Y^2	0	0	1	1	0	0
XZ	0	1	1	0	1	1
YZ	1	1	0	1	0	0
Z^2	0	1	0	1	0	1

The generic situation: five error positions, no three collinear. The matrix is of rank five and the unique solution $Y^2+X^2+XZ+Z^2$ has as its zeros precisely the five error positions.

For larger q we need to reduce the set $\mathcal{P}$ and a five error-correcting code is obtained when the points on a curve of degree four are taken. Part II is devoted to codes from curves. The basic algorithm for these codes coinsides with the procedure above. Various improvements will be discussed. All improvements can tackle four collinear error positions. However, the improvements do not apply to the code $[21,6,12]$, since no curve over $GF(4)$ of degree four contains 21 points.

For the $[21,6,12]$ code, six errors can be detected with permutation decoding. A set of three non-collinear points is called a triangle, a set of six points such that no three are collinear is called a (type II) oval. The latter provides a set of information symbols. It is easy to see that each triangle is contained in three ovals. Thus, there are 168 ovals. For a received word with six errors, we prove that for at least one oval, the errors occur outside the oval. The result can be improved with the knowledge that the ovals fall into three classes such that the three ovals containing a given triangle divide over the classes [6].

Lemma 1.18 *For an arbitrary set $\mathcal{S}$ of six points in the plane that do not form an oval, an oval of a prescribed class exists that does not intersect the six given points. If the six points form an oval, the prescribed class need to be the class of the oval.*

Proof. Let N_i denote the number of ovals, counted with multiplicities, that intersect the set $\mathcal{S}$ in a set of at least i points. By the inclusion-exclusion principle we need to prove $N_0 - N_1 + N_2 - N_3 + N_4 - N_5 + N_6 > 0$. But $N_0 = 56$, $N_1 = 6 \cdot 16$ and $N_2 = 15 \cdot 4$, independent of $\mathcal{S}$. Also, $N_3 = T$ (the number of triangles in $\mathcal{S}$), $N_4 \geq N_5$ and $N_6 \geq 0$. The inequality to be proved, reduces to $T < 20$. This only fails for an oval, which contains the maximum of 20 triangles. If the prescribed class differs $N_4 = N_5 = N_6 = 0$, if it matches $N_4 = 15, N_5 = 6, N_6 = 1$. □

1.5 Additional methods

We give two modifications of the procedure in Theorem 1.2. They apply to pairs (U, V) that do not satisfy the standard conditions. First, let (U, V) be a pair that does satisfy conditions (1.1) and (1.2), but that may not satisfy condition (1.3). The solutions to the key equation are still error-locating if the weaker condition (1.9) is satisfied. This is the case in three of the four examples in the previous section. The situation becomes quite different when also condition (1.9) fails.

Proposition 1.19 *For a given code C, let the pair (U, V) satisfy conditions (1.1),(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 (1.4)

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

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

$$\mathbf{e} * (\lambda_1 \mathbf{u}_1 + \ldots + \lambda_m \mathbf{u}_m) = \mathbf{0}. \tag{1.15}$$

Proof. We may replace $\mathbf{y}$ in (1.4) 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 (1.2). 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 1.20 *For a given code C, let the pair (U, V) satisfy conditions (1.1),(1.2) and let $d(V^{\perp})$ be equal to t. If the key equation (1.4) has a one-dimensional solution space spanned by $\mathbf{u}_1 \in U$ then $\mathbf{e} * \mathbf{u}_1 = \mathbf{0}$ holds. If (1.4) 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 1.19, $k(W)$ is equal to zero and condition (1.9) 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 1.19 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 (1.9) and the error points coincide with the origin. □

Remark 1.21 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.

Secondly, let the pair (U, V) be t-error-locating, but not t-error-correcting. Thus, the conditions (1.1)-(1.3) are satisfied and condition (1.11) is not. 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 [54].

Proposition 1.22 *Let (U, V) be a t-error-locating pair for the code C as in Definition 1.1. 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 1.23 We consider binary Reed-Muller codes of length $n = 2^m$, for $m > 3$. See [32] 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 1.1 and is 3-error-locating for the code C. For a codeword $\mathbf{u}$

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

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.

Part II

Decoding codes from curves

Chapter 2

Basic algorithm

The basic algorithm (BA) [24], [48] traces the error locations in a received word. It yields a so-called error-locating function and the error locations occur among the zeros of this function. In the original set up, two conditions guarantee the determination of an error-locating function with a sufficiently small number of zeros. In Section 2.3, we weaken the two conditions and show that they still yield correct decoding. In both cases, the claims follow from the results of the previous part. In the present context of codes from curves, we have a natural interpretation of the algebraic decoding procedure. It is discussed in Section 2.4. The basic algorithm does not correct up to the designed capability of a code. In Section 2.5, we recall improvements by Pellikaan [38] and by Ehrhard [16]. Two other improvements are treated in separate chapters.

2.1 Notation

We recall some concepts from the theory of algebraic curves and give their notation and some additional assumptions. The concepts are treated in detail in the books: [7], [20] and [23]. More recent books pay special attention to the case of a finite constant field and to the applications in coding theory: [30],[35],[51] and [52]. Although the concepts are fairly standard, their description may differ a lot from one book to another.

Notation 2.1 In the following, a *curve* $\mathcal{X}$, or $\mathcal{X}/\mathbf{F}_q$, is always absolutely irreducible, non-singular, complete and defined over a finite field, of q elements. The field of rational functions is denoted by $\mathbf{F}_q(\mathcal{X})$, the module of rational differential forms by $\Omega(\mathcal{X})$. Points on a curve are identified with places of the function field, rational points with places of degree one. Let t denote a generator of the maximal ideal of a place. For a function f, we define the divisor $(f) = \sum \nu_t(f)P$, where P runs over all places and ν_t denotes the discrete valuation at P. For a differential ω, we define the divisor $(\omega) = \sum \nu_t(\omega)P$, where P runs over all places and $\nu_t(fdt) = \nu_t(f)$.

A divisor is called principal if it is the divisor of a function. The relation $E_1 \sim E_2$ if and only if $E_1 - E_2$ is principal, defines an equivalence relation on divisors. The unique divisor class containing the divisors (ω), ω a rational differential, is called the canonical divisor class. A representative is denoted by K. For a divisor E, the linear spaces $\Omega(E)$ and $L(E)$ are defined by

$$\begin{aligned} \Omega(E) &= \{\omega \in \Omega(\mathcal{X})^* : (\omega) \geq E\} \cup \{0\}, \\ L(E) &= \{f \in \mathbf{F}_q(\mathcal{X})^* : (f) + E \geq 0\} \cup \{0\}. \end{aligned}$$

The integers $i(E)$ and $l(E)$ denote the dimension of the spaces $\Omega(E)$ and $L(E)$ respectively. Each differential ω induces a natural isomorphism

$$L(E) \xrightarrow{\sim} \Omega((\omega) - E), \qquad f \mapsto f\omega. \tag{2.1}$$

The divisor K satisfies: $deg(K) = 2g - 2$ and $l(K) = g$. The integer g is called the genus of the curve. The genus g of a plane curve of degree m satisfies $g = (m-1)(m-2)/2$. For a plane curve, let the divisor L denote the intersection divisor of a line with the curve.

The main results on algebraic curves to be used are

Theorem 2.2 *(Residue theorem) The summation over all places of the residues of a differential is well-defined and equal to zero.*

Theorem 2.3 *(Approximation theorem) For a divisor E and a finite set of places S, there exists a divisor E' that is linearly equivalent to E and that has support outside S.*

Theorem 2.4 *(Riemann-Roch theorem) The dimensions of $L(E)$ and $\Omega(E)$ are related by*

$$l(E) - i(E) = deg(E) + 1 - g.$$

Theorem 2.5 *(Clifford's theorem) For a divisor E with both $L(E)$ and $\Omega(E)$ non-trivial, the following holds*

$$l(E) \leq \frac{deg(E)}{2} + 1.$$

The results and their proofs are described in the literature mentioned above. The latter two theorems are recalled in a different form in the next chapter. For the definition of a linear code with an algebraic curve, we recall the construction of V.D. Goppa.

Let $\mathcal{X}$ be a curve. Let $P_1, P_2, \ldots, P_n$ be n distinct rational points on the curve. Then for the divisors $D(= P_1 + P_2 + \ldots + P_n)$ and G (defined over $\mathbf{F}_q$) one can define algebraic-geometric codes $C_\Omega(D, G)$, known as *residue code*, and $C_L(D, G)$, known as *functional code*. The pair of divisors $\{D, G\}$ that we use to define a code, corresponds with a pair $\{\mathcal{P}, D\}$ in [52].

Definition 2.6 *(Goppa)* We assume that D and G have disjoint support, without loss of generality. By abuse of notation, we will write $P \in D$, rather than $P \in supp(D)$. The codes $C_\Omega(D,G)$ and $C_L(D,G)$ are defined as the images of the linear maps

$$\begin{aligned} \alpha_\Omega &: \Omega(G-D) \longrightarrow \mathbf{F}_q^n, \qquad \omega \mapsto (res_P(\omega))_{P\in D}, \\ \alpha_L &: L(G) \longrightarrow \mathbf{F}_q^n, \qquad f \mapsto (f(P))_{P\in D}. \end{aligned}$$

In particular, α_Ω and α_L induce natural isomorphisms

$$\begin{aligned} \overline{\alpha_\Omega} &: \Omega(G-D)/\Omega(G) \xrightarrow{\sim} C_\Omega(D,G), \\ \overline{\alpha_L} &: L(G)/L(G-D) \xrightarrow{\sim} C_L(D,G). \end{aligned}$$

The isomorphisms yield expressions for the dimension of the codes.

Theorem 2.7 *(Goppa)* *For $deg(G) > deg(K)$, the code $C_\Omega(D,G)$ has parameters*

$$k \geq deg(K+D-G)+1-g, \qquad d \geq d^* = deg(G-K).$$

For $deg(D) > deg(G)$, the code $C_L(D,G)$ has parameters

$$k \geq deg(G)+1-g, \qquad d \geq d^* = deg(D-G).$$

Based on the isomorphism of linear spaces (2.1), a functional code can always be represented by a residue code. The residue code and the functional code are dual by the Residue theorem.

Remark 2.8 In case the divisors D and G have a rational point P in common, we follow the *H-construction* [52]. Let $ord_P(G) = i$ and let t denote a local parameter at P. Then the mappings α_Ω, α_L are modified at the coordinate P:

$$\begin{aligned} \alpha_{\Omega,P} &: \Omega(G-D) \longrightarrow \mathbf{F}_q, \qquad \omega \mapsto (res_P(t^{-i}\omega)), \\ \alpha_{L,P} &: L(G) \longrightarrow \mathbf{F}_q, \qquad f \mapsto ((t^i f)(P))_{P\in D}. \end{aligned}$$

2.2 Description

Let C be a residue code $C_\Omega(D,G)$. It has dual code $C^\perp = C_L(D,G)$. To apply the procedure of the previous chapter to decode C we need an error-locating pair (U,V) as in Definition 1.1. We choose a pair (U,V) of AG-codes. For a divisor F with support disjoint from D, let $U = C_L(D,F)$ and $V = C_L(D,G-F)$. This will satisfy condition (1.1). The other conditions for a t-error-locating pair are

$$\begin{aligned} &k(U) > t, \\ &d(V^\perp) > t. \end{aligned}$$

With Goppa's theorem we write

$$deg(F) + 1 - g \geq t + 1,$$
$$deg(G - F - K) \geq t + 1.$$

The inequalities are satisfied only if $2t \leq deg(G-K)-1-g = d^*-1-g$. For t in that range they are satisfied with F of degree $\lfloor (deg(G-K)-1+g)/2 \rfloor$. The pair is t-error-correcting when condition (1.11) is satisfied

$$d(U) + d(C) > n,$$

Or

$$deg(D - F) + deg(G - K) > deg(D).$$

But the pair fulfills already the stronger condition $deg(G - F - K) > t$ and is thus t-error-correcting. The proof is from [36]. The decoding procedure itself is formulated in [24] and [48]. We recall their description in terms of algebraic functions without explicit reference to the pair (U, V).

Say the code $C = C_\Omega(D, G)$ has parity check matrix H. Let $\mathbf{y} = (y_P)_{P \in D}$ denote a received word with error pattern $\mathbf{e} = (e_P)_{P \in D}$. Thus,

$$H\mathbf{e}^t = H\mathbf{y}^t. \tag{2.2}$$

An *error-locator function* f is defined by the property

$$f(P) \neq 0 \quad \Rightarrow \quad e_P = 0. \tag{2.3}$$

The BA consists of finding a nonzero error-locator function f and then solving (2.2,2.3). To explain how f can be obtained and to formulate the BA we use

Definition 2.9 With a vector $\mathbf{y} = (y_P)_{P \in D}$ we associate a *one-dimensional syndrome* $S(\mathbf{y})$,

$$\begin{aligned} S(\mathbf{y}) : \quad L(G) &\longrightarrow \mathbf{F}_q, \\ h &\longmapsto \sum_{P \in D} y_P\, h(P). \end{aligned}$$

With a divisor F, we associate a *two-dimensional syndrome* $S(F)$,

$$\begin{aligned} S(F) : \quad L(F) \times L(G - F) &\longrightarrow \mathbf{F}_q, \\ (f, g) &\longmapsto S(\mathbf{y})(fg). \end{aligned}$$

Remark 2.10 The syndrome $S(F)$ depends on the vector $\mathbf{y}$, but this is suppressed in the notation. For a fixed received word $\mathbf{y}$, it will often be necessary to consider the syndrome $S(F)$, for various divisors F, and we anticipate this situation.

Lemma 2.11 *The syndrome $S(\mathbf{y})$ in the definition is a coset invariant, that is*

$$S(\mathbf{y}) = S(\mathbf{e}) \quad \Leftrightarrow \quad \mathbf{y} \in \mathbf{e} + C_\Omega(D, G).$$

A fortiori $S(F)$ is a coset invariant.

Proof. From the definition,

$$S(\mathbf{y} - \mathbf{e}) = 0 \;\Leftrightarrow\; \mathbf{y} - \mathbf{e} \in C_L(D, G)^\perp = C_\Omega(D, G).$$

□

Definition 2.12 For a syndrome $S(F)$, the *key equation* is defined as

$$S(F)(f, g) = 0, \quad \forall g \in L(G - F). \tag{2.4}$$

The vector space of solutions $f \in L(F)$ to the key equation is denoted by $K(F)$, its dimension by $k(F)$.

Remark 2.13 The dimension $k(F)$ only depends on the equivalence class of the divisor F. Thus in considering the dimension, we may assume that the divisors F and D have disjoint support. In three of the later improvements of the basic algorithm, in particular in Proposition 2.36, choices of F occur that contain rational points among their support. As long as the evaluation of $(fg)(P)$ takes place after the multiplication of the functions f and g this poses no problem. The interpretation in terms of error-locating pairs only holds when the functions f and g can be evaluated separately. More important, separate evaluation leads to faster procedures [15]. This is achieved by following the H-construction and using a local parameter t at P,

$$(fg)(P) = (t^i f)(P) \cdot (t^{-i} g)(P), \qquad \text{for } i = ord_P(F).$$

Note that we assume that P is not contained in G.

Lemma 2.14 ([48],[38]) *Let the divisor Q consist of the error locations, that is $Q = \sum_{e_P \neq 0} P$. In general, $L(F - Q) \subseteq K(F)$, and*

$$C_\Omega(Q, G - F) = 0 \quad \Rightarrow \quad L(F - Q) = K(F) \tag{2.5}$$

Proof. In the definition of $S(F)$, we may replace $\mathbf{y}$ by $\mathbf{e}$. The inclusion $L(F - Q) \subseteq K(F)$ is obvious. The assumption is needed for the other inclusion. It implies that $(0, \ldots, 1, \ldots, 0) \in C_L(Q, G - F)$ for all unit vectors of length $deg(Q)$. Thus in the definition of $K(F)$, if g runs through $L(G-F)$, the unit vector with support the point $P \in Q$ poses the restriction $K(F) \subseteq L(F - P)$. Together the unit vectors yield $K(F) \subseteq L(F - Q)$. □

Remark 2.15 ([24, plane curves],[48, general]) The main steps of the BA can be summarized as follows:

(B0) Fix a divisor F.
(B1) Calculation of the key matrix $S(F)$.
(B2) Calculation of a nonzero function f in $K(F)$.
(B3) Calculation of the zero divisor of the function f.
(B4) Calculation of the error values in (2.2,2.3).

In case the curve used is the projective line and the divisors G and F are a multiple of the point at infinity, the algorithm reduces to the Peterson-Gorenstein-Ziegler decoder [4].

The divisor F determines the error patterns that will be corrected. The following theorem gives the degree of the divisor F, that optimizes the algorithm.

Theorem 2.16 ([24, plane curves],[48, general]) *Let $C = C_\Omega(D, G)$ be a residue code. The BA with F a divisor of degree $\lfloor (d^* - 1)/2 + g/2 \rfloor$ with support disjoint from D will correct any error pattern of weight up to $\lfloor (d^* - 1)/2 - g/2 \rfloor$.*

Proof. We have $deg(G-F-Q) \geq deg(G)-(d^*-1) = deg(K)+1$. Thus $C_\Omega(Q, G-F) = 0$ and step (B2) yields $f \in L(F-Q)$. With $deg(F-Q) \geq g$ we can take f nonzero. We may assume that $(d^* - 1)/2 - g/2 > 0$ or $deg(F) < d^* - 1$. Thus in step (B3) at most $d^* - 2$ possible error locations are obtained and step (B4) has the error vector as a unique solution. □

The basic algorithm led Pellikaan to the definition of error-locating pairs and the similarity between the two descriptions we have given is obvious. An advantage of the error-locating pairs is their generality. They will be used in the next part on cyclic codes too. The description in terms of functions on the other hand is formulated in terms of the divisor Q and makes clear why some error patterns of a given weight are correctly decoded, while others of the same weight are not.

Another approach to the decoding of AG-codes was taken by Porter [40]. His approach mimics the use of a key equation involving differentials as in the decoding of classical Goppa codes. In a joint paper with Pellikaan and Shen [41], the proofs of [40] are completed and in some cases corrected. Ehrhard [14] generalized the approach in [40] to obtain a description of the key equation for arbitrary AG-codes. The similarity of his description with the basic algorithm is less obvious and was established in [14].

2.3 Sufficient conditions

We study in more detail the dependence of the basic algorithm on the particular error locations, i.e. the divisor Q. In Lemma 2.18, we present two conditions that guarantee error-correction. They have a more general formulation in the form of the conditions (1.13) and (1.14).

Lemma 2.17 *Let* $\mathbf{y}$ *denote an arbitrary vector of length* $n = deg(D)$. *Let two different vectors* $\mathbf{e}_1$ *and* $\mathbf{e}_2 \in \mathbf{y} + C_\Omega(D, G)$ *have as their support the divisors* Q_1 *and* Q_2 *respectively. Then*

$$\Omega(G - F - Q_1) = 0 \quad \Rightarrow \quad L(F - Q_2) = 0.$$

Proof. We prove the reverse direction and assume $F \sim Q_2 + E$, for $E \geq 0$. Let the support of $\mathbf{x} = \mathbf{e}_1 - \mathbf{e}_2 \in C_\Omega(D, G)$ be given by $Q_x \leq Q_1 + Q_2$. Then, $\mathbf{x} = (res_P(\omega))_{P \in D}$ for some nonzero $\omega \in \Omega(G - Q_x)$. A fortiori $\omega \in \Omega(G - Q_1 - Q_2 - E)$. □

Lemma 2.18 *For a vector* $\mathbf{e} \in \mathbf{y} + C_\Omega(D, G)$ *with support* Q, *let the following be fulfilled*

$$\Omega(G - F - Q) = 0, \quad \textit{and} \tag{2.6}$$

$$L(F - Q) \neq 0. \tag{2.7}$$

Then the vector $\mathbf{e}$ *is the unique vector in* $\mathbf{y} + C_\Omega(D, G)$ *with (2.7). Application of the basic algorithm yields in step (B2) an error-locating function for* $\mathbf{e}$ *and in step (B4) the vector* $\mathbf{e}$ *itself.*

Proof. By (2.6) and the previous lemma, any other vector in the coset does not satisafy (2.7). By (2.6) and Lemma 2.14, the function $f \in K(F)$ is error-locating. By (2.7), we may assume f is non-trivial. The vector $\mathbf{e}$ is a solution to the equations (2.2,2.3). Since f is non-trivial, equation (2.3) can only be fulfilled for a vector which support satisfies (2.7). But we saw that $\mathbf{e}$ is the only such vector. □

If the conditions on Q are fulfilled, it is likely to be for a vector $\mathbf{e}$ of small weight. But this need not be the vector of smallest weight in the coset. To give an example we use

Notation 2.19 Let the coset $\mathbf{y} + C_\Omega(D, G)$ contain two vectors $\mathbf{e}_1$ and $\mathbf{e}_2$ that have disjoint supports. Let the divisors Q_1 and Q_2 consist of the points in the support of $\mathbf{e}_1$ and $\mathbf{e}_2$ respectively. Let the weight of the vector $\mathbf{e}_1 - \mathbf{e}_2$ be equal to the designed minimum distance of the code $C_\Omega(D, G)$. In particular, $G - Q_1 - Q_2 \sim K$.

Example 2.20 For the vectors $\mathbf{e}_1$ and $\mathbf{e}_2$ as above, the implication in Lemma 2.17 is in fact an equivalence. The conditions in Lemma 2.18 can be written as

$$\begin{aligned} \mathbf{e} = \mathbf{e}_1: &\quad L(F-Q_2) = 0, \quad L(F-Q_1) \neq 0. \\ \mathbf{e} = \mathbf{e}_2: &\quad L(F-Q_1) = 0, \quad L(F-Q_2) \neq 0. \end{aligned}$$

It is clear that the conditions for $\mathbf{e} = \mathbf{e}_2$ may very well be satisfied, for $\deg(Q_2) > deg(Q_1)$.

Remark 2.21 In [24],[38],[48],[52, Proposition 3.3.2], uniqueness in step (B4) of the basic algorithm is ensured by posing a restriction on the degree of the divisor F : $deg(F) < d^*$. By the lemma, this condition is redundant. In particular the condition $\deg(G) \geq 4g-2$ can be dismissed in [38] and the use of the definition of $s(H)$ in the uniqueness proof can be avoided in [48],[52, Exercise 3.3.10].

To show that $deg(F) < d^*$ does not follow in general from (2.6, 2.7) a small example suffices.

Example 2.22 Consider a plane curve of degree four and let R_1, R_2 be two different points outside D. Let L be the intersection divisor of a line with the curve. Let $G = 2L - R_1$ and $F = L - R_2$. With $K = L$, conditions (2.6,2.7) are satisfied for Q of degree one, but the condition $deg(F) < d^*$ is not.

The next lemma is a slightly stronger version of Lemma 2.14.

Lemma 2.23 *For a coset $\mathbf{e} + C_\Omega(D, G)$, let $K(F)$ denote the space of solutions to the key equation as in Definition 2.12 and $k(F)$ its dimension. Let the divisor Q consist of the support of $\mathbf{e}$. We have*

$$k(F) \leq l(F-Q) + i(G-F-Q) - i(G-F). \tag{2.8}$$

Proof. In the definition of $S(F)$ we replace $\mathbf{y}$ by $\mathbf{e}$. The inclusion $L(F-Q) \subset K(F)$ is obvious and (2.5) follows from (2.8). For (2.8), we also consider the right null space of $S(F)$ and observe that it includes $L(G-F-Q)$. Thus, we are led to consider a bilinear form $\overline{S}$, defined on the product space $L(F)/L(F-Q) \times L(G-F)/L(G-F-Q)$:

$$\overline{S}(\overline{f}, \overline{g}) = \sum_{P \in Q} e_P f(P) g(P).$$

The domain is isomorphic to the product of linear codes $C_L(F, Q) \times C_L(G-F, Q)$. A non-trivial $\overline{f}$ is contained in the left null space of $\overline{S}$ if and only if $(e_P f(P))_{P \in Q}$ is contained in $C_\Omega(Q, G-F)$. Thus the factor $K(F)/L(F-Q)$ has dimension at most

$$\dim C_\Omega(Q, G-F) = i(G-F-Q) - i(G-F).$$

This proves (2.8). □

It is clear from the proof that the factor $K(F)/L(F-Q)$ is determined not only by the error positions but also by the error values. In the situation of Notation 2.19, we can give a description in terms of positions only. The description is useful not so much for the decoding itself, but for an analysis of the basic algorithm in the cases where it fails.

Lemma 2.24 *Let* $\mathbf{e}_1$ *and* $\mathbf{e}_2$ *be as in Notation 2.19. We have*

$$K(F) = L(F-Q_1) + L(F-Q_2).$$

Proof. It suffices to prove $K(F) \subset L(F-Q_1) + L(F-Q_2)$. By assumption $G \sim K + Q_1 + Q_2$. With $Q = Q_1$ we find

$$\begin{aligned} & \dim(L(F-Q_1) + L(F-Q_2)) \\ = \; & l(F-Q) + l(F-(G-K-Q)) - l(F-(G-K)), \\ = \; & l(F-Q) + i(G-F-Q) - i(G-F). \end{aligned}$$

And we use (2.8). □

Example 2.25 Let $\mathcal{X}$ be a plane curve of degree four and let $G = 4L$. For $F = 2L$ and an error pattern of weight at most five, the conditions (2.6) and (2.7) are fulfilled, and the basic algorithm corrects the error. With the exception of error patterns that have four collinear error positions, in which case condition (2.6) fails. Indeed, let $Q_1 \sim L + P$ and $Q_2 \sim 2L - P$. By the lemma, $K(F) \neq L(F-Q_1)$ and the basic algorithm fails.

2.4 Decoding and approximation

Decoding a received word can be interpreted naturally as an approximation problem. For a residue code $C = C_\Omega(G, D)$ this can be done in two different ways. Either one considers approximation by vectors of finite length (codewords) or by vectors of infinite length (differentials). In the basic algorithm, all data involved are formulated in terms of vectors of finite length. Still, we show that the latter interpretation is more appropriate for the basic algortihm.

Let $\mathbf{y} = (y_P)_{P \in D}$ denote a received n-tuple over $\mathbb{F}_q$. The obvious interpretation is

(1) find a word $(c_P)_{P \in D} \in C \subset \mathbf{F}_q^n$ that minimizes $|\{P \in D : y_P \neq c_P\}|$.

This is the formulation in terms of vector spaces in $\mathbb{F}_q^n$ that defines the task of the decoder. The basic algorithm that we use for the decoding however solves a different approximation problem. Recall the mapping

$$\alpha_\Omega \; : \; \Omega(G-D) \;\longrightarrow\; C, \qquad \omega \mapsto (\, res_P(\omega)\,)_{P \in D}.$$

We assume that $d^* = \deg(G - K) > 0$, therefore the mapping α_Ω is an isomorphism. Problem (1) is thus equivalent with

(2) find a differential $\omega \in \Omega(G - D)$ that minimizes $|Q_\omega|$,
with $Q_\omega = \{P \in D : y_P \neq res_P(\omega)\}$.

Let the mimimum be attained for $Q_\omega = Q$. If $|Q|$ is sufficiently small, that is if $\Omega(G - F - Q) = 0$, the basic algorithm yields a function $f \in L(F - Q)$. It is obtained from the equation

$$\forall g \in L(G - F) : \sum_{P \in D} y_p f(P) g(P) = 0. \tag{2.9}$$

We adapt the notation and extend the constant field to its algebraic closure. The vector $(res_P(\omega))_{P \in D}$ is naturally extended to a vector $(res_P(\omega))_{P \notin G}$ of infinite length. For $\omega \in \Omega(G - D)$ this yields an extension with zeros only. The received word $(y_P)_{P \in D}$ is similarly extended to $(y_P)_{P \notin G}$. We choose the extended vector to be zero outside $P \in D$, so that it differs from the differential in at most the transmitted symbols $P \in D$. Equation (2.9) is still well-defined if we take the summation over $P \notin G$ instead of $P \in D$ and if we substitute for $(y_P)_{P \in D}$ the extended vector $(y_P)_{P \notin G}$. Moreover it yields precisely the same equation for f. It is clear that in solving for f, the divisor D plays no role. The problem solved with the basic algorithm can be formulated as

(3) find a differential ω with at most simple poles outside G,
and with $(\omega) \geq_G G$, such that $L(F - Q_\omega) \neq 0$,
with $Q_\omega = \{P \notin G : y_P \neq res_P(\omega)\}$.

We conclude

Lemma 2.26 *For an extension $\mathbb{F}_{q'} \supset \mathbb{F}_q$ of the constant field of the curve, let $D' \geq D$ be a sum of rational points. The code $C_\Omega(G, D')$ contains $C_\Omega(G, D)$ as a shortened subfield subcode. Let $\mathbf{y} \in \mathbf{e} + C_\Omega(G, D)$ be a received word. Let $\mathbf{e}'$ denote the vector $\mathbf{e}$ extended with zeros and let $\mathbf{y}' \in \mathbf{e}' + C_\Omega(G, D')$. For a divisor F, let $K(F)$ (resp. $K'(F)$) be defined as in Definition 2.9 as the space of solutions to the key equation, for the vector $\mathbf{y}$ (resp. $\mathbf{y}'$). Then $K(F) = K'(F)$.*

Proof. In equation (2.9), we may replace $\mathbf{y}$ by $\mathbf{e}$ and $\mathbf{y}'$ by $\mathbf{e}'$ and the equations take the same form. □

Remark 2.27 The lemma shows that if step (B2) of the basic algorithm fails for a given code, it will also fail for a shortened subfield subcode. In particular, to make use of the possibly better parameters of a shortened code, the basic algorithm is of no use. In step (B3), the advantages of the shortened code are apparent, but the basic algorithm only reaches that step if it does so for the non-shortened code.

Example 2.28 The Klein quartic over $GF(8)$ is the plane curve defined by

$$\mathcal{X} \ : \ X^3Y + Y^3Z + Z^3X = 0.$$

It has 24 rational points and 7 points of degree 2. Let $B_1 = (1:0:0), B_2 = (0:1:0)$ and $B_3 = (0:0:1)$. Let the divisor D be the sum of the 21 other rational points. The choice $G = 3(B_1 + B_2 + B_3)$ yields a residue code $C = C_\Omega(D, G)$ of type $[21, 14, \geq 5]/GF(8)$, which actually is of type $[21, 14, 6]/GF(8)$. A longer code is obtained by extending the constant field of the curve to $GF(64)$. Let the divisor D' be the sum of the 21+14 rational points and let G be defined as before. The code $C' = C_\Omega(D', G)$ is of type $[35, 28, \geq 5]/GF(64)$. It has words of weight five at seven mutually disjoint supports. Each of the seven supports contains a pair of two conjugate points over $GF(64)$ and conversely each of the supports is determined by the pair it contains. Let the points $P_1, P_2, P_3, P_{22}, P_{23}$ form the support of a codeword in C' with nonzero coordinates $c_1, c_2, c_3, c_{22}, c_{23}$. We may assume (after multiplication of the given word by a suitable scalar) that the coordinates c_1, c_2, c_3 are in $GF(8)$. Let

$$\mathbf{e} = (c_1, c_2, c_3, 0, \ldots, 0),$$
$$\mathbf{e}' = (c_1, c_2, c_3, 0, \ldots, 0, 0, \ldots, 0),$$

and let $\mathbf{y} \in \mathbf{e} + C$ and $\mathbf{y}' \in \mathbf{e}' + C'$ be received words for the codes C and C' respectively. The coset of $\mathbf{y}$ has the vector $\mathbf{e}$ as coset leader, the coset of $\mathbf{y}'$ has as coset leader the vector $(0, 0, 0, 0, \ldots, 0, c_{22}, c_{23}, 0, \ldots, 0)$. The basic algorithm applied with F defined over $GF(8)$ leads to the same space of solutions $K(F)$ for both received words. With Lemma 2.24,

$$K(F) = L(F - P_1 - P_2 - P_3) + L(F - P_{22} - P_{23}).$$

It is clear that for the code of length 21, the non-rational points play a role in the basic algorithm. Although the vector $\mathbf{e}$ is a unique coset leader, its support cannot be located with the basic algorithm.

2.5 Improvements

For practical purposes the basic algorithm is not fast enough. Also it does not correct up to the designed distance. In this work we focus on the last problem and mention only briefly contributions to the first problem. The basic algorithm solves systems of linear equations and therefore has complexity $O(n^3)$. This is a worst case complexity and may be improved for special curves or by using more sophisticated algorithms for solving linear equations. In case the curve used is the projective line, the basic algorithm reduces to the Peterson-Gorenstein-Ziegler decoder.

For the computations the Berlekamp-Massey algorithm can be used which has complexity $O(n^2)$. In general the complexity will depend on the chosen model of the curve: the dimension of the space in which it is embedded and the form of the equations. For plane curves one can use the generalization of the Berlekamp-Massey algorithm given by Sakata [45]. This is done in [25], where an algorithm with complexity $O(n^{7/3})$ is obtained. For Hermitian curves, an approach similar to Sakata's algorithm is described in [47]. The paper also discusses efficient encoding of the codes. It is shown in [8] that the approach in [25] can be applied to curves in r-dimensional space with complexity $O(n^{3-2/(r+1)})$.

As for correcting more errors, there are several improvements. In the next chapter, we recall a modification given by Skorobogatov and Vlăduţ and we give a generalization of their result. In Chapter 4, we work out an idea of Feng and Rao. In this section, we recall improvements given by Pellikaan and by Ehrhard and we make some remarks.

All improvements use the fact that the divisor F in the basic algorithm can be chosen freely. Pellikaan gives a condition that ensures that a bounded number of suitable applications of the basic algorithm will yield the error pattern. The precise statement is

Proposition 2.29 *([38])* *Let $Div_k = \{E \in Div(\mathcal{X}) : E \geq 0, \deg(E) = k\}$ be the set of effective divisors of degree k. Let $Pic_0(\mathcal{X})$ be the group of all divisors of degree zero modulo linear equivalence. For $s \geq 2$ a mapping ψ_k^s is defined as*

$$\begin{array}{rccc} \psi_k^s : & Div_k^s & \longrightarrow & Pic_0(\mathcal{X})^{s-1}, \\ & (E_1, E_2, \ldots, E_s) & \longmapsto & ([E_1 - E_2], [E_2 - E_3], \ldots, [E_{s-1} - E_s]). \end{array}$$

For $k \geq g$ (and all $s \geq 2$), the mapping is surjective. Let the error pattern $\mathbf{e}$ *be of weight t and let $r = d^* - 2t = deg(G-K) - 2t$. Let s be such that the mapping ψ_{g-r}^s is not surjective. Then, if $(p_1, p_2, \ldots, p_{s-1})$ is without preimage in ψ_{g-r}^s and has preimage $(F_1, F_2, \ldots, F_s)$ in ψ_{g+t}^s, the basic algorithm yields the error pattern* $\mathbf{e}$ *at least once when applied with $F = F_1, F_2, \ldots, F_s$.*

Proof. One can always find a preimage with the mapping ψ_k^s such that the divisors E_i are not effective. For $k \geq g$ we may assume that the E_i are effective by the Riemann-Roch theorem. For the claim on the basic algorithm it suffices to prove that at least one of the F_i satisfies both (2.6) and (2.7). All F_i satisfy $L(F_i - Q) \neq 0$, and we show that $\Omega(G - F_i - Q) \neq 0$ for all F_i yields a contradiction. Indeed,

$$\Omega(G - F_i - Q) \neq 0 \quad \Rightarrow \quad K \sim G - F_i - Q + E_i,$$

for some effective divisor E_i of degree $g - r$, and we find a preimage of $(p_1, p_2, \ldots, p_{s-1})$ in ψ_{g-r}^s. □

The restriction $deg(G) > 4g - 2$ in the original formulation is omitted (see Remark 2.21). It is possible to give upper bounds for the parameter s using only the data from the zeta-function [38, 53]. The bottle-neck in the proposition is to find the element $(p_1, p_2, \ldots, p_{s-1})$. Once this element is known, one has a decoding procedure for all codes from the given curve that decodes up to $(d^* - r)/2$ errors.

Example 2.30 Recall Example 2.25. For a plane curve of degree four and a residue code defined with $G = 4L$, the basic algorithm does not correct all errors when applied with $F = 2L$. The proposition applies with $g = 3, t = 5, r = 2, s = 2$ and $p_1 \neq [P_1 - P_2]$, for any two rational points P_1, P_2. For example, if the tangents of P_1 and P_2 have empty intersection on the curve, the choice $p_1 = [2P_1 - 2P_2]$ and $F_1 = 2L, F_2 \sim 2L + 2P_2 - 2P_1$ will do. For the Klein quartic defined over $GF(8)$, decoding procedures based on the proposition are given in [44].

Ehrhard formulated an effective procedure to decode up to the designed distance. The main idea is contained in the following two lemmas. Proposition 2.36 gives the procedure for $d^* \geq 6g$. We show that the procedure actually holds for $d^* \geq 4g$. Also, a slight modification yields a faster procedure, while the constraint reduces to $d^* \geq 4g - 2\gamma$, where γ denotes the gonality of the curve. For $d^* < 4g - 2\gamma$, we give a family of examples where the lemmas do not apply.

Lemma 2.31 *For an error vector* $\mathbf{e}$ *with support* Q, *let the space* $K(F)$ *be as in Definition 2.9, and let*

$$l(F - Q) < k(F) < 2l(F - Q). \tag{2.10}$$

Let there exist a rational point P, *such that for* $F^* = F - P$,

$$k(F^*) \leq k(F) - 2. \tag{2.11}$$

Then

$$l(F^* - Q) \leq k(F^*) < 2l(F^* - Q). \tag{2.12}$$

Proof. The first inequality in (2.12) holds in general by Lemma 2.14 For the second inequality, we have

$$k(F^*) \leq k(F) - 2 < 2l(F - Q) - 2 \leq 2l(F^* - Q).$$

□

Remark 2.32 Let $\deg(F) < d^*$ or more general $\Omega(G-F) = 0$. In analogy with Lemma 2.17, we note that a vector $\mathbf{e} \in \mathbf{y} + C_\Omega(D,G)$ with (2.10) is unique in its coset. Indeed, let $\mathbf{e}_1 = \mathbf{e}$ and let $\mathbf{e}_2 \in \mathbf{e}_1 + C_\Omega(D,G)$ be a different vector with (2.10). Let their supports be denoted by the divisors $Q_1 = Q$ and Q_2 respectively. Let the support of $\mathbf{x} = \mathbf{e}_1 - \mathbf{e}_2 \in C_\Omega(D,G)$ be given by $Q_x \leq Q_1 + Q_2$. Then, $\mathbf{x} = (res_P(\omega))_{P \in D}$ for some nonzero $\omega \in \Omega(G - Q_x)$. Clearly $L(F-Q_1) \cap L(F-Q_2) \subset L(F-Q_x)$ and $L(F-Q_1) + L(F-Q_2) \subset K(F)$. Application of the bounds (2.10) yields

$$l(F-Q_1) + l(F-Q_2) - l(F-Q_x) < 2l(F-Q_1),$$
$$l(F-Q_1) + l(F-Q_2) - l(F-Q_x) < 2l(F-Q_2).$$

Thus $L(F-Q_x) \neq 0$ and, for $0 \neq f \in L(F-Q_x)$, we obtain the contradiction $0 \neq f\omega \in \Omega(G-F) = 0$.

Lemma 2.33 *Assume $L(F-Q) \neq 0$ and $deg(F) \leq d^* - g - 1$. Then precisely one of the following holds.*

$$K(F) = L(F-Q). \tag{2.13}$$

There exists a rational point $P \in Q$ with

$$k(F-P) \leq k(F) - 2. \tag{2.14}$$

Proof. Let (2.13) hold. With Lemma 2.14, for an arbitrary rational point P,

$$k(F-P) \geq l(F-P-Q) \geq l(F-Q) - 1 = k(F) - 1,$$

and (2.14) fails. Let (2.13) fail. In proving (2.14), we may assume that F has support disjoint from D by Remark 2.13. Then,

$$f \in K(F) \quad \Leftrightarrow \quad (e_P f(P))_{P \in Q} \in C_\Omega(Q, G-F).$$

For $f \in K(F) \backslash L(F-Q)$, let the set $\{P_1, P_2, \ldots, P_l\}$ denote the support of $(e_P f(P))_{P \in Q}$. Thus,

$$K(F) \cap L(F-P_i) \neq K(F), \qquad i = 1, 2, \ldots, l. \tag{2.15}$$

For some P_i, $i = 1, 2, \ldots, l$, let there exist

$$f_i \in L(F-Q) \backslash L(F-P_i-Q). \tag{2.16}$$

In particular, $f_i \in K(F) \cap L(F-P_i)$. Let $g \in L(G-F+P_i) \backslash L(G-F)$. Note that $\deg(F) < d^*$ by assumption, and g exists. We obtain $f_i g \in L(G+P_i-Q) \backslash L(G-Q)$ and $S(F-P_i)(f_i, g) = e_{P_i}(f_i g)(P_i) \neq 0$ and $f_i \notin K(F-P_i)$. We conclude that, provided f_i exists,

$$K(F-P_i) \neq K(F) \cap L(F-P_i). \tag{2.17}$$

Combination of (2.15),(2.17) and $P_i \in Q$ proves (2.14).

To prove that for some P_i a function f_i as in (2.16) exists, it suffices that the number of base points b of $|F-Q|$ is less than l. We have, $\Omega(G-F-P_1-\ldots-P_l) \neq 0$ and $\deg(G-F)-l \leq \deg(K)$ or $l \geq d^*-deg(F)$. Using $g \geq b$ as a bound for the number of base points, the claim follows with the assumption $\deg(F) < d^*-g$. □

Remark 2.34 The original proposition claims the existence of $P \in D$. At this stage of the basic algorithm, the divisor D plays no essential role by Lemma 2.26. The divisor Q does. At least when condition (2.10) holds and when $\Omega(G-F)=0$, by Remark 2.32. Also, in the proof, the points outside Q play no role. The difference is of no importance in proving the claim. The proof follows the original and yields the stronger claim, namely $P \in Q$. The bound on $\deg(F)$ is used twice in the proof. For later use, we note that the existence of the function g follows with the weaker condition $\Omega(G-F)=0$.

Remark 2.35 *([16])* We formulate the decoding procedure.

(E0) Fix a divisor F. Take $F^*=F$.
(E1) Find $P \in D$ with $k(F^*) \leq k(F)-2$, for $F^*=F-P$.
Repeat this step till no such P exists.
(E2) Apply the basic algorithm with F^*.

Proposition 2.36 *([16])* *Let a code $C_\Omega(G,D)$ be given with $d^* \geq 6g$. Let $\mathbf{e}$ be an error vector of weight at most t, for $2t < d^*$. Let F be a divisor of degree $2g+t$. The procedure of the remark corrects the error pattern $\mathbf{e}$.*

Proof. Let Q denote the support of $\mathbf{e}$. We establish condition (2.12), for $F^*=F$, and use Lemma 2.23. We have $\deg(G-F-Q) \geq 2g-2+d^*-2g-t-t > -2$, and therefore $k(F) \leq l(F-Q)+g$. Also, $\deg(F-Q) \geq 2g$, and it follows that $k(F) < 2l(F-Q)$. Next, we establish the condition $\deg(F) \leq d^*-g-1$ in Lemma 2.33. But $6g+2t \leq 2d^*-2$, and $2g+t \leq d^*-g-1$.
It is clear that a divisor $F=F^*$ with (2.12) satisfies either (2.13) or (2.11). The conditions of Lemma 2.33 are fulfilled. Thus, in the former case, the divisor is passed to step (E2). In the latter case, Lemma 2.31 applies and (2.12) holds for the divisor $F^*=F-P$. After finitely many repetitions, a divisor F^* with $K(F^*)=l(F^*-Q) \neq 0$ is passed to step (E2). The basic algorithm yields $f \in K(F^*)$ and with $\deg(F^*) < d^*$ the error vector is uniquely determined by the zeros of f. □

Remark 2.37 The choice $\deg(F) = 2g+t$ implies $\deg(G-F) = d^* - t-2$. By Remarks 2.32 and 2.34, the condition $\Omega(G-F) = 0$ should hold for the application of the Lemmas 2.31 and 2.33 respectively. In case $d^* \geq 4g$, the condition is satisfied, for any choice of F. In case $d^* \geq 2g$, the condition is satisfied for a proper choice of F. In case $d^* < 2g$, no proper choice exists. In the situation $d^* \leq 2g$ however, the choice $\deg(F) = g + 2t$ ensures that condition (2.10) will hold. The choice implies $\deg(G-F) = g-2+d^*-2t$ and $\Omega(G-F) = 0$ for a proper choice of F.

Remark 2.38 A closer look at the base points of $|F-Q|$ in Lemma 2.33 shows that the proposition in fact holds for $d^* \geq 4g$. Let $P_1, P_2, \ldots, P_b$ be b different base points of a non-trivial linear system $|E|$. Clifford's theorem yields,

$$\deg(E) + 1 - g \leq l(E) = l(E - P_1 - \ldots - P_b) \leq (\deg(E) - b)/2 + 1.$$

Or

$$b \leq 2g - \deg(E). \tag{2.18}$$

In the proof of the lemma, it suffices to prove the inequality

$$b < d^* - \deg(F),$$

where b denotes the number of base points of $|F-Q|$. It is trivially fulfilled for $b = 0$. For $b \neq 0$, we apply the bound (2.18), with $E = F - Q$. The inequality holds for $2g + t < d^*$ or $d^* \geq 4g$.

Remark 2.39 Another way to reduce the constraint on d^* to $d^* \geq 4g$ is obtained by using the procedure in parallel with $F = F_0$ and $F = G - F_0$. For at least one of $F_0, G - F_0$, condition (2.10) holds. Indeed, it suffices by Lemma 2.23 that one of the following holds:

$$\begin{aligned} l(F_0 - Q) + i(G - F_0 - Q) &\leq 2l(F_0 - Q), \\ l(G - F_0 - Q) + i(F_0 - Q) &\leq 2l(G - F_0 - Q). \end{aligned}$$

This follows with

$$\begin{aligned} & l(F_0 - Q) - i(G - F_0 - Q) + l(G - F_0 - Q) - i(F_0 - Q) \\ = \quad & \deg(G - 2Q) + 2 - 2g \;\geq\; d^* - 2t \;>\; 0. \end{aligned}$$

For F_0 of degree $g + t$, the conditions $\deg(F) < d^* - g$, for $F = F_0, G - F_0$, are fulfilled for $d^* > 2g + t$, or $d^* \geq 4g$. A further improvement follows by reconsidering the bound g for the number of base points. It is clear that in condition (2.10), the dimension $l(F - Q) > 1$. A bound on the number of base points of $|F - Q|$ is given by $\deg(F - Q) - b \geq \gamma$, where γ denotes the gonality. This implies that for the choice $\deg(F_0) = g + t$, the constraint $d^* \geq 4g - 2\gamma$ suffices. With this choice for F_0, $\deg(G - F) \geq 3g - 1 - \gamma$, for $F = F_0, G - F_0$. Avoiding the choices $F_0 = K, G - K$, in case the gonality $\gamma = g + 1$ ensures that $\Omega(G - F) = 0$.

Example 2.40 Recall Example 2.25. For a plane curve of degree four, a residue code is defined with $G = 4L$. Thus $d^* = 12 = 4g$, and by Remark 2.38, the procedure of Remark 2.35 corrects up to five errors for F in step (E0) of degree $2g+t = 11$. Also, by Remark 2.39, the procedure will be successful for $F = 2L$. In that case, the basic algorithm will in general be applied in step (E2) with $F = 2L$. Only when four error positions are collinear will F be modified in step (E1), and the basic algorithm will be applied with $F = 2L - P'$. Here, P' will be different from the noncollinear error position P.

Example 2.41 The constraint $d^* \geq 4g - 2\gamma$ is sharp. That is, for $d^* < 4g - 2\gamma$, no $P \in Q$ may exist with (2.14), while (2.13) does not hold. Thus, let $d^* = 4g - 2\gamma - 1$ and let $Q = Q_1$ and Q_2 contain the support of error vectors $\mathbf{e} = \mathbf{e}_1$ and $\mathbf{e}_2$ respectively, as in Notation 2.19, such that $\deg(Q_1) = 2g - \gamma - 1$ and $\deg(Q_2) = 2g - \gamma$. As in the proof of Proposition 2.36, we need a divisor F in step (E0), for which (2.12) holds. Following Remark 2.39, we use $F = F_0$ and $F = G - F_0$ in parallel, for a divisor F_0 of degree $g + t$. We give a special choice for Q_1, Q_2 and F_0, such that neither (2.13) nor (2.14) holds. To this end, let Q_γ, Q_{1*} and Q_{10} be divisors, such that

$$\begin{aligned} &\deg(Q_\gamma) = \gamma, && l(Q_\gamma) = 2. \\ &\deg(Q_{1*}) = g - \gamma, && l(Q_\gamma + Q_{1*}) = 2. \\ &\deg(Q_{10}) = g - 1, && l(Q_{10}) = 1. \end{aligned}$$

Let $Q_1 = Q_{1*} + Q_{10}$ and let $Q_2 \sim Q_\gamma + 2Q_{1*}$. For $F_0 \sim Q_\gamma + 2Q_{1*} + Q_{10}$, the divisor F_0 is of degree $g + t$ and (2.12) holds for $F^* = F_0$. In fact, $l(F_0 - Q_1) = l(Q_\gamma + Q_{1*}) = 2$, $l(F_0 - Q_2) = l(Q_{10}) = 1$ and $k(F_0) = 3$ by Lemma 2.24. But $P \in Q = Q_1$ is a basepoint of either $|F_0 - Q_1|$ or $|F_0 - Q_2|$ and no $P \in Q$ exists, such that $k(F_0 - P) \leq k(F_0) - 2$, while $K(F_0) \neq L(F_0 - Q)$.

Chapter 3

Modified algorithm

The modified algorithm (MA) [48] searches for error-locator functions of increasing degree and improves on the BA. The MA can be applied only to a restricted class of codes. The general case was left as an open problem [52, Remark 3.3.13]. In this chapter, we formulate the extended modified algorithm (EMA) that applies to all codes. The bound on error-correction of the MA shows a defect that depends on the particular code. The bound on error-correction of the EMA shows a defect that depends on the curve being used, rather than on the particular code.

3.1 Special divisors

We recall two well known results and we define a parameter that will be used later to measure a defect in a bound on error-correction. The parameter is investigated for hyperelliptic curves and for plane curves. For a curve $\mathcal{X}$, let K be a representative of the canonical divisor class.

Theorem 3.1 (Riemann-Roch) *For an arbitrary divisor E on the curve, we have*

$$\frac{deg(E)}{2} - (l(E) - 1) \quad = \quad \frac{deg(K - E)}{2} - (l(K - E) - 1). \tag{3.1}$$

Proof. See e.g. [7, Chapter 2],[23, Chapter 4]. □

Definition 3.2 A divisor E is called *special* if it is effective and $L(K - E) \neq 0$.

Theorem 3.3 (Clifford) *For a special divisor E on the curve, we have*

$$\frac{deg(E)}{2} - (l(E) - 1) \ \geq \ 0. \tag{3.2}$$

Proof. See e.g. [1, Chapter 3],[23, Chapter 4]. □

The curves considered in [23] are defined over an algebraically closed field. From this the result for finite fields follows. The proof in [1] for characteristic zero is similar for finite characteristic. Theorem 3.3 provides an upper bound for the dimension of a special divisor on a curve. To obtain a lower bound for the dimension of a special divisor we introduce the Clifford defect of a set of divisors. The defect measures the largest deviation from the upper bound.

Definition 3.4 For a curve $\mathcal{X}$, let $\mathcal{E}$ be a finite set of divisors. We define the *Clifford defect* $s(\mathcal{E})$ *of the set* $\mathcal{E}$ by $s(\emptyset) = 0$ and, for $\mathcal{E} \neq \emptyset$,

$$s(\mathcal{E}) \;=\; \max\{\, \frac{\deg(E)}{2} - (\, l(E) - 1\,) \;:\; E \in \mathcal{E}\,\}. \tag{3.3}$$

For our purpose we consider sets $\mathcal{E}$ of special divisors that satisfy

$$\mathcal{E} = \{E_0, E_1, \ldots, E_{2g-2}\}, \quad \deg(E_i) = i, \quad i = 0, 1, \ldots, 2g-2, \tag{3.4}$$

with g the genus of the curve (for $g = 0$ we set $\mathcal{E} = \emptyset$). One verifies that such a set exists if and only if the curve has a rational point. For a fixed $\mathcal{E}$, we write $s = s(\mathcal{E})$ and we define subsets $\mathcal{E}_0, \mathcal{E}_1 \subset \mathcal{E}$ by

$$\begin{aligned} E \in \mathcal{E}_0 \;&\Leftrightarrow\; \deg(E) \equiv 0 \pmod 2. \\ E \in \mathcal{E}_1 \;&\Leftrightarrow\; \deg(E) \equiv 1 \pmod 2. \end{aligned}$$

Also, let $s_0 = s(\mathcal{E}_0)$ and $s_1 = s(\mathcal{E}_1)$.

Lemma 3.5 *For a divisor E, let $s(E)$ denote $s(\{E\}) = deg(E)/2 - (l(E) - 1)$. A set $\mathcal{E}$ as in (3.4) can be modified without increase of its Clifford defect $s(\mathcal{E})$, such that it satisfies*

$$|s(E_i) - s(E_{i+1})| = 1/2, \quad i = 0, 1, \ldots, 2g-3. \tag{3.5}$$

Proof. Besides (3.5), we may distinguish two cases:

(1) $s(E_i) - s(E_{i+1}) > 1/2$, or $l(E_{i+1}) - l(E_i) > 1$.
(2) $s(E_{i+1}) - s(E_i) > 1/2$, or $l(K - E_i) - l(K - E_{i+1}) > 1$.

Let P be a rational point. As a modification in each case, we choose

(1) $E_i \sim E_{i+1} - P, \quad E_i \geq 0.$
(2) $E_{i+1} = E_i + P.$

With each modification the nonnegative number $s(E_0) + s(E_1) + \ldots + s(E_{2g-2})$ will strictly decrease. Regardless of the order of the modifications, after finitely many steps condition (3.5) will hold. □

Remark 3.6 For $\mathcal{E}$ as in (3.4), the maximum in (3.3) is taken over nonnegative values by Theorem 3.3. With Theorem 3.1 we have

$$0 \leq s \leq (g-1)/2, \quad \text{for } g \geq 1.$$

Also note that

$$s_0 \equiv 0 \pmod 1 \quad \text{and} \quad s_1 \equiv 1/2 \pmod 1.$$

A set $\mathcal{E}$ as in (3.4,3.5) satisfies $|s_0 - s_1| = 1/2$. We will later obtain bounds on error-correction, $t \leq (d_C - 1)/2 - s_0$ and $t \leq (d_C - 1)/2 - s_1$ respectively, for the cases of odd and even designed minimum distance respectively (Theorem 3.13).

Definition 3.7 We define the *Clifford defect* $s(\mathcal{X})$ *of the curve* $\mathcal{X}$ as

$$s(\mathcal{X}) = \min\{\, s(\mathcal{E}) : \mathcal{E} \text{ as in (3.4)} \,\}.$$

In particular for elliptic curves we obtain $s(\mathcal{X}) = 0$, with $\mathcal{E} = \{0\}$. From Remark 3.6 we conclude that decoding up to the designed minimum distance is also guaranteed for curves with $s(\mathcal{X}) = 1/2$ (i.e. $s_0 = 0$ and $s_1 = 1/2$).

Proposition 3.8 *The curves with* $s(\mathcal{X}) \leq 1/2$ *can be classified as the curves of genus zero or one and the hyperelliptic curves.*

Proof. Curves of genus $g \geq 2$ with this property have a divisor E, $E = E_2 \in \mathcal{E}$, that satisfies

$$\deg(E) = 2 \text{ and } l(E) = 2. \tag{3.6}$$

That is, the curve is hyperelliptic [23, p.298]. Conversely, for a hyperelliptic curve $\mathcal{X}$ the required set $\mathcal{E} = \mathcal{E}_0 \cup \mathcal{E}_1$ is defined by

$$\mathcal{E}_0 = \{0, E, \ldots, (g-1)E\},$$
$$\mathcal{E}_1 = \{P, E+P, \ldots, (g-2)E+P\},$$

with E as in (3.6) and P a rational point. □

Curves with $s(\mathcal{X}) = 1$ (i.e. $s_0 = 1$ and $s_1 = 1/2$) allow decoding up to the designed minimum distance d_C in case d_C is even. Among these are the plane curves of degree 4.

Proposition 3.9 *Let* $\mathcal{X}$ *be a plane curve of degree* m *with genus* g. *With the assumption that the curve has a rational point and* $m \geq 4$, *a set* $\mathcal{E}$ *can be chosen with*

$$s(\mathcal{E}) = \begin{cases} (m^2 - 4m + 8)/8, & \text{if } m \equiv 0 \pmod 2. \\ (m^2 - 4m + 7)/8, & \text{if } m \equiv 1 \pmod 2. \end{cases} \tag{3.7}$$

In particular $s(\mathcal{X}) \leq (g+1)/4$.

Proof. A construction for $\mathcal{E}$ is given in Section 3.5. With $g = (m-1)(m-2)/2$, we have $(m^2 - 4m + 8)/8 = (g+1)/4 - (m-4)/8$ and the inequality follows. □

3.2 Main lemma

The conditions (2.6,2.7) are conflicting: the former is satisfied for a divisor F of sufficiently low degree, the latter is satisfied for F of sufficiently high degree. The following lemma allows one to choose divisors F of increasing degree, such that for at least one F in the sequence both conditions are satisfied.

Lemma 3.10 (induction step) *Let G, F and Q be given divisors. Let E and F^* be divisors satisfying*

$$l(E) \geq g - deg(F - Q), \quad and \tag{3.8}$$

$$F^* \sim G - F - E. \tag{3.9}$$

Then

$$L(F - Q) = 0 \qquad \Rightarrow \qquad \Omega(G - F^* - Q) = 0.$$

Proof. We assume $\Omega(G - F^* - Q) \neq 0$, say it contains $\omega \neq 0$, and will deduce $L(F - Q) \neq 0$. So let

$$(\omega) = G - F^* - Q + E^*, \qquad E^* \geq 0.$$

Using the definition of F^* (3.9), we obtain

$$\begin{aligned} (\omega) &\sim F + E - Q + E^*, \\ F - Q &\sim K - E - E^*, \end{aligned} \tag{3.10}$$

where K represents the canonical divisor class. Since $E^* \geq 0$, it suffices for $L(F - Q) \neq 0$, to prove

$$\deg(E^*) < l(K - E). \tag{3.11}$$

Substitution of (3.10) in (3.8) yields

$$\begin{aligned} &(g-1) - \deg(K - E - E^*) < l(E), \\ &\deg(E^*) < l(E) - \deg(E) + g - 1, \\ &\deg(E^*) < l(E) - \deg(E)/2 + \deg(K - E)/2, \\ &\deg(E^*) < l(K - E), \end{aligned}$$

where we apply Theorem 3.1 (Riemann-Roch). Thus, we have proven (3.11). □

For a residue code C, we make the bound on error-correction that is contained in (3.8) explicit.

Lemma 3.11 (a bound on error-correction) *Let $C = C_\Omega(D,G)$ be a residue code. It has designed minimum distance $d_C = deg(G-K)$. Also, let $t = deg(Q)$. Then, condition (3.8) in Lemma 3.10 is equivalent to*

$$t \leq \frac{d_C - 1}{2} + (l(E) - 1) - \frac{deg(E)}{2} - \frac{deg(F^* - F) - 1}{2}. \qquad (3.12)$$

Proof. Note that (3.9) implies that the right hand side of (3.12) is a natural number. We will obtain (3.8) from (3.12). To this end, we first multiply and then substitute for the parameters t and d_C,

$$\begin{aligned} 2t &\leq (d_C - 1) + 2(l(E) - 1) - \deg(E) - \deg(F^* - F) + 1, \\ 2\deg(Q) &\leq \deg(G - K) + 2(l(E) - 1) - \deg(E) - \deg(F^* - F). \end{aligned}$$

Next, we use (3.9),

$$\begin{aligned} 2\deg(Q) &\leq 2(l(E) - 1) + \deg(G - K - E) - \deg(G - 2F - E), \\ \deg(K) - 2\deg(F - Q) &\leq 2(l(E) - 1), \\ (g - 1) - \deg(F - Q) &\leq (l(E) - 1). \end{aligned}$$

This clearly yields (3.8). We have equivalence at all steps. □

3.3 Description

Proposition 3.12 (modified algorithm [48]) *Let $C = C_\Omega(D,G)$ be a residue code, with $G = aH$, H an effective divisor, and $h = deg(H)$. Let*

$$s(H) = max\{ \frac{ih + h + 1}{2} - l(iH) : i \in \mathbf{Z} \}.$$

Then, a received word with error pattern of weight t, $t \leq (d_C - 1)/2 - s(H)$, can be corrected by successive applications of the BA with $F = H, 2H, \ldots$. In particular $F = bH$ of lowest degree such that (2.7) holds will satisfy (2.6).

Proof. A slightly improved bound is given in Proposition 3.17. □

Theorem 3.13 (extended modified algorithm) *Let $C = C_\Omega(D,G)$ be a residue code, defined with a curve $\mathcal{X}$, with odd designed minimum distance $d_C = 2e+1$. Let $\mathcal{E}$ be a set of special divisors on $\mathcal{X}$ as in Definition 3.4 and let $\mathcal{E}_0$ be the subset of divisors of even degree, say $\mathcal{E}_0 = \{E_0, E_1, \ldots, E_{g-1}\}$ and $s_0 = s(\mathcal{E}_0)$:*

$$deg(E_i) = 2g - 2 - 2i, \quad deg(E_i)/2 - (l(E_i) - 1) \leq s_0.$$

Also, let $\mathcal{F} = \{F_0, F_1, \ldots, F_g\}$ *be a set of divisors on* $\mathcal{X}$*, with* $deg(F_0) = e$*,* $F_0 \cap D = \emptyset$ *and*

$$F_i \sim G - F_{i-1} - E_{i-1}, \qquad F_i \cap D = \emptyset, \qquad i = 1, 2, \ldots, g. \tag{3.13}$$

Then, a received word with error pattern of weight t*,* $t \leq (d_C - 1)/2 - s_0$*, can be corrected by successive applications of the BA with* $F = F_0, F_1, \ldots, F_g$*. In particular* $F = F_i$ *of lowest degree such that (2.7) holds will satisfy (2.6).*

Proof. With (3.13) we have, for $i = 2, 3, \ldots, g$,

$$\deg(F_i + E_{i-1}) = \deg(G - F_{i-1}) = \deg(F_{i-2} + E_{i-2}),$$
$$\deg(F_i - F_{i-2}) = \deg(E_{i-2} - E_{i-1}) = 2.$$

With $\deg(F_0) = e$ and $\deg(F_1) = e + 1$ we obtain

$$\deg(F_i) = e + i, \quad i = 0, 1, \ldots, g.$$

Let Q denote the divisor of all the error locations. For F_0 we have, with $\deg(Q) = t \leq e - s_0$ and $s_0 \geq 0$,

$$\deg(G - F_0 - Q) \geq \deg(K) + 2e + 1 - e - (e - s_0) \geq \deg(K) + 1.$$

And thus $\Omega(G - F_0 - Q) = 0$. Next, we prove that, for $i = 0, 1, \ldots, g - 1$,

$$L(F_i - Q) = 0 \quad \Rightarrow \quad \Omega(G - F_{i+1} - Q) = 0. \tag{3.14}$$

We use Lemma 3.10. All conditions but (3.8) are trivially fulfilled. We have

$$\deg(F_{i+1} - F_i) = 1.$$
$$\deg(E_i)/2 - (l(E_i) - 1) \leq s_0.$$
$$\deg(Q) \leq e - s_0.$$

We may apply Lemma 3.11. This proves the induction (3.14). For the termination of the algorithm we have for F_g

$$\deg(F_g - Q) \geq (e + g) - (e - s_0) \geq g.$$

Hence $L(F_g - Q) \neq 0$. □

Remark 3.14 (even designed minimum distance) Let $C = C_\Omega(D, G)$ have even designed minimum distance $d_C = 2e + 2$. The following modifications apply to Theorem 3.13: We consider the subset $\mathcal{E}_1$ of $\mathcal{E}$ of divisors of odd degree, say $\mathcal{E}_1 = \{E_1, E_2, \ldots, E_{g-1}\}$ and $s_1 = s(\mathcal{E}_1)$:

$$\deg(E_i) = 2g - 1 - 2i, \quad \deg(E_i)/2 - (l(E_i) - 1) \leq s_1.$$

Also, let $\mathcal{F} = \{F_1, F_2, \ldots, F_g\}$, with $\deg(F_1) = e + 1$, $F_1 \cap D = \emptyset$ and (3.13) for $i = 2, 3, \ldots, g$. Then, a received word with error pattern of weight t, $t \leq (d_C - 1)/2 - s_1$, can be corrected by successive applications of the BA with $F = F_1, F_2, \ldots, F_g$. The proof follows Theorem 3.13.

Example 3.15 Recall Example 2.25. The code $C_\Omega(D,G)$, with $G = 4L$, has distance $d^* = 12$ and the basic algorithm corrects any $(d^* - 1 - g)/2 = 4$ errors. Let P denote a rational point on the curve. The set of special divisors $\mathcal{E}_1 = \{E_1 = L - P, E_2 = P\}$ has Clifford defect $s_1 = 1/2$. The set $\mathcal{F}$ of divisors F is defined inductively, starting with F of degree 6 and using $F^* = G - F - E$:

$$\begin{aligned} F &= L + 2P, & F^* &= 2L - P, & E &= L - P, \\ F &= 2L - P, & F^* &= 2L, & E &= P, \end{aligned}$$

Thus the basic algorithm can be applied with $F \in \mathcal{F}$,

$$\mathcal{F} = \{F_1 = L + 2P, F_2 = 2L - P, F_3 = 2L\}.$$

When applied in this order, the first solution occurring will be error-locating.

Proposition 3.16 *We have the following bounds for error correction with the EMA. Rational, elliptic and hyperelliptic curves:*

$$t \leq \lfloor (d_C - 1)/2 \rfloor.$$

Plane curves of degree m:

$$t \leq \lfloor (d_C - 1)/2 - (m-1)(m-3)/8 \rfloor.$$

Proof. For the first bound we use the set $\mathcal{E}$ as in Proposition 3.8. For the second bound we refer to Section 3.5. □

Proposition 3.17 *We claim that Proposition 3.12 holds with the following: H may be any divisor and $s(H)$ can be defined as*

$$s(H) \;=\; max\{\, \frac{ih + h + 1}{2} - l(iH) \;:\; i \not\equiv a \pmod 2 \,\}.$$

Proof. The improvement is obtained through a different uniqueness proof (Remark 2.21). Note that the bound on error-correction, $t \leq (d_C - 1)/2 - s(H)$, always represents an integer. We use the line of proof in Theorem 3.13. Clearly, we may add $F = 0$ at the beginning of the sequence. The following are trivial: $\Omega(G - 0 - Q) = 0$ and $L(aH - Q) \neq 0$. For the analogue to the induction (3.14), we will apply Lemma 3.10 with $F^* = F + H$. Then $E = G - F - F^*$ runs through

$$E \;=\; (a-1)H, (a-3)H, \ldots .$$

And, for $E = iH$,

$$\deg(E)/2 - (l(E) - 1) \;=\; (ih)/2 - (l(iH) - 1) \;\leq\; s(H) - (h-1)/2.$$

Thus we have, at all steps,

$$\begin{aligned}
&\deg(F^* - F) = h, \quad \text{and} \\
&\deg(E)/2 - (l(E) - 1) \leq s(H) - (h-1)/2, \quad \text{and} \\
&\deg(Q) \leq e - s(H).
\end{aligned}$$

We may apply Lemma 3.11. This proves the induction. □

3.4 Additional lemma

After Lemma 3.10, we present another lemma that allows one to choose divisors F of increasing degree, such that for at least one F in the sequence both conditions (2.6,2.7) are satisfied. Unlike Lemma 3.10, the application of Lemma 3.18 shows some restrictions, see Remark 3.19.

Lemma 3.18 (alternate step) *Let G, F and Q be given divisors. With P be a rational point on the curve, let $F^* = F + P$ and $E = G - F - F^*$. Then $l(E) = l(E - P) + 1$ implies*

$$L(F^* - Q) \neq L(F - Q) \qquad \Rightarrow \qquad \Omega(G - F^* - Q) = \Omega(G - F - Q).$$

Proof. By assumption, there exist a function $f \neq 0$ with

$$(f) = E_1 - E, \qquad E_1 \geq 0, \text{ but not } E_1^* \geq P.$$

We assume $\Omega(G - F^* - Q) \neq \Omega(G - F - Q)$ and will deduce $L(F^* - Q) = L(F - Q)$. Thus, let $\omega \neq 0$ be a differential with divisor

$$(\omega) = G - F^* - Q + E_2, \qquad E_2 \geq 0, \text{ but not } E_2 \geq P.$$

Then $f\omega \in \Omega(F - Q) \backslash \Omega(F^* - Q)$, and $L(F^* - Q) = L(F - Q)$. □

Remark 3.19 (a note on application) The lemma does not apply to two consecutive steps: in case $L(F^* - Q) = L(F - Q) = 0$, no information on $\Omega(G - F^* - Q)$ is obtained and in a next step $\Omega(G - F^* - Q) = \Omega(G - F - Q)$ provides no useful information. It is clear from the proof of Theorem 3.13 that, for a given set of special divisor $\mathcal{E}$, only some induction steps (3.14) determine the bound on error-correction. If one succeeds in replacing these steps by alternate steps, the bound may be improved by one.

Lemma 3.20 *Replacement of Lemma 3.10 with Lemma 3.18 in proving induction for the EMA (Theorem 3.13) yields an improvement only if*

$$s = \max\{s_0, s_1\} = \begin{cases} s_0, & \text{if } d_C \text{ is odd.} \\ s_1, & \text{if } d_C \text{ is even.} \end{cases}$$

For plane curves of degree m the condition becomes

$$\begin{array}{ll} m \equiv 0 \pmod 4 & \text{if } d_C \text{ is odd.} \\ m \not\equiv 0 \pmod 4 & \text{if } d_C \text{ is even.} \end{array}$$

Proof. We consider the case of odd d_C and we may assume that $\mathcal{E}$ satisfies (3.5). With the notation as in (3.4) and Lemma 3.5, let $s(E_{2r}) = s_0$. We have improvement, only if $s(E_{2r-2}) = s(E_{2r+2}) = s_0 - 1$. Then by (3.5): $s(E_{2r-1}) = s(E_{2r+1}) = s_0 - 1/2$. Hence $s_1 < s_0$. The statement on plane curves is now a consequence of Proposition 3.9. □

Example 3.21 Let $\mathcal{X}$ be the Hermitian curve of degree five over $GF(16)$. Let P_0 and P_1 be two rational points. Their tangents have intersection divisor $L_0 = 5P_0$ and $L_1 = 5P_1$ respectively. Let $G = 2a(P_0 + P_1)$ be a divisor. The code $C_\Omega(D, G)$ is of type $[63, 68 - 4a, \geq 4a - 10]$ over $GF(16)$, for $3 \leq a \leq 15$. Proposition 3.16 tells that the EMA corrects $e - 1$ errors (one less than the designed capability). This is obtained with a set of special divisors of Clifford defect one and a half. We may take $\mathcal{E} = \{9P_0, 7P_0, 5P_0, 3P_0, P_0\}$. Only the divisors $7P_0$ and $3P_0$ have a defect of one and a half. We avoid using Lemma 3.10 with these divisors and use Lemma 3.18 instead. The condition $l(E) = l(E - P) + 1$ in the lemma is satisfied for

$$\begin{array}{ll} E = 4P_0 + 3P_1, & P = P_1. \\ E = 5P_0 - 2P_1, & P = P_0. \end{array}$$

For the three remaining induction steps, we may use Lemma 3.10 with $E = 5P_0 + 4P_1, 5P_0, P_0$. Let $\overline{F} = a(P_0 + P_1)$.

$$\begin{array}{lll} F = \overline{F} - 3P_0 - 2P_1, & F^* = \overline{F} - 2P_0 - 2P_1, & E = 5P_0 + 4P_1, \\ F = \overline{F} - 2P_0 - 2P_1, & F^* = \overline{F} - 2P_0 - P_1, & E = 4P_0 + 3P_1, \\ F = \overline{F} - 2P_0 - P_1, & F^* = \overline{F} - 3P_0 + P_1, & E = 5P_0, \\ F = \overline{F} - 3P_0 + P_1, & F^* = \overline{F} - 2P_0 + P_1, & E = 5P_0 - 2P_1, \\ F = \overline{F} - 2P_0 + P_1, & F^* = \overline{F} + P_0 - P_1, & E = P_0. \end{array}$$

Thus the basic algorithm can be applied with $F \in \mathcal{F}$,

$$\begin{array}{rcl} \mathcal{F} & = & \{ \ \overline{F} - 3P_0 - 2P_1, \ \overline{F} - 2P_0 - 2P_1, \ \overline{F} - 2P_0 - P_1, \\ & & \quad \overline{F} - 3P_0 + P_1, \ \overline{F} - 2P_0 + P_1, \ \overline{F} + P_0 - P_1 \ \}. \end{array}$$

When applied in this order, the first solution occurring will be error-locating.

Proof. (Proposition 3.9) We consider the plane curve $\mathcal{X}$ of degree m and genus $g = (m-1)(m-2)/2$. With the assumption $m \geq 4$, we will construct a set $\mathcal{E}$ with

$$s(\mathcal{E}) = \begin{cases} (m^2 - 4m + 8)/8 & \text{if } m \equiv 0 \pmod 2. \\ (m^2 - 4m + 7)/8 & \text{if } m \equiv 1 \pmod 2. \end{cases} \tag{3.15}$$

Let thus L be an intersection divisor of the curve with a line. Also, let B_i be an effective divisor of degree $\deg(B_i) = i$, for $i = 0, 1, \ldots, m-1$. We define the set $\mathcal{E}$ as the set of divisors E, that satisfy one of the following

$$\begin{array}{ll} E = aL + B_i\,, & 0 \leq a < m-3 \text{ and } 0 \leq i \leq m-(a+2). \\ \text{Or } E = (a+1)L - B_{m-i}\,, & 0 \leq a < m-3 \text{ and } m-(a+2) < i < m. \\ \text{Or } E = (m-3)L. & \end{array}$$

One verifies that $\mathcal{E}$ contains divisors E, with degree $\deg(E)$ in the range: $0 \leq \deg(E) \leq (m-3)m = 2g-2$. To each degree in the range there corresponds a unique divisor. With $m \geq 4$ we note: in the calculation of the maximum $s(\mathcal{E})$, we may consider $E \neq (m-3)L$. We have $l(aL) = (a+2)(a+1)/2$, and thus for $l(E)$

$$\begin{aligned} l(aL + B_i) &\geq (a+2)(a+1)/2, \quad \text{and} \\ l((a+1)L - B_{m-i}) &\geq (a+3)(a+2)/2 - (m-i). \end{aligned}$$

For $\deg(E)/2 - (l(E)-1)$ we obtain, with $i = m-(a+2) \mp b, b \geq 0$,

$$\frac{\deg(E)}{2} - (l(E)-1) \leq \frac{-a^2 + (m-4)a + m - 2 - b}{2}. \tag{3.16}$$

The maximum of the right hand side is obtained for $a = (m-4)/2, b = 0$. Looking for integer solutions, we see that the maxima are obtained for

$$(a,b) = \begin{cases} ((m-4)/2\,,\,0), & \text{if } m \equiv 0 \pmod 2. \\ ((m-3)/2\,,\,0), & \text{if } m \equiv 1 \pmod 2. \end{cases}$$

Substitution in (3.16) yields (3.15). □

Proof. (Proposition 3.16) From (3.15) we obtain the inequality

$$(m-1)(m-3)/8 \leq s - 1/2.$$

The gap is at most $1/8$. Also $s - 1/2 \leq s_0$ and $s - 1/2 \leq s_1$, where one inequality is tight and the other shows a gap of $1/2$. For the cases of odd and even designed minimum distance the following represent tight integer bounds for decoding with the EMA (Theorem 3.13)

$$\begin{aligned} t &\leq (d_C - 1)/2 - s_0, \quad \text{for } d_C \text{ odd.} \\ t &\leq (d_C - 1)/2 - s_1, \quad \text{for } d_C \text{ even.} \end{aligned}$$

Hence one verifies by combination of the inequalities that in both cases

$$t \leq (d_C - 1)/2 - (m-1)(m-3)/8,$$

where the gap is at most $5/8$. □

Chapter 4

Majority coset decoding

Justesen et al. [24] observed that the Peterson algorithm could be generalized to codes from plane algebraic curves. As was shown by Skorobogatov and Vlăduţ [48], the restriction to plane curves and to a particular class of codes was not essential. Thus a basic algorithm for the decoding of an arbitrary algebraic geometric code is available. It corrects up to $(d^* - 1)/2 - g/2$ errors. Improvements of this result either are not constructive and of high complexity [38],[53] or do not correct up to the designed minimum distance in general [48],[9]. Both Ehrhard [16] and Feng and Rao [17] formulated procedures that overcome this. Ehrhard's procedure is recalled in Section 2.5.

In this chapter, we show that the procedure of Feng and Rao can be applied to an arbitrary algebraic geometric code. In our set up a reduction step is added to the basic algorithm. In case the basic algorith fails a majority scheme is used to obtain an additional syndrome for the error vector. Thus a strictly smaller coset containing the error vector is obtained. In this way the basic algorithm is applied to a decreasing chain of cosets and after finitely many steps the coset will be small enough for succesful application of the basic algorithm. We call the procedure Majority Coset Decoding (in short MCD).

The repeated applications of the basic algorithm in the procedure can be carried out with one common set of data. For this, Feng and Rao [17] presented a scheme for the computations. The scheme is referred to as Modified Fundamental Iterative Algorithm (in short MFIA). In the last section we point out that also in the general case the computations can be carried out with the MFIA. Thus the decoding procedure has the complexity of the basic algorithm, that is $O(t^2 n + g^2 n)$.

Although based on the same idea, our procedure does not compare immediately with [17] when applied to the the same codes. Our formulation involves square matrices of size $t + g$, whereas in [17] matrices of size $2t + g$ are used. In Section 4.5, we present our procedure in the set up of [17], as to establish the relation.

4.1 Coset decoding

In the following $C_1 = C_\Omega(D, G_1) \subset \mathbf{F}_q^n$ is a fixed residue code [30], [52]. Thus $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$ is a set of n rational points on a non-singular curve $\mathcal{X}/\mathbf{F}_q$, $D = P_1 + P_2 + \ldots + P_n$ and G_1 is a divisor defined over $\mathbf{F}_q$ with support disjoint from $\mathcal{P}$. The codewords of C_1 are of the form $(\mathrm{res}_{P_1}(\eta), \mathrm{res}_{P_2}(\eta), \ldots, \mathrm{res}_{P_n}(\eta))$, for $\eta \in \Omega(G_1 - D)$. With g the genus of the curve $\mathcal{X}$, the code C_1 has designed distance $d_1^* = \deg(G_1) - (2g-2)$. The dual code $C_1^\perp$ is equal to the functional code $C_L(D, G_1)$ with codewords of the form $(h(P_1), h(P_2), \ldots, h(P_n))$, for $h \in L(G_1)$.

For a rational point $P_\infty \notin \mathcal{P}$, let $G_0 = G_1 - P_\infty$ and let $G_2 = G_1 + P_\infty$. We have an extension of residue codes

$$C_0 = C_\Omega(D, G_0) \supset C_1 = C_\Omega(D, G_1) \supset C_2 = C_\Omega(D, G_2). \tag{4.1}$$

Let $\mathbf{e}$ be a vector with

$$\mathrm{wt}(\mathbf{e}) \leq (d_1^* - 1)/2. \tag{4.2}$$

We formulate a *coset decoding procedure with respect to the extension of codes* $C_1 \supset C_2$: for a given $\mathbf{y}_1 \in \mathbf{e} + C_1$ we show how to obtain $\mathbf{y}_2 \in \mathbf{e} + C_2$. Note that with condition (4.2) this is well-defined. In case $\mathbf{y}_2 \neq \mathbf{e}$ the procedure can be repeated, till eventually the error vector is obtained. With a combination of the procedure and known algorithms the number of repetitions required can be bounded by the genus g of the curve (Remark 4.16). Clearly, we may assume that

$$C_1 \neq C_2, \quad \text{and} \quad L(G_2) \neq L(G_1),$$

where the second assumption follows from the first. As with decoding of linear codes in general, syndromes are crucial. We define syndromes as linear maps to the constant field.

Definition 4.1 With a vector $\mathbf{y} = (y_1, y_2, \ldots, y_n) \in \mathbf{F}_q^n$ we associate a *one-dimensional syndrome* $S_i(\mathbf{y})$, for $i = 0, 1, 2$,

$$\begin{aligned} S_i(\mathbf{y}) : \quad L(G_i) &\longrightarrow \mathbf{F}_q, \\ h &\longmapsto \sum_{j=1}^{n} y_j\, h(P_j). \end{aligned}$$

Lemma 4.2 *The one-dimensional syndrome is a coset invariant, that is*

$$S_i(\mathbf{y}) = S_i(\mathbf{e}) \quad \Leftrightarrow \quad \mathbf{y} \in \mathbf{e} + C_i, \qquad i = 0, 1, 2.$$

Proof. From the definition

$$S_i(\mathbf{y} - \mathbf{e}) = 0 \;\Leftrightarrow\; \mathbf{y} - \mathbf{e} \in C_L(D, G_i)^\perp.$$

The result follows with $C_L(D, G_i) = C_i^\perp$. □

Lemma 4.3 *For* $\mathbf{u} \in C_1 \setminus C_2$ *and for* $h \in L(G_2) \setminus L(G_1)$ *we have* $S_2(\mathbf{u})(h) \neq 0$.

Proof. $G_2 = G_1 + P_\infty$ implies $l(G_2) - l(G_1) \leq 1$ and $L(G_2) = L(G_1) \oplus \langle h \rangle$. With the lemma we have

$$\mathbf{u} \in C_1 \wedge \mathbf{u} \notin C_2 \quad \Rightarrow \quad S_1(\mathbf{u}) = 0 \wedge S_2(\mathbf{u}) \neq 0 \quad \Rightarrow \quad S_2(\mathbf{u})(h) \neq 0.$$

□

The following is immediate from the lemmas.

Corollary 4.4 *Let* $\mathbf{u} \in C_1 \setminus C_2$ *and let* $\mathbf{y}_1 \in \mathbf{e} + C_1$. *There exists a unique* λ *such that* $\mathbf{y}_1 - \lambda \mathbf{u} \in \mathbf{e} + C_2$. *For* $h \in L(G_2)$ *it satisfies*

$$\lambda S_2(\mathbf{u})(h) = S_2(\mathbf{y}_1)(h) - S_2(\mathbf{e})(h),$$

where $S_2(\mathbf{u})(h) \neq 0$, *for* $h \in L(G_2) \setminus L(G_1)$.

By the corollary it suffices for coset decoding to find a function $h \in L(G_2) \setminus L(G_1)$ with $S_2(\mathbf{e})(h) = 0$. We give a procedure to obtain functions f, g such that $h = fg$ will do.

4.2 Two-dimensional syndromes

Definition 4.5 For a vector $\mathbf{e}$ with (4.2) we consider the cosets $\mathbf{e} + C_i$ with syndromes $S_i(\mathbf{e})$, for $i = 0, 1, 2$. With a divisor F, that has support disjoint from $\mathcal{P}$, we associate a *two-dimensional syndrome* $S_i(F)$, for $i = 0, 1, 2$,

$$\begin{aligned} S_i(F) \; : \quad & L(F) \times L(G_i - F) \longrightarrow \mathbf{F}_q, \\ & (f, g) \longmapsto S_i(\mathbf{e})(fg). \end{aligned}$$

Let $K_i(F)$ be defined as the subspace of $L(F)$ with

$$f \in K_i(F) \;\Leftrightarrow\; \forall g \in L(G_i - F) : \; S_i(F)(f, g) = 0.$$

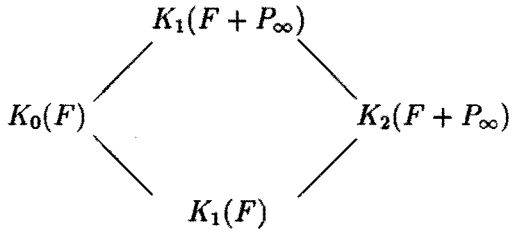
Lemma 4.6 *The containments*

$$\begin{aligned} & K_1(F + P_\infty) \supset K_0(F), \qquad && K_1(F + P_\infty) \supset K_2(F + P_\infty), \\ & K_0(F) \supset K_1(F), \text{ and} && K_2(F + P_\infty) \supset K_1(F) \end{aligned}$$

define factor spaces of dimension at most one.

Proof. The inclusion relations are immediate from the definition. In particular $K_0(F) = K_1(F + P_\infty) \cap L(F)$. Since $K_1(F + P_\infty) \subset L(F + P_\infty)$, the intersection reduces the dimension by at most one. Similar for $K_2(F + P_\infty) \supset K_1(F)$. For the inclusion $K_0(F) \supset K_1(F)$ we observe that $K_1(F)$ is obtained by applying at most one linear condition to $K_0(F)$. Similar for $K_1(F + P_\infty) \supset K_2(F + P_\infty)$. □

The inclusion relations may be depicted as follows



As the next lemma explains, we are interested in the situation where the following conditions are satisfied

$$\begin{array}{ll} (A1) & K_1(F+P_\infty) \neq K_0(F), \\ (A2) & K_0(F) = K_1(F), \\ (A3) & L(G_1 - F) \neq L(G_1 - F - P_\infty). \end{array}$$

$$\begin{array}{ll} (B1) & K_1(F+P_\infty) = K_2(F+P_\infty), \\ (B2) & K_2(F+P_\infty) \neq K_1(F). \end{array}$$

Let $(A) \Leftrightarrow (A1) \wedge (A2) \wedge (A3)$. Also let $(B) \Leftrightarrow (B1) \wedge (B2)$.

Corollary 4.7 *The conditions satisfy*

$$(A1) \wedge (B1) \quad \Leftrightarrow \quad (A2) \wedge (B2).$$

Proof. Immediate from Lemma 4.6. □

Remark 4.8 The vector spaces $K(F)$ that occur in the conditions (A) and (B) are all defined as left null spaces of bilinear forms $S(F)$. The duality in the formulation becomes more obvious if we consider the right null spaces as well. For example, we have

$$(A2) \wedge (A3) \quad \Leftrightarrow \quad K_1(G_1 - F) \neq K_0(G_1 - F - P_\infty).$$

However, to save on computations, the formulation in terms of the left null spaces only is to be preferred.

Lemma 4.9 (main lemma) *Let the functions f, g satisfy*

$$\begin{array}{l} f \in K_1(F+P_\infty) \setminus K_0(F), \quad \textit{and} \\ g \in L(G_1 - F) \setminus L(G_1 - F - P_\infty). \end{array}$$

Then

$$fg \in L(G_2) \setminus L(G_1),$$

and

$$K_1(F+P_\infty) = K_2(F+P_\infty) \quad \Rightarrow \quad S_2(\mathbf{e})(fg) = 0.$$

Proof. From $f \in L(F + P_\infty)$ and $g \in L(G_1 - F)$ we obtain $fg \in L(G_1 + P_\infty) = L(G_2)$. For $fg \notin L(G_1)$ it suffices to consider the pole order $-\nu_\infty(fg)$ at P_∞ and

$$\begin{aligned} -\nu_\infty(fg) &= -\nu_\infty(f) - \nu_\infty(g) \\ &= \mathrm{ord}_{P_\infty}(F + P_\infty) + \mathrm{ord}_{P_\infty}(G_1 - F) \\ &> \mathrm{ord}_{P_\infty}(G_1). \end{aligned}$$

Note that $g \in L(G_2 - F - P_\infty)$. With $f \in K_2(F + P_\infty)$ we have

$$S_2(\mathbf{e})(fg) = S_2(F + P_\infty)(f, g) = 0.$$

□

By the remark following Corollary 4.4 we can choose $h = fg$ with f, g as in the lemma, provided that conditions (A1),(A3) and (B1) hold. With the corollary the conditions (A) and (B) need to be fulfilled. In the decoding situation we cannot determine $K_2(F + P_\infty)$ and we are unable to verify (B). To overcome this we use a majority scheme.

4.3 A majority scheme

Remark 4.10 Let the notation be as in Definition 4.5. Let the divisor $P_\mathbf{e} = \sum_{e_j \neq 0} P_j$. With Definition 4.1 and Definition 4.5 we see, for $i = 0, 1, 2$,

$$f \in L(F - P_\mathbf{e}) \ \vee \ g \in L(G_i - F - P_\mathbf{e}) \ \Rightarrow \ S_i(F)(f, g) = 0.$$

Lemma 4.11 *Consider the conditions*

$$\begin{aligned} &(C1) \quad L(F + P_\infty - P_\mathbf{e}) \neq L(F - P_\mathbf{e}), \\ &(C2) \quad L(G_1 - F - P_\mathbf{e}) \neq L(G_1 - F - P_\infty - P_\mathbf{e}). \end{aligned}$$

With the conditions (A1)-(A3) and (B1)-(B2) as in the previous section the following holds

$$\begin{aligned} (C1) \quad &\Rightarrow \quad (A1) \wedge (B2), \\ (C2) \quad &\Rightarrow \quad (A2) \wedge (B1) \wedge (A3). \end{aligned}$$

Proof. Let (C1) hold. For (A1) and (B2) respectively it suffices to give functions $f_1 \in K_1(F + P_\infty) \setminus L(F)$ and $f_2 \in K_2(F + P_\infty) \setminus L(F)$ respectively. With the remark a function $f \in L(F + P_\infty - P_\mathbf{e}) \setminus L(F - P_\mathbf{e})$ will do in both cases. Let (C2) hold. Recall that $G_0 = G_1 - P_\infty$. We may write $L(G_1 - F) = L(G_0 - F) + L(G_1 - F - P_\mathbf{e})$. With the remark we obtain $K_0(F) \subset K_1(F)$. This proves (A2). For (B1) we use $G_1 = G_2 - P_\infty$ and we write $L(G_2 - F - P_\infty) = L(G_1 - F - P_\infty) + L(G_2 - F - P_\infty - P_\mathbf{e})$ to obtain $K_1(F + P_\infty) \subset K_2(F + P_\infty)$. The condition (A3) follows immediately with $P_\mathbf{e} \geq 0$. □

For the application of the lemma, recall the coset decoding procedure: For the residue codes $C_1 = C_\Omega(G_1, D) \supset C_2 = C_\Omega(G_1 + P_\infty, D)$, let $\mathbf{y}_1 \in \mathbf{e} + C_1$ be a received word. To obtain $\mathbf{y}_2 \in \mathbf{e} + C_2$ with Lemma 4.4, we need a function h. The function h can be obtained with Lemma 4.9, provided that conditions (A) and (B) hold for a divisor F. By Lemma 4.11, the conditions (C1) and (C2) ensure that (A) and (B) will hold.

Lemma 4.12 *Let $deg(G_1) = 2g + 2t - 1$ and let $deg(P_\mathbf{e}) \leq t$. Let the main lemma be applied with divisors F in the range $t - 1 \leq deg(F) \leq t + 2g - 1$, starting with a divisor of degree $t - 1$ and adding a rational point P_∞ each time. Among the divisors F that satisfy condition (A), the divisors that satisfy condition (B) as well form a majority.*

Proof. For the $2g + 1$ divisors F in the range, the conditions (C1) and (C2) will each hold at least $g + 1$ times and fail at most g times. For at least one divisor they both hold. Moreover, the cases that (C1) and (C2) both hold outnumber the cases that (C1) and (C2) both fail. This suffices for the claim, since by Corollary 4.7 and Lemma 4.11,

$$\begin{array}{rcl} (\mathrm{C1}) \wedge (\mathrm{C2}) & \Rightarrow & (\mathrm{A}) \wedge (\mathrm{B}), \\ (\mathrm{A}) \wedge \neg\,(\mathrm{B}) & \Rightarrow & \neg\,(\mathrm{C1}) \wedge \neg\,(\mathrm{C2}). \end{array}$$

□

Repeating the coset decoding procedure will yield a series of vectors $\mathbf{y}_2$, $\mathbf{y}_3$, ..., and finally the vector $\mathbf{e}$. However, we may stop when the basic algorithm applies. Also, with each repetition less applications of the main lemma are required. This is made precise in the following theorem.

Theorem 4.13 (main theorem) *Let $C_0 \supset C_1 \supset C_2$ be an extension of residue codes as in (4.1). Let the genus $g \geq 1$. Let the numbers $t, r \geq 0$ satisfy $2t + r + 1 \leq deg(G_1) - 2g + 2$. For a vector $\mathbf{e}$ of weight $wt(\mathbf{e}) \leq t$ we consider the cosets $\mathbf{e} + C_j$, for $j = 0, 1, 2$. Let F_0 be an arbitrary divisor of degree t with support disjoint from D. Also, we define $F_i = F_0 + iP_\infty$, for $i = 0, 1, \ldots, 2g - 1$. Let*

$$\begin{aligned} I &= \{r, r+1, \ldots, 2g-2\}, \\ I_B &= \{i \in I \mid (A) \wedge (B), \text{ for } F = F_i\}, \\ I_B^* &= \{i \in I \mid (A) \wedge \neg(B), \text{ for } F = F_i\}, \end{aligned}$$

Then at least one of the following holds

$$L(G_1 - F_{2g-1} - P_\mathbf{e} - rP_\infty) \neq 0. \tag{4.3}$$

$$L(F_r - P_\mathbf{e}) \neq 0. \tag{4.4}$$

$$\#I_B \geq \#I_B^* + 1. \tag{4.5}$$

Proof. The assumption on wt(**e**) yields

$$\deg(F_{2g-1} - P_{\mathbf{e}}) \geq 2g - 1, \tag{4.6}$$
$$\deg(G_1 - F_r - P_{\mathbf{e}}) \geq 2g - 1. \tag{4.7}$$

Let the sets I_C and I_C^* be defined as

$$I_C = \{i \in I \mid (C1) \wedge (C2), \text{ for } F = F_i\},$$
$$I_C^* = \{i \in I \mid \neg(C1) \wedge \neg(C2), \text{ for } F = F_i\},$$

By Corollary 4.7 and Lemma 4.11 we have $I_C \subset I_B$ and $I_B^* \subset I_C^*$. Thus for (4.5) to hold it suffices to prove

$$\#I_C \geq \#I_C^* + 1. \tag{4.8}$$

We may assume that (4.3) and (4.4) do not hold. In that case we have

$$l(G_1 - F_{2g-1} - P_{\mathbf{e}}) \leq r,$$
$$l(F_r - P_{\mathbf{e}}) = 0.$$

Combination with (4.6,4.7) yields

$$l(F_{2g-1} - P_{\mathbf{e}}) - l(F_r - P_{\mathbf{e}}) \geq g,$$
$$l(G_1 - F_r - P_{\mathbf{e}}) - l(G_1 - F_{2g-1} - P_{\mathbf{e}}) \geq g - r.$$

And

$$\begin{aligned} \#I - \#I_C^* &= \#\{i \in I \mid (C1)\} + \#\{i \in I \mid (C2)\} - \#I_C, \\ &\geq (2g - r) - \#I_C. \end{aligned}$$

With $\#I = (2g - 1 - r)$ we obtain (4.8). □

4.4 Description

In the following remarks we use the theorem to formulate a decoding procedure. Let $\mathbf{y}_1 \in \mathbf{e} + C_1$ be given. If (4.3) or (4.4) holds we can apply the basic algorithm. If both (4.3) and (4.4) fail, we can apply majority coset decoding.

Remark 4.14 (basic algorithm) We recall the basic algorithm, Lemma 2.18: an error pattern **e** for the residue code $C_\Omega(D, G)$ can be corrected if the following are satisfied for a divisor F:

$$L(F - P_{\mathbf{e}}) \neq 0, \quad \text{and} \quad \deg(G - F - P_{\mathbf{e}}) > 2g - 2. \tag{4.9}$$

In that case we define a syndrome $S(F)$ and a subspace $K(F) \subset L(F)$ as in Definition 4.5 and the following reverse of Remark 4.10 holds

$$S(F)(f,g) = 0, \quad \forall g \in L(G-F) \quad \Rightarrow \quad f \in L(F - P_{\mathbf{e}}).$$

An error locator function $f \in L(F-P_{\mathbf{e}})^*$ can thus be obtained by solving for $f \in K(F)^*$. In the situation of Theorem 4.13 we have (4.6,4.7), and the conditions (4.9) are satisfied for

$$\begin{array}{lll} G = G_1 - rP_\infty, & F = G_1 - F_{2g-1} - rP_\infty, & \text{if (4.3) holds.} \\ G = G_1, & F = F_r, & \text{if (4.4) holds.} \end{array}$$

Remark 4.15 (majority coset decoding) If none of (4.3,4.4) hold, the theorem tells us that (4.5) will hold. With the notation as in the theorem the decoding proceeds as follows. The set $I_A = \{i \in I \mid (A), \text{ for } F = F_i\}$ is determined. We have $I_A = I_B \cup I_B^*$. With the possibly wrong assumption that $I_A = I_B$ application of Lemma 4.9 and Corollary 4.4 yields a vector $\mathbf{y}_2 = \mathbf{y}_1 - \lambda \mathbf{u}$, one for each $i \in I_A$. The unique vector $\mathbf{y}_2$ with $\mathbf{y}_2 \in \mathbf{e} + C_2$ is obtained for $i \in I_B$. By (4.5) this vector will occur with the highest multiplicity.

Remark 4.16 (computations) By the previous remarks we have an effective procedure to decode arbitrary algebraic geometric codes up to the designed distance. With the basic algorithm one has to solve for $f \in K(F)^*$ for suitable G, F. With majority coset decoding one has to solve for $f \in K_1(F+P_\infty) \setminus K_0(F)$ in Lemma 4.9 for a number of G_1, F. These computations clearly dominate the complexity of the procedure. They consist of solving a system of linear equations. We show that it suffices to consider one homogeneous system of linear equations $\mathbf{Sx} = \mathbf{0}$.
Let $C_\Omega(D,G)$ be a code with $\deg(G) = 2g - 2 + 2t + 1$. By replacing G by $G - P_\infty$ if necessary, we may assume that $\deg(G)$ is odd. Let $\mathbf{e}$ be an error pattern of weight $\mathrm{wt}(\mathbf{e}) \leq t$. Let F_0 be a divisor of degree t as in Theorem 4.13. The decoding procedure starts with $G_1 = G$ and if necessary continues with $G_1 = G + rP_\infty, r = 1, 2, \ldots, g$. For $r = g$ condition (4.4) in Theorem 4.13 holds and the basic algorithm will yield the error vector. Thus the decoding procedure terminates after at most g repetitions. For each G_1 the following two-dimensional syndromes are considered

$$\begin{aligned} &S_0(F_i) : L(F_i) \times L(G_0 - F_i) \longrightarrow \mathbf{F}_q, \\ &S_1(F_i) : L(F_i) \times L(G_1 - F_i) \longrightarrow \mathbf{F}_q, \\ &S_1(F_i + P_\infty) : L(F_i + P_\infty) \times L(G_1 - F_i - P_\infty) \longrightarrow \mathbf{F}_q, \\ &S_2(F_i + P_\infty) : L(F_i + P_\infty) \times L(G_2 - F_i - P_\infty) \longrightarrow \mathbf{F}_q, \end{aligned}$$

where $i = r, r+1, \ldots, 2g-2$, for $G_1 = G + rP_\infty$. All syndromes are clearly compatible. They are restrictions of a map

$$S : L(F_{2g-1}) \times L(G - F_0) \longrightarrow \mathbf{F}_q.$$

This map has a representation by a square matrix $\mathbf{S}$ of size $t+g$. To obtain compatible representations for the syndromes we choose the bases for $L(F_{2g-1})$ and $L(G-F_0)$ as follows. Let

$$L(F_0) = \langle f_1, f_2, \ldots, f_l \rangle,$$
$$L(G-F_{2g-1}) = \langle g_1, g_2, \ldots, g_m \rangle.$$

And let

$$L(F_{2g-1}) = L(F_0) + \langle f_{l+1}, f_{l+2}, \ldots, f_{t+g} \rangle,$$
$$L(G-F_0) = L(G-F_{2g-1}) + \langle g_{m+1}, g_{m+2}, \ldots, g_{t+g} \rangle,$$

such that

$$-\nu_\infty(g_{m+1}) < -\nu_\infty(g_{m+2}) < \ldots < -\nu_\infty(g_{t+g}),$$
$$-\nu_\infty(f_{l+1}) < -\nu_\infty(f_{l+2}) < \ldots < -\nu_\infty(f_{t+g}).$$

The functions $f \in K_1(F+P_\infty) \setminus K_0(F)$ and $f \in K(F)^*$ correspond to (partial) relations among the rows of the matrix $\mathbf{S}$. Note that by nature of the procedure some entries of $\mathbf{S}$ only become known in the course of the procedure, but all computations use known entries. Applying Gaussian elimination to the matrix $\mathbf{S}$ gives the partial relations as intermediate results. Thus the overall complexity can be shown to be not larger than one application of the Gaussian elimination algorithm to the matrix $\mathbf{S}$. For this, Feng and Rao formulated the Modified Fundamental Iterative Algorithm [17]. The definition of the matrix $\mathbf{S}$ has complexity $O((t+g)^2 n)$ in general. The Gaussian elimination has complexity $O((t+g)^3)$ and for the overall complexity we obtain $O(t^2 n + g^2 n)$, which is similar to the basic algorithm (of course the constants will be larger).

Remark 4.17 (modified algorithm) The condition (4.4) for the application of the basic algorithm holds for $r = g$ and at most g applications of the majority scheme are required. The modified algorithm improves on the basic algorithm and we note that the solutions for the modified algorithm are just the partial relations that occur as intermediate results in the Gaussian elimination. The modified algorithm basically claims that the first partial relation occurring is error-locating, for $t \leq (d^* - 1)/2 - s$, where s denotes the Clifford defect. Thus, for $r = 2s$ and after $2s < g$ applications of the majority scheme, the modified algorithm applies. In the formulation of the theorem, $s = s(\mathcal{E})$, for $\mathcal{E} = \{G_1 - 2F_i - P_\infty : i \geq 0\}$.

4.5 Comparison

We restrict to a particular class of AG-codes and give the proof of the previous sections in the notation of [17].

Let $\mathcal{P} = \{P_1, P_2, \ldots, P_n, P_\infty\}$ be a set of $n+1$ rational points on a non-singular curve $\mathcal{X}/\mathbf{F}_q$ of genus g, for $g \geq 1$. We consider an algebraic geometric code of type $C_\Omega(D, G) \subset \mathbf{F}_q^n$, where $D = P_1 + P_2 + \ldots + P_n$ and $G = mP_\infty$. The code has designed minimum distance $d^* = m - 2g + 2$. It can correct $t = \lfloor (d^* - 1)/2 \rfloor$ errors. We may assume that $t > 0$ or $m > 2g$. We may also assume that m is odd. For if m is even we can decode with respect to the code with $G = (m-1)P_\infty$. Let m^* denote the dimension of $L(G)$. Thus

$$m = 2t + 2g - 1, \qquad m^* = 2t + g.$$

Recall that a number o_i is a non-gap for P_∞ if $L(o_i P_\infty) \neq L((o_i - 1)P_\infty)$. Then a function $\phi_i \in L(o_i P_\infty)$ exists with pole order o_i at P_∞. As is well-known the non-gaps satisfy

$$0 = o_1 < o_2 < \ldots < o_g < o_{g+1} = 2g,$$

And

$$o_i = i + g - 1, \quad \text{for } i \geq g + 1.$$

The functions $\phi_1, \phi_2, \ldots, \phi_{m^*}$ provide a basis for the space $L(G)$.

Let $\mathbf{e}$ denote an error vector of weight $\tau \leq t$. We define one-dimensional syndromes as

$$s_k = \sum_{l=1}^{n} e_l \, \phi_k(P_l).$$

Also, we define two-dimensional syndromes as

$$S_{i,j} = \sum_{l=1}^{n} e_l \, \phi_i \phi_j(P_l).$$

For the decoding, we define the matrix $\mathbf{S} := (S_{i,j})_{1 \leq i,j \leq t+g}$. The matrix has rank τ (we omit the proof, but note that this will not hold in general for a smaller matrix; in [17] a matrix of size $2t + g$ is used). As is well-known a recurrence among the syndromes provides an error locator function. We state without proof (which follows [24],[48]): If column j of $\mathbf{S}$ is a linear combination of its previous columns, say with coefficients a_i for $1 \leq i \leq j - 1$, then the error locations are among the zeros of the function

$$\phi_j - \sum_{i=1}^{j-1} a_i \, \phi_i. \tag{4.10}$$

Not all syndromes in $\mathbf{S}$ are known, as the values of $s_{m^*+1}, s_{m^*+2}, \ldots$ are unknown. Let us assume the syndromes are known up to s_{m^*+w-1} and that s_{m^*+w} is still unknown, for $w \geq 1$. We also assume that no function (4.10) can be obtained from the known syndromes. Now we want to find s_{m^*+w}.

Let $S_{u,v}$ be a syndrome such that

$$o_u + o_v = o_{m^*+w}. \tag{4.11}$$

The syndrome can be expressed as a linear combination of the s_k, for $k = 1, 2, \ldots, m^* + w$, where the coefficient of s_{m^*+w} is non-zero. Knowledge of $S_{u,v}$ will give us s_{m^*+w}.

With $\mathbf{S}_{u,v} = (S_{i,j})_{1 \leq i \leq u; 1 \leq j \leq v}$, let

$$\operatorname{rank}(\mathbf{S}_{u,v-1}) = \operatorname{rank}(\mathbf{S}_{u-1,v-1}) = \operatorname{rank}(\mathbf{S}_{u-1,v}). \tag{4.12}$$

Then, there exists a unique value for the syndrome $S_{u,v}$, such that

$$\operatorname{rank}(\mathbf{S}_{u,v}) = \operatorname{rank}(\mathbf{S}_{u-1,v-1}). \tag{4.13}$$

A pair (u, v) that satisfies (4.12), but fails (4.13) is called a *discrepancy* of $\mathbf{S}$. A pair (u, v) satisfies (4.12) if and only if there are no discrepancies among $\{(i, v)\}_{i<u}$ and $\{(u, j)\}_{j<v}$. In particular, any row or column contains at most one discrepancy. The number of discrepancies equals the rank τ of the matrix $\mathbf{S}$. All this is straightforward to verify.

A syndrome $S_{u,v}$ with (4.11) is called a *candidate* if condition (4.12) is fulfilled. A candidate is called a *correct candidate* if the condition (4.13) is fulfilled. Otherwise it is called an *incorrect candidate*. With the known information it is possible to determine if $S_{u,v}$ is a candidate, but not if it is a correct or an incorrect candidate. On the assumption that a candidate is correct its value is uniquely determined.

We return to the determination of s_{m^*+w}. For this we seperate the known rows and columns in $\mathbf{S}$ from the rows and columns that contain some unknown entries. Let h be maximal such that $S_{t+g,h} = S_{h,t+g}$ is known.

$$\mathbf{S} = \begin{pmatrix} S_{1,1} & \cdots & S_{1,h} & S_{1,h+1} & \cdots & S_{1,t+g} \\ \vdots & & \vdots & \vdots & & \vdots \\ S_{h,1} & \cdots & S_{h,h} & S_{h,h+1} & \cdots & S_{h,t+g} \\ S_{h+1,1} & \cdots & S_{h+1,h} & S_{h+1,h+1} & \cdots & S_{h+1,t+g} \\ \vdots & & \vdots & \vdots & & \vdots \\ S_{t+g,1} & \cdots & S_{t+g,h} & S_{t+g,h+1} & \cdots & S_{t+g,t+g} \end{pmatrix}.$$

Let the number of known discrepancies in **S** be divided over the four parts as follows.

$$\begin{pmatrix} & \cdots & & \cdots & \\ \vdots & d_0 & \vdots & \vdots & d_1 & \vdots \\ & \cdots & & \cdots & \\ & \cdots & & \cdots & \\ \vdots & d_1 & \vdots & \vdots & d_2 & \vdots \\ & \cdots & & \cdots & \end{pmatrix}.$$

The assumption that $S_{t+g,h+1}$ is unknown yields

$$o_{t+g} + o_{h+1} \geq o_{m^*+w}. \tag{4.14}$$

We look for the number N of (u,v) with (4.11) and $o_u, o_v \leq o_{t+g}$. In particular

$$t + w \;=\; o_{m^*+w} - o_{t+g} \;\leq\; o_u, o_v \;\leq\; o_{t+g} \;=\; t + 2g - 1.$$

To every o_u corresponds an o_v such that (4.11) if $o_{m^*+w} - o_u$ is a non-gap. Thus

$$\begin{aligned} N \quad &\geq \#\text{ non-gaps in } \{t+w, \ldots, t+2g-1\} \;+ \\ &\qquad - \#\text{ gaps in } \{t+w, \ldots, t+2g-1\}, \\ &\geq 2\#\text{ non-gaps in } \{t+w, \ldots, t+2g-1\} \;-\; (2g-w), \\ &\geq 2(t+g-h) - (2g-w) = 2t - 2h + w, \end{aligned}$$

where we used (4.14).

Next, we consider the total number of candidates (correct + incorrect). With the remarks following (4.12,4.13) we look for those (u,v) that have no discrepancy in their row or column. Thus we obtain as a lower bound

$$T + F \geq 2t - 2h + w - 2d_1 - 2d_2. \tag{4.15}$$

An incorrect candidate is itself a discrepancy and the number of incorrect candidates F is bounded by

$$F \leq \tau - d_0 - 2d_1 - d_2. \tag{4.16}$$

The assumption that the known columns are independent yields

$$h = d_0 + d_1. \tag{4.17}$$

Combination of (4.16), (4.17) and (4.15) gives

$$2F \leq 2t - 2h - 2d_1 - 2d_2 \leq T + F - w.$$

And $T \geq F + w$. With the assumption that all candidates are correct their values can be determined according to (4.13). All correct candidates give the true value and by the obtained inequality this value will occur the most often among the values given by all candidates (correct and incorrect). The idea on which the above procedure is based is presented in [17]. The paper contains a scheme for the computations and a complete example. The construction of the example is given in the next section. In the previous sections, the relation to coset decoding is emphasized and a generalisation to all algebraic geometric codes is formulated.

The square matrix $\mathbf{S}$ that we use has size $t + g$, while the matrix $\mathbf{S}$ occurring in [17] has size $2t + g$. We obtain a majority with the candidates from a smaller matrix with the assumption (4.17). Also, the variables h and d_0, d_1 and d_2 were introduced to shorten the counting argument that leads to the inequality $T > F$ and to avoid having to consider special cases.

The notions candidates, correct candidates and incorrect candidates are introduced in [17]. They correspond with the sets I_A, I_B and I_B^* in the previous section. In particular, the inequalities to be proved are $T > F$ and $\#I_B > \#I_B^*$ respectively. The notion of discrepency is more commonly used and is very much related to the way the computations are carried out. The precense of a discrepancy in a row or column means that it is linearly independent of its previous rows or columns respectively. In this section, the numbers of discrepancies d_0, d_1 and d_2 occur in the two inequalities (4.15) and (4.16). In the final inequality $T > F$ they disappear, that is they appear in the irrelevant middle term. The use of discrepancies is avoided in the general proof of the previous section, where instead the conditions (C1) and (C2) are used. These, to the contrary, imply that a certain row or column does depend linearly on its previous rows or columns respectively. There, the inequality $I_C > I_C^*$ is proved. Again, it is irrelevant on its own, but it implies the required $I_B > I_B^*$.

The two omitted proofs relate to the basic algorithm. The two claims follow immediately with the results from Chapter 2. Finally, the matrix $\mathbf{S}$ is symmetric, while the matrix $\mathbf{S}$ occurring in Remark 4.16 is not symmetric in general. In the special case that it is symmetric, we have $G_1 - F_{2g-1} = F_r$ in the proof of Theorem 4.13. In particular, the inequality $l(G_1 - F_{2g-1} - P_e) \leq r$ can be replaced with $l(G_1 - F_{2g-1} - P_e) = 0$. And, the obtained inequality $I_B \geq I_B^* + 1$ can be sharpened to $I_B \geq I_B^* + 1 + r$, for $r \geq 0$. This corresponds indeed with $T \geq F + w$, for $w \geq 1$, that was proved as a special case in this section.

4.6 Example

Recall the situation of Lemma 2.24. The error patterns $\mathbf{e}_1$ and $\mathbf{e}_2$ are in the same coset. It is assumed that their supports are disjoint and that the sum of their weights equals the designed minimum distance of the code. In this situation, the obvious inclusion $L(F - Q_1) + L(F - Q_2) \subset K(F)$ is in fact an equality. In the same situation, the inequalities in the majority scheme can be proved to be sharp. We present results for this special case and give two explicit examples.

Notation 4.18 Let the majority scheme in Theorem 4.13 be applied with the following: $r = 0$ and thus $\deg(G_1) = 2g + 2t - 1$. The divisor F_0 is of degree t and $F_i = F_0 + iP_\infty$, for $i = 0, 1, \ldots, 2g - 1$. The majority scheme is applied when (4.3) and (4.4) fail, and we may assume

$$L(G_1 - F_{2g-1} - P_{\mathbf{e}}) = 0, \tag{4.18}$$
$$L(F_0 - P_{\mathbf{e}}) = 0. \tag{4.19}$$

Also, we consider the situation of Notation 2.19. To distinguish between the cases $\mathbf{e} = \mathbf{e}_1$ and $\mathbf{e} = \mathbf{e}_2$, we write (C1,$\mathbf{e}_1$), if condition (C1) holds with $\mathbf{e} = \mathbf{e}_1$. Similarly, with the other conditions. The sets I_B, I_B^*, I_C and I_C^* in the theorem are indexed with 1 or 2, for the vectors $\mathbf{e}_1$ and $\mathbf{e}_2$ respectively. Note that the set I_A is the same for both vectors.

Lemma 4.19 *With the notation as above, the following equivalences hold:*

$$\begin{aligned} (\text{C1},\mathbf{e}_1) &\Leftrightarrow \neg\,(\text{C2},\mathbf{e}_2) \\ (\text{C2},\mathbf{e}_1) &\Leftrightarrow \neg\,(\text{C1},\mathbf{e}_2) \end{aligned}$$

Proof. By duality, it suffices to prove the first equivalence, and

$$\begin{aligned} & l(F + P_\infty - Q_1) - l(F - Q_1) + \\ & \quad + l(G_1 - F - Q_2) - l(G_1 - F - P_\infty - Q_2) \\ = \; & l(F + P_\infty - Q_1) - l(F - Q_1) + \\ & \quad + l(K + Q_1 - F) - l(K + Q_1 - F - P_\infty) \;\; = \;\; 1. \end{aligned}$$

The claim follows. □

Corollary 4.20 *We have $I_{C,1} = I_{C,2}^*$ and $I_{C,1}^* = I_{C,2}$.*

Proposition 4.21 *In the situation of Notation 4.18, the set I_A has a partition $I_A = I_{C,1} \cup I_{C,2}$. Also, the inclusions $I_C \subset I_B$ and $I_B^* \subset I_C^*$, that occur in the proof of Theorem 4.13, are in fact equalities, for both $\mathbf{e} = \mathbf{e}_1$ and $\mathbf{e} = \mathbf{e}_2$.*

Proof. By definition, we have partitions

$$I_A = I_{B,1} \cup I^*_{B,1}, \quad I_A = I_{B,2} \cup I^*_{B,2}.$$

Also, we have the obvious inclusions

$$I_{C,1} \cup I_{C,2} \subseteq I_{B,1} \cup I_{B,2} \subseteq I_A.$$

To prove the claims, it suffices to prove

$$I_A \subseteq I_{C,1} \cup I_{C,2}, \tag{4.20}$$

and

$$I_{B,1} \cap I_{B,2} = \emptyset. \tag{4.21}$$

For (4.20), we use Lemma 2.24, and $K_1(F) = L(F-Q_1)+L(F-Q_2)$. Thus, (A1), or $K_1(F+P_\infty) \neq K_1(F)$, implies that either (C1,$\mathbf{e}_1$) or (C1,$\mathbf{e}_2$) holds. Similarly (A2,A3), or $K_1((G_1 - F) \neq K_1(G_1 - F - P_\infty)$, implies that either (C2,$\mathbf{e}_1$) or (C2,$\mathbf{e}_2$) holds. Here, we use Remark 4.8. Now use Lemma 4.19. For (4.21), we use Lemma 4.9 and Corollary 4.4. With $F = F_i$, for $i \in I_{B,1} \cap I_{B,2}$, a combination of the lemma and the corollary yields a vector $\mathbf{y}$,

$$\mathbf{y} \in \mathbf{e}_1 + C_\Omega(D, G_1 + P_\infty) \cap \mathbf{e}_2 + C_\Omega(D, G_1 + P_\infty) = \emptyset,$$

a contradiction. □

In the terms of the previous section, the candidates split into two classes. The candidates yield a new syndrome, either for the coset $\mathbf{e}_1 + C_\Omega(D, G_1 + P_\infty)$ or for the coset $\mathbf{e}_2 + C_\Omega(D, G_1 + P_\infty)$. To prove this, we use the notation of Section 4.3. Knowing that the candidates split into two classes, the cardinalities of the two classes can be made precise in either of the two notations.

Lemma 4.22 *For the cardinalities we have*

$$\#I_{C,1} + wt(\mathbf{e}_1) = \#I_{C,2} + wt(\mathbf{e}_2).$$

In the subsequent steps of the majority scheme, the set I_A of condidates will consist of one class of correct candidates.

Proof. By Lemma 4.19, $I_{C1,1} \backslash I_{C,1} = I_{C1,2} \backslash I_{C,2}$, and $\#I_{C,1} - \#I_{C,2} = \#I_{C1,1} - \#I_{C1,2}$. But,

$$\#I_{C1,1} = l(F_{2g-1} - Q_1) - l(F_0 - Q_1),$$
$$\#I_{C1,2} = l(F_{2g-1} - Q_2) - l(F_0 - Q_2).$$

Also,

$$i(F_{2g-1} - Q_1) = l(G_1 - F_{2g-1} - Q_2),$$
$$i(F_{2g-1} - Q_2) = l(G_1 - F_{2g-1} - Q_1).$$

With the assumptions (4.18) and (4.19), we obtain

$$\#I_{C1,1} - \#I_{C1,2} = \deg(F_{2g-1} - Q_1) - \deg(F_{2g-1} - Q_2).$$

Let (A) hold in step j, for some $j \geq 2$. Thus, $K_j(F + P_\infty) \neq K_j(F)$, and a fortiori $K_2(F + P_\infty) \neq K_2(F)$ and $K_1(F + P_\infty) \neq K_1(F)$. By the latter inequality, either (C1,$\mathbf{e}_1$) or (C1,$\mathbf{e}_2$) holds. The candidate will be correct for $\mathbf{e}_1$, if (C1,$\mathbf{e}_1$) holds, and we are done. If (C1,$\mathbf{e}_1$) fails, we have (C,$\mathbf{e}_2$). This in turn implies that (B,$\mathbf{e}_1$) fails, contradicting the former inequality. □

The proof is considerably shorter in the other notation. Let T, F and τ be indexed by 1 and 2, for the vectors $\mathbf{e}_1$ and $\mathbf{e}_2$ respectively.

Lemma 4.23 *For the cardinalities we have*

$$T_1 + \tau_1 = T_2 + \tau_2.$$

After one application of the majority scheme, all discrepancies are known. The inequalities (4.15) and (4.16) are sharp in the situation of Notation 2.19.

Proof. The partition in Proposition 4.21 implies that $T_1 = F_2$ and $T_2 = F_1$. Combination of (4.16) and (4.15), for $w = 1$, yields

$$\tau_1 + \tau_2 - 2d_0 - 4d_1 - 2d_2 \geq F_1 + F_2 \geq 2t - 2h + 1 - 2d_1 - 2d_2.$$

With h as in (4.17)), we have equalities, and the claims follow immediately. □

Remark 4.24 Let $\mathcal{X} : Y^4Z + YZ^4 = X^5$ denote the Hermite curve of degree five over the field $GF(16)$. The curve contains 64 finite rational points, with $Z \neq 0$, and a point $P_\infty = (0 : 1 : 0)$ at infinity. With D the sum of the finite rational points and $G_1 = 23P_\infty$, we define the code $C = C_\Omega(D, G-1)$. It is of type $[64, 46, 13]$. The words $\mathbf{x}$ of minimum weight have support $P_\mathbf{x} \sim 13P_\infty$. Let l_1, l_2 and l_3 be three lines with intersection divisors L_1, L_2 and L_3:

$$\begin{aligned} L_1 &= P_1 + P_2 + P_3 + P_4 + P_5, \\ L_2 &= P_6 + P_7 + P_8 + P_9 + P_\infty, \\ L_3 &= P_{10} + P_{11} + P_{12} + P_{13} + P_\infty, \end{aligned} \tag{4.22}$$

such that $|\{P_1, P_2, \ldots, P_{13}\}| = 13$. Let $\mathbf{x}$ be a codeword with support $Q_{\mathbf{x}} = P_1 + P_2 + \ldots + P_{13}$, say

$$\mathbf{x} = (c_1, c_2, \ldots, c_{12}, c_{13}, 0, 0, \ldots, 0, 0). \tag{4.23}$$

The lines l_1, l_2 and l_3 and the word $\mathbf{x}$ indeed exist. For example, the lines $l_1 : X = Y$, $l_2 : X = Z$ and $l_3 : X = a^3 Z$ give the word

$$\mathbf{x} = (a^{12}, a^4, a^7, a^8, a^9, a^9, a^5, a^{12}, a^{11}, a^4, a^7, a^6, 1, 0, 0, \ldots, 0, 0),$$

where $a \in GF(16)$ satisfies $a^4 + a + 1 = 0$.

We use the codeword $\mathbf{x}$ in (4.23) to illustrate the majority scheme in the situation of Notation 2.19.

Example 4.25 Let

$$\mathbf{e}_1 = (0, 0, 0, c_4, c_5, c_6, c_7, c_8, c_9, 0, 0, 0, 0, 0, 0, \ldots, 0, 0),$$
$$\mathbf{e}_2 = (c_1, c_2, c_3, 0, 0, 0, 0, 0, 0, c_{10}, c_{11}, c_{12}, c_{13}, 0, 0, \ldots, 0, 0).$$

Thus, $\mathbf{e}_1$ is the coset leader and $\mathbf{e}_2 \in \mathbf{e}_1 \in C_\Omega(D, 23P_\infty)$. The table on top of page 64 yields the candidates and their partition. A plus sign indicates that a condition holds. In case it fails, we put a minus sign, unless the failure is due to a gap of the form $L(F + P_\infty) = L(F)$, in which case we put a g. The matrix below is given in the notation of [17]. The positions of the discrepancies (marked $*$) matches the table on top, but slightly different positions may also occur.

The following example was suggested to G.L.Feng for inclusion in [17], as it shows the shortcomings of the modified algorithm and the fruitful use of the majority scheme. In their paper, the example is worked out for the vector $\mathbf{x}$ that is given explicitly in the remark.

Example 4.26 Let

$$\mathbf{e}_1 = (c_1, c_2, c_3, c_4, c_5, c_6, 0, 0, 0, 0, 0, 0, 0, 0, 0, \ldots, 0, 0),$$
$$\mathbf{e}_2 = (0, 0, 0, 0, 0, 0, c_7, c_8, c_9, c_{10}, c_{11}, c_{12}, c_{13}, 0, 0, \ldots, 0, 0),$$

Thus, $\mathbf{e}_1$ is the coset leader and $\mathbf{e}_2 \in \mathbf{e}_1 \in C_\Omega(D, 23P_\infty)$. The difference between the number of correct and incorrect candidates is again one, but the total number of candidates has changed. Recall that the modified algorithm yields as output the first recurrence among columns that holds for the known syndromes.Thus, it yields the function of pole order 8 that corresponds with the fourth column. But, this function locates the positions of $\mathbf{e}_2$ and the modified algorithm fails to locate the error positions of $\mathbf{e}_1$.

(Example 4.25)

	$\mathbf{e}_1$			$\mathbf{e}_2$		
i	$(C1)$	$(C2)$	T/F	$(C1)$	$(C2)$	T/F
0	g	$+$		g	$+$	
1	$-$	$+$		$-$	$+$	
2	$+$	$-$		$+$	$-$	
3	$-$	$+$		$-$	$+$	
4	g	$+$		g	$+$	
5	$+$	$+$	T	$-$	$-$	F
6	$+$	g		$+$	g	
7	$+$	$-$		$+$	$-$	
8	$-$	$+$		$-$	$+$	
9	$+$	$-$		$+$	$-$	
10	$+$	g		$+$	g	

$$I_A = \{5\}, \qquad I_A = \{5\},$$
$$I_{B,1} = I_{C,1} = \{5\}, \qquad I_{B,2} = I_{C,2} = \emptyset,$$
$$I^*_{B,1} = I^*_{C,1} = \emptyset, \qquad I^*_{B,2} = I^*_{C,2} = \{5\}.$$

	0	4	5	8	9	10	12	13	14	15	16	17
0	$*$	0	0	0	0	0	0	0	0	0	0	0
4	s	$*$	0	0	0	0	0	0	0	0	0	0
5	s	s	0	0	0	0	0	0	0	$*$	0	0
8	s	s	0	$*$	0	0	0	0	0	s	@	#
9	s	s	0	s	0	0	0	0	0	@	#	#
10	s	s	0	s	0	$*$	0	0	@	#	#	#
12	s	s	0	s	0	s	$@_7$	#	#	#	#	#
13	s	s	0	s	0	s	#	#	#	#	#	#
14	s	s	0	s	0	@	#	#	#	#	#	#
15	s	s	$*$	s	@	#	#	#	#	#	#	#
16	s	s	s	@	#	#	#	#	#	#	#	#
17	s	s	s	#	#	#	#	#	#	#	#	#

$$@_7 \ : \ \mathbf{e}_1 + C_\Omega(D, 24P_\infty).$$

(Example 4.26)

	$\mathbf{e}_1$			$\mathbf{e}_2$		
i	$(C1)$	$(C2)$	T/F	$(C1)$	$(C2)$	T/F
0	g	$+$		g	$+$	
1	$-$	$-$	F	$+$	$+$	T
2	$+$	$+$	T	$-$	$-$	F
3	$+$	$+$	T	$-$	$-$	F
4	g	$+$		g	$+$	
5	$-$	$-$	F	$+$	$+$	T
6	$+$	g		$+$	g	
7	$+$	$+$	T	$-$	$-$	F
8	$+$	$+$	T	$-$	$-$	F
9	$-$	$-$	F	$+$	$+$	T
10	$+$	g		$+$	g	

$I_A = \{1,2,3,5,7,8,9\}$, $I_A = \{1,2,3,5,7,8,9\}$,
$I_{B,1} = I_{C,1} = \{2,3,7,8\}$, $I_{B,2} = I_{C,2} = \{1,5,9\}$,
$I^*_{B,1} = I^*_{C,1} = \{1,5,9\}$, $I^*_{B,2} = I^*_{C,2} = \{2,3,7,8\}$.

	0	4	5	8	9	10	12	13	14	15	16	17
0	$*$	0	0	0	0	0	0	0	0	0	0	0
4	s	$*$	0	0	0	0	0	0	0	0	0	0
5	s	s	$*$	0	0	0	0	0	0	0	0	0
8	s	s	s	0	0	0	0	0	0	0	$@_{11}$	#
9	s	s	s	0	0	0	0	0	0	$@_{10}$	#	#
10	s	s	s	0	0	0	0	0	$@_9$	#	#	#
12	s	s	s	0	0	0	$@_7$	#	#	#	#	#
13	s	s	s	0	0	0	#	#	#	#	#	#
14	s	s	s	0	0	$@_6$	#	#	#	#	#	#
15	s	s	s	0	$@_5$	#	#	#	#	#	#	#
16	s	s	s	$@_4$	#	#	#	#	#	#	#	#
17	s	s	s	#	#	#	#	#	#	#	#	#

$@_5, @_6, @_9, @_{10}$: $\mathbf{e}_1 + C_\Omega(D, 24P_\infty)$,
$@_4, @_7, @_{11}$: $\mathbf{e}_2 + C_\Omega(D, 24P_\infty)$.

Part III

Decoding cyclic codes

Chapter 5

The BCH-bound and beyond

5.1 Notation

In this chapter we suggest a general format for error-locating pairs that is based on the Roos-bound. In the next chapter two different formats are presented. Chapter 7 gives pairs to decode all but four binary cyclic codes of length less than 63 up to their actual capability.

A cyclic code $C \subset \mathbb{F}^n$ is usually identified with an ideal in the ring $\mathbb{F}[x]/(x^n - 1)$ 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

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

The code is determined by the zeros of $g(x)$. We assume $(\text{char } \mathbb{F}, n) = 1$, so that $x^n - 1$ has n different zeros. Let the extension $\overline{\mathbb{F}}$ of $\mathbb{F}$ contain the n-th roots of unity and let $\alpha \in \overline{\mathbb{F}}$ be a primitive n-th root of unity. Let $m_i(x)$ be the minimal polynomial of α^i over $\mathbb{F}$. If $g(x)$ equals $\text{lcm}\{m_i(x) : \alpha^i \in R\}$ 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 $\overline{C} \subset \overline{\mathbb{F}}^n$. The code C is obtained as the subfield subcode of $\overline{C}$, that means $C = \overline{C} \cap \mathbb{F}^n$.

Definition 5.1 Let R be a *defining set* of a cyclic code $C/\mathbb{F}$. C is then defined as

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

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 notion of generating sets is very convenient to describe error-locating pairs for cyclic codes.

Definition 5.2 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 5.3 *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 $b + cR = \{b + ci \pmod{n} : i \in R\}$. 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.

5.2 Decoding BCH-codes

Theorem 1.2 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 5.4 *(Roos-bound, [31, Theorem 3])* *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 [31]. □

Corollary 5.5 *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 $c_1 A + c_2 B$.*

Proof. First it is immediate from the proof in [31] 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 $c_1 A$ and B. □

Theorem 5.6 *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 $R = b + c_1 I + c_2 J + c_3 K$ has a t-error-locating pair (U, V), where $U/\overline{\mathbb{F}}$ is defined by the generating set $b + c_1 I$ and $V/\overline{\mathbb{F}}$ by the generating set $c_2 J + c_3 K$. 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 (1.1) and (1.2) 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 (1.3) holds. The distance $d(C)$ follows with another application of the lemma, this time with

$$\begin{aligned} &A = c_2 J + c_3 K, \quad d_A \geq t + 1 \quad \text{and} \\ &B = b c_1^{-1} + I, \quad |\overline{B}| \leq (t+1) + (t+1) - 2. \end{aligned}$$

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 5.3) and $d(C) > |\overline{I}| - 1$ yields condition (1.11). □

Example 5.7 *(Example 1 [19])* 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 ([39],[31]). 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 [18]. We recall the two bounds, following [18], and show that the procedure is a special case of Theorem 5.6. 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 5.4. With the further restrictions

$$s + 1 \leq d_0 - 1, \tag{5.1}$$

$$j_{s+1} - j_1 < (d_0 + s - 1)/2. \tag{5.2}$$

the procedure in [18] 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 5.8 *(Theorem 4 and Theorem 5 [18])* *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 (5.1) can be written as $s < t$. In particular the set J is well-defined. The restriction (5.2) yields $|\overline{K}| - 1 < t$. Thus all conditions of the theorem are fulfilled. □

5.3 Recurrences

Some error-locating pairs in Theorem 5.6 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 5.9 *Let the vector* $\mathbf{e}$ *have support in a set* A *and let* R *contain* $|A|$ *consecutive integers. Let the syndromes*

$$S_i = \langle \mathbf{e}, \mathbf{a}(i) \rangle, \quad i \in R, \tag{5.3}$$

be known. Then the set of equations (5.3) 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, a solution with support in A is unique. An efficient way to solve for $\mathbf{e}$ is given by Forney's algorithm [39, p.297]. □

Recall from the computational scheme that in solving (1.4) for $\mathbf{u} \in U \setminus \mathbf{0}$ we actually find a vector σ solving the key equation (1.6). 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 1.8 and for $\sigma = (\sigma_{i_1}, \ldots, \sigma_{i_l})$ we have $\mathbf{u} = \sigma G_U = Ev(\sigma)$ for the polynomial

$$\sigma = \sigma_{i_l} X^{i_l} + \ldots + \sigma_{i_2} X^{i_2} + \sigma_{i_1} X^{i_1}. \tag{5.4}$$

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 5.10 *Let the polynomial* σ *(5.4) have the support of* $\mathbf{e}$ *among its zeros. Then*

$$\sigma_{i_l} S_{i_l+j} + \ldots + \sigma_{i_2} S_{i_2+j} + \sigma_{i_1} S_{i_1+j} = 0, \tag{5.5}$$

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 5.11 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

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

defines an error-locating pair (U, V) by Theorem 5.6. 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 (5.5) 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 5.9 applies.

Remark 5.12 When using recurrences of type (5.5) 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 [19], but there zero coefficients are not considered. Thus, the procedure as described in [19] may fail for the entries 17,84 and 121 in Table 1.

Example 5.13 We consider the code C of Example 5.11. The procedure in [19] corresponds to the choices

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

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 $S_{15+j} = S_j$.

5.4 Correcting more errors

Theorem 5.6 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 5.14 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 5.15 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 [18] 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 $S(\mathbf{y})$ (1.6). 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 5.16 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.

Chapter 6

Pairs from MDS-codes

6.1 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 6.1 *Let C and A denote two cyclic codes over $\overline{\mathbb{F}}$ of length n. Let their defining sets be given by*

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

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 6.2 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 6.3 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 $r \leq 5$.

6.2 Construction of pairs

For a t-error-correcting BCH-code C with defining set $R \supset \{1, 2, \ldots, 2t\}$, we have by Theorem 5.6 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 generally, we have

Proposition 6.4 *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 [31] 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 (1.1)–(1.3) and (1.11) for an error-correcting pair follow immediately. □

Remark 6.5 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 6.1. 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 (1.6) 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 6.6 *(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 (1.6) can be solved with complexity $\mathcal{O}(t^2)$.

Proof. The value of t_l can be obtained with Theorem 6.1. The complete defining set R for C satisfies $R \supset I + J$ and thus $C \perp U * V$. We may use Proposition 6.4. For the solving of the key equation see Section 6.3. □

Theorem 6.7 *(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 (1.6) can be solved with complexity $\mathcal{O}(t^2)$.

Proof. As in Theorem 6.6. □

Example 6.8 We consider codes of length $n = 23$ over $GF(2^{11})$. Let U and V be as in Theorem 6.7 with $q = 2, l = 1, t = 3$, or $I = \{0, 1, 4, 16\}$ and $J = \{0, 2, 8\}$. By Example 6.3, 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 6.9 (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*

$$\begin{array}{ll} i = q^{sl} + 1, \ \textit{for } s \geq 1, & \textit{first format (Theorem 6.6).} \\ i = q^{(2s-1)l} + 1 \ \textit{for } s \geq 1, & \textit{second format (Theorem 6.7).} \end{array}$$

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,

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

Similarly for the second format. □

Example 6.10 We consider codes of length $n = 39$ over $GF(2^{12})$. Let U and V be as in Theorem 6.6 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 5.9.

6.3 Fundamental iterative algorithm

In their paper [18], 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 6.6 and Theorem 6.7. 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 [18].

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 6.6 and Theorem 6.7. 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.

6.4 Reducing complexity

In general the complexity of the procedure described in Section 1.1 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 5.9.

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

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

Lemma 6.11 *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 6.11 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 6.12 *Let $S(\mathbf{y})$ be the $(t-1)\times(t+1)$-matrix of Lemma 6.11. 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 6.13 (on the proof of Theorem 6.6 and Theorem 6.7) To complete the proof of Theorem 6.6, we have to add a row to $S(\mathbf{y})$ of Lemma 6.11. 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 6.7 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 6.14 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 $\overline{\mathbb{F}}$ containing an n-th root of unity. In some cases however U and V can be taken to be codes defined over $\widehat{\mathbb{F}} \subset \overline{\mathbb{F}}$. This implies that $S(\mathbf{y})$ has entries from $\widehat{\mathbb{F}}$ rather than from $\overline{\mathbb{F}}$ which allows us to perform these operations faster.

Example 6.15 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 $GF(4)$.

Chapter 7

Applications

7.1 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 [31]. Equivalent codes and subcodes with the same error correction capability use the same pair. Usage of hyperplanes (Proposition 1.20) 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 [19]. 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 (1.1) and (1.2) of Definition 1.1 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 6.1. Thus, also condition (1.3) 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 6.1 respectively. Cases 106 and 107 are treated in Example 7.3 and Example 7.4 respectively.
To show that the pairs are error-correcting we use either the BCH-bound to show that condition (1.11) is satisfied or we use recurrences to determine unknown syndromes until we can apply Lemma 5.9. Whenever we use the conjugacy format, Lemma 6.9 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 (5.5) in Lemma 5.10,

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

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.

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	FT
11	23	12	7	{1}	{2,8,0}	{1,4,16,0}	4	S0=1
13	31	21	5	{1,5}	{1,4}	{0,1,4}	3	
17		16	7	{1,5,7}	{0,1,2}	{0,7,8,18}	4	FT,S0=1
19		11	11	{1,3,5,11}	{0,...,3}	{1,2,3,8,9,10}	5	H,Ex.7.1
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	FT
					{-2,...,2}	{-13,-12,-11,+11,+12,+13}	6	$GF(32)$,Ex.7.2
36	35	16	7	{1,5,7}	{0,3,6}	{1,2,4,5}	4	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	FT,S0=0
45		15	10	{1,3}	{0,2,4,8}	{1,2,4,8,16}	5	Ex.6.10
					{0,...,3}	{1,2,3,8,9}	5	FT,Ex.5.7
47		13	12	{1,3,13}	{0,...,3}	{1,2,3,8,9,10}	5	H
49	41	21	9	{1}	{8,20,9,0}	{1,23,37,31,0}	5	
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	FT
83		16	10	{0,1,3,7}	{0,...,3}	{-2,...,1,11}	5	FT
84		15	9	{1,3,7,15}	{0,...,3}	{1,11,...,14}	5	Ex.5.11
85		15	10	{1,7,9,15}	{0,1,2,13,14,15}	{13,...,17}	5	FT
90		9	12	{1,5,7,9,15}	{13,...,17}	{0,1,2,13,14,15}	6	
92	47	24	11	{1}	{3,27,8,0}	{1,9,34,24,0}	5	S0=0
96	51	35	5	{1,9}	{0,1}	{1,8,15}	3	FT
					{0,8}	{1,8,13}	3	
98		34	6	{0,1,5}	{0,1}	{0,1,4}	3	FT
106		27	8	{1,3,9}	{0,2,8}	{0,1,4,16}	3	S0=1,Ex.7.3
107		27	9	{1,5,9}	{0,2,7,8,13}	{0,2,7,8,13}	5	S0=0,Ex.7.4
108		27	9	{1,3,19}	{-4,...,-1}	{0,...,4}	5	S0=0,1
119		19	14	{1,3,5,9}	{0,...,4}	{0,...,4,12,6}	6	S0=0,H
121		17	12	{1,3,9,17,19}	{-4,...,0}	{0,...,4,19}	6	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	Ex.7.5
124		17	16	{1,3,5,9,17}	{0,...,5}	{0,...,5,12,13}	7	H,S0=1
128		11	15	{1,3,5,11,19}	{-7,...,-1}	{0,...,7}	8	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	
140		21	15	{1,5,11}	{0,7,16,13,14,32}	{0,1,18,49,2,36,43,4}	7	H,S0=1
144	57	21	14	{1,3}	{0,2,4,8,16}	{0,-1,-2,-4,-8,-16,-32}	6	H,S0=0,1
146		19	16	{1,3,19}	{0,2,4,8,16}	{0,-1,-2,-4,-8,-16,-32}	6	H,S0=0,1

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

Table 1. Error-correcting pairs for binary cyclic codes.

Table 2. Recurrence relations on syndromes.

For details about the notation, see page 85.

Example 7.1 (19) Conditions (1.1),(1.2) and (1.11) are satisfied with $t = 5$, but $d(V^{\perp}) = 5$ and condition (1.3) is not satisfied. We follow Proposition 1.20. Thus, in case we obtain two independent solutions $\mathbf{u}_1$ and $\mathbf{u}_2$ to the key equation, we look for a linear combination of $\mathbf{u}_1$ and $\mathbf{u}_2$ that has at least five zeros. At most six such combinations exist and for each we solve for an error pattern with support among the zeros. Entries 47,119,124,137,140,144,146 proceed likewise.

Example 7.2 (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 (1.6) can be solved over the field $GF(32)$.

Example 7.3 (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 (1.3) are obviously satisfied. In fact $d(V^\perp) = 3$ and condition (1.3) does not hold for $t = 3$. We prove the weaker condition (1.9): $(\mathbf{e} * U) \cap V^\perp = \mathbf{0}$. Words of weight three in $V^\perp / GF(256)$ are in the code with defining set $R = \{1, 4, 16, 13, 0\}$ by Lemma 6.1. 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 (1.9) is satisfied.In [19] the pair $I = \{0, 2, 8, 12\}$ and $J = \{0, 1, 4\}$ is given without the above verification.

Example 7.4 (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 (1.3): $d(V^\perp) > 4$. A word $\mathbf{c} \in V^\perp$ satisfies the checks $\{8, 13, 2, 0\}$ and by theorem 6.1 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 [31] 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 7.5 (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:

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

7.2 Sequences of codes

In a certain range of the minimum distance, the conjugacy formats of Chapter 6 allow the construction of sequences of cyclic codes with the same designed distance and redundancy as BCH-codes.

Proposition 7.6 *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 [32, 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 6.7, 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 (1.1)-(1.3) and (1.11) are fulfilled. Since S_0 occurs only once in the key matrix $S(\mathbf{y})$, by the results of Section 5.4 we can assume that $S_0 \equiv 3 \pmod 2$. □

Remark 7.7 (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 6.6 and 6.7 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 7.8 *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 6.4. □

Example 7.9 (Zetterberg codes [55]) 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 7.8.

Example 7.10 (Melas codes [34]) 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 7.8.

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

Proposition 7.11 *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 5.6. □

Bibliography

[1] E. Arbarello, M. Cornalba, P.A. Griffiths and J. Harris, *Geometry of algebraic curves I.* New York: Springer-Verlag, 1985.

[2] E.R. Berlekamp, "Decoding algorithms for block codes," in *Key Papers in the Development of Coding Theory*, E.R. Berlekamp, Ed.. New York: IEEE Press, 1974.

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

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

[5] 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.

[6] P.J. Cameron and J.H. van Lint, *Designs, Graphs, Codes and their Links.* Cambridge: Cambridge University Press, 1991.

[7] C. Chevalley, *Introduction to the theory of algebraic functions of one variable.* New York: AMS, 1951.

[8] C. Dahl, *Codes and Geometry.* Thesis, Technical University of Denmark, 1991.

[9] I.M. Duursma, "Algebraic decoding using special divisors," *IEEE Trans. Inform. Theory*, vol.IT-39, March 1993.

[10] I.M. Duursma, "On the decoding procedure of Feng and Rao," Proceedings Algebraic and Combinatorial Coding Theory, Voneshta Voda, Bulgaria, 1992.

[11] I.M. Duursma, "Majority coset decoding," *IEEE Trans. Inform. Theory*, vol.IT-39, May 1993.

[12] I.M. Duursma, and R. Kötter, *Error-locating pairs for cyclic codes.* Eindhoven, Linköping: preprint, 1993.

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

[14] D. Ehrhard, "Decoding algebraic-geometric codes by solving a key equation," Proceedings Algebraic Geometry and Coding Theory, Luminy, France, 1991.

[15] D. Ehrhard, "Decoding Hermitian Codes over $\mathbb{F}_{q^2}$ up to designed error capability," Proceedings Algebraic and Combinatorial Coding Theory, Voneshta Voda, Bulgaria, 1992.

[16] D. Ehrhard, "Achieving the designed error capacity in decoding algebraic-geometric codes," to appear in *IEEE Trans. Inform. Theory.*

[17] G.L. Feng and T.R.N.Rao, "Decoding algebraic geometric codes up to the designed minimum distance," *IEEE Trans. Inform. Theory*, vol.IT-39, January 1993.

[18] 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.

[19] 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.

[20] W. Fulton, *Algebraic curves.* Reading, MA: Benjamin, 1969.

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

[22] C.R.P. Hartmann and K.K. Tzeng, "Generalizations of the BCH bound," *Inform. Contr.*, vol.20, pp.489-498, 1972.

[23] R. Hartshorne, *Algebraic geometry.* New York: Springer-Verlag, 1977.

[24] 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.

[25] J. Justesen, K.J. Larsen, H.E. Jensen, and T. Høholdt, "Fast decoding of codes from plane algebraic curves," *IEEE Trans. Inform. Theory*, vol.IT-38, pp.111-119, 1992.

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

[27] V.Yu. Krachkovskii, *Decoding of codes on algebraic curves.* In Russian, preprint, 1988.

[28] G. Lachaud, "Projective Reed-Muller codes," in *Lect. Notes in Comp. Sci.*, vol.311. Berlin: Springer, 1988.

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

[30] J.H. van Lint and G. van der Geer, *Introduction to coding theory and algebraic geometry.* Basel: Birkhäuser Verlag, 1988.

[31] 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.

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

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

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

[35] C. Moreno, *Algebraic curves over finite fields.* Cambridge: Cambridge University Press, 1991.

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

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

[38] R. Pellikaan, "On a decoding algorithm for codes on maximal curves," *IEEE Trans. Inform. Theory*, vol.IT-35, pp.1228-1232, 1989.

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

[40] S.C. Porter, *Decoding codes arising from Goppa's construction on algebraic curves.* Thesis, Yale University, 1988.

[41] S.C. Porter, B. Shen and R. Pellikaan, "On decoding geometric Goppa codes using an extra place," *IEEE Trans. Inform. Theory*, vol.IT-38, pp.1663-1676, 1992.

[42] 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.

[43] C. Roos, "A new lower bound for the minimum distance of a cyclic code," *IEEE Trans. Inform. Theory*, vol.IT-29, pp.330-332, 1983.

[44] D. Rotillon and J.A. Thiong Ly, "Decoding of codes on the Klein quartic," in *Lect. Notes in Comp. Sci.*, vol.514. Berlin: Springer, 1991.

[45] S. Sakata, "Finding a minimal set of linear recurring relations capable of generating a given finite two-dimensional array," *Journal of Symbolic Computation*, vol.5, pp.321-337, 1988.

[46] S. Sakata, "Extension of the Berlekamp-Massey algorithm to N dimensions," *Information and Computation*, vol.84, pp.207-239, 1990.

[47] B. Shen, "On encoding and decoding of the codes from Hermitian curves," in *Cryptography and Coding III*, M. Ganley, Ed.. Oxford: Oxford University Press, 1993.

[48] 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.

[49] A. B. Sørensen, "Projective Reed-Muller Codes," *IEEE Trans. Inform. Theory*, vol.IT-37, pp.1567-1576, 1991.

[50] 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.

[51] H. Stichtenoth, *Algebraic Function Fields and Codes.* Springer Universitext, 1993.

[52] M.A. Tsfasman and S.G. Vlăduţ, *Algebraic Geometric Codes.* Dordrecht: Kluwer Acad. Publ., 1991.

[53] S.G. Vlăduţ, "On the decoding of algebraic-geometric codes over $\mathbf{F}_q$ for $q \geq 16$," *IEEE Trans. Inform. Theory*, vol.IT-36, pp.1461-1463, 1990.

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

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

Samenvatting

Het proefschrift beschrijft de resultaten van mijn onderzoek aan decodeeralgoritmen voor lineaire codes. De algoritmen worden toegepast bij het corrigeren van fouten die onbedoeld optreden bij het verzenden of opslaan van informatie.

Algebraïsch-meetkundige codes (in de zin van Goppa) kennen een eenvoudig te bepalen ondergrens voor het aantal fouten dat gecorrigeerd kan worden. Deze ondergrens wordt de ontwerpcapaciteit genoemd. Bij aanvang van het onderzoek waren twee nauw verwante algoritmen beschikbaar: het basis algoritme en het gemodificeerde algoritme. Beide algoritmen zijn in het algemeen niet in staat om fouten te corrigeren tot de ontwerpcapaciteit van de code. De gemodificeerde versie corrigeert meer fouten maar is slechts toepasbaar op een beperkte klasse van codes. Voor het gemodificeerde algoritme is een formulering gevonden die het toepasbaar maakt op alle algebraïsch-meetkundige codes. Voor de gevallen waarbij de herformulering het algoritme niet wezenlijk verandert wordt bewezen dat de prestaties van het algoritme in feite beter zijn dan aanvankelijk was aangetoond. Gedurende het onderzoek suggereerden Feng en Rao een wezenlijke verbetering van de bestaande algoritmen. Hiermee kunnen fouten worden gecorrigeerd tot de ontwerpcapaciteit. In een voorpublikatie lichten zij dit toe aan de hand van een voorbeeld en ontbreekt een bewijs. Uitwerking van hun idee heeft geresulteerd in een algemeen toepasbaar algoritme met een volledig bewijs. Het huidige bewijs van Feng en Rao gaat uit van een beperkt toepasbaar algoritme, dat een groter beslag legt op computergeheugen en rekentijd.

In samenwerking met R. Kötter, werkzaam aan de universiteit van Linköping, zijn resultaten bereikt voor het decoderen van cyclische codes. Algemene stellingen worden gegeven voor het construeren van decodeeralgoritmen voor deze codes. In het bijzonder worden algoritmen gegeven die meer fouten corrigeren dan het Berlekamp-Massey algoritme als de optredende eenheidswortels verdeeld zijn over een beperkt aantal grote conjugatieklassen. De complexiteit is die van het Berlekamp-Massey algoritme. Toetsing van de stellingen aan binaire codes van lengte kleiner dan 63 leert dat alle codes op vier na gedecodeerd kunnen worden tot de werkelijke capaciteit.

Curriculum vitae

Iwan Maynard Duursma was born on April 19, 1963, in Bussum. He attended the rijksscholengemeenschap Schoonoord in Zeist and completed the gymnasium in 1980. From September 1980 till August 1986, he studied aerospace engineering at the Technische Hogeschool Delft and graduated with the master's thesis: Satellite orbit perturbations due to tidal forces (cum laude, professor K.F. Wakker). From September 1983 till February 1989, he studied mathematics at the Universiteit van Amsterdam and graduated with the master's thesis: Some remarks on Goppa codes (cum laude, professor G. van der Geer).
During his study, he was appointed as student assistant (for September 1985 – April 1987, by professor H.W. Lenstra jr., and for January 1988 – November 1988, by A.M. Cohen). While applying for a position as a Ph.D. student, he was a full-time teacher of mathematics at the Instituut Blankestijn in Utrecht. Since September 1990, he has worked in the Discrete Mathematics group at the Technische Universiteit Eindhoven, with support from the Netherlands Organization for Scientific Research.

Stellingen behorende bij het proefschrift

DECODING CODES FROM CURVES AND CYCLIC CODES

van Iwan M. Duursma

I.

Het projectieve vlak over het lichaam $GF(8)$ bevat 73 rationale punten. Hieronder bevinden zich acht drietallen van de vorm $\{(X : Y : Z), (X^2 : Y^2 : Z^2), (X^4 : Y^4 : Z^4)\}$ met de eigenschap dat de drie punten niet op een lijn liggen. De overige 49 punten liggen op de zeven lijnen gedefinieerd over het lichaam $GF(2)$. De automorfismengroep van de acht drietallen en de automorfismengroep van de zeven lijnen zijn identiek (als ondergroep van de automorfismengroep van het projectieve vlak). Het zijn $L_2(7)$, respectievelijk $L_3(2)$.

II.

Zij k een lichaam van karakteristiek 2. Zij $S_i, T_i \in k[X, Y, Z]$ gedefinieerd door $S_i = X^i + Y^i + Z^i$ en $T_i = S_i + (S_1)^i$, voor $i > 0$. Zij q en r machten van 2. Er geldt

$$T_{q+1}^{r+1} + T_{r+1}^{q+1} = T_{qr+1} T_{q+r}.$$

III.

Zij K de Klein kromme, gedefinieerd door $K : X^3Y + Y^3Z + Z^3X = 0$ over het lichaam der rationale getallen. De 24 flexpunten van K bevinden zich in de doorsnijding met de kromme $H : X^5Z + Y^5X + Z^5Y - 5X^2Y^2Z^2 = 0$. Zij K^* de duale kromme van K. Na reductie modulo $p = 2$, factoriseert het morfisme $K \longrightarrow K^*$ als

$$K \xrightarrow{sep} H \xrightarrow{insep} K^*.$$

(In [Ha,p.305] wordt opgemerkt dat na reductie modulo $p = 3$, het morfisme $K \longrightarrow K^*$ volledig inseparabel is.)

[Ha] Hartshorne, R., Algebraic geometry. New York: Springer-Verlag, 1977.

IV.

De kromme met affiene vergelijking $y^2 + y = x^5$ over het lichaam $GF(16)$ heeft 32 eindige rationale punten en een punt P_∞ in oneindig. De ondergroep van de Picard groep voortgebracht door de divisoren van graad nul is elementair abels van orde 625. De elementen van de vorm $[P - P_\infty]$ en hun tweevouden, met P een eindig rationaal punt, vormen een klasse in een partitie design van regulariteit vier. Dit impliceert voor algebraïsch-meetkundige codes gedefinieerd met de kromme en de 32 rationale punten, dat ten hoogste zes verschillende gewichtsverdelingen optreden bij codes van een gegeven dimensie.

[Ca] Camion, P., Courteau, B., and Delsarte, P.,
On r-partition designs in Hamming spaces,
Applicable Algebra in Eng., Commun. and Comput., vol.2, pp.147-162, 1992.

V.

De Fermat kromme van graad m over het lichaam $GF(q^2)$ bevat ten hoogste $q^2 + 1 + (m-1)(m-2)q$ rationale punten. Het maximum wordt bereikt als $m \mid q+1$. Een elementair bewijs wordt gegeven in [We]. De classificatie van supersinguliere Fermat variëteiten [Sh] laat zien dat de voorwaarde $m \mid q+1$ noodzakelijk is voor het bereiken van het maximum. Een elementair bewijs van de noodzakelijkheid wordt gegeven in [Du]. De tabel in [Se] van maximale Fermat krommen van graad $m \leq 7$ blijkt bij toetsing aan het criterium $m \mid q+1$ niet compleet. De versie van dezelfde tabel in [Go, p.130] bevat enkele krommen die niet maximaal zijn.

[We] Weil, A., Numbers of solutions of equations in finite fields, Bull. Am. Math. Soc., vol.55, pp.497-508, 1949.

[Sh] Shioda, T., and Katsura, T., On Fermat varieties, Tôhoku Math. J., vol.31, pp.97-115, 1979.

[Du] Duursma, I.M., Afstudeerverslag, Universiteit van Amsterdam, 1989.

[Se] Segre, B., Arithmetische Eigenschaften von Galois-Räumen I, Mathematische Annalen, vol.154, pp.195-256, 1964.

[Go] Goppa, V.D., Geometry and Codes. Dordrecht: Kluwer, 1988.

VI.

Zij $\pi : Y \longrightarrow X$ een Galois overdekking met groep G van krommen over een eindig lichaam k. Voor een ondergroep H van G beschouwen we de kromme Y/H. Zij $Pic_0(Y/H)$ het homogene deel van de Picard groep van een kromme Y/H. Zij

$$\{\ ,\ \}_{H,m} \ : \ Pic_0(Y/H)_m \times Pic_0(Y/H)/mPic_0(Y/H) \longrightarrow k/k^m$$

de Tate paring, gedefinieerd als in [Fr]. De groep $Pic_0(Y)$ wordt op de groep $Pic_0(Y/H)$ afgebeeld door restrictie. Op $Pic_0(Y)_m \times Pic_0(Y)/mPic_0(Y)$ definieren we een samengestelde afbeelding

$$\langle\ ,\ \rangle_{H,m} = \{\ ,\ \}_{H,m} \cdot (Res, Res).$$

Zij $\epsilon_H \in Q[G]$ de idempotent van een ondergroep H van G, gedefinieerd als in [Ka]. Laat een relatie op de idempotenten gegeven zijn door $\sum_H a_H \epsilon_H = 0$. Er geldt

$$\prod_H \langle\ ,\ \rangle_{H,m}^{a_H} \ = \ 1.$$

[Fr] Frey, G., and Rück, H., A Remark Concerning m-Divisibility and the Discrete Logarithm in the Divisor Class Group of Curves. Essen: Institut für Experimentelle Mathematik, 1991.

[Ka] Kani, E., and Rosen, M., Idempotent relations and factors of Jacobians, Mathematische Annalen, vol.284, pp.307-327, 1989.

VII.

Het is bekend dat de zeta functie $Z_2(t)$ van een kromme gedefinieerd over een kwadratisch eindig lichaam factoriseert als $Z_2(t^2) = Z_1(t)Z_1(-t)$. De kromme met affiene vergelijking $y^2 = x^p - x + 1$ over het lichaam $GF(p^2)$, p een oneven priemgetal, heeft zeta functie

$$Z(t) = \frac{(1 \pm p^p t^p)/(1 \pm pt)}{(1-t)(1-p^2 t)}.$$

Het plusteken geldt als $p \equiv 3 \pmod 4$ en het minteken als $p \equiv 1 \pmod 4$. De factorisatie van $Z(t^2)$ is een speciaal geval van een Aurifeuillian factorisatie.

[Sc] Schinzel, A., On primitive factors of $a^n - b^n$,
Proc. Cambridge Philos. Soc., vol.58, pp.555-562, 1962.

[St] Stevenhagen, P., On Aurifeuillian factorizations,
Proc. Kon. Ned. Akad. van Wetenschappen, vol.90, pp.451-468, 1987.

VIII.

Zij $f = (X-\alpha_1)(X-\alpha_2)(X-\alpha_3)(X-\alpha_4) \in k[X]$ een deler van $X^{41}-1$, met $k = GF(2^{20})$. Zij $S_i = \alpha_1^i + \alpha_2^i + \alpha_3^i + \alpha_4^i$, $i \geq 0$. Het is bekend dat f volledig wordt bepaald door de waarde van S_1. Met $S_0 = 0$, $S_{2i} = {S_i}^2$ en $S_{i+41} = S_i$ zijn alle elementen in de volgende matrix te berekenen, met uitzondering van S_3.

$$\begin{pmatrix} S_0 & S_{31} & S_{37} & S_{23} & S_1 \\ S_8 & S_{39} & S_4 & S_{31} & S_9 \\ S_{20} & S_{10} & S_{16} & S_2 & S_{21} \\ S_9 & S_{40} & S_5 & S_{32} & S_{10} \\ S_2 & S_{33} & S_{39} & S_{25} & S_3 \end{pmatrix}$$

Singulariteit van de matrix en regulariteit van de minor buiten S_3 geeft een vergelijking voor S_3. Een alternatieve berekening van S_3 wordt gegeven in [Re].

[Du] Duursma, I.M., and Kötter, R., Error-locating pairs for cyclic codes.
Eindhoven-Linköping: preprint, 1993.

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

IX.

Zij gegeven een nevenklasse $\mathbf{y} + C$ van een algebraïsch-meetkundige code C. De oplossingsruimte van het basis decodeeralgoritme bevat functies die nul zijn op de support van een element uit de nevenklasse. Laat de nevenklasse een tweetal vectoren $\{\mathbf{e}_1, \mathbf{e}_2\}$ bevatten met de eigenschap dat het gezamenlijke aantal coördinaten dat verschilt van nul gelijk is aan de ontwerp minimumafstand van de code. De oplossingen van het basis decodeeralgoritme zijn dan te schrijven als lineaire combinatie van een functie die nul is op de niet-nul coördinaten van de eerste vector en een functie die nul is op de niet-nul coördinaten van de tweede vector. In de notatie van dit proefschrift: $K(F) = L(F - Q_1) + L(F - Q_2)$.