A Study on Some Properties of the Feng-Rao Designed Minimum Distance of Binary Linear Codes and Cyclic Codes

Junru Zheng

March 2002

Abstract

The minimum distance of an error correcting code is the most important parameter for evaluating its error correcting ability. The BCH bound, Hartmann-Tzeng bound, Roos bound and Shift bound are the most popular lower bounds for the minimum distance of cyclic codes. It is interesting to find relations between the well-known bounds. The Feng-Rao designed minimum distance and the Feng-Rao decoding were originally introduced into algebraic geometry codes. They have been extended to the case of general linear codes over a finite field by Miura. And recently the definition of the Feng-Rao designed minimum distance of linear codes has been slightly generalized by Matsumoto. According to the definition by Miura the value of the Feng-Rao designed minimum distance for a linear code C over a finite field F_q depends on the choice of the ordered basis of F_q^n, a vector space consisting of all the n-tuples over F_q, used for defining the code C. In practical applications of the Feng-Rao decoding algorithm, it is important to find such an optimum ordered basis for a given linear code.

This dissertation contains the discussion about unknown relation between Roos bound and Shift bound from numerical experiments. It is shown that the Feng-Rao designed minimum distance of binary linear codes can not take an odd value except one, if we use Miura's definition. Matsumoto gave a generalization of Miura's definition with three ordered bases. We have Miura's definition if three ordered bases are same in Matsumoto's definition. Our numerical experiments suggest us conjectures, that Matsumoto's generalization is not so effective for binary linear codes compared with Miura's definition. Other results of this dissertation are investigations for nonbinary cyclic codes and binary cyclic codes. The Type I ordered basis B_n was introduced as a very natural candidate necessary for computing d_{FR}. The ordered basis B_n is Type I if its subset B consists $n-k$ row vectors of the permutation of the usual parity check matrix defined by parity check polynomial of cyclic codes. It was shown that the choice of an ordered basis with Type I is worst in many cases of nonbinary cyclic codes, since the Feng-Rao designed minimum distance is equal to 1 if the check polynomial has a coefficient neither equal to 0 nor 1. It is also shown that in case of binary (n,k) cyclic codes C with $k = 1, 2$, and $n-1$, there exists an ordered basis with Type I such that the Feng-Rao designed minimum distance is equal to $n-1$, $2(\frac{n}{3}-1)$, and 2, respectively.

Acknowledgments

I would like to express my gratitude, and thanks to Professor Kyoki Imamura, the chairman of my dissertation committee, for many advice and the consistent guidance of the research effort, for introducing several research problems and giving his time to me for discussion.

I would like to express my gratitude, and thanks to Doctor Takayasu Kaida with Yatsushiro National College of Technology for many good advice of the research effort, introducing many research problems and giving his a great many times to me for discussion.

I would also like to thank Professor Yuji Oie, Professor Tsutomu Sasao and Professor Takeshi Shinohara, the other thesis committee at Kyushu Institute of Technology. Their comments are very useful for the improvement of presentation.

I would like to thank assistant Professor Satoshi Uehara with The University of Kitakyushu, former assistant at Imamura Laboratory Kyushu Institute of Technology, for advice and helps.

I would like to thank assistant Shunsuke Araki and the members of Imamura Laboratory for advice and helps.

I would like to thank Professor Fumikazu Tamari with Fukuoka University of Education, the supervisor in my muster course, for bringing me this area.

Finally, I would like to thank my parents for their continuous help. I would thank my husband Shuyin Liu for his continuous economical and moral support. And I should thank my baby Masakazu Liu, his lovely actions and filial piety are the best encouragement for me while I was passing at my study.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

The minimum distance of an error correcting code is the most important parameter for evaluating its error correcting ability. For a cyclic code C, Bose-Chaudhuri-Hocquenghem (BCH) bound [1, 2, 6], Hartmann-Tzeng(HT) bound [5], Roos bound [10], the Shift bound by van Lint, Wilson and van Eupen [11] are well-known lower bounds for the minimum distance. It is known that the relation between BCH bound, HT bound and Roos bound, and the relation between HT bound and Shift bound. However it is not known the relation between Roos bound and Shift bound.

A good decoding algorithm for error-correcting codes has a large designed minimum distance d^*, since the algorithm can correct $\lfloor (d^* - 1)/2 \rfloor$ or fewer random errors. In case of algebraic geometry codes, a subclass of linear codes, the Feng-Rao decoding algorithm [3] is known as the best one and has a large designed minimum distance called the Feng-Rao designed minimum distance d_{FR}. The Feng-Rao designed minimum distance and the Feng-Rao decoding algorithm were generalized by Miura [8] to the more general case of linear codes over F_q, a finite field with order q, together with the definition of d_{FR}. According to the definition by Miura the value of d_{FR} for an (n, k) linear code C over F_q depends on the choice of the ordered basis B_n of F_q^n, a vector space consisting of all the n-tuples over F_q, used for defining the code C. In practical applications of the Feng- Rao decoding algorithm, it is important to find such an optimum ordered basis B_n for a given (n, k) linear code C as $d_{FR}(C, B_n)$ takes the maximum value $d_{FR}(C)$.

Miura's definition of the Feng-Rao designed minimum distance d_{FR} for an (n, k) linear code C over a finite field F_q of order q depends on the choice of an ordered basis $B_n =$

$\{\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_n\}$ of the vector space F_q^n with dimension n over F_q. It is interesting to find such an optimum ordered basis B_n as d_{FR} is maximum, since the Feng-Rao decoding algorithm can correct up to $\lfloor (d_{FR}(C, B_n) - 1)/2 \rfloor$ errors. It is shown that some properties of the Feng-Rao designed minimum distance d_{FR} by Miura for binary linear codes and cyclic codes [13, 19, 15].

Recently the definition of d_{FR} of linear codes has been slightly generalized by Matsumoto [7], which uses three ordered bases $U_n = \{\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_n\}$, $V_n = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_n\}$ and $B_n = \{\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_n\}$ of F_q^n instead of one in case of Miura's definition, i.e., B_n is used for defining the linear code, and U_n, V_n are used for computing a syndrome matrix in Matsumoto's definition. Hence Miura's definition is included by assumption $U_n = V_n = B_n$ in Matsumoto's definition.

1.2 In This Dissertation

The minimum distance of an error correcting code is the most important parameter for evaluating its error correcting ability. In case of algebraic geometry codes, a subclass of linear codes, the Feng-Rao decoding algorithm is known as the best one and has a large the Feng-Rao designed minimum distance. This dissertation deals with some properties of the Feng-Rao designed minimum distance by Miura and Matsumoto of binary linear codes and cyclic codes.

Chapter 2 describes briefly construction of the communication system. The linear codes are mostly studied, because they are easier to describe, encode, and decode than nonlinear codes. The cyclic codes include the family of BCH codes are important subclass of linear codes. The properties of linear codes and cyclic codes are presented in Chapter 2.

In Chapter 3, the relation between well-known designed minimum distance of cyclic codes, such as Bose-Chaudhuri-Hocquenghem bound, Hartmann-Tzeng bound, Roos bound and Shift bound are presented. The unknown relation between Roos bound and Shift bound are shown from numerical experiments. This chapter deals with many examples and tables of binary cyclic codes of $n \leq 31$ and ternary cyclic codes of $n \leq 26$.

Chapter 4 is mainly concerned with introductions to the Feng-Rao designed minimum distance and the Feng-Rao decoding of linear codes. The fundamental iterative algorithm and its extension, which are effective algorithms for decoding linear codes up to the designed minimum distance d_{FR}, are introduced. Miura's definition of d_{FR} is described,

and some properties are listed. Moreover the Feng-Rao decoding method is introduced for linear codes.

Chapter 5 deals with the Feng-Rao designed minimum distance d_{FR} of binary linear codes. For an (n, k) linear code over a finite field F_2, Miura's definition of d_{FR} denoted by $d_{FR}(C, B_n)$ depends on the choice of an ordered basis $B_n = \{\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_n\}$ of the vector space F_2^n with dimension n over F_2, and the ordering of n vectors $\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_n$ has meaning. It is interesting to find such an optimum ordered basis B_n as d_{FR} is maximum, since the Feng-Rao decoding can correct up to $\lfloor (d_{FR}(C, B_n) - 1)/2 \rfloor$ errors. In this chapter $d_{FR}(C, B_n)$ of binary linear codes can not take an odd value except one if we use Miura's definition of $d_{FR}(C, B_n)$ is proved.

In Chapter 6, Matsumoto's definition of d_{FR} for linear codes is considered. The definition of d_{FR} of linear codes has been slightly generalized by Matsumoto denoted by $\hat{d}_{FR}$, that uses three ordered bases $U_n = \{\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_n\}, V_n = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_n\}$ and $B_n = \{\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_n\}$ of F_q^n instead of one ordered basis in case of Miura's definition, i.e., B_n is used for defining the linear code, and U_n, V_n are used for computing a syndrome matrix in Matsumoto's definition and Miura's definition is included by assumption $U_n = V_n = B_n$. In this chapter we give conjectures that Matsumoto's generalization is not so effective for binary linear codes compared with Miura's definition from some properties and some numerical examples of Matsumoto's $\hat{d}_{FR}$.

Chapter 7 deals with the Feng-Rao designed minimum distance of cyclic codes. The "Type I" ordered basis B_n is defined, that corresponds to the well-known form of the parity check matrix of an (n, k) cyclic code expressed by its parity check polynomial, i.e., we use a natural choice of B, a subset of B_n, as $(n - k)$ vectors $\{\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_{n-k}\}$, which consists of a permutation of vectors corresponding to the coefficient of the check polynomial and its $(n - k - 1)$ consecutive right cyclic shifts. The possible values of d_{FR} of an (n, k) cyclic code are investigated when the Type I ordered basis B_n is used. For nonbinary cyclic codes $d_{FR}(C, B_n) \leq 1$ is proved, if B_n is Type I and the check polynomial $h(x)$ has a coefficient $\neq 0, 1$. Moreover this chapter shows that, for binary cyclic codes, $d_{FR}(C, B_n) = n - 1$ for $k = 1$ (repetition code), $d_{FR}(C, B_n) \geq 2(\frac{n}{3} - 1)$ for $k = 2$ if B_n is Type I, and $d_{FR}(C, B_n) = 2$ for $k = n - 1$ (parity code), respectively.

Finally conclusions and future works are listed in Chapter 8.

Chapter 2

Preparations

2.1 Introduction

Codes were invented to correct errors on noisy communication channels. Suppose there is a telegraph wire from one place to the other place down, which 0's and 1's can be sent. Usually when a 0 is sent it is received as a 0, or when a 1 is sent it is received as a 1, but occasionally a 0 will be received as a 1, or a 1 as a 0. Let's say that on the average p symbols will be in error, i.e.,for each symbol there is an error probability p that the channel will make a mistake. This is called a *binary symmetric channel* (Fig. 2.1).

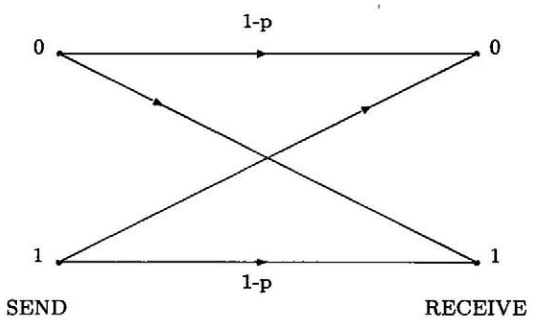

Figure 2.1: The Binary Symmetric Channel

There are a lot of important messages to be sent down this wire, and they must be sent as quickly and reliably as possible. The messages are already written as a string of 0's and 1's perhaps they are being produced by a computer.

We are going to encode these messages to give them some protection against errors on

the channel. A block of message symbols $\boldsymbol{u} = (u_1, u_2, \cdots, u_k)$ where $u_i = 0$ or 1 will be encoded into a *codeword* $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$ where $x_i = 0$ or 1, $n \geq k$ (Fig.2.1); these codewords form a *code.*

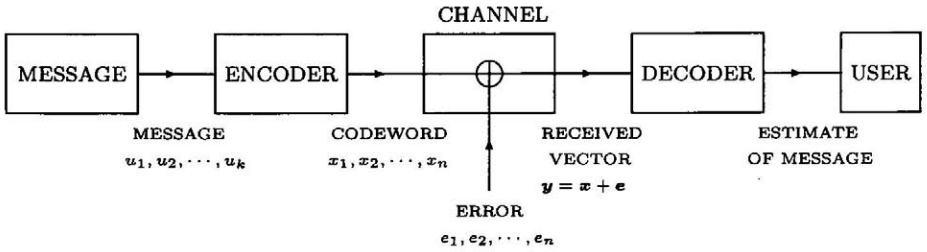

Figure 2.2: The Communication System

In the method of encoding we are about to describe produces what is called a *linear code.* The first part of the codeword consists of the message itself:

$$x_1 = u_1, x_2 = u_2, \cdots, x_k = u_k,$$

followed by $n - k$ check symbols

$$x_{k+1}, x_{k+2}, \cdots, x_n.$$

Suppose the message $\boldsymbol{u} = (u_1, u_2, \cdots, u_k)$ where $u_i = 0$ or 1 is encoded into the codeword $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$, which is then sent through the channel. Because of channel noise, the received vector $\boldsymbol{y} = (y_1, y_2, \cdots, y_n)$ may be different from $\boldsymbol{x}$. Let's define the *error vector*

$$\boldsymbol{e} = \boldsymbol{y} - \boldsymbol{x} = (e_1, e_2, \cdots, e_n).$$

Then $e_i = 0$ with probability $1 - p$ (and the i-th symbol is correct), and $e_i = 1$ with probability p (and the i-th symbol is wrong). So we describe the action of the channel by saying it distorts the codeword $\boldsymbol{x}$ by adding the error vector $\boldsymbol{e}$ to it.

The decoder must decide from $\boldsymbol{y}$ which message $\boldsymbol{u}$ or which codeword $\boldsymbol{x}$ was transmitted. Of course it's enough if the decoder find $\boldsymbol{e}$, for then $\boldsymbol{x} = \boldsymbol{y} - \boldsymbol{e}$. Now the decoder can never be certain what $\boldsymbol{e}$ was. His strategy therefore will be to choose the *most likely* error vector $\boldsymbol{e}$, given $\boldsymbol{y}$ was received. Provided the codewords are all equally likely, this strategy is optimum in the sense that it minimizes the probability of the decoder making a mistake, and is called *maximum likelihood decoding.*

In the next section the linear codes will be stated. And in the last section cyclic codes will be explained.

2.2 Linear Code

Among all types of block codes, linear codes are mostly studied. Because of their algebraic structure, they are easier to describe, encode, and decode than nonlinear codes.

Let F_q^n denote the linear space of all n-tuples over a finite field F_q. A code C with code length n over F_q is a subset of F_q^n. If C is a k dimensional subspace of F_q^n, then C will be called an (n, k) linear code over F_q. We usually write the vectors in F_q^n as words $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$ over the alphabet F_q and call a vector in C a codeword. The field F_2 is very special in coding theory, and codes over F_2 are called *binary codes*. Codes over F_3 are called *ternary codes*. The two most common ways to present a linear code are by a generator matrix and by a parity check matrix.

A *generator matrix* for a linear code C is any $k \times n$ matrix G whose rows form a basis for C. For any set of k independent rows of a generator matrix G, the corresponding set of coordinates forms an information set for C.

Recall that the ordinary inner product of vectors $\boldsymbol{u} = (u_1, u_2, \cdots, u_n)$ and $\boldsymbol{v} = (v_1, v_2, \cdots, v_n)$ in F_q^n is

$$\langle \boldsymbol{u} \cdot \boldsymbol{v} \rangle = \sum_{i=1}^{n} u_i v_i$$

The dual of C is the $(n, n-k)$ linear code $C^{\perp}$ defined by

$$C^{\perp} = \{\boldsymbol{v} \in F_q^n \mid \langle \boldsymbol{u} \cdot \boldsymbol{v} \rangle = 0 \text{ for all } \boldsymbol{u} \in C\}.$$

An $(n-k) \times n$ generator matrix H of $C^{\perp}$ is called a *parity check matrix* for C. So

$$C = \{\boldsymbol{x} \in F_q^n \mid \boldsymbol{x} H^T = 0\}.$$

The *Hamming distance* $\mathrm{d}(\boldsymbol{x}, \boldsymbol{y})$ between two vectors $\boldsymbol{x}$, $\boldsymbol{y} \in F_q^n$ is defined to be the number of coordinates in which $\boldsymbol{x}$ and $\boldsymbol{y}$ differ, i.e.,

$$\mathrm{d}(\boldsymbol{x}, \boldsymbol{y}) = \sharp\{i \mid x_i \neq y_i, 1 \leq i \leq n\},$$

where $\sharp A$ means the cardinality of set A. Distance is a metric on the linear space F_q^n. The *minimum distance* of a code C is the smallest distance between distinct codewords and it

is important in determining its error correcting capabilities. The Hamming weight $\mathrm{wt}(\boldsymbol{x})$ of a vector $\boldsymbol{x} \in F_q^n$ is the number of its nonzero coordinates. Clearly, $\mathrm{d}(\boldsymbol{x}, \boldsymbol{y}) = \mathrm{wt}(\boldsymbol{x} - \boldsymbol{y})$. Thus if C is a linear code, the minimum distance d is the same as the minimum weight of a nonzero codeword for all codewords except zero codeword. If the minimum distance d of an (n, k) code is known, then we refer to the code as an (n, k, d) code.

If the minimum distance of C is d, there exist two distinct codewords such that the spheres of radius $t + 1$ about them are not disjoint, then the packing radius t so that the spheres about the codewords are pairwise disjoint, equals $\lfloor (d-1)/2 \rfloor$. The packing radius t of a code is characterized by the property that nearest neighbor decoding always decodes correctly a received vector in which t, or fewer errors have occurred but will not always decode correctly a received vector in which $t+1$ errors have occurred. Thus C is a *t-error correcting code* but not a $(t+1)$-error correcting code. One way to find a closest codeword to a received vector $\boldsymbol{y}$ is to examine all codewords until one is found with distance t or less from $\boldsymbol{y}$. But obviously this is a realistic decoding algorithm only for codes with a small number of vectors. General decoding algorithm are discussed in Section 4.4.

The minimum distance d is a simple measure of the goodness of a code. For a given length and number of codewords (equivalently, dimension in the case of linear codes), a fundamental problem in coding theory is to determine a code with the largest d.

2.3 Cyclic Code

Cyclic codes are mostly studied of all codes, since they are easy to encode, and include the important family of BCH codes.

A linear code C of length n over F_q is *cyclic* if the vector $(c_{n-1}c_0 \cdots c_{n-2})$ obtained from codeword $\boldsymbol{c} = (c_0 c_1 \cdots c_{n-2} c_{n-1})$ in C by the *cyclic shift* of coordinates $i \mapsto i + 1$ is also in C. Cyclic codes and certain codes related to them are some of the most useful codes known. In particular the Golay codes and the binary Hamming codes can be represented as cyclic codes.

There is a bijective correspondence between the vectors

$$\boldsymbol{c} = (c_0 c_1 \cdots c_{n-1})$$

in F_q^n and the polynomials

$$c(x) = c_0 + c_1 x + \cdots + c_{n-1} x^{n-1}$$

in $F_q[x]$ of degree at most $n-1$. We allow ourselves the latitude of using the vector notation $\boldsymbol{c}$ and the polynomial notation $c(x)$ interchangeably. The fact that a cyclic code C is invariant under a cyclic shift implies that if $c(x)$ is in C then so is $xc(x)$ provided we multiply modulo x^n-1. This suggests that the proper context for studying cyclic codes is the subset of the residue class ring

$$R_n = F_q[x]/(x^n-1).$$

Under the correspondence of vectors with polynomial as given above, cyclic codes correspond bijectively to the ideals of R_n.

To distinguish the ideals $(g(x))$ of $F_q[x]$ from those of R_n, we use the notation $\langle g(x)\rangle$ for the ideal of R_n generated by $g(x)$, where $g(x)$ is called a *generator polynomial* of the ideal. Let C be a nonzero cyclic code in R_n. There exists a polynomial $g(x) \in C$ with the following properties.

1 . $g(x)$ is the unique monic polynomial of minimum degree in C.

2 . $C = \langle g(x)\rangle$.

3 . $g(x) \mid (x^n-1)$.

Let $k = n - \deg(g(x))$, and let $g(x) = \sum_{i=0}^{n-1} g_i x^i$ where $g_{n-k+1} = \cdots = g_{n-1} = 0$. Then

4 . the dimension of C is k and $\{g(x), xg(x), \cdots, x^{k-1}g(x)\}$ is a basis for C, i.e.,

$$G = \begin{bmatrix} g_0 & g_1 & g_2 & \cdots & g_{n-k} & 0 & 0 & \cdots & 0 \\ 0 & g_0 & g_1 & \cdots & g_{n-k-1} & g_{n-k} & 0 & \cdots & 0 \\ 0 & 0 & g_0 & \cdots & g_{n-k-2} & g_{n-k-1} & g_{n-k} & \cdots & 0 \\ & & & \vdots & & & & & \\ 0 & 0 & 0 & \cdots & & & & \cdots & g_{n-k} \end{bmatrix}$$

$$= \begin{bmatrix} g(x) \\ xg(x) \\ \vdots \\ x^{k-1}g(x) \end{bmatrix}$$

is a generator matrix for C.

Let C be a cyclic code with generator polynomial $g(x)$. Then,

$$h(x) = (x^n-1)/g(x) = \sum_{i=0}^{k} h_i x^i, \quad (h_k \neq 0),$$

is called the *check polynomial* of C. Let

$$H = \begin{bmatrix} h_k & h_{k-1} & \cdots & h_0 & 0 & 0 & \cdots & 0 \\ 0 & h_k & \cdots & h_1 & h_0 & 0 & \cdots & 0 \\ & & \vdots & & & & & \\ 0 & 0 & h_k & \cdots & & & & h_0 \end{bmatrix}$$

using an obvious notation. If $\boldsymbol{x} \in C$ then $\boldsymbol{x}H^T = 0$. Since

$$k = \deg(h(x)) = n - \deg(g(x)) = \dim(C),$$

and the rows of H are obviously linearly independent, the condition $\boldsymbol{x}H^T = 0$ is also sufficient for $\boldsymbol{x}$ to be in the code. Thus H is a parity check matrix for C.

Chapter 3

Designed Minimum Distances of Cyclic Codes

3.1 Introduction

It is surveyed various bounds on the minimum distance of cyclic codes in this chapter. For a cyclic code C the Bose-Chaudhuri-Hocquenghem bound $d_{BCH}(C)$ [1, 2, 6], Hartmann-Tzeng bound $d_{HT}(C)$ [5], Roos bound $d_R(C)$ [10], the Shift bound $d_S(C)$ by van Lint, Wilson and van Eupen [11] are well-known lower bounds for the minimum distance . It is known that $d_{BCH}(C) \leq d_{HT}(C) \leq d_R(C)$ and $d_{HT}(C) \leq d_S(C)$. However the relation between $d_R(C)$ and $d_S(C)$ dose not be known. In this chapter we will discuss this relation with some examples by author's computing program.

In this chapter, firstly it is referred to the definition of BCH bound $d_{BCH}(C)$, Hartmann-Tzeng bound $d_{HT}(C)$, Roos bound $d_R(C)$ and the Shift bound $d_S(C)$ for a cyclic code C. Secondly the relations between designed minimum distances will be discussed from numerical experiments.

3.2 Well-Known Designed Minimum Distances

Some lower bounds will be introduced on the minimum distance of cyclic codes in this section. A finite filed is denoted by F and the multiplicative group of non-zero elements is denoted by F^*. The finite filed with q elements is denoted by F_q.

A cyclic code C of length n over F_q will be identified with the corresponding ideal in the ring $F_q[x]/(x^n - 1)$. This ideal C is generated by a polynomial $g(x)$ which divides

$x^n - 1$. The *true minimum distance* of C is denoted by d. If α is a primitive nth root of unity in an extension field F_{q^m} of F_q, then $g(x)$ is a product of polynomials $m_i(x)$, where $m_i(x)$ denotes the minimal polynomial for α^i over F_q. The following set

$$R = \{i | g(\alpha^i) = 0, 0 \leq i \leq n-1\}$$

is called *defining set* of C with $g(x)$.

A well-known lower bound for the minimum distance of cyclic codes is the so-called Bose-Chaudhuri-Hocquenghem (BCH) bound [6].

Definition 3.2.1 *(BCH bound) Let the largest $\delta \leq n$ such that $\{i, i+1, \cdots, i+\delta-2\} \subseteq R$ for some i, then $d_{BCH}(C) = \delta$.*

The BCH bound was generalized by Hartmann and Tzeng [5]. Their result which we call the *HT bound*, was slightly modified by Roos [10].

Definition 3.2.2 *(HT bound) Let the largest number $\delta + s$, such that there exist i, a and s with the property that $\gcd(a, n) = 1$ and $\{i + j + ka | 1 \leq j < \delta, 0 \leq k \leq s\} \subseteq R$, then $d_{HT}(C) = \delta + s$.*

Definition 3.2.3 *(Roos bound) Let b and n be two positive integers such that $\gcd(b, n) = 1$. If $A \subset R, B = \{i_1 b, i_2 b, \cdots, i_t b\}$ where $0 \leq i_1 < \cdots < i_t < n$, and $i_t - i_1 + 1 \leq t + d_A - 2$, then the true minimum distance of C is $d(C) \geq t + d_A - 1$. The Roos bound $d_R(C)$ is the largest number $t + d_A - 1$ such that there exist defining sets A and B, where $A + B \subseteq R$, and d_A is minimum distance of $C(A)$ over F_{q^m}, where $C(A)$ is defined by defining set A.*

Let $g(x)$ be the generator polynomial of cyclic code C, the check polynomial $h(x) = (x^n - 1)/g(x)$ consists of the produce of irreducible polynomial $h_1(x), h_2(x), \cdots, h_u(x)$. The Shift bound [12] of cyclic codes C is recursively defined as following.

Definition 3.2.4 *(Shift bound)*

1. *In case of $u = 1$ ($h(x)$ is irreducible polynomial):*

   Let R be defining set of C and $\{r_1, r_2, \cdots, r_w\} \subseteq R$, for w sequences $a_1, a_2, \cdots, a_w$, for $w = 1, 2, \cdots, n$, the largest number i such that

   $$\{a_i + r_1, a_i + r_2, \cdots, a_i + r_{i-1}\} \subset R, \quad a_i + r_i \notin R, \tag{3.1}$$

   we have $d_S(C) = w + 1$.

2. *In case of* $u > 1$*:*

   Let $h_0(x)|h(x), h_0(x) \neq h(x)$*, the cyclic code* C_0 *with generator polynomial* $g_0(x) = g(x)h_0(x)$ *is true subcode of* C*, we write* $C_0 < C$*. Let*

$$\bar{d}_S(C) = \min_{C_0<C} d_S(C_0).$$

   For $\{r_1, r_2, \cdots, r_w\} \subseteq R, w+1 \leq \bar{d}_S(C)$ *and* $i = 1, 2, \cdots, w$*, (3.1) is hold, for the largest number* w *such that there exist the sequences* $a_1, a_2, \cdots, a_w$*, we have* $d_S(C) = w + 1$*.*

3.3 Relations Between Designed Minimum Distances from Numerical Experiments

It follows directly from the definitions that the HT bound is a generalization of the BCH bound and the Roos bound is a generalization of the HT bound, so we have $d_{BCH} \leq d_{HT} \leq d_R$.

A set of integers is denoted by Z, and the integers modulo by n is denoted by Z_n.

Theorem 3.3.1 *The following inequalities hold*

$$d_S(C) \geq d_{HT}(C) \geq d_{BCH}(C).$$

Proof The last inequality is obvious. Let $J = \{i + j + ka | 1 \leq j < \delta, 0 \leq k \leq s\}$ be a subset of Z_n, and $J \subseteq R \neq Z_n$. Then there is a $\delta' \geq \delta$ such that $i+j \in R$ for all $1 \leq j < \delta'$ and $i + \delta' \notin R$. The set $\{i + j + ka | 1 \leq j < \delta, k \in Z_n\}$ is equal to Z_n, since $\gcd(a, n) < \delta$. So there exist $s' \geq s$ and j' such that $i + j + ka \in R$ for all $1 \leq j < \delta$ and $0 \leq k \leq s'$, and $1 \leq j' < \delta, i + j' + (s' + 1)a \notin R$. Let $w = \delta + s'$.

Let $i_k = (k-1)a$ for all $1 \leq k \leq s' + 1$, and $i_k = \delta' - \delta - s' - 1 + k$ for all k such that $s' + 2 \leq k \leq \delta + s'$. Let $j_l = i + l$ for all $1 \leq l \leq \delta - 1$, and let $j_l = i + j' + (l - \delta + 1)a$ for all l such that $\delta \leq l \leq \delta + s'$. Then one easily checks that $i_k + j_l \in R$ for all $k + l \leq w$, and $i_k + j_{w-k+1} = i + j' + (s' + 1)a \notin R$ for all $1 \leq k \leq s' + 1$, and $i_k + j_{w-k+1} = i + \delta' \notin R$ for all $s' + 2 \leq k \leq \delta + s'$. So we have a set which is independent with respect to R and has size $w = \delta + s' \geq \delta + s$.

From the definition of Shift bound, we have $d_S(C) > \delta + s$ for all defining sets R which contain J and are not equal to Z_n. Therefore $d_S(C) \geq d_{HT}(C)$. □

The Roos bound is a generalization of the HT bound, so $d_R(C) \geq d_{HT}(C)$. The next we discuss about the relation between d_R and d_S from numerical experiments that use the computing program of Roos bound and Shift bound by author.

Explanation of Computing Program

It is introduced the method for computing program of Roos bound and Shift bound as follows.

1. For the Computing Program of Roos Bound

   Let R be defining set, and $A+B \subseteq R$. We choice the set A is the subset of R, which the largest continuous elements, such that $d_A = d_{BCH}(C)$. From $A + B \subseteq R$, we decide the set B. Let ib for $0 \leq i < n$, we choice the elements that belong to set B by the order of i, where $b \in \{0, 1, \cdots, n-1\}, \gcd(b, n) = 1$. We choice B that satisfies $i_t - i_1 + 1 \leq t + d_A - 2$, and compute $d_R(C) = t + d_A - 1$.

2. For the Computing Program of Shift Bound

   It is made out computing program that search all sequences satisfying the condition of definition.

In many cases of binary codes of length at most 62, the Shift bound is equal to the true minimum distance [12]. In about 95% of all ternary codes of length at most 40, the Shift bound is equal to the true minimum distance [11].

3.3.1 For Binary Cyclic Codes of $n \leq 31$

In case of $d_R(C) < d_S(C)$ we can find $d_R(C) = d_{BCH}(C)$ or $d_R(C) = d_{HT}(C)$.

Example 3.3.1 *Let* $n = 21, R = \{0, 1, 2, 3, 4, 6, 7, 8, 11, 12, 14, 16\}$, *we have* $d_{BCH}(C) = d_{HT}(C) = d_R(C) = 6$, *and* $d_S = 8$. *The minimum distance of this cyclic code is 8.*

Example 3.3.2 *Let* $n = 31, R = \{1, 2, 4, 7, 8, 14, 16, 19, 25, 28\}$, *we have* $d_{BCH}(C) = 3, d_{HT}(C) = d_R(C) = 4$ *and* $d_S = 5$. *The minimum distance of this cyclic code is 5.*

In case of $d_R = d_S$, we have $d_{BCH}(C) = d_R(C) = d_S(C)$ or $d_{HT}(C) = d_R(C) = d_S(C)$ in many cases. We show the example which is not so, i.e.,in case of $d_{BCH}(C) < d_{HT}(C) < d_R(C) = d_S(C)$.

Example 3.3.3 *Let* $n = 21, R = \{1, 2, 3, 4, 6, 7, 8, 9, 11, 12, 14, 15, 16, 18\}$. *We have* $d_{BCH}(C) = 5, d_{HT}(C) = 6$. *Let* $A\{1, 2\}, b = 4, B = \{11, 12, 13, 14, 16, 17\}$, *we have* $d_R(C) = 8, d_S(C) = 8$. *The minimum distance of this cyclic code is 8.*

For the binary cyclic codes of $n \leq 31$, about 25% is $d_R(C) < d_S(C)$, about 75% is $d_R(C) = d_S(C)$, and there is no case of $d_R(C) > d_S(C)$. Therefore we have $d_R(C) \leq d_S(C)$ for binary cyclic codes of $n \leq 31$.

3.3.2 For Ternary Cyclic Codes of $n \leq 26$

All ternary cyclic codes of $n < 26$ are $d_R(C) \leq d_S(C)$. There are two examples with $d_R(C) > d_S(C)$. One of the example is in [11], and the other one is found by author.

Example 3.3.4 *[11] For* $n = 26, R = \{0, 13, 14, 16, 17, 22, 23, 25\}$, *let* $A = \{13, 14\}, b = 3, B = \{0, 1, 3, 4\}$, *we have* $d_R = 6$ *and* $d_S = 5$. *The minimum distance of this cyclic code is 6.*

Example 3.3.5 *For* $n = 26, R = \{0, 5, 8, 13, 14, 15, 16, 17, 19, 20, 22, 23, 24, 25\}$, *let* $A = \{13, 14, 15\}, b = 9, B = \{0, 1, 3, 4, 6, 7\}$, *we have* $d_R = 9$ *and* $d_S = 8$. *The minimum distance of this cyclic code is 9.*

For the ternary cyclic codes of $n \leq 26$, about 32% is $d_R(C) < d_S(C)$, about 67% is $d_R(C) = d_S(C)$, and there are two examples with $d_R(C) > d_S(C)$.

Explanation of Tables

It is computed all binary cyclic codes of $n \leq 31$, and all ternary cyclic codes of $n \leq 26$ by author's computing program. In many cases we have $d_R(C) = d_S(C)$, then the tables only give the case of $d_R(C) < d_S(C)$. And the tables give the code length n, dimension k, Roos bound $d_R(C)$, Shift bound $d_S(C)$, and the set G such that $g(x) = \prod_{i \in G} m_i(x)$, in other words, $R = \{\alpha^i | i \in G\}$ is a defining set.

3.4 Conclusion

It is surveyed various bounds on the minimum distance of cyclic codes. For cyclic codes the BCH bound $d_{BCH}(C)$, HT bound $d_{HT}(C)$, Roos bound $d_R(C)$ and the Shift bound $d_S(C)$ for the minimum distance are well-known. It is known that $d_{BCH}(C) \leq d_{HT}(C) \leq d_R(C)$

and $d_{HT}(C) \leq d_S(C)$. But the relation between $d_R(C)$ and $d_S(C)$ is not known. It is discussed with some examples in this chapter.

For all binary cyclic codes of $n \leq 31$, about 25% is $d_R(C) < d_S(C)$, about 75% is $d_R(C) = d_S(C)$, and there is no case of $d_R(C) > d_S(C)$. Therefore we have $d_R(C) \leq d_S(C)$ for binary cyclic codes of $n \leq 31$.

For all ternary cyclic codes of $n \leq 26$, about 32% is $d_R(C) < d_S(C)$, about 67% is $d_R(C) = d_S(C)$, and there are two examples with $d_R(C) > d_S(C)$.

Table 3.1: Binary Cyclic Codes of $n \leq 31$

NO.	n	k	$d_R(C)$	$d_S(C)$	G
1	21	15	3	4	0,3,7
2			3	4	0,7,9
3		9	5	6	1,3,9
4			6	8	0,1,3,7
5		6	7	8	1,3,5
6	23	12	5	6	1
7		11	6	7	0,1
8	31	21	4	5	1,5
9			4	5	1,7
10			4	5	3,7
11			4	5	3,11
12		20	5	6	0,1,5
13			4	6	0,1,7
14		16	6	7	1,5,7
15		15	6	7	0,1,5,7

Table 3.2: Ternary Cyclic Codes of $n \leq 26$

NO.	n	k	$d_R(C)$	$d_S(C)$	G
1	13	6	5	6	0,1,2
2			5	6	0,1,7
3			5	6	0,2,7
4	16	9	4	5	0,1,10
5			4	5	1,2,8
6			4	5	0,2,5
7		7	5	6	1,2,4,8
8	20	15	3	4	0,1
9			3	4	0,11
10		13	3	4	0,1,5
11			3	4	1,5,10
12		10	5	6	0,1,4,10
13			5	6	0,1,2,10
14		9	5	6	1,4,5,10
15			5	6	0,1,2,5
16		8	5	8	0,1,2,5,10
17			5	8	0,1,4,5,10
18	22	16	3	4	0,1
19			3	4	2,11
20		15	3	4	0,1,11
21			3	4	0,2,11
22		7	7	8	1,2,4
23		6	9	10	1,2,4,11
24	22	5	9	10	0,1,2,4,11
25			9	10	0,1,2,7,11
26	23	12	5	6	1
27		11	6	7	0,1
28	26	19	3	4	0,7,17
29		18	3	4	0,7,13,17
30		17	4	5	2,8,17
31			4	5	4,14,17
32		16	5	6	2,4,13,17
33			5	6	4,7,13,17

Table 3.3: Ternary Cyclic Codes of $n \leq 26$ (*Continued*)

NO.	n	k	$d_R(C)$	$d_S(C)$	G
34	26	16	5	6	1,8,13,17
35			4	6	4,8,13,17
36			4	6	7,8,13,17
37			5	6	1,13,14,17
38			5	6	2,13,14,17
39			5	6	4,13,14,17
40			5	6	7,13,14,17
41		15	5	6	0,7,8,13,17
42			5	6	0,4,8,13,17
43			5	6	1,2,8,17
44			5	6	4,7,14,17
45			5	6	1,7,8,17
46		14	4	6	1,7,14,17
47			5	7	1,8,14,17
48			5	6	2,8,14,17
49			5	6	4,8,14,17
50		13	5	7	1,8,13,14,17
51			5	6	1,7,8,13,17
52			5	6	4,7,8,13,17
53			5	8	1,2,13,14,17
54			5	6	1,4,13,14,17
55			5	6	2,4,13,14,17
56			7	8	4,5,13,14,17
57			5	6	1,7,13,14,17
58			6	7	4,7,13,14,17
59			5	6	4,8,13,14,17
60		12	7	8	0,4,7,13,14,17
61			5	6	0,4,7,8,13,17
62			7	8	0,1,8,13,14,17
63		11	6	7	0,1,8,13,14,17
64			5	7	1,4,8,14,17
65			5	6	2,4,8,14,17
66			7	8	2,7,8,14,17

Chapter 4

The Feng-Rao Designed minimum Distance and the Feng-Rao Decoding

4.1 Introduction

In this chapter we will briefly review Miura's definition [8] of the Feng-Rao designed minimum distance d_{FR} of linear codes and related facts including the Feng-Rao decoding of linear codes, since d_{FR} is closely related to the Feng-Rao decoding.

The Feng-Rao decoding [3, 8] of an (n, k) linear code C over F_q is as follows.

(1) We consider an ordered basis

$$B_n = \{\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_n\} \tag{4.1}$$

of F_q^n and define an (n, k) linear code C over F_q as

$$C = \mathrm{Span}\{B\}^{\perp}, \tag{4.2}$$

where

$$B = \{\boldsymbol{b}_{u_1}, \boldsymbol{b}_{u_2}, \cdots, \boldsymbol{b}_{u_{n-k}}\} \subset B_n, \tag{4.3}$$

and $\mathrm{Span}\{B\}$ means a subspace of F_q^n spanned by B.

(2) Let $\boldsymbol{c} = (c_1, c_2, \cdots, c_n) \in F_q^n$ be a codeword of C. If $\boldsymbol{y} = (y_1, y_2, \cdots, y_n) \in F_q^n$ is received when the code word $\boldsymbol{c}$ was sent, we define as $y = c + e$. Then the Feng-Rao decoding will find the error vector $\boldsymbol{e} = (e_1, e_2, \cdots, e_n) \in F_q^n$ correctly as

$$\boldsymbol{e} = \boldsymbol{y} - \boldsymbol{c} \tag{4.4}$$

when

$$\mathrm{wt}(\boldsymbol{e}) \leq \lfloor (d_{FR} - 1)/2 \rfloor. \tag{4.5}$$

(3) The determination of $\boldsymbol{e}$ will be done as follows. Define an $n \times n$ syndrome matrix $S(\boldsymbol{e})$ over F_q as

$$S(\boldsymbol{e}) = H(B_n)\mathrm{diag}(\boldsymbol{e})^t H(B_n), \tag{4.6}$$

where

$$H(B_n) = \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \vdots \\ \boldsymbol{b}_n \end{bmatrix}, \tag{4.7}$$

$$\mathrm{diag}(\boldsymbol{e}) = \begin{bmatrix} e_1 & & & \\ & e_2 & & 0 \\ & & \ddots & \\ 0 & & & e_n \end{bmatrix}, \tag{4.8}$$

and ${}^tH(B_n)$ is the transposed matrix of $H(B_n)$. If we can compute $S(\boldsymbol{e})$ from the received word $\boldsymbol{y}$, then we can determine the error vector $\boldsymbol{e}$ by the inverse matrixes of $H(B_n)$ and ${}^tH(B_n)$ as

$$\mathrm{diag}(\boldsymbol{e}) = H(B_n)^{-1} S(\boldsymbol{e})^t H(B_n)^{-1} \tag{4.9}$$

since $H(B_n)$ is nonsingular.

The main effort of the Feng-Rao decoding is the computation of $S(\boldsymbol{e})$ from $\boldsymbol{y}$. This effort can be done as follows.

The (i, j) element of $S(\boldsymbol{e})$ is given by $\langle \boldsymbol{e}, \boldsymbol{b}_i \boldsymbol{b}_j \rangle$. For two vectors $\boldsymbol{x} = (x_1, x_2, \cdots, x_n) \in F_q^n$ and $\boldsymbol{y} = (y_1, y_2, \cdots, y_n) \in F_q^n$ we define the inner product

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = x_1 y_1 + x_2 y_2 + \cdots + x_n y_n \in F_q^n$$

and vector product

$$\boldsymbol{xy} = (x_1 y_1, x_2 y_2, \cdots, x_n y_n) \in F_q^n,$$

respectively. By expanding $\boldsymbol{b}_i\boldsymbol{b}_j$ with respect to the ordered basis $B_n = \{\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_n\}$ the element $\langle \boldsymbol{e}, \boldsymbol{b}_i\boldsymbol{b}_j \rangle$ can be computed from

$$\langle \boldsymbol{e}, \boldsymbol{b}_1 \rangle, \langle \boldsymbol{e}, \boldsymbol{b}_2 \rangle, \cdots, \langle \boldsymbol{e}, \boldsymbol{b}_n \rangle. \tag{4.10}$$

Among the n values of (4.10), the $n-k$ values

$$\langle \boldsymbol{e}, \boldsymbol{b}_{u_i} \rangle \quad (i = 1, 2, \cdots, n-k) \tag{4.11}$$

can be easily computed by

$$\langle \boldsymbol{e}, \boldsymbol{b}_{u_i} \rangle = \langle \boldsymbol{y}, \boldsymbol{b}_{u_i} \rangle,$$

since $\langle \boldsymbol{c}, \boldsymbol{b}_{u_i} \rangle = 0$ from the definition of C, i.e., (4.2) and (4.3). The remaining k values

$$\langle \boldsymbol{e}, \boldsymbol{b}_j \rangle \quad j \notin \{u_1, u_2, \cdots, u_{n-k}\}$$

will be determined by applying the Extended Fundamental Iterative Algorithm (EFIA) to the $n \times n$ matrix $S(\boldsymbol{y})$ using the majority voting principle together [3, 4].

Therefore we will briefly review the EFIA in Section 4.2, the definition and useful properties of d_{FR} in Section 4.3, and the Feng-Rao decoding in Section 4.4.

4.2 The Fundamental Iterative Algorithm and Its Extension

In this section we explain the fundamental iterative algorithm and its extension as reference for the Feng-Rao decoding.

Let

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & \cdots & \vdots \\ a_{M1} & a_{M2} & \cdots & a_{MN} \end{bmatrix}.$$

be an $M \times N$ matrix over a field F. We assume $rank(A) < N$. Then the columns of A are linearly dependent. Let

$$C(\boldsymbol{x}) = c_0 + c_1 x + \cdots + c_l x^l, \quad c_0 = 1$$

$$a^{(i)}(\boldsymbol{x}) = a_{i,0} + a_{i,1} x + \cdots + a_{i,N} x^N, \quad a_{i,0} = 1, \quad i = 1, 2, \cdots, M.$$

Let $[C(\boldsymbol{x})a^{(i)}(\boldsymbol{x})]_n$, $\quad l+1 \leq n \leq N$, be the coefficient of x^n in $C(\boldsymbol{x})a^{(i)}(\boldsymbol{x})$. $\deg(C(\boldsymbol{x})) \leq l$ such that $[C(\boldsymbol{x})a^{(i)}(\boldsymbol{x})]_{l+1} = 0$ for $i = 1, 2, \cdots, M$

For each column j, we define $C^{(i-1,j)}(\boldsymbol{x}) = \sum_{k=0}^{j-1} c_k^{(i-1,j)} x^k$, where $1 \leq i \leq M, \quad c_0^{(i-1,j)} = 1$, to be the polynomial with the property that

$$\begin{aligned}[C^{(i-1,j)}(x)a^{(h)}(x)]_j &= a_{h,j} + c_1^{(i-1,j)} a_{h,j-1} + \cdots + c_{j-1}^{(i-1,j)} a_{h,1} \\ &= 0 \qquad h \leq i-1.\end{aligned}$$

$C^{(0,j)}(\boldsymbol{x})$ is then referred as the initial polynomial for column j. We have $C^{(0,1)}(\boldsymbol{x}) = 1$ for the first column. We refer to

$$d_{i,j} = [C^{(i-1,j)}(x)a^{(i)}(x)]_j = a_{i,j} + c_1^{(i-1,j)} a_{i,j-1} + \cdots + c_{j-1}^{(i-1,j)} a_{i,1}$$

as the *discrepancy* at the row i and column j. We define the final polynomial at column j to be $C^{(j)}(\boldsymbol{x}) = C^{(r-1,j)}(\boldsymbol{x})$ and refer to this nonzero $d_{r,j}$ as the final discrepancy in column j. We will also refer to a final discrepancy as a *primary discrepancy* and mark its existence with an "$\times$".

(1) Fundamental Iterative Algorithm (FIA)

Step 1 : Empty Table D and C, $1 \to s, 1 \to r, 1 \to C^{(0,s)}(\boldsymbol{x})$.

Stap 2 : Compute $d_{r,s} = [C^{(r-1,s)}(\boldsymbol{x})a^{(r)}(\boldsymbol{x})]_s$.

Stap 3 : If $d_{r,s} = 0$, then

(a) if $r = M$, then $s-1 \to l, \quad C^{(r-1),s}(\boldsymbol{x}) \to C(\boldsymbol{x})$, stop;

(b) otherwise $C^{(r-1,s)}(\boldsymbol{x}) \to C^{(r,s)}(\boldsymbol{x}), \quad r+1 \to r$, and return to Step 2.

Stap 4 : If $d_{r,s} \neq 0$, then

(a) if there exists a $d_{r,u} \in D$ for some $1 \leq u < s$, then

$$C^{(r-1,s)}(\boldsymbol{x}) - \frac{d_{r,s}}{d_{r,u}} C^{(u)}(\boldsymbol{x}) x^{s-u} \to C^{(r-1,s)}(\boldsymbol{x})$$

and return to Step 3(a);

(b) otherwise, $d_{r,s}$ is stored in D,

$$C^{(r-1,s)}(\boldsymbol{x}) \to C^{(s)}(\boldsymbol{x}) \to C^{(0,s+1)}(\boldsymbol{x})$$

and $C^{(s)}(\boldsymbol{x})$ is stored in C, then $s+1 \to s, 1 \to r$, and return to Step 2.

The final s and $C^{(r-1,s)}(\boldsymbol{x})$ give the solutions to l and $C(\boldsymbol{x})$. It is seen that whenever $d_{i,j} = 0$, the top i components of column j is a linear combination of the top i components of its preceding $j-1$ columns.

Example 4.2.1 *Let the binary matrix*

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{16} & \cdots \\ a_{21} & a_{22} & a_{23} & \cdots & a_{26} & \cdots \\ \vdots & \vdots & \vdots & & \vdots & \\ a_{61} & a_{62} & a_{63} & \cdots & a_{66} & \cdots \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 1 & \cdots \\ 1 & 0 & 0 & 0 & 0 & 1 & \cdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 1 & 0 & 1 & 0 & 0 & \cdots \\ 0 & 0 & 0 & 1 & 1 & 1 & \cdots \\ 0 & 0 & 1 & 0 & 1 & 1 & \cdots \end{bmatrix}.$$

If we use FIA, we have the following matrix D

$$D = \begin{bmatrix} 0 & 0 & 1 & (1) & 0 & \cdots \\ 1 & 0 & & 0 & 0 & \cdots \\ & 0 & & 0 & 0 & \cdots \\ & 1 & & (1) & (1) & \cdots \\ & & & 1 & 0 & \cdots \\ & & & & 0 & \cdots \end{bmatrix},$$

and $l = s - 1 = 4$, $C(\boldsymbol{x}) = C^{(5,5)}(\boldsymbol{x}) = 1 + x + x^2 + x^3$.

Now we consider the application of the FIA to the matrix A. For any column j, assume we have obtained $C^{(r-1,j)}(\boldsymbol{x})$ such that $[C^{(r-1,j)}(\boldsymbol{x})a^{(i)}(\boldsymbol{x})]_j = 0$ for $i = 1, 2, \cdots, r-1$ and $a_{r,j}$ is unknown. Then there are two cases to be considered, if there is no primary discrepancy at row r to the left of $a_{r,j}$.

(1) In case of $d_{r,j} = 0$:

$$d_{r,j} = [C^{(r-1,j)}(x)a^{(r)}(x)]_j = a_{r,j} + c^{(r-1,j)}a_{r,j-1} + \cdots + c^{(r-1,j)}a_{r,1} = 0,$$

then

$$a_{r,j} = -\sum_{k=1}^{j-1} c^{(r-1,j)}a_{r,j-k},$$

the value computed for $a_{r,j}$ is true, if $a_{r,j}$ is unknown syndrome.

(2) In case of $d_{r,j} \neq 0$: $d_{r,j}$ is a primary discrepancy, the value of $a_{r,j}$ can not be decided.

(2) Extended Fundamental Iterative Algorithm (EFIA)

Step 1 : Empty Table D, C, E and F, $1 \to s, 1 \to r, 1 \to C^{(0,s)}(\boldsymbol{x})$.

Stap 2 : Compute $d_{r,s} = [C^{(r-1,s)}(\boldsymbol{x})a^{(r)}(\boldsymbol{x})]_s$.

Stap 3 : If $d_{r,s} = 0$, then

(a) if $r = 2t + 1 - s$, then if there is no $d_{r+1,u} \in D$, then go to (a1), otherwise go to (a2);

(a1) calculate

$$a_{r+1,s} = \sum_{k=1}^{s-1} c_k^{(r,s)} a_{r+1,s-k},$$

$a_{r+1,s}$ and $C^{(r-1,s)}(\boldsymbol{x})$ are stored in E and F, respectively, then go to (a2);

(a2) if $s = 2t$, then stop, otherwise

$$C^{(r-1),s}(\boldsymbol{x}) \to C^{(s)}(\boldsymbol{x}) \to C^{(0,s+1)}(\boldsymbol{x}),$$

$s + 1 \to s, \quad 1 \to r$, and return to Step 2;

(b) otherwise $C^{(r-1,s)}(\boldsymbol{x}) \to C^{(r,s)}(\boldsymbol{x}), r + 1 \to r$, and return to Step 2.

Stap 4 : If $d_{r,s} \neq 0$, then

(a) if there exists a $d_{r,u} \in D$ for some $1 \leq u < s$, then

$$C^{(r-1,s)}(\boldsymbol{x}) - \frac{d_{r,s}}{d_{r,u}} C^{(u)}(\boldsymbol{x}) x^{s-u} \to C^{(r-1,s)}(\boldsymbol{x})$$

and return to Step 3(a);

(b) otherwise, $d_{r,s}$ is stored in D, and $C^{(s)}(\boldsymbol{x})$ is stored in C, then go to Step 3(a2).

When the algorithm stops, Table E contains all the values computed for unknown syndromes. Obviously, the complexity of this algorithm is $O(n^3)$.

4.3 The Definition of the Feng-Rao Designed Minimum Distance by Miura

We will call $B_n = \{\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_n\} \subseteq F_q^n$ an ordered basis of F_q^n if B_n is a basis of F_q^n and the ordering of n vectors $\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_n$ has meaning. The subset B_i of B_n is defined by $B_i = \{\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_i\}$ for $1 \leq i \leq n$.

Definition 4.3.1 *For a vector* $\boldsymbol{b} \in F_q^n$, *the map* σ: $F_q^n \to \{0, 1, 2, \cdots, n\}$ *is defined as*

$$\sigma(\boldsymbol{b}) = \min\{i | \boldsymbol{b} \in \mathrm{Span}\{B_i\},\ 0 \le i \le n\},$$

where $\mathrm{Span}\{B_i\}$ *is a subspace of* F_q^n *spanned by* B_i *and* $\mathrm{Span}\{B_0\} = \{\mathbf{0}\}$.

For two vectors $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$, $\boldsymbol{y} = (y_1, y_2, \cdots, y_n) \in F_q^n$, their product $\boldsymbol{xy}$ is defined as $\boldsymbol{xy} = (x_1y_1, x_2y_2, \cdots, x_ny_n) \in F_q^n$.

Definition 4.3.2 *The product* $\boldsymbol{b}_i\boldsymbol{b}_j$ *is said to be well-behaved if* $\sigma(\boldsymbol{b}_u\boldsymbol{b}_v) < \sigma(\boldsymbol{b}_i\boldsymbol{b}_j)$ *for any* u, v *satisfying* $1 \le u \le i, \quad 1 \le v \le j, \quad (u, v) \ne (i, j)$.

Definition 4.3.3 *The product* $\boldsymbol{b}_i\boldsymbol{b}_j$ *is said to be weakly well-behaved if* $\sigma(\boldsymbol{b}_u\boldsymbol{b}_v) < \sigma(\boldsymbol{b}_i\boldsymbol{b}_j)$ *for any* u *satisfying* $1 \le u < i, v = j$, *and any* v *satisfying* $u = i, 1 \le v < j$.

Definition 4.3.4 *For* $1 \le s \le n$, *we define* $N(s), N^*(s)$ *as*

$$N(s) = \sharp\{(i, j) | \sigma(\boldsymbol{b}_i\boldsymbol{b}_j) = s,\ 1 \le i, j \le n, \boldsymbol{b}_i\boldsymbol{b}_j \textit{is well-behaved}\},$$

$$N^*(s) = \sharp\{(i, j) | \sigma(\boldsymbol{b}_i\boldsymbol{b}_j) = s,\ 1 \le i, j \le n, \boldsymbol{b}_i\boldsymbol{b}_j \textit{is weakly well-behaved}\},$$

where $\sharp A$ *means the cardinality of set* A. *For an ordered basis* B_n *of* F_q^n *we define* $N(B_n), N^*(B_n)$ *as*

$$N(B_n) = (N(1), N(2), \cdots, N(n)),$$

$$N^*(B_n) = (N^*(1), N^*(2), \cdots, N^*(n)).$$

Lemma 4.3.1 *For a given ordered basis* $B_n = \{\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_n\}$, *we can determine* $N(B_n)$ *in computational complexity* $O(n^4)$.

Proof We can calculate $N(B_n)$ from the $n \times n$ matrix

$$[\sigma(\boldsymbol{b}_i\boldsymbol{b}_j)] \quad (1 \le i, j \le n),$$

where $\sigma(\boldsymbol{b}_i\boldsymbol{b}_j)$ can be determined as follows. Let

$$\begin{aligned} \boldsymbol{b}_i\boldsymbol{b}_j &= \sum_{k=1}^{n} \alpha_k \boldsymbol{b}_k \\ &= \alpha_1\boldsymbol{b}_1 + \alpha_2\boldsymbol{b}_2 + \cdots + \alpha_n\boldsymbol{b}_n \\ &= \alpha H(B_n), \end{aligned}$$

where

$$\alpha := (\alpha_1, \alpha_2, \cdots, \alpha_n)$$

and

$$H(B_n) = \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \vdots \\ \boldsymbol{b}_n \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{nn} \end{bmatrix}.$$

We have $\alpha = \boldsymbol{b}_i \boldsymbol{b}_j H(B_n)^{-1}$ and

$$\sigma(\boldsymbol{b}_i \boldsymbol{b}_j) = \max\{k | \alpha_k \neq 0, 1 \leq k \leq n\},$$

where $H(B_n)^{-1}$ is the inverse matrix of $H(B_n)$. The computational complexity of $H(B_n)^{-1}$ and α are $O(n^3)$ and $O(n^2)$, respectively. Therefore we can compute the $n \times n$ matrix $[\sigma(\boldsymbol{b}_i \boldsymbol{b}_j)]$ with the computational complexity of $O(n^4)$.

□

Lemma 4.3.2 *Let* $\boldsymbol{a}$, $\boldsymbol{b} \in F_q^n, \alpha, \beta \in F_q$, *we have* $\sigma(\alpha \boldsymbol{a} + \beta \boldsymbol{b}) \leq \max\{\sigma(\boldsymbol{a}), \sigma(\boldsymbol{b})\}$. *The equality holds if* $\alpha \neq 0, \beta \neq 0, \sigma(\boldsymbol{a}) \neq \sigma(\boldsymbol{b})$.

Proof Let $\sigma(\boldsymbol{a}) = u, \sigma(\boldsymbol{b}) = v$. We have

$$\boldsymbol{a} = c_1 \boldsymbol{b}_1 + c_2 \boldsymbol{b}_2 + \cdots + c_u \boldsymbol{b}_u,$$

$$\boldsymbol{b} = c_1' \boldsymbol{b}_1 + c_2' \boldsymbol{b}_2 + \cdots + c_v' \boldsymbol{b}_v,$$

where $c_i, c_j' \in F_q \quad (1 \leq i \leq u, 1 \leq j \leq v)$.

If $u > v$ and $\alpha \neq 0$, then we have

$$\begin{aligned} & \alpha \boldsymbol{a} + \beta \boldsymbol{b} \\ = \; & (\alpha c_1 + \beta c_1') \boldsymbol{b}_1 + (\alpha c_2 + \beta c_2') \boldsymbol{b}_2 + \cdots + (\alpha c_v + \beta c_v') \boldsymbol{b}_v + \alpha c_{v+1} \boldsymbol{b}_{v+1} \\ & + \cdots + \alpha c_u \boldsymbol{b}_u, \end{aligned}$$

and $\sigma(\alpha \boldsymbol{a} + \beta \boldsymbol{b}) = u$.

If $u < v$ and $\beta \neq 0$, then we have $\sigma(\alpha \boldsymbol{a} + \beta \boldsymbol{b}) = v$.

If $u = v$, then we have $\sigma(\alpha \boldsymbol{a} + \beta \boldsymbol{b}) \leq u = v$.

□

Lemma 4.3.3 *For* $1 \leq t \leq n, \boldsymbol{b} \in F_q^n$, *if* $\sigma(\boldsymbol{b}_t\boldsymbol{b}) < t$, *then there exists at least one* i *with* $1 \leq i < t$ *such that* $\sigma(\boldsymbol{b}_t\boldsymbol{b}) \leq \sigma(\boldsymbol{b}_i\boldsymbol{b})$.

Proof See the proof of Lemma 5.2.1 in the next chapter.

□

This Lemma tells us that $\boldsymbol{b}_i\boldsymbol{b}_j$ is not weakly well-behaved, therefore $\boldsymbol{b}_i\boldsymbol{b}_j$ is not also well-behaved, if $\sigma(\boldsymbol{b}_i\boldsymbol{b}_j) < i$ or $\sigma(\boldsymbol{b}_i\boldsymbol{b}_j) < j$.

Corollary 4.3.1 *If* $\sigma(\boldsymbol{b}_i\boldsymbol{b}_j) = s$ *and* $\boldsymbol{b}_i\boldsymbol{b}_j$ *is weakly well-behaved, then we have* $1 \leq i \leq s,\ 1 \leq j \leq s$.

Proof If we assume that $i \geq s$, then we have $\sigma(\boldsymbol{b}_i\boldsymbol{b}_j) = s < i$. From Lemma 4.3.3 there exists at least one u with $1 \leq u < i$ such that

$$\sigma(\boldsymbol{b}_u\boldsymbol{b}_j) \geq \sigma(\boldsymbol{b}_i\boldsymbol{b}_j),$$

which is a contradiction to the assumption that $\boldsymbol{b}_i\boldsymbol{b}_j$ is weakly well-behaved.

The proof of $j \leq s$ is the similar.

□

Lemma 4.3.4 *For* $1 \leq s \leq n$, *we have* $0 \leq N(s) \leq N^*(s) \leq s$.

Proof $0 \leq N(s) \leq N^*(s)$ is obvious. $N^*(s) \leq s$ is shown as follows.

(1) When u is fixed, $N^*(s) = \sharp\{(i,j) | \sigma(\boldsymbol{b}_i\boldsymbol{b}_j) = s,\ 1 \leq i,j \leq n, \boldsymbol{b}_i\boldsymbol{b}_j$ is well-behaved$\}$, there exists at most one v for $1 \leq v \leq n$. From Corollary 5.2.1, we have $1 \leq v \leq s, N^*(s) \leq s$.

(2) When v is fixed, the proof is as same as (1).

□

Let B be a subset of an ordered basis B_n of F_q^n. The linear code $C(B_n, B)$ over F_q is defined as

$$C(B_n, B) = \mathrm{Span}\{B\}^{\perp},$$

where $\mathrm{Span}\{B\}^{\perp}$ means the set of all vectors in F_q^n orthogonal to $\mathrm{Span}\{B\}$.

Definition 4.3.5 *The Feng-Rao designed minimum distance of the linear code C is denoted as $d_{FR}(C, B_n)$ and defined as*

$$d_{FR}(C, B_n) = \min\{N(s)|\boldsymbol{b}_s \in B_n\backslash B,\ 1 \leq s \leq n\},$$

$$d^*_{FR}(C, B_n) = \min\{N^*(s)|\boldsymbol{b}_s \in B_n\backslash B,\ 1 \leq s \leq n\},$$

where $B_n\backslash B$ is the subset of B_n without the elements of B.

4.4 The Feng-Rao Decoding

Definition 4.4.1 *For $\boldsymbol{y} = (y_1, y_2, \cdots, y_n) \in F_q^n$, the $n \times n$ diagonal matrix $\mathrm{diag}(\boldsymbol{y})$ over F_q, is defined as*

$$\mathrm{diag}(\boldsymbol{y}) = \begin{bmatrix} y_1 & & & \\ & y_2 & & 0 \\ & & \ddots & \\ 0 & & & y_n \end{bmatrix}.$$

Let $B_n = \{\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_n\}$ be an ordered basis of F_q^n. Let $H(B_n)$ be an $n \times n$ nonsingular matrix defined as

$$H(B_n) = \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \vdots \\ \boldsymbol{b}_n \end{bmatrix}.$$

Definition 4.4.2 *For $\boldsymbol{y} \in F_q^n$, the $n \times n$ matrix $S(\boldsymbol{y})$ over F_q, is defined as*

$$S(\boldsymbol{y}) := H(B_n)\mathrm{diag}(\boldsymbol{y})^t H(B_n),$$

where ${}^tH(B_n)$ is the transposition of $H(B_n)$. $S(\boldsymbol{y})$ is called the syndrome matrix of $\boldsymbol{y} \in F_q^n$.

Lemma 4.4.1 *For $1 \leq i, j \leq n$, we have $S(\boldsymbol{y}) = [\langle \boldsymbol{y}, \boldsymbol{b}_i\boldsymbol{b}_j \rangle]$.*

Proof From above definition, we have

$$\begin{aligned} & S(\boldsymbol{y}) \\ &= H(B_n)\mathrm{diag}(\boldsymbol{y})^t H(B_n) \end{aligned}$$

$$
\begin{aligned}
&= \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{nn} \end{bmatrix} \begin{bmatrix} y_1 & & & \\ & y_2 & & 0 \\ & & \ddots & \\ & 0 & & y_n \end{bmatrix} \begin{bmatrix} b_{11} & b_{21} & \cdots & b_{n1} \\ b_{12} & b_{22} & \cdots & b_{n2} \\ \vdots & \vdots & & \vdots \\ b_{1n} & b_{2n} & \cdots & b_{nn} \end{bmatrix} \\
&= \begin{bmatrix} \sum_{j=1}^n b_{1j}b_{1j}y_j & \sum_{j=1}^n b_{1j}b_{2j}y_j & \cdots & \sum_{j=1}^n b_{1j}b_{nj}y_j \\ \sum_{j=1}^n b_{2j}b_{1j}y_j & \sum_{j=1}^n b_{2j}b_{2j}y_j & \cdots & \sum_{j=1}^n b_{2j}b_{nj}y_j \\ & \vdots & & \vdots \\ \sum_{j=1}^n b_{nj}b_{1j}y_j & \sum_{j=1}^n b_{nj}b_{2j}y_j & \cdots & \sum_{j=1}^n b_{nj}b_{nj}y_j \end{bmatrix} \\
&= \begin{bmatrix} \sum_{j=1}^n b_{1j}b_{1j}y_j & \cdots & \sum_{j=1}^n b_{1j}b_{nj}y_j \\ \vdots & \sum_{j=1}^n b_{ij}b_{kj}y_j & \vdots \\ \sum_{j=1}^n b_{nj}b_{1j}y_j & \cdots & \sum_{j=1}^n b_{nj}b_{nj}y_j \end{bmatrix}.
\end{aligned}
$$

On the other hand,

$$
\begin{aligned}
&[\langle \boldsymbol{y}, \boldsymbol{b}_i\boldsymbol{b}_j\rangle] \\
&= \begin{bmatrix} \langle \boldsymbol{y}, \boldsymbol{b}_1\boldsymbol{b}_1\rangle & \langle \boldsymbol{y}, \boldsymbol{b}_1\boldsymbol{b}_2\rangle & \cdots & \langle \boldsymbol{y}, \boldsymbol{b}_1\boldsymbol{b}_n\rangle \\ \langle \boldsymbol{y}, \boldsymbol{b}_2\boldsymbol{b}_1\rangle & \langle \boldsymbol{y}, \boldsymbol{b}_2\boldsymbol{b}_2\rangle & \cdots & \langle \boldsymbol{y}, \boldsymbol{b}_2\boldsymbol{b}_n\rangle \\ \vdots & \vdots & & \vdots \\ \langle \boldsymbol{y}, \boldsymbol{b}_n\boldsymbol{b}_1\rangle & \langle \boldsymbol{y}, \boldsymbol{b}_n\boldsymbol{b}_2\rangle & \cdots & \langle \boldsymbol{y}, \boldsymbol{b}_n\boldsymbol{b}_n\rangle \end{bmatrix} \\
&= \begin{bmatrix} \sum_{j=1}^n b_{1j}b_{1j}y_j & \sum_{j=1}^n b_{1j}b_{2j}y_j & \cdots & \sum_{j=1}^n b_{1j}b_{nj}y_j \\ \sum_{j=1}^n b_{2j}b_{1j}y_j & \sum_{j=1}^n b_{2j}b_{2j}y_j & \cdots & \sum_{j=1}^n b_{2j}b_{nj}y_j \\ \vdots & \vdots & & \vdots \\ \sum_{j=1}^n b_{nj}b_{1j}y_j & \sum_{j=1}^n b_{nj}b_{2j}y_j & \cdots & \sum_{j=1}^n b_{nj}b_{nj}y_j \end{bmatrix} \\
&= \begin{bmatrix} \sum_{j=1}^n b_{1j}b_{1j}y_j & \cdots & \sum_{j=1}^n b_{1j}b_{nj}y_j \\ \vdots & \sum_{j=1}^n b_{ij}b_{kj}y_j & \vdots \\ \sum_{j=1}^n b_{nj}b_{1j}y_j & \cdots & \sum_{j=1}^n b_{nj}b_{nj}y_j \end{bmatrix}.
\end{aligned}
$$

Consequently, we have $S(\boldsymbol{y}) = [\langle \boldsymbol{y}, \boldsymbol{b}_i\boldsymbol{b}_j\rangle]$.

□

Lemma 4.4.2 *For $\boldsymbol{y} \in F_q^n$, we have* $\mathrm{wt}(\boldsymbol{y}) = \mathrm{rank}(\mathrm{diag}(\boldsymbol{y})) = \mathrm{rank}(S(\boldsymbol{y}))$.

Proof $\mathrm{wt}(\boldsymbol{y}) = \mathrm{rank}(\mathrm{diag}(\boldsymbol{y}))$ is obvious. Since $H(B_n)$ is nonsingular matrix, $\mathrm{rank}(H(B_n)) = \mathrm{rank}({}^tH(B_n)) = n$. From

$$S(\boldsymbol{y}) = H(B_n)\mathrm{diag}(\boldsymbol{y})^tH(B_n),$$

we have

$$\mathrm{rank}(H(B_n)\mathrm{diag}(\boldsymbol{y})) = \mathrm{rank}(\mathrm{diag}(\boldsymbol{y})).$$

In the same way, from

$$\text{rank}(\text{diag}(\boldsymbol{y})^t H(B_n)) = \text{rank}(\text{diag}(\boldsymbol{y})),$$

we have

$$\text{rank}(S(\boldsymbol{y})) = \text{rank}(\text{diag}(\boldsymbol{y})).$$

Consequently, we can prove

$$\text{wt}(\boldsymbol{y}) = \text{rank}(\text{diag}(\boldsymbol{y})) = \text{rank}(S(\boldsymbol{y})).$$

□

Lemma 4.4.3 *Let d be the true minimum distance of $C(B_n, B)$. Then we have $d \geq d^*_{FR}(C, B_n) \geq d_{FR}(C, B_n)$.*

Proof Let $\boldsymbol{c} \in C(B_n, B) = \text{Span}\{\text{B}\}^{\perp}$, $\boldsymbol{c} \neq 0$. For $1 \leq s \leq n$, let

$$\langle \boldsymbol{c}, \boldsymbol{b}_1 \rangle = \langle \boldsymbol{c}, \boldsymbol{b}_2 \rangle = \cdots = \langle \boldsymbol{c}, \boldsymbol{b}_{s-1} \rangle = 0,$$

and $\langle \boldsymbol{c}, \boldsymbol{b}_s \rangle \neq 0$, then $\boldsymbol{b}_s \in B_n \backslash B$. For

$$^{\forall}(k,l) \in \{(i,j) | \sigma(\boldsymbol{b}_i \boldsymbol{b}_j) = s,\ 1 \leq i, j \leq n, \boldsymbol{b}_i \boldsymbol{b}_j \text{is weakly well-behaved}\}$$

the element $\langle \boldsymbol{c}, \boldsymbol{b}_k \boldsymbol{b}_l \rangle$ of $S(\boldsymbol{c})$ is nonzero, because

$$\begin{aligned}
\langle \boldsymbol{c}, \boldsymbol{b}_k \boldsymbol{b}_l \rangle &= \langle \boldsymbol{c}, \sum_{t=1}^{s} \alpha_t \boldsymbol{b}_t \rangle \\
&= \sum_{t=1}^{s} \alpha_t \langle \boldsymbol{c}, \boldsymbol{b}_t \rangle \\
&= \alpha_s \langle \boldsymbol{c}, \boldsymbol{b}_s \rangle + \sum_{t=1}^{s-1} \alpha_t \langle \boldsymbol{c}, \boldsymbol{b}_t \rangle \\
&= \alpha_s \langle \boldsymbol{c}, \boldsymbol{b}_s \rangle \\
&\neq 0.
\end{aligned}$$

On the other hand, for $u = i, 1 \leq v < j$ or $1 \leq u < i, v = j$, $\langle \boldsymbol{c}, \boldsymbol{b}_u \boldsymbol{b}_v \rangle = 0$. The ith column and jth row of $[\langle \boldsymbol{c}, \boldsymbol{b}_i \boldsymbol{b}_j \rangle]$ are linearly independent. Therefor at least the matrix $[\langle \boldsymbol{c}, \boldsymbol{b}_i \boldsymbol{b}_j \rangle]$ has $N^*(s)$ independent row or column. From Lemma 4.4.1 and Lemma 4.4.2, we have

$$\text{wt}(\boldsymbol{c}) = \text{rank}(S(\boldsymbol{c})) = \text{rank}([\langle \boldsymbol{c}, \boldsymbol{b}_i \boldsymbol{b}_j \rangle]) \geq N^*(s) \geq N(s).$$

Moreover, from the Definition of $d^*_{FR}(C, B_n)$ and $d_{FR}(C, B_n)$, we have

$$d \geq d^*_{FR}(C, B_n) \geq d_{FR}(C, B_n).$$

□

For linear codes $C(B_n, B)$, let $\boldsymbol{y} \in F_q^n$ be received vector, and $\boldsymbol{e} \in F_q^n$ be error vector. Then $\boldsymbol{c} = \boldsymbol{y} - \boldsymbol{e} \in C(B_n, B)$. We assume $\mathrm{wt}(\boldsymbol{e}) \leq \lfloor (d_{FR}(C, B_n) - 1)/2 \rfloor$, for $\boldsymbol{b}_s \in B_n \backslash B$ such that $s, 1 \leq s \leq n$, there is $\mathrm{rank}(S(\boldsymbol{e})) = \mathrm{wt}(\boldsymbol{e}) \leq \lfloor (N(s) - 1)/2 \rfloor$. From Lemma 4.3.4, we have $\mathrm{wt}(\boldsymbol{e}) \leq \lfloor (s-1)/2 \rfloor$. If $s \leq 2$, there is no error because $\mathrm{wt}(\boldsymbol{e}) = 0$. Therefore we can assume $\boldsymbol{b}_1, \boldsymbol{b}_2 \in B$. Let $\langle \boldsymbol{e}, \boldsymbol{b}_1 \rangle, \langle \boldsymbol{e}, \boldsymbol{b}_2 \rangle, \cdots, \langle \boldsymbol{e}, \boldsymbol{b}_{s-1} \rangle$ be known.

In case of $\boldsymbol{b}_s \in B$, we can determine $\langle \boldsymbol{e}, \boldsymbol{b}_s \rangle$, from $\boldsymbol{y} - \boldsymbol{e} \in C(B_n, B)$, we have $\langle \boldsymbol{y}, \boldsymbol{b}_s \rangle - \langle \boldsymbol{e}, \boldsymbol{b}_s \rangle = 0$ and $\langle \boldsymbol{e}, \boldsymbol{b}_s \rangle = \langle \boldsymbol{y}, \boldsymbol{b}_s \rangle$.

In case of $\boldsymbol{b}_s \notin B$, i.e., $\boldsymbol{b}_s \in B_n \backslash B$, let

$$(k, l) \in \{(i, j) | \sigma(\boldsymbol{b}_i \boldsymbol{b}_j) = s, \quad 1 \leq i, j \leq n. \boldsymbol{b}_i \boldsymbol{b}_j \text{is well-behaved}\}.$$

We have $\sigma(\boldsymbol{b}_u \boldsymbol{b}_v) < \sigma(\boldsymbol{b}_k \boldsymbol{b}_l) = s$ for $1 \leq u \leq k,\ 1 \leq v \leq l, (u, v) \neq (k, l)$. Let $\sigma(\boldsymbol{b}_u \boldsymbol{b}_v) \leq s - 1$. We can determine $\langle \boldsymbol{e}, \boldsymbol{b}_u \boldsymbol{b}_v \rangle$, since $\langle \boldsymbol{e}, \boldsymbol{b}_1 \rangle, \langle \boldsymbol{e}, \boldsymbol{b}_2 \rangle, \cdots, \langle \boldsymbol{e}, \boldsymbol{b}_{s-1} \rangle$ are known. The portion $\langle \boldsymbol{e}, \boldsymbol{b}_k \boldsymbol{b}_l \rangle$ of $S(\boldsymbol{e})$ with the location

$$(k, l) \in \{(i, j) | \sigma(\boldsymbol{b}_i \boldsymbol{b}_j) = s, 1 \leq i, j \leq n, \boldsymbol{b}_i \boldsymbol{b}_j \text{is well-behaved}\}$$

can be determined from the EFIA in Section 4.2.

Let $\mathrm{rank}(S(\boldsymbol{e})) = \lfloor (N(s) - 1)/2 \rfloor = \nu \leq t$, then there are at most ν primary discrepancies. For each

$$(k, l) \in \{(i, j) | \sigma(\boldsymbol{b}_i \boldsymbol{b}_j) = s, 1 \leq i, j \leq n,\ \boldsymbol{b}_i \boldsymbol{b}_j \text{is well-behaved}\},$$

the number of primary discrepancy at its internal is at most μ where $\mu \leq \nu$. The elements at lower or right of a primary discrepancy can not determine, then there are at most 2μ unknown positions. If $\mu = \nu$, the number of unknown as follows

$$\lfloor (N(s) - 1)/2 \rfloor \times 2 = \begin{cases} N(s) - 1 & \text{if } N(s) - 1 \text{ is even,} \\ N(s) - 2 & \text{if } N(s) - 1 \text{ is odd.} \end{cases}$$

Then the number of the true values is at least one, and the number of the true values is greater than the number of the false values.

Since $\sigma(\boldsymbol{b}_k\boldsymbol{b}_l) = s$, we can write

$$\boldsymbol{b}_k\boldsymbol{b}_l = \alpha_1\boldsymbol{b}_1 + \alpha_2\boldsymbol{b}_2 + \cdots + \alpha_s\boldsymbol{b}_s.$$

Then we can calculate by

$$\begin{aligned}\langle \boldsymbol{e}, \boldsymbol{b}_k\boldsymbol{b}_l\rangle &= \langle \boldsymbol{e}, \alpha_1\boldsymbol{b}_1 + \alpha_2\boldsymbol{b}_2 + \cdots + \alpha_s\boldsymbol{b}_s\rangle \\ &= \alpha_1\langle \boldsymbol{e}, \boldsymbol{b}_1\rangle + \alpha_2\langle \boldsymbol{e}, \boldsymbol{b}_2\rangle + \cdots + \alpha_{s-1}\langle \boldsymbol{e}, \boldsymbol{b}_{s-1}\rangle + \alpha_s\langle \boldsymbol{e}, \boldsymbol{b}_s\rangle \\ &= \sum_{k=1}^{s-1}\alpha_k\langle \boldsymbol{e}, \boldsymbol{b}_k\rangle + \alpha_s\langle \boldsymbol{e}, \boldsymbol{b}_s\rangle.\end{aligned}$$

Hence we have

$$\langle \boldsymbol{e}, \boldsymbol{b}_s\rangle = (\langle \boldsymbol{e}, \boldsymbol{b}_k\boldsymbol{b}_l\rangle - \sum_{k=1}^{s-1}\alpha_k\langle \boldsymbol{e}, \boldsymbol{b}_k\rangle)/\alpha_s,$$

therefore, $\langle \boldsymbol{e}, \boldsymbol{b}_s\rangle$ can be computed. Consequently, the element $\langle \boldsymbol{e}, \boldsymbol{b}_i\boldsymbol{b}_j\rangle$ of $S(\boldsymbol{e})$ such that $\sigma(\boldsymbol{b}_i\boldsymbol{b}_j) \leq s, 1 \leq i, j \leq n$ are a known.

From

$$S(\boldsymbol{e}) := H(B_n)\mathrm{diag}(\boldsymbol{e})^t H(B_n),$$

we have

$$\mathrm{diag}(\boldsymbol{e}) := H(B_n)^{-1} S(\boldsymbol{e})^t H(B_n)^{-1},$$

then the error vector $\boldsymbol{e}$ can be computed.

Chapter 5

The Feng-Rao Designed Minimum Distance of Binary Linear Codes

5.1 Introduction

The Feng-Rao designed minimum distance d_{FR} and the Feng-Rao decoding were originally introduced into algebraic geometry codes by G. L. Feng and T. R. N. Rao [3]. They have been extended to the case of general linear codes over a finite field by Miura [8].

Miura's definition of $d_{FR}(C, B_n)$ for an (n, k) linear code C over a finite field F_q with order q depends on the choice of an ordered basis $B_n = \{\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_n\}$ of the vector space F_q^n with dimension n over F_q, and the ordering of n vectors $\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_n$ has meaning. It is interesting to find such an optimum ordered basis B_n as d_{FR} is maximum, since the Feng-Rao decoding can correct up to $\lfloor (d_{FR}(C, B_n) - 1)/2 \rfloor$ errors.

This chapter shows that $d_{FR}(C, B_n)$ of binary linear codes can not take an odd value except one if we use Miura's definition of $d_{FR}(C, B_n)$.

Recently the definition of $d_{FR}(C, B_n)$ of linear codes has been slightly modified by Matsumoto [7] which uses three ordered basis of F_q^n instead of one in case of Miura's definition. This chapter does not discuss whether the freedom introduced in Matsumoto's definition will result in another conclusion or not, although it is a very interesting problem. We will discuss about this problem in Chapter 6.

5.2 The Feng-Rao Designed Minimum Distance of Binary Linear Codes

For a given (n, k) linear code C over F_q, the value of the Feng-Rao designed minimum distance $d_{FR}(C, B_n)$ depends on the value of the number of well-behaved, since the definition of $d_{FR}(C, B_n)$.

The matrix $[\sigma(\boldsymbol{b}_i\boldsymbol{b}_j)]$ is

$$\begin{bmatrix} \sigma(\boldsymbol{b}_1\boldsymbol{b}_1) & \sigma(\boldsymbol{b}_1\boldsymbol{b}_2) & \cdots & \sigma(\boldsymbol{b}_1\boldsymbol{b}_n) \\ \sigma(\boldsymbol{b}_2\boldsymbol{b}_1) & \sigma(\boldsymbol{b}_2\boldsymbol{b}_2) & \cdots & \sigma(\boldsymbol{b}_2\boldsymbol{b}_n) \\ \vdots & \vdots & \ddots & \vdots \\ \sigma(\boldsymbol{b}_n\boldsymbol{b}_1) & \sigma(\boldsymbol{b}_n\boldsymbol{b}_2) & \cdots & \sigma(\boldsymbol{b}_n\boldsymbol{b}_n) \end{bmatrix},$$

from the property of symmetry, we have $\sigma(\boldsymbol{b}_i\boldsymbol{b}_j) = \sigma(\boldsymbol{b}_j\boldsymbol{b}_i)$. Let

$$\boldsymbol{b}_i\boldsymbol{b}_j = \sum_{i=1}^{n} \alpha_i \boldsymbol{b}_i, \quad \alpha = (\alpha_1, \alpha_2, \cdots, \alpha_n) \in F_q^n,$$

if $\boldsymbol{b}_i\boldsymbol{b}_j \neq 0$ then we have

$$\sigma(\boldsymbol{b}_i\boldsymbol{b}_j) = \max\{i \mid \alpha_i \neq 0, \ \ 1 \leq i \leq n\}.$$

From Lemma 4.3.4, the ideal $N(s)$ is equal to s, therefore the matrix $[\sigma(\boldsymbol{b}_i\boldsymbol{b}_j)]$ becomes the following ideal form

$$\begin{bmatrix} 1 & 2 & 3 & 4 & \cdots & n-3 & n-2 & n-1 & n \\ 2 & 3 & 4 & 5 & \cdots & n-2 & n-1 & n & \\ 3 & 4 & 5 & 6 & \cdots & n-1 & n & & \\ 4 & 5 & 6 & 7 & \cdots & n & & & \\ \vdots & & & & & & & & \\ n & & & & & & & & \end{bmatrix}.$$

Particularly, for $(n = q-1, k)$ Reed-Solomon code C over $F_q(q = p^m)$, $d_{FR}(C) = n-k+1$ which is equal to the Singleton bound and the true minimum distance.

However, in case of binary linear code when $i = j$, we have $\sigma(\boldsymbol{b}_i\boldsymbol{b}_i) = \sigma(\boldsymbol{b}_i^2) = \sigma(\boldsymbol{b}_i) = i$, then the matrix $[\sigma(\boldsymbol{b}_i\boldsymbol{b}_j)]$ is

$$\begin{bmatrix} 1 & \sigma(\boldsymbol{b}_1\boldsymbol{b}_2) & \sigma(\boldsymbol{b}_1\boldsymbol{b}_3) & \cdots & \sigma(\boldsymbol{b}_1\boldsymbol{b}_n) \\ \sigma(\boldsymbol{b}_2\boldsymbol{b}_1) & 2 & \sigma(\boldsymbol{b}_2\boldsymbol{b}_3) & \cdots & \sigma(\boldsymbol{b}_2\boldsymbol{b}_n) \\ \sigma(\boldsymbol{b}_3\boldsymbol{b}_1) & \sigma(\boldsymbol{b}_3\boldsymbol{b}_2) & 3 & \cdots & \sigma(\boldsymbol{b}_3\boldsymbol{b}_n) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sigma(\boldsymbol{b}_n\boldsymbol{b}_1) & \sigma(\boldsymbol{b}_n\boldsymbol{b}_2) & \sigma(\boldsymbol{b}_n\boldsymbol{b}_n) & \cdots & n \end{bmatrix},$$

The next will consider only binary linear codes over F_2. The following lemma [[8], Lemma 3.3] and its corollary [[8], Corollary 3.4] are essential in our discussions in this section. They will be quoted in case of binary linear codes, although Miura proved them in case of linear codes over any F_q. Their proof will be given for the convenience of readers who will find difficulty in obtaining Miura's thesis [8].

Lemma 5.2.1 *Let $B_n = \{\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_n\}$ be an ordered basis of F_q^n. If $\sigma(\boldsymbol{b}_t\boldsymbol{b}) < t \leq n$, then there exists at least one i such that $\sigma(\boldsymbol{b}_t\boldsymbol{b}) \leq \sigma(\boldsymbol{b}_i\boldsymbol{b})$ and $1 \leq i < t$.*

Proof We will show a contradiction if we assume

$$\sigma(\boldsymbol{b}_i\boldsymbol{b}) < \sigma(\boldsymbol{b}_t\boldsymbol{b}) \quad \text{for} i = 1, 2, \cdots, t-1. \tag{5.1}$$

Let $\sigma(\boldsymbol{b}_t\boldsymbol{b}) = s < t$. We have

$$\boldsymbol{b}_t\boldsymbol{b} = \alpha_1\boldsymbol{b}_1 + \alpha_2\boldsymbol{b}_2 + \cdots + \alpha_s\boldsymbol{b}_s, \quad \alpha_s = 1. \tag{5.2}$$

$\alpha_i \in F_2$ for $1 \leq i \leq s$. From the assumption (5.1) we also have

$$\boldsymbol{b}_i\boldsymbol{b} = \beta_{i,1}\boldsymbol{b}_1 + \beta_{i,2}\boldsymbol{b}_2 + \cdots + \beta_{i,s-1}\boldsymbol{b}_{s-1}. \tag{5.3}$$

$\beta_{i,j} \in F_2$ for $1 \leq i < t, 1 \leq j < s$. We have $\boldsymbol{b}\boldsymbol{b} = \boldsymbol{b}$ for any binary vector $\boldsymbol{b}$ and we have

$$\begin{aligned} \boldsymbol{b}_t\boldsymbol{b} &= \boldsymbol{b}_t\boldsymbol{b}\boldsymbol{b} = (\boldsymbol{b}_t\boldsymbol{b})\boldsymbol{b} \\ &= (\alpha_1\boldsymbol{b}_1 + \alpha_2\boldsymbol{b}_2 + \cdots + \alpha_s\boldsymbol{b}_s)\boldsymbol{b} \\ &= \alpha_1\boldsymbol{b}_1\boldsymbol{b} + \alpha_2\boldsymbol{b}_2\boldsymbol{b} + \cdots + \alpha_s\boldsymbol{b}_s\boldsymbol{b} \end{aligned} \tag{5.4}$$

from (5.2). From (5.3) we have

$$\begin{aligned} \boldsymbol{b}_t\boldsymbol{b} &= \alpha_1(\beta_{1,1}\boldsymbol{b}_1 + \beta_{1,2}\boldsymbol{b}_2 + \cdots + \beta_{1,s-1}\boldsymbol{b}_{s-1}) \\ &\quad +\alpha_2(\beta_{2,1}\boldsymbol{b}_1 + \beta_{2,2}\boldsymbol{b}_2 + \cdots + \beta_{2,s-1}\boldsymbol{b}_{s-1}) \\ &\quad + \cdots + \alpha_s(\beta_{s,1}\boldsymbol{b}_1 + \beta_{s,2}\boldsymbol{b}_2 + \cdots + \beta_{s,s-1}\boldsymbol{b}_{s-1}) \\ &= (\alpha_1\beta_{1,1} + \alpha_2\beta_{2,1} + \cdots + \alpha_s\beta_{s,1})\boldsymbol{b}_1 \\ &\quad +(\alpha_1\beta_{1,2} + \alpha_2\beta_{2,2} + \cdots + \alpha_s\beta_{s,2})\boldsymbol{b}_2 \\ &\quad + \cdots + (\alpha_1\beta_{1,s-1} + \alpha_2\beta_{2,s-1} + \cdots + \alpha_s\beta_{s,s-1})\boldsymbol{b}_{s-1} \\ &= \sum_{j=1}^{s-1}(\sum_{i=1}^{s}\alpha_i\beta_{i,j})\boldsymbol{b}_j, \end{aligned}$$

then $\sigma(\boldsymbol{b}_t\boldsymbol{b}) < s$. However (5.2) and (5.4) are contradiction, since the term $\alpha_s\boldsymbol{b}_s$ in (5.2) is missing in the right-hand side of (5.4).

□

Corollary 5.2.1 *If $\sigma(\boldsymbol{b}_i\boldsymbol{b}_j) = s$ and $\boldsymbol{b}_i\boldsymbol{b}_j$ is well-behaved, then we have $1 \leq i, j \leq s$.*

Proof If we assume $i > s$, then application of Lemma 5.2.1 with $t = i$ and $\boldsymbol{b} = \boldsymbol{b}_j$ shows that $\boldsymbol{b}_i\boldsymbol{b}_j$ is not well-behaved from the definition of well-behavedness.

The proof of $j \leq s$ is the same.

□

Next we will prove the following theorem.

Theorem 5.2.1 *Any binary linear code has $d_{FR}(C, B_n)$ equal to either one or an even number.*

This theorem tells that binary linear codes can not have an odd $d_{FR}(C, B_n) \geq 3$. Our proof of Theorem 5.2.1 is very simple as shown below.

First we use the following property which is obvious from the definition.

Lemma 5.2.2 *If $\boldsymbol{b}_i\boldsymbol{b}_j (i \neq j)$ is well-behaved, then $\boldsymbol{b}_j\boldsymbol{b}_i$ is also well-behaved.*

Proof Since the property of symmetry, the proof of this Lemma is obvious.

□

Next we use the following lemma which is almost obvious from Corollary 5.2.1.

Lemma 5.2.3 *If $\boldsymbol{b}_i\boldsymbol{b}_i$ is well-behaved, then we have $N(i) = 1$.*

Proof For a binary vector $\boldsymbol{b}_i$ we have $\boldsymbol{b}_i\boldsymbol{b}_i = \boldsymbol{b}_i$ and $\sigma(\boldsymbol{b}_i\boldsymbol{b}_i) = i$. There can not exist another $\boldsymbol{b}_u\boldsymbol{b}_v, (u, v) \neq (i, i)$ which satisfies the condition that $\sigma(\boldsymbol{b}_u\boldsymbol{b}_v) = i$ and $\boldsymbol{b}_u\boldsymbol{b}_v$ is well-behaved, since such $\boldsymbol{b}_u$ and $\boldsymbol{b}_v$ must satisfy $1 \leq u, v \leq i$ from Corollary 5.2.1 and $\sigma(\boldsymbol{b}_u\boldsymbol{b}_v)$ must be less than i because of $\boldsymbol{b}_i\boldsymbol{b}_i$ being well-behaved.

Proof of Theorem 5.2.1 We have $N(1) = 1$ because of $\sigma(\boldsymbol{b}_1\boldsymbol{b}_1) = 1$.

If $\boldsymbol{b}_i\boldsymbol{b}_i$ is not well-behaved for $i \geq 2$, then $N(i)$ is even for $i \geq 2$ from Lemma 5.2.2.

If there exists such a $\boldsymbol{b}_i \in B_n \backslash B$ as $\boldsymbol{b}_i\boldsymbol{b}_i$ is well-behaved, then we have $N(i) = 1$ from Lemma 5.2.3 and $d_{FR}(C, B_n) = 1$ from the definition of $d_{FR}(C, B_n)$.

Therefore we have Theorem 5.2.1.

□

5.3 Conclusion

This Chapter showed that $d_{FR}(C, B_n)$ of binary linear codes can not take an odd number greater than or equal to 3 if we use Miura's definition [8] of $d_{FR}(C, B_n)$ for linear codes.

Recently Miura's definition of $d_{FR}(C, B_n)$ has been modified by Matsumoto [7] so that we can include the case where Lemma 5.2.2 does not hold. Therefore it is an interesting problem to examine whether Theorem 5.2.1 still holds or not if we use Matsumoto's definition of $d_{FR}(C, B_n)$ for linear codes. In Chapter 6 we will give some conjectures for Matsumoto's definition of $d_{FR}(C, B_n)$.

Chapter 6

Matsumoto's Definition and Some Conjectures

6.1 Introduction

Theorem 5.2.1 in the previous chapter is closely related to the fact that the matrix $[\sigma(\boldsymbol{b}_i\boldsymbol{b}_j)]$ is symmetric in Miura's definition [8] of d_{FR}.

Recently Matsumoto [7] slightly extended Miura's definition of d_{FR} by using three ordered basis $U_n = \{\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_n\}, V_n = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_n\}$ and $B_n = \{\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_n\}$ instead of the only one ordered basis B_n in case of Miura. This generalization induce that the matrix $[\sigma(\boldsymbol{u}_i\boldsymbol{v}_j)]$ is not symmetric in general.

In this chapter we will discuss the effect of the unsymmetry of $[\sigma(\boldsymbol{u}_i\boldsymbol{v}_j)]$ on the possible value of d_{FR} of binary cyclic codes. Our numerical experiments suggest a conjecture that this generalization is not so effective.

6.2 Matsumoto's Definition of $\hat{d}_{FR}$ and Some Conjectures

Miura's definition of the Feng-Rao designed minimum distance in Section 4.3, has been generalized to $\hat{d}_{FR}$ by Matsumoto [7].

Definition 6.2.1 *For a vector $\boldsymbol{b} \in F_q^n$, the map σ: $F_q^n \to \{0, 1, 2, \cdots, n\}$ is defined as*

$$\sigma(\boldsymbol{b}) = \min\{i | \boldsymbol{b} \in \mathrm{Span}\{B_i\},\ 0 \le i \le n\}.$$

Definition 6.2.2 *Let $\boldsymbol{u}_i \in U_n$, $\boldsymbol{v}_j \in V_n$. The product $\boldsymbol{u}_i\boldsymbol{v}_j$ of $\boldsymbol{u}_i$ and $\boldsymbol{v}_j$ for an ordered basis B_n is said to be well-behaved if $\sigma(\boldsymbol{u}_u\boldsymbol{v}_v) < \sigma(\boldsymbol{u}_i\boldsymbol{v}_j)$ for any $1 \le u \le i$, $1 \le v \le$*

j, $(u,v) \neq (i,j)$.

Definition 6.2.3 *For $1 \leq s \leq n$, we define $\hat{N}(s)$ as*

$$\hat{N}(s) = \sharp\left\{(i,j) \;\middle|\; \begin{array}{l} \sigma(\boldsymbol{u}_i\boldsymbol{v}_j) = s,\ 1 \leq i,j \leq n, \\ \boldsymbol{u}_i\boldsymbol{v}_j \text{ is well-behaved} \end{array}\right\}.$$

For an ordered basis B_n, U_n and V_n of F_q^n we define $\hat{N}(B_n, U_n, V_n)$ as $\hat{N}(B_n, U_n, V_n) = (\hat{N}(1), \hat{N}(2), \cdots, \hat{N}(n))$.

Our numerical experiments in case of all $(7,k)$ linear codes by generating all the possible set of three bases show $\hat{N}(1) \leq 1$ and $\hat{N}(2) \leq 2$. So we have the following conjecture.

Conjecture 6.2.1 *We have $0 \leq \hat{N}(s) \leq s$, for $1 \leq s \leq n$.*

Definition 6.2.4 *The Feng-Rao designed minimum distance by Matsumoto of the linear code $C(B_n, B)$ is denoted as $\hat{d}_{FR}(C, B_n, U_n, V_n)$ and defined as*

$$\hat{d}_{FR}(C, B_n, U_n, V_n) = \min\left\{\hat{N}(s) \;\middle|\; \begin{array}{l} \boldsymbol{b}_s \in B_n \backslash B. \\ 1 \leq s \leq n \end{array}\right\},$$

Definition 6.2.5 *For $\boldsymbol{y} = (y_1, y_2, \cdots, y_n) \in F_q^n$, the syndrome matrix $S(\boldsymbol{y})$ over F_q, is defined as*

$$\begin{aligned} S(\boldsymbol{y}) &= H(U_n)\mathrm{diag}(\boldsymbol{y})^t H(V_n) \\ &= \begin{bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \\ \vdots \\ \boldsymbol{u}_n \end{bmatrix} \begin{bmatrix} \boldsymbol{y}_1 & & & \\ & \boldsymbol{y}_2 & & 0 \\ & & \ddots & \\ 0 & & & \boldsymbol{y}_n \end{bmatrix}^t \begin{bmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \\ \vdots \\ \boldsymbol{v}_n \end{bmatrix}. \end{aligned}$$

Lemma 6.2.1 *For $1 \leq i,j \leq n$, we have $S(\boldsymbol{y}) = [\langle \boldsymbol{y}, \boldsymbol{u}_i\boldsymbol{v}_j \rangle]$.*

Proof We can refer to the proof of Lemma 4.4.1.

□

Lemma 6.2.2 *For $\boldsymbol{y} \in F_q^n$, we have* $\mathrm{wt}(\boldsymbol{y}) = \mathrm{rank}(\mathrm{diag}(\boldsymbol{y})) = \mathrm{rank}(S(\boldsymbol{y}))$.

Proof See the proof of Lemma 4.4.2.

□

Lemma 6.2.3 *Let d be the true minimum distance of $C(B_n, B)$. Then we have $d \geq \hat{d}_{FR}(C, B_n, U_n, V_n)$.*

Proof Let $\boldsymbol{c} \in C(B_n, B) = \mathrm{Span}\{\mathrm{B}\}^{\perp}$, $\boldsymbol{c} \neq 0$. Let

$$\langle \boldsymbol{c}, \boldsymbol{b}_1 \rangle = \langle \boldsymbol{c}, \boldsymbol{b}_2 \rangle = \cdots = \langle \boldsymbol{c}, \boldsymbol{b}_{s-1} \rangle = 0,$$

and $\langle \boldsymbol{c}, \boldsymbol{b}_s \rangle \neq 0$, then $\boldsymbol{b}_s \in B_n \backslash B$. For

$$^{\forall}(k,l) \in \{(i,j) | \sigma(\boldsymbol{u}_i \boldsymbol{v}_j) = s,\ 1 \leq i,j \leq n, \boldsymbol{u}_i \boldsymbol{v}_j \text{is well-behaved}\}$$

the elements $\langle \boldsymbol{c}, \boldsymbol{u}_k \boldsymbol{v}_l \rangle$ of $S(\boldsymbol{c})$ are nonzero, because of

$$\begin{aligned} \langle \boldsymbol{c}, \boldsymbol{u}_k \boldsymbol{v}_l \rangle &= \langle \boldsymbol{c}, \sum_{t=1}^{s} \alpha_t \boldsymbol{b}_t \rangle \\ &= \sum_{t=1}^{s} \alpha_t \langle \boldsymbol{c}, \boldsymbol{b}_t \rangle \\ &= \alpha_s \langle \boldsymbol{c}, \boldsymbol{b}_s \rangle + \sum_{t=1}^{s-1} \alpha_t \langle \boldsymbol{c}, \boldsymbol{b}_t \rangle \\ &= \alpha_s \langle \boldsymbol{c}, \boldsymbol{b}_s \rangle \\ &\neq 0. \end{aligned}$$

On the other hand, for $u = i, 1 \leq v < j$ or $1 \leq u < i, v = j$, $\langle \boldsymbol{c}, \boldsymbol{u}_u \boldsymbol{v}_v \rangle = 0$. The i-th column and j-th row of $[\langle \boldsymbol{c}, \boldsymbol{u}_i \boldsymbol{v}_j \rangle]$ are linearly independent. Therefor at least the matrix $[\langle \boldsymbol{c}, \boldsymbol{u}_i \boldsymbol{v}_j \rangle]$ has $\hat{N}(s)$ independent row or column. From Lemma 6.2.1 and Lemma 6.2.2, we have

$$\mathrm{wt}(\boldsymbol{c}) = \mathrm{rank}(S(\boldsymbol{c})) = \mathrm{rank}([\langle \boldsymbol{c}, \boldsymbol{u}_i \boldsymbol{v}_j \rangle]) \geq \hat{N}(s).$$

Moreover, from the definition of $\hat{d}_{FR}(C, B_n, U_n, V_n)$, we have

$$d \geq \hat{d}_{FR}(C, B_n, U_n, V_n).$$

□

Definition 6.2.6 *For a linear code C we define the set of all triples with three ordered bases such that C can be defined by an ordered basis B_n, i.e.,*

$$\mathcal{T}(C) = \left\{ (B_n, U_n, V_n) \;\middle|\; \begin{array}{l} ^{\exists} B \subseteq B_n \text{ s.t. } C = C(B_n, B) \text{ and} \\ U_n, V_n \text{ are ordered bases of } F_q^n \end{array} \right\}.$$

Note that B is uniquely determined from C and B_n. Moreover U_n and V_n are not concerned with the linear code C. The purpose is to give an optimum triple of three ordered bases (B_n, U_n, V_n) for a given linear code C.

Definition 6.2.7 *The Feng-Rao designed minimum distance $\hat{d}_{FR}(C)$ of C by Matsumoto is defined as*

$$\hat{d}_{FR}(C) = \max\{\hat{d}_{FR}(C, B_n, U_n, V_n) | (B_n, U_n, V_n) \in \mathcal{T}(C)\}.$$

The triple of three ordered bases (B_n, U_n, V_n) satisfying

$$\hat{d}_{FR}(C) = \hat{d}_{FR}(C, B_n, U_n, V_n)$$

is called an optimum triple for C.

The next shows an example of a triple of three bases in order to discuss Matsumoto's generalization of $\hat{d}_{FR}$.

Example 6.2.1 *For (7,4) binary linear code, let three ordered basis be as follows:*

$$\begin{aligned}
\boldsymbol{b}_1 = \boldsymbol{u}_1 &= (1\ 0\ 1\ 1\ 1\ 0\ 0),\\
\boldsymbol{b}_2 = \boldsymbol{u}_2 &= (0\ 1\ 0\ 1\ 1\ 1\ 0),\\
\boldsymbol{b}_3 = \boldsymbol{u}_3 &= (0\ 0\ 1\ 0\ 1\ 1\ 1),\\
\boldsymbol{b}_4 = \boldsymbol{u}_4 &= (0\ 0\ 0\ 1\ 1\ 0\ 0),\\
\boldsymbol{b}_5 = \boldsymbol{u}_5 &= (0\ 0\ 0\ 0\ 0\ 1\ 1),\\
\boldsymbol{b}_6 = \boldsymbol{u}_6 &= (0\ 0\ 0\ 1\ 0\ 1\ 0),\\
\boldsymbol{b}_7 = \boldsymbol{u}_7 &= (0\ 0\ 0\ 0\ 0\ 1\ 0),
\end{aligned}$$

and

$$\begin{aligned}
\boldsymbol{v}_1 &= (0\ 0\ 0\ 0\ 0\ 1\ 0),\\
\boldsymbol{v}_2 &= (0\ 0\ 0\ 0\ 0\ 1\ 1),\\
\boldsymbol{v}_3 &= (1\ 0\ 1\ 1\ 1\ 0\ 0),\\
\boldsymbol{v}_4 &= (0\ 0\ 0\ 1\ 1\ 0\ 0),\\
\boldsymbol{v}_5 &= (0\ 1\ 0\ 1\ 1\ 1\ 0),\\
\boldsymbol{v}_6 &= (0\ 0\ 1\ 0\ 1\ 1\ 1),\\
\boldsymbol{v}_7 &= (0\ 0\ 0\ 1\ 0\ 1\ 0).
\end{aligned}$$

Then the matrix of $\sigma(\boldsymbol{u}_i\boldsymbol{v}_j)$ is

$$\begin{bmatrix}
0 & 0 & 1 & 4 & 4 & 5 & 7\\
7 & 7 & 4 & 4 & 2 & 6 & 6\\
7 & 5 & 5 & 7 & 6 & 3 & 7\\
0 & 0 & 4 & 4 & 4 & 7 & 7\\
7 & 5 & 0 & 0 & 7 & 5 & 7\\
7 & 7 & 7 & 7 & 6 & 7 & 6\\
7 & 7 & 0 & 0 & 7 & 7 & 7
\end{bmatrix}.$$

Above matrix $\sigma(\boldsymbol{u}_i\boldsymbol{v}_j)$ shows that $\sigma(\boldsymbol{u}_1\boldsymbol{v}_3) = 1$ and $\boldsymbol{u}_1\boldsymbol{v}_3$ is well-behaved, which contradicts with Corollary 5.2.1. Therefore in Matsumoto's definition we don't have Lemma 5.2.1 and Corollary 5.2.1, which are used in proving Theorem 5.2.1.

□

However our numerical experiments on $(7, k)$ binary codes strongly suggest the following conjecture which is the same our previous Theorem 5.2.1.

Conjecture 6.2.2 *Any binary linear code has $\hat{d}_{FR}(C, B_n, U_n, V_n)$ equal to either one or an even number.*

6.3 Conclusion

In this chapter we discussed Miura's definition and Masumoto's definition of the Feng-Rao designed minimum distance for binary linear codes. Matsumoto's definition is a generalization of Miura's definition. Some properties and examples induce some conjectures which tell Matsumoto's generalization is not so effective compared with Miura's definition for binary linear codes.

Future works are giving proofs for these conjectures and investigate nonbinary linear codes.

Chapter 7

The Feng-Rao Designed Minimum Distance of Cyclic Codes

7.1 Introduction

This chapter is a first trial to find an optimum ordered basis B_n for (n, k) cyclic codes over F_q, which are in a class of the most useful linear codes. For a cyclic code C there exists a good designed minimum distance such as BCH designed minimum distance denoted by d_{BCH}. It is interesting to compare $d_{FR}(C)$ with d_{BCH} of a cyclic code C.

The "Type I" ordered basis B_n will be introduced, which corresponds to the well-known form of the parity check matrix of an (n, k) cyclic code expressed by its parity check polynomial, i.e., we use a natural choice of B, a subset of B_n, as $(n-k)$ vectors $\{\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_{n-k}\}$, which consists of a permutation of vectors corresponding to the check polynomial and its $(n-k-1)$ consecutive right cyclic shifts. The possible values of d_{FR} of an (n, k) cyclic code will be investigated when we use the Type I ordered basis B_n.

Firstly, it is shown that for nonbinary cyclic codes, i.e., in case of $q \neq 2$ we have $d_{FR}(C, B_n) \leq 1$ if B_n is Type I and the check polynomial $h(x)$ has a coefficient $\neq 0, 1$.

Secondly, for binary cyclic codes, i.e., in case of $q = 2$ it is shown that $d_{FR}(C, B_n) = n - 1$ for $k = 1$ (repetition code), $d_{FR}(C, B_n) \geq 2(\frac{n}{3} - 1)$ for $k = 2$ if B_n is Type I, and $d_{FR}(C, B_n) = 2$ for $k = n - 1$ (parity code), respectively.

7.2 The Feng-Rao Designed Minimum Distance of Cyclic Codes and Type I Ordered Basis

In this section the Feng-Rao designed minimum distance $d_{FR}(C)$ for (n,k) cyclic code C over F_q is investigated. We are interest in the relation between the choice of ordered basis $B_n = \{\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_n\}$ and the Feng-Rao designed minimum distance $d_{FR}(C, B_n)$, since the value of $d_{FR}(C, B_n)$ depends on the choice of B_n [8]. If we define $d_{FR}(C)$ as the maximum value of $d_{FR}(C, B_n)$ among all the possible choices of B_n for C, it is believed that $d_{FR}(C)$ is a good lower bound of the minimum distance of C.

For a fixed linear code C we can choose many bases B_n such that $C = C(B_n, B)$.

Definition 7.2.1 *For a cyclic code C we define the set $\mathcal{B}(C)$ of all ordered bases such that C can be defined by this ordered basis, i.e.,*

$$\mathcal{B}(C) = \{B_n | {}^{\exists} B \subseteq B_n \ s.t.\ C = C(B_n, B)\}.$$

Note that B is uniquely determined from C and B_n. Our purpose is to give an optimum ordered basis B_n for a given cyclic code C.

Definition 7.2.2 *The Feng-Rao designed minimum distance $d_{FR}(C)$ of C is defined as*

$$d_{FR}(C) = \max\{d_{FR}(C, B_n) | B_n \in \mathcal{B}(C)\}.$$

The ordered basis B_n satisfying $d_{FR}(C) = d_{FR}(C, B_n)$ is called an optimum ordered basis for C.

The computation of $d_{FR}(C)$ needs the choice of an ordered basis of F_q^n, i.e.,

$$B_n = \{\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_n\}. \tag{7.1}$$

In this chapter we will consider an (n,k) cyclic code C over a base field F_q. Let $g(x)$ be the generator polynomial of C. Let

$$h(x) = (x^n - 1)/g(x) = x^k + b_2 x^{k-1} + \cdots + b_{k+1} \tag{7.2}$$

be the check polynomial of C. It is well known that C consists of all vectors in F_q^n orthogonal to

$$\boldsymbol{b} = (1, b_2, \cdots, b_{k+1}, 0, \cdots, 0) \in F_q^n, \quad b_{k+1} \neq 0 \tag{7.3}$$

and

$$\delta(\boldsymbol{b}), \cdots, \delta^{n-k-1}(\boldsymbol{b}),$$

where $\delta(\boldsymbol{c}) = (c_n, c_1, \cdots, c_{n-1})$ for $\boldsymbol{c} = (c_1, c_2, \cdots, c_n)$ is the right cyclic shift of $\boldsymbol{c}$. Therefore without loss of generality we will assume that

$$C = \mathrm{Span}\{\boldsymbol{b}, \delta(\boldsymbol{b}), \cdots, \delta^{n-k-1}(\boldsymbol{b})\}^{\perp}. \tag{7.4}$$

There are many choices for B_n. From Lemma 4.3.4 and Definition 4.3.5 we need to set $\boldsymbol{b}_s \in B$ for small s in order to obtain a large value of d_{FR}.

The following B_n is a natural choice for computing $d_{FR}(C, B_n)$ in case of (n, k) cyclic codes C.

Definition 7.2.3 *The ordered basis B_n is called "Type I" if $B = \{\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_{n-k}\}$ is a permutation of*

$$\{\boldsymbol{b}, \delta(\boldsymbol{b}), \cdots, \delta^{n-k-1}(\boldsymbol{b})\},$$

where $\boldsymbol{b}$ is the vector of (7.3).

7.3 The Feng-Rao Designed Minimum Distance of Nonbinary Cyclic Codes

First the following proposition will be proved about the product of two vectors of B.

Proposition 7.3.1 *Let $B = \{\boldsymbol{b}_{u_1}, \boldsymbol{b}_{u_2}, \cdots, \boldsymbol{b}_{u_{n-k}}\}$ be a subset of an ordered basis B_n with Type I. For $1 \leq i \leq n-k, 1 \leq j \leq n-k$, we have following two statements:*

1. *$\boldsymbol{b}_{u_i}\boldsymbol{b}_{u_i} = \alpha \boldsymbol{b}_{u_i}$ with $\alpha \in F_q \backslash \{0\}$ or $\boldsymbol{b}_{u_i}\boldsymbol{b}_{u_i} \notin \mathrm{Span}\{B\}$.*
2. *if $i \neq j$, then $\boldsymbol{b}_{u_i}\boldsymbol{b}_{u_j} = \boldsymbol{0}$ or $\boldsymbol{b}_{u_i}\boldsymbol{b}_{u_j} \notin \mathrm{Span}\{B\}$.*

Proof Let $\boldsymbol{t}_1 = \boldsymbol{b}$ and $\boldsymbol{t}_i = \delta^{i-1}(\boldsymbol{b})$ for $i = 2, 3, \cdots, n-k$. From Definition 7.2.3 we have

$$B = \{\boldsymbol{b}_{u_1}, \boldsymbol{b}_{u_2}, \cdots, \boldsymbol{b}_{u_{n-k}}\} = \{\boldsymbol{t}_1, \boldsymbol{t}_2, \cdots, \boldsymbol{t}_{n-k}\}.$$

We will first prove 1. We have

$$\boldsymbol{t}_1\boldsymbol{t}_1 = (1^2, \cdots, b_{k+1}^2, 0, \cdots, 0) \neq \boldsymbol{0} \tag{7.5}$$

When we assume $\boldsymbol{t}_1\boldsymbol{t}_1 \in \mathrm{Span}\{B\}$, we have

$$\boldsymbol{t}_1\boldsymbol{t}_1 = \sum_{i=1}^{n-k} \beta_i \boldsymbol{t}_i.$$

In order to satisfy (7.5), we need $\beta_i = 0$ for $i = n-k, n-k-1, \cdots, 2$ from (7.3) and (7.5). Then we have $\beta_1 = \alpha \neq 0$ from $\boldsymbol{t}_1\boldsymbol{t}_1 \neq \boldsymbol{0}$. Consequently we have $\boldsymbol{t}_1\boldsymbol{t}_1 = \alpha\boldsymbol{t}_1$. It is obvious that $\boldsymbol{t}_1\boldsymbol{t}_1 \notin \mathrm{Span}\{B\}$ if $\boldsymbol{t}_1\boldsymbol{t}_1 \neq \alpha\boldsymbol{t}_1$.

The proof of $\boldsymbol{t}_i\boldsymbol{t}_i = \alpha\boldsymbol{t}_i$ or $\boldsymbol{t}_i\boldsymbol{t}_i \notin \mathrm{Span}\{B\}$ is similar, since $\boldsymbol{t}_i\boldsymbol{t}_i = \alpha\boldsymbol{t}_i$ is equivalent to $\boldsymbol{t}_1\boldsymbol{t}_1 = \alpha\boldsymbol{t}_1$ from $\boldsymbol{t}_i = \delta^{i-1}(\boldsymbol{t}_1)$ and $\boldsymbol{t}_i\boldsymbol{t}_i = \delta^{i-1}(\boldsymbol{t}_1\boldsymbol{t}_1)$.

Next we will prove 2. Since $\boldsymbol{t}_i\boldsymbol{t}_j = \boldsymbol{t}_j\boldsymbol{t}_i$ and

$$\boldsymbol{t}_i\boldsymbol{t}_j = \delta(\boldsymbol{t}_{i-1}\boldsymbol{t}_{j-1}) = \delta^{i-1}(\boldsymbol{t}_1\boldsymbol{t}_{j-i+1})$$

for $i < j$, it is enough to prove the case of $i = 1$ and $2 \leq j \leq n-k$. We have

$$\boldsymbol{t}_1\boldsymbol{t}_j = (0^{j-1}, *, \cdots, *, 0^{n-k-1}), \tag{7.6}$$

where 0^u denotes the zero vector of length u and $*$ means any value in F_q. When we assume $\boldsymbol{t}_1\boldsymbol{t}_j \in \mathrm{Span}\{B\}$, we have

$$\boldsymbol{t}_1\boldsymbol{t}_j = \sum_{i=1}^{n-k} \gamma_i \boldsymbol{t}_i.$$

From (7.3) and (7.6) we have $\boldsymbol{t}_1\boldsymbol{t}_j = \boldsymbol{0}$.

□

The following proposition shows that the choice of an ordered basis B_n with Type I is worst in many cases of nonbinary cyclic codes.

Proposition 7.3.2 *Let C be a nonbinary (n,k) cyclic code. We have*

$$d_{FR}(C, B_n) \leq 1, \tag{7.7}$$

if B_n is Type I and the check polynomial $h(x)$ of (7.2)has a coefficient $b_i \neq 0, 1$.

Proof If $\boldsymbol{b}_1 \notin B$ we have $d_{FR}(C, B_n) \leq 1$ from Lemma 4.3.4 and Definition 4.3.5. Then we consider the case of $\boldsymbol{b}_1 \in B$. From Proposition 7.3.1,

$$\begin{aligned} \boldsymbol{b}_1\boldsymbol{b}_1 &= (0, \cdots, 0, 1^2, b_2^2, \cdots, b_{k+1}^2, 0, \cdots, 0) \\ &= \alpha\boldsymbol{b}_1, \quad \alpha \in F_q \backslash \{0\} \end{aligned} \tag{7.8}$$

or

$$\boldsymbol{b}_1\boldsymbol{b}_1 \notin \text{Span}\{B\}. \tag{7.9}$$

From (7.8) we need $b_i = 0$ or α for all $i = 1, 2, \cdots, k+1$. Since $b_1 = 1 \neq 0$, we have $\alpha = 1$. If there exists a $b_i \neq 0, 1$ with $i = 2, 3, \cdots, k+1$, then (7.8) can not hold.

In case of (7.9), we have $\sigma(\boldsymbol{b}_1\boldsymbol{b}_1) = r > 1$ and $N(r) = 1$, which means $d_{FR}(C, B_n) \leq 1$ because of $\boldsymbol{b}_r \in B_n \backslash B$.

□

Proposition 7.3.2 shows that the choice of any Type I ordered basis is not good for most of nonbinary cyclic codes, although no efficient method has been known for computing good ordered basis. Another possible approach is to modify the definition of d_{FR} for linear codes[7].

7.4 The Feng-Rao Designed Minimum Distance of Binary Cyclic Codes

We consider (n, k) cyclic codes over F_2 with an odd length n and $k = 1$, $k = 2$ and $k = n - 1$ in this section. We will show that there exists not so bad ordered basis with Type I.

7.4.1 $(n, 1)$ Binary Cyclic Code (Repetition Code)

In this case the code has only two code words, i.e., $\mathbf{0} = (0, 0, \cdots, 0)$ and $\mathbf{1} = (1, 1, \cdots, 1)$. Since its generator polynomial $g(x) = x^{n-1} + x^{n-2} + \cdots + x + 1$ has $n - 1$ roots, i.e., α, α^2, $\cdots$, α^{n-1} with α being a primitive n-th root of unity, we have

$$d_{BCH} = n. \tag{7.10}$$

Lemma 7.4.1 *For a binary repetition code C ($k = 1$), we have $d_{FR}(C, B_n) \leq n - 1$.*

Proof If $\boldsymbol{b}_n \in B$, then we have $B_n \backslash B = \{\boldsymbol{b}_s\}(1 \leq s \leq n - 1)$ and $d_{FR}(C, B_n) = N(s) \leq n - 1$ from Lemma 4.3.4.

Therefore we consider the case where $B_n \backslash B = \{\boldsymbol{b}_n\}$ and $d_{FR}(C, B_n) = N(n)$. We have $N(n) = n$ if and only if

$$\begin{cases} \sigma(\boldsymbol{b}_i\boldsymbol{b}_j) < n & \text{if } i + j < n + 1, \\ \sigma(\boldsymbol{b}_i\boldsymbol{b}_{n+1-i}) = n & \text{if } 1 \leq i \leq n. \end{cases} \tag{7.11}$$

For a binary vector $\boldsymbol{b}_i$ we have $\boldsymbol{b}_i\boldsymbol{b}_i = \boldsymbol{b}_i$ and $\sigma(\boldsymbol{b}_{\frac{n+1}{2}}\boldsymbol{b}_{\frac{n+1}{2}}) = \frac{n+1}{2} < n$. Therefore (7.11) can not hold.

□

Proposition 7.4.1 *For a binary $(n,1)$ cyclic code C, we have $d_{FR}(C, B_n) = n-1$.*

Proof We will show the existence of a Type I ordered basis $B_n = \{\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_n\}$ which satisfies $d_{FR}(C, B_n) = n-1$. Let

$$\boldsymbol{e}_i = (0^{i-1}, 1, 0^{n-i}) \in F_2^n, \tag{7.12}$$

From $h(x) = (x^n - 1)/g(x) = x + 1$, we can choose B_n as

$$\begin{cases} \boldsymbol{b}_i = \boldsymbol{e}_{2i-1} + \boldsymbol{e}_{2i} & \text{for } 1 \le i \le \frac{n-1}{2}, \\ \boldsymbol{b}_{\frac{n-1}{2}+i} = \boldsymbol{e}_{n-(2i-2)} + \boldsymbol{e}_{n-(2i-1)} & \text{for } 1 \le i \le \frac{n-1}{2}, \\ \boldsymbol{b}_n = \boldsymbol{e}_1, \end{cases}$$

and $B = B_{n-1}$. Since $\boldsymbol{e}_i \notin \mathrm{Span}\{B_{n-1}\}$ for $i = 1, 2, \cdots, n$, we have

$$\begin{cases} \sigma(\boldsymbol{b}_i\boldsymbol{b}_i) = i & \text{if } 1 \le i \le n, \\ \sigma(\boldsymbol{b}_i\boldsymbol{b}_j) = 0 & \text{if } 1 \le i < j \le n-1-i, \\ \sigma_(\boldsymbol{b}_i\boldsymbol{b}_{n-i}) = n & \text{if } 1 \le i \le n-1. \end{cases} \tag{7.13}$$

We have $N(n) \ge n-1$ and $d_{FR}(C, B_n) \ge n-1$. From Lemma 7.4.1 we have $d_{FR}(C, B_n) = n-1$.

□

Therefore from (7.10) and Proposition 7.4.1, we have $\lfloor \frac{d_{FR}-1}{2} \rfloor = \lfloor \frac{d_{BCH}-1}{2} \rfloor - 1$.

Example 7.4.1 *For the binary (7,1) cyclic code with the check polynomial $h(x) = x+1$, we can choose $B = \{\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_6\}$ and $B_n \backslash B = \{\boldsymbol{b}_7\}$ as follows:*

$$\begin{aligned} \boldsymbol{b}_1 &= (1\ 1\ 0\ 0\ 0\ 0\ 0), \\ \boldsymbol{b}_2 &= (0\ 0\ 1\ 1\ 0\ 0\ 0), \\ \boldsymbol{b}_3 &= (0\ 0\ 0\ 0\ 1\ 1\ 0), \\ \boldsymbol{b}_4 &= (0\ 0\ 0\ 0\ 0\ 1\ 1), \\ \boldsymbol{b}_5 &= (0\ 0\ 0\ 1\ 1\ 0\ 0), \\ \boldsymbol{b}_6 &= (0\ 1\ 1\ 0\ 0\ 0\ 0), \\ \boldsymbol{b}_7 &= (1\ 0\ 0\ 0\ 0\ 0\ 0). \end{aligned}$$

Since the matrix of $\sigma(\boldsymbol{b}_i\boldsymbol{b}_j)$ is

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 7 & 7 \\ 0 & 2 & 0 & 0 & 7 & 7 & 0 \\ 0 & 0 & 3 & 7 & 7 & 0 & 0 \\ 0 & 0 & 7 & 4 & 0 & 0 & 0 \\ 0 & 7 & 7 & 0 & 5 & 0 & 0 \\ 7 & 7 & 0 & 0 & 0 & 6 & 0 \\ 7 & 0 & 0 & 0 & 0 & 0 & 7 \end{bmatrix},$$

we have $d_{FR}(C, B_n) = 6$. *From* $d_{BCH} = 7$, *we have* $d_{FR}(C, B_n) = d_{BCH} - 1$.

□

7.4.2 $(n, 2)$ Binary Cyclic Code

Let C be a binary $(n, 2)$ cyclic code. In this case, it is obvious that the check polynomial of C is only $h(x) = x^2 + x + 1$. We have $x^3 - 1 | x^n - 1$ and $3|n$, since $h(x)|x^n - 1$, $x - 1|x^n - 1$, $x^3 - 1 = (x - 1)h(x)$, and $x^n - 1 = x^r(x^3 - 1)(x^{3(m-1)} + x^{3(m-2)} + \cdots + x^3 + 1) + (x^r - 1)$ for $n = 3m + r$ and $0 \leq r \leq 2$.

Lemma 7.4.2 *The odd code length of binary* $(n, 2)$ *cyclic codes* C *is a multiple of 3.*

Let α be a primitive n-th root of unity, then two roots of the check polynomial of C are $\alpha^{n/3}$ and $\alpha^{2n/3} = \alpha^{-n/3}$. Since the number of consecutive roots of the generator polynomial of C is $\frac{2}{3}n - 1$, we have

$$d_{BCH} = \frac{2}{3}n. \tag{7.14}$$

Proposition 7.4.2 *For a binary* $(n, 2)$ *cyclic code* C, *we have*

$$d_{FR}(C, B_n) \geq 2(\frac{n}{3} - 1) \tag{7.15}$$

with a Type I ordered basis B_n.

Proof A lower bound of (7.15) is showed by finding an ordered basis B_n satisfying equation (7.15). Let $B_n = \{\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_n\}$ be an ordered basis with

$$\begin{cases} \boldsymbol{b}_i = \boldsymbol{e}_{3i-2} + \boldsymbol{e}_{3i-1} + \boldsymbol{e}_{3i} \quad \text{for} \quad 1 \leq i \leq \frac{n}{3}, \\ \boldsymbol{b}_{\frac{n}{3}+2k-1} = \boldsymbol{e}_{n-3k} + \boldsymbol{e}_{n-3k+1} + \boldsymbol{e}_{n-3k+2} \\ \qquad\qquad\qquad\qquad \text{for} \quad 1 \leq k \leq \frac{n}{3} - 1, \\ \boldsymbol{b}_{\frac{n}{3}+2k} = \boldsymbol{e}_{n-3k-1} + \boldsymbol{e}_{n-3k} + \boldsymbol{e}_{n-3k+1} \\ \qquad\qquad\qquad\qquad \text{for} \quad 1 \leq k \leq \frac{n}{3} - 1, \\ \boldsymbol{b}_{n-1} = \boldsymbol{e}_3, \qquad \boldsymbol{b}_n = \boldsymbol{e}_2, \end{cases}$$

where $\boldsymbol{e}_i$ is defined by (7.12). From this we have $B = B_{n-2}$ and $d_{FR}(C, B_n) = \min\{N(n-1), N(n)\}$. We can write $\boldsymbol{e}_{3i}$, $\boldsymbol{e}_{3i-1}$ and $\boldsymbol{e}_{3i-2}$ for $1 \leq i \leq \frac{n}{3}$ as

$$\boldsymbol{e}_{3i} = \boldsymbol{b}_{n-1} + (\sharp), \quad \boldsymbol{e}_{3i-1} = \boldsymbol{b}_n + (\sharp), \quad \boldsymbol{e}_{3i-2} = \boldsymbol{b}_n + \boldsymbol{b}_{n-1} + (\sharp),$$

where $(\sharp)$ represents a vector in $\mathrm{Span}\{B_{n-2}\}$. Therefore we have

$$\begin{cases} \sigma(\boldsymbol{b}_i\boldsymbol{b}_j) = 0, \quad \text{if} \quad i+1 \leq j < n-2i-1, \\ \sigma(\boldsymbol{b}_i\boldsymbol{b}_{n-2i-1}) = n-1, \\ \sigma(\boldsymbol{b}_i\boldsymbol{b}_{n-2i}) = n \end{cases} \tag{7.16}$$

for $1 \leq i \leq \frac{n}{3} - 1$, since $\boldsymbol{b}_i\boldsymbol{b}_j = \boldsymbol{0}$ for $i+1 \leq j < n-2i-1$, $\boldsymbol{b}_i\boldsymbol{b}_{n-2i-1} = \boldsymbol{e}_{3i}$, and $\boldsymbol{b}_i\boldsymbol{b}_{n-2i} = \boldsymbol{e}_{3i} + \boldsymbol{e}_{3i-1}$.

Hence we have $N(n-1) = N(n) = 2(\frac{n}{3} - 1)$ and $d_{FR}(C, B_n) = 2(\frac{n}{3} - 1)$.

□

Example 7.4.2 *For the binary (9,2) cyclic code with the check polynomial* $h(x) = x^2 + x + 1$, *we can choose* $B = \{\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_7\}$ *and* $B_n \backslash B = \{\boldsymbol{b}_8, \boldsymbol{b}_9\}$ *as follows:*

$$\begin{array}{lcccccccccc} \boldsymbol{b}_1 = & (1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0), \\ \boldsymbol{b}_2 = & (0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0), \\ \boldsymbol{b}_3 = & (0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1), \\ \boldsymbol{b}_4 = & (0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0), \\ \boldsymbol{b}_5 = & (0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0), \\ \boldsymbol{b}_6 = & (0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0), \\ \boldsymbol{b}_7 = & (0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0), \\ \boldsymbol{b}_8 = & (0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0), \\ \boldsymbol{b}_9 = & (0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0). \end{array}$$

The matrix $\sigma(\boldsymbol{b}_i\boldsymbol{b}_j)$ *is*

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 8 & 9 & 8 & 9 \\ 0 & 2 & 0 & 8 & 9 & 8 & 9 & 0 & 0 \\ 0 & 0 & 3 & 8 & 9 & 0 & 0 & 0 & 0 \\ 0 & 8 & 8 & 4 & 9 & 0 & 0 & 0 & 0 \\ 0 & 9 & 9 & 9 & 5 & 9 & 0 & 0 & 0 \\ 8 & 8 & 0 & 0 & 9 & 6 & 9 & 8 & 0 \\ 9 & 9 & 0 & 0 & 0 & 9 & 7 & 8 & 9 \\ 8 & 0 & 0 & 0 & 0 & 8 & 8 & 8 & 0 \\ 9 & 0 & 0 & 0 & 0 & 0 & 9 & 0 & 9 \end{bmatrix}. \tag{7.17}$$

and we have $d_{FR}(C, B_n) = 4$. *On the other hand we have* $d_{BCH} = 6$. □

Although Proposition 7.4.2 gives a lower bound of $d_{FR}(C, B_n)$ for a binary $(n, 2)$ cyclic code C with a Type I ordered basis B_n, the following consideration suggests us to conjecture that the lower bound is also an upper bound. Let $\boldsymbol{b}_i' = \delta^{i-1}(\boldsymbol{b})$ $(1 \leq i \leq 7)$, i.e.,

$$\begin{aligned}
\boldsymbol{b}_1' &= (1\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0),\\
\boldsymbol{b}_2' &= (0\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 0),\\
\boldsymbol{b}_3' &= (0\ 0\ 1\ 1\ 1\ 0\ 0\ 0\ 0),\\
\boldsymbol{b}_4' &= (0\ 0\ 0\ 1\ 1\ 1\ 0\ 0\ 0),\\
\boldsymbol{b}_5' &= (0\ 0\ 0\ 0\ 1\ 1\ 1\ 0\ 0),\\
\boldsymbol{b}_6' &= (0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 0),\\
\boldsymbol{b}_7' &= (0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1).
\end{aligned}$$

From this for each i $(1 \leq i \leq 7)$ the number of the pair $\{\boldsymbol{b}_i', \boldsymbol{b}_j'\}$ $(i \neq j)$ satisfying $\boldsymbol{b}_i'\boldsymbol{b}_j' \neq \boldsymbol{0}$ $(1 \leq j \leq 7)$ is equal to 2 $(i = 1, 7)$, 3 $(i = 2, 6)$, or 4 $(i = 3, 4, 5)$. For such a pair we have

$$\boldsymbol{0} \neq \boldsymbol{b}_i'\boldsymbol{b}_j' \in B_n \backslash B = \{\boldsymbol{b}_8, \boldsymbol{b}_9\}$$

from Proposition 7.3.1. Therefore we can make $d_{FR}(C, B_n) = 6 > 4$, if $\sigma(\boldsymbol{b}_i\boldsymbol{b}_j)$ is the form

$$\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 8 & 9 & * & *\\
0 & 2 & 0 & 0 & 8 & 9 & * & * & *\\
0 & 0 & 3 & 8 & 9 & * & * & * & *\\
0 & 0 & 8 & 4 & * & * & * & * & *\\
0 & 8 & 9 & * & 5 & * & * & * & *\\
8 & 9 & * & * & * & 6 & * & * & *\\
9 & * & * & * & * & * & 7 & * & *\\
* & * & * & * & * & * & * & 8 & *\\
* & * & * & * & * & * & * & * & 9
\end{bmatrix}, \tag{7.18}$$

which gives $N(8) = N(9) = 6$.

The first row of (7.18) requests that $\boldsymbol{b}_1$ should be $\boldsymbol{b}_1'$ or $\boldsymbol{b}_7'$. We consider the case of $\boldsymbol{b}_1 = \boldsymbol{b}_1'$. The case of $\boldsymbol{b}_1 = \boldsymbol{b}_7'$ is similar.

From the form of the first row of (7.18), we have $\boldsymbol{b}_6 = \boldsymbol{b}_2'$, $\boldsymbol{b}_7 = \boldsymbol{b}_3'$ or $\boldsymbol{b}_6 = \boldsymbol{b}_3'$, $\boldsymbol{b}_7 = \boldsymbol{b}_2'$. However any choice of $\boldsymbol{b}_2 \in \{\boldsymbol{b}_4', \boldsymbol{b}_5', \boldsymbol{b}_6', \boldsymbol{b}_7'\}$ can not satisfy the form of the second row of (7.18). Therefore (7.18) is impossible. From the symmetric property $\sigma(\boldsymbol{b}_i\boldsymbol{b}_j) = \sigma(\boldsymbol{b}_j\boldsymbol{b}_i)$ and $\sigma(\boldsymbol{b}_i\boldsymbol{b}_i) = i$ $(1 \leq i \leq n)$ for binary codes, the value of $d_{FR}(C, B_n)$ is even if $d_{FR}(C, B_n) > 1$ [19] and so $d_{FR}(C, B_n) < 6$ means $d_{FR}(C, B_n) \leq 4$.

From the above discussion we strongly conjecture that $d_{FR}(C, B_n) = 2(\frac{n}{3} - 1) < d_{BCH}$ and $\lfloor \frac{d_{FR}-1}{2} \rfloor = \lfloor \frac{d_{BCH}-1}{2} \rfloor - 1$ with a Type I ordered basis B_n.

7.4.3 $(n, n-1)$ Binary Cyclic Code (Parity Code)

Let C be a parity $(n, n-1)$ code with $h(x) = x^{n-1} + x^{n-2} + \cdots + x + 1$. From $\boldsymbol{b}_1 = (1, 1, \cdots, 1) \in F_2^n$, we have $\sigma(\boldsymbol{b}_1\boldsymbol{b}_j) = j$ for $j = 1, 2, \cdots, n$ and $N(2) = 2$.We have $d_{FR}(C, B_n) = 2$, since $B = \{\boldsymbol{b}_1\}$ and $B_n \backslash B = \{\boldsymbol{b}_2, \boldsymbol{b}_3, \cdots, \boldsymbol{b}_n\}$. Since the minimum distance of the parity code is 2, we have following proposition.

Proposition 7.4.3 *For the parity $(n, n-1)$ code C, we have $d_{FR}(C, B_n) = d_{BCH} = d = 2$.*

7.5 Conclusion

The relation between the choice of an ordered basis $B_n = \{\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_n\}$ of F_q^n and $d_{FR}(C, B_n)$ was discussed, where $d_{FR}(C, B_n)$ is the Feng-Rao designed minimum distance of an (n, k) cyclic code C, computed by using B_n.

The ordered basis B_n with Type I is defined such that the subset $B = \{\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_{n-k}\}$ of B_n satisfies $C = \mathrm{Span}\{B\}^{\perp}$ and B is a permutation of $\{\boldsymbol{b}, \delta(\boldsymbol{b}), \cdots, \delta^{n-k-1}(\boldsymbol{b})\}$, where $\boldsymbol{b} = (1, b_2, \cdots, b_{k+1}, 0, \cdots, 0)$ corresponds to the check polynomial $h(x) = x^k + b_2 x^{k-1} + \cdots + b_{k+1}$ and $\delta(\boldsymbol{b})$ is the right cyclic shift of $\boldsymbol{b}$.

It was shown that the choice of an ordered basis B_n with Type I is worst in many cases of nonbinary cyclic codes, since $d_{FR}(C, B_n) \leq 1$ if the check polynomial $h(x)$ has a coefficient neither equal to 0 nor 1.

Recently it is shown that d_{FR} of binary linear codes can not be an odd number greater than one [19]. This is due to the symmetric property of $\sigma(\boldsymbol{b}_i\boldsymbol{b}_j)$, i.e., $\sigma(\boldsymbol{b}_i\boldsymbol{b}_j) = \sigma(\boldsymbol{b}_j\boldsymbol{b}_i)$ in Miura's definition[8] which is not assumed in [7].

It was also shown that in case of binary (n, k) cyclic codes C with $k = 1, 2$, and $n-1$, there exists an ordered basis B_n with Type I such that $d_{FR}(C, B_n) = n - 1$, $d_{FR}(C, B_n) = 2(\frac{n}{3} - 1)$, and $d_{FR}(C, B_n) = 2$ for $k = 1, 2$, and $n-1$, respectively. Especially we have $d_{FR}(C) = n - 1$ in case of $k = 1$ and $d_{FR}(C) = 2$ in case of $k = n - 1$ for binary cyclic codes with any B_n.

For cyclic codes conventional decoding methods up to BCH bound, HT bound and Roos bound need the computation over the extension field, on the other hand the Feng-Rao decoding up to the Feng-Rao designed minimum distance needs only the computation over its base field. We showed that the Feng-Rao designed minimum distance is inferior only by one or equal to the BCH designed minimum distance in the ability of error correction for binary (n, k) cyclic codes with odd length n and $k = 1$, $k = 2$, and $k = n - 1$.

Finding an optimum ordered basis B_n for computing $d_{FR}(C)$ of any (n, k) cyclic code C is a desirable future work. Further investigation of the computational complexity of the Feng-Rao decoding of linear codes [8] and their modification for cyclic codes are very important future problems.

Chapter 8

Conclusions and Future Works

8.1 Conclusions

Some lower bounds for the minimum distance of cyclic codes are surveyed. For cyclic codes the BCH bound $d_{BCH}(C)$, HT bound $d_{HT}(C)$, Roos bound $d_R(C)$ and the Shift bound $d_S(C)$ for the minimum distance are well-known. It is known that $d_{BCH}(C) \leq d_{HT}(C) \leq d_R(C)$ and $d_{HT}(C) \leq d_S(C)$. But the relation between $d_R(C)$ and $d_S(C)$ has been not known. In Chapter 3 this relation is discussed with some numerical examples. For the binary cyclic codes with $n \leq 31$, about 25% is $d_R(C) < d_S(C)$, about 75% is $d_R(C) = d_S(C)$, and there is no case of $d_R(C) > d_S(C)$. Therefore we have $d_R(C) \leq d_S(C)$ for binary cyclic codes with $n \leq 31$. For the ternary cyclic codes with $n \leq 26$, about 32% is $d_R(C) < d_S(C)$, about 67% is $d_R(C) = d_S(C)$, and there are two cases with $d_R(C) > d_S(C)$. One of them was reported in [11] and another one found by the author.

Miura's definition of $d_{FR}(C, B_n)$ for an (n, k) linear code C over a finite field F_q depends on the choice of an ordered basis $B_n = \{\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_n\}$ of the vector space F_q^n. In Chapter 5 we proved that $d_{FR}(C, B_n)$ of binary linear codes can not take an odd number greater than or equal to 3 if we use Miura's definition [8] of $d_{FR}(C, B_n)$ for binary linear codes.

Recently Miura's definition of $d_{FR}(C, B_n)$ has been modified by Matsumoto [7] so that we can show the case where Lemma 5.2.2 does not hold. Therefore it is an interesting problem to examine whether Theorem 5.2.1 still holds or not if we use Matsumoto's definition of $d_{FR}(C, B_n)$ for binary linear codes. In Chapter 6 we gave some conjectures for Matsumoto's definition of $d_{FR}(C, B_n)$.

The ordered basis B_n with Type I is defined such that the subset $B = \{\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_{n-k}\}$ of B_n satisfies $C = \mathrm{Span}\{B\}^{\perp}$ and B is a permutation of $\{\boldsymbol{b}, \delta(\boldsymbol{b}), \cdots, \delta^{n-k-1}(\boldsymbol{b})\}$, where

$\boldsymbol{b} = (1, b_2, \cdots, b_{k+1}, 0, \cdots, 0)$ corresponds to the check polynomial $h(x) = x^k + b_2 x^{k-1} + \cdots + b_{k+1}$ and $\delta(\boldsymbol{b})$ is the right cyclic shift of $\boldsymbol{b}$. In Chapter 7, it was shown that the choice of an ordered basis B_n with Type I is worst in many cases of nonbinary cyclic codes, since $d_{FR}(C, B_n) \leq 1$ if the check polynomial $h(x)$ has a coefficient neither equal to 0 nor 1. It was also shown that in case of binary (n, k) cyclic codes C with $k = 1, 2$, and $n-1$, there exists an ordered basis B_n with Type I such that $d_{FR}(C, B_n) = n - 1$, $d_{FR}(C, B_n) = 2(\frac{n}{3} - 1)$, and $d_{FR}(C, B_n) = 2$ for $k = 1, 2$, and $n-1$, respectively. Especially we have $d_{FR}(C) = n - 1$ in case of $k = 1$ and $d_{FR}(C) = 2$ in case of $k = n - 1$ for binary cyclic codes with any ordered basis B_n.

For cyclic codes conventional decoding methods up to BCH designed minimum distance need the computation over the extension field, on the other hand the Feng-Rao decoding method up to the Feng-Rao designed minimum distance needs only the computation over its base field. We showed that the Feng-Rao designed minimum distance is inferior only by one or equal to the BCH designed minimum distance in the ability of error correction for binary (n, k) cyclic codes with odd length n and $k = 1$, $k = 2$, and $k = n - 1$.

8.2 Future Works

In Chapter 6 some properties and examples suggest us some conjectures which tell Matsumoto's generalization is not so effective compared with Miura's definition for binary linear codes. Giving proofs for these conjectures and investigating nonbinary linear codes are future works.

Chapter 7 discussed the case of binary (n, k) cyclic codes C with $k = 1, 2$, and $n - 1$. A derivation of $d_{FR}(C, B_n)$ for any k is very interesting work. We need that the upper bound of $d_{FR}(C, B_n)$ with $k = 2$ is proved as same as the lower bound.

Finding an optimum ordered basis B_n for computing $d_{FR}(C)$ of any (n, k) cyclic code C or any linear code C is a desirable future work. Further investigation of the computational complexity of the Feng-Rao decoding of linear codes and their modification for cyclic codes are very important future problems.

Bibliography

[1] R. C. Bose and D. K. Ray-Chaudhuri, "On a Class of Error Correcting Binary Group Codes," *Information and Control* vol. 3, pp. 68-79, 1960

[2] R. C. Bose and D. K. Ray-Chaudhuri, "Further Results on Error Correcting Binary Group Codes," *Information and Control* vol. 3, pp. 279-290, 1960

[3] G.L. Feng, T.R.N. Rao, "Decoding Algebraic Geometric Codes up to the Designed Minimum Distance," *IEEE Transaction on Information Theory*, vol. 39, pp. 36-47, Jan. 1993.

[4] G.L. Feng, K.K. Tzeng, "A New Procedure for Decoding Cyclic and BCH Codes up to Actual Minimum Distance," *IEEE Transaction on Information Theory*, vol. 40, No. 5, pp. 1364-1374, Sep. 1994.

[5] C.R.P. Hartmann and K.K. Tzeng, "Generalization of the BCH-bound," *Information and Control*, vol. 20, pp. 489-498, 1972.

[6] F.J. MacWiliams and N.J.A. Sloane, The Theory of Error-Correcting Codes, Amsterdam: North Holland 1977.

[7] R. Matsumoto and S. Miura, "On the Feng-Rao Bound for the L-construction of Algebraic Geometric Codes," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E83-A, No.5, pp. 923-926, May 2000.

[8] S. Miura, "Study on Error Correcting Codes Based on Algebraic Geometry," Ph. D. Dissertation, the University of Tokyo, 1997. (in Japanese)

[9] R. Pellikaan, "The Sift Bound for Cyclic, Reed-Muller and Geometric Goppa Codes," *Arithmetic, Geometry and Coding Theory 4*, pp. 155-174, 1996.

[10] C. Roos, "A New Lower Bound for the Minimum Distance of a Cyclic Code," *IEEE Transaction on Information Theory*, vol. IT-29, pp. 330-332, May 1983.

[11] M. van Eupen and J.H. van Lint, "On the Minimum Distance of Ternary Cyclic Codes," *IEEE Transaction on Information Theory*, vol. 39, NO. 2, March 1993.

[12] J.H. van Lint and R.M. Wilson, "On the Minimum Distance of Cyclic Codes," *IEEE Transaction on Information Theory*, vol. 32, NO. 1, January 1986.

[13] J. Zheng, T. Kaida, K. Imamura, "A Note on Feng-Rao Designed Minimum Distance for Cyclic Codes," *The Third International Conference on Information, Communications & Signal Processing (ICICS 2001)*, pp. 1-5 (1AI.2), Oct., 2001.

[14] J. Zheng, T. Kaida, K. Imamura, "The Feng-Rao Designed Minimum Distance of Binary Linear Codes Does not Have an Odd Number Except One," *The First International Workshop on Sequence Designed and Applications for CDMA Systems (IWSDA 2001)*, pp. 146-149, Sep., 2001.

[15] J. Zheng, T. Kaida, K. Imamura, "Is Matsumoto's Generalization of the Feng-Rao Designed Minimum Distance for Binary Linear Codes Effective?," *6th International Symposium on Digital Signal Processing for Communication Systems (DSPCS2002)*, pp. 84-87, Jan., 2002.

[16] J. Zheng, T. Kaida, K. Imamura, 'A Note on Lower Bound for the Minimum Distance of a Cyclic Codes," *The 21st Symposium on Information Theory and its Applications (SITA98)*, pp. 21-24, Dec., 1998.

[17] J. Zheng, T. Kaida, K. Imamura, 'A Note on Feng-Rao Designed Minimum Distance of Binary and Cyclic Codes," *The 22nd Symposium on Information Theory and its Applications (SITA99)*, pp. 511-514, Nov., 1999.

[18] J. Zheng, T. Kaida, K. Imamura, 'Further Note on the Feng-Rao Designed Minimum Distance for Cyclic Codes," *The 23rd Symposium on Information Theory and its Applications (SITA2000)*, pp. 659-662, Oct., 2000.

[19] J. Zheng, T. Kaida, K. Imamura, "Some Conjectures for Matsumoto's Generalization of the Feng-Rao Designed Minimum Distance of Binary Linear Codes," *The 24th*

Symposium on Information Theory and its Applications (SITA2001), pp. 191-193, Dec., 2001.