CONSTRUCTION OF QUASI-CYCLIC SELF-DUAL CODES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF APPLIED MATHEMATICS
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

PINAR ÇOMAK

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
CRYPTOGRAPHY

SEPTEMBER 2013

Approval of the thesis:

CONSTRUCTION OF QUASI-CYCLIC SELF-DUAL CODES

submitted by **PINAR ÇOMAK** in partial fulfillment of the requirements for the degree of **Master of Science in Department of Cryptography, Middle East Technical University** by,

Bülent Karasözen _____
Director, Graduate School of **Applied Mathematics**

Ferruh Özbudak _____
Head of Department, **Cryptography**

Ferruh Özbudak _____
Supervisor, **Department of Cryptography, METU**

Jon Lark Kim _____
Co-supervisor, **Department of Mathematics, Sogang University**

Examining Committee Members:

Asst. Prof. Dr. Ömer Küçüksakallı (Head of the examining com.) _____
Department of Mathematics, METU

Prof. Dr. Ferruh Özbudak(Supervisor) _____
Department of Mathematics, METU

Asst. Prof. Dr. Sedat Akleylek _____
Department of Computer Engineering, Ondokuz Mayıs University

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: PINAR ÇOMAK

Signature :

ABSTRACT

CONSTRUCTION OF QUASI-CYCLIC SELF-DUAL CODES

Çomak, Pınar
M.S., Department of Cryptography
Supervisor : Ferruh Özbudak
Co-Supervisor : Jon Lark Kim

September 2013, 58 pages

Quasi-cyclic and self-dual codes are interesting classes of linear codes. Quasi-cyclic codes are linear codes which takes maximum possible value of minimum distance among the codes with the same length and same dimension. Another class of interesting linear codes is the self-dual codes. Self-dual codes have close connections with group theory, lattice theory and design theory. There has been an active research on the classification of self-dual codes over finite fields and over rings. We study on construction of quasi-cyclic self-dual codes, especially binary cubic ones. With a new algebraic approach, binary quasi-cyclic codes of length 3ℓ over a field are defined by the linear codes of length ℓ over the ring $\mathbb{F}_2 \times \mathbb{F}_4$. In this thesis, we improve the result for the cubic self-dual binary codes, by finding two new self-dual codes with the algebraic approach.

Keywords: self-dual codes, quasi-cyclic codes, Chinese Remainder Theorem, cubic

construction

ÖZ

QUASI-CYCLIC SELF-DUAL KODLARIN YAPILANDIRILMASI

Çomak, Pınar
Yüksek Lisans, Kriptografi Bölümü
Tez Yöneticisi : Ferruh Özbudak
Ortak Tez Yöneticisi : Jon Lark Kim

Eylül 2013, 58 sayfa

Yarı-devirli ve self-dual kodlar, lineer kodların ilgi çekici sınıflarıdır. Aynı uzunluk ve boyuta sahip lineer kodlar arasında, olası en yüksek minimum uzaklığa sahip kodlar yarı-devirli kodlardır. Lineer kodların diğer bir ilgi çekici sınıfı ise self-dual kodlardır. Self-dual kodların, grup teori, kafes teorisi ve dizayn teori ile yakın bir bağlantısı vardır. Sonlu cisimler ile halkalar üzerindeki self-dual kodların üzerine çalışan aktif araştırma grupları bulunmaktadır. Yarı-devirli self-dual kodlar arasında özellikle ikilik ve kübik olanlarının yapılandırılması üzerine çalıştık. Yeni cebirsel yaklaşım ile, sonlu cisimler üzerinde tanımlanan 3ℓ uzunluğundaki ikilik quasi-cyclic kodlar, $\mathbb{F}_2 \times \mathbb{F}_4$ halkası üzerinde tanımlanan ℓ uzunluğunda bir lineer kod olarak tanımlanmıştır. Bu tezde, cebirsel yaklaşım ile iki yeni self-dual kod bularak, ikilik kübik self-dual kodların sonucu geliştirilmiştir.

Anahtar Kelimeler: self-dual kodlar, yarı-devirli kodlar, Çin Kalan Teoremi,

kübik yapılandırılma

To mum and Azra

ACKNOWLEDGMENTS

I want to thank my supervisor Prof. Dr. Ferruh Özbudak for his guidance and helpful suggestions during the preparation of this thesis.

I do specially appreciate the help of my co-supervisor Prof. Dr. Jon-Lark Kim who played the main role during this research. I would like to thank my friends Nari and Sung in Seoul for providing me a pleasant time during my visit. I would also like to take this opportunity to thank for the hospitality of Department of Mathematics, Sogang University.

I am also thankful to my thesis defence committee members for their useful comments and discussions.

I deeply appreciate everybody who helped me directly or indirectly to accomplish this thesis.

Also, I want to thank my friends for their helps in LaTeX.

This work in this thesis is partially supported by Turkish Scientific and Technical Research Council (TÜBİTAK) under project no 112T011.

Finally, I appreciate the financial support from Council of Higher Education (YÖK) for funding me during my master study in Seoul, South Korea.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Coding theory has become a fast growth mathematical theory which has a wide area of applications especially in communication system and information theory. Among all types of block codes, linear codes are the most studied. Because of their algebraic structure they are easier to define, encode and decode when compared with nonlinear codes. The best known error correcting codes are Hamming, Golay, Bose-Chaudhuri-Hocquenghem and Reed-Solomon, which are all subclasses of cyclic codes, because of their rich mathematical structure. The Reed-Muller codes are also important because they are Majority Logic Decodable, a scheme which is fast and simple.

Linear codes which are quasi-cyclic and self-dual simultaneously are an interesting class of codes. Quasi-cyclic codes have been discovered with minimum distance exceeding that previously known for any linear code of the same length and dimension, or, indeed, taking the maximum possible value. From this point of view, this family of codes is very interesting. Moreover, quasi-cyclic codes were studied for their application in some cryptosystems, McEliece, Niederreiter's. Indeed they allow an interesting key reduction compared to Goppa codes.

One class of codes which has many well-known best error correcting codes is linear self-dual code, one is the Reed-Muller code that was used in the spacecraft Mariner 9 to send the gray image of Mars on 19 January 1972. Self-dual codes have a rich mathematical theory and strong connections with other areas of combinatorics, group theory and lattice. Self-dual codes are important because many of the best known codes are of this type.

Quasi-cyclic codes can be considered as modules over the group algebra of the cyclic group, from the module theory. Quasi-cyclic codes are remarkably a generalization of cyclic codes. The authors of [10] introduced the algebraic approach to quasi-cyclic codes. In their paper, they consider linear codes over a ring $\mathcal{R}$, and they use the one-to-one correspondence ϕ between (self-dual) quasi-cyclic codes over a field $\mathbb{F}_q$ and (self-dual) linear codes over an auxiliary ring $\mathcal{R} := \mathcal{R}(\mathbb{F}_q, m) = \mathbb{F}_q[Y]/(Y^m - 1)$ where m is coprime with the characteristic of $\mathbb{F}$. They show that all binary extended quadratic residue codes of length 3ℓ are attainable by the cubing construction. When $q = 2$, $m = 3$ and the code is self-dual, the code is called cubic self-dual binary codes [3].

In this thesis, this type of codes is studied. The outline is as follows: In Chapter 2, linear codes, containing especially cyclic codes and their properties, containing the self-duality, are introduced. In Chapter 3, quasi-cyclic codes and the one-to-one correspondence between ℓ-quasi-cyclic codes over a field and linear codes over a ring are mentioned. In Chapter 4, the ring decomposition and some theorems about existence of self-dual codes are given. Chapter 5 contains the building-up construction and some other constructions of quasi-cyclic self-dual codes depending on the values m. Chapter 6 presents the $(a+x \mid b+x \mid a+b+x)$ construction among the other ones. Our contribution of this thesis is to find new weight enumerators of cubic self-dual $[54, 27, 10]$ codes with $(a + x \mid b + x \mid a + b + x)$ construction. In this chapter in this construction, we choose some other possible and suitable linear codes for $\mathcal{C}_2$. As a result, we found two new codes with different weight distributions other than previously found. In last chapter, the work is concluded with the algorithm we used in MAGMA.

CHAPTER 2

CODES

Quasi-cyclic code is a type of linear codes and it is a generalization of cyclic codes. The properties of quasi-cyclic codes are inherited from the ones of the linear codes. Therefore, it is more convenient to give the general properties.

2.1 Linear Codes

Most practical error-correcting codes in use are linear codes. Any linear combination of codewords is also a codeword. Its advantages over arbitrary codes are as follows [1]:

1. It is much easier to evaluate the distance $d(\mathcal{C})$. We will see below that $d(\mathcal{C}) = w(\mathcal{C})$.
2. To specify a non-linear code, we may have to list all the codewords. We can easily specify a linear $[n, k]$ code by giving a basis of k codewords (from a generator matrix).
3. Encoding is fast and requires little storage.
4. To determine which errors are correctable and detectable is much easier.
5. The probability of correct decoding is much easier to calculate.
6. There are many nice decoding techniques for linear codes.

Definition 2.1.1 *A q-ary linear code $\mathcal{C}$ is a linear subspace of $\mathbb{F}_q^n$. If $\mathcal{C}$ has dimension k then $\mathcal{C}$ is called an $[n, k]$ linear code.*

The number of codewords in the form $[n, k]$ over $\mathbb{F}_q$ equals q^k. The code rate is $R = \frac{k}{n}$.
Because of linearity, for $c, c' \in \mathcal{C}$ and for $a \in \mathbb{F}_q$, the followings must be satisfied:

1. $c + c' \in \mathcal{C}$
2. $ac \in \mathcal{C}$.

The all-zero vector $\mathbf{0}$ automatically belongs to a linear code.

2.1.1 Preliminaries

2.1.1.1 Minimum Hamming Distance and Weight

The minimum Hamming distance $d(\mathcal{C})$ is the minimum number of distinct coordinates between any pair of distinct codewords. *The weight* $w(c)$ of a codeword c in $\mathbb{F}_q^n$ is defined to be the number of non-zero entries of c. One of the most useful properties of a linear code is that its *minimum distance* is equal to the smallest of the weights of the nonzero codewords, i.e.

Theorem 2.1.2 [6] *Let $\mathcal{C}$ be a linear code and let $w(\mathcal{C})$ be the smallest of the weights of the nonzero codewords of $\mathcal{C}$. Then*

$$d(\mathcal{C}) = w(\mathcal{C}).$$

in other words,

$$d = \min_{\forall c \neq c'} dist(c, c') = \min_{\forall c \neq 0} wt(c)$$

Proof. There exist codewords c and c' of $\mathcal{C}$ such that $d(\mathcal{C}) = d(c, c')$. Then

$$d(\mathcal{C}) = w(c - c') \geq w(\mathcal{C}),$$

since $c - c'$ is a codeword of the linear code $\mathcal{C}$.
On the other hand, for some codeword $c \in \mathcal{C}$,

$$w(\mathcal{C}) = w(c) = d(c, \mathbf{0}) \geq d(\mathcal{C}),$$

since $\mathbf{0}$ belongs to the linear code $\mathcal{C}$.
Hence $d(\mathcal{C}) \geq w(\mathcal{C})$ and $w(\mathcal{C}) \geq d(\mathcal{C})$ which gives $d(\mathcal{C}) = w(\mathcal{C})$. ■

It is important because if a code $\mathcal{C}$ has minimal Hamming weight d, then $\mathcal{C}$ can correct
$\lfloor (d-1)/2 \rfloor$ errors. In other words, to find a linear code that can correct t errors, one must find a linear code with minimum weight satisfying $d \geq 2t + 1$, sphere packing bound [2]. Another parameter that is often optimized is the information rate of the code. The information rate of an $[n, k]$ code is defined to be k/n. It states that for every k bits of useful information, the code generates a total of n bits of data, of which $n - k$ are redundant. When the ratio is closed to 1, it is more efficient to encode information using the code. Efficiency refers to the length of messages that are used to encode the information. *The extremal code* is used for the codes which has the possible largest minimum weight of a given length code. *The optimal code* is one having the highest minimal distance of any self-dual code of that length. An extremal code is automatically optimal.

2.1.1.2 Permutation Equivalence of Linear Codes

Two codes are said to be *equivalent up to permutation* if they differ only in the order of their coordinates. We will use only equivalent during the work since it means equivalent up to permutation. The rowspace of generator matrix is equal to the code. We can permute the rows of matrix since it does not change the rowspace. However, the column permutation changes the rowspace, so the code.

If we apply column permutation, we get an equivalent code, since the length, dimension and weight structure are unchanged.

Theorem 2.1.3 [6] *Two $k \times n$ matrices generate equivalent linear $[n, k]$ codes over $\mathbb{F}_q$ if one matrix can be obtained from the other by a sequence of a sequence of operations of the following types:*
(R1) Permutation of the rows.
(R2) Multiplication of a row by a nonzero scalar.
(R3) Addition of a scalar multiple of one row to another.
(C1) Permutation of the columns.
(C2) Multiplication of any column by a nonzero scalar.

Proof. The row operations (R1), (R2) and (R3) preserve the linear independence of the rows of a generator matrix and simply replace one basis by another of the same code. The column operations (C1) and (C2) convert a generator matrix to one for an equivalent code. ■

2.1.1.3 Automorphism Groups

The subset of transformations that preserve the code forms *Automorphism Group $Aut(\mathcal{C})$ (or Permutation Group $Perm(\mathcal{C})$)* . For binary codes, $Aut(\mathcal{C})$ is the subgroup of the permutation group S_n where n is the length of the codewords of $\mathcal{C}$, [18].
Let G denote the group of any transformation. The order of G is $n!$ for binary codes.
The number of the codes that are equivalent to a given code $\mathcal{C}$ is $|G|/|Aut(\mathcal{C})|$.

2.1.1.4 Weight Enumerators

The Hamming Weight of a vector $u = (u_1, \ldots, u_n) \in \mathbb{F}^n$, denoted by $wt(u)$, is the number of components of u, which are nonzero.

We denote the number of vectors of the code $\mathcal{C}$ having Hamming Weight equal to i by A_i. The *Hamming Weight Enumerator* of the code $\mathcal{C}$ is defined to be

$$W_{\mathcal{C}}(x, y) = \sum_{c \in \mathcal{C}} x^{n-wt(c)} y^{wt(c)} = \sum_{i=0}^{n} A_i x^{n-i} y^i.$$

Often, the x is replaced by 1 and we write it as a polynomial in the single variable y [18].

2.1.2 Some Examples of Self-dual Codes with their Weight Enumerator

We denote a linear code with length n, dimension k and minimum distance d over a field $\mathbb{F}_q$ by $[n, k, d]_q$. We omit q for binary codes (i.e. $q = 2$).
The parentheses in a vector mean that all permutations indicated by the parentheses are applied to that vector. For example, 1(1101000) stands for the seven vectors (11101000), (11010001), (10100011), (11000110), (10001101), (10011010), (10110100).
Here is some self-dual codes:

- The $[2, 1, 2]$ repetition code $i_2 = \{00, 11\}$ is a binary self-dual code with weight enumerator

$$W_{i_2}(x, y) = x^2 + y^2.$$

- The $[8, 4, 4]$ Hamming code e_8 generated by 1(1101000) has weight enumerator

$$W_{e_8}(x, y) = x^8 + 14x^4y^4 + y^8.$$

- The $[24, 12, 8]$ binary Golay code g_{24} generated by $1(10101110001100000000000)$ has weight enumerator

$$W_{g_{24}}(x, y) = x^{24} + 759x^{16}y^8 + 2576x^{12}y^{12} + 759x^8y^{16} + y^{24}.$$

2.1.3 Inner products

In order to define dual codes, we must define inner products.
We denote inner product by $(\ ,\ )$, [18]. The inner product of the codewords $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n)$ in $\mathbb{F}^n$ is defined by

$$(x, y) = \sum_{i=1}^{n} (x_i, y_i).$$

and it satisfies the following conditions:

$$\begin{aligned}
(x + y, z) &= (x, z) + (y, z), \\
(x, y + z) &= (x, y) + (x, z), \\
(ax, y) &= a(x, y) = (x, ay), \\
&\text{if } (x, y) = 0 \text{ for all } x \text{ then } y = 0, \\
&\text{if } (x, y) = 0 \text{ for all } y \text{ then } x = 0.
\end{aligned}$$

To define the dual code a linear code we must define *conjugacy* operation (which may be identity), denoted by a bar. It satisfies

$$\overline{\overline{x}} = x, \quad \overline{x + y} = \overline{x} + \overline{y}, \quad \overline{xy} = \overline{x}\ \overline{y}.$$

Then the inner product must satisfy

$$(x, y) = \overline{(y, x)}, \quad (ax, y) = (x, \overline{a}y).$$

Inner products for linear codes over $\mathbb{F}_q^{\ell m}$ and over $\mathcal{R}^\ell$ are as follows, respectively, [18].

2.1.3.1 Euclidean inner product

Standard (Euclidean) inner product is defined on $\mathbb{F}_q^{\ell m}$ as

$$(a, b) = a \cdot b = \sum_{i=0}^{m-1} \sum_{j=0}^{\ell-1} a_{ij} b_{ij}$$

for

$$a = (a_{0,0}, a_{0,1}, \ldots, a_{0,\ell-1}, a_{1,0}, \ldots, a_{1,\ell-1}, \ldots, a_{m-1,0}, \ldots, a_{m-1,\ell-1})$$

and

$$b = (b_{0,0}, b_{0,1}, \ldots, b_{0,\ell-1}, b_{1,0}, \ldots, b_{1,\ell-1}, \ldots, b_{m-1,0}, \ldots, b_{m-1,\ell-1})$$

2.1.3.2 Hermitian inner product

Hermitian inner product is defined on $\mathcal{R}^\ell$ as

$$(x, y) = \langle x, y \rangle = \sum_{j=0}^{\ell-1} x_j \overline{y_j}$$

for

$$x = (x_0, x_1, \ldots, x_{\ell-1}) \quad \textit{and} \quad y = (y_0, y_1, \ldots, y_{\ell-1})$$

Here the conjugation map $^-$ on $\mathcal{R}$ is a map sending Y to $Y^{-1} = Y^{m-1}$ and it acts as the identity map on $\mathbb{F}_q$.

2.1.4 Dual Codes

We can define the *dual* of a code $\mathcal{C}$ to be

$$\mathcal{C}^\perp = \{u \in \mathbb{F}^n : (u, v) = 0 \text{ for all } v \in \mathcal{C}\}.$$

The dual of a binary linear code is again a binary linear code. The *dual* code $\mathcal{C}^\perp$ of the code $\mathcal{C}$ is understood with respect to standard (Euclidean) inner product.

Theorem 2.1.4 [6] *Suppose $\mathcal{C}$ is an $[n, k]$ code over $\mathbb{F}_q$. Then the dual code $\mathcal{C}^\perp$ of $\mathcal{C}$ is a linear $[n, n-k]$ code.*

Proof. First, we show that $\mathcal{C}^{\perp}$ is a linear code.
Suppose $u, v \in \mathcal{C}^{\perp}$ and $a, b \in \mathbb{F}_q$. Then for all w in $\mathcal{C}$,

$$\begin{aligned}(au + bv) \cdot w &= a(u \cdot w) + b(v \cdot w) \\ &= a0 + b0 = 0.\end{aligned}$$

Hence, $au + bv \in \mathcal{C}^{\perp}$, so $\mathcal{C}^{\perp}$ is linear.
Now, we show that dimension of $\mathcal{C}^{\perp}$ is $n-k$. Let $G = [g_{ij}]$ be a generator matrix of $\mathcal{C}$. The elements of $\mathcal{C}^{\perp}$ are all orthogonal to the rows of that matrix such that

$$\sum_{j=1}^{n} g_{ij} v_j = 0 \quad \text{for} \quad i = 1, 2, \ldots, k$$

for the vectors $v = (v_1 v_2 \ldots v_n)$ in $\mathcal{C}^{\perp}$. This is a system of k independent homogeneous equations in n unknowns. By linear algebra, the dimension of $\mathcal{C}^{\perp}$ is $n-k$. ■

Example 2.1.5 *It is easy to check that*

$$\textit{if} \quad \mathcal{C} = \begin{cases} 0000 \\ 1100 \\ 0011 \\ 1111 \end{cases}, \quad \textit{then} \quad \mathcal{C}^{\perp} = \mathcal{C}.$$

$$\textit{if} \quad \mathcal{C} = \begin{cases} 000 \\ 110 \\ 011 \\ 101 \end{cases}, \quad \textit{then} \quad \mathcal{C}^{\perp} = \begin{cases} 000 \\ 111 \end{cases}.$$

Theorem 2.1.6 *For any $[n, k]$ code $\mathcal{C}$, $(\mathcal{C}^{\perp})^{\perp} = \mathcal{C}$.*

Proof. Clearly, $\mathcal{C} \subseteq (\mathcal{C}^{\perp})^{\perp}$ since every vector in $\mathcal{C}$ is orthogonal to every vector in $\mathcal{C}^{\perp}$. But $dim((\mathcal{C}^{\perp})^{\perp}) = n - (n-k) = k = dim(\mathcal{C})$, and so $\mathcal{C} = (\mathcal{C}^{\perp})^{\perp}$. ■

We will show that a parity-check matrix H of a code $\mathcal{C}$ is a generator matrix of $\mathcal{C}^{\perp}$.

2.1.4.1 Self-dual Codes

A code $\mathcal{C}$ is said to be *self-dual* if $\mathcal{C} = \mathcal{C}^{\perp}$. $\mathcal{C}$ is *self-orthogonal* if $\mathcal{C} \subset \mathcal{C}^{\perp}$.
If $\mathcal{C}$ is self-dual then

$$|\mathcal{C}| = |\mathbb{F}|^{n/2},$$

and if $|\mathbb{F}|$ is not a square then n must be even. In particular, if $\mathcal{C}$ is linear over a field, then n is even and $\mathcal{C}$ is a subspace of dimension $n/2$. Since each codeword in $\mathcal{C}$ is orthogonal to the all codewords in $\mathcal{C}$, the weight of all codewords must be even. The self-dual codes in which there is at least one codeword with weight not divisible by 4 are called *Type I* or *singly-even* self-dual binary codes. The self-dual codes in which the weight of each codeword is divisible by 4 are called *Type II* or *doubly-even* self-dual binary codes. For self-dual codes the transmission rate k/n is always $1/2$.

2.1.4.2 Generator Matrix

As $\mathcal{C}$ is a subspace, there exists a basis $c_1, c_2, \ldots, c_k$ where k is the dimension of the subspace. The linear combinations of these basis vectors give all the codewords. These vectors in matrix form can be written as the columns of a $k \times n$ matrix. Such a matrix is called a *generator matrix*.

Definition 2.1.7 *Let $\mathcal{C} \subseteq \mathbb{F}_q^n$ be a linear code with dimension k. We say that a matrix $G \in \mathbb{F}_q^{k \times n}$ is a generator matrix for $\mathcal{C}$ if its k rows span $\mathcal{C}$.*

Note that by choosing different basis for the code as a vector space, we get different generator matrices. So, the generator matrix is not unique.

With the generator matrix G, we encode a message $x \in \mathbb{F}_q^k$ as the codeword $xG \in \mathcal{C} \subseteq \mathbb{F}_q^n$. We take a matrix G whose rows are the codewords of any basis of $\mathcal{C}$, say $c_1, c_2, \ldots, c_k$, and define for each message m the corresponding codeword c by $c = mG$.
Thus a linear code has an encoding map $E : \mathbb{F}_q^k \to \mathbb{F}_q^n$ which is a linear transformation $x \to xG$.

Example 2.1.8 *Let $c_1 = 12043$, $c_2 = 23104$ and $c_3 = 40211$ be a basis for a 3-dimensional code over $\mathbb{F}_5$. So the message $m = 123$ is encoded as*

$$mG = \begin{pmatrix} 1 & 2 & 3 \end{pmatrix} \begin{pmatrix} 1 & 2 & 0 & 4 & 3 \\ 2 & 3 & 1 & 0 & 4 \\ 4 & 0 & 2 & 1 & 1 \end{pmatrix} = 23324$$

So $\mathcal{C}$ is just the span of the rows of G, $Im(G)$.

2.1.4.3 Standard Form of Generator Matrix

Particularly convenient linear codes are those which have a $k \times n$ generator matrix in which the first k columns forms the identity matrix I_k, because in this case the message coincides with the first k symbols of its codeword. This is a time saving feature because the vast majority of words are received with no errors so for these words all the receiver needs to do with the received codeword is read off its first k symbols.

Definition 2.1.9 *The form of the generator matrix of a code $\mathcal{C}$*

$$G = [I_k \mid A],$$

where I_k is the $k \times k$ identity matrix, and A is a $k \times (n-k)$ matrix, called standard or systematic form.

By performing operations of types (R1), (R2), (R3), (C1) and (C2), G can be transformed to the standard form.

Example 2.1.10 *Let C be the $[7,4]$ code of $\mathbb{F}_2^7$ generated by the rows of G (in standard form):*

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

We get the 16 codewords by multiplying G on the left by the 16 different binary row vectors of length 4. The list of all the codewords is:
$(0000000), (1101000), (0110100), (0011010), (0001101), (1000110), (0100011),$
$(1010001), (1111111), (0010111), (1001011), (1100101), (1110010), (0111001),$
$(1011100), (0101110).$
Notice that there are 7 codewords of weight 3, 7 of weight 4, 1 of weight 7 and 1 of weight 0. Hence, the minimum distance of this code is 3 because of linearity of the code, and so it is a 1-error correcting code, because $d \geq 2t+1$.
This $[7,4,3]$ code is called the $[7,4]$ Hamming Code.

2.1.4.4 Parity-check Matrix

When generator matrix in standard form is used, the first k symbols of the codeword are just the message symbols, and the last $n-k$ symbols are redundant check symbols. Note that, by permuting coordinates if needed, every linear code can have a generator matrix in standard form.

Example 2.1.11 *For $c_1 = 12043$, $c_2 = 23104$ and $c_3 = 40211$ being a basis for a 3-dimensional code over $\mathbb{F}_5$, the standard form of the generator matrix is*

$$G = \begin{pmatrix} 1 & 2 & 0 & 4 & 3 \\ 2 & 3 & 1 & 0 & 4 \\ 4 & 0 & 2 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 1 & 4 \\ 0 & 1 & 0 & 4 & 2 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

and the message $m = 123$ is encoded as

$$mG = \begin{pmatrix} 1 & 2 & 3 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 1 & 4 \\ 0 & 1 & 0 & 4 & 2 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix} = 12323$$

123 (the first 3 symbols of the codeword) are the message symbols, and the last 2 symbols 23 are redundant check symbols.

For $[7, 4]$ Hamming code, the messages are all sixteen 4-bit binary words which are encoded as 7-bit codewords. The three additional bits are fixed by requiring that the total number of ones in each of the three sets is even. Using modulo 2 addition and denoting a codeword by $c_1c_2 \dots c_7$ these conditions become:

$$c_1 + c_3 + c_4 + c_5 = 0$$

$$c_1 + c_2 + c_4 + c_6 = 0$$

$$c_1 + c_2 + c_3 + c_7 = 0$$

So this Hamming code $\mathcal{C}$ can be specified as the set of all 7-bit strings which satisfy these equations. The equations can be written in matrix form as

$$cH^\intercal = \mathbf{0}$$

where c is the codeword $(c_1c_2 \dots c_7)$, regarded as a row vector, H is the matrix

$$\begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$\mathbf{0}$ is the zero column vector

$$\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

and $H^\intercal$ is the transpose of H. Notice that the left hand sides of the three equations are just the dot products of c with the rows of H, so another description of $\mathcal{C}$ is that it is just $S^\perp$ where S is $1011100, 1101010, 1110001$.

Definition 2.1.12 [1] *H is called a parity check matrix for a linear code $\mathcal{C}$ if*
(i) *its rows are independent,*

(ii) *$\mathcal{C}$ is the set of all words satisfying $cH^\intercal = \boldsymbol{0}$. That is, $\mathcal{C}$ is the null space of H. We can write the code $\mathcal{C}$ as*

$$\mathcal{C} = \{x \in \mathbb{F}_q^n \mid xH^\intercal = \boldsymbol{0}\}.$$

Theorem 2.1.13 [1] *H is a parity check matrix for the $[n, k]$ linear code $\mathcal{C}$ if and only if it is a generator matrix for $\mathcal{C}^\perp$.*

Proof.
($\Rightarrow$) H is a parity check matrix for $\mathcal{C}$
$\Rightarrow \mathcal{C} = null(H)$
$\Rightarrow k = n - dim(Im(H))$
$\Rightarrow dim(Im(H)) = n - k$
$\Rightarrow dim(Im(H)) = dim(\mathcal{C}^\perp)$
But $Im(H) \subseteq \mathcal{C}^\perp$, so $Im(H) = \mathcal{C}^\perp$.
That is, H is a generator matrix of $\mathcal{C}^\perp$.

($\Leftarrow$) H is a generator matrix for $\mathcal{C}^\perp$.
$\Rightarrow$ rows of H are independent and $\mathcal{C}^\perp = Im(H)$
$\Rightarrow c \cdot u = 0$ for all $c \in \mathcal{C}$where u is any linear combination of H.
$\Rightarrow \mathcal{C} \subseteq Null(H)$
$\Rightarrow \mathcal{C} = Null(H)$ since
That is, H is a parity check matrix for $\mathcal{C}$. ■

It follows that from this theorem every linear code $\mathcal{C}$ has a parity check matrix as any generator matrix of $\mathcal{C}^\perp$. In other words, the code generated by H is called the dual code of $\mathcal{C}$, $\mathcal{C}^\perp$.
Thus H is an $(n - k) \times n$ matrix satisfying $GH^\intercal = \mathbf{0}$, where $H^\intercal$ denotes the transpose of H and $\mathbf{0}$ is an all-zero matrix.

2.1.4.5 Standard Form of Parity-check Matrix

Theorem 2.1.14 [6] *If $G = [I_k \mid A]$ is the generator matrix in standard form for the $[n, k]$ code $\mathcal{C}$, then $H = [-A^\intercal \mid I_{n-k}]$ is the parity check matrix for $\mathcal{C}$.*

Proof. By previous theorem, we know H is a generator matrix for $\mathcal{C}^\perp$.
Now, $GH^\intercal = [I_k \mid A]\,[-A^\intercal \mid I_{n-k}] = I_k(-A) + (A)I_{n-k} = \mathbf{0}$ which implies the rows of H are orthogonal to the rows of G, therefore $Im(H)$ = row space of H is contained in $\mathcal{C}^\perp$ ■

Definition 2.1.15 [6] *The form of the parity-check matrix of a code $\mathcal{C}$*

$$H = [B \mid I_{n-k}]$$

where B is the $(n-k \times k)$ matrix $-A^\intercal$ and I_{n-k} is the $(n-k) \times (n-k)$ identity matrix, is called standard form.

Many codes are most easily defined by specifying a parity-check matrix, or a set of parity-check equations. If a code is given by a parity-check matrix H which is not in standard form, then H can be reduced to standard form in the same way as for a generator matrix

Example 2.1.16 *For $[7, 4]$ Hamming code, a generator matrix G is*

$$G = [I_k \mid A] = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

and the parity-check matrix H is

$$H = [-A^\intercal \mid I_{n-k}] = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

where the matrix A is $\begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$. It is easily calculated that $GH^\intercal = \boldsymbol{0}$.

2.1.5 Cyclic codes

As one of the linear error-correcting codes, cyclic codes have convenient algebraic structures for efficient error detection and correction. Cyclic codes have some additional structural constraint on the codes. They are based on Galois fields and they are very useful for error controls because of their structural properties. The encoding and decoding algorithms for cyclic codes are computationally efficient. Cyclic codes are linear codes for which the automorphism group contains the cyclic group of order n, where n is the length of the word.

Definition 2.1.17 [6] *An $[n, k]$ linear code $\mathcal{C}$ is said to be cyclic if for every codeword $c = (c_0, c_1, \ldots, c_{n-1}) \in \mathcal{C}$, then there is the corresponding codeword $c' = (c_{n-1}, c_0, \ldots, c_{n-2})$ in $\mathcal{C}$ where c' is a cyclic shift of c.*

It is more convenient to represent the codewords as polynomials. The codeword

$$c = (c_0, c_1, \ldots, c_{n-1})$$

is represented by the polynomial

$$c(x) = c_0 + c_1 x + c_2 x^2 + \cdots + c_{n-1} x^{n-1}.$$

$\mathbb{F}_q[x]$ is the set of polynomials in x with coefficients in $\mathbb{F}_q$. Let $f(x)$ be a fixed polynomial in $\mathbb{F}_q[x]$. We denote by $\mathbb{F}_q[x]/f(x)$ the set of polynomials in $\mathbb{F}_q[x]$ of degree less than the degree of $f(x)$, with addition and multiplication in modulo $f(x)$.
With polynomial representation, a cyclic shift can be represented as follows:

$$xc(x) = c_0 x + c_1 x^2 + c_2 x^3 + \cdots + c_{n-1} x^n$$

in *mod* $(x^n - 1)$ is

$$xc(x) \quad mod\ (x^n - 1) = c_{n-1} + c_0 x + c_1 x^2 + c_2 x^3 + \cdots + c_{n-2} x^{n-1}$$

So multiplication by x in the ring $\mathbb{F}_q[x]/(x^n - 1)$ corresponds to a cyclic shift. In the same way, any power of x times a codeword gives a codeword, so that, for

example,

$$\begin{aligned}(c_0, c_1, \ldots, c_{n-1}) &\leftrightarrow c(x) = x^n c(x)\\ (c_{n-1}, c_0, \ldots, c_{n-2}) &\leftrightarrow xc(x)\\ (c_{n-2}, c_{n-1}, c_0, \ldots, c_{n-3}) &\leftrightarrow x^2 c(x)\\ &\vdots\\ (c_1, c_2, \ldots, c_0) &\leftrightarrow x^{n-1} c(x)\end{aligned}$$

where the arithmetic is done in the ring $\mathbb{F}_q[x]/(x^n - 1)$. So, it is easy to see that if we take a polynomial $a(x) \in \mathbb{F}_q[x]/(x^n - 1)$ of the form

$$a(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{n-1} x^{n-1}$$

then

$$c(x)a(x)$$

is a linear combination of cyclic shifts of $c(x)$ also is a codeword of $\mathcal{C}$. Hence, a cyclic code is an ideal in $\mathbb{F}_q[x]/(x^n - 1)$.

Theorem 2.1.18 [1] *A set I of polynomials in the ring $\mathcal{R} = F[x]/(x^n - 1)$ represents a cyclic code $\mathcal{C}$ if and only if I is an ideal of $\mathcal{R}$.*

Proof.

($\Leftarrow$) Let I be an ideal and let c_1, c_2 be members of I, and λ any constant member of $\mathcal{R}$. Then $c_1 + c_2 \in I$ by subring property of I, and $\lambda c_1 \in I$ by the ideal property. In terms of code $\mathcal{C}$, it is proved that $\mathcal{C}$ is closed under addition and scalar multiplication, i.e. $\mathcal{C}$ is linear. In order to show that it is cyclic, we can use the ideal property as $\forall x \in \mathcal{R}$, $xc(x) \in I$ also. This means that $\mathcal{C}$ is closed under the cyclic shift operation. Hence $\mathcal{C}$ is a cyclic code.

($\Rightarrow$) Conversely, let $\mathcal{C}$ be cyclic. We have to show that $I \in \mathcal{R}$ has all these properties:

1. $r + s \in I$, $r \times s \in I$,
2. $r + s = s + r$,
3. $(r + s) + t = r + (s + t)$, $(r \times s) \times t = r \times (s \times t)$,

4. $r \times (s+t) = (r \times s) + (r \times t)$, $(s+t) \times r = (s \times r) + (t \times r)$,

5. I has 0, such that $r + 0 = r$,

6. I has $-r$, such that $r + (-r) = 0$.

where $r, s, t \in I$, any set of polynomials in $\mathcal{R}$.

Properties 2,3,4 hold for all members of $\mathcal{R}$, in other words, I inherits these properties from $\mathcal{R}$. $\mathcal{C}$ is linear so $\mathbf{0} \in \mathcal{C}$ and $c \in \mathcal{C} \Rightarrow (-1)c \in \mathcal{C}$, means properties 5 and 6 also hold for I. The addition part of property 1 also follows from the linearity of $\mathcal{C}$.

And also we need to show I has the ideal property. Let c be any codeword, represented by $c(x)$ in I, and $g(x) = g_0 + g_1 x + ... + g_{n-1}x^{n-1}$ be any polynomial in $\mathcal{R}$. Then

$$g(x)c(x) = g_0\underline{c(x)} + g_1\underline{xc(x)} + ... + g_{n-1}\underline{x^{n-1}c(x)}$$

and this represents the word

$$g_0 c + g_1 c^1 + g_2 c^2 + ... + g_{n-1}c^{n-1}$$

which is a linear combination of cyclic shifts of c, so must be in $\mathcal{C}$. Hence $g(x)c(x) \in I$ so I has the ideal property.

And for the multiplicative part of property 1, it is automatically shown by: $g(x)c(x) \in I$ for all $g \in \mathcal{R}, c \in I$, so in particular $g(x)c(x) \in I$ for all $g \in I, c \in I$.

■

By this theorem, we can say that $\mathcal{C}$ is a cyclic code if and only if its set of representative polynomials in $\mathcal{R}$ is the set of all multiples of some single polynomial, and conversely, every such set of polynomials represents a cyclic code. We write $I = \langle g \rangle$ for the ideal consisting of all multiples of g, where g is generator of I. Then $g(x)$ is the *generator polynomial* of $\mathcal{C}$.

Theorem 2.1.19 [6] *Let $\mathcal{C}$ be a nonzero cyclic code in $\mathcal{R}$. Then,*

(i) *there exists a unique monic polynomial $g(x)$ of smallest degree in $\mathcal{C}$,*

(ii) $\mathcal{C} = \langle g(x) \rangle$,

(iii) $g(x)$ *is a factor of* $x^n - 1$.

Proof.

(i) Suppose both of the monic polynomials $g(x)$ and $h(x)$ are of least degree in $\mathcal{C}$. Then $g(x) - h(x) \in \mathcal{C}$ and has smaller degree. This gives a contradiction if $g(x) \neq h(x)$, for then a suitable scalar multiple of $g(x) - h(x)$ is monic, is in $\mathcal{C}$ and is of smaller degree than $g(x)$.

(ii) Suppose $a(x) \in \mathcal{C}$. By the division algorithm for $\mathbb{F}_q[x]$, $a(x) = q(x)g(x) + r(x)$, where $deg(r(x)) < deg(g(x))$. But, $r(x) = a(x) - q(x)g(x) \in \mathcal{C}$. By the minimality of $deg(g(x))$, we must have $r(x) = 0$ and so $a(x) \in \langle g(x) \rangle$.

(iii) By the division algorithm,

$$x^n - 1 = q(x)g(x) + r(x)$$

where $deg(r(x)) < deg(g(x))$. But then $r(x) \equiv -q(x)g(x)\ mod\ (x^n - 1)$, and so $r(x) \in \langle g(x) \rangle$. By the minimality of $deg(g(x))$, we must have $r(x) = 0$, which implies that $g(x)$ is a factor of $x^n - 1$.

■

2.1.5.1 Generator Matrix and Generator Polynomial

We have seen that linear codes have generator matrix and the subclass of cyclic codes have generator polynomials.

Definition 2.1.20 *In a nonzero cyclic code $\mathcal{C}$, the monic polynomial of least degree is called the generator polynomial of $\mathcal{C}$.*

The generator polynomial of a cyclic $[n, k]$ code has degree $n - k$.

Theorem 2.1.21 [6] *Suppose nonzero cyclic code $\mathcal{C}$ with generator polynomial*

$$g(x) = g_0 + g_1 x + \cdots + g_{n-k} x^{n-k}$$

of degree $n-k$. Then $dim\mathcal{C} = k$ and it has a generator matrix G of the form

$$\begin{pmatrix} g_0 & g_1 & g_2 & \cdots & g_{n-k} & 0 & 0 & \cdots & 0 \\ 0 & g_0 & g_1 & \cdots & g_{n-k-1} & g_{n-k} & 0 & \cdots & 0 \\ \vdots & & \ddots & & & & \ddots & & \\ \vdots & & & \ddots & & & & \ddots & \\ 0 & 0 & \cdots & 0 & g_0 & g_1 & \cdots & \cdots & g_{n-k} \end{pmatrix}$$

in which each row is the first cyclic shift of the previous one, and neither g_0 nor g_{n-k} is zero.

Proof. First observe that this matrix has k rows, which is the dimension of $\mathcal{C}$. If the first row is a codeword of $\mathcal{C}$, then so are all the rest by the cyclic property. Then we need to show that rows are linearly independent and that $\mathcal{C}$ has a codeword of the form given by first row.
These k rows represent the codewords $g(x), xg(x), \ldots, x^{k-1}g(x)$, and it remains only to show that every codeword in $\mathcal{C}$ can be expressed as a linear combination of them. Let $a(x)$ be a codeword in $\mathcal{C}$. Then,

$$a(x) = q(x)g(x)$$

for some polynomial $q(x)$, and no need modulo $x^n - 1$ since $deg(a(x)) < n$. It follows that $deg(q(x)) < k$. Hence,

$$\begin{aligned} q(x)g(x) &= (q_0 + q_1x + \cdots + q_{k-1}x^{k-1})g(x) \\ &= q_0g(x) + q_1xg(x) + \cdots + q_{k-1}x^{k-1}g(x), \end{aligned}$$

which is the desired linear combination. ■

Remark 2.1.22 *g_0 and g_{n-k} must be nonzero because, if $g_0 = 0$, then the first column of G is all zero, so every codeword of $\mathcal{C}$ has zero as its first place. But this is impossible because $\mathcal{C}$ has nonzero words so some cyclic shift of such a word will have non-zero digit as its first symbol. Similarly, by considering the last digit of the codewords, g_{n-k} cannot be zero.*

2.1.5.2 Parity-check Matrix and Parity-check Polynomial

Since the given generator matrix of a cyclic code is not in the standard form, it is not appropriate to write the parity-check matrix from standard form of G for cyclic codes. However, it is closely related to a polynomial interpretation of the parity check matrix of a cyclic code.
Let $\mathcal{C} = \langle g(x) \rangle$ be a cyclic $[n, k]$ code with generator polynomial $g(x)$. It is known that $g(x)$ is a factor of $x^n - 1$. So $g(x)h(x) = x^n - 1$ for some polynomial $h(x)$. Now $deg(g) = n - k$ so $deg(h) = k$. Also since $g(x)$ and $x^n - 1$ are monic, $h(x)$ is monic.

Definition 2.1.23 *The polynomial $h(x)$ introduced above is called the check polynomial of $\mathcal{C}$.*

Theorem 2.1.24 [1] *$c(x)$ corresponds to a codeword of $\mathcal{C}$ if and only if $c(x)h(x) \equiv 0$ mod $(x^n - 1)$.*

Proof.
($\Rightarrow$)

$$c \in \mathcal{C} \Rightarrow c(x) \equiv a(x)g(x) \ mod \ x^n - 1 \text{ for some } a(x) \in \mathbb{F}[X]$$
$$\Rightarrow c(x)h(x) \equiv a(x)g(x)h(x) \equiv 0 \ mod \ x^n - 1$$

($\Leftarrow$) Conversely, suppose $c(x)h(x) \equiv 0$. Divide $c(x)$ by $g(x)$ to get $c(x) = g(x)q(x) + r(x)$ with $\deg(r) < n - k$. Then

$$c(x)h(x) \equiv 0 \Rightarrow g(x)q(x)h(x) + r(x)h(x) \equiv 0$$
$$\Rightarrow r(x)h(x) \equiv 0$$
$$\Rightarrow r(x)h(x) \text{ is a multiple of } x^n - 1$$

But, $\deg(r) < n - k$ and $\deg(h)=k$, so $\deg(rh) < n$ and hence $rh = 0$. Therefore $r = 0$ so $c(x) = g(x)q(x)$; that is, $c \in \mathcal{C}$. ■

Theorem 2.1.25 *If $\mathcal{C}$ is cyclic then so is $\mathcal{C}^{\perp}$.*

Proof. Let $\mathcal{C}$ be a cyclic $[n,k]$ code. Let c be any codeword, c^t be its tth cyclic shift; that is, if $c=(c_1,c_2,\ldots,c_n)$ then $c^t=(c_{n-t+1},c_{n-t+2},\ldots,c_n,c_1,c_2,\ldots,c_{n-t})$. Note that is immediate from the definition of a cyclic code that if $c\in\mathcal{C}$, then $c^t\in\mathcal{C}$ for all t. We shall show $h^1\in\mathcal{C}^\perp$ whenever $h\in\mathcal{C}^\perp$ thus providing the cyclicity of $\mathcal{C}^\perp$.

$$\begin{aligned} h^1\cdot c &= h_nc_1+h_1c_2+\cdots+h_{n-1}c_n \\ &= h_1c_2+\cdots+h_{n-1}c_n+h_nc_1 \\ &= h\cdot c^{n-1} \text{ because } h\in\mathcal{C}^\perp \text{ and } c^{n-1}\in\mathcal{C} \end{aligned}$$

Hence $h^1\in\mathcal{C}^\perp$ as required. ■

In spite of its name, the polynomial $h(x)$ does not generate $\mathcal{C}^\perp$. The point is that the product of $c(x)$ and $h(x)$ being zero in $\mathcal{R}$ is not the same thing as the corresponding codewords in $\mathbb{F}_q^n$ being orthogonal. But there is a close connection between $h(x)$ and $\mathcal{C}^\perp$.

So, using this connection, we can find the generator matrix for the dual $\mathcal{C}^\perp$, in other words, the parity-check matrix for $\mathcal{C}$. And we can write the generator polynomial for $\mathcal{C}^\perp$.

Theorem 2.1.26 [6] *Suppose $\mathcal{C}$ is a $[n,k]$ cyclic code with check polynomial*

$$h(x)=h_0+h_1x+\cdots+h_kx^k.$$

Then,
(i) *a parity-check matrix for $\mathcal{C}$ is*

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

(ii) *$\mathcal{C}^\perp$ is a cyclic code which is generated by the reciprocal polynomial of $h(x)$*

$$\bar{h}(x) = h_k + h_{k-1}x + \cdots + h_0 x^k.$$

Proof.

(i) A polynomial $c(x) = c_0 + c_1 x + \cdots + c_{n-1}x^{n-1}$ is a codeword if and only if $c(x)h(x) = 0$. Now for $c(x)h(x)$ to be zero, then in particular the coefficient of $x^k, x^{k+1}, \ldots, x^{n-1}$ must all be zero, i.e.

$$\begin{array}{rcccccccc}
c_0h_k+ & c_1h_{k-1}+ & \cdots & + & c_kh_0 & & & = & 0 \\
 & c_1h_k+ & c_2h_{k-1}+ & \cdots & + \; c_{k+1}h_0 & & & = & 0 \\
 & \ddots & & & & & \ddots & \vdots & \\
 & & c_{n-k-1}h_k+ & \cdots & & & +c_{n-1}h_0 & = & 0
\end{array}$$

Thus any codeword $c_0c_1\cdots c_{n-1}$ of $\mathcal{C}$ is orthogonal to the vector $h_kh_{k-1}\cdots h_000\cdots 0$ and its cyclic shifts. So the rows of the matrix H given in the statement of the theorem are all codewords of $\mathcal{C}^\perp$. We have already observed that $h(x)$ is monic of degree k and so $h_k = 1$; it means that the rows of H are linearly independent, since it is in the echelon form and there are nothing other than zero below 1s. The number of rows of H is $n-k$, which is the dimension of $\mathcal{C}^\perp$. Hence H is a generator matrix of $\mathcal{C}^\perp$, i.e. a parity-check matrix for $\mathcal{C}$.

(ii) If we can show that $\bar{h}(x)$ is a factor of $x^n - 1$, then we will say $\langle \bar{h}(x) \rangle$ is a cyclic code whose generator matrix is the above matrix H, and hence that $\langle \bar{h}(x) \rangle = \mathcal{C}^\perp$. We observe that $\bar{h}(x) = x^k h(x{-}1)$. Since $h(x^{-1})g(x^{-1}) = (x^{-1})^n - 1$, we have $x^k h(x^{-1})x^{n-k}g(x^{-1}) = x^n(x^{-n} - 1) = 1 - x^n$, and so $\bar{h}(x) = x^k h(x^{-1})$ is a factor of $x^n - 1$.

■

CHAPTER 3

QUASI-CYCLIC CODES

3.1 Preliminaries

Let $\mathbb{F}_q$ be a finite field and m be a positive integer coprime with the characteristic of $\mathbb{F}_q$.

Definition 3.1.1 [5] *A linear code $\mathcal{C}$ of length ℓm over $\mathbb{F}_q$ is called quasi-cyclic code if the codeword*

$$(c_{0,0}, \ldots, c_{0,\ell-1}, c_{1,0}, \ldots, c_{1,\ell-1}, \ldots, c_{m-1,0}, \ldots, c_{m-1,\ell-1}) \in \mathcal{C}$$

then

$$(c_{m-1,0}, \ldots, c_{m-1,\ell-1}, c_{0,0}, \ldots, c_{0,\ell-1}, \ldots, c_{m-2,0}, \ldots, c_{m-2,\ell-1}) \in \mathcal{C}.$$

We denote the standard shift operator as T on $\mathbb{F}_q$. This code is invariant under T^ℓ (so under ℓ-shift) and this codes are called as *ℓ-quasi-cyclic codes* or *quasi-cyclic codes of index ℓ*. The quasi-cyclic codes are the generalization of cyclic codes, when $\ell = 1$ it is just a cyclic code.

Example 3.1.2 *The binary $[6, 3]$ code with generator matrix*

$$\begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$

is quasi-cyclic code of index 2.

For binary codes, permutations of the coordinates of a code form another equivalent code. For this code, these two matrices

$$G = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix} \quad \textit{and} \quad G' = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

G and G' generate equivalent codes for $\ell = 2$ and $m = 3$.

3.2 1-1 correspondence:

Let $\mathcal{R}$ be a commutative ring with identity. A linear code $\mathcal{C}$ of length n over $\mathcal{R}$ is defined to be a $\mathcal{R}$-submodule of $\mathcal{R}^n$. If $\mathcal{R}$ is a finite field $\mathbb{F}_q$ of order q, the linear code $\mathcal{C}$ of order n over $\mathbb{F}_q$ is an $\mathbb{F}_q$-vector subspace of $\mathbb{F}_q^n$.

Let $\mathbb{F}_q[Y]$ denote the polynomials, Y be indeterminate and coefficients be in $\mathbb{F}_q$. Define the ring as $\mathcal{R} := \mathcal{R}(\mathbb{F}_q, m) = \mathbb{F}_q[Y]/(Y^m - 1)$. This ring is the same in the polynomial representation of cyclic codes of length m over $\mathbb{F}_q$. Namely, cyclic codes of length m over $\mathbb{F}_q$ are ideals of $\mathcal{R}(\mathbb{F}_q, m)$.

Modules over $\mathcal{R}$ is closely related to the ideals in $\mathbb{F}_q[Y]/(Y^m - 1)$. In fact, ideals are just 1-dimensional $\mathcal{R}$-submodules.

Let $\mathcal{C}$ be a ℓ-quasi-cyclic code over $\mathbb{F}_q$ of length ℓm. Let

$$c = (c_{0,0}, \ldots, c_{0,\ell-1}, c_{1,0}, \ldots, c_{1,\ell-1}, \ldots, c_{m-1,0}, \ldots, c_{m-1,\ell-1})$$

denote a codeword in $\mathcal{C}$.

Define a map $\phi : \mathbb{F}_q^{\ell m} \to \mathcal{R}^\ell$ by

$$\phi(c) = (c_0(Y), c_1(Y), \ldots, c_{\ell-1}(Y)) \in \mathcal{R}^\ell$$

where

$$c_j(Y) = \sum_{i=0}^{m-1} c_{ij} Y^i \in \mathcal{R}, \quad j = 0, \ldots, \ell - 1.$$

We denote by $\phi(\mathcal{C})$ the image of $\mathcal{C}$ under ϕ.

Lemma 3.2.1 [10] *The map ϕ gives a one-to-one correspondence between ℓ-quasi-cyclic codes over $\mathbb{F}_q$ of length ℓm and linear codes over $\mathcal{R}$ of length ℓ.*

Proof.

($\Rightarrow$) $\phi(\mathcal{C})$ is closed under scalar multiplication by elements of $\mathbb{F}_q$ because of the linearity of $\mathcal{C}$ over $\mathbb{F}_q$. Since $Y^m = 1$ in $\mathcal{R}$,

$$Yc_j(Y) = \sum_{i=0}^{m-1} c_{ij} Y^{i+1} = \sum_{i=0}^{m-1} c_{i-1,j} Y^i$$

where the subscripts are taken modulo m.

The word

$$(Yc_0(Y), Yc_1(Y), \ldots, Yc_{\ell-1}(Y)) \in \mathcal{R}^\ell$$

corresponds to the word

$$(c_{m-1,0}, \ldots, c_{m-1,\ell-1}, c_{0,0}, \ldots, c_{0,\ell-1}, \ldots, c_{m-2,0}, \ldots, c_{m-2,\ell-1}) \in \mathbb{F}_q^{\ell m}$$

which is in $\mathcal{C}$ since $\mathcal{C}$ is ℓ-quasi-cyclic code. Therefore, $\phi(\mathcal{C})$ is closed under multiplication by Y, and hence $\phi(\mathcal{C})$ is an $\mathcal{R}$-submodule of $\mathcal{R}^\ell$.

($\Leftarrow$) It is easy to see that every linear code over $\mathcal{R}$ of length ℓ comes from a quasi-cyclic code of index ℓ and length ℓm over $\mathbb{F}_q$. ■

Proposition 3.2.2 [10] *Let $a, b \in \mathbb{F}_q^{\ell m}$. Then $(T^{\ell k}(a)) \cdot b = 0$ for $0 \leq k \leq m-1$ if and only if $\langle \phi(a), \phi(b) \rangle = 0$.*

Proof.

$\langle \phi(a), \phi(b) \rangle = 0$ means

$$0 = \sum_{j=0}^{\ell-1} a_j \overline{b_j} = \sum_{j=0}^{\ell-1} \left(\sum_{i=0}^{m-1} a_{ij} Y^i \right) \left(\sum_{k=0}^{m-1} b_{kj} Y^{-k} \right). \tag{3.1}$$

Comparing the coefficients of Y^h on both sides, the equation 3.1 is equivalent to

$$\sum_{j=0}^{\ell-1} \sum_{i=0}^{m-1} a_{i+h,j} b_{ij} = 0, \quad \text{for all} \quad 0 \leq h \leq m-1 \tag{3.2}$$

where the subscripts $i+h$ is considered to be in $\{0, 1, \ldots, m-1\}$ by taking modulo m. The equation 3.2 means that $(T^{-\ell h}(a)) \cdot b = 0$. Since $T^{-\ell h} = T^{\ell(m-h)}$, it follows that the equation 3.2, and hence $\langle \phi(a), \phi(b) \rangle = 0$, is equivalent to $(T^{\ell k}(a)) \cdot b = 0$ for $0 \leq k \leq m-1$. ■

From this proposition, it follows that a quasi-cyclic code $\mathcal{C}$ is self-dual with respect to the Euclidean inner product if and only if $\phi(\mathcal{C})$ is self-dual with respect to the Hermitian inner product, where $\mathcal{C}$ is an ℓ-quasi-cyclic code over $\mathbb{F}_q$ of length ℓm and $\phi(\mathcal{C})$ is its image in $\mathcal{R}^\ell$ under ϕ. And also $\phi(\mathcal{C})^\perp = \phi(\mathcal{C}^\perp)$, where the dual in $\mathbb{F}_q^{\ell m}$ is taken with respect to the Euclidean inner product, while the dual in $\mathcal{R}^\ell$ is taken with respect to the Hermitian inner product.

CHAPTER 4

CONSTRUCTION OF QUASI-CYCLIC SELF-DUAL CODES

4.1 Ring Decomposition

Let

$$\mathcal{R} = \mathcal{R}(\mathbb{F}_q, m) = \mathbb{F}_q[Y]/(Y^m - 1).$$

When m is coprime with the characteristic of $\mathbb{F}_q$, so q, the ring can be decomposed into a direct sum of fields by Chinese Remainder Theorem (CRT) or Discrete Fourier Transform (DFT) [10]. There are two benefits of this approach, as investigating self-dual quasi-cyclic codes in a systematic way and decomposing the quasi-cyclic codes into codes of lower lengths.

The polynomial $Y^m - 1$ factors completely into distinct irreducible factors in $\mathbb{F}_q[Y]$, so it can be written as

$$Y^m - 1 = \delta g_1 \dots g_s h_1 h_1^* \dots h_t h_t^* \tag{4.1}$$

where δ is nonzero in $\mathbb{F}_q$, $g_1, \dots, g_s$ are the polynomials which are self-reciprocal, and h_i^*'s are reciprocals of h_i's, for all $1 \leq i \leq t$.

4.1.1 Decomposition by the Chinese Remainder Theorem

The ring $\mathcal{R}$ can be written as by CRT, [10]

$$\mathcal{R} = \frac{F_q[Y]}{(Y^m - 1)} = \left(\bigoplus_{i=1}^{s} \frac{F_q[Y]}{(g_i)} \right) \oplus \left(\bigoplus_{j=1}^{t} \left(\frac{F_q[Y]}{(h_j)} \oplus \frac{F_q[Y]}{(h_j^*)} \right) \right) \tag{4.2}$$

The direct sum on the right-hand side is endowed with the coordinate-wise addition and multiplication.

Let $F_q[Y]/(g_i)$ be denoted by G_i, and in the same way $F_q[Y]/(h_j)$ by H'_j and $F_q[Y]/(h_j^*)$ by H''_j for simplicity of notation. Then

$$\mathcal{R}^\ell = \left(\bigoplus_{i=1}^{s} G_i^\ell\right) \oplus \left(\bigoplus_{j=1}^{t}\left(H_j'^\ell \oplus H_j''^\ell\right)\right).$$

In particular, every $\mathcal{R}$-linear code $\mathcal{C}$ of length ℓ can be decomposed as the direct sum

$$\mathcal{C} = \left(\bigoplus_{i=1}^{s} \mathcal{C}_i\right) \oplus \left(\bigoplus_{j=1}^{t}\left(\mathcal{C}'_j \oplus \mathcal{C}''_j\right)\right)$$

where $\mathcal{C}_i$, $\mathcal{C}'_j$ and $\mathcal{C}''_j$ are linear codes over G_i, H'_j and H''_j, respectively, all of length ℓ for each $1 \le i \le s$, and for each $1 \le j \le t$.

Every element of $\mathcal{R}$ can be written as $c(Y)$, for some polynomial $c \in \mathbb{F}_q[Y]$. The decomposition (4.2) shows that $c(Y)$ may also be written as an $(s+2t)$-tuple

$$(c_1(Y), \ldots, c_s(Y), c'_1(Y), c''_1(Y), \ldots, c'_t(Y), c''_t(Y)) \tag{4.3}$$

where $c_i(Y) \in G_i$, $(1 \le i \le s)$, $c'_j(Y) \in H'_j$, $(1 \le j \le t)$ and $c''_j(Y) \in H''_j$, $(1 \le j \le t)$.

Here, we can consider the c_i, c'_j and c''_j as polynomials in $\mathbb{F}_q[Y]$. Also [10] we can write $\overline{c(Y)}$ as

$$(\overline{c_1(Y)}, \ldots, \overline{c_s(Y)}, c''_1(Y), c'_1(Y), \ldots, c''_t(Y), c'_t(Y)).$$

Theorem 4.1.1 [10] *A linear code $\mathcal{C}$ over $\mathcal{R}$ of length ℓ is self-dual with respect to Hermitian inner product, or equivalently, an ℓ-quasi-cyclic code of length ℓm over $\mathbb{F}_q$ is self-dual with respect to Euclidean inner product, if and only if*

$$\mathcal{C} = \left(\bigoplus_{i=1}^{s} \mathcal{C}_i\right) \oplus \left(\bigoplus_{j=1}^{t}\left(\mathcal{C}'_j \oplus (\mathcal{C}'_j)^\perp\right)\right)$$

where, for $1 \le i \le s$, $\mathcal{C}_i$ is a self-dual code over G_i of length ℓ with respect to the Hermitian inner product and for $1 \le j \le t$, $\mathcal{C}'_j$ is a linear code of length ℓ over H'_j and $(\mathcal{C}')^\perp$ is its dual with respect to the Euclidean inner product.

4.1.2 The Discrete Fourier Transform

Let m be coprime with q. In the case $m \in \mathbb{F}_q^* := \mathbb{F}_q - \{0\}$, and the isomorphism (4.2) can be described by DFT, [10].

There is a one-to-one correspondence between the factors g_i, h_j and h_j^* and the q-cyclotomic cosets of $\mathbb{Z}/m\mathbb{Z}$. The q-cyclotomic cosets corresponding to g_i, h_j and h_j^* are U_i, V_j and W_j respectively. For

$$c = \sum_{g \in \mathbb{Z}/m\mathbb{Z}} c_g Y^g \in \mathbb{F}_q[Y]/(Y^m - 1),$$

its Fourier transform is

$$\hat{c} = \sum_{h \in \mathbb{Z}/m\mathbb{Z}} \hat{c_h} Y^h,$$

where the Fourier coefficient $\hat{c_h}$ is defined as

$$\hat{c_h} = \sum_{g \in \mathbb{Z}/m\mathbb{Z}} c_g \omega^{gh}$$

where ω is a primitive mth root of unity in some Galois extension of $\mathbb{F}_q$.

For a vector x, by its Fourier transform, it is meant that the vector whose ith entry is the Fourier transform of the ith entry of x. This gives the following trace parametrization for quasi-cyclic codes over finite fields.

Theorem 4.1.2 [10] *Let m is coprime with q. Then, the following construction gives the quasi-cyclic codes over $\mathbb{F}_q$ of length ℓm and of index ℓ, for any ℓ.*

Let $Y^m - 1 = \delta g_1 \dots g_s h_1 h_1^* \dots h_t h_t^*$, with the assumptions mentioned above, and also

$F_q[Y]/(g_i) = G_i$, $F_q[Y]/(h_j) = H_j'$ and $F_q[Y]/(h_j^*) = H_j''$.

Let U_i (respectively, V_j and W_j) denote the cyclotomic coset of $\mathbb{Z}/m\mathbb{Z}$ corresponding to G_i, (respectively H_j' and H_j'') and fix $u_i \in U_i$, $v_j \in V_j$ and $w_j \in W_j$.

For each i, let $\mathcal{C}_i$ be a code over G_i of length ℓ, for each j, let $\mathcal{C}_j'$ be a code over H_j' of length ℓ and let $\mathcal{C}_j''$ be a code over H_j'' of length ℓ.

For $x_i \in \mathcal{C}_i$, $y'_j \in \mathcal{C}'_j$ and $y''_j \in \mathcal{C}''_j$, and for each $0 \leq g \leq m-1$, let

$$c_g((x_i),(y'_j),(y''_j)) = \sum_{i=1}^{s} Tr_{G_i/\mathbb{F}_q}(x_i \omega^{-gu_i}) + \sum_{j=1}^{t} (Tr_{H'_j/\mathbb{F}_q}(y'_j \omega^{-gv_j}) + Tr_{H''_j/\mathbb{F}_q}(y''_j \omega^{-gw_j})).$$

Then the code

$$\mathcal{C} = \{(c_0((x_i),(y'_j),(y''_j)), \ldots, c_{m-1}((x_i),(y'_j),(y''_j))) \mid \forall x_i \in \mathcal{C}_i, \forall y'_j \in \mathcal{C}'_j \text{ and } \forall y''_j \in \mathcal{C}''_j\}$$

is a quasi-cyclic code over $\mathbb{F}_q$ of length ℓm and of index ℓ. Conversely, every ℓ-quasi-cyclic code over $\mathbb{F}_q$ of length ℓm is obtained through this construction.

Moreover, $\mathcal{C}$ is self-dual with respect to the Euclidean inner product if and only if the $\mathcal{C}_i$ are self-dual with respect to the Hermitian inner product and $\mathcal{C}''_j = (\mathcal{C}'_j)^{\perp}$ for each j with respect to Euclidean inner product.

4.2 Existence of Self-dual Codes

Let $\mathcal{R} = \mathcal{R}(\mathbb{F}_q, m) = \mathbb{F}_q[Y]/(Y^m - 1)$.
Self-dual codes over $\mathcal{R}$ are understood self-dual codes with respect to Hermitian inner product.
This section contains some lemmas regarding the length and possible weight enumerators of self-dual codes.

Proposition 4.2.1 [10] *Let m be relatively prime to q and let ℓ be odd. Then no self-dual ℓ-quasi-cyclic codes over $\mathbb{F}_q$ of length ℓm exist.*

Proof. Since $Y - 1$ is a factor of $Y^m - 1$, $\mathbb{F}_q$ is always a direct factor of $\mathcal{R}$ in the decomposition (4.1). Since ℓ is odd, no self-dual code of length ℓ exists over $\mathbb{F}_q$. ■

The following lemma gives more information about length.

Lemma 4.2.2 [5] *Let* $\mathcal{R} = \mathcal{R}(\mathbb{F}_q, m) = \mathbb{F}_q[Y]/(Y^m - 1)$.

(i) If $char(\mathbb{F}_q) = 2$ *or* $q \equiv 1 \ (mod\ 4)$*, then there exists a self-dual code of length* ℓ *over* $\mathcal{R}$ *if and only if* $2 \mid \ell$.

(ii) If $q \equiv 3 \ (mod\ 4)$*, then there exists a self-dual code of length* ℓ *over* $\mathcal{R}$ *if and only if* $4 \mid \ell$.

Proof.

To prove the lemma, suppose $\mathcal{C}$ is a self-dual code of length ℓ over $\mathcal{R}$. Assume that $\mathcal{C}_1$ in the decomposition of $\mathcal{C}$ in (4.1.1) is a Euclidean self-dual code over $\mathbb{F}_q$ of length ℓ.

For *(i)*,

($\Rightarrow$) If a code is self-dual, then its length must be even. So $2 \mid \ell$.

($\Leftarrow$) Suppose $2 \mid \ell$. Let $\ell = 2k$.

We take an Euclidean self-dual code over $\mathbb{F}_q$ of length 2 using the generator matrix:

$G = \begin{bmatrix} 1 & c \end{bmatrix}$; where $c^2 = -1$ in $\mathbb{F}_q$. It is easily seen that a self-dual code $\mathcal{C}$ over $\mathcal{R}$ of length 2 is generated by this matrix. Then the direct sum of the k-copies of $\mathcal{C}$ form a self-dual code over $\mathcal{R}$ of length $2k = \ell$.

Remark that the reason how such a c exists is $char(\mathbb{F}_q) = 2$ or $q \equiv 1 \ (mod\ 4)$.

For *(ii)*,

($\Rightarrow$) It is well-known if $q \equiv 3 \ (mod\ 4)$, then a self-dual code of length n over $\mathbb{F}_q$ exists if and only if n is a multiple of 4.

By this observation, $4 \mid \ell$.

($\Leftarrow$) Suppose $4 \mid \ell$. Let $\ell = 4k$.

We take an Euclidean self-dual code over $\mathbb{F}_q$ of length 4 using the generator matrix

$G = \begin{bmatrix} 1 & 0 & \alpha & \beta \\ 0 & 1 & -\beta & \alpha \end{bmatrix}$ where $\alpha^2 + \beta^2 + 1 = 0$ in $\mathbb{F}_q$. A self-dual code $\mathcal{C}$ over $\mathcal{R}$ of length 4 is generated by this matrix. Then the direct sum of the k-copies of $\mathcal{C}$ form a self-dual code of length $4k = \ell$ over $\mathcal{R}$.

Remark that there exist such α and β in $\mathbb{F}_q$ because $q \equiv 3 \ (mod\ 4)$. ■

By the following lemma, possible weight enumerators of a binary ℓ-quasi-cyclic self-dual code of length ℓm with a prime m are determined.

Lemma 4.2.3 [11] *Let $\mathcal{C}$ be a binary code and H any subgroup of $Aut(\mathcal{C})$. If A_i is the total number of codewords in $\mathcal{C}$ of weight i and $A_i(H)$ is the number of codewords which are fixed by some non-identity element of H, then*

$$A_i \equiv A_i(H) \pmod{|H|}$$

Proof. We need to consider some non-identity element of H because the identity of H always fixes any codeword. Thus, we can divide the codewords of weight i into two classes: first one is the codewords fixed by some element of H, and the rest is second. The number of codewords fixed by an element in H is $A_i(H)$. Let $a \in \mathcal{C}$ be not fixed by any element of H. Then the $|H|$ codewords ga for $g \in H$ must all be distinct. Thus the number of these codewords must be multiple of $|H|$, say $m|H|$.
$\therefore A_i = A_i(H) + m|H|$
$\therefore A_i \equiv A_i(H) \pmod{|H|}$ ■

Proposition 4.2.4 [10] *A code $\mathcal{C}$ of length ℓm is quasi-cyclic code of index ℓ if and only if $Perm(\mathcal{C})$ contains a fixed point free (fpf) permutation consisting of ℓ disjoint m-cycles. In particular, if m denotes a prime, a fpf permutation with ℓ disjoint m-cycles means a fpf permutation group of order m.*

Proof.
($\Rightarrow$) If $\mathcal{C}$ is ℓ-quasi-cyclic then T^ℓ is the permutation sought for, where T denotes the cyclic shift.
($\Leftarrow$) If $Perm(\mathcal{C})$ contains such a permutation σ, then up to coordinate labeling, we can assume that $\sigma = T^\ell$. ■

Remark 4.2.5 *In this work, we study binary self-dual codes with a fixed point free automorphism of order three, so-called binary cubic self-dual codes.*

Lemma 4.2.6 [5] *Let $\mathcal{C}$ be a binary ℓ-quasi-cyclic self-dual code of length $m\ell$ with m prime. If m does not divide the weight i, then m must divide A_i.*

Proof. From previous proposition, $\mathcal{C}$ must contain an fpf permutation σ of order m. Let
$H = \langle\sigma\rangle$ whose order m. Since σ is an fpf of order m and any codewords of weight i with $m \nmid i$, cannot be fixed by any non-identity element of H, we have $A_i(H) = 0$.
$\therefore$ by previous lemma $A_i \equiv A_i(H) = 0 \ (mod\ m)$ ■

Example 4.2.7 *For $\ell = 12$ and $m = 3$.*
There are two weight enumerators for $[36, 18, 8]$ self-dual code.

$$\begin{aligned} W_1 &= 1 + 225y^8 + 2016y^{10} + \dots \\ W_2 &= 1 + 289y^8 + 1632y^{10} + \dots \end{aligned} \tag{4.4}$$

By previous lemma, we can directly say that there is no binary cubic self dual code having the weight enumerator W_2.
Since 3 should divide A_8 (if $m \nmid i$, then $m \mid A_i$).
But $3 \nmid 289$. So there is no code having W_2.

CHAPTER 5

APPLICATIONS

5.1 Constructions of Self-dual codes

This chapter contains some methods for combining codes to get new codes with greater length. The building-up construction and some constructions depending on m for self-dual codes over the ring $\mathcal{R} = \mathbb{F}_q[Y]/(Y^m - 1)$ are given. For more than the constructions which are mentioned in this chapter, one can look [11].

5.1.1 Building-up Construction

The following theorem is the building-up construction for self-dual codes over the ring $\mathcal{R}$, equivalently ℓ-quasi-cyclic self-dual code over $\mathbb{F}_q$. The construction is given for the case $char(\mathbb{F}_q) = 2$, although the method holds not only for the fields with even characteristic, but also for the other cases. However, we are interested in only that case. The construction is as follows.

Proposition 5.1.1 [5], [8] *Let $G_0 = (r_i)$ be a matrix generating the self-dual code $\mathcal{C}_0$ over $\mathcal{R}$ of length 2ℓ, where r_i is the ith row of the $k \times 2\ell$ matrix, G_0, for $1 \leq i \leq k$. Let $x = (x_1, \ldots, x_{2\ell})$ be a vector in $\mathcal{R}^{2\ell}$ with $\langle x, x \rangle = -1$ and let c be in $\mathcal{R}$ such that $c\bar{c} = -1$. Set $y_i = \langle r_i, x \rangle$ for $1 \leq i \leq k$. Then the following*

matrix

$$G = \left[\begin{array}{cc|c} 1 & 0 & x_1 \\ \hline y_1 & cy_1 & r_1 \\ \vdots & \vdots & \vdots \\ y_k & cy_k & r_k \end{array}\right]$$

generates a self-dual code $\mathcal{C}$ *over* $\mathcal{R}$ *of length* $2\ell + 2$.

Every self-dual code $\mathcal{C}$ over $\mathcal{R} = \mathbb{F}_q[Y]/(Y^m - 1)$ of length $2\ell + 2$ can be obtained by the building-up construction in the previous proposition (5.1.1) up to permutation equivalence, provided that $char(\mathbb{F}_q) = 2$ or $q \equiv 1 \ (mod\ 4)$, m is a prime p, and q is a primitive element of $\mathbb{F}_p$, [5].

The rest of this chapter includes some constructions for self-dual codes over $\mathcal{R} = \mathbb{F}_q[Y]/(Y^m - 1)$ which depends on m.

5.1.2 The $(\ u + v \mid u - v\ )$ Construction for $m = 2$

In this subsection, we consider ℓ-quasi-cyclic codes over the finite field $\mathbb{F}_q$ of length 2ℓ.

(i) q is odd. If $\mathcal{C}_1$ and $\mathcal{C}_2$ are codes of length ℓ over $\mathbb{F}_q$, then

$$\mathcal{C}\{\ (\ u + v \mid u - v) \mid u \in \mathcal{C}_1,\ v \in \mathcal{C}_2\}$$

is an ℓ-quasi-cyclic code of length 2ℓ over $\mathbb{F}_q$. All ℓ-quasi-cyclic codes of length 2ℓ over $\mathbb{F}_q$ are constructed by this construction. Moreover, $\mathcal{C}$ self-dual if and only if $\mathcal{C}_1$ and $\mathcal{C}_2$ are self-dual with respect to the Euclidean inner product, [10].

(ii) q is even. If q is a power of 2, then $Y^2 - 1 = (Y - 1)^2$, so $\mathcal{R}$ is the ring $\mathbb{F}_q + u\mathbb{F}_q$, where $u^2 = 0$. Therefore, every ℓ-quasi-cyclic code of length 2ℓ over $\mathbb{F}_q$ (q even) can be realized as a code of length ℓ over $\mathbb{F}_q + u\mathbb{F}_q$, [10].

5.1.3 Construction for $m = 3$

This is the case which this work focuses on. Assume that $m = 3$ and that q is not a power of 3.

(i) $q \equiv 2 \ (mod\ 3)$

$Y^2 + Y + 1$ is irreducible in $\mathbb{F}_q[Y]$, so

$$Y^3 - 1 = (Y - 1)(Y^2 + Y + 1)$$

as a product of irreducible factors. By (4.2), $\mathcal{R}$ can be decomposed as

$$\mathcal{R} = \frac{\mathbb{F}_q[Y]}{(Y^3 - 1)} = \mathbb{F}_q \oplus \mathbb{F}_{q^2}.$$

This isomorphism gives a correspondence between the ℓ-quasi-cyclic codes $\mathcal{C}$ of length 3ℓ over $\mathbb{F}_q$ and a pair $(\mathcal{C}_1, \mathcal{C}_2)$, where $\mathcal{C}_1$ is a linear code over $\mathbb{F}_q$ of length ℓ and $\mathcal{C}_2$ is a linear code over F_{q^2} of length ℓ. Using the Discrete Fourier Transform [10] and the theorem (4.1.2), we have

$$\mathcal{C} = \{ \ (\ x + 2a - b \mid x - a + 2b \mid x - a - b \ ) \mid x \in \mathcal{C}_1, \ a + \omega b \in \mathcal{C}_2 \} \quad (5.1)$$

where $\omega^2 + \omega + 1 = 0$.

Moreover, $\mathcal{C}$ is self-dual if and only if $\mathcal{C}_1$ is self-dual with respect to Euclidean inner product and $\mathcal{C}_2$ is self-dual with respect to Hermitian inner product.

(ii) $q \equiv 1 \ (mod\ 3)$

In this case, $Y^3 - 1$ factors completely into $(Y - 1)(Y - \omega)(Y - \omega^2)$, where $\omega^2 + \omega + 1 = 0$ and $\omega \in \mathbb{F}_q$. An ℓ-quasi-cyclic code $\mathcal{C}$ over $\mathbb{F}_q$ of length ℓ decomposes into $\mathcal{C}_1 \oplus \mathcal{C}_2 \oplus \mathcal{C}_3$, where $\mathcal{C}_1, \mathcal{C}_2$ and $\mathcal{C}_3$ are codes over $\mathbb{F}_q$ of length ℓ. Moreover, $\mathcal{C}$ is self-dual if and only if $\mathcal{C}_1$ is self-dual with respect to the Euclidean inner product and $\mathcal{C}_3 = \mathcal{C}_2^{\perp}$ with respect to the Euclidean inner product, [10].

In the next chapter, for binary case, more details and some examples are given about construction.

5.1.4 Construction for $m = 5$

In this subsection, we consider ℓ-quasi-cyclic codes over the finite field $\mathbb{F}_q$ of length 5ℓ.

Suppose that $m = 5$ and q is such that $Y^4 + Y^3 + Y^2 + Y + 1$ is irreducible in $\mathbb{F}_q[Y]$. Let $\omega \in \mathbb{F}_{q^4}$ be such that $\omega^4 + \omega^3 + \omega^2 + \omega + 1 = 0$ and let Tr denote the trace from $\mathbb{F}_{q^4}$ to $\mathbb{F}_q$. Then, for $\mathcal{C}_1$ a code of length ℓ over $\mathbb{F}_q$ and $\mathcal{C}_2$ a code of length ℓ over $\mathbb{F}_{q^4}$, the code

$$\mathcal{C} = \{ \, (\, x + Tr(y) \mid x + Tr(y\omega^{-1}) \mid Tr(y\omega^{-2}) \mid x + Tr(y\omega^{-3}) \mid x + Tr(y\omega^{-4})) \\ \mid x \in \mathcal{C}_1, \; y \in \mathcal{C}_2 \}$$

is an ℓ-quasi-cyclic code of length 5ℓ over $\mathbb{F}_q$. All such codes are constructed by this way, [10].

Moreover, $\mathcal{C}$ is a self-dual code if and only if $\mathcal{C}_1$ and $\mathcal{C}_2$ are self-dual codes with respect to the Euclidean and the Hermitian inner product, respectively.

Remark 5.1.2 *When $q = 2^t$, the above construction is equivalent to the construction*
$(\, x+a \mid x+a+b \mid x+b+c \mid x+c+d \mid x+d \,)$, where $x \in \mathcal{C}_1$ and $a+b\omega+c\omega^2+d\omega^3 \in \mathcal{C}_2$.

5.1.5 Construction for $m = 7$

In this subsection, we consider ℓ-quasi-cyclic codes over the finite field $\mathbb{F}_q$ of length 7ℓ.

Suppose that $m = 7$ and $q = 2^t$ is such that $Y^7 - 1$ factors into irreducible polynomials such as $(Y - 1)(Y^3 + Y + 1)(Y^3 + Y^2 + 1)$. Let ω be a root of $Y^3 + Y + 1$ in $\mathbb{F}_{q^3}$. Let $\mathcal{C}_1$ be a code of length ℓ over $\mathbb{F}_q$ and let $\mathcal{C}_2, \mathcal{C}_3$ be codes of length ℓ over $\mathbb{F}_{q^3}$. Let Tr denote the trace from $\mathbb{F}_{q^3}$ to $\mathbb{F}_q$. Then the code

$$C = \{ \, (\, c_0, \ldots, c_6 \,) \mid c_i = x + Tr(y\omega^{-i}) + Tr(z\omega^i), \; x \in \mathcal{C}_1, \; y \in \mathcal{C}_2, \; z \in \mathcal{C}_3 \}$$

is an ℓ-quasi-cyclic code over $\mathbb{F}_q$ of length 7ℓ. All such codes are constructed by this way, [10]. Conversely, all ℓ-quasi-cyclic codes of length 7ℓ over $\mathbb{F}_q$ are

constructed by this way. Moreover, $\mathcal{C}$ is self-dual if and only if $\mathcal{C}_1$ is self-dual and $\mathcal{C}_3 = \mathcal{C}_2^{\perp}$.

Example 5.1.3 [10] *There is an extremal Type I code of length 42 which is cyclic, hence 6-quasi-cyclic. Its binary component $\mathcal{C}_1$ has to be equivalent to the unique [6,3,2] self-dual code.*

CHAPTER 6

CUBIC SELF-DUAL BINARY CODES

In this chapter, we assume that $m = 3$ and that q is not a power of 3. We study the ℓ-quasi-cyclic self-dual codes of length 3ℓ over $\mathbb{F}_q$, in other words, binary self-dual codes with a fixed point free automorphism with order 3 from the proposition (4.2.4). It is shown in [10] that all such codes can be obtained by a generalized cubic construction of Turyn's, from a code over $\mathbb{F}_2$ and a code over $\mathbb{F}_4$ both of length ℓ (binary and quaternary, respectively). Cubic binary codes of length 3ℓ are viewed as codes of length ℓ over the ring $\mathbb{F}_2 \times \mathbb{F}_4$, [3].
In the construction (5.1), when $q = 2^t$ (t odd) and for any ℓ

$$\mathcal{C} = \{ \ (\ x + b \mid x + a \mid x + a + b \ ) \mid x \in \mathcal{C}_1, \ a + \omega b \in \mathcal{C}_2 \}.$$

It is easily verified that, if $a, b \in \mathcal{C}_2'$ for some linear code $\mathcal{C}_2'$ over $\mathbb{F}_q$, then
$\mathcal{C}_2 := \{ \ a + b\omega \mid a, b \in \mathcal{C}_2' \}$ is a linear code over F_{q^2}, where $\omega^2 + \omega + 1 = 0$. So, if we begin with two $\mathbb{F}_q$-linear codes $\mathcal{C}_2'$ and $\mathcal{C}_1$, the construction in (6) gives Turyn's $(\ a + x \mid b + x \mid a + b + x \ )$-construction. Particularly, we obtain

Theorem 6.0.4 [10] *The* $(\ a+x \mid b+x \mid a+b+x \ )$*-construction which is applied to two linear codes* $\mathcal{C}_1$ *and* $\mathcal{C}_2'$ *over* $\mathbb{F}_{2^t}$ *(where* t *odd) of length* ℓ*, gives an* $\mathbb{F}_{2^t}$*-linear code* $\mathcal{C}$ *of length* 3ℓ *that is quasi-cyclic of index* ℓ*.*

Example 6.0.5 [11] *The binary extended Golay code* $\mathcal{G}_{24}$ *is obtained by Turyn's construction.*
If we choose $\mathcal{C}_2'$ *as the binary extended Hamming code* ($[8, 4, 4]$ *code obtained by*

adding overall parity checks to $[7, 4, 3]$ *Hamming code) with generator matrix:*

$$A_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

and $\mathcal{C}_1$ *as equivalent code of this code by reversing the order of the coordinates of words with generator matrix:*

$$A'_8 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

then, the code $\mathcal{C}$ *given by the construction* (6) *is the binary extended* $[24, 12, 8]$ *Golay code as a quasi-cyclic code of index 8.*

No simple formula is known for the minimum distance of $\mathcal{C}$, although a lower bound is given by the following proposition:

Proposition 6.0.6 [11] *For any binary vectors* a, b, x*,*

$$\begin{aligned} wt(\ a + x \mid b + x \mid a + b + x\ ) &= 2wt(a \oplus b) - wt(x) + 4s, \\ &\geq 2wt(a \oplus b) - wt(x), \end{aligned}$$

where s *is the number of times* a *is 0,* b *is 0 and* x *is 1.*

In the paper [5], the classification of binary cubic self-dual codes of lengths up to 42 (up to permutation equivalence) is completed by building-up construction. After that, in [4], the classification of cubic self-dual $[48, 24, 10]$ code is completed. Up to permutation equivalence the numbers of cubic self-dual codes of lengths up to 48 are as follows [5]:

There is/are
for $\ell = 2$, unique binary cubic self-dual code of length 6,
for $\ell = 4$, 2 binary cubic self-dual codes of length 12,

for $\ell = 6$, 3 binary cubic self-dual codes of length 18,
for $\ell = 8$, 16 binary cubic self-dual codes of length 24,
for $\ell = 10$, 8 binary cubic self-dual codes of length 30,
for $\ell = 12$, 13 binary cubic self-dual codes of length 36,
for $\ell = 14$, 1569 binary cubic self-dual codes of length 42,
for $\ell = 16$, 264 binary cubic self-dual codes of length 48.

For $\ell = 18$, we have tried to find more codes by the cubic construction (6). Our motivation is that the number of inequivalent codes which are found is at only 7. Also, it is the lowest length which is not completed the classification. In order to increase the number, we use the cubic construction since it is proven that all binary cubic self-dual codes can be found by this construction.

For self-dual $[54, 27, 10]$ codes, there are two weight enumerators [5]

$$\begin{aligned} W_1 &= 1 + (351 - 8\beta)y^{10} + (5031 + 24\beta)y^{12} + \dots \quad (0 \le \beta \le 43) \\ W_2 &= 1 + (351 - 8\beta)y^{10} + (5543 + 24\beta)y^{12} + (43884 + 32\beta)y^{14} + \dots \end{aligned} \tag{6.1}$$

In [5], by building-up construction four inequivalent codes with W_1 for $\beta = 0, 3, 6, 9$ and three inequivalent codes with W_2 for $\beta = 12, 15, 18$ are found.

For cubic construction, we use the notations as in [3].
Binary codes $\mathcal{C}$ of length 54 are formed by this construction from a binary code $\mathcal{C}_1$ of length 18 and a quaternary code $\mathcal{C}_2$ of length 18. If A, B and X are binary vectors of length 18 then let

$$\begin{aligned} U &= X + A \\ V &= X + B \\ W &= X + A + B \end{aligned}$$

and writing $\mathbb{F}_4 = \mathbb{F}_2(\omega)$ we can define a Gray map from $\mathbb{F}_2^{18} \times \mathbb{F}_4^{18} \to \mathbb{F}_2^{54}$ as

$$\phi(X,\ A + \omega B) := (U \mid V \mid W). \tag{6.2}$$

With this notations, the constructed code $\mathcal{C}$ is $\phi(\mathcal{C}_1,\ \mathcal{C}_2)$.

For $\ell = 18$,

In [3], by this construction one $[54, 27, 10]$ code with weight enumerator W_1 for $\beta = 0$ and one $[54, 27, 10]$ code with weight enumerator W_2 for $\beta = 12$ are found by taking $\mathcal{C}_1 = H_{18}, I_{18}$, respectively and $\mathcal{C}_2 = S_{18}$, [17], where

$$H_{18} = \begin{bmatrix}
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{bmatrix},$$

$$I_{18} = \begin{bmatrix}
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{bmatrix}$$

and

$$S_{18} = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \omega & 1 & 1 & \omega^2 & 1 & 1 & \omega & \omega^2 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \omega & \omega & 0 & \omega^2 & 0 & 1 & \omega^2 & \omega & \omega \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & \omega & 1 & 0 & \omega & \omega & \omega & \omega^2 & 0 & \omega^2 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & \omega & 1 & 0 & \omega & \omega & \omega & \omega^2 & \omega^2 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & \omega^2 & 1 & 1 & \omega & \omega & 1 & 1 & \omega^2 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \omega^2 & \omega & \omega & \omega & 0 & 1 & \omega & 0 & \omega^2 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \omega^2 & \omega & \omega & \omega & 0 & 1 & \omega & \omega^2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \omega & \omega^2 & 1 & 0 & \omega^2 & 0 & \omega & \omega & \omega \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \omega & 1 & 1 & \omega^2 & 1 & 1 & \omega & 1 & \omega^2
\end{bmatrix}$$

Our aim here is to find more codes for other β values, by taking $\mathcal{C}_1 = H_{18}$ and $\mathcal{C}_2 = \sigma(M_{18})$ where the matrix M_{18} is any self-dual quaternary code, but not extremal, (since the only extremal one is S_{18}, [17]) and σ is a suitable permutation, which is chosen by using Random of Magma.

We tried at least 20 different Hermitian self-dual quaternary codes for $\mathcal{C}_2$, we took $\mathcal{C}_1 = H_{18}$ and as a result we get 2 new inequivalent codes, both are with weight enumerator W_1 for $\beta = 12$ and $\beta = 15$. The codes $\sigma(M_{18})$ and $\sigma'(M'_{18})$ which we use for $\mathcal{C}_2$ and the corresponding new $[54, 27, 10]$ codes, and the permutations σ and σ' are as follows:

For the matrix M_{18}

$$\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \omega & 1 & 1 & \omega^2 & 1 & 1 & \omega & \omega^2 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \omega^2 & \omega^2 & 1 & 1 & \omega & \omega^2 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \omega & \omega^2 & \omega^2 & \omega^2 & \omega^2 & \omega & 1 & \omega \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \omega^2 & 1 & \omega & 1 & \omega^2 & \omega^2 & \omega^2 & \omega & \omega \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \omega & \omega & \omega^2 & \omega^2 & 0 & 0 & \omega & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & \omega^2 & 1 & 0 & \omega^2 & 1 & \omega \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \omega & 1 & 0 & 1 & 1 & \omega^2 & \omega & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \omega^2 & \omega & 0 & \omega & 1 & \omega & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \omega & 1 & 1 & \omega^2 & 1 & 1 & \omega & 1 & \omega^2
\end{bmatrix}$$

and for the permutation

$$\sigma = (1, 14, 13, 10)(2, 18, 4, 17, 11, 6, 16, 8, 7, 9, 12, 3, 15, 5),$$

we use $\mathcal{C}_1 = H_{18}$ where H_{18} is mentioned before, and $\mathcal{C}_2 = \sigma(M_{18})$, then we get the following $[54, 27, 10]$ code

[100000000000000000000000001001011000001110011010010101 0]
[010000000000000000000000001001101010010000010110001011 1]
[001000000000000000000000001001011111001001001001010100 0]
[000100000000000000000000001000001111110000111101001111 1]
[000010000000000000000000000011011011010100010101101000]
[000001000000000000000000001000101100100001000100011110 0]
[000000100000000000000000001001010100010011010100010010 0]
[000000010000000000000000000011010100011111011110101010]
[000000001000000000000000000010101110101001000101001011]
[000000000100000000000000000010111010010101001010111 0]
[000000000010000000000000000011000111100110000110111 11]
[000000000001000000000000000011111000001100100111011101]
[000000000000100000000000000010001110101010001110111011]
[000000000000010000000000001001111001001010110000000111]
[000000000000001000000000000010010110010110010110110 1]
[000000000000000100000000100011001010110100010111011 01]
[000000000000000010000000000000000010110100101010011 11]

[000000000000000001000000010001001110100101111011000000]
[000000000000000001000000000001100110010100011001010 01]
[000000000000000000010000010001000000000110011101011010]
[000000000000000000001000010010101101000111011101100010]
[000000000000000000000100000000011011111000111100011011]
[000000000000000000000010000001101110010100001010101 1100]
[000000000000000000000001000001001001111101000001000110]
[000000000000000000000000100010011001010110100001001100]
[000000000000000000000000010111010111111111010010000001]
[000000000000000000000000001001100000011001011101001 10]

with weight distribution $W = [\langle 0, 1\rangle, \langle 10, 255\rangle, \langle 12, 5319\rangle, \langle 14, 48876\rangle,$
$\langle 16, 313278\rangle, \langle 18, 1443468\rangle, \langle 20, 4791612\rangle, \langle 22, 11630505\rangle, \langle 24, 20897964\rangle,$
$\langle 26, 27977586\rangle, \ldots, \langle 54, 1\rangle]$
in other words, with W_1 and $\beta = 12$.

For the matrix M'_{18}

$$\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \omega^2 & 0 & \omega^2 & \omega^2 & 1 & \omega^2 & \omega^2 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \omega & 1 & \omega & \omega^2 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \omega^2 & \omega & 0 & 0 & 0 & 1 & \omega & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \omega & 0 & \omega^2 & \omega & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & \omega^2 & \omega^2 & 0 & 0 & \omega^2 & \omega^2 & 1 & \omega^2 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \omega & \omega^2 & 0 & \omega & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \omega & 1 & 0 & 0 & 0 & \omega & \omega^2 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & \omega^2 & \omega & 1 & \omega & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \omega^2 & \omega^2 & 1 & \omega^2 & \omega^2 & 0 & \omega^2 & 1
\end{bmatrix}$$

and for the permutation

$$\sigma' = (1, 6, 2, 14, 4, 8, 16, 7, 12, 18, 9, 3, 5, 15, 13, 17),$$

we use $\mathcal{C}_1 = H_{18}$ and $\mathcal{C}_2 = \sigma'(M'_{18})$, then we get the following $[54, 27, 10]$ code

[100000000000000000000000001001010100001010011111101000]
[010000000000000000000000001001101011010000101111111110]
[001000000000000000000000001100101110000100100110000100]
[000100000000000000000000001000111101111011010001011000]
[000010000000000000000000001000110001010111100001000001]
[000001000000000000000000001100101101010111011010101101]
[000000100000000000000000001001101001000000010101011010]
[000000010000000000000000001010110101111111010101010010]
[000000001000000000000000001101000100011001111010011110]
[000000000100000000000000001000101101100110100101011101]
[000000000010000000000000001101111111100101101010111111]
[000000000001000000000000001000110011010100010011100000]
[000000000000100000000000001011010000101010010100111110]
[000000000000010000000000000111110111111100101000100100]
[000000000000001000000000000101110000111010100010010001]
[000000000000000100000000001011111011010010001011011111]
[000000000000000010000000001000110100010101111011100000]
[000000000000000001000000001101010001111111010100001001]
[000000000000000000100000001000110101001010111000100000]
[000000000000000000010000000100001011010111111000010100001]
[000000000000000000001000001001100111110111001001010100]
[000000000000000000000100001011010111111100001010011000]
[000000000000000000000010011011110001010000000010011101]
[000000000000000000000001000011100110110101110110000011]
[000000000000000000000000101001010010110101001111100000]
[000000000000000000000000011001111110000110011001100]

with weight distribution $W' = [\langle 0, 1\rangle, \langle 10, 231\rangle, \langle 12, 5391\rangle, \langle 14, 48972\rangle,$
$\langle 16, 312798\rangle, \langle 18, 1443468\rangle, \langle 20, 4792956\rangle, \langle 22, 11629833\rangle, \langle 24, 20895948\rangle,$
$\langle 26, 27979266\rangle, \ldots, \langle 54, 1\rangle]$,
in other words, with W_1 and $\beta = 15$.

Recall that we say a binary code is of Type II if and only if it is self-dual and all its codewords have Hamming weights a multiple of 4.
For a binary ℓ-quasi-cyclic code of length 3ℓ, i.e., if $m = 3$, its binary component $\mathcal{C}_1$, means the component in the decomposition corresponding to the polynomial $Y - 1$. Also the quaternary component $\mathcal{C}_2$ of the code $\mathcal{C}$ corresponds $Y^2 + Y + 1$.

Proposition 6.0.7 [10] *A self-dual binary code $\mathcal{C}$ is a Type II ℓ-quasi-cyclic code of length 3ℓ if and only if its binary component $\mathcal{C}_1$ is of Type II.*

Proof.
($\Rightarrow$) Taking $a = b = 0$ in the $(x + a \mid x + b \mid x + a + b)$ construction, it is seen that $(x \mid x \mid x) \in \mathcal{C}$ for all $x \in \mathcal{C}_1$. Thus, $\mathcal{C}_1$ is Type II.
($\Leftarrow$) The weight of $(a \mid b \mid a + b)$ is twice the Hamming weight of $(a + \omega b)$, where $\omega^2 + \omega + 1 = 0$. From the Hermitian self-duality of $\mathcal{C}_2$, it follows that the Hamming weight of $(a + \omega b)$ is even. Hence the weight of $(a \mid b \mid a + b)$ is a multiple of 4. ■

Remark 6.0.8 *These* $[54, 27, 10]$ *codes are of Type II 18-quasi-cyclic self-dual codes of length 54 since their binary component H_{18} is of Type II and self-dual with respect to Euclidean inner product, by the previous proposition.*

CHAPTER 7

CONCLUSION

In this thesis, we have studied ℓ-quasi-cyclic self-dual codes over the field $\mathbb{F}_2$. Before this special type of linear codes, we gave basic information about linear codes and cyclic codes.

We showed the one-to-one correspondence between quasi-cyclic codes over a field $\mathbb{F}$ of index ℓ with length ℓm and linear codes over an auxiliary ring $\mathcal{R}$ of length ℓ. By Chinese Remainder Theorem, we decomposed that ring into a direct product of fields. That ring decomposition provides a code construction from codes of lower lengths. We used the cubic construction ($x+a \mid x+b \mid x+a+b$) among all constructions to find more new codes, since it is easy to implement and also it is proven that all codes can be found by this one.

By this construction, with suitable choices of binary and quaternary codes of length 18, we found two more $[54, 27, 10]$ codes than previously known. We gave all information about the construction step-by-step.

Appendix A

ALGORITHM

In this work, to construct new code, the MAGMA Computational Algebra System is used. We use ($x + a \mid x + b \mid x + a + b$) construction method in [10]. Below is the algorithm.

The following binary code is used for $\mathcal{C}_1$ from [17] :

$$\begin{bmatrix}
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{bmatrix}$$

And the following two quaternary codes are used for $\mathcal{C}_2$ from [12] :

$$\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \omega & 1 & 1 & \omega^2 & 1 & 1 & \omega & \omega^2 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \omega^2 & \omega^2 & 1 & 1 & \omega & \omega^2 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \omega & \omega^2 & \omega^2 & \omega^2 & \omega^2 & \omega & 1 & \omega \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \omega^2 & 1 & \omega & 1 & \omega^2 & \omega^2 & \omega^2 & \omega & \omega \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \omega & \omega & \omega^2 & \omega^2 & 0 & 0 & \omega & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & \omega^2 & 1 & 0 & \omega^2 & 1 & \omega \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \omega & 1 & 0 & 1 & 1 & \omega^2 & \omega & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \omega^2 & \omega & 0 & \omega & 1 & \omega & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \omega & 1 & 1 & \omega^2 & 1 & 1 & \omega & 1 & \omega^2
\end{bmatrix}$$

and

$$\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \omega^2 & 0 & \omega^2 & \omega^2 & 1 & \omega^2 & \omega^2 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \omega & 1 & \omega & \omega^2 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \omega^2 & \omega & 0 & 0 & 0 & 1 & \omega & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \omega & 0 & \omega^2 & \omega & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & \omega^2 & \omega^2 & 0 & 0 & \omega^2 & \omega^2 & 1 & \omega^2 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \omega & \omega^2 & 0 & \omega & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \omega & 1 & 0 & 0 & 0 & \omega & \omega^2 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & \omega^2 & \omega & 1 & \omega & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \omega^2 & \omega^2 & 1 & \omega^2 & \omega^2 & 0 & \omega^2 & 1
\end{bmatrix}.$$

We want to have a bigger matrix whose first 9 rows are rows of the generator matrix GM of $\mathcal{C}_2$, and the second 9 rows are the rows of the matrix GM*ω and the last 9 rows are from GM*ω.

Algorithm 1 Multiplying $\mathcal{C}_2$ by ω and ω^2

A 9×18 generator matrix of $\mathcal{C}_2$: GM := GeneratorMatrix($\mathcal{C}_2$) A 27×18 matrix GMlong$i$ in $[1 \dots 9]$ temp1$[i]$ $\leftarrow$ ω*GM$[i]$ temp2$[i]$ $\leftarrow$ ω^2*GM$[i]$ GMlong$[i]$ $\leftarrow$GM$[i]$ GMlong$[i+9]$ $\leftarrow$temp1$[i]$ GMlong$[i+18]$ $\leftarrow$temp2$[i]$

We use the following function to separate a quaternary matrix into two binary matrices such that $A + \omega B = C$ for binary matrices A and B, quaternary matrix

C.

```
function AF4toF2(codeword)
V := VectorSpace(GF(2),18);
ets := [ElementToSequence(codeword[i]) : i in [1..18]];
return V![ets[i][1] : i in [1..18]];
end function;
function BF4toF2(codeword)
V := VectorSpace(GF(2),18);
ets := [ElementToSequence(codeword[i]) : i in [1..18]];
return V![ets[i][2] : i in [1..18]];
end function;
```

where codeword is the rows of matrices.

Algorithm 2 Separation of the matrix GMlong as $A + \omega B$ =GMlong

The 27×18 matrix GMlong The 27×18 matrices A, B where $A + \omega B$ =GMlong$i$ in $[1 \dots 27]$ $A[i] \leftarrow A\mathbb{F}_4 to \mathbb{F}_2(\text{GMlong}[i])$ $B[i] \leftarrow B\mathbb{F}_4 to \mathbb{F}_2(\text{GMlong}[i])$

We use $\sigma(A)$ and $\sigma(B)$ with some permutations found by using the function *Random of Magma*.

Algorithm 3 Permutation on the matrices A and B

The matrices A and B, any permutation σ from $Random(Sym(18))$ The two permuted matrices $\sigma(A)$ and $\sigma(B)k$ in $[1 \dots 27]$ $\sigma(A)[\text{k}] \leftarrow A[\text{k}]^\sigma$ $\sigma(B)[\text{k}] \leftarrow B[\text{k}]^\sigma$

The($x+a \mid x+b \mid x+a+b$) construction is done with the following algorithm.

Algorithm 4 The ($x+a \mid x+b \mid x+a+b$) construction

The generator matrix of $\mathcal{C}_1$, GM2 and $\sigma(A), \sigma(B)$A generator matrix G of the code constructed by cubic construction i in $[1 \dots 9]$ j in $[1 \dots 27]$ m in $[1 \dots 18]$ $G[27*(i-1)+j, m] \leftarrow GM2[i,m] + \sigma(A)[j,m]$ $G[27*(i-1)+j, m+18] \leftarrow GM2[i,m] + \sigma(B)[j,m]$ $G[27*(i-1)+j, m+36] \leftarrow GM2[i,m] + \sigma(A)[j,m] + \sigma(B)[j,m]$

With the function of Magma *LinearCode(G)*, we get the code $\mathcal{C}$. Then with the function *MinimumWeight($\mathcal{C}$)*, we learn minimum weight, so minimum distance. We searched the codes with minimum distance 10 among all codes. And among this codes, we tried to find some codes with weight enumerator W_1 and $\beta \neq 0, 3, 6, 9$ and with W_2 and $\beta \neq 12, 15, 18$ since they are found before [5]. Consequently, after this process we found two codes with weight enumerator W_1 and $\beta = 12$ and 15. They are given in Chapter 6.

Appendix B

SOME ALGEBRA

B.1 Group

Definition B.1.1 [9] *A group is a set G together with a binary operation $*$ on G such that the following three properties hold:*
1) $*$ is associative; that is, for all a, b, c in G,

$$a * (b * c) = (a * b) * c;$$

2) There is an identity element e in G such that for all a in G

$$a * e = e * a = a;$$

3) For each $a \in G$, there exists an inverse element $ga^{-1} \in G$ such that

$$a * a^{-1} = a^{-1} * a = e.$$

If the group also satisfies
4) For all $a, b \in G$,

$$a * b = b * a,$$

then the group is called abelian (or commutative).

B.2 Ring

Definition B.2.1 [9] *A ring $(\mathcal{R}, +, \cdot)$ is a set $\mathcal{R}$, together with two binary operations, denoted by $+$ and $\cdot$ such that:*

1) $\mathcal{R}$ *is an abelian group with respect to* $+$.

2) $\cdot$ *is associative-that is,* $(a \cdot b) \cdot c = a \cdot (b \cdot c)$ *for all* $a, b, c \in \mathcal{R}$.

3) The distributive laws hold; that is, for all $a, b, c \in \mathcal{R}$ *we have* $a \cdot (b+c) = a \cdot b + a \cdot c$ *and* $(b + c) \cdot a = b \cdot a + c \cdot a$.

Definition B.2.2 [9]

1) A ring is called a ring with identity if the ring has multiplicative identity-that is, if there is an element e *such that* $ae = ea = a$ *for all* $a \in \mathcal{R}$.

2) A ring is called commutative if $\cdot$ *is commutative.*

3) A ring is called an integral domain if it is commutative ring with identity $e \neq 0$ *in which* $ab = 0$ *implies* $a = 0$ *or* $b = 0$.

4) A ring is called a division ring (or skew field) if the nonzero elements of $\mathcal{R}$ *for a group under* $\cdot$.

5) A commutative division ring is called a field.

Definition B.2.3 [9] *A subset* S *of a ring* $\mathcal{R}$ *is called a subring of* $\mathcal{R}$ *provided* S *is closed under* $+$ *and* $\cdot$ *and forms a ring under these operations.*

B.3 Ideal

Definition B.3.1 [9] *A subset* J *of a ring* $\mathcal{R}$ *is called an ideal provided* J *is a subring of* $\mathcal{R}$ *and for all* $a \in J$ *and* $r \in \mathcal{R}$ *we have* $ar \in J$ *and* $ra \in J$.

REFERENCES

[1] J. Baylis, *Error Correcting Codes A Mathematical Introduction*, Chapman and Hall Mathematics, 1998.

[2] R. E. Blahut, *Theory and Practice of Error Control Codes*, Addison-Wesley Publishing Company, 1984.

[3] A. Bonnecaze, A.D. Bracco, S.T. Dougherty, L.R. Nochefranca, P. Solé, *Cubic self-dual binary codes*, IEEE Trans. Inform. Theory., vol. 49, no. 9, pp. 2253-2259, Sep. 2003.

[4] S. Bouyuklieva, N. Yankov, J.-L. Kim, *Classification of binary self-dual* $[48, 24, 10]$ *codes with an automorphism of odd prime order*, Finite Fields and Their Appl., vol. 18, no. 6, pp. 1104-1113, 2012

[5] S. Han, J.-L. Kim, H. Lee and Y. Lee, *Construction of quasi-cyclic self-dual codes*, Finite Fields and Their Appl., vol. 18, no. 3, pp. 613-633, 2012.

[6] R. Hill, *A First Course in Coding Theory*, Clarendon Press, Oxford, 1986.

[7] W.C. Huffman, V. Pless, *Fundamentals of Error-correcting Codes*, Cambridge University Press, Cambridge, 2003.

[8] J.-L. Kim, Y. Lee, *Euclidean and Hermitian self-dual MDS codes over large finite fields*, J. Combin. Theory Ser. A. vol. 105, pp. 79-95, 2004.

[9] R. Lidl, H. Niederreiter, *Finite Fields*, Addison-Wesley Publishing Company, 1983

[10] S. Ling, P. Solé, *On the algebraic structure of quasi-cyclic codes I, Finite fields* IEEE Trans. Inform. Theory. vol. 47, pp. 2751-2760, 2001.

[11] F.J. MacWilliams, N.J.A. Sloane, *The Theory of Error-Correcting Codes*, Amsterdam, The Netherlands, North-Holland, 1977.

[12] A.Munemasa, http://www.math.is.tohoku.ac.jp/∼munemasa/research/codes/sd2.html

[13] http://www.coursehero.com/file/6259603/CHAPTER-03-Cyclic-codes.pdf

[14] http://www-math.ucdenver.edu/ wcherowi/courses/m5410/codingintro.pdf

[15] http://www.cs.cmu.edu/ venkatg/teaching/codingtheory/notes/notes1.pdf

[16] http://www.neng.usu.edu/classes/ece/7670/lecture5.pdf

[17] V. Pless, *A classification of self-orthogonal codes over GF(2)*, Discrete Math., vol. 3, pp. 209-246, 1972.

[18] E. Rains and N.J.A. Sloane, *Self-dual codes*, in *Handbook of Coding Theory*, V.S. Pless and W.C. Huffman, Eds. Amsterdam, The Netherlands: Elsevier, 1998.