5-2015

Quaternary Affine-Invariant Codes

Badria H Omar Salih

UAEU

جامعة الإمارات العربيـة المتحدة
United Arab Emirates University

United Arab Emirates University

College of Science

Department of Mathematical Sciences

QUATERNARY AFFINE-INVARIANT CODES

Badria H. Omar Salih

This thesis is submitted in partial fulfillment of the requirements
for the degree of Master of Science in Mathematics

Under the Supervision of Dr. Kanat Abdukhalikov

May 2015

Approval of the Master Thesis

This Master Thesis is approved by the following Examining Committee Members:

1) Advisor (Committee Chair): Dr. Kanat Abdukhalikov

   Title: Associate Professor

   Department of Mathematical Sciences

   College of Science

   Signature Abdukhalikov Date 29. 04. 15

2) Member: Dr. Adama Diene

   Title: Associate Professor

   Department of Mathematical Sciences

   College of Science

   Signature Adama Diene Date 29/04/2015

3) Member (External Examiner): Dr. Taher Abualrub

   Title: Professor

   Department of Mathematics and Statistics

   Institution: American University of Sharjah

   Signature [illegible] Date 29/4/2015

Title and Abstract in Arabic

الرُموز الرُباعَيّة الافينية الثَابتة

تَهتَمُ هٰذِهِ أَلأُطرُوحَة بِالرُموز الدَورِيّة المُوَسّعَة. مَوضُوع أَلأُطرُوحَة هُو التَحدِيد الكَامِل للرُموز الثُنَائِيّة وَالرُموز الرُبَاعَيّة الأَفينية الثَابتة ذَات الأَبعَاد الصَغرَى. الرُموز الدَورِيّة المُوَسّعَة تَتِمُ دِرَاسَتُهَا بِطرق الحلقَات الزُمَرِيّة. الرُموز الأَفينية الثَابتة يتمّ تَحدِيدهَا بمجمُوعَاتهَا التّعرِيفية. نقُوم بتقدِيم النتَائِج بتعدَاد المجمُوعَات التّعرِيفية. نعطِي تَحدِيدًا كَاملًا للرُموز الأَفينية الثَابتة لِلَابعَاد الصَغرَى.

Acknowledgments

I would like to thank my supervisor, Dr. Kanat Abdukhalikov who introduced me to the exciting field of codes, for his great support, help and patience with my lack of knowledge. I would like to thank my committee for their guidance, support, and assistance throughout my preparation.

I would like to thank the chair and all members of the Department of Mathematical Sciences at the United Arab Emirates University for assisting me all over my studies and research.

Special thanks go to my parents, my husband, and my children who helped me along the way.

Dedication

To my beloved parents, husband and children

Table of Contents

List of Tables

Chapter 1: Introduction

Let p be a prime number. Consider cyclic codes of length $p^n - 1$ (primitive cyclic codes) over a ring F, where F is either a field of characteristic p or the ring $\mathbb{Z}_{p^e}$ (integers modulo p^e). The extended code is defined by adding an overall parity check. The permutation group of the extended cyclic code contains the cyclic group $GL_1(p^n) \cong \mathbb{F}^*_{p^n}$ by definition. Let $n = mt$. Then the extended cyclic code may admit the affine group $G_m = AGL_m(p^t)$ as a permutation group. The affine group $AGL_m(p^t)$ is defined in the following way. Let $V = \mathbb{F}^+_{p^n}$ be the additive group of a finite field $\mathbb{F}_{p^n}$, and $\mathbb{F}_{p^t}$ be a subfield of $\mathbb{F}_{p^n}$, $n = mt$. Consider V as a vector space over $\mathbb{F}_{p^t}$ of dimension m. Then $G_m = AGL_m(p^t) = V \cdot GL_m(p^t)$.

Codes invariant under the group G_1 are called affine-invariant codes. Note that all groups G_m contain the group G_1. Our goal is to get full classification of binary and quaternary extended cyclic codes that are invariant under the group G_1 for small dimensions.

The problem is well studied in case of codes over a field. Extended cyclic codes of length p^n that are invariant under the affine group $G_1 = AGL_1(p^n)$ were characterized by Kasami, Lin and Peterson [16]. The general case is studied by Delsarte [11]. He gave a necessary and sufficient condition for extended cyclic codes of length p^n to be invariant under G_m. Berger and Charpin [9] reformulated this condition in other terms and used it to calculate permutation groups of some codes. These results are also described in the Handbook of Coding Theory [14].

In [3, 5] it is given one more necessary and sufficient condition for codes to be invariant under G_m. Of course, it is equivalent to conditions of Delsarte, but it looks much simpler. Moreover, this approach is generalized and applied to codes over $\mathbb{Z}_{p^e}$ (the ring of integers modulo p^e). There is growing interest in such codes. In particular, it turns out that codes over $\mathbb{Z}_4$ give a systematic way of constructing very good binary codes. For example, the famous Kerdock and Preparata codes are nonlinear binary codes that contain more codewords than any comparable linear codes presently known. Hammons et al. [13] showed that the Kerdock and Preparata codes can be very simply constructed as binary images under a certain map, called the Gray map, of linear codes over $\mathbb{Z}_4$. The Kerdock and Preparata codes, considered as codes over $\mathbb{Z}_4$, are analogs of the classical Reed-Muller codes: they have length 2^n and they are invariant under the affine group $G_1 = AGL_1(2^n)$. Recently connections of these codes to mutually unbiased bases, association schemes, spherical configurations, etc. were discovered (see, for example, [4]). In case of $e \geq 2$ there are only a few more works: the paper [10] deals with the case $m = 1$, $e = 2$, and the paper [12] deals with the case $m = 1$, and $e = 2$ or $p = 2$, with different approaches. On the other hand, case of nonprimitive cyclic codes is considered in [6, 7, 8].

Despite the fact that in the general case there are necessary and sufficient conditions for codes to be invariant under G_m, there are no appropriate enumerations of such codes. Two important extremal cases, $m = 1$ and $m = n$, were studied in detail in [1, 2]. In the current paper we present a full description of binary and quaternary affine-invariant codes for $n \leq 7$.

The work is organized as follows. In Chapters 2 - 4 we recall basic notions and theorems from coding theory [15, 17, 18, 19]. In Chapter 5 we consider binary and quaternary affine-invariant codes and present our results.

Chapter 2: Basic properties of codes

Let $F = \{a_1, a_2, \cdots, a_q\}$ be a set of size q, which we consider as a code alphabet and whose elements are called code symbols. In our thesis F mostly will be a finite field $\mathbb{F}_q$ of order q elements or ring $\mathbb{Z}_4$ of integers modulo 4.

A q-ary word of length n over F is a sequence $w = w_1 w_2 \cdots w_n$ with each $w_i \in F$ for all i. Sequence w may also be considered as the vector $(w_1, \cdots, w_n)$. A q-ary block code of length n over F is a nonempty set C of q-ary words with the same length n. An element of C is called a codeword in C. The number $|C|$ of codewords of C is called the size of C.

A code of length n and size k is called an (n, k)-code. The number $(\log_q |C|)/n$ is called the information rate of a code C of length n.

Codes over $\mathbb{F}_2$ and $\mathbb{Z}_4$ are called binary and quaternary codes, respectively.

2.1 Hamming distance

Let $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n)$ be n-tuples. The Hamming distance or distance $d(x, y)$, between x and y is the number of coordinates in which x and y differ, so the distance between two codewords is the minimum number of transmission errors required to change one codeword into the other. The minimum distance for a code d_{min} is the minimum of all distances $d(x, y)$, where x and y are distinct codewords. We have

$$d(x, y) = d(x_1, y_1) + \ldots + d(x_n, y_n),$$

where x_i and y_i are considered as words of length 1, and

$$d(x_i, y_i) = \begin{cases} 1, & x_i \neq y_i, \\ 0, & x_i = y_i. \end{cases} \tag{2.1}$$

Example. 1. Let $x = 0101$, $y = 1101$, $z = 1000$, over $\mathbb{F}_2$. Then

$$d(x, y) = 1,$$

$$d(y, z) = 2,$$

$$d(z, x) = 3.$$

2. Let $x = 123$, $y = 132$, $z = 321$ over $\mathbb{Z}_4$. Then

$$d(x, y) = 2,$$

$$d(y, z) = 3,$$

$$d(z, x) = 2.$$

Proposition 2.1.1 *Let x, y, z be words of length n over F. Then we have*

1. $0 \leq d(x, y) \leq n$,
2. $d(x, y) = 0$ *if and only if* $x = y$,
3. $d(x, y) = d(y, x)$,
4. $d(x, z) \leq d(x, y) + d(y, z)$ *(triangle inequality).*

Therefore, F^n is a metric space with respect to distance d.

2.2 Nearest neighbour decoding

Suppose that codewords from a code C are being sent over a communication channel. If a word x is received, the nearest neighbour decoding rule (or minimum distance decoding rule) will decode x to c_x if $d(x, c_x)$ is minimal among all the codewords in C, i.e.,

$$d(x, c_x) = \min \{d(x, c), c \in C\}$$

2.3 Distance of a code

Apart from the length and size of a code, another important and useful characteristic of a code is its distance. For a code C containing at least two words, the (minimum) distance of C, denoted by $d(C)$, is

$$d(C) = \min\{d(x,y) : x, y \in C, x \neq y\}.$$

A code of length n, size M and distance d is called an (n, M, d)-code. The numbers n, M and d are called the parameters of the code.

Example. Let $C = \{00000, 00101, 11100\}$ be a binary code. Then $d(C) = 2$ since

$$d(00000, 00101) = 2,$$

$$d(00000, 11100) = 3,$$

$$d(00101, 11100) = 3.$$

Therefore, C is a binary $(5, 3, 2)$-code.

It turns out that the distance of a code is related to the error detecting and error-correcting capabilities of the code.

Let r be a positive integer. A code C is r-error-detecting if, whenever a codeword incurs at least one but at most r errors, the resulting word is not a codeword. A code C is exactly r-error-detecting if it is r-error-detecting but not $(r+1)$-error-detecting.

Example. 1. The binary code $C = \{00000, 00101, 11011\}$ is 1-error detecting since changing any codeword in one position does not result in another codeword. In other words,

$00000 \to 00101$ needs to change two bits

$00000 \to 11011$ needs to change four bits

$00101 \to 11011$ needs to change four bits

In fact, C is exactly 1-error-detecting, as changing the third and fifth position of 00000 will result in another codeword 00101. So C is not a 2-error-detecting code.

2. The ternary code $C = \{00000000, 00001111, 11112222\}$ is 3-error-detecting since changing any codeword in one or two or three positions does not result in another codeword. In other words,

00000000 $\rightarrow$ 00001111 needs to change four positions;

00000000 $\rightarrow$ 11112222 needs to change eight positions;

00001111 $\rightarrow$ 11112222 needs to change eight positions.

In fact, C is exactly 3-error-detecting, as changing each of the last four positions of 00000000 to 1 will result in the codeword 00001111 (so C is not 4-error-detecting).

Theorem 2.3.1 *A code C is r-error-detecting if and only if $d(C) \geq r + 1$, that is, a code with distance d is an exactly $(d - 1)$-error-detecting code.*

Proof: If $d(C) < r + 1$, that is, $d(C) \leq r$, then there exist $c_1, c_2 \in C$ such that $1 \leq d(c_1, c_2) = d(C) \leq r$. Therefore for the codeword c_1 it is possible that $d(C) \leq r$ errors produce a codeword $c_2 \in C$. Hence, C is not an r-error-detecting code.

Conversely, let $d(C) \geq r + 1$. If $c \in C$ and x are such that $1 \leq d(c, x) \leq r < d(C)$, then $x \notin C$, hence, C is r-error-detecting. $\square$

Let r be a positive integer. A code C is r-error-correcting if minimum distance decoding is able to correct r or fewer errors, assuming that the incomplete decoding rule is used. A code C is exactly r-error-correcting if it is r-error-correcting but not $(r + 1)$-error-correcting.

Theorem 2.3.2 *A code C is r-error-correcting if and only if $d(C) \geq 2r + 1$, that is, a code with distance d is an exactly $[(d - 1)/2]$-error-correcting code. Here, $[r]$ is the greatest integer less than or equal to r.*

Proof: Suppose that C is r-error-correcting. If $d(C) < 2r + 1$, then there are distinct codewords $c, c' \in C$ such that $d(c, c') = d(C) \leq 2r$. Then there exists word c'' such that $d(c, c'') \leq r$ and $d(c', c'') \leq r$. Therefore, the word c'' cannot be decoded uniquely.

Conversely, suppose that $d(C) \geq 2r+1$. Let c be the codeword sent and let x be the word received. If r or fewer errors occur in the transmission, then $d(x,c) \leq r$. Hence, for any codeword $c' \in C$, $c' \neq c$, we have

$$d(x,c') \geq d(c,c') - d(x,c) \geq 2r+1-r = r+1 > d(x,c).$$

Therefore, x will be decoded (correctly) to c by using the minimum distance decoding rule. Hence C is r-error-correcting. $\square$

Chapter 3: Linear codes

A linear code of length n over the finite field $\mathbb{F}_q$ is simply a subspace of the vector space $\mathbb{F}_q^n$. Since linear codes are vector spaces, their algebraic structures often make them easier to describe and use than nonlinear codes.

3.1 Vector spaces over finite fields

Let $\mathbb{F}_q$ be the finite field of order q. A nonempty set V, together with addition operation, and scalar multiplication by elements of $\mathbb{F}_q$, is called a vector space (or linear space) over $\mathbb{F}_q$ if it satisfies the following conditions: for all $u, v, w \in V$ and for all $\lambda, \mu \in \mathbb{F}_q$:

1. $u + v \in V$;
2. $(u + v) + w = u + (v + w)$;
3. there is an element $0 \in V$ with the property $0 + v = v = v + 0$ for all $v \in V$;
4. for each $u \in V$ there is an element of V, called $-u$, such that $u + (-u) = 0 = (-u) + u$;
5. $u + v = v + u$;
6. $\lambda v \in V$;
7. $\lambda(u + v) = \lambda u + \lambda v, (\lambda + \mu)u = \lambda u + \mu u$;
8. $(\lambda\mu)\, u = \lambda\, (\mu u)$;
9. if 1 is the multiplicative identity of $\mathbb{F}_q$, then $1u = u$.

Example. It is easy to verify that the following are vector spaces over the field $\mathbb{F}_q$:

1. $C_1 = \mathbb{F}_q^n$ and $C_2 = \{0\}$.

2. $C_3 = \{(a, ..., a) : a \in \mathbb{F}_q\}$.

3. $C_4 = \{(0, 0, 0, 0), (1, 0, 1, 0), (0, 1, 0, 1), (1, 1, 1, 1)\}$, $q = 2$.

4. $C_5 = \{(0, 0, 0), (0, 1, 2), (0, 2, 1)\}$, $q = 3$.

A nonempty subset C of a vector space V is called a subspace of V if it is itself a vector space with the same vector addition and scalar multiplication as the space V.

Let V be a vector space over $\mathbb{F}_q$. A linear combination of $v_1, \ldots, v_r \in V$ is a vector of the form $\lambda_1 v_1 + \ldots + \lambda_r v_r$, where $\lambda_1, \ldots, \lambda_r \in \mathbb{F}_q$ are some scalars.

Let V be a vector space over $\mathbb{F}_q$. A set of vectors $\{v_1, \ldots, v_r\}$ in V is called linearly independent if

$$\lambda_1 v_1 + \ldots + \lambda_r v_r = 0 \Rightarrow \lambda_1 = \ldots = \lambda_r = 0$$

The set is linearly dependent if it is not linearly independent; i.e., if there are $\lambda_1, \ldots, \lambda_r \in \mathbb{F}_q$, not all zero, such that $\lambda_1 v_1 + \ldots + \lambda_r v_r = 0$.

If $S = \{v_1, \ldots, v_r\}$ is a collection of vectors, then the set $\langle S \rangle$ generated by S is the set of all linear combinations of $v_1, \ldots, v_r$.

3.2 Linear codes

A linear code C of length n over $\mathbb{F}_q$ is a subspace of $\mathbb{F}_q^n$. We consider below some examples.

1. $C = \{(\lambda, \lambda, \ldots, \lambda) : \lambda \in \mathbb{F}_q\}$. This code is called a repetition code.

2. $C = \{000, 100, 010, 110\}$, $q = 2$.

3. $C = \{0000, 1010, 2020, 2200, 1100, 0210, 2110, 1220, 0120\}$, $q = 3$.

Let C be a linear code in $\mathbb{F}_q^n$. For $x = (x_1, \ldots, x_n) \in \mathbb{F}_q^n$ and $y = (y_1, \ldots, y_n) \in \mathbb{F}_q^n$ we define inner product

$$x \cdot y = x_1 y_1 + \ldots + x_n y_n.$$

The dual code of C is $C^{\perp}$, the orthogonal complement of the subspace C of $\mathbb{F}_q^n$:

$$C^{\perp} = \left\{x \in \mathbb{F}_q^n \mid x \cdot y = 0 \text{ for all } c \in C\right\}$$

Dimension of the linear code C is the dimension of C as a vector space over $\mathbb{F}_q$.

Theorem 3.2.1 *Let C be a linear code of length n over $\mathbb{F}_q$. Then,*

1. $|C| = q^{\dim(C)}$ *i.e.*, $\dim(C) = \log_q |C|$.

2. $C^{\perp}$ *is a linear code and* $\dim(C) + \dim\left(C^{\perp}\right) = n$.

3. $\left(C^{\perp}\right)^{\perp} = C$.

A linear code C of length n and dimension k over $\mathbb{F}_q$ is often called a q-ary $[n, k]$-code or, if q is clear from the context, an $[n, k]$-code. It is also an $\left(n, q^k\right)$-linear code. If the distance d of C is known, it is also called an $[n, k, d]$-linear code.

Let C be a linear code. Code C is self-orthogonal if $C \subseteq C^{\perp}$, and C is self-dual if $C = C^{\perp}$. The dimension of a self-orthogonal code of length n must be $\leq n/2$, and the dimension of a self-dual code of length n is $n/2$.

3.3 Hamming weight

Let x be a word in $\mathbb{F}_q^n$. The Hamming weight of x, denoted by $wt(x)$, is defined to be the number of nonzero coordinates in x; that is, $wt(x) = d(x, 0)$, where 0 is the zero word.

For every element $x \in \mathbb{F}_q$, we can define the Hamming weight as follows

$$wt(x) = d(x, 0) = \begin{cases} 1, & x \neq 0, \\ 0, & x = 0. \end{cases} \tag{3.1}$$

Then, writing $x \in \mathbb{F}_q^n$ as $x = (x_1, x_2, \ldots, x_n)$, the Hamming weight of x can also be equivalently defined as $wt(x) = wt(x_1) + wt(x_2) + \ldots + wt(x_n)$.

For any prime power q and $x, y \in \mathbb{F}_q^n$ we have

$$d(x, y) = wt(x - y),$$

$$wt(x) + wt(y) \geq wt(x + y) \geq wt(x) - wt(y).$$

Let C be a code (not necessarily linear). The minimum Hamming weight of C, denoted $wt(C)$, is the smallest of the weights of the nonzero codewords of C.

Theorem 3.3.1 *Let C be a linear code over $\mathbb{F}_q$. Then $d(C) = wt(C)$.*

Example. Consider the binary linear code $C = (0000, 1101, 0011, 1110\}$. We see that

$$wt(1101) = 3,$$

$$wt(0011) = 2,$$

$$wt(1110) = 3,$$

Hence, $d(C) = 2$.

3.4 Bases for linear codes

A linear code is a vector space, all its elements can be described in terms of a basis. Let A be a matrix over $\mathbb{F}_q$, an elementary row operation performed on A is any one of the following three operations:

1. Interchanging two rows,
2. Multiplying a row by a nonzero scalar,
3. Replacing a row by its sum with the scalar multiple of another row.

The following are well known facts from linear algebra:

1. Any matrix M over $\mathbb{F}_q$ can be put in row echelon form (REF) or reduced row echelon form (RREF) by a sequence of elementary row operations. In other words, a matrix is row equivalent to a matrix in REF or in RREF.

2. For a given matrix. its RREF is unique, but it may have different REFs.

   Recall that the difference between the RREF and the REF is that the leading nonzero entry of a row in the RREF is equal to 1 and it is the only nonzero entry in its column.

Algorithm 1 (for finding a basis of a linear code)

Input: A nonempty subset S of $\mathbb{F}_q$

Output: A basis for $C =< S >$, the linear code generated by S.

Description: Form the matrix A whose rows are the words in S. Use elementary row operations to find an REF of A. Then the nonzero rows of the REF form a basis for C.

Algorithm 2 (choosing basis elements from generating set)

Input: A nonempty subset S of $\mathbb{F}_q^n$.

Output: A basis for $C =< S >$, the linear code generated by S.

Description: Form the matrix A whose columns are the words in S. Use elementary row operations to put A in REF and locate the leading columns in the REF. Then the original columns of A corresponding to these leading columns form a basis for C.

Algorithm 3 (finding a basis of the dual code)

Input: A nonempty subset S of $\mathbb{F}_q^n$.

Output: A basis for the dual code $C^{\perp}$, where $C =< S >$.

Description: Form the matrix A whose rows are the words in S. Use elementary row operations to place A in RREF. Let G be the $k \times n$ matrix consisting of all the nonzero rows of the RREF:

$$A = \begin{pmatrix} G \\ 0 \end{pmatrix} \tag{3.2}$$

The matrix G contains k leading columns. Permute the columns of G to form

$$G' = (I_k \mid X),$$

where I_k denotes the $k \times k$ identity matrix. Form a matrix H' as follows:

$$H' = (-X^T \mid I_{n-k})$$

where X^T denotes the transpose of X. Apply the inverse of the permutation applied to the columns of G to the columns of H' to form H. Then the rows of H form a basis for $C^{\perp}$.

3.5 Generator matrix and parity-check matrix

In coding theory, a basis for a linear code is often represented in the form of a matrix, called a generator matrix, while a matrix that represents a basis for the dual code is called a parity-check matrix.

A generator matrix for a linear code C is a matrix G whose rows form a basis for C. A parity-check matrix H for a linear code C is a generator matrix for the dual code $C^{\perp}$.

1. If C is an $[n,k]$-linear code, then a generator matrix for C must be a $k \times n$ matrix and a parity-check matrix for C must be $(n-k) \times n$ matrix.

2. Algorithm 3 can be used to find generator and parity-check matrices for a linear code.

3. A permutation (different from the identity) of the rows of a generator matrix also leads to a different generator matrix.

4. The rows of a generator matrix are linearly independent. The same holds for the rows of a parity-check matrix. To show that a $k \times n$ matrix G is indeed a generator matrix for a given $[n, k]$-linear code C, it suffices to show that the rows of G are codewords in C and that they are linearly independent.

Alternatively, one may also show that C is contained in the row space of G. A generator matrix of the form $(I_k \mid X)$ is said to be in standard form. A parity-check matrix in the form $(Y \mid I_{n-k})$ is said to be in standard form.

Theorem 3.5.1 *Let C be an $[n, k]$*
Then $v \in \mathbb{F}_q^n$ belongs to $C^\perp$ if and only if v is orthogonal to every row of G, that is, $v \in C^\perp \Leftrightarrow vG^T = 0$. In particular, given an $(n-k) \times n$ matrix H, then H is a parity-check matrix for C if and only if the rows of H are linearly independent and $HG^T = 0$.

Theorem 3.5.2 *Let C be a linear code and let H be a parity-check matrix for C. Then*

1. *C has distance $\geqslant d$ if and only if any $d-1$ columns of H are linearly independent;*

2. *C has distance $\leqslant d$ if and only if H has d columns that are linearly dependent.*

Proof: Let $v = (v_1, \ldots, v_n) \in C$ be a word of weight $e > 0$. Suppose the nonzero coordinates are in the positions $i_1, \ldots, i_e$, so that $v_j = 0$ if $j \notin \{i_1, \ldots, i_e\}$. Let c_i $(1 \leq i \leq n)$ denote the ith column of H. By Theorem 3.5.1 code C contains a nonzero word $(v_1, \ldots, v_n)$ of weight e if and only if

$$0 = vH^T = v_{i_1} c_{i_1}^T + \ldots + v_{i_e} c_{i_e}^T,$$

this is true if and only if there are e columns of H that are linearly dependent.

To prove (1): the statement that the distance of C is $\geq d$ is equivalent to saying that C does not contain any nonzero word of weight $\leq d-1$, which is in turn equivalent to saying that any $d-1$ columns of H are linearly independent.

Similarly to prove (2): the statement that the distance of C is $\leq d$ is equivalent to saying that C contains a nonzero word of weight $\leq d$, which is in turn equivalent to saying that H has $\leq d$ columns that are linearly dependent. □

Corollary 3.5.3 *Let C be a linear code and let H be a parity-check matrix for C. Then the following statements are equivalent:*

1. *C has distance d;*

2. *any $d-1$ columns of H are linearly independent and H has d columns that are linearly dependent.*

Theorem 3.5.4 *If $G = (I_k \mid X)$ is the standard form generator matrix of an $[n,k]$-code C, then a parity-check matrix for C is $H = \left(-X^T \mid I_{n-k}\right)$.*

3.6 Encoding with a linear code

Let C be an $[n,k,d]$-linear code over the finite field $\mathbb{F}_q$. Each codeword of C can represent one piece of information, so C can represent q^k distinct pieces of information. Once a basis $\{r_1,\ldots,r_k\}$ is fixed for C, each codeword v, or, equivalently, each of the q^k pieces of information, can be uniquely written as a linear combination $v = u_1r_1 + \ldots + u_kr_k$, where $u_1,\ldots,u_k \in F_q$.

Equivalently, we may set G to be the generator matrix of C whose ith row is the vector r_i in the chosen basis. Given a vector $u = (u_1,\ldots,u_k) \in \mathbb{F}_q^k$, it is clear that $v = uG = u_1r_1 + \ldots + u_kr_k$ is a codeword in C. Conversely, any $v \in C$ can be written uniquely as $v = uG$, where $u = (u_1,...,u_k) \in \mathbb{F}_q^k$.

Hence, every word $u \in \mathbb{F}_q^k$ can be encoded as $v = uG$. The process of representing the elements u of $\mathbb{F}_q^k$ as codewords $v = uG$ in C is called encoding.

Example. Let C be the binary $[7,3]$-linear code with the generator matrix

$$G = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix} \tag{3.3}$$

then the message x = 110 is encoded as

$$v = xG = (110)\begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix} = 1110110 \qquad (3.4)$$

Remark: Some of the advantages of having the generator matrix of a linear code in standard form are as follows:

1. If a linear code C has a generator matrix G in standard form, $G = (I \mid X)$, then Algorithm 3 at once yields $H = \left(-X^T \mid I\right)$ as a parity-check matrix for C.

2. If an $[n, k, d]$- linear code C has a generator matrix G in standard form, $G = (I \mid X)$, then it is trivial to recover the message u from the codeword $v = uG$ since

$$v = uG = u(I \mid X) = (u, uX)$$

3.7 Decoding of linear codes

Let C be a linear code of length n over $\mathbb{F}_q$, and let $u \in \mathbb{F}_q^n$ be any vector of length n, we define the coset of C determined by u to be the set

$$C + u = \{v + u : v \in C\}.$$

We note that, under the vector addition, $\mathbb{F}_q^n$ is a finite abelian group, and a linear code C over $\mathbb{F}_q$ of length n is also a subgroup of $\mathbb{F}_q^n$. The coset of a linear code defined above coincides with the usual notion of a coset in group theory.

Example. Let $q = 2$ and $C = \{0000, 0101, 1010, 1111\}$

$$C + 1000 = \{1000, 1101, 0010, 0111\}$$

$$C + 0100 = \{0100, 0001, 1110, 1011\}$$

$$C + 0001 = \{0001, 0100, 1011, 1110\}$$

Theorem 3.7.1 *Let C be an $[n, k, d]$-linear code over the finite field $\mathbb{F}_q$. Then*

1. *every vector of $\mathbb{F}_q^n$ is contained in some coset of C;*
2. *for all $u \in \mathbb{F}_q^n$, $|C + u| = |C| = q^k$;*
3. *for all $u, v \in \mathbb{F}_q^n$, $u \in C + v$ implies that $C + u = C + v$;*
4. *two cosets are either identical or they have empty intersection,*
5. *there are q^{n-k} different cosets of C;*
6. *for all $u, v \in \mathbb{F}_q^n$, $u - v \in C$ if and only if u and v are in the same coset.*

Example. The cosets of the binary linear code $C = \{0000, 1101, 0011, 1110\}$ are as follows:

$$0000 + C = \{0000, 1101, 0011, 1110\}$$

$$1000 + C = \{1000, 0101, 1011, 0110\}$$

$$0100 + C = \{0100, 1001, 0111, 1010\}$$

$$0001 + C = \{0001, 1100, 0010, 1111\}$$

The above array is called a standard array. A word of the least Hamming weight in a coset is called a coset leader. In the previous example the first vectors in cosets are coset leaders.

3.8 Nearest neighbour decoding for linear codes

Let C be a linear code. Assume the codeword v is transmitted and the word w is received, then the error pattern (or error string) is $e = w - v \in w + C$. Therefore $w - e = v \in C$, so, by part (6) of Theorem above, the error pattern e and the received word w are in the same coset.

Since error patterns of small weight are the most likely to occur, nearest neighbour decoding works for a linear code C in the following manner. Upon receiving the word w, we choose a word e of least weight in the coset $w + C$ and conclude that $v = w - e$ was the codeword transmitted.

Example. Let $q = 2$ and $C = \{0000, 1101, 0011, 1110\}$. Decode the following received words: (a) $w = 1011$, (b) $w = 1111$. First, we write down the standard array of C

$$0000 + C = \{0000, 1101, 0011, 1110\}$$

$$1000 + C = \{1000, 0101, 1011, 0110\}$$

$$0100 + C = \{0100, 1001, 0111, 1010\}$$

$$0001 + C = \{0001, 1100, 0010, 1111\}$$

(a) $w = 1011$: $w + C$ is the second coset. The word of least weight in this coset is 1000 (note that this is the unique coset leader of this coset). Hence, $1011 - 1000 = 0011$ was the most likely codeword transmitted.

(b) $w = 1111$: $w + C$ is the fourth coset. There are two words of smallest weight, 0001 and 0010, in this coset. If we are doing incomplete decoding, we ask for a retransmission. If we are doing complete decoding, we arbitrarily choose one of the words of smallest weight, say 0001, to be the error pattern, and conclude that $1111 - 0001 = 1110$ was a most likely codeword sent.

3.9 Cyclic codes

The linear code C of length n is a cyclic code if it is invariant under a cyclic shift:

$$c = (c_0, c_1, c_2, \ldots c_{n-2}, c_{n-1}) \in C \Rightarrow c' = (c_{n-1}, c_0, c_1, c_2, \ldots c_{n-2}) \in C.$$

Let F be a field. With every codeword $c = (c_0, c_1, c_2, \ldots c_{n-2}, c_{n-1}) \in F^n$ we associate the polynomial

$$c(x) = c_0 + c_1x + c_2x^2 + \ldots + c_{n-2}x^{n-2} + c_{n-1}x^{n-1} \in F[x]/(x^n - 1).$$

If c is a codeword of the code C, then we call $c(x)$ the associated code polynomial. The shifted codeword c' has associated code polynomial

$$c'(x) = c_{n-1} + c_0x + c_1x^2 + c_2x^3 + \ldots + c_{n-2}x^{n-1} \in F[x]/(x^n - 1).$$

Thus $c'(x)$ is equal to the polynomial $xc(x)$ modulo $(x^n - 1)$. If C is closed under the cyclic shift maps $p(x) \to x^j p(x)$ then C is a cyclic code. Therefore a linear code $C \subseteq F[x]/(x^n - 1)$ is cyclic if and only if C is an ideal of the quotient ring $F[x]/(x^n - 1)$.

Theorem 3.9.1 *Let C be a cyclic $[n, k]$ code over a field F with $k > 0$. Then there is a unique monic polynomial $g(x)$ such that for every $c(x) \in F[x]$ with* $\deg c(x) < n$, *we have*

$$c(x) \in C \Longleftrightarrow g(x)|c(x).$$

Proof. First, if $g(x)$ exists then it must be a codeword of C (since obviously $g(x)|g(x)$ and it is unique (since it divides all other monic codewords in C).

Let $g(x)$ be a monic nonzero codeword with a smallest degree in C (if nonzero codeword with a smallest degree is not monic, we can make it monic multiplying by a constant). For every $u(x) \in F[x]$ we have $u(x)g(x) \bmod (x^n - 1) \in C$. In particular, for every $u(x) \in F[x]$, $\deg u(x) < n - \deg g(x)$, we have $u(x)g(x) \in C$. Therefore, all the polynomial multiples of $g(x)$ of degree less than n are codewords of C.

Let $c(x) \in C$ and write $c(x) = u(x)g(x) + r(x)$ where $\deg r(x) < \deg g(x)$. Since both $c(x)$ and $u(x)g(x)$ are in C then, $r(x) = c(x) - u(x)g(x) \in C$. From the minimality of $\deg g$ we get that $r(x) = 0$, i.e., $c(x)$ is divisible by $g(x)$. □

The polynomial $g(x)$ in Theorem 3.9.1 is called the *generator polynomial* of the cyclic code C.

Theorem 3.9.1 states that a cyclic $[n, k]$ code C can be written as

$$C = \{u(x)g(x) \mid u(x) \in F[x],\ \deg u(x) < n - \deg g(x)\}.$$

where $g(x)$ is the generator polynomial of C.

Theorem 3.9.2 *Let $f(x)$ be the generator polynomial of a cyclic $[n, k]$ code over a field F. Then*

$$g(x) | x^n - 1.$$

Proof. Write $x^n - 1 = h(x)g(x) + r(x)$, where $\deg r(x) < \deg g(x)$. We have

$$r(x) = -h(x)g(x) \bmod (x^n - 1)$$

and, from the property of cyclic codes, it follows that $r(x) \in C$. This means that $r(x) = 0$, since no other codeword in C can have degree smaller than $\deg g$. $\square$

We can also state a converse to Theorem 3.9.2 : if $g(x)$ is a polynomial over F that divides $x^n - 1$, then the set

$$C = \{u(x)g(x) \mid u(x) \in F[x],\ \deg u(x) < n - \deg g(x)\}$$

is a cyclic code .

Let C be a cyclic $[n, k]$ code with a generator polynomial $g(x)$. The *check polynomial* of C, denoted as $h(x)$, is the monic polynomial of degree k obtained by

$$h(x) = \frac{x^n - 1}{g(x)}.$$

Then

$$C = \{c(x) \mid \deg c(x) < n,\ c(x)h(x) \equiv 0 (\mathrm{mod}\, (x^n - 1))\}.$$

Theorem 3.9.3 *Let C be a cyclic $[n, k]$ code over a field F and let $h(x) = \sum_{j=0}^{k} h_j x^j$ be the check polynomial of C. Then the dual code of C is a cyclic $[n, n-k]$ code over*

F whose generator polynomial is

$$g^{\perp}(x) = \frac{1}{h_0}\sum_{j=0}^{k} h_{k-j}x^j = \frac{1}{h_0}x^k h(x^{-1}).$$

Therefore, $g^{\perp}(x)$ is a scaled reciprocal polynomial of $h(x)$.

Chapter 4: Quaternary linear codes

Let $\mathbb{Z}_4$ be the ring of integers modulo 4, n be a positive integer, and $\mathbb{Z}_4^n$ be the set of n-tuples over $\mathbb{Z}_4$, i.e.

$$\mathbb{Z}_4^n = \{(x_1, \ldots, x_n) | x_i \in \mathbb{Z}_4 \text{ for } i = 1, \ldots, n\}$$

Any non-empty subset C of $\mathbb{Z}_4^n$ is called a quaternary code or, simply and more precisely, a $\mathbb{Z}_4$-code or a code over $\mathbb{Z}_4$, and n is called the length of the code. The n-tuples in $\mathbb{Z}_4^n$ are called words and n-tuples in a quaternary code C are called codewords of C.

For all $(x_1, \ldots, x_n)$ and $(y_1, \ldots, y_n) \in \mathbb{Z}_4^n$ define a component-wise addition

$$(x_1, \ldots, x_n) + (y_1, \ldots, y_n) = (x_1 + y_1, \ldots, x_n + y_n),$$

then $\mathbb{Z}_4^n$ becomes an additive abelian group of order 4^n.

Any subgroup of $\mathbb{Z}_4^n$ is called a quaternary linear code, or simply, $\mathbb{Z}_4$ -linear code. For all $(x_1, \ldots, x_n)$ and $(y_1, \ldots, y_n)$ in $\mathbb{Z}_4^n$ define

$$(x_1, \ldots, x_1) \cdot (y_1, \ldots, y_n) = x_1 y_1 + \cdots + x_n y_n,$$

which is called the inner product of x and y.

Let C be a quaternary linear code of length n. Define

$$C^\perp = \{x \in \mathbb{Z}_4^n | x \cdot y = 0 \text{ for all } y \in C\}.$$

Then $C^\perp$ is a subgroup of $\mathbb{Z}_4^n$. Hence $C^\perp$ is also a quaternary linear code, called the dual code of C. If $C \subset C^\perp$, C is called a self orthogonal code. If $C = C^\perp$ then C is called a self-dual code. Two quaternary codes C_1 and C_2 both of length n are said to be equivalent, if one can be obtained from the other by permuting the coordinates. Quaternary codes differ only by a permutation of coordinates are said to be permutation-equivalent.

A quaternary cyclic code C of length n is a quaternary linear code C of length n with the property

$$(c_0, c_1, \ldots, c_{n-1}) \in C \Rightarrow (c_{n-1}, c_0, c_1, , c_{n-2}) \in C.$$

As in the binary case, we have a bijection

$$\mathbb{Z}_4^n \rightarrow \mathbb{Z}_4[X]/(X^n - 1),$$

$$(c_0, c_1, \ldots, c_{n-1}) \rightarrow (c_0 + c_1 X + \ldots + c_{n-1} X^{n-1})$$

A nonempty set of $\mathbb{Z}_4^n$ is a $\mathbb{Z}_4$-cyclic code if and only if the image under the above map is an ideal of the residue class ring $\mathbb{Z}_4[X]/(X^n - 1)$.

4.1 Generator Matrices

Let C be a $\mathbb{Z}_4$-linear code of length n. A $(k \times n)$-matrix G over $\mathbb{Z}_4$ is called a generator matrix of C if the rows of G generate C and no proper subset of the rows of G generates C.

Any $\mathbb{Z}_4$-linear code C containing some nonzero codewords is permutation-equivalent to a $\mathbb{Z}_4$-linear code with a generator matrix of the form

$$X = \begin{pmatrix} I_{k_1} & A & B \\ 0 & 2I_{k_2} & 2D \end{pmatrix} \tag{4.1}$$

where I_{k_1} and I_{k_2} denote the $k_1 \times k_1$ and $k_2 \times k_2$ identity matrices, respectively, A and D are Z_2-matrices, and B is a $\mathbb{Z}_4$-matrix. Then C is an abelian group of type $4^{k_1}2^{k_2}$, C contains $2^{2k_1+k_2}$ codewords, and C is a free $\mathbb{Z}_4$-module if and only if $k_2 = 0$.

4.2 The Gray Map and the Lee weight

The Gray map ψ is defined in the following way:

$$\psi : \mathbb{Z}_4 \to Z_2^2,$$

$$0 \to 00,$$

$$1 \to 01,$$

$$2 \to 11,$$

$$3 \to 10.$$

The map ψ is a bijection from $\mathbb{Z}_4$ to Z_2^2. But ψ is not an additive group homomorphism from $\mathbb{Z}_4$ to Z_2^2, since $\psi(x+y) \neq \psi(x) + \psi(y)$.

The Gray map is very related to the Lee weight. The Lee weights of 0, 1, 2, $3 \in \mathbb{Z}_4$ are defined as follows:

$$wt_L(0) = 0, \ wt_L(1) = 1, \ wt_L(2) = 2, \ wt_L(3) = 1.$$

Then the Lee weight $wt_L(x)$ of $x = (x_1, \ldots, x_n) \in \mathbb{Z}_4^n$ is the sum of the Lee weights of its components

$$wt_L(x) = \sum_{k=1}^{n} wt_L(x_k).$$

Lee distance is defined by

$$wt_L(x, y) = wt_L(x - y).$$

It is easy to see that Lee weight of a quaternary codeword is equal to Hamming (binary) weight of its Gray image.

4.3 The Quaternary Kerdock Codes

Let m be any integer ≥ 2 and $h(X)$ be a basic primitive polynomial of degree m over $\mathbb{Z}_4$ such that $h(X)|(X^{2^m-1}-1)$. The existence of such a polynomial $h(X)$ follows from the existence of the Hensel lift [19] of the binary primitive polynomial $h(X)$ of degree m. Let $n = 2^m - 1$ and $g(X)$ be the reciprocal polynomial to the polynomial $(X^n - 1)/(X-1)h(X)$. The shortened quaternary Kerdock code $K'(m)$ is the quaternary cyclic code of length $2^m - 1$ with generator polynomial $g(X)$. The positions of the coordinates of codewords of $K'(m)$ are numbered as $0, 1, 2, \ldots, 2^m - 2$.

The quaternary Kerdock code $K(m)$ is the code obtained from $K'(m)$ by adding a zero-sum check symbol to each codeword of $K'(m)$ at position ∞, which is situated in front of the position 0. When m is an odd integer ≥ 3, the binary image of $K(m)$ (under the Gray map) is the binary Kerdock code K_{m+1} of length 2^{m+1}.

4.4 The Quaternary Preparata Codes

Let m is an integer ≥ 2, $h(X)$ is a basic primitive polynomial of degree m dividing X^n-1 in $\mathbb{Z}_4[X]$, where $n = 2^m-1$, ξ is a root of $h(X)$ in a Galois ring $GR(4^n)$, and $g(X)$ is the reciprocal polynomial to the polynomial $(X^n-1)/(X-1)h(X)$. The $\mathbb{Z}_4$-cyclic code of length n with generator polynomial $h(X)$ is called the shortened quaternary Preparata code [19] and denoted by $P'(m)$. The $\mathbb{Z}_4$-linear code obtained from $P'(m)$ by adding a zero-sum check symbol to each codeword of $P'(m)$ is called the quaternary Preparata code and denoted by $P(m)$. Code $P'(m)$ has parity check matrix

$$\left(1, \xi, \xi^2, \ldots, \xi^{n-1}\right).$$

$P(m)$ is the dual code of $K(m)$ and has parity check matrix

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & \ldots & 1 \\ 0 & 1 & \xi & \xi^2 & \xi^3 & \ldots & \xi^{n-1} \end{pmatrix}$$

Chapter 5: Affine-invariant codes

Let $\mathbb{F}_q$ be a finite field of p^n elements, p be a prime number. An extended cyclic code of length p^n is called affine-invariant if it is invariant under the action of the affine group

$$G_1 = AGL_1(p^n) = V \cdot GL_1(p^n).$$

It is a semidirect product of the additive group $V = \mathbb{F}_q^+$ and the multiplicative group $GL_1(p^n) = \mathbb{F}_q^*$ of the field $\mathbb{F}_q$.

Let $A = F[V]$ be the group ring of the abelian group $V = \mathbb{F}_q^+$ over a ring F:

$$A = \left\{ \sum_{v \in V} a_v X^v \mid a_v \in F \right\}.$$

Operations in A are given by:

$$\sum a_v X^v + \sum b_v X^v = \sum (a_v + b_v) X^v,$$

$$c \sum a_v X^v = \sum c a_v X^v, \quad c \in F,$$

$$\left(\sum a_v X^v\right) \cdot \left(\sum b_v X^v\right) = \sum_{u,v} a_u b_v X^{u+v} = \sum_w \left(\sum_u a_u b_{w-u}\right) X^w.$$

The element X^0 is the unity of the group ring A and A is a module over F of rank p^n with basis $\{X^v \mid v \in V\}$.

We consider F-submodules of A as F-linear codes of length p^n based on the alphabet F.

The affine group $G_1 = AGL_1(p^n) = V \cdot GL_1(p^n)$ acts on A:

$$\widehat{u}(X^v) = X^{u+v}, \quad u \in V,$$

$$\widehat{g}(X^v) = X^{gv}, \quad g \in \mathbb{F}_q^*$$

G_1-invariant submodules of A are called affine-invariant codes, and $GL_1(p^n)$-invariant codes in

$$A' = \{\sum_{v\neq 0} a_v X^v \mid a_v \in F\}$$

are cyclic codes. If C' is a cyclic code in A', then the extended cyclic code C is obtained by embedding:

$$\sum_{v\neq 0} a_v X^v \mapsto (-\sum_{v\neq 0} a_v)X^0 + \sum_{v\neq 0} a_v X^v.$$

From a cyclic code C' one can construct an extended cyclic code C of length p^n . It consists of codewords $(c_\infty, c_0, c_1, \ldots, c_{p^n-2}) \in C$, where $(c_0, c_1, \ldots, c_{p^n-2}) \in C'$ and $c_\infty + c_0 + c_1 + \ldots + c_{p^n-2} = 0$.

Invariance of a code $C \subseteq A$ under the action of the group V means that C is an ideal in the group ring A. Therefore, the question of the classification of affine-invariant codes is equivalent to classification of ideals in A, invariant under the group $GL_1(p^n)$.

5.1 Defining sets

A cyclic code C' of length $p^n - 1$ over a field F is an ideal of $F[x]/(x^{p^n-1} - 1)$, where the codeword

$$c = (c_0, c_2, \ldots, c_{p^n-2}) \in C'$$

is represented by the class of polynomial

$$c(x) = c_0 + c_2 x + \ldots + c_{p^n-2} x^{p^n-2}$$

Cyclic code C' is uniquely determined by its generating polynomial $f(x)$. Let ω be a primitive element of the field $\mathbb{F}_q$. Then the set T' of all numbers s, such that $0 < s \leq p^n - 1$ and $f(\omega^s) = 0$, is called the defining set of C'. So elements ω^s, $s \in T'$, are all zeros of the polynomials of the cyclic code C'.

Consider the following F-linear map of A:

$$\varphi_s\left(\sum_{\alpha\in\mathbb{F}_q} a_\alpha X^\alpha\right) = \sum a_\alpha \alpha^s.$$

If $C' \subseteq A'$ is a cyclic code then

$$T = \{s \mid \varphi_s(c) = 0 \; \forall c \in C'\}$$

is the defining set of C' (see [1, 3, 5]). Then $T \cup \{0\}$ will be the defining set of the extended cyclic code C.

For s, $0 \leq s \leq p^n - 1$, the p-adic expansion is

$$s = \sum_{i=0}^{n-1} s_i p^i, \quad (0 \leq s_i \leq p-1).$$

The partial order relation $\prec$ on $\{0, 1, \dots, p^n - 1\}$ is defined as follows:

$$\forall s, r \in \{0, 1, \dots, p^n - 1\}: \; s \prec r \iff s_i \leq r_i, \; 0 \leq i \leq n-1.$$

Theorem 5.1.1 **([1, 3, 5, 9])** *Let T be the defining set of an extended cyclic code C. Then C is affine-invariant if and only if the condition $s \in T$ implies $t \in T$ for any $t \prec s$.*

5.2 Quaternary affine-invariant codes

Now we consider extended cyclic codes of length 2^n over the ring $\mathbb{Z}_4$ of integers modulo 4. The ambient space will be

$$A = \left\{ \sum_{v\in V=\mathbb{F}_{2^n}} a_v X^v \mid a_v \in \mathbb{Z}_4 \right\}.$$

Let C be an extended cyclic code over $\mathbb{Z}_4$ in A (i.e. invariant under $GL_1(2^n)$).

There are two canonical subcodes of C:

$$C_1 = (C + 2A)/2A \quad \text{(residue code)},$$

$$C_2 = C \cap 2A = \{c \in C \mid 2c = 0\} \quad \text{(torsion code)}.$$

They can be considered as linear codes over $\mathbb{F}_2$. We say that (T_1, T_2) is the defining set of C if T_1 and T_2 are the defining sets of C_1 and C_2 respectively. Assuming that C_i is naturally embedded in A mod 2, we have

$$C_1 \subseteq C_2, \quad T_1 \supseteq T_2.$$

We will say that a quaternary code is affine-invariant if it is invariant under action of the group $AGL_1(2^n)$.

The following theorem is the main tool for our investigation.

Theorem 5.2.1 ([1, 3, 5]) *Let (T_1, T_2) be the defining set of an extended cyclic code C of length 2^n over $\mathbb{Z}_4$. Then C is affine-invariant if and only if the following two properties hold:*

(i) $s \in T_d,\ r \prec s \Rightarrow r \in T_d$ *for* $d = 1, 2$;

(ii) $s = s_0 + \cdots + s_i \cdot 2^i + 0 \cdot 2^{i+1} + 1 \cdot 2^{i+2} + \cdots \in T_2 \Rightarrow s_0 + \cdots + s_i \cdot 2^i + 1 \cdot 2^{i+1} + 0 \cdot 2^{i+2} + \cdots \in T_1$. *(Subscripts and superscripts* mod$n$.*)*

Now we show that all known good series of quaternary codes [13] are affine-invariant. Note that if $s = s_0 \cdot 2^0 + s_1 \cdot 2^1 + \cdots + s_{n-1} \cdot 2^{n-1} \in T_d$ then $s \cdot 2 \bmod (2^n - 1) = s_{n-1} \cdot 2^0 + s_0 \cdot 2^1 + \cdots + s_{n-2} \cdot 2^{n-1} \in T_d$, so we will denote by $Cl(s_0, s_1, \ldots, s_{n-1}) = Cl(s_0 + s_1 \cdot 2^1 + \cdots + s_{n-1} \cdot 2^{n-1})$ the cyclotomic coset of the number $s = s_0 \cdot 2^0 + s_1 \cdot 2^1 + \cdots + s_{n-1} \cdot 2^{n-1}$, that is, numbers s, $s \cdot 2 \bmod (2^n - 1)$, $s \cdot 2^2 \bmod (2^n - 1)$, …, $s \cdot 2^{n-1} \bmod (2^n - 1)$.

Preparata code is given by the defining set (T_1, T_2), where

$$T_1 = T_2 = \{Cl(0, \ldots, 0), Cl(1, 0, \ldots, 0)\}.$$

It is clear that conditions of Theorem 5.2.1 are satisfied, so this $\mathbb{Z}_4$-code is affine-invariant. For odd n, the Gray image of the $\mathbb{Z}_4$-Preparata code determines binary $(2^{n+1}, 2^{2^{n+1}-2n-2}, 6)$ code [13]. (Gray map sends elements $0, 1, 2, 3$ of $\mathbb{Z}_4$ to the binary combinations $00, 01, 11, 10$ respectively).

Kerdock code is the dual code to Preparata code, considered as $\mathbb{Z}_4$-code. It is given by the following defining set:

$$T_1 = T_2 = \{0, 1, 2, \ldots, p^n - 1\} \setminus \{Cl(1, \ldots, 1), Cl(0, 1, \ldots, 1)\}.$$

For odd n, the Gray image of the $\mathbb{Z}_4$-Kerdock code is binary $(2^{n+1}, 4^n, 2^n - 2^{(n-1)/2})$ code [13].

Quaternary Preparata and Kerdock codes are particular cases of **quaternary Reed-Muller codes** $QRM(r, n)$, defined by

$$T_1 = T_2 = \{Cl(s_0, \ldots, s_{n-1}) \mid s_0 + \cdots + s_{n-1} \leq n - 1 - r\}.$$

Quaternary Reed-Muller codes are also affine-invariant. Codes $QRM(n-2, n)$ and $QRM(1, n)$ are Preparata and Kerdock codes respectively. Note that $QRM(r, n)$ is a lifted Reed-Muller code, in the sense that $T_1 = T_2$ and T_1 determines binary Reed-Muller code $RM(r, n)$.

5.3 Binary affine-invariant codes for small dimensions

In this section we present a classification of binary affine-invariant codes of dimensions ≤ 7. (In fact, we have full description of codes for $n = 8$, but we decided not put them in this thesis, since results take too much pages).

Tables 5.1, 5.4,5.7, 5.10 and 5.13 present structures of extended cyclic codes in dimensions 3, 4, 5, 6 and 7 respectively. Here n-tuple $(s_0, s_1, \ldots, s_{n-1})$ denotes the cyclotomic coset $Cl(s_0, s_1, \ldots, s_{n-1}) = Cl(s_0 + s_1 \cdot 2^1 + \cdots + s_{n-1} \cdot 2^{n-1})$ of the number $s = s_0 \cdot 2^0 + s_1 \cdot 2^1 + \cdots + s_{n-1} \cdot 2^{n-1}$, that is, numbers s, $s \cdot 2 \bmod (2^n - 1)$, $s \cdot 2^2 \bmod (2^n - 1)$, $\ldots$, $s \cdot 2^{n-1} \bmod (2^n - 1)$. Using Theorem 5.1.1 we can find all

n-tuples, corresponding to binary affine-invariant codes. They are presented in the Tables 5.2, 5.5, 5.8, 5.11 and 5.14. We denote here

$$r_k = \{Cl(s_0, \dots, s_{n-1}) \mid s_0 + \cdots + s_{n-1} \leq k\},$$

so r_k determines Reed-Muller code $RM(n-k-1, n)$ for $k < n$ (r_n determines the trivial code $C = 0$). In our tables, we put a defining set T into a cell corresponding to r_k if $T \subseteq r_k$ and $T \not\subseteq r_{k-1}$.

Now we explain how to construct codes from defining sets. Let's consider the defining set generated by 5-tuple $\langle 11000 \rangle$. It contains all 5-tuples that are less than $\langle 11000 \rangle$ with respect to the order $\prec$, that is,

$$T = \langle 11000 \rangle = \{(11000), (10000), (00000)\} = \{3, 6, 12, 24, 17, 1, 2, 4, 8, 16, 0\}.$$

Then the affine-invariant code with the defining set T is

$$C = \{a = \sum_{\alpha \in \mathbb{F}_q} a_\alpha X^\alpha \mid \varphi_s(a) = 0 \ \text{ all } s \in T\}.$$

Other description is the following. Let ω be a primitive element of the field $\mathbb{F}_q$. Consider polynomial

$$g(x) = (x - \omega^1)(x - \omega^2)(x - \omega^4)(x - \omega^8)(x - \omega^{16})$$

$$\cdot (x - \omega^3)(x - \omega^6)(x - \omega^{12})(x - \omega^{24})(x - \omega^{17}) \in \mathbb{F}_2[x].$$

The polynomial $g(x)$ generates a cyclic code C', and the extension of this code gives us an affine-invariant code C with defining set T.

For $n = 3$ the defining set $\langle 100 \mid 100 \rangle$ determines $\mathbb{Z}_4$-Nordstrom-Robinson code. It is simultaneously a Kerdock and Preparata code, and its Gray image is a binary nonlinear $(16, 2^8, 6)$-code.

5.4 Quaternary affine-invariant codes for small dimensions

In this section we present a classification of quaternary affine-invariant codes of dimensions ≤ 7.

Using the classification of binary affine-invariant codes from the previous section and with the help of Theorem 5.2.1 we can find all n-tuples, corresponding to quaternary affine-invariant codes. They are presented in the Tables 5.3, 5.6, 5.9, 5.12 and 5.15, corresponding to the cases $n = 3, 4, 5, 6$ and 7 respectively. We presented "minimal" defining sets (T_1, T_2) of quaternary affine-invariant codes, in the following sense. If (T_1, T_2) is from our tables, $T' \supseteq T_1$ and T' is a defining set of some binary affine-invariant code, then (T', T_2) is also the defining set of a quaternary affine-invariant code (since in this case (T', T_2) also satisfies conditions of Theorem 5.2.1). The enumeration of the defining sets is organized in the following way. We put a defining set (T_1, T_2) into a cell corresponding to R_k if $(T_1 \cup T_2) \subseteq r_k$ and $(T_1 \cup T_2) \not\subseteq r_{k-1}$.

As an example, now we show how to construct the quaternary affine-invariant code from the defining set $T = (T_1, T_2) = \langle 10100, 11000 \mid 11000 \rangle$. As in the previous section, we have

$$T_2 = \langle 11000 \rangle = \{(11000), (10000), (00000)\} = \{3, 6, 12, 24, 17, 1, 2, 4, 8, 16, 0\},$$

$$T_1 = \langle 10100, 11000 \rangle = \{(10100), (11000), (10000), (00000)\}$$

$$= \{5, 10, 20, 9, 18, 3, 6, 12, 24, 17, 1, 2, 4, 8, 16, 0\}.$$

We compute polynomials

$$g_2(x) = (x - \omega^1)(x - \omega^2)(x - \omega^4)(x - \omega^8)(x - \omega^{16})$$

$$\cdot (x - \omega^3)(x - \omega^6)(x - \omega^{12})(x - \omega^{24})(x - \omega^{17}) \in \mathbb{F}_2[x],$$

and

$$g_1(x) = (x-\omega^1)(x-\omega^2)(x-\omega^4)(x-\omega^8)(x-\omega^{16})(x-\omega^3)(x-\omega^6)$$

$$\cdot(x-\omega^{12})(x-\omega^{24})(x-\omega^{17})(x-\omega^5)(x-\omega^{10})(x-\omega^{20})(x-\omega^9)(x-\omega^{18}) \in \mathbb{F}_2[x].$$

The polynomial $g_1(x)$ generates a cyclic binary code D_1', we lift this code to a quaternary code C_1' with the help of Hensel lifting and the extension of this code denote by C_1. The polynomial $g_2(x)$ generates a cyclic binary code D_2', we lift this code to a quaternary code C_2' with the help of Hensel lifting and the extension of this code denote by C_2. Then our quaternary affine-invariant code with the defining set T is

$$C = C_1 + 2C_2.$$

Table 5.1: Binary codes for $n = 3$

111
110
100
000

Table 5.2: Defining sets of binary affine-invariant codes for $n = 3$

r_3	$\langle 111 \rangle$
r_2	$\langle 110 \rangle$
r_1	$\langle 100 \rangle$
r_0	$\langle 000 \rangle$

Table 5.3: Defining sets of quaternary affine-invariant codes for $n = 3$

R_3	$\langle 111 \mid 111 \rangle$
R_2	$\langle 110 \mid 110 \rangle$
R_1	$\langle 100 \mid 100 \rangle$
R_0	$\langle 000 \mid 000 \rangle$

Table 5.4: Binary codes for $n = 4$

1111	
1110	
1100	1010
1000	
0000	

Table 5.5: Defining sets of binary affine-invariant codes for $n = 4$

r_4	$\langle 1111 \rangle$
r_3	$\langle 1110 \rangle$
r_2	$\langle 1100, 1010 \rangle, \langle 1100 \rangle, \langle 1010 \rangle$
r_1	$\langle 1000 \rangle$
r_0	$\langle 0000 \rangle$

Table 5.6: Defining sets of quaternary affine-invariant codes for $n = 4$

R_4	$\langle 1111 \mid 1111 \rangle$
R_3	$\langle 1110 \mid 1110 \rangle$
R_2	$\langle 1100, 1010 \mid 1100, 1010 \rangle$, $\langle 1100, 1010 \mid 1100 \rangle, \langle 1010, 1100 \mid 1010 \rangle$
R_1	$\langle 1000 \mid 1000 \rangle$
R_0	$\langle 0000 \mid 0000 \rangle$

Table 5.7: Binary codes for $n = 5$

11111	
11110	
11100	11010
10100	11000
10000	
00000	

Table 5.8: Defining sets of binary affine-invariant codes for $n = 5$

r_5	$\langle 11111 \rangle$
r_4	$\langle 11110 \rangle$
r_3	$\langle 11100, 11010 \rangle, \langle 11100 \rangle, \langle 11010 \rangle$
r_2	$\langle 10100, 11000 \rangle, \langle 11000 \rangle, \langle 10100 \rangle$
r_1	$\langle 10000 \rangle$
r_0	$\langle 00000 \rangle$

Table 5.9: Defining sets of quaternary affine-invariant codes for $n = 5$

R_5	$\langle 11111 \mid 11111 \rangle$
R_4	$\langle 11110 \mid 11110 \rangle$
R_3	$\langle 11100, 11010 \mid 11100, 11010 \rangle, \langle 11100, 11010 \mid 11100 \rangle,$ $\langle 11010, 11100 \mid 11010 \rangle$
R_2	$\langle 11000, 10100 \mid 10100, 11000 \rangle, \langle 11000, 10010 \mid 11000 \rangle,$ $\langle 10100, 11000 \mid 10100 \rangle$
R_1	$\langle 10000 \mid 10000 \rangle$
R_0	$\langle 00000 \mid 00000 \rangle$

Table 5.10: Binary codes for $n = 6$

111111			
111110			
111100	111010	110110	
111000	101100	110100	101010
101000	110000	100100	
100000			
000000			

Table 5.11: Defining sets of binary affine-invariant codes for $n = 6$

r_6	$\langle 111111 \rangle$
r_5	$\langle 111110 \rangle$
r_4	$\langle 111100, 110110, 111010 \rangle$ $\langle 111100 \rangle$, $\langle 111010 \rangle$, $\langle 110110 \rangle$ $\langle 111100, 111010 \rangle$, $\langle 111100, 110110 \rangle$, $\langle 111010, 110110 \rangle$, $\langle 111100, 110110, 101010 \rangle$, $\langle 110110, 111000, 101010 \rangle$, $\langle 111100, 101010 \rangle$, $\langle 110110, 111000 \rangle$, $\langle 110110, 101010 \rangle$
r_3	$\langle 110100, 101100, 111000, 101010 \rangle$, $\langle 110100 \rangle$, $\langle 101100 \rangle$ $\langle 111000 \rangle$, $\langle 101010 \rangle$, $\langle 110100, 101100 \rangle$, $\langle 110100, 111000 \rangle$, $\langle 110100, 101010 \rangle$, $\langle 101100, 111000 \rangle$, $\langle 101100, 101010 \rangle$, $\langle 111000, 101010 \rangle$, $\langle 110100, 101100, 111000 \rangle$, $\langle 101100, 111000, 101010 \rangle$, $\langle 111000, 101010, 110100 \rangle$, $\langle 101010, 110100, 101100 \rangle$, $\langle 111000, 101010, 100100 \rangle$, $\langle 101010, 110000, 100100 \rangle$, $\langle 111000, 100100 \rangle$, $\langle 101010, 100100 \rangle$
r_2	$\langle 101000, 110000, 100100 \rangle$, $\langle 110000 \rangle$, $\langle 101000 \rangle$, $\langle 100100 \rangle$, $\langle 110000, 101000 \rangle$, $\langle 110000, 100100 \rangle$, $\langle 101000, 100100 \rangle$
r_1	$\langle 100000 \rangle$
r_0	$\langle 000000 \rangle$

Table 5.12: Defining sets of quaternary affine-invariant codes for $n = 6$

R_6	$\langle 111111 \mid 111111 \rangle$
R_5	$\langle 111110 \mid 111110 \rangle$
R_4	$\langle 111100, 111010, 110110 \mid 111100, 111010, 110110 \rangle$, $\langle 111100, 111010 \mid 111100 \rangle$, $\langle 111010, 111100, 110110 \mid 111010 \rangle$, $\langle 110110, 111010 \mid 110110 \rangle$, $\langle 111100, 111010, 110110 \mid 111100, 111010 \rangle$, $\langle 111100, 110110, 111010 \mid 111100, 110110 \rangle$, $\langle 111010, 110110, 111100 \mid 111010, 110110 \rangle$, $\langle 111100, 110110, 101010, 111010 \mid 111100, 110110, 101010 \rangle$, $\langle 110110, 111000, 101010, 111010 \mid 110110, 111000, 101010 \rangle$, $\langle 111100, 101010, 111010 \mid 111100, 101010 \rangle$, $\langle 110110, 111000, 111010 \mid 110110, 111000 \rangle$, $\langle 110110, 101010, 111010 \mid 110110, 101010 \rangle$
R_3	$\langle 111000, 101100, 110100, 101010 \mid 111000, 101100, 110100, 101010 \rangle$, $\langle 110100, 111000, 101010 \mid 110100 \rangle$ $\langle 101100, 110100 \mid 101100 \rangle$, $\langle 111000, 101100 \mid 111000 \rangle$ $\langle 101010, 110010 \mid 101010 \rangle$, $\langle 110100, 101100, 111000, 101010 \mid 110100, 101100 \rangle$, $\langle 110100, 111000, 101010, \mid 110100, 111000 \rangle$, $\langle 110100, 101010, 111000, 110010 \mid 110100, 101010 \rangle$, $\langle 101100, 111000, 110100 \mid 101100, 111000 \rangle$, $\langle 101100, 101010, 110100 \mid 101100, 101010 \rangle$, $\langle 111000, 101010, 101100 \mid 111000, 101010 \rangle$, $\langle 110100, 101100, 111000, 101010 \mid 110100, 101100, 111000 \rangle$, $\langle 101100, 111000, 101010, 110100 \mid 101100, 111000, 101010 \rangle$, $\langle 111000, 101010, 110100, 101100 \mid 101010, 110100 \rangle$, $\langle 101010, 110100, 101100, 111000 \mid 101010, 110100, 101100 \rangle$, $\langle 111000, 101010, 101100 \mid 111000, 101010, 100100 \rangle$, $\langle 101010, 110000, 100100, 110010 \mid 101010, 110000, 100100 \rangle$, $\langle 111000, 101100, 101000 \mid 111000, 100100 \rangle$, $\langle 101010, 110010, \mid 101010, 100100 \rangle$, $\langle 101010, 110010 \mid 101010, 110000 \rangle$
R_2	$\langle 110000, 101000, 100100 \mid 110000, 101000, 100100 \rangle$, $\langle 110000, 101000 \mid 110000 \rangle$, $\langle 101000, 110000, 100100 \mid 101000 \rangle$ $\langle 100100, 101000 \mid 100100 \rangle$, $\langle 110000, 101000, 100100 \mid 110000, 101000 \rangle$ $\langle 110000, 100100, 101000 \mid 110000.100100 \rangle$ $\langle 101000, 100100, 110000 \mid 101000, 100100 \rangle$
R_1	$\langle 100000 \mid 100000 \rangle$
R_0	$\langle 000000 \mid 000000 \rangle$

Table 5.13: Binary codes for $n = 7$

1111111				
1111110				
1111100		1111010	1110110	
1111000	1110100	1101100	1011100	1110000
1110000	1101000	1011000	1010100	1100100
1010000		1100000	1001000	
1000000				
0000000				

Table 5.14: Defining sets of binary affine-invariant codes for $n = 7$

r_7	⟨1111111⟩
r_6	⟨1111110⟩
r_5	⟨1111100, 1111010, 1110110⟩ , ⟨1111100⟩ , ⟨1111010⟩ , ⟨1110110⟩ , ⟨1111100, 1111010⟩ , ⟨1111100, 1110110⟩ , ⟨1111010, 1110110⟩ , ⟨1111100, 1101010⟩ , ⟨1111010, 1101100⟩ , ⟨1110110, 1111000⟩
r_4	⟨1111000, 1110100, 1101100, 1011100, 1101010⟩ ⟨1110100⟩ , ⟨1111000⟩ , ⟨1011100⟩ , ⟨1101100⟩ , ⟨1101010⟩ , ⟨1111000, 1101100⟩ , ⟨1111000, 1110100⟩ , ⟨1111000, 1011100⟩ , ⟨1111000, 1101010⟩ , ⟨1110100, 1101100⟩ , ⟨1110100, 1011100⟩ , ⟨1110100, 1101010⟩ , ⟨1101100, 1011100⟩ , ⟨1101100, 1101010⟩ , ⟨1011100, 1101010⟩ , ⟨1111000, 1110100, 1101100⟩ ⟨1101100, 1011100, 1101010⟩ , ⟨1110100, 1101100, 1011100⟩ , ⟨1101010, 1111000, 1110100⟩ , ⟨1011100, 1101010, 1111000⟩ , ⟨1111000, 1110100, 1011100⟩ , ⟨1101100, 1110100, 1111000⟩ , ⟨1110100, 1101100, 1101010⟩ , ⟨1011100, 1101010, 1110100⟩ , ⟨1101010, 1110100, 1101100⟩ , ⟨1111000, 1110100, 1101100, 1101010⟩ , ⟨1110100, 1101100, 1101010, 1011100⟩ , ⟨1101100, 1101010, 1011100, 1111000⟩ , ⟨1101010, 1011100, 1111000, 1110100⟩ , ⟨1011100, 1111000, 1110100, 1101100⟩ , ⟨1111000, 1010100⟩ , ⟨1111000, 1100100⟩ , ⟨1110100, 1011000⟩ , ⟨1101100, 1010100⟩ , ⟨1101100, 1110000⟩ , ⟨1011100, 1101000⟩ , ⟨1101010, 1100100⟩ , ⟨1101010, 1110000⟩ , ⟨1111000, 1010100, 1100100⟩ , ⟨1101100, 1010100, 1110000⟩ , ⟨1101010, 1100100, 1110000⟩ , ⟨1101100, 1101010, 1110000⟩ , ⟨1101100, 1010100, 1111000⟩ , ⟨1100100, 1111000, 1101010⟩
r_3	⟨1110000, 1101000, 1011000, 1010100, 1100100⟩ , ⟨1110000⟩ , ⟨1101000⟩ , ⟨1011000⟩ , ⟨1010100⟩ , ⟨1100100⟩ , ⟨1110000, 1101000⟩ , ⟨1110000, 10110000⟩ , ⟨1110000, 1010100⟩ , ⟨1110000, 1100100⟩ , ⟨1101000, 1011000⟩ , ⟨1101000, 1010100⟩ , ⟨1101000, 1100100⟩ , ⟨1011000, 1010100⟩ , ⟨1011000, 1100100⟩ , ⟨1010100, 1100100⟩ , ⟨1110000, 1101000, 1011000⟩ , ⟨1101000, 1011000, 1010100⟩ , ⟨1011000, 1010100, 1100100⟩ , ⟨1010100, 1100100, 1110000⟩ , ⟨1100100, 1110000, 1101000⟩ , ⟨1110000, 1101000, 1010100⟩ , ⟨1101000, 1011000, 1100100⟩ , ⟨1011000, 1010100, 1110000⟩ , ⟨1010100, 1100100, 1101000⟩ , ⟨1100100, 1110000, 1011000⟩ , ⟨1110000, 1101000, 1011000, 1010100⟩ , ⟨1101000, 1011000, 1010100, 1100100⟩ , ⟨1011000, 1010100, 1100100, 1110000⟩ , ⟨1010100, 1100100, 1110000, 1101000⟩ , ⟨1100100, 1110000, 1101000, 1011000⟩ , ⟨1110000, 1001000⟩ , ⟨1010100, 1100000⟩ , ⟨1100100, 1010000⟩
r_2	⟨1100000, 1010000, 1001000⟩ , ⟨1100000⟩ , ⟨10100000⟩ , ⟨1001000⟩ ⟨1100000, 1010000⟩ , ⟨1100000, 1001000⟩ , ⟨1010000, 1001000⟩
r_1	⟨1000000⟩
r_0	⟨0000000⟩

Table 5.15: Defining sets of quaternary affine-invariant codes for $n = 7$

R_7	$\langle 1111111 \mid 1111111 \rangle$
R_6	$\langle 1111110 \mid 1111110 \rangle$
R_5	$\langle 1111100, 1111010, 1110110 \mid 1111100, 1111010, 1110110 \rangle$, $\langle 1111100, 1111010 \mid 1111100 \rangle$, $\langle 1111010, 1111100, 1110110 \mid 1111010 \rangle$, $\langle 1110110, 1111010 \mid 1110110 \rangle$, $\langle 1111100, 1111010, 1110110 \mid 1111100, 1111010 \rangle$, $\langle 1111100, 1110110, 1111010 \mid 1111100, 1110110 \rangle$, $\langle 1111010, 1110110, 1111100 \mid 1111010, 1110110 \rangle$, $\langle 1111100, 1111010 \mid 1111100, 1101010 \rangle$, $\langle 1111010, 1111100, 1110110, 1011010 \mid 1111010, 1101100 \rangle$, $\langle 1110110, 1111000 \mid 1110110, 1111000 \rangle$

Table 5.16: Defining sets of quaternary affine-invariant codes for $n = 7$ (cont.)

R_4	$\langle 1111000, 1110100, 1101100, 1011100, 1101010 \mid$ $1111000, 1110100, 1101100, 1011100, 1101010\rangle$, $\langle 1111000, 1110010 \mid 1111000\rangle$, $\langle 1110100, 1111000, 1101010 \mid 1110100\rangle$, $\langle 1101100, 1110100, 1011010 \mid 1101100\rangle$, $\langle 1011100, 1101100, 1110100 \mid 1011100\rangle$, $\langle 1101010, 1110010, 1101100 \mid 1101010\rangle$, $\langle 1111000, 1110100, 1110010, 1101010 \mid 1111000, 1110100\rangle$, $\langle 1111000, 1101100, 1110010, 1110100, 1011010 \mid 1111000, 1101100\rangle$, $\langle 1111000, 1011100, 1101100, 1110100 \mid 1111000, 1011100\rangle$, $\langle 1111000, 1101010, 1110010, 1101100 \mid 1111000, 1101010\rangle$, $\langle 1110100, 1101100, 1111000, 1101010 \mid 1110100, 1101100\rangle$, $\langle 1110100, 1011100, 1111000, 1101010, 1101100 \mid 1110100, 1011100\rangle$, $\langle 1110100, 1101010, 1111000, 1110010, 1101100 \mid 1110100, 1101010\rangle$, $\langle 1101100, 1011100, 1110100, 1101010 \mid 1101100.1011100\rangle$, $\langle 1101100, 1101010, 1110100, 1110010 \mid 1101100, 1101010\rangle$, $\langle 1011100, 1101010, 1101100, 1110100 \mid 1011100, 1101010\rangle$, $\langle 1111000, 1110100, 1101100, 1011100, 1101010 \mid 1111000, 1110100, 1101100\rangle$, $\langle 1110100, 1101100, 1011100, 1111000, 1101010 \mid 1110100, 1101100, 1011100\rangle$, $\langle 1101100, 1011100, 1101010, 1110100 \mid 1101100, 1011100, 1101010\rangle$, $\langle 1011100, 1101010, 1111000, 1101100, 1110100 \mid 1011100, 1101010, 1111000\rangle$, $\langle 1101010, 1111000, 1110100, 1110010, 1101100 \mid 1101010, 1111000, 1110100\rangle$, $\langle 1111000, 1110100, 1011100, 1101010, 1101100 \mid 1111000, 1110100, 1011100\rangle$, $\langle 1110100, 1101100, 1101010, 1111000, 1110010 \mid 1110100, 1101100, 1101010\rangle$, $\langle 1101100, 1110100, 1111000, 1011010 \mid 1101100, 1110100, 1111000\rangle$, $\langle 1011100, 1101010, 1110100, 1101100, 1111000 \mid 1011100, 1101010, 1110100\rangle$, $\langle 1101010, 1110100, 1101100, 1110010, 1111000 \mid 1101010, 1110100, 1101100\rangle$, $\langle 1111000, 1110100, 1101100, 1101010, 1011100 \mid 1111000, 1110100, 1101100, 1101010\rangle$ $\langle 1110100, 1101100, 1101010, 1011100, 1111000 \mid 1110100, 1101100, 1101010, 1011100\rangle$, $\langle 1101100, 1101010, 1011100, 1111000, 1110100 \mid 1101100, 1101010, 1011100, 1111000\rangle$, $\langle 1101010, 1011100, 1111000, 1110100 \mid 1101010, 1011100, 1111000, 1110100\rangle$, $\langle 1011100, 1111000, 1110100, 1101100, 1101010 \mid 1011100, 1111000, 1110100, 1101100\rangle$, $\langle 1111000, 1010100, 1110010 \mid 1111000, 1010100\rangle$, $\langle 1111000, 1100100, 1011100 \mid 1111000, 1100100\rangle$ $\langle 1110100, 1111000, 1101010 \mid 1110100, 1011000\rangle$ $\langle 1101100, 1110100, 1011010 \mid 1101100, 1010100\rangle$, $\langle 1101100, 1110100, 1011010 \mid 1101100, 1110000\rangle$, $\langle 1011100, 1101100, 1110100 \mid 1011100, 1101000\rangle$, $\langle 1101010, 1110010, 1101100, 1010100 \mid 1101010, 1100100\rangle$, $\langle 1101010, 1110010, 1101100 \mid 1101010, 1110000\rangle$, $\langle 1111000, 1011100, 1101000 \mid 1111000, 1010100, 1100100\rangle$, $\langle 1101100, 1110100, 1010110 \mid 1101100, 1010100, 1110000\rangle$, $\langle 1101010, 1110010, 1101100 \mid 1101010, 1100100, 1110000\rangle$, $\langle 1101100, 1101010, 1110000 \mid 1101100, 1101010, 1110000\rangle$, $\langle 1101100, 1111000, 1011100, 1010110 \mid 1100100, 1111000, 1101010\rangle$, $\langle 1111000, 1101010, 1011100 \mid 1100100, 1111000, 1101010\rangle$

Table 5.17: Defining sets of quaternary affine-invariant codes for $n = 7$ (more cont.)

R_3	⟨1110000, 1101000, 1011000, 1010100, 1100100 | 1110000, 1101000, 1011000, 1010100, 1100100⟩ , ⟨1110000, 1011000 | 110000⟩ , ⟨1101000, 1110000, 1001010 | 1101000⟩ , ⟨1011000, 11010000, 1100100 | 1011000⟩ , ⟨1010100, 1100100, 1011000 | 1010100⟩ , ⟨1100100, 1101000, 1001010 | 1100100⟩ ⟨1110000, 1101000, 1011000, 1001010 | 1110000, 1101000⟩ , ⟨1110000, 1011000, 1101000, 1100100 | 1110000, 1011000⟩ , ⟨1110000, 1010100, 1011000, 1100100 | 1110000, 1010100⟩ , ⟨1110000, 1100100, 1100010, 1101000, 1001010 | 1100100⟩ , ⟨1101000, 1011000, 1110000, 1001010, 1100100 | 1101000, 1011000⟩ , ⟨1101000, 1010100, 1110000, 1100100, 1011000 | 1010100⟩ , ⟨1101000, 1100100, 1110000, 1001010 | 1101000, 1100100⟩ , ⟨1011000, 1010100, 1101000, 1100100 | 1011000, 1010100⟩ , ⟨1011000, 1100100, 1101000 | 1011000, 1100100⟩ , ⟨1010100, 1100100, 1011000, 1101000 | 1010100, 1100100⟩ , ⟨1110000, 1101000, 1011000, 1001010, 1100100 | 1101000, 1011000⟩ , ⟨1101000, 1011000, 1010100, 1110000, 1100100 | 1011000, 1010100⟩ , ⟨1011000, 1010100, 1100100, 1101000 | 1011000, 1010100, 1100100⟩ , ⟨1010100, 1100100, 1110000, 1011000, 1101000 | 1010100, 1100100, 1110000⟩ , ⟨1100100, 1110000, 1101000, 1001010, 1011000 | 1100100, 1110000, 1101000⟩ , ⟨1110000, 1101000, 1010100, 1100010, 1100100 | 1110000, 1101000; 1010100⟩ , ⟨1101000, 1011000, 1100100, 1110000, 1001010 | 1101000, 1011000, 1100100⟩ , ⟨1011000, 1010100, 1110000, 1101000, 1100100 | 1011000, 1010100, 1110000⟩ , ⟨1010100, 1100100, 1101000, 1011000, 1110000 | 1100100, 1101000⟩ , ⟨1100100, 1110000, 1011000, 1101000, 1001010 | 1100100, 1011000⟩, ⟨1110000, 1101000, 1011000, 1010100, 1100100 | 1110000, 1101000, 1011000, 1010100⟩, ⟨1101000, 1011000, 1010100, 1100100, 1110000 | 1101000, 1011000, 1010100, 1100100⟩ , ⟨1011000, 1010100, 1100100, 1110000, 1101000 | 1011000, 1010100, 1100100, 1110000⟩ , ⟨1010100, 1100100, 1110000, 1101000, 1011000 | 1010100, 1100100, 1110000, 1101000⟩ , ⟨1100100, 1110000, 1101000, 1011000, 1001010 | 1100100, 1110000, 1101000, 1011000⟩ , ⟨1110000, 1011000 | 1110000, 1001000⟩ , ⟨1010100, 1100100, 1011000 | 1010100, 1100000⟩ , ⟨1100100, 1101000, 1001010 | 1100100, 1010000⟩
R_2	⟨1100000, 1010000, 1001000 | 1100000, 1010000, 1001000⟩, ⟨1100000, 1010000 | 1100000⟩ , ⟨10100000, 1100000, 1001000 | 10100000⟩, ⟨1001000, 1010000 | 1001000⟩ , ⟨1100000, 1010000, | 1100000, 1010000⟩, ⟨1100000, 1001000, 1010000 | 1100000, 1001000⟩, ⟨1010000, 1001000, 1100000 | 1010000, 100100⟩
R_1	⟨1000000 | 1000000⟩
R_0	⟨0000000 | 0000000⟩

Bibliography

[1] K. Abdukhalikov. Affine invariant and cyclic codes over p-adic numbers and finite rings, *Des. Codes Cryptogr.* 23 (2001), 343–370.

[2] K. Abdukhalikov, Codes over p-adic numbers and finite rings invariant under the full affine group, *Finite Fields Appl.* 7 (2001), no. 4, 449–467.

[3] K. Abdukhalikov, Defining sets of extended cyclic codes invariant under the affine group, *J. Pure Appl. Algebra* 196 (2005), 1–19.

[4] K. Abdukhalikov, E. Bannai and S. Suda, Association schemes related to universally optimal configurations, Kerdock codes and extremal Euclidean line-sets, *J. Combin. Theory. Ser. A* 116 (2009), 434–448.

[5] K. Abdukhalikov, On codes over rings invariant under affine groups, *Adv. Math. Commun.* 7 (2013), 253–265.

[6] T. Abualrub, A. Ghrayeb, R. H. Oehmke, A mass formula and rank of $\mathbb{Z}_4$ cyclic codes of length 2^e, *IEEE Trans. Inform. Theory* 50 (2004), no. 12, 33063312.

[7] T. Abualrub, R. H. Oehmke, On the generators of $\mathbb{Z}_4$ cyclic codes of length 2^e, *IEEE Trans. Inform. Theory* 49 (2003), no. 9, 21262133.

[8] T. Abualrub, R. H. Oehmke, Cyclic codes of length 2^e over $\mathbb{Z}_4$, International Workshop on Coding and Cryptography (WCC 2001) (Paris), *Discrete Appl. Math.* 128 (2003), no. 1, 39.

[9] T. Berger and P. Charpin, The permutation group of affine-invariant extended cyclic codes, *IEEE Trans. Inform. Theory* 42 (1996), No. 6, 2194–2209.

[10] D. K. Ray-Chaudhuri, J. T. Blackford, A transform approach to permutation groups of cyclic codes over Galois rings, *IEEE Trans. Inform. Theory* 46 (2000), 2050–2058.

[11] P. Delsarte, On cyclic codes that are invariant under the general linear group. *IEEE Trans. Inform. Theory* 16 (1970), 760–769.

[12] B. K. Dey, B. S. Rajan, Affine invariant extended cyclic codes over Galois rings, *IEEE Trans. Inform. Theory* 50, No. 4 (2004), 691–698.

[13] R. Hammons, P. V. Kumar, A. R. Calderbank, N. J. A. Sloane and P. Solé, The Z_4-linearity of Kerdock, Preparata, Goethals, and related codes, *IEEE Trans. Inform. Theory* 40, No. 2 (1994), 301–319.

[14] W. Cary Huffman, Codes and groups, In "Handbook of Coding Theory", (V. S. Pless and W. C. Huffman, Eds.) Volume 2, Chapter 17, 1345–1440. Elsevier, Amsterdam, 1998.

[15] Huffman W. C., Pless V. Fundamentals of Error-Correcting Codes. Cambridge University Press, 2003.

[16] T. Kasami, S. Lin and W. W. Peterson, Some results on cyclic codes which are invariant under the affine group and their applications, *Inform. and Control* 11 (1967), 475–496.

[17] San Ling and Chaoping Xing, Coding Theory: A First Course, Cambridge University Press, 2004.

[18] Ron M. Roth, Introduction to Coding Theory, Cambridge University Press, 2006.

[19] Z-X. Wan, Quaternary codes, Series on Applied Mathematics: Volume 8, World Scientific, 1997.

[20] Z.-X. Wan, Lectures on finite fields and Galois rings, World Scientific, Singapore, 2003.