UNIVERSITY OF ZURICH

Master Thesis

CLASSIFICATION AND CONSTRUCTION OF SELF-DUAL CONVOLUTIONAL CODES

Author:
Sebastian Heri

Supervisor:
Prof. Dr. Joachim Rosenthal

Co-Supervisor:
Dr. Julia Lieb

Universität Zürich[UZH]

Department of Mathematics
University of Zurich
March 16, 2021

Abstract

This thesis discusses the subject of self-dual convolutional codes, whereby some characteristics, a few classifications and constructions of such are presented.
In particular, we will begin by introducing similar characteristics for self-dual linear block codes and their most popular construction methods. Next, we will study convolutional codes, while firstly focusing on the idea of a non-catastrophic convolutional code, which will be shown to be a characteristic of a self-dual convolutional code. Further, self-dual codes will be studied and equivalent conditions of self-duality will be presented. Using these conditions, multiple groups of self-dual convolutional codes will be completely classified.
Then, we will the shift focus to binary self-dual convolutional codes and dive into the construction of such codes, whereby the “Building-up construction" and the “Harada-Munemasa construction" will be generalized and closely examined. Finally, we will construct two examples of binary self-dual $(10, 5)$ convolutional codes with the help of the “generalized Building-up construction".

Acknowledgements

I would like to thank Prof. Dr. Joachim Rosenthal, my advisor, for suggesting this interesting topic and educating me in many algebraic subjects, ranging from Algebra to Coding Theory. His teachings gave me the necessary skills to understand the underlying mathematical theory of this thesis.
Further, I am grateful for the generous help of Dr. Julia Lieb, my co-advisor, in shape of countless email exchanges and numerous meetings, where she defused my misunderstandings and suggested many great ideas. Without her this thesis would not have been possible.
Then, I thank Pascal Studer and Andreia Venzin for being such amazing fellow students on this mathematical journey.
Finally, I would like to express my gratitude towards my family and friends for their extraordinary and loving support in every step of my life.

Contents

1 Introduction **3**
1.1 Historical Background 3
1.2 Basic Notions 4

2 Self-Dual Linear Block Codes **5**
2.1 Prelimenaries 5
2.2 Construction Methods 10

3 Convolutional Codes **14**
3.1 Prelimenaries 14
3.2 Non-Catastrophic Codes 16
3.3 Self-Dual Convolutional Codes 22

4 Classification of Self-Dual Convolutional Codes **33**
4.1 Self-Dual (2,1) Convolutional Codes 33
4.2 Binary Self-Dual (4,2) Convolutional Codes 35
4.3 Double Diagonal Generator Matrix 36
4.4 Double Upper Triangular Generator Matrices 36

5 Construction of Self-Dual Convolutional Codes **39**
5.1 Construction Methods 39
5.2 Examples of Self-Dual Convolutional Codes 62

6 Conclusion **67**

7 Appendix **69**

1 Introduction

1.1 Historical Background

The idea of using codes to transmit information was first introduced in 1948 by Claude Shannon in his famous work “A Mathematical Theory on Communication”. Shortly after in 1950, Richard Hamming invented the notion of the “Hamming distance" with a new code, where errors in transmission can not only be detected, but also corrected, called the Hamming code and generated by

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

This wasn‘t the first so called error-correcting code as the binary Golay code was already known, yet the new concept of distance lead to the further study of error-correction.
The particular branch of coding theory we are interested in, are the so called self-dual codes. Beginning in 1971, Vera Pless published a series of papers on the complete classification of binary self-dual linear block codes up to length 30 ([15, 16]). An example of such a code is the $[8, 4, 4]$ Hamming code generated by G and adding $(1, 1, 1, 0)^T$ as a column.
At this point, all binary self-dual linear block codes up to length 36 have been completely classified ([7]) and a lot of research has been done on self-dual block codes over other finite fields ([14]). For self-dual convolutional codes on the other hand, not much is known. One interesting paper on the subject is [5], where the authors present an algorithm to find systematic self-dual convolutional codes, but they do not dive into the theory much. We will seek to do that dive and develop some basic theory surrounding self-dual convolutional codes in the following pages.

1.2 Basic Notions

Let $\mathbb{F}_q$ be the field with q elements and $R := \mathbb{F}_q[z]$ its polynomial ring.

Definition 1.2.1. *Let*

$$G = \begin{pmatrix} — & g_1 & — \\ & \vdots & \\ — & g_k & — \end{pmatrix} \in \mathbb{F}_q^{k \times n}.$$

*Then its **rowspan** is defined to be all linear combinations of rows*

$$rowspan(G) := \left\{ \sum_{i=1}^{k} a_i g_i \mid a_i \in \mathbb{F}_q \right\}$$

*and its **kernel***

$$ker(G) := \left\{ v \in \mathbb{F}_q^n \mid Gv^T = 0 \right\}.$$

Definition 1.2.2. *Let*

$$G(z) = \begin{pmatrix} — & g_1(z) & — \\ & \vdots & \\ — & g_k(z) & — \end{pmatrix} \in R^{k \times n}.$$

*Then its **rowspan** is defined to be all linear combinations of rows*

$$rowspan(G(z)) := \left\{ \sum_{i=1}^{k} a_i(z) g_i(z) \mid a_i(z) \in R \right\}$$

*and its **kernel***

$$ker(G(z)) := \left\{ f(z) \in R^n \mid G(z) f(z)^T = 0 \right\}.$$

Definition 1.2.3. *Let*

$$G(z) = \begin{pmatrix} | & & | \\ g_1(z) & \cdots & g_n(z) \\ | & & | \end{pmatrix} \in R^{k \times n}$$

with $k \leq n$.

*A $k \times k$ **submatrix** $\tilde{G}(z)$ of $G(z)$ is defined to be*

$$\tilde{G}(z) = \begin{pmatrix} | & & | \\ g_{i_1}(z) & \cdots & g_{i_k}(z) \\ | & & | \end{pmatrix} \in R^{k \times k}$$

for any distinct $i_1, ..., i_k \in \{1, ..., n\}$.

*A k**-th minor** is defined to be the determinant of a $k \times k$ submatrix and the **set of** k**-th minors** is the set of determinants of all $k \times k$ submatrices.*

Definition 1.2.4. *Let $M = \{m_1(z), \ldots, m_l(z)\}$ be a subset of R. Then*

$$< M >:= \left\{ \sum_{i=1}^{l} a_i(z) m_i(z) \mid a_i(z) \in R \right\}$$

*denotes the **ideal generated by M**.*

2 Self-Dual Linear Block Codes

We start off by discussing known properties of self-dual linear block codes and then use these properties to present two popular construction methods, where binary self-dual linear block codes of smaller lengths are used to find ones of bigger lengths. These construction methods will be very important to our results in chapter 5, as we will present some generalisations thereof.

2.1 Prelimenaries

Let $\mathbb{F}_q$ denote the finite field with q elements.

Definition 2.1.1. *A (n, k) **linear block code** $\mathcal{C}$ over $\mathbb{F}_q$ is a k-dimensional linear subspace $\mathcal{C} \subset \mathbb{F}_q^n$, where $k \leq n$.*

Being a k-dimensional linear subspace implies that there exists a $k \times n$ matrix G, called a **Generator matrix** of $\mathcal{C}$, such that $\mathcal{C} = rowspan(G)$. This G uniquely determines the code and is commonly used in such a way.
In this chapter, only linear block codes will be discussed and therefore the words “code", “block code" and “linear block code" shall be interpreted the same way.

Definition 2.1.2. *We say that two generator matrices G_1 and G_2 are **equivalent** if there exists an invertible matrix $A \in \mathbb{F}_q^{k\times k}$ such that*

$$G_1 = AG_2.$$

In other words, two generator matrices are equivalent if they differ by exchanging, multiplying or adding rows.

Definition 2.1.3. *Let $\mathcal{C}$ be a (n, k) linear block code. Then there exists $H \in \mathbb{F}_q^{n-k\times n}$ such that*

$$\mathcal{C} = ker(H).$$

*We call H a **Parity-check matrix** of $\mathcal{C}$.*

Neither H nor G are unique as every invertible row operation on either leaves the code unchanged.

Definition 2.1.4. *A code over $\mathbb{F}_2$ is called **binary**.*

Definition 2.1.5. $\mathcal{C}^\perp = \left\{ x \in \mathbb{F}_q^n \mid xc^T = 0 \ \ \forall c \in C \right\}$ *is called the **dual** of a code $\mathcal{C} \subset \mathbb{F}_q^n$.*

If the condition of $xc^T = 0$ holds, x and c are said to be **orthogonal**. Furthermore, the dual of a linear block code is always a linear block code.

Definition 2.1.6. *We say that a code $\mathcal{C}$ is **self-orthogonal** if $\mathcal{C} \subset \mathcal{C}^\perp$ and **self-dual** if $\mathcal{C} = \mathcal{C}^\perp$.*

So self-orthogonality is fulfilled if all codewords are orthogonal to each other, whereas self-duality demands the further restriction of no other element in the vector space $\mathbb{F}_q^n$ being orthogonal to all codewords.

An immediate consequence of the definition of the dual is that any generator matrix of the code is a parity-check matrix of the dual.

Lemma 2.1.7. *[12] Let G be a generator matrix of $\mathcal{C}$. Then*

$$\mathcal{C}^{\perp} = ker(G)$$

or in other words, G is a parity-check matrix of $\mathcal{C}^{\perp}$.

Proof. Let $c \in \mathcal{C}^{\perp}$ and g_i be the i-th row of G. Then

$$g_i c^T = 0$$

as c is orthogonal to every codewords and g_i is a codeword. Therefore

$$Gc^T = \begin{pmatrix} g_1 c^T \\ \vdots \\ g_k c^T \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$$

and equivalently $c \in ker(G)$.

Conversely let $c_1 \in ker(G)$. We want to show that c_1 is orthogonal to every codeword. For this let $c_2 \in \mathcal{C}$, then by definition 2.1.1 there exists $m \in \mathbb{F}_q^k$ such that

$$c_2 = mG.$$

But now

$$c_2 c_1^T = mGc_1^T = m(Gc_1^T) = 0,$$

where the last equality followed from c_1 being in the kernel of G. Hence $c_1 \in \mathcal{C}^{\perp}$. □

An important consequence of this lemma is that the generator matrix of a self-dual code is also a parity-check matrix of the given code as $\mathcal{C} = \mathcal{C}^{\perp} = ker(G)$.

Lemma 2.1.8. *[11] Let $\mathcal{C}$ be a (n, k) code with generator matrix G. Then*

$$n = dim(\mathcal{C}) + dim(\mathcal{C}^{\perp}).$$

Proof. By the Rank-Nullity Theorem

$$n = dim(rowspan(G)) + dim(ker(G))$$

holds and we conclude with $\mathcal{C} = rowspan(G)$ and $\mathcal{C}^{\perp} = ker(G)$. □

Notice that we did no use the fact of $\mathcal{C}$ being a linear block code, meaning that this lemma also holds in more generality, e.g. for convolutional codes.

Corollary 2.1.9. *[12] Any self-dual (n, k) code fulfils $n = 2k$.*

Proof. By the previous Lemma and $\mathcal{C} = \mathcal{C}^{\perp}$,

$$n = dim(\mathcal{C}) + dim(\mathcal{C}^{\perp}) = 2dim(\mathcal{C}) = 2k.$$

□

So from now on we may assume every self-dual code to be a $(n, n/2)$ code or equivalently a $(2k, k)$ code.

The next lemma is a well-known result on self-dual codes.

Lemma 2.1.10. *Let $\mathcal{C}$ be a $(2k, k)$ code over $\mathbb{F}_q$ with generator matrix G. Then the following statements are equivalent:*

(i) *$\mathcal{C}$ is self-dual;*

(ii) *$\mathcal{C}$ is self-orthogonal;*

(iii) *$GG^T = 0$;*

(iv) *G is a parity-check matrix of $\mathcal{C}$.*

Proof. $(i) \implies (ii)$ This is true by definition.

$(ii) \implies (iii)$ Let g_i, g_j be two rows of G, then they both are codewords and hence

$$g_i g_j^T = 0$$

as $\mathcal{C}$ is self-orthogonal. Consequently

$$GG^T = \begin{pmatrix} g_1 g_1^T & \cdots & g_1 g_k^T \\ \vdots & & \vdots \\ g_k g_1^T & \cdots & g_k g_k^T \end{pmatrix} = \begin{pmatrix} 0 & \cdots & 0 \\ \vdots & & \vdots \\ 0 & \cdots & 0 \end{pmatrix}.$$

$(iii) \implies (i)$ Let $c_1, c_2 \in \mathcal{C}$, then there exists $m_1, m_2 \in \mathbb{F}_q$ such that

$$c_1 = m_1 G \text{ and } c_2 = m_2 G.$$

All codewords are orthogonal to each other as

$$c_1 c_2^T = m_1 G (m_2 G)^T = m_1 (GG^T) m_2^T = 0$$

and therefore

$$\mathcal{C} \subset \mathcal{C}^\perp.$$

From $\mathcal{C}^\perp$ being a linear block code, we know that there exists a generator matrix H of $\mathcal{C}^\perp$. Combining this and $\mathcal{C} \subset \mathcal{C}^\perp$, we get

$$rowspan(G) \subset rowspan(H)$$

and hence every row g_i of G can be written as a linear combination of rows h_i of H

$$g_i = \sum_{j=1}^{k} u_{ij} h_j$$

for some $u_{ij} \in \mathbb{F}_q$ (up to k, because H has $dim(\mathcal{C}^\perp) = 2k - dim(\mathcal{C}) = k$ rows).

Let $U = (u_{ij})_{1 \le i,j \le k}$, then

$$UH = \begin{pmatrix} u_{11} & \cdots & u_{1k} \\ \vdots & & \vdots \\ u_{k1} & \cdots & u_{kk} \end{pmatrix} \begin{pmatrix} — & h_1 & — \\ & \vdots & \\ — & h_k & — \end{pmatrix} =$$

$$= \begin{pmatrix} u_{11}h_1 + \cdots + u_{1k}h_k \\ \vdots \\ u_{k1}h_1 + \cdots + u_{kk}h_k \end{pmatrix} = \begin{pmatrix} — & g_1 & — \\ & \vdots & \\ — & g_k & — \end{pmatrix} = G.$$

But now by elementary rank properties

$$k = rank(G) = rank(UH) \leq min\{rank(U), rank(H)\}$$

and we find that $rank(U) = k$ or equivalently U is invertible.

Conclusively H and G are equivalent and therefore

$$\mathcal{C} = \mathcal{C}^{\perp}.$$

So statements (i)-(iii) were shown to be equivalent and we just need to connect with (iv).

$(i) \Longrightarrow (iv)$ Let $\mathcal{C}$ be self-dual, then

$$\mathcal{C} = \mathcal{C}^{\perp} = ker(G)$$

by Lemma 2.1.7, making G a parity-check matrix of $\mathcal{C}$.

$(iv) \Longrightarrow (iii)$ Let G be a parity-check matrix of $\mathcal{C}$, then

$$rowspan(G) = \mathcal{C} = ker(G)$$

and hence

$$GG^T = 0.$$

□

The equivalence between (i) and (iii) makes checking for self-duality way easier as we only need to check k rows for being orthogonal to each other ($\sum_{i=1}^{k} i = \frac{k(k+1)}{2}$ equations) instead of having to check all q^k codewords for orthogonality ($\sum_{i=1}^{q^k} i = \frac{q^k(q^k+1)}{2}$ equations) and then all non-codewords if neither of them is orthogonal to all codewords (at most $(q^n - q^k)q^k$ equations).

Example 2.1.11. *We claim that*

$$G = \begin{pmatrix} 4 & 0 & 5 & 6 \\ 5 & 6 & 0 & 4 \end{pmatrix} \in \mathbb{F}_{11}[z]^{2\times 4}$$

generates a self-dual code.

Since G generates a $(2k, k)$ code, it is enough to check the rows for orthogonality (by the previous Lemma) and indeed

$$\begin{array}{lcccccc} (4,0,5,6)(4,0,5,6)^T & = & 16+25+36 & = & 77 & = & 0 \\ (4,0,5,6)(5,6,0,4)^T & = & 20+24 & = & 44 & = & 0 \\ (5,6,0,4)(5,6,0,4)^T & = & 25+36+16 & = & 77 & = & 0 \end{array}.$$

We conclude that $GG^T = 0$ and therefore G generates a self-dual code.

Definition 2.1.12. *Let $c_1 = (x_1, \ldots, x_n)$ and $c_2 = (y_1, \ldots, y_n)$ be elements of a code. Then their **Hamming distance** is defined to be*

$$d(c_1, c_2) := \{i | x_i \neq y_i\}$$

*and the **Hamming weight** of c_1 is $wt(c_1) := d(c_1, 0)$.*

As we will not introduce any other weights in this chapter, we will use the words “weight" and “Hamming weight" interchangeably.

Definition 2.1.13. *The **distance** d of a linear block code $\mathcal{C}$ is given by*

$$d(C) = min\left\{wt(c) \mid c \in \mathcal{C} \setminus \{0\}\right\}.$$

A commonly used fact in coding theory classes to determine the distance is the following.

Lemma 2.1.14. *Let $\mathcal{C}$ be a linear block code and H its parity-check matrix. Then $d(\mathcal{C})$ is equal to the minimal number of linearly dependent columns of H.*

Proof. Let d denote the minimal number of linearly dependent columns of H and $d(\mathcal{C})$ the distance of $\mathcal{C}$.

Assume by contradiction that $d(\mathcal{C}) < d$. There always exists $c \in \mathcal{C}$ such that $wt(c) = d(\mathcal{C})$. But H is a parity-check matrix and therefore

$$Hc^T = 0,$$

implying that there are $wt(c) < d$ linearly dependent columns, a contradiction to the minimality of d. Hence $d(\mathcal{C}) \geq d$.

To show the other inequality, having d linearly dependent columns suggests that there exists $c \in ker(H)$ with $wt(c) = d$ and

$$Hc^T = 0.$$

But $ker(H) = \mathcal{C}$ by definition and therefore $c \in \mathcal{C}$, implying $d(\mathcal{C}) \leq wt(c) = d$. □

2.2 Construction Methods

There are three popular construction methods for binary self-dual codes in the Building-up, Harada-Munemasa and recursive construction. The latter will not be presented, but interested readers may consult [9].

The first method we dive into, is the so called “Bulding-up construction" due to J.-L. Kim and Y.Lee.

Theorem 2.2.1 (Building-up construction). *[8] Let $\mathcal{C}$ be a binary self-dual $(2k, k)$ code and $G = (g_i)_{i\in\{1,\dots,k\}}$ its generator matrix, where g_i is the i-th row of G. Let $x \in \mathbb{F}_2^n$ be a binary vector with odd weight and define $y_i := xg_i^T$ for $1 \leq i \leq k$. Then*

$$\tilde{G} = \begin{pmatrix} 1 & 0 & x \\ y_1 & y_1 & \\ \vdots & \vdots & G \\ y_k & y_k & \end{pmatrix} \in \mathbb{F}_2^{(k+1)\times(2k+2)}$$

generates a binary self-dual $(2k+2, k+1)$ code $\tilde{\mathcal{C}}$.

Proof. We notice that $\tilde{\mathcal{C}}$ is a $(2(k+1), k+1)$ code and hence we may apply Lemma 2.1.10. So our goal is to show $\tilde{G}\tilde{G}^T = 0$.

Let $\tilde{g}_i$ be the i-th row of $\tilde{G}$ and g_i the i-th row of G. Then $\tilde{g}_1$ is

$$\tilde{g_1}\tilde{g_1}^T = 1^2 + xx^T = 1 + wt(x) = 0$$

orthogonal to itself and

$$\tilde{g_1}\tilde{g_{i+1}}^T = y_i + xg_i^T = xg_i^T + xg_i^T = 0$$

to every other row for $i \in \{1, \dots, k\}$.

Also, all the other rows are orthogonal to each other as

$$\tilde{g}_{j+1}\tilde{g}_{i+1}^T = y_jy_i + y_jy_i + g_jg_i^T = g_jg_i^T = 0,$$

where $1 \leq i, j \leq k$. The last equality followed, because G generates a self-dual code and equivalently $GG^T = 0$ (by Lemma 2.1.10).

Finally, we conclude that

$$\tilde{G}\tilde{G}^T = 0$$

and equivalently $\tilde{G}$ generates a self-dual code. □

J.-L.Kim went even one step further and established that all binary self-dual block codes with $d > 2$ can be found with this construction (allowing permutation of columns).

Theorem 2.2.2. *[8] Any binary self-dual code of length n with distance $d > 2$ can be obtained by some binary self-dual code of length $n - 2$ with the construction in Theorem 2.2.1 (up to column permutation).*

A proof of this fact can be found in [8].

Example 2.2.3.

$$G_1 = (g_1) = \begin{pmatrix} 1 & 1 \end{pmatrix}$$

is a binary self-dual $(2,1)$ code (actually the only one). Our goal is now to apply Theorem 2.2.1 and find other binary self-dual codes.

Let $x = (x_1, x_2) = (1,0)$, where $wt(x)$ is odd as desired. Then

$$y_1 = xg_1^T = x(1,1)^T = 1$$

and by Theorem 2.2.1

$$G_2 = \begin{pmatrix} 1 & 0 & x_1 & x_2 \\ y_1 & y_1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

generates a binary self-dual code.

Similarly, let f_1 be the first and f_2 the second row of G_2 and $x = (x_1, x_2, x_3, x_4) = (1,0,1,1)$, where again $wt(x)$ is odd. Then

$$y_1 = xf_1^T = x(1,0,1,0)^T = 0$$

and

$$y_2 = xf_2^T = x(1,1,1,1)^T = 1.$$

Applying Theorem 2.2.1, we conclude that

$$G_3 = \begin{pmatrix} 1 & 0 & x_1 & x_2 & x_3 & x_4 \\ y_1 & y_1 & 1 & 0 & 1 & 0 \\ y_2 & y_2 & 1 & 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

generates a binary self-dual code.

We have argued in the previous section that the generator matrix of a self-dual code is also a parity-check matrix. So to calculate the distance of the code $\mathcal{C}_3$ generated by G_3, we may use G_3 as a parity-check matrix and Lemma 2.1.14, which states that $d(\mathcal{C}_3)$ is equal to the minimal number of dependent columns of its parity-check matrix. By inspection, this number is 2 and hence $d(\mathcal{C}_3) = 2$.

We conclude that G_3 generates a binary self-dual $[6,3,2]$ code.

Finally, we present the “Harada-Munemasa construction".

Theorem 2.2.4 (Harada-Munemasa construction). *[7] Let G be the generator matrix of a binary self-dual $[2k,k,d]$ code $\mathcal{C}$. Then the matrix*

$$G_1 = \begin{pmatrix} a_1 & a_1 & \\ \vdots & \vdots & G \\ a_k & a_k & \end{pmatrix} \in \mathbb{F}_2^{k \times (2k+2)},$$

where $a_i \in \mathbb{F}_2$, generates a binary self-orthogonal $[2k+2, k, \geq d]$ code $\mathcal{C}_1$. Moreover, there exists $x \in \mathcal{C}_1^{\perp} \setminus \mathcal{C}_1$ such that

$$G_2 = \begin{pmatrix} \text{—} & x & \text{—} \\ a_1 & a_1 & \\ \vdots & \vdots & G \\ a_k & a_k & \end{pmatrix} \in \mathbb{F}_2^{(k+1) \times (2k+2)}$$

generates a binary self-dual $(2k+2, k+1)$ code.

Proof. Thanks to $GG^T = 0$, all the rows of G_1 are orthogonal to each other and as codewords are linear combinations of rows,

$$\mathcal{C}_1 \subset \mathcal{C}_1^{\perp},$$

follows, concluding the first part.

Furthermore, x is orthogonal to all other rows by definition of $\mathcal{C}_1^{\perp}$ and we only need to argue that there is a x that is also orthogonal to itself to conclude the self-duality of the code generated by G_2.

By the Rank-Nullity Theorem, $dim(\mathcal{C}_1^{\perp}) = 2k + 2 - dim(\mathcal{C}_1) = k + 2$, implying that

$$dim(\tilde{\mathcal{C}_1}^{\perp} / \tilde{\mathcal{C}_1}) = dim(\mathcal{C}_1^{\perp}) - dim(\mathcal{C}_1) = 2.$$

Let a and b be the two basis vectors of $\tilde{\mathcal{C}_1}^{\perp} / \tilde{\mathcal{C}_1}$. If neither a nor b have even weight, then

$$(a+b)(a+b)^T = aa^T + ab^T + ba^T + bb^T = aa^T + bb^T = 0,$$

proving that there always exists a $x \in \tilde{\mathcal{C}}^{\perp} / \tilde{\mathcal{C}}$ such that $wt(x) = xx^T = 0$. But now after choosing this x of even weight, all rows of G_2 are orthogonal, which lets us conclude with Lemma 2.1.10 that G_2 generates a self-dual code. □

Theorem 2.2.5. *Any binary self-dual code of length n with distance $d > 2$ can be obtained by some binary self-dual code of length $n-2$ with the construction in Theorem 2.2.4 (up to column permutation).*

A proof of this fact can be found in [7].

Example 2.2.6. *In Example 2.2.3 we have seen that*

$$G_3 = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

generates a binary self-dual $(6, 3)$ code.

We will use the Harada-Munemasa construction to find a binary self-dual $(8, 4)$ code.

Let $a_1 = a_3 = 1$ and $a_2 = 0$. Then by Theorem 2.2.4,

$$G_4 = \begin{pmatrix} a_1 & a_1 & 1 & 0 & 1 & 0 & 1 & 1 \\ a_2 & a_2 & 0 & 0 & 1 & 0 & 1 & 0 \\ a_3 & a_3 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

generates a self-orthogonal code.
We now want to find a $x = (x_1, ..., x_8) \in \mathbb{F}_2^8$ that is orthogonal to every row of G_4 and itself or equivalently

$$\begin{array}{rcccl} x(1,1,1,0,1,0,1,1)^T & = & x_1 + x_2 + x_3 + x_5 + x_7 + x_8 & = & 0 \\ x(0,0,0,0,1,0,1,0)^T & = & x_5 + x_7 & = & 0 \\ x(1,1,1,1,1,1,1,1)^T & = & x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 & = & 0 \\ xx^T & = & x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 & = & 0 \end{array}.$$

Consequently $x_5 = x_7$ and plugging this into the first and third condition we find

$$\begin{array}{rcccl} x(1,1,1,0,1,0,1,1)^T & = & x_1+x_2+x_3+x_8 & = & 0 \\ x(1,1,1,1,1,1,1,1)^T & = & x_1+x_2+x_3+x_4+x_6+x_8 & = & 0 \end{array}.$$

Finally by combining these two conditions, we find $x_4 = x_6$ and $x_1 + x_2 + x_3 + x_8 = 0$ as the last two restrictions on x.

Therefore $x = (1,1,0,1,0,1,0,0)$ is a valid choice and indeed

$$G_5 = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

generates a binary self-dual code. This may be verified by checking $G_5 G_5^T = 0$.

By inspection, 2 is the minimal number of dependent columns of G_5 and hence $d(\mathcal{C}_5) = 2$.

We conclude that $\mathcal{C}_5$ is a binary self-dual $[8,4,2]$ code.

An interesting difference between the two constructions is that the Building-up construction chooses the row first and then simply calculates the entries of the new columns based on the chosen row, whereas the Harada-Munemasa construction chooses the columns first and then determines the to be added row based on it being orthogonal to all other rows plus itself.

3 Convolutional Codes

In this section, we start off by introducing the idea of a convolutional code. Further, we will study non-catastrophic codes, whereby a very powerful result on some equivalent conditions will be presented. Finally, we will connect non-catastrophic codes to self-dual codes and present equivalent properties to self-duality, which will be used to classify and construct in later chapters.

3.1 Prelimenaries

Let $\mathbb{F}_q$ denote the finite field with q elements and $R := \mathbb{F}_q[z]$ its polynomial ring.

Definition 3.1.1. *A (n, k) **convolutional code** is a R-submodule of R^n with rank k.*

R is a PID and Modules over a PID always admit a basis. Therefore, one always finds a $G(z) \in R^{k\times n}$ such that $\mathcal{C} = rowspan(G(z))$. This $G(z)$ is called a **Generator Matrix** of $\mathcal{C}$.

Definition 3.1.2. *Let $H(z) \in R^{n-k\times n}$ such that*

$$\mathcal{C} = ker(H(z)).$$

*Then we call $H(z)$ a **Parity-Check Matrix** of $\mathcal{C}$.*

Other than for block codes, there are some convolutional codes that do not admit a parity-check matrix. More details on the specific characteristics that permit the existence of a parity-check matrix will be presented in chapter 3.2.

Definition 3.1.3. *We say that a $k \times k$ matrix $U(z)$ with entries in R is **unimodular** if there exists another $k \times k$ matrix $V(z)$ with entries in R such that*

$$V(z)U(z) = I_k.$$

Definition 3.1.4. *Two generator matrices $G(z) \in R^{k\times n}$ and $\tilde{G}(z) \in R^{k\times n}$ are **equivalent** if there exists an unimodular matrix $U(z) \in R^{k\times k}$ such that*

$$G(z) = U(z)\tilde{G}(z).$$

Equivalent generator matrices generate the same code.

Unimodular matrices will be of major importance in the following chapters, mostly in determining the Smith form (to be introduced in 3.2.4) or finding equivalent generator matrices. Consequently, we will have a closer look at which row or column operations are done vian unimodular matrices.

First, we observe that $U(z) \in R^{k\times k}$ is unimodular if and only if $det(U(z)) \in R^*$. This is a consequence of elementary properties of the determinant in

$$1 = det(I_k) = det(U(z)U(z)^{-1}) = det(U(z))det(U(z)^{-1}) = det(U(z))det(U(z))^{-1}.$$

Definition 3.1.5. *We say that a row or column operation is **unimodular** if it is done via multiplying by an unimodular matrix.*

There are some commonly known unimodular operations that will be used in the further studies.

Remark 3.1.6. *The following row and column operations are all unimodular:*

(i) Exchanging rows;

(ii) Exchanging columns;

(iii) Taking a row $r_i(z)$ and adding $f(z)r_i(z)$ to another row for any $f(z) \in \mathbb{F}_q[z]$.;

(iv) Taking a column $c_i(z)$ and adding $f(z)c_i(z)$ to another column for any $f(z) \in \mathbb{F}_q[z]$.

The determinant of a matrix product is equal to the product of their determinants, making the matrix product unimodular whenever both matrices are. Therefore any sequence of unimodular operations is still unimodular.
An additional observation is that (i) and (iii) are both operations that can be used on a generator matrix, resulting in an equivalent generator matrix.

Furthermore, we introduce a notion that in a way tells us how “far away" a convolutional code is from a block code.

Definition 3.1.7. *The **degree** δ of a convolutional code is defined to be the highest degree of any k-th minor of its generator matrix.*

As the generator matrices of two equivalent codes differ by the multiplication of an unimodular matrix, which has a constant determinant, the degree δ is the same for all equivalent generator matrices, making this a valid definition.

Remark 3.1.8. *The convolutional codes of degree 0 are essentially the linear block codes.*

Definition 3.1.9. *Let $\mathcal{C}$ be a convolutional code generated by $G(z) = (g_{ij}(z))_{1 \leq i \leq k, 1 \leq j \leq n}$. Then*

$$\delta_i := max\left\{deg(g_{ij}(z)) \mid j \in \{1, ..., n\}\right\}$$

*is called the **i-th row degree**.*

Remark 3.1.10. *Let δ be the degree and δ_i the row degrees of a convolutional code, then*

$$\delta \leq \sum_{i=1}^{k} \delta_i.$$

The row degrees of two equivalent matrices may be different, but an generator matrix fulfilling

$$\delta = \sum_{i=1}^{k} \delta_i$$

can always be found. This generator matrix is called a **minimal encoder**.

We say that $\mathcal{C}$ is a $[n, k, \delta]$ convolutional code if $\mathcal{C}$ has length n, dimension k and degree δ. An alternative way to describe the dimension and length is with the rate of the code, written as k/n.

Definition 3.1.11. *The **weight** of a polynomial vector*

$$c(z) = \sum_{i=0}^{deg(c(z))} c_i z^i \in \mathbb{F}_q[z]^n$$

is defined to be

$$wt(c(z)) := \sum_{i=0}^{deg(c(z))} wt_{Ham}(c_i),$$

where $wt_{Ham}(c_i(z))$ denotes the Hamming weight of $c_i(z)$ (as defined for block codes in 2.1.12).

In other words, the weight of a polynomial vector is just the sum of all non-zero coefficients of all its entries.

Definition 3.1.12. *The **free distance** of a convolutional code $\mathcal{C}$ is given by*

$$d_{free}(\mathcal{C}) = min\left\{wt(c(z)) \mid c(z) \in \mathcal{C} \setminus \{0\}\right\}.$$

3.2 Non-Catastrophic Codes

Definition 3.2.1. *A polynomial matrix $G(z) \in R^{k\times n}$ is said to be **left-prime** if in all factorizations*

$$G(z) = A(z)\tilde{G}(z), \text{ with } A(z) \in R^{k\times k} \text{ and } \tilde{G}(z) \in R^{k\times n},$$

the left factor $A(z)$ is unimodular.

If one generator matrix is left-prime, then all equivalent generator matrices are also left-prime as they differ by left-multiplication of an unimodular matrix. And conversely if it doesn't hold for one, then it holds for none. We call codes that admit a left-prime generator matrix **non-catastrophic** and **catastrophic** if they do not.

Example 3.2.2.

$$G(z) = \begin{pmatrix} z+1 & z & z^2+z+1 \\ z & 0 & z^2+z \end{pmatrix} \in \mathbb{F}_2[z]^{2\times 3}$$

generates a catastrophic code, because

$$G(z) = \begin{pmatrix} 1 & z+1 \\ z & z \end{pmatrix} \begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & z \end{pmatrix}$$

is a factorisation, but the left factor is not unimodular as

$$det\left[\begin{pmatrix} 1 & z+1 \\ z & z \end{pmatrix}\right] = z^2 \notin \mathbb{F}_2^*$$

and therefore any equivalent generator matrix

$$\tilde{G}(z) = U(z)G(z)$$

is not left-prime, since

$$\tilde{G}(z) = \left[U(z)\begin{pmatrix} 1 & z+1 \\ z & z \end{pmatrix}\right]\begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & z \end{pmatrix}$$

they have a not unimodular left-factor.

We will see way more practical methods to check whether codes are non-catastrophic in Theorem 3.2.6, where we will also be proving that being non-catastrophic is a characteristic of the code, but first we introduce some necessary notions.

Definition 3.2.3. *Let $G(z) \in R^{k\times n}$ with $k \leq n$. Then there exists an unimodular matrix $U(z) \in R^{k\times k}$ such that*

$$H(z) = U(z)G(z) = \begin{pmatrix} h_{11}(z) & h_{12}(z) & \cdots & h_{1k}(z) & h_{1k+1}(z) & \cdots & h_{12k}(z) \\ & h_{22}(z) & \cdots & h_{2k}(z) & h_{2k+1}(z) & \cdots & h_{22k}(z) \\ & & \ddots & \vdots & \vdots & & \vdots \\ & & & h_{kk}(z) & h_{kk+1}(z) & \cdots & h_{k2k}(z) \end{pmatrix},$$

*where $(h_{ii}(z))_{i=1}^{k}$ are monic polynomials such that $deg(h_{ii}(z)) > deg(h_{ji}(z))$ for $i > j$. H(z) is called the (unique) **column Hermite Form** of $G(z)$.*

The uniqueness of the column Hermite Form gives us the option to characterize equivalent generator matrices by their respective column Hermite Form.

Definition 3.2.4. *Let $G(z) \in R^{k\times n}$ with $k \leq n$. Then there exists an unimodular matrix $V(z) \in R^{n\times n}$ such that*

$$H(z) = G(z)V(z) = \begin{pmatrix} h_{11}(z) & & & & 0 & \cdots & 0 \\ h_{21}(z) & h_{22}(z) & & & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & & \vdots & & \vdots \\ h_{k1}(z) & h_{k2}(z) & \cdots & h_{kk}(z) & 0 & \cdots & 0 \end{pmatrix},$$

*where $(h_{ii}(z))_{i=1}^{k}$ are monic polynomials such that $deg(h_{ii}(z)) > deg(h_{ij}(z))$ for $i > j$. H(z) is called the (unique) **row Hermite Form** of $G(z)$.*

Definition 3.2.5. *Let $G(z) \in R^{k\times n}$ with $k \leq n$. Then there exist two unimodular matrices $U(z) \in R^{k\times k}$ and $V(z) \in R^{n\times n}$ such that*

$$S(z) = U(z)G(z)V(z) = \begin{pmatrix} \gamma_1(z) & & & & 0 & \cdots & 0 \\ & \gamma_2(z) & & & 0 & \cdots & 0 \\ & & \ddots & & \vdots & & \vdots \\ & & & \gamma_k(z) & 0 & \cdots & 0 \end{pmatrix},$$

*where $(\gamma_i(z))_{i=1}^{k}$ are monic polynomials such that $\gamma_{i+1}(z)|\gamma_i(z)$ for all $i = 1, ..., k-1$. $S(z)$ is called the (unique) **Smith form** of $G(z)$.*

Now we are ready to present the key result, that will be used in later chapters to determine whether a code is non-catastrophic.

Theorem 3.2.6. *[1, 2] Let $R = \mathbb{F}_q[z]$, $\mathcal{C}$ a convolutional code and $G(z) \in R^{k\times n}$ its generator matrix, with $k \leq n$. The following are equivalent:*

1. *$\mathcal{C}$ is non-catastrophic;*
2. *$G(z)$ is left-prime;*
3. *the Smith form of $G(z)$ is $[I_k\ 0]$;*
4. *the row Hermite form of $G(z)$ is $[I_k\ 0]$;*
5. *the ideal generated by all the k-th minors of $G(z)$ is R;*
6. *there exists a parity-check matrix $H(z)$ of $\mathcal{C}$;*

7. *$G(z)$ can be completed to an unimodular matrix, e.g. there exists $L(z) \in R^{n-k\times n}$ such that $\left[\begin{smallmatrix} G(z) \\ L(z) \end{smallmatrix}\right]$ is unimodular.*

Proof. Our approach will be the following: First, we clarify our previous argument for $1.$ and $2.$ being equivalent statements, then we show that $3., 4.$ and $5.$ are equivalent statements, and connect $(1.) - (5.)$. Finally, we connect the already shown to $7.$ and show $6.$ and $7.$ to be equivalent.

$(1. \implies 2.)$ Let $\mathcal{C}$ be non-catastrophic, or equivalently let there exist a generator matrix $G'(z) \in R^{k\times n}$ that is left-prime. Furthermore, $G(z)$ and $G'(z)$ are equivalent generator matrices and hence

$$G'(z) = U(z)G(z)$$

for some unimodular $U(z)$.

Assume conversely that $G(z)$ is not left-prime. Then there exists a factorisation

$$G(z) = A(z)\tilde{G}(z),$$

where $A(z)$ is not unimodular. But now

$$G'(z) = U(z)G(z) = (U(z)A(z))\tilde{G}(z)$$

is a factorisation of $G'(z)$, whereby the left factor $(U(z)A(z))$ is not unimodular, contradicting left-primeness of $G'(z)$.

$(2. \implies 1.)$ Let $G(z)$ be left-prime, then $\mathcal{C}$ is non-catastrophic by definition.

$(5. \implies 4.)$ Let $H(z) = G(z)V(z)$ be $G(z)$'s row Hermite form for some unimodular $V(z) \in R^{n\times n}$. Then $H(z)$'s k-th minors generate the same ideal as $G(z)$'s k-th minors, which generate R. Therefore, since

$$det\left[\begin{pmatrix} h_{11}(z) & & & \\ h_{21}(z) & h_{22}(z) & & \\ \vdots & \vdots & \ddots & \\ h_{k1}(z) & h_{k2}(z) & \cdots & h_{kk}(z) \end{pmatrix}\right] = \prod_{i=1}^{k} h_{ii}(z)$$

is the only non-zero k-th minor of $H(z)$, all $h_{ii}(z)$'s must be constants. Furthermore, the $h_{ii}(z)$'s are monic by definition and hence $h_{i,i}(z) = 1$. Finally, as all non-diagonal entries of $H(z)$ can be eliminated with unimodular column operations, we conclude that

$$H(z) = [I_k\ 0]$$

is the row Hermite form of $G(z)$.

$(4. \implies 3.)$ If $G(z)$ has row Hermite form $[I_k\ 0]$, then its Smith form is also $[I_k\ 0]$.

$(3. \implies 5.)$ Let $[I_k\ 0]$ be the Smith form of $G(z)$ or equivalently there exist unimodular matrices $U(z) \in R^{k\times k}$ and $V(z) \in R^{n\times n}$ such that

$$[I_k\ 0] = U(z)G(z)V(z).$$

Then the ideal generated by $G(z)$'s k-th minors is the same as the ideal generated by $[I_k\ 0]$'s k-th minors. Finally, the ideal generated by the minors of $[I_k\ 0]$ is R, concluding this part.

$(5. \implies 2.)$ We will show the contrapositive. Let $G(z)$ not be left-prime. Then there exists a factorisation with $A(z) \in R^{k\times k}$ and $\tilde{G}(z) \in R^{k\times n}$ such that

$$G(z) = A(z)\tilde{G}(z),$$

where $det(A(z)) \notin R^*$. But $det(A(z))$ is a factor of every k-th minor, implying that the ideal generated by the set of k-th minors M is not R or rather

$$< M > \subset < det(A(z)) > \subsetneq R.$$

$(2. \implies 5.)$ We again will show the contrapositive. Let $S(z) \neq [I_k\ 0]$ be the Smith form of $G(z)$. Then by definition, there exist unimodular $U(z) \in R^{k\times k}$ and $V(z) \in R^{n\times n}$ such that

$$U(z)G(z)V(z) = S(z) = \tilde{S}(z)[I_k\ 0],$$

where $\tilde{S}(z)$ is just the first k columns of $S(z)$. Equivalently

$$G(z) = (U(z)^{-1}\tilde{S}(z))([I_k\ 0]V(z)^{-1}).$$

But as $det(\tilde{S}(z)) \notin R^*$, we know that $(U(z)^{-1}\tilde{S}(z))$ is a not unimodular left factor of a factorisation of $G(z)$, making $G(z)$ not left-prime.

At this point, $(1.) - (5.)$ are established to be equivalent statements.

$(4. \implies 7.)$ Since $G(z)$'s row Hermite Form is $[I_k\ 0]$, we know there to exist an unimodular $V(z) \in R^{n\times n}$ such that

$$G(z)V(z) = [I_k\ 0].$$

As $V(z)$ is unimodular, we know there also exists $V(z)^{-1}$ such that

$$V(z)^{-1}V(z) = \begin{pmatrix} I_k & 0 \\ 0 & I_{n-k} \end{pmatrix} = \begin{pmatrix} G(z)V(z) \\ L(z)V(z) \end{pmatrix},$$

where $L(z)$ is the $n-k \times n$ matrix containing the last $n-k$ rows of $V(z)^{-1}$. But now

$$\begin{pmatrix} G(z) \\ L(z) \end{pmatrix} V(z) = \begin{pmatrix} G(z)V(z) \\ L(z)V(z) \end{pmatrix} = I_n$$

$G(z)$ can be unimodularly completed.

$(7. \implies 4.)$ Let $L(z) \in R^{n-k\times n}$ be an unimodular completion of $G(z)$. Then there exist $N(z) \in R^{k\times n}$ and $H(z) \in R^{n-k\times n}$ such that

$$\begin{pmatrix} G(z) \\ L(z) \end{pmatrix} \begin{pmatrix} N(z)^T & H(z)^T \end{pmatrix} = \begin{pmatrix} G(z)N(z)^T & G(z)H(z)^T \\ L(z)N(z)^T & L(z)H(z)^T \end{pmatrix} = \begin{pmatrix} I_k & 0 \\ 0 & I_{n-k} \end{pmatrix}$$

and hence

$$G(z) \begin{pmatrix} N(z)^T & H(z)^T \end{pmatrix} = [I_k\ 0]$$

is the row Hermite form of $G(z)$.

$(7. \implies 6.)$ Let $L(z) \in R^{n-k \times n}$ complete $G(z)$ unimodularly. Then there exist $N(z) \in R^{k \times n}$ and $H(z) \in R^{n-k \times n}$ such that

$$\begin{pmatrix} G(z) \\ L(z) \end{pmatrix} \begin{pmatrix} N(z)^T & H(z)^T \end{pmatrix} = \begin{pmatrix} G(z)N(z)^T & G(z)H(z)^T \\ L(z)N(z)^T & L(z)H(z)^T \end{pmatrix} = \begin{pmatrix} I_k & 0 \\ 0 & I_{n-k} \end{pmatrix}.$$

We now claim and show that $H(z)$ is a parity-check matrix of the code generated by $G(z)$ or rather

$$rowspan(G(z)) = ker(H(z)). \tag{1}$$

Let $c(z) \in \mathcal{C} = rowspan(G(z))$, then there exists $m(z) \in R^k$ such that $c(z) = m(z)G(z)$ and hence

$$H(z)c(z)^T = H(z)(m(z)G(z))^T = H(z)G(z)^T m(z)^T = 0,$$

which lets us conclude that $\mathcal{C} \subset ker(H(z))$.

Conversely, let $f(z) \in ker(H(z)$, then $f(z)H(z)^T = 0$ by definition. Furthermore,

$$f(z) = f(z)I_n = f(z) \begin{pmatrix} N(z)^T & H(z)^T \end{pmatrix} \begin{pmatrix} G(z) \\ L(z) \end{pmatrix} = \begin{pmatrix} f(z)N(z)^T & f(z)H(z)^T \end{pmatrix} \begin{pmatrix} G(z) \\ L(z) \end{pmatrix} =$$

$$= \begin{pmatrix} f(z)N(z)^T & 0 \end{pmatrix} \begin{pmatrix} G(z) \\ L(z) \end{pmatrix} = (f(z)N(z)^T)G(z)$$

and therefore $f(z) \in rowspan(G(z)) = \mathcal{C}$.

$(6. \implies 7.)$ This proof is due to [4, Thm. 1.2.8].

Let $H(z)$ be a parity-check matrix of $\mathcal{C}$. Then $H(z)$ is not necessarily left-prime, but there exist $X(z) \in R^{n-k \times n-k}$ with full rank and $\tilde{H}(z) \in R^{n-k \times n}$ left-prime such that

$$H(z) = X(z)\tilde{H}(z).$$

Notice that necessarily $\mathcal{C} = ker(H(z)) = ker(\tilde{H}(z))$. Furthermore, $\tilde{H}(z)$ being left-prime implies $(2. \implies 7.)$ that there exist $L(z) \in R^{k \times n}$, $N(z) \in R^{n-k \times n}$ and $M(z) \in R^{k \times n}$ such that

$$\begin{pmatrix} \tilde{H(z)} \\ L(z) \end{pmatrix} \begin{pmatrix} N(z)^T & M(z)^T \end{pmatrix} = I_n$$

or rather

$$\begin{pmatrix} M(z) \\ N(z) \end{pmatrix} \begin{pmatrix} L(z)^T & \tilde{H}(z)^T \end{pmatrix} = I_n.$$

But we have shown in (1) that in this case $rowspan(M(z)) = ker(\tilde{H}(z))$ and therefore

$$\mathcal{C} = ker(\tilde{H}(z)) = rowspan(M(z)),$$

concluding that one generator matrix of $\mathcal{C}$ in $M(z)$ can be unimodularly completed.

Now let $G(z)$ be any generator matrix of $\mathcal{C}$, then there exists an unimodular $U(z)$ such that

$$G(z) = U(z)M(z).$$

But now

$$\begin{pmatrix} G(z) \\ N(z) \end{pmatrix}$$

is still an unimodular completion as

$$\begin{pmatrix} L(z)^T U(z)^{-1} & \tilde{H}(z)^T \end{pmatrix}$$

is its inverse.

This conludes the proof as $6.$ and $7.$ have been successfully shown to be equivalent to the first five statements. □

Corollary 3.2.7. *Let $G(z)$ generate a non-catastrophic convolutional code, then for any unimodular $U(z) \in R^{k\times k}$ and $V(z) \in R^{n\times n}$,*

$$U(z)G(z)V(z)$$

also generates a non-catastrophic code. Or in other words, a non-catastrophic code stays non-catastrophic under unimodular operations.

This corollary is due to the fact that the Smith normal form does not get changed after multiplying with unimodular matrices. It further shows that we may use unimodular operations on a generator matrix to make it easier to determine either of the properties presented in Theorem 3.2.6.

Definition 3.2.8. *For two matrices $G_1(z), G_2(z) \in R^{k\times n}$, we write*

$$G_1(z) \sim G_2(z),$$

if there exist unimodular matrices $A(z) \in R^{k\times k}$ and $B(z) \in R^{n\times n}$ such that

$$G_1(z) = A(z)G_2(z)B(z).$$

This is similar to the condition that is sometimes used to define equivalent codes, but we will apply it with a different purpose here. It will mainly be used in the calculation of the Smith form or in other characterising properties of non-catastrophic codes as done in the following example.

Example 3.2.9. *We've shown in Example 3.2.2 that*

$$G(z) = \begin{pmatrix} z+1 & z & z^2+z+1 \\ z & 0 & z^2+z \end{pmatrix} \in \mathbb{F}_2[z]^{2\times 3}$$

generates a catastrophic code by finding a factorisation, where the left factor is not unimodular. Finding such a left factor can be rather hard. In fact, we did not find the left factor, but we built $G(z)$ by choosing a not unimodular left factor and a random right factor.
So let's show that $G(z)$ is not left-prime in an easier fashion: By computing its Smith form.

$$\begin{pmatrix} z+1 & z & z^2+z+1 \\ z & 0 & z^2+z \end{pmatrix} \overset{R_1'=R_1+R_2}{\sim} \begin{pmatrix} 1 & z & 1 \\ z & 0 & z^2+z \end{pmatrix} \overset{R_2'=zR_1+R_2}{\sim} \begin{pmatrix} 1 & z & 1 \\ 0 & z^2 & z^2 \end{pmatrix} \sim$$

$$\overset{\substack{C_2'=zC_1+C_2 \\ C_3'=C_1+C_3}}{\sim} \begin{pmatrix} 1 & 0 & 0 \\ 0 & z^2 & z^2 \end{pmatrix} \overset{C_3'=C_2+C_3}{\sim} \begin{pmatrix} 1 & 0 & 0 \\ 0 & z^2 & 0 \end{pmatrix} \sim \begin{pmatrix} z^2 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}.$$

Therefore $G(z)$ does not have Smith form $[I_k\ 0]$ and equivalently by Theorem 3.2.6, $G(z)$ generates a catastrophic code.

Remark 3.2.10. *$\sim$ is an equivalence relation.*

3.3 Self-Dual Convolutional Codes

We proceed to the heart of the third chapter, where some basic theory about self-dual convolutional codes will be presented.

Definition 3.3.1. $\mathcal{C}^{\perp} = \left\{ f(z) \in R^n \mid f(z)c(z)^T = 0 \quad \forall c(z) \in C \right\}$ *is called the* ***dual*** *of a convolutional code* $\mathcal{C}$*.*

The dual of a block code is always a linear block code and a similar statement is true for convolutional codes.

Lemma 3.3.2. *[5] The dual of a convolutional code* $\mathcal{C}$ *is also a convolutional code.*

Proof. We will show that $\mathcal{C}^{\perp}$ fulfils Definition 3.1.1 and indeed is a R-submodule of R^n.

Let $c_1(z), c_2(z) \in \mathcal{C}^{\perp}$ and $c(z) \in \mathcal{C}$. Then

$$(c_1(z) + c_2(z))c(z)^T = c_1(z)c(z)^T + c_2(z)c(z)^T = 0$$

and equivalently $\mathcal{C}^{\perp}$ is closed under addition. Further,

$$(0, \ldots, 0)c(z)^T = 0$$

the additive identity $(0, \ldots, 0)$ is in $\mathcal{C}^{\perp}$ and

$$-c_1(z)c(z)^T = (-1)(c_1(z)c(z)^T) = 0$$

the additive inverse $-c_1(z)$ of $c_1(z)$ is in $\mathcal{C}^{\perp}$.

Finally, commutativity and associativity follow from having the same additive operation as $(R^n, +)$ and being a subset of R^n. We conclude that $(\mathcal{C}^{\perp}, +)$ forms an abelian group.

The multiplicative operation is given by

$$\begin{array}{rccc} \cdot : & R \times \mathcal{C}^{\perp} & \longrightarrow & \mathcal{C}^{\perp} \\ & (r, c_1(z)) & \longmapsto & rc_1(z) \end{array} .$$

This is a well-defined map as

$$(r \cdot c_1(z))c(z)^T = r(c_1(z)c(z)^T) = 0$$

or rather $rc_1(z) \in \mathcal{C}^{\perp}$.

Finally, $\cdot$ fulfils all the necessary requirements in

$$\begin{array}{rcl} r \cdot (c_1(z) + c_2(z)) & = & r \cdot c_1(z) + r \cdot c_2(z) \\ (r + s) \cdot c_1(z) & = & r \cdot c_1(z) + s \cdot c_1(z) \\ (rs) \cdot c_1(z) & = & r \cdot (s \cdot c_1(z)) \\ 1 \cdot c_1(z) & = & c_1(z) \end{array}$$

for $r, s \in R$.

We conclude that $\mathcal{C}^{\perp}$ indeed is a convolutional code. □

Definition 3.3.3. *Two vectors $u(z), v(z) \in R^n$ are said to be **orthogonal** if*

$$u(z)v(z)^T = 0.$$

Not only does the dual code always have a generator matrix (Lemma 3.3.2), but the dual code also has a parity-check matrix as established in the following Lemma.

Lemma 3.3.4. *Let*

$$G(z) = \begin{pmatrix} — & g_1(z) & — \\ & \vdots & \\ — & g_k(z) & — \end{pmatrix}$$

be the generator matrix of a convolutional code $\mathcal{C}$, then

$$\mathcal{C}^\perp = ker(G(z))$$

or equivalently $G(z)$ is a parity-check matrix of $\mathcal{C}^\perp$.

Proof. We first show $\mathcal{C}^\perp \subset ker(G(z))$.

Let $c(z) \in \mathcal{C}^\perp$, then $g_i(z)c(z)^T = 0$ as $g_i(z)$ is a codeword and $c(z)$ is orthogonal to all codewords. Therefore

$$G(z)c(z)^T = \begin{pmatrix} g_1(z)c(z)^T \\ \vdots \\ g_k(z)c(z)^T \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$$

and equivalently $c(z) \in ker(G(z))$.

Secondly, we show $ker(G(z)) \subset \mathcal{C}^\perp$.

Let $f(z) \in ker(G(z))$ and $c(z) \in \mathcal{C}$, then $c(z) = m(z)G(z)$ for some $m(z) \in \mathbb{F}_q[z]^k$. We know by the definition 3.1.1 of a convolutional code that such an $m(z)$ must exist. But now

$$c(z)f(z)^T = (m(z)G(z))f(z)^T = m(z)(G(z)f(z)^T) = 0$$

as $f(z)$ is in the kernel of $G(z)$ and we conclude that $f(z)$ is orthogonal to every codeword or equivalently $f(z) \in \mathcal{C}^\perp$. □

So it is true that any generator matrix of the code is a parity-check matrix of the dual code. Consequently, the dual is always non-catastrophic by Theorem 3.2.6.

Lemma 3.3.5. *Let $\mathcal{C}$ be a (n, k) convolutional code with generator matrix $G(z)$. Then*

$$\mathcal{C} \text{ is non-catastrophic}$$

if and only if

$$\mathcal{C} = (\mathcal{C}^\perp)^\perp.$$

Proof. ($\Longrightarrow$) Let $c(z) \in \mathcal{C}$ and $c_1(z) \in \mathcal{C}^\perp$, then by definition

$$c_1(z)c(z)^T = 0$$

and hence

$$\mathcal{C} \subset (\mathcal{C}^\perp)^\perp.$$

As $(\mathcal{C}^\perp)^\perp$ again is a convolutional code, there exists a generator matrix $\tilde{G}(z)$. Combining this with the previous inclusion, we find

$$rowspan(G(z)) \subset rowspan(\tilde{G}(z)).$$

Therefore, there exist $u_{ij}(z) \in R$ with $1 \leq i, j \leq k$ such that

$$g_i(z) = \sum_{j=1}^{k} u_{ij}(z)\tilde{g}_j(z),$$

where $g_i(z)$ is the i-th row of $G(z)$ and $\tilde{g}_i(z)$ is the i-th row of $\tilde{G}(z)$. This sum goes up to k, because $(\mathcal{C}^\perp)^\perp$ has the same dimension as $\mathcal{C}$,

$$dim((\mathcal{C}^\perp)^\perp) = dim((\mathcal{C}^\perp)^\perp) + dim(\mathcal{C}^\perp) - dim(\mathcal{C}^\perp) = n - dim(\mathcal{C}^\perp) = dim(\mathcal{C}) = k,$$

thanks to the Rank-Nullity Theorem for codes (Lemma 2.1.8).

Let $U(z) = (u_{ij}(z))_{1 \leq i,j \leq k}$, then

$$U(z)\tilde{G}(z) = \begin{pmatrix} u_{11}(z) & \cdots & u_{1k}(z) \\ \vdots & & \vdots \\ u_{k1}(z) & \cdots & u_{kk}(z) \end{pmatrix} \begin{pmatrix} — & \tilde{g}_1(z) & — \\ & \vdots & \\ — & \tilde{g}_k(z) & — \end{pmatrix} =$$

$$= \begin{pmatrix} — & u_{11}(z)\tilde{g}_1(z) + \cdots + u_{1k}(z)\tilde{g}_k(z) & — \\ & \vdots & \\ — & u_{k1}(z)\tilde{g}_1(z) + \cdots + u_{kk}(z)\tilde{g}_k(z) & — \end{pmatrix} = \begin{pmatrix} — & g_1(z) & — \\ & \vdots & \\ — & g_k(z) & — \end{pmatrix} = G(z).$$

This means that $U(z)\tilde{G}(z)$ is a factorisation of $G(z)$. But $G(z)$ is left-prime, because $\mathcal{C}$ was assumed to be non-catastrophic, and therefore the left-factor $U(z)$ is unimodular.

We conclude that $G(z)$ and $\tilde{G}(z)$ are equivalent and thereby

$$\mathcal{C} = (\mathcal{C}^\perp)^\perp.$$

($\Longleftarrow$) We know $\mathcal{C}^\perp$ to be a convolutional code thanks to Lemma 3.3.2 and therefore it has a generator matrix $G(z)$. But now

$$(\mathcal{C}^\perp)^\perp = ker(G(z))$$

by the previous lemma, implying with Theorem 3.2.6 that $(\mathcal{C}^\perp)^\perp = \mathcal{C}$ is non-catastrophic. □

Definition 3.3.6. *We say that a convolutional code $\mathcal{C}$ is **self-orthogonal** if $\mathcal{C} \subset \mathcal{C}^\perp$ and **self-dual** if $\mathcal{C} = \mathcal{C}^\perp$.*

From this definition the following Lemma immediately follows.

Lemma 3.3.7. *Let a (n, k) convolutional code $\mathcal{C}$ with generator matrix $G(z)$ be self-dual, then $\mathcal{C}$ is non-catastrophic.*

Proof. We just showed in Lemma 3.3.4 that $\mathcal{C}^\perp = ker(G(z))$ and now self-duality lets us conclude with

$$\mathcal{C} = \mathcal{C}^\perp$$

that $G(z)$ is a parity-check matrix of $\mathcal{C}$. By Theorem 3.2.6, this means that $\mathcal{C}$ is non-catastrophic. □

Lemma 3.3.8. *[5] If $\mathcal{C}$ is a self-dual (n,k) convolutional code, then $n = 2k$.*

Proof. By the Rank-Nullity Theorem,

$$n = dim(rowspan(G(z))) + dim(ker(G(z))$$

and as $\mathcal{C} = rowspan(G(z))$ and $\mathcal{C}^{\perp} = ker(G(z))$, we find

$$n = dim(\mathcal{C}) + dim(\mathcal{C}^{\perp}) = 2dim(\mathcal{C}) = 2k.$$

□

Therefore, we may assume every self-dual convolutional code to be a $(2k,k)$ or equivalently a $(n,n/2)$ code.

In a next step, we will establish the main result of this chapter, which will be used for determining whether a convolutional code is self-dual, in later sections.

Theorem 3.3.9. *Let $\mathcal{C}$ be a $(2k,k)$ convolutional code with generator matrix $G(z)$, then the following statements are equivalent:*

(i) $\mathcal{C}$ is self-dual;

(ii) $\mathcal{C}$ is non-catastrophic and self-orthogonal;

(iii) $\mathcal{C}$ is non-catastrophic and $G(z)G(z)^T = 0$;

(iv) $G(z)$ is a parity-check matrix of $\mathcal{C}$.

Proof. $(i) \iff (ii)$ Let $\mathcal{C}$ be self-dual, then the code is self-orthogonal by definition and non-catastrophic by Lemma 3.3.7.

For the other direction, let $\mathcal{C}$ be non-catastrophic and self-orthogonal. From Lemma 3.3.2, we know that there exists a generator matrix $H(z)$ of $\mathcal{C}^{\perp}$ and from self-orthogonality

$$\mathcal{C} \subset \mathcal{C}^{\perp}$$

follows. Equivalently

$$rowspan(G(z)) \subset rowspan(H(z))$$

and hence every row $g_i(z)$ of $G(z)$ can be written as a linear combination of rows $h_i(z)$ of $H(z)$

$$g_i(z) = \sum_{j=1}^{k} u_{ij}(z)h_j(z)$$

for some $u_{ij}(z) \in R$ (up to k, because $H(z)$ has $dim(\mathcal{C}^{\perp}) = 2k - dim(\mathcal{C}) = k$ rows).

Let $U(z) = (u_{ij}(z))_{1 \le i,j \le k}$, then

$$U(z)H(z) = \begin{pmatrix} u_{11}(z) & \cdots & u_{1k}(z) \\ \vdots & & \vdots \\ u_{k1}(z) & \cdots & u_{kk}(z) \end{pmatrix} \begin{pmatrix} — & h_1(z) & — \\ & \vdots & \\ — & h_k(z) & — \end{pmatrix} =$$

$$= \begin{pmatrix} u_{11}(z)h_1(z) + \cdots + u_{1k}(z)h_k(z) \\ \vdots \\ u_{k1}(z)h_1(z) + \cdots + u_{kk}(z)h_k(z) \end{pmatrix} = \begin{pmatrix} \sum_{j=1}^{k} u_{1j}(z)h_j(z) \\ \vdots \\ \sum_{j=1}^{k} u_{kj}(z)h_j(z) \end{pmatrix} = \begin{pmatrix} — & g_1(z) & — \\ & \vdots & \\ — & g_k(z) & — \end{pmatrix} = G(z).$$

We assumed $\mathcal{C}$ to be non-catastrophic and equivalently $G(z)$ is left-prime. Furthermore, $G(z) = U(z)H(z)$ is a factorisation of $G(z)$. Combining the two, we find that the left factor $U(z)$ must be unimodular.

But this means that $G(z)$ and $H(z)$ are equivalent as generator matrices and we conclude that

$$\mathcal{C} = \mathcal{C}^{\perp}.$$

$(ii) \iff (iii)$ Let $\mathcal{C}$ be self-orthogonal. Take two rows $g_i(z), g_j(z)$ of $G(z)$, they both are codewords and therefore

$$g_i(z)g_j(z)^T = 0.$$

But as the two rows were chosen arbitrarily,

$$G(z)G(z)^T = 0$$

follows.

For the other direction, let $\mathcal{C}$ fulfil $G(z)G(z)^T = 0$. Take $c_1(z), c_2(z) \in \mathcal{C}$, then there exist $m_1(z), m_2(z) \in \mathbb{F}_q[z]^k$ such that

$$\begin{array}{ll} c_1(z) & = m_1(z)G(z) \\ c_2(z) & = m_2(z)G(z) \end{array}.$$

But now

$$c_1(z)c_2(z)^T = m_1(z)G(z)(m_2(z)G(z))^T = m_1(z)(G(z)G(z)^T)m_2(z)^T = 0$$

and therefore all codewords are orthogonal or equivalently $\mathcal{C}$ is self-orthogonal.

At this point, we have established that $(i) - (iii)$ are equivalent statements and need to connect them to the fourth statements.

$(i) \implies (iv)$ By assumption,

$$\mathcal{C} = \mathcal{C}^{\perp}$$

and by Lemma 3.3.4,

$$\mathcal{C}^{\perp} = ker(G(z)),$$

making $G(z)$ a parity check matrix of $\mathcal{C}$

$(iv) \implies (ii)$ Let $G(z)$ be a parity-check of $\mathcal{C}$, then $\mathcal{C}$ is non-catastrophic as it has a parity-check matrix and

$$G(z)G(z)^T = 0,$$

because $rowspan(G(z)) = \mathcal{C} = ker(G(z))$. □

One might observe that self-orthogonality and $G(z)G(z)^T = (0)$ are equivalent statements as the assumption of the code being non-catastrophic was not used in the proof of $(ii) \iff (iii)$.

For $(2k, k)$ linear block codes, self-duality and self-orthogonality were equivalent properties. For $(2k, k)$ convolutional codes however, this is not necessarily true as was shown in the previous Lemma and is further demonstrated in the following example.

Example 3.3.10. *Let*

$$G(z) = \begin{pmatrix} z^2+z+1 & z^2 & z & 1 \\ 1 & z & z^2 & z^2+z+1 \end{pmatrix} \in \mathbb{F}_2[z]^{2\times 4}$$

be the generator matrix of a binary $(4,2)$ *convolutional code* $\mathcal{C}$*.*

We want to show that $\mathcal{C} = rowspan(G(z))$ *is a proper subset of* $\mathcal{C}^\perp = ker(G(z))$*.*

Let $c(z) \in \mathcal{C}$*, then by definition there exists* $m(z) \in \mathbb{F}_2[z]^k$ *such that* $c(z) = m(z)G(z)$*. We further notice that*

$$G(z)G(z)^T = \begin{pmatrix} z^4+z^2+1+z^4+z^2+1 & z^2+z+1+z^3+z^3+z^2+z+1 \\ z^2+z+1+z^3+z^3+z^2+z+1 & z^4+z^2+1+z^4+z^2+1 \end{pmatrix} = 0$$

and hence

$$G(z)c(z)^T = G(z)(m(z)G(z))^T = G(z)G(z)^T m(z)^T = 0$$

or equivalently $c(z) \in \mathcal{C}^\perp$*.*

This establishes

$$\mathcal{C} \subset \mathcal{C}^\perp.$$

Finally, we observe that $(1,1,1,1) \in \mathcal{C}^\perp$ *or rather*

$$G(z)(1,1,1,1)^T = 0,$$

but $(1,1,1,1) \notin \mathcal{C}$*, which shows that* $\mathcal{C}$ *was indeed a proper subset of* $\mathcal{C}^\perp$*.*

Conclusively, $\mathcal{C}$ *is a self-orthogonal* $(2k,k)$ *convolutional codes that is not self-dual.*

As a consequence of Theorem 3.3.9, the previously presented code must be a catastrophic code.

Our classification and results will mostly be over the binary polynomial ring and determining, whether a vector is orthogonal to itself, will be an important part. This determination will be simplified by the following commonly known Lemma.

Lemma 3.3.11. *Let* $f_i(z) \in \mathbb{F}_2[z]$*, for* $i \in \{1,\dots,m\}$*, and* $f(z) = (f_1(z),\dots,f_m(z))$*, then*

$$f(z)f(z)^T = (f(z)(1,\dots,1)^T)^2,$$

where the equality is over $\mathbb{F}_2$*.*

Proof. First we remark that

$$f(z)f(z)^T = f_1(z)^2 + \cdots + f_m(z)^2$$

and

$$f(z)(1,\dots,1)^T = f_1(z) + \cdots + f_m(z).$$

But now

$$(f(z)(1,\dots,1)^T)^2 = (f_1(z) + \cdots + f_m(z))^2 =$$

$$= f_1(z)^2 + f_1(z)\sum_{i=2}^{m} f_i(z) + \cdots + f_m(z)^2 + f_m(z)\sum_{i=1}^{m-1} f_i(z) =$$

$$= f_1(z)^2 + \cdots + f_m(z)^2 + 2\sum_{\substack{i=1\\ j>i}}^{k} f_i(z)f_j(z) =$$

$$= f_1(z)^2 + \cdots + f_m(z)^2 = f(z)f(z)^T.$$

□

As a consequence, to determine whether a binary polynomial vector is orthogonal to itself, we may just check if the sum of its entries adds up to zero instead of having to check whether the sum of squares is zero. In mathematical terms, we mean that

$$f(z)f(z)^T = 0$$

and

$$f(z)(1,\ldots,1)^T = 0$$

are equivalent conditions.

Let us further discuss this property by presenting an example of a binary self-dual convolutional code.

Example 3.3.12. *We claim that*

$$G(z) = \begin{pmatrix} 1 & 0 & 0 & z^3+z^2+z & z^3+z & z^2+1 \\ 0 & 1 & 0 & z^3+z & z^3+z^2+1 & z^2+z \\ 0 & 0 & 1 & z^2+1 & z^2+z & z \end{pmatrix} \in \mathbb{F}_2[z]^{3\times 6}$$

generates a binary self-dual convolutional code.

It definitely is a $(2k,k)$ code and a non-catastrophic code as the Smith form of $G(z)$ is $[I_3\ 0]$. So to conclude with Theorem 3.3.9, $G(z)G(z)^T = 0$ is the only thing left to show.

Let $g_i(z)$ be the i-th row of $G(z)$. If we now want to know whether e.g.

$$g_1(z)g_1(z)^T = 0,$$

then by the previous lemma it is enough to check if the sum of the entries in the first row adds up to 0 and indeed

$$1 + 0 + 0 + (z^3+z^2+z) + (z^3+z) + (z^2+1) = 0.$$

By inspection, each rows entries add up to 0 and therefore $g_i(z)g_i(z)^T = 0$ for $1 \le i \le 3$.

Finally,

$$\begin{aligned} g_1(z)g_2(z)^T &= (z^3+z^2+z)(z^3+z) + (z^3+z)(z^3+z^2+1) + (z^2+1)(z^2+z) = \\ &= (z+1)(z^3+z) + (z^3+z)(z+1) = 0 \end{aligned}$$

and

$$\begin{aligned} g_1(z)g_3(z)^T &= (z^3+z^2+z)(z^2+1) + (z^3+z)(z^2+z) + (z^2+1)z = \\ &= (z^3+z^2)(z^2+1) + (z^2+1)(z^3+z^2) = 0 \end{aligned}$$

and

$$\begin{aligned} g_2(z)g_3(z)^T &= (z^3+z)(z^2+1)+(z^3+z^2+1)(z^2+z)+(z^2+z)z &= \\ &= (z^3+z)(z^2+1)+(z^3+z^2+z+1)(z^2+z) &= \\ &= z^5+z+z^5+z &= 0. \end{aligned}$$

We successfully showed

$$G(z)G(z)^T = \begin{pmatrix} g_1(z)g_1(z)^T & g_1(z)g_2(z)^T & g_1(z)g_3(z)^T \\ g_2(z)g_1(z)^T & g_2(z)g_2(z)^T & g_2(z)g_3(z)^T \\ g_3(z)g_1(z)^T & g_3(z)g_2(z)^T & g_3(z)g_3(z)^T \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

and conclude with Theorem 3.3.9 that the claim was true.

This is just one minor simplification that Lemma 3.3.11 helped us with. We will see more interesting and complex applications thereof, as e.g. the following lemmas.

Lemma 3.3.13. *All codewords of a binary self-dual code have even weight.*

Proof. Let $c(z) = \sum_{i=0}^{l} c_i z^i$ be a codeword of a binary self-dual (n,k) convolutional code, where $l = deg(c(z))$ and $c_i \in \mathbb{F}_2^n$. By the definition of self-duality, every codeword must be orthogonal to itself, or rather

$$c(z)c(z)^T = 0.$$

Furthermore, it was established to be a consequence of Lemma 3.3.11 that this is equivalent to demanding

$$c(z)(1,\ldots,1)^T = 0$$

or equivalently

$$\sum_{i=0}^{l} c_i z^i (1,\ldots,1)^T = \sum_{i=0}^{l} z^i c_i (1,\ldots,1)^T = 0.$$

Every coefficient of the last sum must be equal to zero or in other terms,

$$c_i(1,\ldots,1)^T = 0$$

must hold for every $i \in \{0,\ldots,l\}$ and hence $wt(c_i)$ is even. But then

$$wt(c(z)) = \sum_{i=0}^{l} wt(c_i)$$

is also even. □

In [11], the following result was proven for block codes. We did not find a work, where it was proven for convolutional codes, but we assume it was already known.

Lemma 3.3.14. *Every binary self-dual convolutional code contains* $(1,\ldots,1) \in \mathbb{F}_2^n$.

Proof. Let $g_i(z)$ be a row of the generator matrix $G(z)$, then

$$g_i(z)g_i(z)^T = 0$$

or equivalently (Lemma 3.3.11)

$$g_i(z)(1,\ldots,1)^T = 0$$

and therefore

$$G(z)(1,\ldots,1)^T = 0.$$

Hence $(1,\ldots,1)$ is part of the dual of the code and, thanks to self-duality, part of the code itself. □

Lemma 3.3.15. *Let $\mathcal{C}$ be a binary self-dual (n,k) convolutional code, then there exists a generator matrix with just ones in the first row*

Proof. The idea of this proof is due to Dr. Julia Lieb.

Let $G(z) = (g_{ij}(z))_{1\leq i\leq k, 1\leq j\leq n}$ be a generator matrix of $\mathcal{C}$, then we know by the previous lemma that there exists $b(z) = (b_1(z), \dots, b_k(z)) \in \mathbb{F}_2[z]^k$ such that

$$b(z)G(z) = (1, \dots, 1).$$

Therefore $b_1(z)g_{11}(z) + \cdots b_k(z)g_{1k}(z) = 1$, making the vector $b(z)$ interpreted as a $1\times n$ matrix fulfil the condition of its minors generating $\mathbb{F}_2[z]$ and equivalently $b(z)$ can be completed to an unimodular matrix (Theorem 3.2.6 (7.)). Meaning that there exists $L(z) \in \mathbb{F}_2[z]^{n-1\times n}$ such that

$$U(z) := \begin{pmatrix} b(z) \\ L(z) \end{pmatrix}$$

is unimodular. Finally, with

$$U(z)G(z) = \begin{pmatrix} b(z)G(z) \\ * \end{pmatrix} = \begin{pmatrix} 1 & \cdots & 1 \\ * & * & * \end{pmatrix},$$

we conclude that $G(z)$ is equivalent to a generator matrix with just ones in the first row. □

This generator matrix with just ones in the first row, $U(z)G(z)$, is further equivalent to its own column Hermite form, making them also generate the same code. Therefore, to find all binary self-dual convolutional codes, we may characterize them by their generator matrix that has a first row of just ones and which is in column Hermite form.

Remark 3.3.16. *Let*

$$G(z) = \begin{pmatrix} 1 & 1 & 1 & 1 & \cdots & \cdots & 1 \\ 0 & g_{22} & g_{23} & g_{24} & \cdots & \cdots & g_{2n} \\ \vdots & 0 & g_{33} & g_{34} & \cdots & \cdots & g_{3n} \\ \vdots & \vdots & \ddots & \ddots & \ddots & & \vdots \\ 0 & 0 & \cdots & 0 & g_{kk} & \cdots & g_{kn} \end{pmatrix}$$

be a generator matrix of a self-dual code in the just discussed form. Then we choose this form of $G(z)$ as the representative of any binary self-dual convolutional code for our classifications.

Multiplying the generator matrix of a self-dual code with an unimodular matrix from the left-hand side leaves the code unchanged and therefore leaves the code self-dual. But what about multiplication from the right-hand side?
We claim that exchanging columns is the only column operation, which surely has no influence on self-duality, whereas multiplying or adding columns might tamper with self-duality.

Lemma 3.3.17. *Let*

$$G(z) = \begin{pmatrix} | & & | \\ g_1 & \cdots & g_{2k} \\ | & & | \end{pmatrix}$$

generate a self-dual convolutional code, then for distinct $i_1, \dots, i_{2k} \in \{1, \dots, 2k\}$

$$\tilde{G}(z) = \begin{pmatrix} | & & | \\ g_{i_1} & \cdots & g_{i_{2k}} \\ | & & | \end{pmatrix}$$

generates a self-dual convolutional code.

Proof. This lemma is due to the fact that $\mathcal{C}^{\perp} = ker(G(z))$ and $\tilde{\mathcal{C}}^{\perp} = ker(\tilde{G}(z))$ differ by exchanging columns of their $1 \times n$ vector elements in the same way $\mathcal{C} = rowspan(G(z))$ and $\tilde{\mathcal{C}} = rowspan(\tilde{G}(z))$ do differ. Therefore

$$\tilde{\mathcal{C}} = \tilde{\mathcal{C}}^{\perp}.$$

□

Previously, it was claimed that adding and multiplying columns might tamper with self-duality. This statement will be further examined in the following example.

Example 3.3.18. *We claim that*

$$G(z) = \begin{pmatrix} 3 & z & 1 & 3z \\ 1 & 2z+4 & 2 & z+2 \end{pmatrix} \in \mathbb{F}_5[z]^{2\times 4}$$

generates a self-dual code.

We will use Theorem 3.3.9 to proof this. Firstly,

$$G(z)G(z)^T = \begin{pmatrix} 3^2 + z^2 + 1^2 + (3z)^2 & 3 + 2z^2 + 4z + 2 + 3z^2 + 6z \\ 3 + 2z^2 + 4z + 2 + 3z^2 + 6z & 1^2 + (2z)^2 + 16z + 4^2 + 2^2 + z^2 + 4z + 2^2 \end{pmatrix} =$$

$$= \begin{pmatrix} 10z^2 + 10 & 5z^2 + 10z + 5 \\ 5z^2 + 10z + 5 & 5z^2 + 20z + 25 \end{pmatrix} = 0$$

and secondly

$$det\left[\begin{pmatrix} 3 & z \\ 1 & 2z+4 \end{pmatrix}\right] = 6z + 12 - z = 2,$$

which implies that the set of $G(z)$'s minors generates $\mathbb{F}_5[z]$ or equivalently $G(z)$ generates a non-catastrophic code.

By Theorem 3.3.9, these two facts let us conclude that the claim was indeed true.

We now want to establish that adding and multiplying columns might lead to problems. For this, add the second column of $G(z)$ to the first to find

$$\begin{pmatrix} 3 & z & 1 & 3z \\ 1 & 2z+4 & 2 & z+2 \end{pmatrix} \sim \begin{pmatrix} 3+z & z & 1 & 3z \\ 2z & 2z+4 & 2 & z+2 \end{pmatrix} =: G_1(z).$$

Then the first row of $G_1(z)$ is not orthogonal to itself as

$$(3+z, z, 1, 3z)(3+z, z, 1, 3z)^T = 9 + 6z + z^2 + z^2 + 1 + 9z^2 = z^2 + z \neq 0,$$

meaning that it does not generate a self-dual code anymore.

Similarly, if we multiply the first column of $G(z)$ by 2 we find

$$\begin{pmatrix} 3 & z & 1 & 3z \\ 1 & 2z+4 & 2 & z+2 \end{pmatrix} \sim \begin{pmatrix} 1 & z & 1 & 3z \\ 2 & 2z+4 & 2 & z+2 \end{pmatrix}$$

and again the first row is not orthogonal to itself as

$$(1, z, 1, 3z)(1, z, 1, 3z)^T = 1 + z^2 + 1 + 9z^2 = 2 \neq 0.$$

This ends our discussion of the theory about self-dual convolutional codes and we will now dive into the classification of different types of self-dual convolutional codes, primarily using the equivalence between

$$\mathcal{C} \textit{ being self-dual}$$

and

$$\mathcal{C} \textit{ being non-catastrophic and its generator matrix } G(z) \textit{ fullfilling } G(z)G(z)^T = 0$$

as presented in Theorem 3.3.9.

4 Classification of Self-Dual Convolutional Codes

In this section, we will be classifying all self-dual $(2,1)$ convolutional codes over finite fields and all binary self-dual $(4,2)$ convolutional codes. We will further classify all self-dual convolutional codes with double diagonal matrices over finite fields and all binary self-dual convolutional codes with double upper triangular generator matrices.

4.1 Self-Dual (2,1) Convolutional Codes

We will try to find all self-dual $(2,1)$ convolutional codes in $R := \mathbb{F}_q[z]$, where $q = p^l$ is a prime power. But first, we state two commonly known facts.

Lemma 4.1.1. *Let $\mathcal{C}$ be a self-dual $(2k,k)$ convolutional code. If a row of $G(z)$ has a non-constant polynomial entry, then there must be at least two entries in this row that are equal to the degree of the row.*

Proof. Let $G = (g_{ij}(z))_{1\leq i\leq k, 1\leq j\leq 2k}$ and assume this statement to not be true for the row $g_i(z)$. Then there exists exactly one $j \in \{1,\dots,n\}$ such that $rowdeg(g_i(z)) = deg(g_{ij}(z))$. But now

$$deg(g_i(z)g_i(z)^T) = deg(g_{i1}(z)^2 + \cdots + g_{in}(z)^2) = deg(g_{ij}(z)^2) > 0$$

and therefore $g_i(z)g_i(z)^T \neq 0$, which is a contradiction to $\mathcal{C}$ being self-dual. □

Theorem 4.1.2. *[13] Let $q = p^l$ be a prime power, then there exist $a, b \in R \setminus \{0\}$ such that*

$$a^2 + b^2 = 0 \ mod \ p$$

if and only if either

$$p = 1 \ mod \ 4$$

or

$$p = 0 \ mod \ 2$$

or

$$l \text{ is even}.$$

Proof. See Appendix □

In other words, if $q = p^l$ fulfils $p = 3 \ mod \ 4$ and l is odd, then there exists no solution to $a^2 + b^2 = 0 \ mod \ p$. In all other cases, a solution exists.

Denote by $G(z) = (g_1(z), g_2(z)) \in R^{1\times 2}$ the generator matrix of a self-dual $(2,1)$ convolutional code $\mathcal{C}$. By Theorem 3.3.9,

$$G(z)G(z)^T = 0$$

is a necessary condition and hence

$$g_1(z)^2 + g_2(z)^2 = 0.$$

Firstly, $g_1(z)$ and $g_2(z)$ must have the same degree by Lemma 4.1.1. Secondly, the highest degree coefficients of $g_1(z)^2$ and $g_2(z)^2$ have to add up to a multiple of p, which leads to the problem of $a^2 + b^2 = 0 \ mod \ p$ having a solution. This problem has no solutions whenever $p = 3 \ mod \ 4$ and l is odd. Hence there are no self-dual $(2,1)$ convolutional codes for these q.

Let $p = 1 \ mod \ 4$, $p = 2$ or l even, then there exists a solution to

$$a^2 + b^2 = 0 \ mod \ p$$

for some $a, b \in R \setminus \{0\}$. Fix (a, b) as a solution and let

$$\begin{array}{rcl} \tilde{a} & := & a \ mod \ p \\ \tilde{b} & := & b \ mod \ p \end{array}.$$

Lemma 4.1.3. *$C = rowspan((\tilde{a} \ \tilde{b}))$ is a self-dual convolutional code over R.*

Proof. The set of k-th minors M is equal to $\left\{\tilde{a}, \tilde{b}\right\} \subset R^*$ and hence

$$< M >=< \tilde{a}, \tilde{b} >= R$$

as an ideal over R. By Theorem 3.2.6, this implies that $\mathcal{C}$ is non-catastrophic.

Finally

$$G(z)G(z)^T = \tilde{a}^2 + \tilde{b}^2 = a^2 + b^2 = 0 \ mod \ p,$$

lets us conclude with Theorem 3.3.9. □

We further claim that these are the only self-dual $(2, 1)$ convolutional codes.

Lemma 4.1.4. *There are no other self-dual $(2, 1)$ convolutional codes over R.*

Proof. There are no other generator matrices with constant entries, because we need

$$G(z)G(z)^T = 0,$$

and for the reasoning why $G(z)$ may not contain non-constant entries, we present the following argument:

Let $G(z) = (g_1(z) \ g_2(z))$ generate a self-dual code. Then $G(z)$ should also be generating a non-catastrophic code and equivalently by Theorem 3.2.6. its set of first degree minors in $\{g_1(z), g_2(z)\}$ generates R. In other terms, this means that there exists $a(z), b(z) \in R$ such that

$$a(z)g_1(z) + b(z)g_2(z) = 1.$$

Secondly, $G(z)G(z)^T = 0$ or rather

$$g_1(z)^2 + g_2(z)^2 = 0$$

must hold. Multiplying the second equation with $a(z)^2$ and applying the first equation we find

$$\begin{array}{rcccl} a(z)^2(g_1(z)^2 + g_2(z)^2) & = & (a(z)g_1(z))^2 + a(z)^2g_2(z)^2 & = & \\ (1 - b(z)g_2(z))^2 + a(z)^2g_2(z)^2 & = & 1 - 2b(z)g_2(z) + b(z)^2g_2(z)^2 + a(z)^2g_2(z)^2 & = & 0. \end{array}$$

But the last equation can be rearranged to

$$-1 = g_2(z)(-2b(z) + b(z)^2g_2(z) + a(z)^2g_2(z)),$$

implying that $g_2(z)$ and therefore also $g_1(z)$ (by Lemma 4.1.1) are actually constants. □

Therefore we fully classified all self-dual $(2, 1)$ convolutional codes over finite fields and they are exactly the self-dual $(2, 1)$ block codes.

4.2 Binary Self-Dual (4,2) Convolutional Codes

The goal in this section is to find all binary self-dual $(4, 2)$ convolutional codes $\mathcal{C}$ classified by their representative as defined in Remark 3.3.16. To get there we use the equivalence between

$$\textit{$\mathcal{C}$ being self-dual}$$

and

$$\textit{$\mathcal{C}$ being non-catastrophic and } G(z)G(z)^T = 0.$$

Let $\mathcal{C}$ be a binary self-dual $(4, 2)$ convolutional code. By Remark 3.3.16, we may assume that

$$G(z) = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & g_{22}(z) & g_{23}(z) & g_{24}(z) \end{pmatrix} \in \mathbb{F}_2[z]^{2\times 4}$$

is a generator matrix of $\mathcal{C}$ for some $g_{22}(z), g_{23}(z), g_{24}(z) \in \mathbb{F}_2[z]$.

Next, we want to find restrictions, which follow from the mentioned equivalence.
$G(z)G(z)^T = 0$ tells us that

$$g_{22}(z)^2 + g_{23}(z)^2 + g_{24}(z)^2 = 0$$

or equivalently (by Lemma 3.3.11)

$$g_{22}(z) + g_{23}(z) + g_{24}(z) = 0. \tag{2}$$

By Theorem 3.2.6, being non-catastrophic is equivalent to the set of 2-nd minors M of $G(z)$

$$M = \{g_{22}(z), g_{23}(z), g_{24}(z), g_{22}(z) + g_{23}(z), g_{22}(z) + g_{24}(z), g_{23}(z) + g_{24}(z)\}$$

generating $\mathbb{F}_2[z]$ over itself in

$$< M >=< g_{22}(z), g_{23}(z), g_{24}(z) > \stackrel{!}{=} \mathbb{F}_2[z].$$

The last equation holds if and only if $1 \in < M >$ or rather

$$gcd(g_{22}(z), g_{23}(z), g_{24}(z)) = 1. \tag{3}$$

But we also know from (2) that

$$g_{22}(z) = g_{23}(z) + g_{24}(z)$$

simplifying (3) into

$$gcd(g_{23}(z), g_{24}(z)) = 1.$$

We conclude that for $g_{23}(z), g_{24}(z) \in \mathbb{F}_2[z]$ such that $gcd(g_{23}(z), g_{24}(z)) = 1$,

$$G(z) = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & g_{23}(z) + g_{24}(z) & g_{23}(z) & g_{24}(z) \end{pmatrix}$$

generates a binary self-dual $(4, 2)$ convolutional code and that every binary self-dual $(4, 2)$ convolutional code has a generator matrix of this form.

4.3 Double Diagonal Generator Matrix

In this subsection, we want to find all self-dual convolutional codes with generator matrices of the form

$$G(z) = \begin{pmatrix} g_{1,1}(z) & & & & g_{1,k+1}(z) & & & \\ & g_{2,2}(z) & & & & g_{2,k+1}(z) & & \\ & & \ddots & & & & \ddots & \\ & & & g_{k,k}(z) & & & & g_{k,2k}(z) \end{pmatrix} \in \mathbb{F}_q^{k\times 2k},$$

where $q = p^l$ for some $l \in \mathbb{N}$.

Let $i \in \{1, ..., k\}$. By Theorem 3.3.9,

$$G(z)G(z)^T = 0$$

must hold and hence

$$g_{i,i}(z)^2 + g_{i,k+i}(z)^2 = 0. \tag{4}$$

Furthermore, as $G(z)$ generates a non-catastrophic code, its row Hermite form must be $[I_k\ 0]$, implying that there exists $a(z), b(z) \in R$ such that

$$a(z)g_{i,i}(z) + b(z)g_{i,k+i}(z) = 1. \tag{5}$$

The very attentive reader might have seen this exact situation before. It was in Lemma 4.1.4, where we used (4) and (5) to conclude that $g_{i,i}(z)$ and $g_{i,k+i}(z)$ must be constants. We do the same here.

Conclusively

$$G(z) = \begin{pmatrix} a_1 & & & & b_1 & & & \\ & a_2 & & & & b_2 & & \\ & & \ddots & & & & \ddots & \\ & & & a_k & & & & b_k \end{pmatrix} \in \mathbb{F}_q^{k\times 2k},$$

whereby $a_i, b_i \in \mathbb{F}_{p^l}^*$ such that $a_i^2 + b_i^2 = 0\ mod\ p$, are the only double diagonal generator matrices of self-dual convolutional codes. It was established in Theorem 4.1.2 that such a_i and b_i do not exist whenever $p = 3\ mod\ 4$ and l is odd.

4.4 Double Upper Triangular Generator Matrices

Let $\mathcal{C}$ be a binary self-dual $(2k, k)$ convolutional code with a generator matrix $G(z)$ of the form

$$G(z) = \begin{pmatrix} g_{1,1} & \cdots & \cdots & g_{1,k} & g_{1,k+1} & g_{1,k+2} & \cdots & g_{1,2k} \\ & g_{2,2} & \cdots & g_{2,k} & & g_{2,k+2} & \cdots & g_{2,2k} \\ & & \ddots & \vdots & & & \ddots & \vdots \\ & & & g_{k,k} & & & & g_{k,2k} \end{pmatrix} \in \mathbb{F}_2[z]^{k\times 2k}.$$

We want to show that

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

is also a generator matrix of $\mathcal{C}$.

For simplicity we did and will write $g_{i,j}$ instead of $g_{i,j}(z)$.

Let $g_i(z)$ be the i-th row of $G(z)$, then $g_k(z)g_i(z)^T = 0$ for $i \in \{1, \ldots, k\}$ leads to

$$\begin{array}{rcl} g_{k,k} & = & g_{k,2k} \\ g_{k,k}(g_{i,k} + g_{i,2k}) & = & 0 \end{array}$$

for $1 \leq i \leq k$.
$g_{k,k}$ is non-zero (as $rank(G(z)) = k$) and hence $g_{i,k} = g_{i,2k}$. In terms of $G(z)$, this means that

$$G(z) = \begin{pmatrix} g_{1,1} & g_{1,2} & \cdots & \cdots & g_{1,k} & g_{1,k+1} & g_{1,k+2} & \cdots & \cdots & g_{1,k} \\ & g_{2,2} & g_{2,3} & \cdots & g_{2,k} & & g_{2,k+2} & g_{2,k+3} & \cdots & g_{2,k} \\ & & \ddots & \ddots & \vdots & & & \ddots & \vdots & \\ & & & g_{k-1,k-1} & g_{k-1,k} & & & & g_{k-1,2k-1} & g_{k-1,k} \\ & & & & g_{k,k} & & & & & g_{k,k} \end{pmatrix}.$$

We further notice that $g_{k,k}$ is a common factor of every k-th minor and hence must be equal to 1 to guarantue the code to be non-catastrophic. Therefore

$$G(z) = \begin{pmatrix} g_{1,1} & g_{1,2} & \cdots & \cdots & g_{1,k} & g_{1,k+1} & g_{1,k+2} & \cdots & \cdots & g_{1,k} \\ & g_{2,2} & g_{2,3} & \cdots & g_{2,k} & & g_{2,k+2} & g_{2,k+3} & \cdots & g_{2,k} \\ & & \ddots & \ddots & \vdots & & & \ddots & \vdots & \\ & & & g_{k-1,k-1} & g_{k-1,k} & & & & g_{k-1,2k-1} & g_{k-1,k} \\ & & & & 1 & & & & & 1 \end{pmatrix}.$$

Furthermore, $G(z)$ is equivalent to

$$\tilde{G}(z) = \begin{pmatrix} g_{1,1} & g_{1,2} & \cdots & g_{1,k-1} & 0 & g_{1,k+1} & g_{1,k+2} & \cdots & g_{1,2k-1} & 0 \\ & g_{2,2} & \cdots & g_{2,k-1} & 0 & & g_{2,k+2} & \cdots & g_{2,2k-1} & 0 \\ & & \ddots & & \vdots & & & \ddots & \vdots & \\ & & & g_{k-1,k-1} & 0 & & & & g_{k-1,2k-1} & 0 \\ & & & & 1 & & & & & 1 \end{pmatrix}$$

by doing the unimodular operation of adding the k-th row times $g_{i,k}$ to every i-th row, where $i \in \{1, \ldots, k-1\}$. This still generates the same code.

Let $\tilde{g}_i(z)$ be the i-th row of $\tilde{G}(z)$. Now if we calculate $\tilde{g}_i(z)\tilde{g}_{k-1}(z)^T$ for $i \in \{1, ..., k-1\}$, we find that similarly as before

$$\begin{array}{rcl} g_{k-1,k-1} & = & g_{k-1,2k-1} \\ g_{k-1,k-1}(g_{i,k-1} + g_{i,2k-1}) & = & 0 \end{array}.$$

Therefore we may use the exact same argument to find that $\tilde{G}(z)$ is equivalent to

$$\begin{pmatrix} g_{1,1} & \cdots & g_{1,k-2} & 0 & 0 & g_{1,k+1} & \cdots & g_{1,2k-2} & 0 & 0 \\ & \ddots & \vdots & \vdots & \vdots & & \ddots & \vdots & \vdots & \vdots \\ & & g_{k-2,k-2} & 0 & 0 & & & g_{k-2,2k-2} & 0 & 0 \\ & & & 1 & 0 & & & & 1 & 0 \\ & & & & 1 & & & & & 1 \end{pmatrix}.$$

After repeating this argument until all (i,i)-th and $(i,k+i)$-th entries are 1, we find that every double upper triangle generator matrix of a binary self-dual code is equivalent to

$$G'(z) = \begin{pmatrix} 1 & & & & 1 & & \\ & 1 & & & & 1 & \\ & & \ddots & & & & \ddots \\ & & & 1 & & & & 1 \end{pmatrix}.$$

Conclusively, there exists exactly one binary self-dual $(2k,k)$ convolutional code that has a double triangular generator matrix, being $\mathcal{C} = rowspan(G'(z))$.

5 Construction of Self-Dual Convolutional Codes

In this section, multiple results on the construction of self-dual convolutional codes will be presented. Further, we will try to generalize the two construction methods for block codes, we presented in the second chapter, to convolutional codes. Finally, we will have a closer look at some examples for binary self-dual convolutional codes of different lengths, dimensions and degrees.

5.1 Construction Methods

From Scratch

In 2012, H. Palines and V. Sison presented an interesting result on systematic self-dual convolutional codes, systematic meaning that there exists a generator matrix of the form $G(z) = [I_k \ A(z)]$ for some polynomial matrix $A(z)$. They have proven their result for any commutative ring, satisfying the "Descending Chain Condition".

Lemma 5.1.1. *[5] Let $A(z)$ be a polynomial matrix over $\mathbb{F}_q[z]$. If $A(z)$ is invertible and $A(z)^{-1} = -A(z)^T$, then $G(z) = [I_k \ A(z)]$ is a $k \times 2k$ generator matrix of a convolutional code $\mathcal{C}$ and $H(z) = [-A(z)^T \ I_k]$ is the parity-check matrix of $\mathcal{C}$ and consequently $\mathcal{C}$ is self-dual.*

Proof. We observe that $G(z)$ has Smith form $[I_k \ 0]$ and therefore generates a non-catastrophic code. Further,

$$\begin{array}{rclcrcl} G(z)G(z)^T & = & [I_k \ A(z)]G(z)^T & = & A(z)[A(z)^{-1} \ I_k]G(z)^T & = & \\ & = & A(z)[-A(z)^T \ I_k]G(z)^T & = & A(z)H(z)G(z)^T & = & 0, \end{array}$$

proving the lemma with Theorem 3.3.9. □

As a consequence of their more general Lemma 5.1.1, Palines and Sison suggested the following algorithm to construct self-dual convolutional codes over $\mathbb{F}_q(z)$ or also $\mathbb{Z}_{p^r}(z)$:

1. Construct set P of polynomials of degree less than or equal to L;
2. Construct set Q of all possible rational functions from the elements in P;
3. Construct $k \times k$ matrices $A_i(z)$ with entries coming from Q.
4. For each i, test whether $A_i(z)$ is invertible and if $A_i(z)^{-1} = -A_i(z)^T$ is satisfied.
5. If $A_i(z)^{-1} = -A_i(z)^T$, then $G(z) = [I_k \ A_i(z)]$ generates a self-dual convolutional code by Lemma 5.1.1.

One way to find self-dual convolutional codes with entries in our usual polynomial ring $\mathbb{F}_q[z]$ and not in its field of rational functions, we skip step 2 and in step 3 we look for matrices with entries in P instead. Unfortunately, this decreases our chances of finding an invertible matrix substantially as we now need $det(A(z)) \in \mathbb{F}_q[z]^* = \mathbb{F}_q \setminus \{0\}$ instead of just $det(A(z)) \in \mathbb{F}_q(z)^* = \mathbb{F}_q(z) \setminus \{0\}$.

The other way that was actually used by H. Palines and V. Sison, was to find a $G(z) \in \mathbb{F}_q(z)^{k \times n}$ and then calculate a basic generator matrix, which always exists and has entries in the polynomial ring.

Example 5.1.2. *If k is even and $f(z) \in \mathbb{F}_2[z]$, then*

$$G(z) = [I_k \; A(z)] \in \mathbb{F}_2^{k \times 2k}[z]$$

with

$$A(z) := \begin{pmatrix} f(z) & f(z)+1 & 0 & 0 & \cdots & \cdots & 0 & 0 \\ f(z)+1 & f(z) & 0 & 0 & \cdots & \cdots & 0 & 0 \\ 0 & 0 & f(z) & f(z)+1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & & 0 & f(z)+1 & f(z) & 0 & 0 \\ 0 & 0 & \cdots & \cdots & 0 & 0 & f(z) & f(z)+1 \\ 0 & 0 & \cdots & \cdots & 0 & 0 & f(z)+1 & f(z) \end{pmatrix} \in \mathbb{F}_2[z]^{k \times k}$$

generates a self-dual convolutional code.

Notice that $A(z)A(z) = I_k$, plus $A(z) = A(z)^T$ and conclude with

$$A(z)^{-1} = A(z) = A(z)^T = -A(z)^T$$

that Lemma 5.1.1 applies.

We further claim that $G(z)$ generates a code of degree $\delta = \frac{k}{2} deg(f(z))$.

Firstly, there exists a minor with the claimed degree in

$$det \left[\begin{pmatrix} f(z) & 0 & 0 & 0 & \cdots & \cdots & 0 & 0 \\ f(z)+1 & 1 & 0 & 0 & \cdots & \cdots & 0 & 0 \\ 0 & 0 & f(z) & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & \cdots & \cdots & 0 & 0 & f(z) & 0 \\ 0 & 0 & \cdots & \cdots & 0 & 0 & f(z)+1 & 1 \end{pmatrix} \right] = f(z)^{\frac{k}{2}}$$

and hence $\delta \geq \frac{k}{2} deg(f(z))$. Secondly, by adding rows we find (add second to first, fourth to third, etc.)

$$A(z) \sim \begin{pmatrix} 1 & 1 & 0 & 0 & \cdots & \cdots & 0 & 0 \\ f(z)+1 & f(z) & 0 & 0 & \cdots & \cdots & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & & 0 & f(z)+1 & f(z) & 0 & 0 \\ 0 & 0 & \cdots & \cdots & 0 & 0 & 1 & 1 \\ 0 & 0 & \cdots & \cdots & 0 & 0 & f(z)+1 & f(z) \end{pmatrix} = B(z),$$

where $B(z)$'s row degrees add up to $\frac{k}{2} deg(f(z))$. If we did the same row operations on $[I_k \; A(z)]$, this observation about the sum of row degrees would still be true. Finally adding rows does not change the degree of a code and therefore $\delta \leq \frac{k}{2} deg(f(z))$ by Remark 3.1.11.

For the interested reader, in [5] one example over $\mathbb{F}_2(z)$ and one over $\mathbb{Z}_4(z)$ was presented. Contrary to our example they were not obtained by thought, but by a computational search.

From Other Self-Dual Codes

In this subsection, we will have a closer look at some constructions, where already known self-dual convolutional codes are used to find new ones. The first two results are true over any polynomial ring of a general finite field and the rest only over the polynomial ring of the binary field.

Proposition 5.1.3. *If $G(z) \in R^{k\times n}$ and $\tilde{G}(z) \in R^{k'\times n'}$ are generator matrices of two self-dual convolutional codes, then the code $\mathcal{C}'$ generated by*

$$G'(z) = \begin{pmatrix} G(z) & 0 \\ 0 & \tilde{G}(z) \end{pmatrix} \in R^{(k+k')\times(n+n')}$$

is also self-dual.

Proof. We first compute the Smith normal form of $G'(z)$:

$$G'(z) = \begin{pmatrix} G(z) & 0 \\ 0 & \tilde{G}(z) \end{pmatrix} \sim \begin{pmatrix} [I_k\ 0] & 0 \\ 0 & [I_{k'}\ 0] \end{pmatrix} \sim \begin{pmatrix} I_{k+k'} & 0 \end{pmatrix},$$

where the first equivalence follows from $G(z)$ and $\tilde{G}(z)$ generating non-catastrophic codes. Therefore the Smith normal form of $G'(z)$ is $[I_{k+k'}\ 0]$ and equivalently $\mathcal{C}'$ is non-catastrophic.

Finally

$$G'(z)G'(z)^T = \begin{pmatrix} G(z) & 0 \\ 0 & \tilde{G}(z) \end{pmatrix} \begin{pmatrix} G(z) & 0 \\ 0 & \tilde{G}(z) \end{pmatrix}^T = \begin{pmatrix} G(z)G(z)^T & 0 \\ 0 & \tilde{G}(z)\tilde{G}(z)^T \end{pmatrix} = 0$$

lets us conclude with Theorem 3.3.9. □

The following proposition is a generalisation of a result presented in [6] by P. Gaborit and A. Otmani.

Proposition 5.1.4. *Let $\mathcal{C}$ be a self-dual convolutional code over R of length n with generator matrix $G(z)$. Now let $M_i(z) \in R^{n\times n}$ such that*

$$M_i(z)M_i(z)^T = \lambda_i I_n$$

for λ_i invertible in R.
Let $A_1, \ldots, A_r$ be permutations of the symmetric group S_n ($1 \leq r$). Then

$$G_r(z) = G(z)M_1(z)A_1 \cdots M_r(z)A_r$$

generates a self-dual code.

Proof. P. Gaborit and A. Otmani have proven this proposition for linear block codes in [6, Prop. 3.1]. Our proof is inspired by theirs.

We know that $G(z)$ has Smith form $[I_k\ 0]$ and $M_i(z)$ is unimodular as

$$M_i(z)(\lambda_i^{-1}M_i^T) = \lambda_i^{-1}(M_i(z)M_i^T) = \lambda_i^{-1}\lambda_i I_n = I_n.$$

Therefore $G(z)M_1(z)$ has Smith form $[I_k\ 0]$.

By definition, the A_i's are also unimodular, implying that

$$G_r(z) = G(z)M_1(z)A_1 \cdots M_r(z)A_r$$

has Smith form $[I_k\ 0]$ aswell, which lets us conclude with Theorem 3.2.6 that $G_r(z)$ generates a non-catastrophic code.

Finally thanks to $A_iA_i^T = I_n$, we find

$$\begin{aligned} G_r(z)G_r(z)^T &= G(z)M_1(z)A_1\cdots M_r(z)(A_rA_r^T)M_r(z)^T\cdots A_1^TM_1(z)^TG(z)^T &&= \\ &= G(z)M_1(z)A_1\cdots M_r(z)I_nM_r(z)^T\cdots A_1M_1(z)^TG(z)^T &&= \\ &= G(z)M_1(z)A_1\cdots A_{r-1}(M_r(z)M_r(z)^T)A_{r-1}\cdots A_1M_1(z)^TG(z)^T &&= \\ &= G(z)M_1(z)A_1\cdots M_{r-1}(z)A_{r-1}\lambda_rA_{r-1}M_{r-1}(z)^T\cdots A_1M_1(z)^TG(z)^T &&= \\ &= \lambda_1\lambda_2\cdots\lambda_rG(z)G(z)^T &&= 0, \end{aligned}$$

which lets us conclude with Theorem 3.3.9 that $G_r(z)$ indeed generates a self-dual convolutional code. □

In section 2.2, two construction methods for binary self-dual block codes were introduced in the Building-up and Harada-Munemasa construction. We will discuss in the following pages, how they are best implemented for convolutional codes and which interesting properties their generalisations possess.

Proposition 5.1.5 (Generalized Building-up construction). *Let $\mathcal{C}$ be a binary self-dual $(2k, k)$ convolutional code and $G(z)$ its generator matrix with $g_i(z)$ being the i-th row of $G(z)$. Let $f(z) \in \mathbb{F}_2[z]^{2k}$ be a $1 \times 2k$ vector with $f(z)(1,\ldots,1)^T = 1$ and define $y_i(z) := f(z)g_i(z)^T$ for $1 \le i \le k$. Then*

$$\tilde{G}(z) = \begin{pmatrix} 1 & 0 & f(z) \\ y_1(z) & y_1(z) & \\ \vdots & \vdots & G(z) \\ y_k(z) & y_k(z) & \end{pmatrix} \in \mathbb{F}_2[z]^{(k+1)\times(2k+2)}$$

generates a binary self-dual $(2k+2, k+1)$ convolutional code $\tilde{\mathcal{C}}$.

Proof. The goal is to show that $\tilde{G}(z)$ generates a non-catastrophic code and $\tilde{G}(z)\tilde{G}(z)^T = 0$ to then conclude with Theorem 3.3.9.

First, we compute $\tilde{G}(z)$'s Smith form.

Take the second column and add it to the first

$$\tilde{G}(z) = \begin{pmatrix} 1 & 0 & f(z) \\ y_1(z) & y_1(z) & \\ \vdots & \vdots & G(z) \\ y_k(z) & y_k(z) & \end{pmatrix} \sim \begin{pmatrix} 1 & 0 & f(z) \\ 0 & y_1(z) & \\ \vdots & \vdots & G(z) \\ 0 & y_k(z) & \end{pmatrix}.$$

Then use the first column to the delete the first row

$$\begin{pmatrix} 1 & 0 & f(z) \\ 0 & y_1(z) & \\ \vdots & \vdots & G(z) \\ 0 & y_k(z) & \end{pmatrix} \sim \begin{pmatrix} 1 & 0 & 0 \\ 0 & y_1(z) & \\ \vdots & \vdots & G(z) \\ 0 & y_k(z) & \end{pmatrix}.$$

Finally use the fact that $G(z)$ has row Hermite form $[I_k\ 0]$ to find

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & y_1(z) & \\ \vdots & \vdots & G(z) \\ 0 & y_k(z) & \end{pmatrix} \sim \begin{pmatrix} 1 & 0 & 0 \\ 0 & y_1(z) & \\ \vdots & \vdots & [I_k\ 0] \\ 0 & y_k(z) & \end{pmatrix} \sim \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & \\ \vdots & \vdots & [I_k\ 0] \\ 0 & 0 & \end{pmatrix} \sim [I_{k+1}\ 0],$$

where we now conclude with Theorem 3.2.6 that $\tilde{G}(z)$ generates a non-catastrophic code.

Denote by $\tilde{g}_i(z)$ the i-th row of $\tilde{G}(z)$ and by $g_i(z)$ the i-th row of $G(z)$. As a consequence of Lemma 3.3.11,

$$f(z)f(z)^T = 1$$

and

$$f(z)(1,\dots,1)^T = 1$$

are equivalent conditions. Therefore the first row is orthogonal to itself as

$$\tilde{g}_1(z)\tilde{g}_1(z)^T = 1^2 + f(z)f(z)^T = 1 + 1 = 0 \tag{6}$$

and to every other row as

$$\tilde{g}_1(z)\tilde{g}_{i+1}(z) = y_i(z) + f(z)g_i(z)^T = f(z)g_i(z)^T + f(z)g_i(z)^T = 0,$$

where $1 \leq i \leq k$.

Finally let $1 \leq i, j \leq k$, then

$$\tilde{g}_{i+1}(z)\tilde{g}_{j+1}(z)^T = y_i(z)y_j(z) + y_i(z)y_j(z) + g_i(z)g_j(z)^T = g_i(z)g_j(z)^T = 0$$

as $G(z)G(z)^T = 0$.

Conclusively, we just showed that all rows of $\tilde{G}(z)$ are orthogonal to each other or rather

$$\tilde{G}(z)\tilde{G}(z)^T = 0$$

and conclude with Theorem 3.3.9 that $\tilde{G}(z)$ generates a self-dual code. □

In equation (6) we see that the first row of $\tilde{G}(z)$ is orthogonal to itself if and only if $f(z)f(z)^T = 1$, meaning that the condition we set on $f(z)$ is not only a sufficient, but also a necessary condition.

Example 5.1.6. *We know thanks to section 4.2, that*

$$G_1(z) = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & z+1 & z \end{pmatrix}$$

generates a binary self-dual convolutional code. Let's apply the generalized Building-up construction with $f(z) = (1, z, z^2, z^2 + z) \in \mathbb{F}_2[z]^4$.

The condition of $f(z)(1,1,1,1)^T = 1$ *is fulfilled and*

$$\begin{array}{rcrcl} y_1(z) & = & f(z)(1,1,1,1)^T & = & 1 \\ y_2(z) & = & f(z)(0,1,z+1,z)^T & = & z + z^3 + z^2 + z^3 + z^2 = z. \end{array}$$

Therefore

$$G_2(z) = \begin{pmatrix} 1 & 0 & — & f(z) & — & — \\ y_1(z) & y_1(z) & 1 & 1 & 1 & 1 \\ y_2(z) & y_2(z) & 0 & 1 & z+1 & z \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & z & z^2 & z^2+z \\ 1 & 1 & 1 & 1 & 1 & 1 \\ z & z & 0 & 1 & z+1 & z \end{pmatrix}$$

generates a binary self-dual convolutional code.

We will dive further into this and other examples in section 5.2.

Next we show that the generalisation of Theorem 2.2.2 does not hold for convolutional codes.

Lemma 5.1.7. *Not every binary self-dual convolutional code with free distance $d_{free} > 2$ can be constructed with the generalized Building-up construction (up to permutation of columns).*

Proof. By section 4.2,

$$G(z) = \begin{pmatrix} 0 & z^2+z+1 & z & z^2+1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \in \mathbb{F}_2[z]^{2\times 4}$$

generates a binary self-dual convolutional code.

We claim that $G(z)$ cannot be constructed with the generalized Building-up construction, because there is no codeword in $\mathcal{C} = rowspan(G(z))$, having a 1 and a 0 entry simultaneously. To proof this we go through every possible entry and show that having a 0 entry and a 1 is mutually exclusive.

Assume that

$$c(z) = a(z)(1,1,1,1) + b(z)(0, z^2+z+1, z, z^2+1) = (0, *, *, *)$$

for some $a(z), b(z) \in \mathbb{F}_2[z]$. Then $a(z) = 0$ and $b(z)$ is arbitrary. Hence

$$c(z) = b(z)(0, z^2+z+1, z, z^2+1),$$

which has no 1 entry.

Next, assume that

$$c(z) = a(z)(1,1,1,1) + b(z)(0, z^2+z+1, z, z^2+1) = (*, 0, *, *)$$

for some $a(z), b(z) \in \mathbb{F}_2[z]$. Then $a(z) = (z^2+z+1)b(z)$ and hence

$$c(z) = ((z^2+z+1)b(z), 0, (z^2+1)b(z), zb(z)),$$

which has no 1 entry.

Then, assume that

$$c(z) = a(z)(1,1,1,1) + b(z)(0, z^2+z+1, z, z^2+1) = (*, *, 0, *)$$

for some $a(z), b(z) \in \mathbb{F}_2[z]$. Then $a(z) = zb(z)$ and hence

$$c(z) = (zb(z), (z^2+1)b(z), 0, (z^2+z+1)b(z)),$$

which has no 1 entry.

Finally, assume that

$$c(z) = a(z)(1,1,1,1) + b(z)(0, z^2+z+1, z, z^2+1) = (*, *, *, 0)$$

for some $a(z), b(z) \in \mathbb{F}_2[z]$. Then $a(z) = (z^2+1)b(z)$ and hence

$$c(z) = ((z^2+1)b(z), zb(z), (z^2+z+1)b(z), 0),$$

which has no 1 entry.

We conclude that there is no codeword in $\mathcal{C}$ that has a 0 entry and a 1 entry simultaneously, meaning that the top-left $(1\ 0)$ in the Building-up construction can never be achieved by row operation or column permutations on $G(z)$. Hence $\mathcal{C}$ cannot be constructed with the generalized Building-up construction.

So the only thing left to show is $d_{free}(\mathcal{C}) > 2$. As a consequence of Lemma 3.3.13, every binary self-dual code has an even free distance and as the free distance is non-zero, we just need to show that there is no codeword of weight 2.

Assume conversely that there exists $c(z) = (c_1(z), c_2(z), c_3(z), c_4(z)) \in \mathcal{C}$ such that

$$c(z) = a(z)(1,1,1,1) + b(z)(0, z^2+z+1, z, z^2+1)$$

and $wt(c(z)) = 2$. We want to show that necessarily $a(z) = b(z) = 0$, which would contradict as

$$wt(c(z)) = wt(0*(1,1,1,1) + 0*(0, z^2+z+1, z, z^2+1)) = wt((0,0,0,0)) = 0.$$

The entries of $c(z)$ are given by

$$\begin{array}{rcl} c_1(z) &=& a(z) \\ c_2(z) &=& a(z) + (z^2+z+1)b(z) \\ c_3(z) &=& a(z) + zb(z) \\ c_4(z) &=& a(z) + (z^2+1)b(z) \end{array}$$

and at most two $c_i(z)$ can be non-zero.

If $a(z) = 0$, then $c_2(z), c_3(z)$ and $c_4(z)$ are three non-zero entries unless $b(z) = 0$ aswell. Hence $a(z) \neq 0$. But now we have at atleast three non-zero $c_i(z)$, since $c_1(z)$ is non-zero and atleast two of $c_2(z)$,$c_3(z)$ and $c_4(z)$ must be non-zero.
Another way to understand this argument is by looking at the $c(z)$ we found when we assumed there to be a 0 in any entry of $c(z)$ to show that there is no simultaneous 1 entry, and notice that all other entries were always non-zero (for $b(z) \neq 0$), implying $wt(c(z)) \geq 3$.

We conclude that $\mathcal{C}$ has free distance $d_{free} > 2$ and cannot be constructed with the generalized Building-up construction. □

As a consequence, the full classification of binary self-dual convolutional codes cannot be done by using the generalized Building-up construction, like some classification for block codes were. But maybe with the Harada-Munemasa construction?

First, we present a minor generalisation of the Harada-Munemasa construction.

Proposition 5.1.8. *Let $G(z) \in \mathbb{F}_2[z]^{k\times 2k}$ be the generator matrix of a binary self-dual $(2k, k)$ convolutional code and $a_i(z) \in \mathbb{F}_2[z]$, for $i \in \{1, \ldots, k\}$. Then*

$$\tilde{G}(z) = \begin{pmatrix} a_1(z) & a_1(z) & \\ \vdots & \vdots & G(z) \\ a_k(z) & a_k(z) & \end{pmatrix} \in \mathbb{F}_2[z]^{k\times(2k+2)}$$

generates a binary self-orthogonal convolutional code $\tilde{\mathcal{C}}$.

Proof. Let $g_i(z), g_j(z)$ be any two rows of $G(z)$ and $\tilde{g}_i(z)$ and $\tilde{g}_j(z)$ be the two corresponding rows of $\tilde{G}(z)$. Then

$$\tilde{g}_i(z)\tilde{g}_j(z)^T = a_i(z)a_j(z) + a_i(z)a_j(z) + g_i(z)g_j(z)^T = g_i(z)g_j(z)^T = 0$$

and hence every row of $\tilde{G}(z)$ is orthogonal to every other row and itself. Consequently, $\tilde{G}(z)$ generates a self-orthogonal code. □

Definition 5.1.9. *Let $G(z) \in \mathbb{F}_2[z]^{k\times 2k}$ be a generator matrix of a binary self-dual $(2k, k)$ convolutional code and for some $a_i(z) \in \mathbb{F}_2[z]$*

$$\tilde{G}(z) = \begin{pmatrix} a_1(z) & a_1(z) & \\ \vdots & \vdots & G(z) \\ a_k(z) & a_k(z) & \end{pmatrix} \in \mathbb{F}_2[z]^{k\times(2k+2)}$$

like in Proposition 5.1.8. If there exists $f(z) \in \mathbb{F}_2[z]^{2k+2}$ such that

$$G_1(z) = \begin{pmatrix} — & f(z) & — \\ a_1(z) & a_1(z) & \\ \vdots & \vdots & G(z) \\ a_k(z) & a_k(z) & \end{pmatrix} \in \mathbb{F}_2[z]^{(k+1)\times(2k+2)}$$

*generates a self-dual code, then we say that $G_1(z)$ is a **self-dual completion** or just a **completion** of $\tilde{G}(z)$.*

Naturally, we want to figure out when and how a self-dual completion can be found.

Lemma 5.1.10. *Choosing $f(z) = (1, 1, 0, \ldots, 0)$ in Definition 5.1.9 leads to a generator matrix of a self-dual code. More concretely,*

$$G_1(z) = \begin{pmatrix} 1 & 1 & 0 \\ a_1(z) & a_1(z) & \\ \vdots & \vdots & G(z) \\ a_k(z) & a_k(z) & \end{pmatrix} \in \mathbb{F}_2[z]^{(k+1)\times(2k+2)}$$

is a self-dual completion for any $a_i(z) \in \mathbb{F}_2[z]$.

Proof. Let

$$G_1(z) = \begin{pmatrix} 1 & 1 & 0 \\ a_1(z) & a_1(z) & \\ \vdots & \vdots & G(z) \\ a_k(z) & a_k(z) & \end{pmatrix}$$

be as in Definition 5.1.9 with $G(z)$ generating a binary self-dual $(2k, k)$ convolutional code. Then

$$G_1(z) = \begin{pmatrix} 1 & 1 & 0 \\ a_1(z) & a_1(z) & \\ \vdots & \vdots & G(z) \\ a_k(z) & a_k(z) & \end{pmatrix}$$

and

$$G_2(z) = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & \\ \vdots & \vdots & G(z) \\ 0 & 0 & \end{pmatrix}$$

generate the same code, making the choice of the $a_i(z)$ pointless. But still

$$G_2(z)G_2(z)^T = 0$$

and

$$G_2(z) \sim [I_{k+1}\ 0],$$

implying that the code generated by $G_2(z)$ is self-dual. □

As mentioned in the proof, this result is to no purpose to the Harada-Munemasa construction, because the application of the construction was pointless. This inspires us to make the following definition.

Definition 5.1.11. *Let*

$$G_1(z) = \begin{pmatrix} — & f(z) & — \\ a_1(z) & a_1(z) & \\ \vdots & \vdots & G(z) \\ a_k(z) & a_k(z) & \end{pmatrix} \in \mathbb{F}_2[z]^{(k+1)\times(2k+2)}$$

*be a self-dual completion. Then we say that the completion was **trivial** if $G_1(z)$ and*

$$\begin{pmatrix} 1 & 1 & 0 \\ a_1(z) & a_1(z) & \\ \vdots & \vdots & G(z) \\ a_k(z) & a_k(z) & \end{pmatrix} \in \mathbb{F}_2[z]^{(k+1)\times(2k+2)}$$

*generate the same code. If they do not, we call the completion **non-trivial**.*

An important observation here is that $(1, 1, 0, \ldots, 0)$ can never be a linear combination of the last k rows of $G_1(z)$, because the rows of $G(z)$ are linearly independent. On the other hand, we will see that the existence of a non-trivial completion will be depending only on the last k rows of $G_1(z)$, as the possibilities for $f(z)$ are uniquely determined by the last k rows of $G_1(z)$.

In the following example, we demonstrate how the choice of the $a_i(z)$ affects the existence of non-trivial self-dual completions.

Example 5.1.12. *We know that $(1\ 1)$ is a binary self-dual $(2, 1)$ convolutional code (actually the only one). Let's try to implement the Harada-Munemasa construction for $a_1(z) = z$ and show that for this choice of $a_1(z)$, there are only trivial self-dual completions.*

By Proposition 5.1.8,

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

generates a self-orthogonal code. We are now looking for $f(z) = (f_1, f_2, f_3, f_4) \in \mathbb{F}_2[z]^4$ such that

$$G(z) = \begin{pmatrix} f_1 & f_2 & f_3 & f_4 \\ z & z & 1 & 1 \end{pmatrix}$$

generates a self-dual code $\mathcal{C}$.

By Lemma 3.3.14, the all one vector must be part of the code. Therefore

$$b(z)G(z) = (1, 1, 1, 1)$$

for some $b(z) = (b_1, b_2) \in \mathbb{F}_2[z]^2$. Notice that necessarily $b_1 \neq 0$.
This translates into the following system of congruences over $\mathbb{F}_2[z]$

$$\begin{array}{rcl} b_1 f_1 + z b_2 & = & 1 \\ b_1 f_2 + z b_2 & = & 1 \\ b_1 f_3 + b_2 & = & 1 \\ b_1 f_4 + b_2 & = & 1 \end{array}.$$

Since $b_1 \neq 0$, by adding the last two equations we find that $f_4 = f_3$ and the last equation is equivalent to

$$b_2 = b_1 f_4 + 1.$$

Hence the system simplifies into

$$\begin{array}{rcl} b_1 f_1 + z(b_1 f_4 + 1) & = & 1 \\ b_1 f_2 + z(b_1 f_4 + 1) & = & 1 \end{array}$$

and similarly we find $f_1 = f_2$ by adding the two equations.

Finally $b_1 f_1 + z(b_1 f_4 + 1) = 1$ is equivalent to

$$b_1(f_1 + f_4 z) = 1 + z$$

and we conclude that either $b_1 = 1$ or $b_1 = z + 1$.

Let $b_1 = 1$, then $f_1 = 1 + z + z f_4$ and the generator matrix is given by

$$G(z) = \begin{pmatrix} 1 + z + f_4 z & 1 + z + f_4 z & f_4 & f_4 \\ z & z & 1 & 1 \end{pmatrix}$$

depending only on $f_4 \in \mathbb{F}_2[z]$.

We will show that this generator matrix necessarily generates a catastrophic code, by computing its Smith form.

Take the second row, multiply it by f_4 and add it to the first row

$$\begin{pmatrix} 1 + z + f_4 z & 1 + z + f_4 z & f_4 & f_4 \\ z & z & 1 & 1 \end{pmatrix} \sim \begin{pmatrix} 1 + z & 1 + z & 0 & 0 \\ z & z & 1 & 1 \end{pmatrix}.$$

Finally, use the last column to delete the second row and add the first column to the second to find

$$\begin{pmatrix} 1 + z & 1 + z & 0 & 0 \\ z & z & 1 & 1 \end{pmatrix} \sim \begin{pmatrix} 1 + z & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \sim \begin{pmatrix} 1 + z & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

that the Smith form of $G(z)$ is not $[I_k\ 0]$, implying that $G(z)$ generates a catastrophic and thereby not self-dual code.

Let $b_1 = z + 1$, then $f_1 = 1 + f_4 z$ and

$$G(z) = \begin{pmatrix} 1 + f_4 z & 1 + f_4 z & f_4 & f_4 \\ z & z & 1 & 1 \end{pmatrix}.$$

But $G(z)$ and

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

generate the same code, implying that $G(z)$ was a trivial self-dual completion.

We conclude that for this choice of $a_1(z)$ there are only trivial self-dual completions.

Now let $a_1(z) = 1$ instead of z, then

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

generates a self-orthogonal code and a non-trivial self-dual completion exists in the form of

$$\begin{pmatrix} 0 & z^2+z+1 & z & z^2+1 \\ 1 & 1 & 1 & 1 \end{pmatrix}.$$

This generates a self-dual code by section 4.2.

Therefore, the existence of non-trivial completions is depending on the choice of $a_i(z)$. This naturally steers to the search for viable conditions on the $a_i(z)$ that admit non-trivial self-dual completions.

Theorem 5.1.13. *Let*

$$\tilde{G}(z) = \begin{pmatrix} a_1(z) & a_1(z) & \\ \vdots & \vdots & G(z) \\ a_k(z) & a_k(z) & \end{pmatrix} \in \mathbb{F}_2[z]^{k\times(2k+2)}$$

Then a non-trivial self-dual completion exists if and only if

$$(1,\ldots,1) \in rowspan(\tilde{G}(z)).$$

Proof. Let $\tilde{\mathcal{C}} = rowspan(\tilde{G}(z))$,

$$G_1(z) = \begin{pmatrix} — & f(z) & — \\ a_1(z) & a_1(z) & \\ \vdots & \vdots & G(z) \\ a_k(z) & a_k(z) & \end{pmatrix}$$

be a self-dual completion for some $f(z) \in \mathbb{F}_2[z]^{2k+2}$ and $\mathcal{C}_1 = rowspan(G_1(z))$.

The main fact we will be using to prove this theorem will be that if

$$(1,\ldots,1) \in \tilde{\mathcal{C}},$$

then every possible new row $f(z) \in \tilde{\mathcal{C}}^{\perp}$ that we want to add to $\tilde{G}(z)$, fulfils

$$f(z)f(z) = (f(z)(1,\ldots,1)^T)^2 = 0,$$

as $(1,\ldots,1) \in \tilde{\mathcal{C}}$, implying that for

$$G_1(z) = \begin{pmatrix} f(z) \\ \tilde{G}(z) \end{pmatrix},$$

all rows of $G_1(z)$ are orthogonal.

On the other hand, if

$$(1,\ldots,1) \notin \tilde{\mathcal{C}},$$

then most $f(z) \in \tilde{\mathcal{C}}^{\perp}$ will not be orthogonal to themselves, meaning that

$$G_1(z) = \begin{pmatrix} f(z) \\ \tilde{G}(z) \end{pmatrix}$$

will not be a self-dual completion given these $f(z)$, as its first row is not orthogonal to itself.

($\Longrightarrow$) We will prove the contrapositive by showing that $(1, \ldots, 1) \notin \tilde{\mathcal{C}}$ implies that there exists exactly one self-dual completion (up to equivalent generator matrices) and it is the trivial one.

We know by the Rank-Nullity Theorem that $dim(\tilde{\mathcal{C}}^{\perp} / \tilde{\mathcal{C}}) = 2$ ($\tilde{\mathcal{C}} \subset \tilde{\mathcal{C}}^{\perp}$ by Proposition 5.1.8). Then, we also know that $(1, 1, 0, \ldots, 0) \in \tilde{\mathcal{C}}^{\perp} / \tilde{\mathcal{C}}$ by Lemma 5.1.10. So let

$$\tilde{\mathcal{C}}^{\perp} / \tilde{\mathcal{C}} = < (1, 1, 0, \ldots, 0), g(z) >$$

for some $g(z) \in \mathbb{F}_2[z]^{2k+2}$.

The reason, why we are so interested in $\tilde{\mathcal{C}}^{\perp} / \tilde{\mathcal{C}}$ is, because every existing self-dual completion of $\tilde{G}(z)$ can be achieved by adding a vector $f(z)$ of $\tilde{\mathcal{C}}^{\perp} / \tilde{\mathcal{C}}$ as a new row. And that's because adding a codeword of $\tilde{\mathcal{C}}$ to the to be added row $f(z)$ will not alter the new code $\mathcal{C}_1$ as the codewords of $\tilde{\mathcal{C}}$ are linear combinations of the last k rows of $G_1(z)$. As a consequence, all self-dual completions of $\tilde{\mathcal{C}}$ can be found by looking only at all possible new rows $f(z) \in \tilde{\mathcal{C}}^{\perp} / \tilde{\mathcal{C}}$ and checking the resulting $\mathcal{C}_1$ for self-duality.

We start off by showing that $g(z)$ can never be orthogonal to itself.

Assume conversely that $g(z)g(z)^T = 0$, then

$$(g(z)(1, \ldots, 1)^T)^2 = g(z)g(z)^T = 0.$$

Next, we know that $(1, \ldots, 1) \in rowspan(G(z))$, since $G(z)$ generates a binary self-dual code, and therefore

$$(b(z), b(z), 1, \ldots, 1) \in \tilde{\mathcal{C}}$$

for some $b(z) \in \mathbb{F}_2[z]$, where $b(z) \neq 1$, because $(1, \ldots, 1) \notin \tilde{\mathcal{C}}$ by assumption. Now as $g(z) \in \tilde{\mathcal{C}}^{\perp}$,

$$g(z)(b(z), b(z), 1, \ldots, 1)^T = 0$$

and hence

$$g(z)(b(z) + 1, b(z) + 1, 0, \ldots, 0)^T = g(z)(b(z), b(z), 1, \ldots, 1)^T + g(z)(1, \ldots, 1)^T = 0,$$

implying with $b(z) + 1 \neq 0$, that

$$g(z)(1, 1, 0, \ldots, 0)^T = 0.$$

Finally, we know by Lemma 5.1.10 that

$$G_2(z) := \begin{pmatrix} 1 & 1 & 0 \\ a_1(z) & a_2(z) & \\ \vdots & \vdots & \tilde{G}(z) \\ a_k(z) & a_k(z) & \end{pmatrix}$$

generates a binary self-dual convolutional code. We just argued that $g(z)$ is orthogonal to the first row of $G_2(z)$ and it is orthogonal to every other row by definition, implying

$$g(z) \in ker(G_2) = rowspan(G_2),$$

where the last equality follows from $G_2(z)$ generating a self-dual code. But this contradicts the assumption of $g(z)$ generating $\tilde{\mathcal{C}}^{\perp} / \tilde{\mathcal{C}}$ with $(1, 1, 0, \ldots, 0)$, since $g(z)$ was just shown to be a linear combination of a codeword of $\tilde{\mathcal{C}}$ and $(1, 1, 0, \ldots, 0)$.

We conclude that $g(z)$ cannot be orthogonal to itself.

Now let $f(z) \in \tilde{\mathcal{C}}^{\perp} / \tilde{\mathcal{C}}$ or equivalently $f(z) = a(z)(1, 1, 0, \ldots, 0) + b(z)g(z)$ for some $a(z), b(z) \in \mathbb{F}_2[z]$, then

$$\begin{aligned} f(z)f(z)^T &= (a(z)(1,1,0,\ldots,0) + b(z)g(z))(a(z)(1,1,0,\ldots,0) + b(z)g(z))^T = \\ &= a(z)^2(1,1,0,\ldots,0)(1,1,0,\ldots,0)^T + 2a(z)b(z)(1,1,0,\ldots,0)g(z)^T + b(z)^2 g(z)g(z)^T = \\ &= b(z)^2 g(z)g(z)^T. \end{aligned}$$

Hence the only $f(z)$ that are orthogonal to themselves are the ones, where $b(z) = 0$, meaning that these are the only possible new rows leading to a self-dual completion.

In other words,

$$G_1(z) = \begin{pmatrix} a(z) & a(z) & 0 \\ a_1(z) & a_2(z) & \\ \vdots & \vdots & G(z) \\ a_k(z) & a_k(z) & \end{pmatrix}$$

are the only possibly self-dual completions left. Now if $a(z) = 0$, we have a generator matrix that does not have full rank, and if $a(z)$ is non-constant, $G_1(z)$ does not generate a non-catastrophic code, since its row Hermite form is not $[I_k \; 0]$.

Conclusively,

$$G_1(z) = \begin{pmatrix} 1 & 1 & 0 \\ a_1(z) & a_2(z) & \\ \vdots & \vdots & \tilde{G}(z) \\ a_k(z) & a_k(z) & \end{pmatrix}$$

is the only existing self-dual completion of $\tilde{G}(z)$ (up to equivalent generator matrices) and it indeed is the trivial one.

($\Longleftarrow$) $dim(\tilde{\mathcal{C}}^{\perp} / \tilde{\mathcal{C}}) = 2$ still holds, so let

$$\tilde{\mathcal{C}}^{\perp} / \tilde{\mathcal{C}} = < (1, 1, 0, \ldots), f(z) >$$

for some $f(z) = (f_1(z), \ldots, f_{2k+2}) \in \mathbb{F}_2[z]^{2k+2}$. We want to show that the code $\mathcal{C}_1$ generated by

$$G_1(z) = \begin{pmatrix} f(z) \\ \tilde{G}(z) \end{pmatrix}$$

is a non-trivial completion.

First off, we will show that $(1,1,0,\ldots,0) \notin \mathcal{C}_1^{\perp}$.

Assume conversely that $(1,1,0,\ldots,0) \in \mathcal{C}_1^{\perp}$, then

$$f(z)(1,1,0,\ldots,0)^T = 0,$$

implying $f_1(z) = f_2(z)$.

Let $g_i(z)$ be the i-th row of $G(z)$ and $g_{ij}(z)$ its i,j-th entry. Since $f(z) \in \tilde{\mathcal{C}}^{\perp}$, it is orthogonal to the i-th row of $\tilde{G}(z)$ in

$$f(z)(a_1(z), a_2(z), g_{i1}(z), \ldots, g_{i2k}(z))^T = 0,$$

where $1 \leq i \leq k$. But also

$$f(z)(a_1(z), a_2(z), g_{i1}(z), \ldots, g_{i2k}(z))^T = a_1(z)f_1(z) + a_1(z)f_2(z) + (f_3(z), \ldots, f_{2k+2}(z))g_i(z)^T =$$

$$= (f_3(z), \ldots, f_{2k+2}(z))g_i(z)^T.$$

Hence $(f_3(z), \ldots, f_{2k+2}(z))$ is also orthogonal to every row of $G(z)$ and as $G(z)$ generates a self-dual convolutional code,

$$(f_3(z), \ldots, f_{2k+2}(z)) \in rowspan(G(z))$$

follows.

Using this fact, we know that there exists $\tilde{c}(z) \in \tilde{\mathcal{C}}$ such that

$$\tilde{c}(z) = (*, *, f_3(z), \ldots, f_{2k+2}(z)),$$

where the first two entries are some linear combination of the $a_i(z)$. Therefore

$$(b(z), b(z), 0, \ldots, 0) = f(z) + \tilde{c}(z) \in \mathcal{C}_1$$

for some $b(z) \in \mathbb{F}_2[z]$, which in turn implies

$$G_1(z) \sim \begin{pmatrix} b(z) & b(z) & 0 & \ldots & 0 \\ & & \tilde{G}(z) & & \end{pmatrix}$$

(notice that we established $\tilde{c} \in \tilde{\mathcal{C}}$ to make sure that we can apply the unimodular operation of adding multiples of the last k rows to the first row of $G_1(z)$). As both matrices must have row Hermite form $[I_{k+1}\ 0]$, we conclude that $b(z) = 1$. But now

$$f(z) + (1,1,0,\ldots,0) = \tilde{c}(z) \in \tilde{\mathcal{C}},$$

which is impossible as $f(z)$ and $(1,1,0\ldots,0)$ are basis vectors and therefore linearly independent vectors of $\tilde{\mathcal{C}}^{\perp}/\tilde{\mathcal{C}}$.

We conclude that $(1,1,0\ldots,0) \notin \mathcal{C}_1^{\perp}$.

Our next goal is to show that $G_1(z)$ is not only a generator matrix of $\mathcal{C}_1$, but also a parity-check matrix to then conclude with Theorem 3.3.9. But first we compute the generator matrix of $\tilde{\mathcal{C}}^{\perp}$.

We want to show that

$$\tilde{H}(z) = \begin{pmatrix} 1 & 1 & 0 \\ — & f(z) & — \\ a_1(z) & a_1(z) & \\ \vdots & \vdots & G(z) \\ a_k(z) & a_k(z) & \end{pmatrix}$$

is a generator matrix of $\tilde{\mathcal{C}}^{\perp}$ as in

$$\tilde{\mathcal{C}}^{\perp} = rowspan(\tilde{H}(z)).$$

For this let $h(z) \in \tilde{\mathcal{C}}^{\perp}$, then either

$$h(z) \in \tilde{\mathcal{C}} \text{ or } h(z) \in \tilde{\mathcal{C}}^{\perp} \setminus \tilde{\mathcal{C}}$$

as $\tilde{\mathcal{C}} \subset \tilde{\mathcal{C}}^{\perp}$ by Proposition 5.1.8. But

$$\tilde{\mathcal{C}} = rowspan(\tilde{G}(z)) \text{ and } \tilde{\mathcal{C}}^{\perp} / \tilde{\mathcal{C}} =< f(z), (1, 1, 0, \ldots) >,$$

showing $h(z) \in rowspan(\tilde{H}(z))$.

Conversely, let $h(z) \in rowspan(\tilde{H}(z))$. All rows of $\tilde{H}(z)$ are orthogonal to the rows of $\tilde{G}(z)$. More specifically, the first two rows of $\tilde{H}(z)$ are in the dual of $\tilde{\mathcal{C}}$ as

$$\tilde{\mathcal{C}}^{\perp} / \tilde{\mathcal{C}} =< f(z), (1, 1, 0, \ldots) >$$

and the last k rows are in the dual, since $\tilde{\mathcal{C}}$ is self-orthogonal. Conclusively, $h(z)$ is a linear combination of rows that are orthogonal to every row of $\tilde{G}(z)$ and hence

$$\tilde{G}(z)h(z)^T = 0$$

or rather

$$h(z) \in ker(\tilde{G}(z)) = \tilde{\mathcal{C}}^{\perp}.$$

Now we use the fact that

$$\mathcal{C}_1^{\perp} \subset \tilde{\mathcal{C}}^{\perp} = rowspan(\tilde{H}(z)) \tag{7}$$

to show that

$$\mathcal{C}_1^{\perp} = rowspan(G_1(z)),$$

where still $G_1(z) = \begin{pmatrix} f(z) \\ \tilde{G}(z) \end{pmatrix}$.

Let $h(z) \in rowspan(G_1(z))$. The last k rows of $G_1(z)$ are orthogonal to themselves as $\tilde{\mathcal{C}}$ is self-orthogonal and the first row is orthogonal to all others as $f(z) \in \tilde{\mathcal{C}}^{\perp}$. Furthermore, the first row is also orthogonal to itself as

$$f(z)f(z)^T = (f(z)(1, \ldots, 1)^T)^2 = 0,$$

since $(1, \ldots, 1)$ is a codeword of $\tilde{\mathcal{C}}$ by assumption. Finally as $h(z)$ is a linear combination of rows of $G_1(z)$,

$$G_1(z)h(z)^T = 0$$

or rather

$$h(z) \in ker(G_1(z)) = \mathcal{C}_1^{\perp}$$

follows.

Conversely let $h(z) \in \mathcal{C}_1^{\perp}$, then there are two things we know about $h(z)$:
Firstly by (7), $h(z) \in rowspan(\tilde{H}(z))$ or rather there exists $m(z) = (m_1(z), \ldots, m_{k+2}(z)) \in \mathbb{F}_2[z]^{k+2}$ such that

$$h(z) = m(z)\tilde{H}(z).$$

And secondly,

$$G_1(z)h(z)^T = 0.$$

Combining the two, we find

$$G_1(z)h(z)^T = G_1(z)\tilde{H}(z)^T m(z)^T = 0. \tag{8}$$

Our goal is now to show that $m_1(z) = 0$.

If we look at $G_1(z)\tilde{H}(z)^T$ closely, we see that

$$\begin{aligned} G_1(z)\tilde{H}(z)^T &= G_1(z)\begin{pmatrix} 1 & 1 & 0 & \cdots & 0 \\ & & G_1(z) & & \end{pmatrix}^T = \\ &= \begin{pmatrix} G_1(z)(1,1,0,\ldots,0)^T & G_1(z)G_1(z)^T \end{pmatrix} = \\ &= \begin{pmatrix} f(z)(1,1,0,\ldots,0)^T & f(z)f(z)^T & 0 \\ 0 & 0 & 0 \end{pmatrix} = \\ &= \begin{pmatrix} f_1(z) + f_2(z) & 0 \\ 0 & 0 \end{pmatrix}. \end{aligned}$$

But we have seen in the proof of $(1,1,0,\ldots,0) \notin \mathcal{C}_1^{\perp}$ that necessarily $f_1(z) \neq f_2(z)$ and therefore there is exactly one non-zero entry in this matrix product and it is the entry in the first row and first column. Hence we need $m_1(z) = 0$ for (8) to be fulfilled and we conclude that $h(z)$ is a linear combination of the the last $k+1$ rows of $\tilde{H}(z)$ or equivalently

$$h(z) \in rowspan(G_1(z)).$$

Conclusively, we started off by showing $(1,1,0,\ldots,0) \notin \mathcal{C}_1^{\perp}$ and then showed that $G_1(z)$ is not only a generator matrix matrix, but also a parity-check matrix of $\mathcal{C}_1$. By Theorem 3.3.9, this implies that $\mathcal{C}_1$ is a self-dual convolutional code or equivalently $\mathcal{C}_1 = \mathcal{C}_1^{\perp}$, also proving non-triviality as now $(1,1,0,\ldots,0) \notin \mathcal{C}_1$ follows. □

So, if we are looking for non-trivial completions, we can find one in two steps:

1. Make sure that $(1,\ldots,1) \in rowspan(\tilde{G}(z)) = \tilde{\mathcal{C}}$, which means we must establish that the system
$$b(z)\tilde{G}(z) = (1,\ldots,1) \in \mathbb{F}_2[z]^{2k+2}, \tag{9}$$
where $b(z) = (b_1,\ldots,b_k) \in \mathbb{F}_2[z]^k$, has a solution for $b(z)$.

2. Find a vector $f(z) \in \tilde{\mathcal{C}}^{\perp}$ that forms a basis of $\tilde{\mathcal{C}}^{\perp}/\tilde{\mathcal{C}}$ with $(1,1,0,\ldots,0)$ and then
$$G_1(z) = \begin{pmatrix} — & f(z) & — \\ a_1(z) & a_1(z) & \\ \vdots & \vdots & G(z) \\ a_k(z) & a_k(z) & \end{pmatrix}$$
generates a binary self-dual convolutional code.

Let's try to clarify and simplify these two steps.

Lemma 5.1.14. *An equivalent way to making sure that* (9) *has a solution is by finding a solution to*

$$b(z)G(z) = (1, \dots, 1) \in \mathbb{F}_2[z]^{2k},$$

which always exists, and then choosing the $a_i(z) \in \mathbb{F}_2[z]$ *based on*

$$\sum_{i=1}^{k} a_i(z)b_i(z) = 1$$

holding.

Proof. Remember that

$$\tilde{G}(z) = \begin{pmatrix} a_1(z) & a_1(z) & \\ \vdots & \vdots & G(z) \\ a_k(z) & a_k(z) & \end{pmatrix}$$

for some $a_i(z) \in \mathbb{F}_2[z]$.

The existence of $b(z) = (b_1(z), \dots, b_k(z)) \in \mathbb{F}_2[z]^k$ follows from $G(z)$ generating a binary self-dual code (and Lemma 3.3.14). And the existence of $a_i(z)$ fulfilling $\sum_{i=1}^{k} a_i(z)b_i(z) = 1$, follows from

$$gcd(b_1(z), \dots, b_k(z)) = 1.$$

Let $b(z)$ fulfil

$$b(z)G(z) = (1, \dots, 1) \in \mathbb{F}_2[z]^{2k}$$

and

$$\sum_{i=1}^{k} a_i(z)b_i(z) = 1,$$

then

$$b(z)\tilde{G}(z) = (\sum_{i=1}^{k} a_i(z)b_i(z), \sum_{i=1}^{k} a_i(z)b_i(z), b(z)G(z)) = (1, \dots, 1).$$

Conversely, let there be a $b(z) \in \mathbb{F}_2[z]^k$ such that

$$b(z)\tilde{G}(z) = (1, \dots, 1) \in \mathbb{F}_2[z]^{2k+2},$$

then

$$\sum_{i=1}^{k} a_i(z)b_i(z) = b(z)(a_1(z), \dots, a_k(z))^T = 1,$$

since $(a_1(z), \dots, a_k(z))^T$ is the first column of $\tilde{G}(z)$, and

$$b(z)G(z) = (1, \dots, 1),$$

since $G(z)$ are the last $2k$ columns of $\tilde{G}(z)$. □

So we have successfully found conditions for the $a_i(z)$ that lead to the existence of non-trivial completions, but given such $a_i(z)$, how do we find a $f(z)$? One thing we know for sure is that the first two entries of $f(z)$ cannot be equal (see proof of Theorem 5.1.13). This property and

$$< f(z), (1,1,0,\ldots,0) >= \tilde{\mathcal{C}}^{\perp} / \tilde{\mathcal{C}}$$

are all the conditions we have at this point.

Lemma 5.1.15. *Let* $f(z) = (f_1(z), \ldots, f_{2k+2}(z)) \in \tilde{\mathcal{C}}^{\perp} / < \tilde{\mathcal{C}}, (1,1,0,\ldots,0) >$ *and*

$$gcd(f_1(z), \ldots, f_{2k+2}(z)) = 1,$$

then

$$< f(z), (1,1,0,\ldots,0) >= \tilde{\mathcal{C}}^{\perp} / \tilde{\mathcal{C}}.$$

Proof. Let $f(z)$ be as assumed and assume conversely that there exists $h(z) \in \tilde{\mathcal{C}}^{\perp} / \tilde{\mathcal{C}}$ such that $h(z) \neq f(z)$ and

$$< h(z), (1,1,0,\ldots,0) >= \tilde{\mathcal{C}}^{\perp} / \tilde{\mathcal{C}}.$$

Let $\tilde{h}(z)$ be the element in $\tilde{\mathcal{C}}^{\perp} / < \tilde{\mathcal{C}}, (1,1,0,\ldots,0) >$ corresponding to $h(z)$.

As $f(z) \in \tilde{\mathcal{C}}^{\perp} / < \tilde{\mathcal{C}}, (1,1,0,\ldots,0) > =< \tilde{h}(z) >$, $f(z)$ must be a multiple of $\tilde{h}(z)$. But since the entries of $f(z)$ have no common non-trivial divisor,

$$f(z) = \tilde{h}(z)$$

follows immediately. Finally

$$< f(z), (1,1,0,\ldots,0) >=< \tilde{h}(z), (1,1,0,\ldots,0) >=< h(z), (1,1,0,\ldots,0) >= \tilde{\mathcal{C}}^{\perp} / \tilde{\mathcal{C}}$$

□

So given a good choice of the $a_i(z)$ ("good choice" meaning that a non-trivial completion exists), we may find a $f(z)$ in $\tilde{\mathcal{C}}^{\perp} / < \tilde{\mathcal{C}}, (1,1,0,\ldots,0) >$ and divide it by the greatest common divisor of its entries, leading to $\tilde{f}(z)$. Finally adding any $g(z)$, that is in the same equivalence class as $\tilde{f}(z)$ with respect to $\tilde{\mathcal{C}}^{\perp} / < \tilde{\mathcal{C}}, (1,1,0,\ldots,0) >$, as a new row will hand us a self-dual code.

Proposition 5.1.16 (Finding non-trivial completions)**.** *Let* $G(z)$ *be the generator matrix of a binary self-dual* $(2k, k)$ *convolutional code. If we want to construct a new binary self-dual* $(2k+2, k+1)$ *convolutional code that was non-trivially completed, we do the following:*

1. *Find* $b(z) \in \mathbb{F}_2[z]^k$ *such that*

$$b(z)G(z) = (1,\ldots,1) \in \mathbb{F}_2[z]^{2k}.$$

2. *Find* $a_i(z) \in \mathbb{F}_2[z]$*, for* $1 \leq i \leq k$*, fulfilling*

$$\sum_{i=1}^{k} a_i(z)b_i(z) = 1.$$

3. *Find a non-zero* $f(z) \in \tilde{\mathcal{C}}^{\perp} / < \tilde{\mathcal{C}}, (1,1,0,\ldots,0) >$ *and divide it by the greatest common divisor of its entries, calling the result* $\tilde{f}(z)$*.*

4. *Finally, every element $g(z)$ that is in the same equivalence class as $\tilde{f}(z)$ with respect to $\tilde{\mathcal{C}}^{\perp}/<\tilde{\mathcal{C}},(1,1,0,\ldots,0)>$ will lead to a binary self-dual convolutional code with*

$$G_1(z)=\begin{pmatrix} — & g(z) & — \\ a_1(z) & a_1(z) & \\ \vdots & \vdots & G(z) \\ a_k(z) & a_k(z) & \end{pmatrix}.$$

Proof. It was shown in Lemma 5.1.14 that solutions to step 1 and 2 can always be found. Further, given this unique $b(z)$ and these $a_i(z)$,

$$b(z)\begin{pmatrix} a_1(z) & a_1(z) & \\ \vdots & \vdots & G(z) \\ a_k(z) & a_k(z) & \end{pmatrix}=(1,\ldots,1)$$

holds true and as a consequence there exists a non-trivial completion (Theorem 5.1.13). Moreover, it was shown in Theorem 5.1.13 that if we find a basis vector $f(z)$ that generates $\tilde{\mathcal{C}}^{\perp}/\tilde{\mathcal{C}}$ with $(1,1,0,\ldots,0)$, then

$$\begin{pmatrix} — & f(z) & — \\ a_1(z) & a_1(z) & \\ \vdots & \vdots & G(z) \\ a_k(z) & a_k(z) & \end{pmatrix}$$

is a non-trivial self-dual completion.

In the fourth step, we assumed $g(z)$ to be in the same equivalence class as $\tilde{f}(z)$ or equivalently

$$g(z)=\tilde{f}(z)+a(z)(1,1,0,\ldots,0)+\sum_{i=1}^{k}d_i(z)\tilde{g}_i(z)$$

for some $a(z),d_i(z)\in\mathbb{F}_2[z]$ and $\tilde{g}_i(z)$ being the i-th row of $\tilde{G}(z)$.

It was shown in Lemma 5.1.15, that the $\tilde{f}(z)$ we find in the third step fulfils the property of

$$<\tilde{f}(z),(1,1,0,\ldots,0)=\tilde{\mathcal{C}}^{\perp}/\tilde{\mathcal{C}}.$$

But also,

$$G_1(z)=\begin{pmatrix} — & g(z) & — \\ a_1(z) & a_1(z) & \\ \vdots & \vdots & G(z) \\ a_k(z) & a_k(z) & \end{pmatrix}$$

and

$$G_2(z)=\begin{pmatrix} — & \tilde{f}(z)+a(z)(1,1,0,\ldots,0) & — \\ a_1(z) & a_1(z) & \\ \vdots & \vdots & G(z) \\ a_k(z) & a_k(z) & \end{pmatrix}$$

generate the same code, whereby

$$<\tilde{f}(z)+a(z)(1,1,0,\ldots,0),(1,1,0,\ldots,0)>=<\tilde{f}(z),(1,1,0,\ldots,0)>=\tilde{\mathcal{C}}^{\perp}/\tilde{\mathcal{C}}.$$

Hence $G_2(z)$ generates a binary self-dual convolutional code and so does $G_1(z)$. □

Example 5.1.17. *Let*

$$G(z) = \begin{pmatrix} 0 & z^2+z+1 & z & z^2+1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \in \mathbb{F}_2[z]^{2\times 4}.$$

We know by section 4.2, that $G(z)$ generates a self-dual convolutional code. Let's look for non-trivial completions by applying the steps of Proposition 5.1.16.

By inspection,

$$b(z)G(z) = (1,1,1,1)$$

holds exactly for $b(z) = (0,1)$. Hence we need

$$0 * a_1(z) + 1 * a_2(z) = 1$$

or equivalently $a_2(z) = 1$ and $a_1(z)$ may be chosen arbitrarily. So let $a_1(z) = z^2+1$ and $a_2(z) = 1$, then

$$\tilde{G}(z) = \begin{pmatrix} z^2+1 & z^2+1 & 0 & z^2+z+1 & z & z^2+1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

We are now looking for a $f(z) \in \tilde{\mathcal{C}}^{\perp} / < \tilde{\mathcal{C}}, (1,1,0,\ldots,0) >$. One thing we know about $f(z)$ is that $f_1(z) \neq f_2(z)$ and not surprisingly this guarantees $f(z) \notin < \tilde{\mathcal{C}}, (1,1,0,0,0,0) >$.

Further, we need $f(z)$ to be orthogonal to both rows of $\tilde{G}(z)$ as in

$$\begin{aligned} f(z)(z^2+1, z^2+1, 0, z^2+z+1, z, z^2+1)^T &= 0 \\ f(z)(1,1,1,1,1,1)^T &= 0 \end{aligned}$$

and see that this system holds for

$$f(z) = (0,1,0,0,0,1).$$

As the entries of $f(z)$ have no common polynomial divisor, we may add

$$(z, z+1, 0, 0, 0, 1) = f(z) + z(1,1,0,0,0,0)$$

as a new row and hence

$$G_1(z) = \begin{pmatrix} z & z+1 & 0 & 0 & 0 & 1 \\ z^2+1 & z^2+1 & 0 & z^2+z+1 & z & z^2+1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

generates a binary self-dual convolutional code by Proposition 5.1.16.

Let's proof this last step more thoroughly.

Firstly, $G_1(z)G_1(z)^T = 0$ holds as we chose $f(z)$ based on this condition and $f(z)f(z)^T = 0$.

Secondly, there are two 3-rd minors in

$$det\left[\begin{pmatrix} 0 & 0 & 1 \\ 0 & z & z^2+1 \\ 1 & 1 & 1 \end{pmatrix}\right] = z$$

and

$$det\left[\begin{pmatrix} 0 & 0 & 1 \\ 0 & z^2+z+1 & z^2+1 \\ 1 & 1 & 1 \end{pmatrix}\right] = z^2+z+1$$

that generate $\mathbb{F}_2[z]$ together. Hence $G_1(z)$ generates a non-catastrophic and self-orthogonal code, which let's us conclude that the code is indeed self-dual.

An immediate consequence of the second step of Proposition 5.1.16, is that there must be a linear combination of the $a_i(z)$ that is equal to 1. But this is not possible, if they have a common non-trivial divisor, giving us the following corollary.

Corollary 5.1.18. *Let*

$$G_1(z) = \begin{pmatrix} — & f(z) & — \\ a_1(z) & a_1(z) & \\ \vdots & \vdots & G(z) \\ a_k(z) & a_k(z) & \end{pmatrix} \in \mathbb{F}_2[z]^{(k+1)\times(2k+2)}$$

be a self-dual completion, where

$$gcd(a_1(z), \ldots, a_k(z)) \neq 1.$$

Then the completion was trivial.

The converse is not true in general as can be seen in the next example.
Furthermore, we illustrate in the upcoming example that a good choice of the $a_i(z)$ ("good choice" meaning that a non-trivial completion exists) differs for equivalent generator matrices.

Example 5.1.19. *By Lemma 5.1.13, if we chose $a_1 = a_2 = 1$ for*

$$\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \in \mathbb{F}_2[z]^{2\times 4},$$

as in

$$\tilde{G}(z) = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 \end{pmatrix},$$

then $\tilde{G}(z)$ can only be trivially completed as $(1,1,1,1,1,1) \notin rowspan(\tilde{G}(z))$. Now, we claim that another generator matrix of the same code in

$$G(z) = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \in \mathbb{F}_2[z]^{2\times 4}$$

has a non-trivial completion for $a_1(z) = a_2(z) = 1$ in

$$G_1(z) = \begin{pmatrix} h(z) & h(z)+1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 \end{pmatrix},$$

where $h(z) \in \mathbb{F}_2[z]$.

$G_1(z)$ generates a non-catastrophic code, since

$$det\left[\begin{pmatrix} h(z) & h(z)+1 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix}\right] = h(z) + 1 + h(z) = 1$$

and

$$G_1(z)G_1(z)^T = 0$$

by inspection. Hence $G_1(z)$ generates a self-dual code by Theorem 3.3.9.

So the only thing left to show is non-triviality.

Assume by contradiction that there exists $b(z) = (b_1, b_2, b_3) \in \mathbb{F}_2[z]^3$ *such that*

$$b(z)G_1(z) = (1, 1, 0, 0, 0, 0),$$

or rather, in terms of a system of congruences,

$$\begin{array}{rcl} b_1h(z) + b_2 + b_3 & = & 1 \\ b_1(h(z) + 1) + b_2 + b_3 & = & 1 \\ b_2 & = & 0 \\ b_2 & = & 0 \\ b_2 + b_3 & = & 0 \\ b_1 + b_2 + b_3 & = & 0 \end{array}.$$

But the last three equations imply $b_1 = b_2 = b_3 = 0$*, which is a contradiction and we conclude that the completion was non-trivial.*

Furthermore, if there are two generator matrices of the same code that differ by given row operations, then from a good choice of the $a_i(z)$ for one matrix, we can find a good choice for the other matrix by applying the given row operations to the already known good choice in $(a_1(z), \ldots, a_k(z))^T$.

We were previously asking the question, whether all binary self-dual convolutional codes with $d_{free} > 2$ can be constructed with the generalized Harada-Munemasa construction. This question will be left unanswered for codes of lengths greater than 4. But still there are some things that can be said about it.
To find a contradicting code, one needs to find a generator matrix, where there are no two columns (up to row operations) that have matching polynomials in each expect at most one entry (otherwise it can be constructed using the generalized Harada-Munemasa construction).
To go the other way and proof the statement, one needs to show that there exists a generator matrix for every binary self-dual convolutional code that contains two columns, having matching polynomials in all expect at most one entry.

Lemma 5.1.20. *All binary self-dual* $(4, 2)$ *convolutional codes can be constructed with the generalized Harada-Munemasa construction.*

Proof. Let $\mathcal{C}$ be a binary self-dual $(4, 2)$ convolutional code. Then by section 4.2, there exists a generator matrix of the form

$$G_1(z) = \begin{pmatrix} 0 & g(z) + h(z) & g(z) & h(z) \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

for some $g(z), h(z) \in \mathbb{F}_2[z]$.

But now by setting $G(z) = (1\ 1)$, $a_1(z) = 1$ and $f(z) = (0, g(z) + h(z), g(z), h(z))$, we may construct $G_1(z)$ and therefore $\mathcal{C}$ with the generalized Harada-Munemasa construction. □

Finally, we want to connect the two constructions.

Lemma 5.1.21. *Every code that was constructed with the generalized Building-up construction, can be constructed with the generalized Harada-Munemasa construction, but not vice versa.*

Proof. Let $G(z)$ be a generator matrix of a binary self-dual convolutional code and let

$$G_1(z) = \begin{pmatrix} 1 & 0 & f(z) \\ y_1(z) & y_1(z) & \\ \vdots & \vdots & G(z) \\ y_k(z) & y_k(z) & \end{pmatrix}$$

be a generator matrix freshly constructed with the Building-up construction, for some $f(z) \in \mathbb{F}_2[z]^{2k}$. To construct the same code with the Harada-Munemasa construction, just set

$$a_i(z) := y_i(z)$$

to get

$$\tilde{G}(z) = \begin{pmatrix} a_1(z) & a_1(z) & \\ \vdots & \vdots & G(z) \\ a_k(z) & a_k(z) & \end{pmatrix} = \begin{pmatrix} y_1(z) & y_1(z) & \\ \vdots & \vdots & G(z) \\ y_k(z) & y_k(z) & \end{pmatrix}$$

and then add $(1, 0, f(z))$ as a new row. And we have just successfully constructed

$$G_1(z) = \begin{pmatrix} 1 & 0 & f(z) \\ y_1(z) & y_1(z) & \\ \vdots & \vdots & G(z) \\ y_k(z) & y_k(z) & \end{pmatrix}.$$

Finally, in Lemma 5.1.7, it was shown that

$$\begin{pmatrix} 0 & z^2+z+1 & z & z^2+1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

cannot be constructed using the generalized Building-up construction. But by the previous lemma, all binary self-dual $(4, 2)$ convolutional codes can be constructed with the generalized Harada-Munemasa construction. □

We will conclude this section by again presenting all the facts we know about the generalized Harada-Munemasa construction:

1. A trivial completion always exists;
2. A good choice of the $a_i(z)$ might work for one generator matrix, but not for an equivalent one;
3. A non-trivial completion exists if and only if $(1, \ldots, 1) \in rowspan(\tilde{G}(z))$;
4. All binary self-dual $(4, 2)$ convolutional codes can be constructed with the generalized Harada-Munemasa construction;
5. The generalized Harada-Munemasa construction can construct strictly more codes than the generalized Building-up construction.

5.2 Examples of Self-Dual Convolutional Codes

In this subsection, we want to first find a binary self-dual $(10, 5)$ convolutional code of arbitrary degree and then we will be providing another example of a binary self-dual $(10, 5)$ convolutional code with degree 1. All examples are constructed by applying the generalized Building-up construction.

Example 5.2.1. *We found in Example 5.1.6*

$$G_2(z) = \begin{pmatrix} 1 & 0 & 1 & z & z^2 & z^2+z \\ 1 & 1 & 1 & 1 & 1 & 1 \\ z & z & 0 & 1 & z+1 & z \end{pmatrix}$$

to generate a binary self-dual convolutional code. Let's go one step further and find a binary self-dual $(8, 4)$ convolutional code.

Let $f(z) = (z, z^2+z, z^3, z^3+1, 0, z^2) \in \mathbb{F}_2[z]^6$, then $f(z)(1,1,1,1,1,1)^T = 1$ as desired and

$$\begin{array}{rclcrcl} y_1(z) & = & f(z)(1,0,1,z,z^2,z^2+z)^T & = & 0 \\ y_2(z) & = & f(z)(1,1,1,1,1,1)^T & = & 1 \\ y_3(z) & = & f(z)(z,z,0,1,z+1,z)^T & = & z^3+1 \end{array} .$$

Therefore, by the generalized Building-up construction (Proposition 5.1.5)

$$G_3(z) = \begin{pmatrix} 1 & 0 & z & z^2+z & z^3 & z^3+1 & 0 & z^2 \\ 0 & 0 & 1 & 0 & 1 & z & z^2 & z^2+z \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ z^3+1 & z^3+1 & z & z & 0 & 1 & z+1 & z \end{pmatrix}$$

generates a binary self-dual $(8, 4)$ convolutional code $\mathcal{C}_3$.

Moreover, we want to calculate $\mathcal{C}_3$'s degree δ.
Firstly, δ is definitely equal to or smaller than the sum of row degrees, being 8 (Remark 3.1.14).
Secondly, the determinant of the submatrix containing the 2nd, 4th, 5th and 7th column is

$$det\left[\begin{pmatrix} 0 & z^2+z & z^3 & 0 \\ 0 & 0 & 1 & z^2 \\ 1 & 1 & 1 & 1 \\ z^3+1 & z & 0 & z+1 \end{pmatrix}\right] = det\left[\begin{pmatrix} z^2+z & z^3 & 0 \\ 0 & 1 & z^2 \\ z & 0 & z+1 \end{pmatrix}\right] + (z^3+1) det\left[\begin{pmatrix} z^2+z & z^3 & 0 \\ 0 & 1 & z^2 \\ 1 & 1 & 1 \end{pmatrix}\right] =$$

$$= z^6 + z^3 + z + (z^3+1)(z^5+z^4+z^3+z^2+z) = z^8 + z^7 + z^2,$$

implying $\delta \geq 8$ and letting us conclude that $\delta = 8$.

Hence $\mathcal{C}_3$ is a binary self-dual $[8, 4, 8]$ convolutional code.

Then we construct a $(10, 5)$ code.

Example 5.2.2. *Let $f(z) = (0,0,1,0,0,0,0,0)$, then $f(z)(1,1,1,1,1,1,1,1)^T = 1$ as desired and*

$$\begin{array}{rclcrcl} y_1(z) & = & f(z)(1,0,z,z^2+z,z^3,z^3+1,0,z^2)^T & = & z \\ y_2(z) & = & f(z)(0,0,1,0,1,z,z^2,z^2+z)^T & = & 1 \\ y_3(z) & = & f(z)(1,1,1,1,1,1,1,1)^T & = & 1 \\ y_4(z) & = & f(z)(z^3+1,z^3+1,z,z,0,1,z+1,z)^T & = & z \end{array} .$$

Therefore

$$G_4(z) = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ z & z & 1 & 0 & z & z^2+z & z^3 & z^3+1 & 0 & z^2 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & z & z^2 & z^2+z \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ z & z & z^3+1 & z^3+1 & z & z & 0 & 1 & z+1 & z \end{pmatrix}$$

generates a binary self-dual $(10, 5)$ *convolutional code.*

To calculate the degree we again notice that $\delta \leq 8$*, because* 8 *is the sum of the row degrees. From*

$$det\left[\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ z & 0 & z^2+z & z^3 & 0 \\ 1 & 0 & 0 & 1 & z^2 \\ 1 & 1 & 1 & 1 & 1 \\ z & z^3+1 & z & 0 & z+1 \end{pmatrix}\right] = det\left[\begin{pmatrix} 0 & z^2+z & z^3 & 0 \\ 0 & 0 & 1 & z^2 \\ 1 & 1 & 1 & 1 \\ z^3+1 & z & 0 & z+1 \end{pmatrix}\right] = z^8+z^7+z^2,$$

where the last step was shown in the previous example, we find that $\delta \geq 8$ *and conclude* $\delta = 8$*.*

Consequently $G_4(z)$ *generates a binary self-dual* $[10, 5, 8]$ *convolutional code.*

So we have successfully found a binary self-dual $(10, 5)$ convolutional code. Finally, we want to find such a code of degree 1.

Example 5.2.3. *We will try to go from a self-dual* $(4, 2)$ *code to a* $(10, 5)$ *code of degree* 1*.*

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & z+1 & z \end{pmatrix} \in \mathbb{F}_2[z]^{2\times 4}$$

was shown to be a binary self-dual code. Now let $f(z) = (0, 0, 1, 0)$*, then* $f(z)(1, 1, 1, 1)^T = 1$ *as desired and*

$$\begin{array}{rcccl} y_1(z) & = & f(z)(1,1,1,1)^T & = & 1 \\ y_2(z) & = & f(z)(0,1,z+1,z)^T & = & z+1 \end{array}.$$

Therefore

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

generates binary self-dual $(6, 3)$ *code.*

Then let $f(z) = (0, 0, 1, 0, 0, 0)$ *and*

$$\begin{array}{rcccl} y_1(z) & = & f(z)(1,0,0,0,1,0)^T & = & 0 \\ y_2(z) & = & f(z)(1,1,1,1,1,1)^T & = & 1 \\ y_3(z) & = & f(z)(z+1,z+1,0,1,z+1,z)^T & = & 0 \end{array}.$$

Therefore

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

generates a binary self-dual $(8,4)$ *code.*

Finally let $f(z) = (0,0,0,0,1,1,1,0)$*, then*

$$\begin{array}{rcrcl} y_1(z) & = & f(z)(1,0,0,0,1,0,0,0)^T & = & 1 \\ y_2(z) & = & f(z)(0,0,1,0,0,0,1,0)^T & = & 1 \\ y_3(z) & = & f(z)(1,1,1,1,1,1,1,1)^T & = & 1 \\ y_4(z) & = & f(z)(0,0,z+1,z+1,0,1,z+1,z)^T & = & z \end{array}.$$

and therefore

$$G(z) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ z & z & 0 & 0 & z+1 & z+1 & 0 & 1 & z+1 & z \end{pmatrix}$$

generates a binary self-dual $(10,5)$ *convolutional code.*

The row degrees sum up to 1*, so again* $\delta \leq 1$*. Furthermore, the following* 5*-th minor has degree* 1

$$det\left[\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ z & z & 0 & 0 & z \end{pmatrix}\right] = det\left[\begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ z & 0 & 0 & z \end{pmatrix}\right] =$$

$$= det\left[\begin{pmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & z \end{pmatrix}\right] + det\left[\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ z & 0 & z \end{pmatrix}\right] = 0 + z = z,$$

which implies that indeed $\delta = 1$*.*

We conclude that $G(z)$ *generates a binary self-dual* $[10,5,1]$ *convolutional code* $\mathcal{C}$*.*

Let's go one step further and calculate the free distance of $\mathcal{C}$*.*

As a consequence of Lemma 3.3.13, $\mathcal{C}$ *must have an even free distance. Moreover, there exists a row in*

$$(1,0,0,0,0,0,1,1,1,0)$$

that has weight 4*, implying* $d_{free}(\mathcal{C}) \leq 4$*.*

Given the fact that $d_{free}(\mathcal{C})$ *is even and non-zero, we only need to exclude the case of it being equal to* 2*, to conclude* $d_{free}(\mathcal{C}) = 4$

Assume conversely that there exists $c(z) = (c_1(z), \ldots, c_{10}(z)) \in \mathcal{C}$ *of weight* 2*. Then there exists* $m(z) = (m_1(z), m_2(z), m_3(z), m_4(z), m_5(z)) \in \mathbb{F}_2[z]^5$ *such that*

$$c(z) = m(z)G(z).$$

In terms of a system of congruences, this means that (writing m_i instead of $m_i(z)$ and c_i instead of $c_i(z)$ for simplicity)

$$\begin{aligned}
c_1 &= m_1 + m_2 + m_3 + m_4 + zm_5 && (10)\\
c_2 &= m_2 + m_3 + m_4 + zm_5 && (11)\\
c_3 &= m_2 + m_4 && (12)\\
c_4 &= m_4 && (13)\\
c_5 &= m_3 + m_4 + (1+z)m_5 && (14)\\
c_6 &= m_4 + (1+z)m_5 && (15)\\
c_7 &= m_1 + m_2 + m_4 && (16)\\
c_8 &= m_1 + m_4 + m_5 && (17)\\
c_9 &= m_1 + m_3 + m_4 + (1+z)m_5 && (18)\\
c_{10} &= m_4 + zm_5, && (19)
\end{aligned}$$

where at most 2 of the c_i are non-zero. We will argue, that at most 2 of the c_i being non-zero, is only possible whenever $m(z) = (0,0,0,0,0)$, a contradiction.

Assume conversely that $m_5 \neq 0$, then thanks to (15) and (19), $m_4 \neq 0$ must also hold. This follows, because if m_4 was equal to zero as well, we would have $wt(c_{10}) \geq 1$ and $wt(c_6) \geq 2$, contradicting $wt(c(z)) = 2$.
A consequence of $m_4 \neq 0$ is that $c_4 \neq 0$ and $c_{10} \neq 0$, implying that all other c_i must be equal to zero. But (12) implies that

$$m_2 = m_4$$

and (16) plus (17) imply

$$m_2 = m_5.$$

Combining the two, we find $m_4 = m_5$, implying with (19) that $wt(c_{10}) = wt((1+z)m_5) \geq 2$. But also $wt(c_4) = wt(m_4) \geq 1$, a contradiction.

We conclude that $m_5 = 0$.

Similarly, assume $m_4 \neq 0$, then $(13), (15),$ and (19) imply that $m_5 \neq 0$. But we just showed that necessarily $m_5 = 0$.

We conclude that $m_4 = 0$ and update the system:

$$\begin{aligned}
c_1 &= m_1 + m_2 + m_3\\
c_2 &= m_2 + m_3\\
c_3 &= m_2\\
c_4 &= 0\\
c_5 &= m_3\\
c_6 &= 0\\
c_7 &= m_1 + m_2\\
c_8 &= m_1\\
c_9 &= m_1 + m_3\\
c_{10} &= 0
\end{aligned}\quad .$$

By inspection, $m_1 = m_2 = m_3 = 0$ and hence $m(z) = (0,0,0,0,0)$, contradicting the existence of a codeword of weight 2.

Conclusively $G(z)$ *generates a binary self-dual* $[10, 5, 1]$ *convolutional code with its free distance being equal to* 4*.*

6 Conclusion

We started off by finding equivalent conditions to self-duality. This was the foundation for all further research. First, we used it to fully classify all self-dual $(2, 1)$ convolutional codes, all binary self-dual $(4, 2)$ convolutional codes, all self-dual convolutional codes with double diagonal generator matrices and all binary self-dual convolutional codes with double triangular generator matrices.
Then, we dove into the construction of self-dual convolutional codes, where the Building-up construction and the Harada-Munemasa construction were generalized. For the latter, conditions on the $a_i(z)$ for the existence of non-trivial self-dual completions were established and the generalized Harada-Munemasa construction was shown to be able to construct strictly more codes.
Further, we presented a binary self-dual convolutional code with free distance $d_{free} > 2$, which cannot be constructed with the generalized Building-up construction. As a consequence, the generalized Building-up construction is not viable for the full classification of binary self-dual convolutional codes, like it was for block codes. For the generalized Harada-Munemasa construction this question about viability is still open for codes of lengths greater than 4.
Finally, a few examples of binary self-dual convolutional codes of rates ranging from $(4, 2)$ to $(10, 5)$ and differing degrees were presented in section 5.2.

7 Appendix

Theorem 7.0.1. ***(Prime power being the sum of two squares)*** *[13] Let $q = p^l$ be a prime power and $R := \mathbb{F}_q[z]$, then there exist $a, b \in R \setminus \{0\}$ such that*

$$a^2 + b^2 = 0 \; mod \; p$$

if and only if either

$$p = 1 \; mod \; 4$$

or

$$p = 2$$

or

$$l \text{ is even}.$$

Proof. $(\Longleftarrow)$ Let $p = 1 \; mod \; 4$, then -1 is known to be a square mod p and hence a square in R.

Similarly for $p = 2$, $1^2 = 1 = -1$ and therefore -1 is a square mod p, implying it being a square in R.

Finally let l be even, then in terms of Jacobi symbols

$$\left(\frac{-1}{p^l}\right) = \left(\frac{-1}{p}\right)^l = 1$$

and we conclude that -1 is a square in R.

So either way -1 is a square in R. Let $a \in R$ be the square root of -1 in

$$a^2 = -1 \; mod \; p.$$

Then

$$1^2 + a^2 = 1 - 1 = 0 \; mod \; p,$$

proving our claim.

$(\Longrightarrow)$ The proof idea of this direction is closely inspired by J-L. Lagrange.

We know that $a, b \in R$ such that

$$a^2 + b^2 = 0 \; mod \; p$$

exist. In other words, we know that the equation

$$x^2 + y^2 = 0 \tag{20}$$

has a non-trivial solution as an equation over the finite field R. If we look at (20) in terms of a functional equation in the variable x, then

$$D(y) := -4y^2 = (-1)(2y)^2$$

is its discriminant.

Furthermore, (20) has a solution if and only if $D(y)$ is a square over R. We did already show that -1 is a square for any q, except when $p = 3 \mod 4$ and l is odd, implying that for

these q, solutions do exist.

Finally let $p = 3\ mod\ 4$ and l odd, then in terms of Jacobi Symbols

$$\left(\frac{-1}{p^l}\right) = \left(\frac{-1}{p}\right)^l = \left(\frac{-1}{p}\right) = -1,$$

where the last step follows from -1 not being a square mod p, if $p = 3 \mod 4$.

Conclusively $D(y)$ is not a square for these q and hence if there exists a solution to (20), then $p = 3\ mod\ 4$ and l being odd may not hold simultaneously. Or equivalently, then either

$$p = 1\ mod\ 4$$

or

$$p = 2$$

or

$$l \text{ is even.}$$

□

References

[1] T.Kailath. *Linear Systems*, published by Englewood Cliffs, N.J.: Prentice Hall, 1980.

[2] E.V. York. *Algebraic description and construction of error correcting codes: A linear systems point of view*, Ph.D. dissertation, University of Notre Dame, 1997.

[3] B.M. Allen. *Linear system analysis and decoding of convolutional code*, Ph.D. dissertation, University of Notre Dame, Indiana, 1999.

[4] J. Lieb, R. Pinto, J. Rosenthal. *Convolutional codes.* To be published in "A concise encyclopedia of coding theory" (eds. Huffman, C; Kim, J.; Sole, P.), CRC Press, 2020, arXiv:2001.08281.

[5] H.S. Palines and V.P. Sison. *On self-dual convolutional codes over rings*, East-West Journal of Mathematics Vol. 14 (2012), No. 1, pp. 28-36.

[6] P. Gaborit, A. Otmani. *Experimental construction of self-dual codes*, Finite Fields and Their Applications Vol. 9 (2003), pp. 372-394.

[7] M. Harada and A. Munemasa. *Classification of self-dual codes of length 36*, Advances in Mathematics of Communications Vol. 6 (2012), No. 2, pp. 229-235.

[8] J.-L. Kim. *New extremal self-dual codes of length 36, 38 and 58*, IEEE Transactions on Information Theory Vol. 47 (2001), No. 1, pp. 386-393.

[9] C.A. Melchor, P. Gaborit, J.-L. Kim, L. Sok and P. Solé. *Classification of extremal and s-extremal binary self-dual codes of length 36*, IEEE Transactions on Information Theory 58 (2012), No. 4, pp. 2253-2262.

[10] S.H. Friedberg, A.J. Insel and L.E. Spence. *Linear Algebra*, published by Springler, 2003.

[11] N.J. Russell. *Self-dual codes and indecomposable building blocks*, Master Thesis, Eastern Kentucky University, Kentucky, 2016.

[12] W.C. Hufmann and V. Pless. *Fundamentals of error-correcting codes*, published by Cambridge University Press, 2003.

[13] C.F. Gauss. *Disquisitiones Arithmeticae (English Edition)*, Transl. by Arthur A. Clarke, published by Springer, 1986.

[14] E.M. Rains and N.J.A Sloane. *Self-dual codes*, in: V.S. Pless and W.C. Huffman, Handbook of Coding Theory, publishhed by Elsevier, 1998, pp. 177–294..

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

[16] J.H. Conway and V. Pless. *On the enumeration of self-dual codes*, Combin. Theory Vol. 28 (1980), Ser. A, pp. 26-53.