Self-Dual Convolutional Codes

Sebastian Heri, Julia Lieb, and Joachim Rosenthal, Fellow, IEEE

Abstract—This paper investigates the concept of self-dual convolutional codes. We derive the basic properties of this interesting class of codes and we show how some of the techniques to construct self-dual linear block codes generalize to self-dual convolutional codes. As for self-dual linear block codes we are able to give a complete classification for some small parameters.

Index Terms—Convolutional codes, self-dual codes, classification of codes, construction of codes.

I. INTRODUCTION

A \((n,k)\) linear block code \(C\) is by definition a linear subspace \(C \subset F_q^n\), where \(F_q\) is a finite field and \(\dim_q C = k\). By considering the natural bilinear form \(\langle , \rangle\) on the vector space \(F_q^n\) one obtains the notion of the dual code

\[
C^\perp = \{ x \in F_q^n \mid \langle x, c \rangle = 0 \ \forall c \in C \}.
\]

Clearly \(C^\perp\) is an \((n,n-k)\) linear block code. Note also that the bilinear form \(\langle , \rangle\) is in general not positive definite and when \(n = 2k\) it can happen that \(C^\perp = C\). A code having the particular property that \(C^\perp = C\) is called a self-dual linear block code. Self-dual block codes are a highly interesting class of linear block codes. They led to the construction of codes with good parameters and also some MDS block codes have been achieved via self-dual codes, see e.g. [6]. Moreover, self-dual codes are important for the construction of quantum codes, see e.g. [2]. There have been large efforts in the classification of self-dual codes. In [3] and [17] a complete classification of binary self-dual block codes up to length 30 is provided. This is extended in [7] to a complete classification up to length 36. Moreover, in [4], [7], and [12], different techniques how to construct new self-dual block codes from known self-dual block codes are provided. In this paper, we will generalize all of these techniques to obtain new self-dual convolutional codes from known self-dual convolutional codes. In general there still exist many interesting and open questions related to self-dual block codes. For more information regarding this the reader is referred to the survey articles [1], [18].

In this paper we introduce the concept of a self-dual convolutional code. One can view a convolutional code \(C\) as a submodule \(C \subset F_q[x]^n\), where \(F_q[x]\) is the polynomial ring over \(F_q\). A natural bilinear form \(\langle , \rangle\) on \(F_q[x]^n\) can be defined and this induces the notion of duality in convolutional codes as well. In [9], examples of particular self-dual convolutional codes are provided, however, using another definition of self-duality than the one presented in our paper. In [16], an algorithm to find systematic self-dual convolutional codes is presented. However, theoretical approaches concerning properties and classifications of this kind of codes are still lacking and are the objectives of our paper.

The paper is structured as follows. In Section II, we present the basics of self-dual block codes and their construction from smaller self-dual codes. In Section III, we introduce convolutional codes. In Section IV, we present equivalent conditions for convolutional codes to be self-dual, as well as some properties of self-dual convolutional codes. In Section V, we classify all self-dual convolutional codes with certain code parameters. In Section V-A, we classify all self-dual \((2,1)\) convolutional codes, in Section V-B all binary self-dual \((4,2)\) convolutional codes, in Section V-C all self-dual convolutional codes with double diagonal generator matrices and in Section V-D all binary self-dual convolutional codes with double triangular generator matrices. In Section VI, we generalize the building-up construction and the Harada-Munemasa construction from block to convolutional codes, where the building-up construction is further generalized to arbitrary finite fields. Moreover, we prove that every binary code obtained with the generalized building-up construction can also be obtained with the generalized Harada-Munemasa construction, but the opposite is not true. Furthermore, we show that not all binary self-dual convolutional codes with free distance \(d_{free} > 2\) can be constructed with the generalized building-up construction, as it was shown to be true for block codes. We also show that all binary self-dual \((4,2)\) convolutional codes can be constructed with the generalized Harada-Munemasa construction, but for general code parameters this question remains open.

II. SELF-DUAL LINEAR BLOCK CODES

In this section, we present some results on self-dual linear block codes, that we will generalize later to obtain results for self-dual convolutional codes.

A. Preliminaries

Definition 1: An \((n,k)\) (linear) block code \(C\) over \(F_q\) is a \(k\)-dimensional subspace \(C \subset F_q^n\), where \(k \leq n\). A matrix
$G \in \mathbb{F}_q^{k \times n}$ such that $C = \text{rowspan}(G)$ is called a generator matrix of $C$.

**Definition 2:** Two generator matrices $G_1$ and $G_2$ are equivalent if there exists an invertible matrix $A \in GL_k(\mathbb{F}_q)$ such that

$$G_1 = AG_2.$$ 

In other words, two generator matrices are equivalent if they generate the same code.

**Definition 3:** Let $C$ be a $(n, k)$ linear block code. Then there exists $H \in \mathbb{F}_q^{(n-k)\times n}$ such that

$$C = \ker(H) = \{v \in \mathbb{F}_q^n \mid Hv^\top = 0\}.$$ 

We call $H$ a parity-check matrix of $C$.

Like for generator matrices, also a parity-check matrix of a code is in general not unique and elementary row operations on either of these matrices leave the code unchanged.

**Definition 4:** For a code $C \subset \mathbb{F}_q^n$, $C^\perp = \{x \in \mathbb{F}_q^n \mid xc^\top = 0 \ \forall c \in C\}$ is called the dual of $C$. If the condition $xc^\top = 0$ holds, then $x$ and $c$ are said to be orthogonal.

The dual of a linear block code is always a linear block code.

**Definition 5:** We say that a code $C$ is self-orthogonal if $C \subset C^\perp$ and self-dual if $C = C^\perp$.

Thus, 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.

The following lemma is an immediate consequence of the definition of the dual code.

**Lemma 6 [8]:** Let $G$ be a generator matrix of $C$. Then

$$C^\perp = \ker(G)$$

i.e., $G$ is a parity-check matrix of $C^\perp$.

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 $C = C^\perp = \ker(G)$.

**Corollary 1 [8]:** Any self-dual $(n, k)$ code fulfills $n = 2k$.

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

**Lemma 7 [14, Chapter 1.8]:** Let $C$ be a $(2k, k)$ block code over $\mathbb{F}_q$ with generator matrix $G$. Then the following statements are equivalent:

(i) $C$ is self-dual;
(ii) $C$ is self-orthogonal;
(iii) $GG^\top = 0$;
(iv) $G$ is a parity-check matrix of $C$.

**Definition 8:** 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

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

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

**Definition 9:** The (minimum) distance $d$ of a linear block code $C$ is given by

$$d = d(C) = \min \{wt(c) \mid c \in C \setminus \{0\}\},$$

i.e. for linear codes the minimum Hamming distance is equal to the minimum Hamming weight. One uses the notation $[n, k, d]$ code for an $(n, k)$ block code with distance $d$.

### B. Construction Methods

There are three popular construction methods for binary self-dual codes, namely the building-up construction, the Harada-Munemasa construction and the recursive construction. The latter will not be presented, but interested readers may consult [15].

**Theorem 10 (Building-Up Construction [12]):** Let $C$ be a binary self-dual $(2k, k)$ code and $G = (g_i)_{i \in \{1, \ldots, k\}}$ 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 $y_i := xg_i^\top$ for $1 \leq i \leq k$. Then

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

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

**Theorem 11 [12]:** Any binary self-dual code of length $n$ with distance $d > 2$ can be obtained from some binary self-dual code of length $n - 2$ with the construction in Theorem 10 (up to column permutations in the generator matrix).

As second construction method we present the Harada-Munemasa construction.

**Theorem 12 (Harada-Munemasa Construction [7]):** Let $G$ be the generator matrix of a binary self-dual $[2k, k, d]$ code $C$. Then the matrix

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

where $a_i \in \mathbb{F}_2$, generates a binary self-orthogonal $[2k + 2, k, d']$ code $C_1$ with $d' \geq d$. Moreover, there exists $x = (x_1 \cdots x_{k+1}) \in C_1^\perp \setminus C_1$ such that

$$G_2 = \begin{pmatrix} x_1 & \cdots & x_{k+1} \\ a_1 & a_1 \\ \vdots & \vdots \\ a_k & a_k \end{pmatrix} \in \mathbb{F}_2^{(k+1)\times(2k+2)}$$

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

**Theorem 13:** 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 12 (up to column permutations in the generator matrix).

The 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.
III. CONVOLUTIONAL CODES

In this section, we first introduce some basics about convolutional codes with a particular focus on non-catastrophic convolutional codes. Then, we will connect non-catastrophic codes to self-dual codes and present equivalent properties to self-duality, which will be used for classifications and constructions in later sections.

A. Preliminaries

Definition 14: An \((n, k)\) convolutional code is a \(\mathbb{F}_q[z]\)-submodule of \(\mathbb{F}_q[z]^n\) with rank \(k\).

\(\mathbb{F}_q[z]\) is a PID and modules over a PID always admit a basis. Thus, one always finds a full rank matrix \(G(z) \in \mathbb{F}_q[z]^{k \times n}\) such that
\[
C = \text{rowspan}_{\mathbb{F}_q[z]}(G(z)) = \{c(z) = m(z)G(z) \text{ for some } m(z) \in \mathbb{F}_q[z]^k\}.
\]

This \(G(z)\) is called a generator matrix of \(C\).

Definition 15: Let \(H(z) \in \mathbb{F}_q[z]^{(n-k) \times n}\) such that
\[C = \ker(H(z)).\]

Then we call \(H(z)\) a parity-check matrix of \(C\).

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 Section III-B.

Definition 16: We say that \(U(z) \in \mathbb{F}_q[z]^{k \times k}\) is unimodular if there exists \(V(z) \in \mathbb{F}_q[z]^{k \times k}\) such that
\[V(z)U(z) = I_k.\]

Definition 17: Two generator matrices \(G(z) \in \mathbb{F}_q[z]^{k \times n}\) and \(\tilde{G}(z) \in \mathbb{F}_q[z]^{k \times n}\) are equivalent if there exists a unimodular matrix \(U(z) \in \mathbb{F}_q[z]^{k \times k}\) such that
\[G(z) = U(z)\tilde{G}(z).\]

Equivalent generator matrices generate the same code.

Definition 18: We say that a row or column operation is unimodular if it is done via multiplication with a unimodular matrix.

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

Definition 19: The degree \(\delta\) 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 multiplication with a unimodular matrix, which has a constant determinant, the degree \(\delta\) is the same for all equivalent generator matrices, making it a well-defined notion. We say that \(C\) is an \((n, k, \delta)\) convolutional code if \(C\) has length \(n\), dimension \(k\) and degree \(\delta\). Moreover, the rate of the code is defined as \(k/n\).

Remark 1: The convolutional codes of degree 0 are essentially the linear block codes.

Definition 20 [10]: Let \(C\) be a convolutional code with generator matrix \(G(z) = (g_j(z))_{1 \leq i \leq k, 1 \leq j \leq n}\).

Then \(\nu_i := \max \{\deg(g_j(z)) \mid j \in \{1, \ldots, n\}\}\) is called the \(i\)-th row degree or \(i\)-th constraint length of \(G(z)\).

Moreover, \(\mu = \max_{i=1, \ldots, k} \nu_i\) is called the memory and \(\nu = \sum_{i=1}^{k} \nu_i\) is called the overall constraint length of \(G(z)\).

Remark 2: Let \(\delta\) be the degree and \(\nu\) the overall constraint length of some generator matrix of a convolutional code, then \(\delta \leq \nu\). The row degrees of two equivalent matrices may be different, but a generator matrix fulfilling \(\delta = \nu\) can always be found. Such a generator matrix is called a minimal generator matrix.

Definition 21: 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
\[wt(c(z)) := \sum_{i=0}^{\deg(c(z))} wt(c_i),\]
where \(wt(c_i)\) denotes the Hamming weight of \(c_i\), i.e. the weight of a polynomial vector is the sum of the weights of its coefficients.

Definition 22: The free distance of a convolutional code \(C\) is defined as
\[d_{free}(C) = \min \{wt(c(z)) \mid c(z) \in C \setminus \{0\}\}.\]

B. Non-Catastrophic Codes

Definition 23: A polynomial matrix \(G(z) \in \mathbb{F}_q[z]^{k \times n}\) is said to be left-prime if in all factorizations
\[G(z) = A(z)\tilde{G}(z), \text{ with } A(z) \in \mathbb{F}_q[z]^{k \times k}, \tilde{G}(z) \in \mathbb{F}_q[z]^{k \times n},\]
the left factor \(A(z)\) is unimodular.

If one generator matrix is left-prime, then all generator matrices of the same convolutional code are also left-prime as they differ by left-multiplication with a unimodular matrix.

Definition 24: We call codes that admit a left-prime generator matrix to be non-catastrophic and catastrophic otherwise.

Remark 3: The notion non-catastrophic convolutional code has its origin in the fact that convolutional codes with left prime generator matrices avoid so-called catastrophic situations where a small number of transmission errors can cause an arbitrarily large number of decoding errors. Take e.g. the catastrophic (2,1,1) convolutional code with generator matrix
\[G(z) = [1 - z \quad 1 - z] = (1 - z)[1 \quad 1].\]

To send the message \(m(z) = \sum_{i=0}^{n} z^i\), encode it to the codeword
\[c(z) = m(z)G(z) = [1 - z^{n+1} \quad 1 - z^{n+1}].\]

If due to (four) transmission errors in the channel, the vector [0 0] is received instead of \([1 - z^{n+1} \quad 1 - z^{n+1}]\), one would decode it to 0, which means one has \(n+1\) decoding errors.

The module approach relates naturally to linear systems theory and is in some way also simpler. At the same time (see Corollary 3) one readily also gets results for convolutional codes defined over formal Laurent series.

Remark 4: We defined convolutional codes as modules over the ring of polynomials, i.e. messages are polynomial vectors,
and as explained above, with this definition of convolutional code, being catastrophic or non-catastrophic is a property of the code itself. Earlier works on convolutional codes, such as [10], define convolutional codes as vector spaces over the field of rational functions \( \mathbb{F}_q(z) \) or the field of formal Laurent series \( \mathbb{F}_q((z)) \). Also with these definitions, each convolutional code possesses polynomial generator matrices, but always some of its polynomial generator matrices are left-prime and some are not left-prime. Hence, in these scenarios, it makes no sense to speak of non-catastrophic convolutional codes but rather to speak of non-catastrophic generator matrices. With these definitions, we could choose \( G(z) = [1 \ 1] \) as non-catastrophic generator matrix in the above example.

We will see more practical methods to check whether codes are non-catastrophic in Theorem 28, but first we introduce some necessary notions.

Definition 25 [5, Section 2.1]: Let \( G(z) \in \mathbb{F}_q[z]^{k \times n} \) with \( k \leq n \). Then, there exists a unimodular matrix \( U(z) \in \mathbb{F}_q[z]^{k \times k} \) such that

\[
G_{rH}(z) = U(z)G(z) = \begin{pmatrix}
h_{11}(z) & h_{12}(z) & \cdots & h_{1k}(z) & h_{1k+1}(z) & \cdots & h_{12k}(z) \\
h_{21}(z) & h_{22}(z) & \cdots & h_{2k}(z) & h_{2k+1}(z) & \cdots & h_{22k}(z) \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
h_{k1}(z) & h_{k2}(z) & \cdots & h_{kk}(z) & 0 & \cdots & 0
\end{pmatrix}
\]

where \( (h_{ij}(z))_{i=1}^k \) are monic polynomials such that \( \deg(h_{i1}(z)) > \deg(h_{ij}(z)) \) for \( i > j \) if \( h_{ij}(z) \) is not identically equal to zero. The matrix \( G_{rH}(z) \) is unique and is called the row Hermite form of \( G(z) \).

The uniqueness of the row Hermite form gives us the option to characterize equivalent generator matrices by their respective row Hermite forms.

Definition 26: Let \( G(z) \in \mathbb{F}_q[z]^{k \times n} \) with \( k \leq n \). Then there exists a unimodular matrix \( V(z) \in \mathbb{F}_q[z]^{n \times n} \) such that

\[
G_{cH}(z) = G(z)V(z)
\]

\[
= \begin{pmatrix}
h_{11}(z) & 0 & \cdots & 0 \\
h_{21}(z) & h_{22}(z) & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \ddots & \vdots \\
h_{k1}(z) & h_{k2}(z) & \cdots & h_{kk}(z) & 0 & \cdots & 0
\end{pmatrix}
\]

where \( (h_{ii}(z))_{i=1}^k \) are monic polynomials such that \( \deg(h_{i1}(z)) > \deg(h_{ij}(z)) \) for \( i > j \) if \( h_{ij}(z) \) is not identically equal to zero. \( G_{cH}(z) \) is called the (unique) column Hermite form of \( G(z) \).

Definition 27: Let \( G(z) \in \mathbb{F}_q[z]^{k \times n} \) be full rank with \( k \leq n \). Then there exist two unimodular matrices \( U(z) \in \mathbb{F}_q[z]^{k \times k} \) and \( V(z) \in \mathbb{F}_q[z]^{n \times n} \) such that

\[
S(z) = U(z)G(z)V(z)
\]

\[
= \begin{pmatrix}
\gamma_1(z) & 0 & \cdots & 0 \\
\gamma_2(z) & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
\gamma_k(z) & 0 & \cdots & 0
\end{pmatrix}
\]

where \( (\gamma_i(z))_{i=1}^k \) are monic polynomials such that \( \gamma_1(z) | \gamma_i(z) \) for all \( i = 1, \ldots, k - 1 \). \( S(z) \) is called the (unique) Smith form of \( G(z) \).

The following result will be crucial in later sections to determine whether a code is non-catastrophic.

Theorem 28 [11], [21]: Let \( C \) be an \( (n, k) \) convolutional code and \( G(z) \in \mathbb{F}_q[z]^{k \times n} \) its generator matrix. The following are equivalent:

1. \( C \) is non-catastrophic;
2. \( G(z) \) is left-prime;
3. the Smith form of \( G(z) \) is \([I_k \ 0]\);
4. the column Hermite form of \( G(z) \) is \([I_k \ 0]\);
5. the ideal generated by all the \( k \)-th minors of \( G(z) \) is \( \mathbb{F}_q[z] \);
6. there exists a parity-check matrix \( H(z) \) of \( C \);
7. \( G(z) \) can be completed to an unimodular matrix, e.g. there exists \( L(z) \in \mathbb{F}_q[z]^{(n-k) \times n} \) such that \( [G(z) \ L(z)] \) is unimodular.

Corollary 2 Let \( G(z) \) be the generator matrix of a non-catastrophic \((n, k)\) convolutional code. Then, for any unimodular \( U(z) \in \mathbb{F}_q[z]^{k \times k} \) and \( V(z) \in \mathbb{F}_q[z]^{n \times n} \),

\[
U(z)G(z)V(z)
\]

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 change when multiplying it 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 28.

Definition 29 For two matrices \( G_1(z), G_2(z) \in \mathbb{F}_q[z]^{k \times n} \), we write \( G_1(z) \sim G_2(z) \), if there exist unimodular matrices \( A(z) \in \mathbb{F}_q[z]^{k \times k} \) and \( B(z) \in \mathbb{F}_q[z]^{n \times n} \) such that \( A(z)G_1(z)B(z) \).

IV. SELF-DUAL CONVOLUTIONAL CODES

In this section, we present the basic theory of self-dual convolutional codes.

Definition 30: Let \( C \) be an \((n, k)\) convolutional code. Then,

\[ C^\perp = \{ f(z) \in \mathbb{F}_q[z]^n \mid f(z)c(z)^\top = 0 \ \forall c(z) \in C \} \]

is called the dual of \( C \).

Lemma 31 [16]: The dual of an \((n, k)\) convolutional code \( C \) is an \((n, n-k)\) convolutional code.

Definition 32: Two vectors \( u(z), v(z) \in \mathbb{F}_q[z]^n \) are said to be orthogonal if

\[ u(z)v(z)^\top = 0. \]

A. Characterization of Self-Dual Convolutional Codes Over Arbitrary Finite Fields

The following lemma establishes that the dual code always has a parity-check matrix.

Lemma 33: Let

\[
G(z) = \begin{pmatrix}
g_1(z) \\
\vdots \\
g_k(z)
\end{pmatrix}
\]
be the generator matrix of a convolutional code $C$, then $C^\perp = \ker(G(z))$ or equivalently $G(z)$ is a parity-check matrix of $C^\perp$.

**Proof:** To show $C^\perp \subset \ker(G(z))$, let $c(z) \in C^\perp$. One has $g_i(z)c(z)^\top = 0$ for $i = 1, \ldots, k$ and therefore $G(z)c(z)^\top = 0$, i.e., $c(z) \in \ker(G(z))$.

To show $\ker(G(z)) \subset C^\perp$, let $f(z) \in \ker(G(z))$ and $c(z) \in C$. Then, $c(z) = m(z)G(z)$ for some $m(z) \in \mathbb{F}_q[z]^k$. But now

$$c(z)f(z)^\top = (m(z)G(z))f(z)^\top = m(z)(G(z)f(z)^\top) = 0$$

as $f(z)$ is in the kernel of $G(z)$. We conclude that $f(z)$ is orthogonal to every codeword of $C$ or equivalently $f(z) \in C^\perp$.

Thus, any generator matrix of a convolutional code is a parity-check matrix of the dual code. Consequently, the dual is always non-catastrophic by Theorem 28.

**Lemma 34:** Let $C$ be an $(n, k)$ convolutional code with generator matrix $G(z)$. Then, $C$ is non-catastrophic if and only if $C = (C^\perp)^\perp$.

**Proof:** $(\Rightarrow)$ Let $c(z) \in C$ and $c_i(z) \in C^\perp$, then by definition $c_1(z)c(z)^\top = 0$ and hence $C \subset (C^\perp)^\perp$. As $(C^\perp)^\perp$ is again a convolutional code, it has a generator matrix $G(z)$ with $\text{rowspan}(G(z)) \subset \text{rowspan}(G(z))$. Moreover, $\dim((C^\perp)^\perp) = n - \dim(C^\perp) = \dim(C) = k$ and one obtains $U(z)G(z) = G(z)$ for some $U(z) \in \mathbb{F}_q[z]^{k \times k}$. But $G(z)$ is left-prime, because we assumed that $C$ is non-catastrophic, and hence the left-factor $U(z)$ is unimodular. We conclude that $G(z)$ and $G(z)$ are equivalent and $C = (C^\perp)^\perp$.

$(\Leftarrow)$ Follows directly from the previous lemma as $C = (C^\perp)^\perp$ is the dual of $C^\perp$.

**Definition 35:** We say that a convolutional code $C$ is self-orthogonal if $C \subset C^\perp$ and self-dual if $C = C^\perp$.

Note that the term convolutional self-orthogonal code (CSOC) is also used with a different meaning in the literature; see e.g. [19]. Moreover, in [9], a convolutional code is called self-dual when $GG^\top = 0$ where $G = \begin{pmatrix} G_0 & \cdots & G_\mu \\ 0 & G_0 & \cdots & G_\mu \\ \vdots & \ddots & \ddots & \ddots \end{pmatrix}$ is the semi-infinite sliding generator matrix associated to the generator matrix $G(z) = \sum_{i=0}^\mu G_iz^i \in \mathbb{F}_q[z]^{k \times n}$. This differs from our definition. However, we chose the definition of duality for convolutional codes in a way such that one obtains natural generalizations of the theory of dual and self-dual block codes. The following lemma is immediate.

**Lemma 36:** Let $C$ be a self-dual $(n, k)$ convolutional code with generator matrix $G(z)$. Then, $C$ is non-catastrophic.

Like for block codes, we may assume every self-dual convolutional code to be a $(2k, k)$ or equivalently an $(n, n/2)$ code as stated in the following lemma.

**Lemma 37** [16]: If $C$ is a self-dual $(n, k)$ convolutional code, then $n = 2k$.

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

**Theorem 38:** Let $C$ be a $(2k, k)$ convolutional code with generator matrix $G(z)$, then the following statements are equivalent:

(i) $C$ is self-dual;

(ii) $C$ is non-catastrophic and self-orthogonal;

(iii) $C$ is non-catastrophic and $G(z)G(z)^\top = 0$;

(iv) $G(z)$ is a parity-check matrix of $C$.

**Proof:** (i) $\Rightarrow$ (ii) Let $C$ be self-dual, then the code is self-orthogonal by definition and non-catastrophic by Lemma 26.

(ii) $\Rightarrow$ (iii) Let $C$ be non-catastrophic and self-orthogonal and let $H(z)$ be a generator matrix of $C^\perp$. $C \subset C^\perp$ implies $\text{rowspan}(G(z)) \subset \text{rowspan}(H(z))$ and hence $U(z)H(z) = G(z)$ for some $U(z) \in \mathbb{F}_q[z]^{k \times k}$. We assumed $C$ to be non-catastrophic, i.e., $G(z)$ is left-prime and hence, $U(z)$ must be unimodular. But this means that $G(z)$ and $H(z)$ are generator matrices of the same code and $C = C^\perp$.

(iii) $\Rightarrow$ (i) Let $C$ be self-orthogonal with generator matrix $G(z)$. Each pair of rows $g_i(z), g_j(z)$ of $G(z)$ is orthogonal, i.e., $g_i(z)g_j(z)^\top = 0$ and $G(z)G(z)^\top = 0$ follows.

(iv) $\Rightarrow$ (ii) Let $G(z)$ be a parity-check of $C$, then $C$ is non-catastrophic and $G(z)G(z)^\top = 0$, because $\text{rowspan}(G(z)) = \ker(G(z))$.

One might observe that self-orthogonality and $G(z)G(z)^\top = 0$ 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 are equivalent properties, which is due to the fact that each linear block code possesses a parity-check matrix. For $(2k, k)$ convolutional codes however, this is not necessarily true as illustrated in the following example.

**Example 1:** Let $G(z) = \begin{pmatrix} z^2 + z + 1 & z^2 & z & 1 \\ 1 & z^2 & z^2 & z^2 + z + 1 \end{pmatrix} \in \mathbb{F}_2[z]^{2 \times 4}$ be the generator matrix of a binary $(4, 2)$ convolutional code $C$. We want to show that $C = \text{rowspan}(G(z))$ is a proper subset of $C^\perp = \ker(G(z))$. As $G(z)G(z)^\top = 0$, $C \subset C^\perp$. Moreover, we observe that $(1, 1, 1, 1) \in C^\perp$ but $(1, 1, 1, 1) \notin C$, which shows that $C$ is indeed a proper subset of $C^\perp$. Conclusively, $C$ is a self-orthogonal $(2k, k)$ convolutional codes that is not self-dual. As a consequence of Theorem 38, $C$ must be a catastrophic code.
Corollary 3: If \( C \) is a subspace of \( F_q(z)^n \) or \( F_q((z))^n \), then \( C = C^\perp \) if and only of \( G(z)G(z)^T = 0 \) for any of the (rational) generator matrices of \( C \), i.e. \( C \subseteq C^\perp \) already implies \( C = C^\perp \).

Proof: To show that self-orthogonality implies self-duality when working with subspaces of \( F_q(z)^n \) or \( F_q((z))^n \), we can proceed as in part (ii) \( \Rightarrow \) (i) of the proof of the preceding theorem and use that \( U(z)H(z) = G(z) \) for \( H(z) \), \( G(z) \in F_q(z)^{k \times k} \) and some \( U(z) \in F_q(z)^{n \times k} \) already implies that \( H(z) \) and \( G(z) \) generate the same convolutional code since that \( G(z) \) is full rank implies \( \det(U(z)) \neq 0 \).

B. Special Properties of Binary Self-Dual Convolutional Codes

In this subsection, we present some further results for binary self-dual convolutional codes.

The following well-known lemma is an immediate consequence of the “Freshman dream” and helps us to determine whether a binary polynomial vector is orthogonal to itself.

**Lemma 39:** Let \( f_i(z) \in F_2[z] \), for \( i \in \{1, \ldots, m\} \), and \( f(z) = (f_1(z), \ldots, f_m(z)) \), then

\[
f(z)f(z)^T = (f(z)(1, \ldots, 1)^T)^2.
\]

As a consequence, to determine whether a binary polynomial vector is orthogonal to itself, we may just check if the sum of its entries is equal to zero.

**Lemma 40:** 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 \( c_i \in F_2 \). By definition of self-duality, one obtains \( c(z)c(z)^T = 0 \). It follows from Lemma 39 that \( c(z)(1, \ldots, 1)^T = 0 \) 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, i.e. \( c_i(1, \ldots, 1)^T = 0 \) 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 [20], the following result was proven for block codes. Up to our knowledge it has not been proven for convolutional codes yet.

**Lemma 41:** Every binary self-dual convolutional code contains \((1, \ldots, 1) \in F_2^n \).

**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 39), \( g_i(z)(1, \ldots, 1)^T = 0 \) and therefore \( G(z)(1, \ldots, 1)^T = 0 \). Hence \((1, \ldots, 1)\) is contained in the dual of the code and, because of self-duality, contained in the code itself.

**Corollary 4:** Let \( C \) be a binary self-dual \((n,k)\) convolutional code, then there exists a generator matrix whose first row is the all one vector.

This generator matrix with just ones in the first row is equivalent to its row Hermite form. Therefore, to find all binary self-dual convolutional codes, we may characterize them via generator matrices that have a first row of just ones and are in row Hermite form.

**Remark 5:** Let

\[
G(z) = \begin{pmatrix}
1 & 1 & \cdots & \cdots & 1 \\
0 & g_{22}(z) & g_{23}(z) & \cdots & g_{2n}(z) \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
0 & \cdots & 0 & g_{kk}(z) & g_{kn}(z)
\end{pmatrix}
\]

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 a unimodular matrix from the left-hand side leaves the code unchanged and therefore leaves the code self-dual. Obviously, exchanging columns has also no influence on self-duality since it only results in renumbering the components of the codewords. However, the following example shows that multiplying a column with a constant or adding columns does not preserve self-duality.

**Example 2:** We claim that

\[
G(z) = \begin{pmatrix}
3 & z & 1 & 3z \\
1 & 2z + 4 & 2 & z + 2
\end{pmatrix} \in F_5[z]^{2 \times 4}
\]

generates a self-dual code and will use Theorem 38 to prove this. Firstly, \( G(z)G(z)^T = 0 \) and secondly

\[
det \begin{pmatrix}
3 & z \\
1 & 2z + 4
\end{pmatrix} = 2,
\]

which implies that the set of \( G(z) \)'s fullsize minors generates \( F_5[z] \) or equivalently \( G(z) \) generates a non-catastrophic code. Hence, the code is self-dual. We now want to establish that adding and multiplying columns might change this property. For this, add the second column of \( G(z) \) to the first to find

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

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

\[
(3 + z, z, 1, 3z)(3 + z, z, 1, 3z)^T = z^2 + z \neq 0,
\]

meaning that \( G_1(z) \) does not generate a self-dual code. 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

\[
(1, z, 1, 3z)(1, z, 1, 3z)^T = 2 \neq 0.
\]
V. CLASSIFICATION OF SELF-DUAL CONVOLUTIONAL CODES

In this section, we will classify 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.

We denote $F_q^* = F_q \setminus \{0\}$ and say that $r \in F_q$ is a square in $F_q$ if $r = s^2$ for some $s \in F_q$.

A. Self-Dual (2,1) Convolutional Codes

In this subsection, we consider $F_q[z]$, where $q = p^l$ is a prime power.

**Theorem 42:** The self-dual (2,1) convolutional codes are exactly the (2,1) self-dual block codes. In particular, these are exactly the codes with a generator matrix of the form $C = \begin{pmatrix} a & b \end{pmatrix}$ where $a^2 = -b^2$ in $F_q$. Clearly, such a generator matrix exists if and only if $-1$ is a square in $F_q$, i.e., for $p \equiv 1 \pmod{4}$ or $p = 2$ or $l \equiv 0 \mod{2}$ even.

**Proof:** Denote by $G(z) = (g_1(z), g_2(z)) \in F_q[z]^{1 \times 2}$ a generator matrix of a self-dual (2,1) convolutional code $C$. By Theorem 38, the code is self-dual if and only if $g_1(z)^2 + g_2(z)^2 = 0$ and it is non-catastrophic. However, such a code is non-catastrophic if and only if $g_1(z)$ and $g_2(z)$ are coprime, which can only be fulfilled together with $g_1(z)^2 = -g_2(z)^2$ if $g_1(z)$ and $g_2(z)$ are constant. □

B. Binary Self-Dual (4,2) Convolutional Codes

The goal in this section is to find all binary self-dual (4,2) convolutional codes $C$ classified by their representative as defined in Remark 5, i.e., we may assume that

$$G(z) = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & g_{22}(z) & g_{23}(z) & g_{24}(z) \end{pmatrix} \in F_2[z]^{2 \times 4}$$

is a generator matrix of $C$ for some $g_{22}(z), g_{23}(z), g_{24}(z) \in F_2[z]$. $G(z)G(z)^\top = 0$ tells us that $g_{22}(z)^2 + g_{23}(z)^2 + g_{24}(z)^2 = 0$ or equivalently (by Lemma 39),

$$g_{22}(z) + g_{23}(z) + g_{24}(z) = 0. \quad (1)$$

Moreover, $C$ is non-catastrophic if and only if the elements of the set of 2-nd minors $M = \{g_{22}(z), g_{23}(z), g_{24}(z), g_{22}(z) + g_{23}(z), g_{22}(z) + g_{24}(z), g_{23}(z) + g_{24}(z)\}$ of $G(z)$ are coprime, i.e., $gcd(g_{22}(z), g_{23}(z), g_{24}(z)) = 1$. Using (1) this simplifies to $gcd(g_{23}(z), g_{24}(z)) = 1$.

We conclude that for $g_{23}(z), g_{24}(z) \in F_2[z]$ 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.

**Example 3:**

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

is the generator matrix of a binary (4,2,1) self-dual convolutional code $C$. Since

$$d_{free}(C) \geq min_{m_0 \neq 0} wt \left( \begin{pmatrix} m_0 & m_1 \\ G_0 & G_1 \end{pmatrix} \right) = 4$$

and the first row of $G(z)$ is a codeword of weight 4, we obtain $d_{free}(C) = 4$, i.e., $C$ is MDS.

C. Double Diagonal Generator Matrix

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

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

Let $i \in \{1, \ldots, k\}$. Self-orthogonality is equivalent to $g_{i,i}(z)^2 + g_{i,k+i}(z)^2 = 0$ and non-catastrophicity is equivalent to $gcd(g_{i,i}(z), g_{i,k+i}(z)) = 1$. As in Theorem 42, one concludes that

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

where $a_i, b_i \in \mathbb{F}_q^*$ such that $a_i^2 = -b_i^2$ in $\mathbb{F}_q$. Therefore, self-dual convolutional codes of this form exist for exactly the same field sizes as (2,1) self-dual block codes.

D. Double Upper Triangular Generator Matrices

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

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

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

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

is also a generator matrix of $C$. Let $g_i$ be the $i$-th row of $G(z)$, then $g_k g_i = 0$ for $i \in \{1, \ldots, k\}$ leads to

$$g_{k,k} = g_{k,2k},$$

$$g_{k,k} g_{i,k} + g_{k,2k} g_{i,2k} = g_{k,k}(g_{i,k} + g_{i,2k}) = 0.$$
Moreover, \( g_{k,k} \) is non-zero (as \( \text{rank}(G(z)) = k \)) and hence \( g_{1,k} = g_{1,2k-1} \). Moreover, \( g_{k,k} \) is a common factor of every \( k \)-th minor of \( G(z) \) and hence must be equal to 1 because the code is non-catastrophic. Therefore,

\[
G(z) = \begin{pmatrix}
g_{1,1} & \cdots & g_{1,k} & g_{1,k+1} & \cdots & g_{1,k} \\
g_{k-1,k-1} & \cdots & g_{k-1,k} & g_{k-1,k+1} & \cdots & g_{k-1,k} \\
1 & \cdots & g_{1,k-1} & g_{1,k} & \cdots & g_{1,k} \\
\end{pmatrix}
\]

Furthermore, \( G(z) \) is equivalent to

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

by adding the \( k \)-th row times \( g_{i,k} \) to the \( i \)-th row for \( i \in \{1, \ldots, k-1 \} \). Let \( \tilde{g}_i \) be the \( i \)-th row of \( \tilde{G}(z) \). Similarly to before, from \( \tilde{g}_i \tilde{g}_k \) for \( i \in \{1, \ldots, k-1 \} \), we get

\[
g_{k-1,k-1} = g_{k-1,2k-1} = g_{k-1,k-1} + g_{k,2k-1} = 0.
\]

Hence, using the same argument as before we get that \( \tilde{G}(z) \) is equivalent to

\[
\begin{pmatrix}
g_{1,1} & \cdots & g_{1,k-2} & 0 & g_{1,k} & \cdots & g_{1,2k-2} & 0 & 0 \\
\vdots & \ddots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\
g_{2,1} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
1 & 0 & \cdots & 1 & 0 & \cdots & 1 & 0 & \cdots & 1 \\
\end{pmatrix}
\]

Repeating this reasoning, we find that every double upper triangle generator matrix of a binary self-dual code is equivalent to

\[
G'(z) = \begin{pmatrix}
1 & & & & & \\
& 1 & & & & \\
& & \ddots & & & \\
& & & 1 & & \\
& & & & 1 & \\
\end{pmatrix}
\]

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

### VI. CONSTRUCTION OF SELF-DUAL CONVOLUTIONAL CODES

In this section, we will generalize known construction methods for block codes to construction methods for convolutional codes. Finally, we will have a closer look at some examples for binary self-dual convolutional codes of different lengths \( n \), dimensions \( k \) and degrees \( \delta \).

#### A. Construction Methods

In this subsection, we will consider some constructions, where already known self-dual convolutional codes are used to find new ones. The first three results are true over any polynomial ring over a general finite field and the rest only over the polynomial ring over the binary field. One immediately sees the following:

**Remark 6:** If \( G(z) \in \mathbb{F}_q[z]^{k \times n} \) and \( \tilde{G}(z) \in \mathbb{F}_q[z]^{k' \times n'} \) are generator matrices of two self-dual convolutional codes, then the code \( \mathcal{C}' \) generated by

\[
G'(z) = \begin{pmatrix}
G(z) & 0 \\
0 & \tilde{G}(z)
\end{pmatrix} \in \mathbb{F}_q[z]^{(k+k') \times (n+n')}
\]

is also self-dual.

The following proposition is a generalization of a result presented in [4].

**Proposition 43:** Let \( \mathcal{C} \subset \mathbb{F}_q[z]^n \) be a self-dual convolutional code with generator matrix \( G(z) \). Let \( r \in \mathbb{N} \) and for \( i = 1, \ldots, n \), let \( M_i(z) \in \mathbb{F}_q[z]^{n \times n} \) be such that \( M_i(z)M_i(z)^T = \lambda_i I_n \) for \( \lambda_i \in \mathbb{F}_q^* \). Let \( A_1, \ldots, A_r \) be \( n \times n \) permutation matrices. Then

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

generates a self-dual code.

**Proof:** The idea of the proof is similar to that of [4, Prop. 3.1]. We know that \( G(z) \) has Smith form \([I_k \ 0]\) and \( M_i(z) \) is unimodular as

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

Therefore also \( G(z)M_i(z) \) 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 \([I_k \ 0]\) as well, which lets us conclude with Theorem 28 that \( G_r(z) \) generates a non-catastrophic code. Finally due to \( A_i A_i^T = I_n \), we find

\[
G_r(z)G_r(z)^T = G(z)M_1(z)A_1 \cdots M_r(z)(A_rA_r^T)M_r(z)^T \cdots A_1^T M_1(z)^T G(z)^T = \lambda_1 \lambda_2 \cdots \lambda_r G(z)G(z)^T = 0.
\]

Hence, with Theorem 38 we can conclude that \( G_r(z) \) generates a self-dual convolutional code.

**Example 4:** Take the self-dual convolutional code from Example 2, i.e.

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

Let \( r = 1 \) and

\[
M_1 = \begin{pmatrix} 1 & 4 & 0 & 0 \\ -4 & 1 & 0 & 0 \\ 0 & 1 & -4 \\ 0 & 0 & 4 & 1 \end{pmatrix} \in \mathbb{F}_5[z]^{4 \times 4},
\]

i.e. \( \lambda_1 = 2 \), and

\[
A_1 = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}.
\]

Then

\[
G_1(z) = \begin{pmatrix} 1 + 3z & 4z & 3 \ast + z \\ 2z & 3z & 4z & 3 + z \ast \end{pmatrix} \in \mathbb{F}_5[z]^{2 \times 4}
\]
generates a self-dual convolutional code as well.
In Section II-B, 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 generalizations possess. At first, we will consider the building-up construction, which we not only generalize from block to convolutional codes but also from binary fields to any finite field that has the property that \(-1\) is a square in this field.

**Proposition 44 (Generalized Building-Up Construction):** Let \(G(z)\) be a generator matrix of a self-dual \((2k, k)\) convolutional code \(\mathcal{C}\) over \(\mathbb{F}_q[z]\). Let \(q = p^j\), where \(p = 3 \mod 4\) and \(l\) being odd does not hold simultaneously. Further, let

\[
a^2 + b^2 = 0 \mod p
\]

for some \(a, b \in \mathbb{F}_q^*\), \(f(z) \in \mathbb{F}_q[z]^{2k}\) with

\[
f(z)f(z)^T = -(a^{-1})^2
\]

and define

\[
y_i(z) := f(z)g_i(z)^T
\]

for \(1 \leq i \leq k\). Then

\[
\tilde{G}(z) = \begin{pmatrix}
-a^{-1} & 0 & f(z) \\
ay_1(z) & by_1(z) \\
\vdots & \vdots & G(z) \\
ay_k(z) & by_k(z)
\end{pmatrix} \in \mathbb{F}_q[z]^{(k+1)\times(2k+2)}
\]

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

**Proof:** According to Theorem 38, we have to show that \(G(z)\) generates a non-catastrophic code and that \(G(z)G(z)^T = 0\). Computing the Smith form of \(G(z)\) and using that \(G(z)\) has column Hermite form \([I_k \ 0]\) yields

\[
\tilde{G}(z) = \begin{pmatrix}
-a^{-1} & 0 & f(z) \\
ay_1(z) & by_1(z) \\
\vdots & \vdots & G(z) \\
ay_k(z) & by_k(z)
\end{pmatrix} \sim \begin{pmatrix}
1 & 0 & 0 \\
0 & by_1(z) \\
\vdots & \vdots & G(z) \\
0 & by_k(z)
\end{pmatrix} \sim [I_{k+1} \ 0],
\]

Applying Theorem 28 we obtain that \(\tilde{G}(z)\) generates a non-\(\tilde{G}(z)\)\-catastrophic code. Let \(\hat{g}_i(z)\) be the \(i\)-th row of \(\tilde{G}(z)\) and \(g_i(z)\) be the \(i\)-th row of \(G(z)\). To see that \(\tilde{C}\) is self-dual, we compute

\[
\hat{g}_1(z)\hat{g}_1(z)^T = -(a^{-1})^2 + f(z)f(z)^T = 0
\]

\[
\hat{g}_1(z)g_1(z)^T = -a^{-1}ay_1(z) + f(z)g_1(z) = 0
\]

\[
\hat{g}_{i+1}(z)\hat{g}_{j+1}(z)^T = a^2y_i(z)y_j(z) + b^2y_i(z)g_j(z)^T + g_i(z)g_j(z)^T = g_i(z)g_j(z)^T = 0
\]

**Remark 7:** If we set \(p = 2\) in the preceding theorem, one has \(a = b = 1\) and obtains a straightforward generalization of Theorem 10.

**Example 5:** From Section V-B, we know that

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

generates a binary self-dual convolutional code. We apply the generalized building-up construction with \(f(z)f(z)^T = (1, z, z^2, z^2 + z \in \mathbb{F}_2[z]^4\). The condition of \(f(z)f(z)^T = a = 1\) is fulfilled and

\[
y_1(z) = f(z)(1, 1, 1, 1)^T = 1,
\]

\[
y_2(z) = f(z)(0, 1, z + 1, z)^T = z.
\]

Therefore

\[
G_2(z) = \begin{pmatrix}
1 & 0 & f_1(z) & f_2(z) & f_3(z) & f_4(z) \\
y_1(z) & y_1(z) & 1 & 1 & 1 & 1 \\
y_2(z) & y_2(z) & 0 & 1 & z + 1 & z
\end{pmatrix}
\]

\[
= \begin{pmatrix}
1 & 0 & 1 & z & z^2 & z^2 + z \\
1 & 1 & 1 & 1 & 1 & 1 \\
z & 0 & 1 & z + 1 & z
\end{pmatrix}
\]

generates a binary self-dual convolutional code.

Next we show that Theorem 11 cannot be generalized to convolutional codes.

**Lemma 45:** 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 V-B,

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

generates a binary self-dual convolutional code. We claim that \(G(z)\) cannot be constructed with the generalized building-up construction, because every codeword of the corresponding code \(\tilde{C}\) has the property that if one of its entries is equal to 0, then none of its entries can be equal to 1. To see this, assume that for some \(a(z), b(z) \in \mathbb{F}_2[z],
\]

\[
c(z) = a(z)(1, 1, 1, 1) + b(z)(0, z^2 + z + 1, z, z^2 + 1)
\]

has one entry that is equal to 0. This implies \(a(z) = b(z)d(z)\) for some \(d(z) \in \{0, z^2 + z + 1, 1, z, z^2 + 1\}\) and hence, \(gcd(a(z), b(z)) = a(z) \neq 1\). Consequently, \(a(z) + b(z)d(z) \neq 1\) (by the Lemma of Bezout). We conclude that there is no codeword that has a entry equal to 0 as well as an entry equal to 1, meaning that the top-left \((1, 0)\) in the building-up construction cannot be achieved by row operations or column permutations on \(G(z)\). Hence, \(\tilde{C}\) cannot be constructed with the generalized building-up construction.

It remains to show that \(d_{free}(\mathcal{C}) > 2\). Indeed,

\[
d_{free}(\mathcal{C}) \geq \min_{m_0 \neq 0} wt \left( \begin{pmatrix}
m_0 & m_1 \\
G_0 & G_1
\end{pmatrix} \right)
\]

\[
= \min_{m_0 \neq 0} wt \left( \begin{pmatrix}
m_0 & m_1 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 & 1
\end{pmatrix} \right)
\]

\[
= 4.
\]
We conclude that $C$ has free distance $d_{\text{free}} = 4 > 2$. Since the code is self-dual and hence the parity-check matrix is equal to the generator matrix, one could also use the fact that each set of three columns of $G(z)$ is linearly independent to conclude $d_{\text{free}} = 4$.

As a consequence, the full classification of (binary) self-dual convolutional codes cannot be done by using the generalized building-up construction. In the following, we will investigate generalizations of the Harada-Munemasa construction.

**Proposition 46:** 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 $a_i(z) \in \mathbb{F}_2[z]$, for $i \in \{1, \ldots, k\}$. Then
\[
\tilde{G}(z) = \begin{pmatrix}
a_1(z) & a_1(z) \\
\vdots & \vdots \\
a_k(z) & a_k(z)
\end{pmatrix} G(z) \in \mathbb{F}_2[z]^{k \times (2k+2)}
\]
generates a binary self-orthogonal convolutional code $\tilde{G}$.

**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 $G(z)$. Then
\[
g_i(z)g_j(z)^\top = a_i(z)a_j(z) + a_i(z)a_j(z) + g_i(z)g_j(z)^\top = g_i(z)g_j(z)^\top = 0.
\]

**Definition 47:** 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 $a_i(z) \in \mathbb{F}_2[z]$, $i \in \{1, \ldots, k\}$,
\[
\tilde{G}(z) = \begin{pmatrix}
a_1(z) & a_1(z) \\
\vdots & \vdots \\
a_k(z) & a_k(z)
\end{pmatrix} G(z) \in \mathbb{F}_2[z]^{k \times (2k+2)}
\]
like in Proposition 46. If there exists $f(z) \in \mathbb{F}_2[z]^{2k+2}$ such that
\[
G_1(z) = \begin{pmatrix}
f_1(z) & \cdots & f_{2k+2}(z) \\
a_1(z) & a_1(z) \\
\vdots & \vdots \\
a_k(z) & a_k(z)
\end{pmatrix} \in \mathbb{F}_2[z]^{(k+1) \times (2k+2)}
\]
generates a self-dual code, then we say that $G_1(z)$ is a self-dual completion or just a completion of $G(z)$.

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

**Lemma 48:** Choosing $f(z) = (1, 1, 0, \ldots, 0)$ in Definition 47 leads to a generator matrix of a self-dual code. More concretely,
\[
G_1(z) = \begin{pmatrix}
1 & 1 & 0 \\
a_1(z) & a_1(z) \\
\vdots & \vdots \\
a_k(z) & a_k(z)
\end{pmatrix} \in \mathbb{F}_2[z]^{(k+1) \times (2k+2)}
\]
is a self-dual completion for any $a_i(z) \in \mathbb{F}_2[z]$, $i \in \{1, \ldots, k\}$.

**Proof:** Obviously, $G_1(z)$ and
\[
G_2(z) = \begin{pmatrix}
1 & 1 & 0 \\
0 & 0 & \vdots \\
0 & 0 & G(z)
\end{pmatrix}
\]
generate the same code, making the choice of the $a_i(z)$ pointless. But still $G_2(z)G_2(z)^\top = 0$ and $G_2(z) \sim [I_{k+1} \ 0]$, implying that the code generated by $G_2(z)$ is self-dual.

Because of the preceding lemma we make the following definition.

**Definition 49:** Let
\[
G_1(z) = \begin{pmatrix}
f_1(z) & \cdots & f_{2k+2}(z) \\
a_1(z) & a_1(z) \\
\vdots & \vdots \\
a_k(z) & a_k(z)
\end{pmatrix} \in \mathbb{F}_2[z]^{(k+1) \times (2k+2)}
\]
be a self-dual completion. Then we say that the completion was trivial if $G_1(z)$ and
\[
\begin{pmatrix}
1 & 1 & 0 \\
a_1(z) & a_1(z) \\
\vdots & \vdots \\
a_k(z) & a_k(z)
\end{pmatrix} \in \mathbb{F}_2[z]^{(k+1) \times (2k+2)}
\]
generate the same code. If they do not, we call the completion non-trivial.

**Remark 8:** 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.

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

**Example 6:** We know that $(1, 1)$ is a binary self-dual $(2, 1)$ convolutional code (actually the only one). We 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 46, $(z \ z \ 1 \ 1)$ 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 $C$. By Lemma 41, the all one vector must be part of the code. Therefore, there is $b(z) = (b_1, b_2) \in \mathbb{F}_2[z]^2$ such that
\[
b(z)G(z) = (1, 1, 1, 1).
\]

Note that we abbreviate $b(z)$ and $f_i(z)$ with $b_i$ and $f_i$, respectively. Besides, notice that necessarily $b_1 \neq 0$. This translates into the following system of equations over $\mathbb{F}_2[z]$:
\[
\begin{align*}
b_1f_1 + zb_2 &= 1 \\
b_1f_2 + zb_3 &= 1 \\
b_1f_3 + b_2 &= 1 \\
b_1f_4 + b_2 &= 1.
\end{align*}
\]

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 $b_2 = b_1f_4 + 1$. Hence the system simplifies into
\[
\begin{align*}
b_1f_1 + z(b_1f_4 + 1) &= 1 \\
b_1f_2 + z(b_1f_4 + 1) &= 1.
\end{align*}
\]
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 $f_4 \in \mathbb{F}_2[z]$. Computing its Smith form, one obtains

$$
\left(1 + z + f_4 z \quad 1 + z + f_4 z \quad f_4 \quad f_4 \right) \\
\left(\begin{array}{cccc}
z & z & 1 & 1
\end{array}\right) \sim \left(\begin{array}{cccc}
1 + z & 0 & 0 & 0 \\
z & 0 & 0 & 0
\end{array}\right)
$$

Since the Smith form of $G(z)$ is not $[I_k \ 0]$, $G(z)$ generates a catastrophic and therefore 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 $a_1(z) = 1$. Then $(1 \ 1 \ 1 \ 1)$ 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 V-B.

Therefore, the existence of non-trivial completions depends on the choice of $a_1(z)$. This naturally steers the search for viable conditions on the $a_i(z)$ that admit non-trivial self-dual completions.

**Theorem 50:** Let

$$
\tilde{G}(z) = \begin{pmatrix}
a_1(z) & a_1(z) \\
\vdots & \vdots \\
a_k(z) & a_k(z)
\end{pmatrix} G(z),
$$

Then a non-trivial self-dual completion exists if and only if $(1, \ldots, 1) \in \text{rowspan}(G(z)).$

**Proof:** Let $\tilde{C} = \text{rowspan}(\tilde{G}(z))$,

$$
G_1(z) = \begin{pmatrix}
f_1(z) & \cdots & f_{2k+2}(z) \\
a_1(z) & a_1(z) \\
\vdots & \vdots \\
a_k(z) & a_k(z)
\end{pmatrix} G(z)
$$

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

For submodules $V \subseteq U \subseteq \mathbb{F}_q[z]^n$ denote by $W = U / V$ the submodule $W$ such that $U = V \oplus W$ is a direct sum. Obviously, all self-dual completions of $\tilde{C}$ can be found by looking at all possible new rows $f(z) \in \tilde{C}^\perp / \tilde{C}$ and checking the resulting $C_1$ for self-duality. The main fact we will use to prove this theorem is that if $(1, \ldots, 1) \in \tilde{C}$, then every possible new row $f(z) \in \tilde{C}^\perp$ that we want to add to $G(z)$, fulfills

$$
f(z) f(z)^\top = (f(z)(1, \ldots, 1)) \in \tilde{C}^\perp
$$

as $(1, \ldots, 1) \in \tilde{C}$, which implies that each pair of rows of $G_1(z)$ is orthogonal.

$(\Rightarrow)$ We will prove that $(1, \ldots, 1) \not\in \tilde{C}$ implies that there exists exactly one self-dual completion (up to equivalent generator matrices) and it is the trivial one.

If $C_1$ with generator matrix $G_1(z)$ is assumed to be self-dual (and hence, also non-catastrophic), then $(1, \ldots, 1) \in C_1$. Thus, $\left(\begin{array}{ccc} 1 \\
\vdots \\
a_k(z) \\
C(z)
\end{array}\right)$ is a generator matrix for $C_1$. As $C$ is self-dual, $(1, \ldots, 1) \in \text{rowspan}(G(z))$ and therefore, $C_1$ has also a generator matrix of the form

$$
\begin{pmatrix}
b(z) & b(z) & 0 & \cdots & 0 \\
a_1(z) & a_1(z) \\
\vdots & \vdots \\
a_k(z) & a_k(z)
\end{pmatrix},
$$

for some $b(z) \in \mathbb{F}_2[z]$. As $C_1$ is non-catastrophic $b(z) = 1$ and hence, the self-dual completion is trivial.

$(\Leftarrow)$ We have $\dim(\tilde{C}^\perp / \tilde{C}) = n - k - k = 2$. Let

$$
\tilde{C}^\perp / \tilde{C} = \text{span}\{(1, 1, 0, \ldots, f(z))\}
$$

for some $f(z) = (f_1(z), \ldots, f_{2k+2})(z) \in \mathbb{F}_2[z]^{2k+2}$. As $\tilde{C}^\perp$ is non-catastrophic and $f(z)$ can be completed to a basis of this code, $\gcd(f_1(z), \ldots, f_{2k+2}(z)) = 1$. We want to show that the code $C_1$ generated by

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

is a non-trivial self-dual completion.

First, we observe that $G_1(z) G_1(z)^\top = 0$ as $\tilde{C}$ is self-orthogonal and $f(z) \in \tilde{C}^\perp$ (and $(1,1,1,1) \in \tilde{C}$ implies $f(z) f(z)^\top = 0$). Next, we will show that $C_1$ is non-catastrophic, which then shows that $C_1$ is self-dual. Applying column operations, we obtain

$$
G_1(z) \sim \begin{pmatrix}
f_1(z) & \cdots & f_{k+2}(z) & \hat{f}_{k+3}(z) & \cdots & \hat{f}_{2k+2}(z) \\
a_1(z) & a_1(z) \\
\vdots & \vdots \\
a_k(z) & a_k(z)
\end{pmatrix} I_k
$$

where $\hat{f}_1(z), \ldots, \hat{f}_{2k+2}(z) \in \mathbb{F}_2[z]$ with

$$
\gcd(\hat{f}_1(z), \ldots, \hat{f}_{2k+2}(z)) = \gcd(f_1(z), \ldots, f_{2k+2}(z)) = 1.
$$

Consequently,

$$
G_1(z) \sim \begin{pmatrix} 1 & 0 & 0 \\
0 & I_k & 0
\end{pmatrix}
$$

and $C_1$ is non-catastrophic.

Finally, we will show that $(1,1,0,\ldots,0) \not\in C_1$. Assume conversely that $(1,1,0,\ldots,0) \in C_1$. As $(1,1,1,1) \in \tilde{C}$ implies $f(z) \in \tilde{C}^\perp$ and one has $\tilde{C} \subseteq C_1 \subseteq \tilde{C}^\perp$, one gets that $\tilde{C}_1 = \tilde{C}^\perp$. However, $\dim(C_1) = n - 1 \neq k + 2 = \dim(\tilde{C})$, a contradiction.
Conclusively, we showed \((1, 1, 0, \ldots, 0) \not\in C_1^\perp\) and \(C_1 = C_1^\perp,\) i.e. we have a non-trivial completion to a self-dual convolutional code.

Thus, we can find non-trivial completions with the following two steps:

1. Make sure that \((1, \ldots, 1) \in \text{rowspan}(\tilde{G}(z)) = \tilde{\mathcal{C}},\) which means we must establish that the system

\[
b(z)\tilde{G}(z) = (1, \ldots, 1) \in \mathbb{F}_2[z]^{2k+2},
\]

where \(b(z) = (b_1(z), \ldots, b_k(z)) \in \mathbb{F}_2[z]^k,\) 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

\[
G_1(z) = \begin{pmatrix}
f_1(z) & \cdots & f_{2k+2}(z) \\
a_1(z) & a_1(z) \\
\vdots & \vdots & G(z) \\
a_k(z) & a_k(z)
\end{pmatrix}
\]

generates a binary self-dual convolutional code.

We will further investigate these two steps in the following.

**Remark 9:** A way to make sure that \((2)\) has a solution is by finding a solution to

\[
b(z)G(z) = (1, \ldots, 1) \in \mathbb{F}_2[z]^{2k},
\]

which always exists since \(G(z)\) generates a binary self-dual code (see Lemma 41), and then choosing the \(a_i(z) \in \mathbb{F}_2[z]\) such that

\[
\sum_{i=1}^k a_i(z)b_i(z) = 1,
\]

which is possible since \(b(z)G(z) = (1, \ldots, 1)\) implies \(\gcd(b_1(z), \ldots, b_k(z)) = 1.\) In particular, for any \(j = 1, \ldots, n, a_i(z) = g_{ij}(z)\) for \(i = 1, \ldots, k,\) yields such a solution.

The following lemma shows, how to do the second step, i.e. how to find a suitable \(f(z)\).

**Lemma 51:** Let

\[
f(z) = (f_1(z), \ldots, f_{2k+2}(z)) \in \tilde{\mathcal{C}}^\perp / \text{span}\{\tilde{\mathcal{C}}, (1, 1, 0, \ldots, 0)\}
\]

and \(\gcd(f_1(z), \ldots, f_{2k+2}(z)) = 1.\) Then

\[
\text{span}\{f(z), (1, 1, 0, \ldots, 0)\} = \tilde{\mathcal{C}}^\perp / \tilde{\mathcal{C}}.
\]

**Proof:** Assume conversely that there exists \(h(z) \in \tilde{\mathcal{C}}^\perp / \tilde{\mathcal{C}}\) such that \(h(z) \neq f(z)\) and

\[
\text{span}\{h(z), (1, 1, 0, \ldots, 0)\} = \tilde{\mathcal{C}}^\perp / \tilde{\mathcal{C}}.
\]

As \(f(z) \in \tilde{\mathcal{C}}^\perp / \text{span}\{\tilde{\mathcal{C}}, (1, 1, 0, \ldots, 0)\} = \text{span}\{h(z)\},\)

\(f(z)\) must be a multiple of \(h(z).\) But since the entries of \(f(z)\) have no common non-trivial divisor, \(f(z) = h(z)\) follows immediately. \(\square\)

**Example 7:** 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 V-B, that \(G(z)\) generates a self-dual convolutional code. By inspection, \(b(z)G(z) = (1, 1, 1, 1)\) holds exactly for \(b(z) = (0, 1).\) Hence we need

\[
0 \cdot a_1(z) + 1 \cdot a_2(z) = 1
\]

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 \(f(z) \in \tilde{\mathcal{C}}^\perp / \text{span}\{\tilde{\mathcal{C}}, (1, 1, 0, 0, 0)\}.\) If \(f_1(z) \neq f_2(z),\) we have \(f(z) \not\in \text{span}\{\tilde{\mathcal{C}}, (1, 1, 0, 0, 0)\}.\)

Further, we need \(f(z)\) to be orthogonal to both rows of \(\tilde{G}(z),\) i.e.,

\[
f(z)(z^2 + 1, z^2 + 1, 0, z^2 + z + 1, z, z^2 + 1)^T = 0
\]

and see that this holds for \(f(z) = (0, 1, 0, 0, 0, 1).\) One obtains that

\[
G_1(z) = \begin{pmatrix}
0 & 1 & 0 & 0 & 0 & 1 \\
1 & 1 & 1 & 1 & 1 & 1
\end{pmatrix}
\]

generates a binary self-dual convolutional code.

An immediate consequence of Remark 9 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:** Let

\[
G_1(z) = \begin{pmatrix}
f_1(z) & \cdots & f_{2k+2}(z) \\
a_1(z) & a_1(z) \\
\vdots & \vdots & G(z) \\
a_k(z) & a_k(z)
\end{pmatrix} \in \mathbb{F}_2[z]^{(k+1) \times (2k+2)}
\]

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 8:** By Theorem 50, if we chose \(a_1 = a_2 = 1\) for

\[
\begin{pmatrix}
1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1
\end{pmatrix} \in \mathbb{F}_2[z]^{2 \times 4},
\]

to obtain

\[
\tilde{G}(z) = \begin{pmatrix}
1 & 1 & 1 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 & 1 & 1
\end{pmatrix},
\]

then \(\tilde{G}(z)\) can only be trivially completed as \((1, 1, 1, 1, 1, 1) \not\in \text{rowspan}(G(z)).\) Now, we claim that another generator matrix of the same code, namely

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

has a non-trivial completion for \(a_1(z) = a_2(z) = 1\) in

\[
G_1(z) = \begin{pmatrix}
h(z) & h(z) + 1 & 0 & 0 & 0 & 1 \\
1 & 1 & 1 & 1 & 1 & 1
\end{pmatrix},
\]

where \(h(z) = 1.\)
where \( h(z) \in \mathbb{F}_2[z] \). \( G_1(z) \) generates a non-catastrophic code, since
\[
\det \begin{pmatrix}
 h(z) & h(z) + 1 & 0 \\
 1 & 1 & 1 \\
 1 & 1 & 0
\end{pmatrix} = h(z) + 1 + h(z) = 1.
\]

Moreover, \( G_1(z)G_1(z)^T = 0 \) by inspection. Hence, \( G_1(z) \) generates a self-dual code by Theorem 38.

So the only thing left to show is non-triviality. Assume by contradiction that there exists \( b(z) = (b_1, b_2, b_3) \in \mathbb{F}_2[z]^3 \) (again we use the abbreviation \( b_i \) for \( b_i(z) \)) such that
\[
b(z)G_1(z) = (1, 1, 0, 0, 0, 0),
\]
or rather, in terms of a system of equations,
\[
\begin{align*}
b_1 h + b_2 + b_3 &= 1 \\
b_1 (h + 1) + b_2 + b_3 &= 1 \\
b_2 &= 0 \\
b_2 &= 0 \\
b_2 + b_3 &= 0 \\
b_1 + b_2 + b_3 &= 0.
\end{align*}
\]

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 \( (a_1(z), \ldots, a_k(z))^T \).

We were previously asking the question, whether all binary self-dual convolutional codes with \( d_{\text{free}} > 2 \) can be constructed with the generalized Harada-Munemasa construction. This question will be left open for codes of lengths greater than 4.

**Lemma 52:** All binary self-dual (4, 2) convolutional codes can be constructed with the generalized Harada-Munemasa construction.

**Proof:** Let \( C \) be a binary self-dual (4, 2) convolutional code. Then by Section V-B, 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 \( C \) with the generalized Harada-Munemasa construction.

Finally, we want to connect the generalized building-up construction and the generalized Harada-Munemasa construction.

**Lemma 53:** 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
\[
G_1(z) = \begin{pmatrix}
 1 & 0 & f(z) \\
y_1(z) & y_1(z) & \vdots & \vdots \\
y_k(z) & y_k(z)
\end{pmatrix}
\]
be a generator matrix 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
\[
\tilde{G}(z) = \begin{pmatrix}
a_1(z) & a_1(z) \\
\vdots & \vdots \\
a_k(z) & a_k(z)
\end{pmatrix}
\]
and then add \((1, 0, f(z))\) as a new row. On the other hand, in Lemma 45, 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.

**VII. Conclusion**

We started with finding equivalent conditions to self-duality for convolutional codes. This forms the foundation for the other results obtained in this paper. We continued with the full classification of 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 investigated 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 \( d_{\text{free}} > 2 \), 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 is for block codes. For the generalized Harada-Munemasa construction this question about viability is still open for codes of lengths greater than 4.

**References**

[1] S. Bouyuklieva, “Self-dual codes,” in Concise Encyclopedia of Coding Theory, C. Huffman, J. Kim, and P. Sole, Eds. Boca Raton, FL, USA: CRC Press, 2021.

[2] A. R. Calderbank, E. M. Rains, P. M. Shor, and N. J. A. Sloane, “Quantum error correction via codes over GF(4),” IEEE Trans. Inf. Theory, vol. 44, no. 4, pp. 1369–1378, Jul. 1998.

[3] J. H. Conway and V. Pless, “On the enumeration of self-dual codes,” J. Combinat. Theory A, vol. 28, no. 1, pp. 26–53, Jan. 1980.
[4] P. Gaborit and A. Otmani, “Experimental construction of self-dual codes,” *Finite Fields Their Appl.*, vol. 9, no. 3, pp. 372–394, 2003.

[5] F. R. Gantmacher, *The Theory of Matrices*, vol. 1, New York, NY, USA: Chelsea Publishing Company, 1977.

[6] M. Grass and T. A. Gulliver, “On self-dual MDS codes,” in *Proc. IEEE Int. Symp. Inf. Theory*, Toronto, ON, Canada, Jul. 2008, pp. 1954–1957.

[7] M. Harada and A. Munemasa, “Classification of self-dual codes of length 36,” *Adv. Math. Commun.*, vol. 6, no. 2, pp. 229–235, 2012.

[8] W. C. Huffman and V. Pless, *Fundamentals of Error-Correcting Codes*. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[9] R. Johannesson, P. Stahl, and E. Wittenmark, “A note on type II convolutional codes,” *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1510–1514, Jul. 2000.

[10] R. Johannesson and K. S. Zigangirov, *Fundamentals of Convolutional Coding*. New York, NY, USA: IEEE Press, 1999.

[11] T. Kailath, *Linear Systems*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1980.

[12] J.-L. Kim, “New extremal self-dual codes of length 36, 38, and 58.” *IEEE Trans. Inf. Theory*, vol. 47, no. 1, pp. 386–393, Jan. 2001.

[13] J. Lieb, R. Pinto, and J. Rosenthal, *Convolutional Codes Concise Encyclopedia of Coding Theory*, C. Huffman, J. Kim, and P. Sole, Eds. Boca Raton, FL, USA: CRC Press, 2021.

[14] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. Amsterdam, The Netherlands: North-Holland Publishing Co., 1977.

[15] C. A. Melchor, P. Gaborit, J.-L. Kim, L. Sok, and P. Sole, “Classification of extremal and s-extremal binary self-dual codes of length 36,” *IEEE Trans. Inf. Theory*, vol. 58, no. 4, pp. 2253–2262, Nov. 2012.

[16] H. S. Palines and V. P. Sison, “On self-dual convolutional codes over rings,” *East-West J. Math.*, vol. 14, no. 1, pp. 28–36, 2012.

[17] V. Pless, “A classification of self-orthogonal codes over GF(2),” *Discrete Math.*, vol. 3, no. 1, pp. 209–246, 1972.

[18] E. M. Rains and N. J. A. Sloane, “Self-dual codes,” in *Handbook of Coding Theory*, V. S. Pless and W. C. Huffman, Eds. Amsterdam, The Netherlands: Elsevier, 1998, pp. 177–294.

[19] J. Robinson, “Error propagation and definite decoding of convolutional codes,” *IEEE Trans. Inf. Theory*, vol. IT-14, no. 1, pp. 121–128, Jan. 1968.

[20] N. J. Russell, “Self-dual codes and indecomposable building blocks,” M.S. thesis, Fac. Graduate School Eastern Kentucky Univ., Richmond, KY, USA, 2016.

[21] E. V. York, “Algorithmic description and construction of error correcting codes: A linear systems point of view,” Ph.D. dissertation, Univ. Notre Dame, Notre Dame, IN, USA, 1997.

**Sebastian Heri** received the bachelor’s and master’s degrees in mathematics from the University of Zürich. During his master’s studies, he focused on various aspects of algebra, such as mathematical coding theory and mathematical cryptography. In his master thesis, he developed the basic theory for self-dual convolutional codes and the basis for the present article. After completing the master’s degree, he started to work as a Data Analytics Specialist in the industry.

**Julia Lieb** received the bachelor’s, master’s, and Ph.D. degrees from the University of Wuerzburg, Germany, in 2012, 2014, and July 2017, respectively, all in mathematics. In the fall of 2016, she was a visiting Ph.D. student with the Applied Algebra Group at the University of Zürich. Her Ph.D. thesis is based on the interface of the theory of polynomial matrices, linear systems theory, and coding theory. After completing her Ph.D., she received several fellowships to do research in coding theory and worked as postdoctoral researcher at the Universities of Wuerzburg, Aveiro, Portugal, and Zurich. Additionally, in Summer 2021, she was substitute Assistant Professor for Discrete Mathematics at the University of Passau, Germany. She is currently an Assistant Professor for Discrete Mathematics at TU Ilmenau, Germany.

**Joachim Rosenthal** (Fellow, IEEE) received the Diploma degree in mathematics from the University of Basel in 1986 and the Ph.D. degree in mathematics from Arizona State University in 1990. From 1990 to 2006, he was with the University of Notre Dame, USA, where he was the holder of an Endowed Chair of applied mathematics and he was also a Concurrent Professor of electrical engineering. Since 2004, he has been a Professor of applied mathematics with the University of Zürich, where was the past Chair of the Institute of Mathematics and the Vice Dean of the College of Science. He will be (2024–2025) the President of the Swiss Mathematical Society. His current research interests include coding theory and cryptography. He has served as an organizer or the program chair for numerous international conferences, e.g., he is one of the Technical Program Chair of ISIT 2024, Athens, Greece. In addition, he has served on numerous editorial boards.