Carl von Ossietzky
Universität Oldenburg

Diplomstudiengang Mathematik

DIPLOMARBEIT

Minimal Cyclic Convolutional Codes

Vorgelegt von: Barbara Langfeld
Betreuende Gutachterin: PD Dr. Heide Glüsing-Lüerßen
Zweiter Gutachter: Prof. Dr. Wiland Schmale

Eingereicht in Oldenburg am 30. Juni 2003

Das vorliegende Exemplar der Diplomarbeit
unterscheidet sich an wenigen Stellen von der eingereichten Version
aufgrund von sprachlicher Glättung und Korrektur von Tippfehlern.
Augsburg im August 2003, Barbara Langfeld

Für meine Eltern

Contents

List of Symbols

The following symbols will be used in this thesis without further explanation:

$0_{k\times n}, 0_k$	The $k \times n$ zero matrix resp. the zero vector of length k.
$\langle\, g \,\rangle$	The ideal generated by g in a commutative ring (specified in the context).
$A^{\mathrm{t}}, v^{\mathrm{t}}$	The transpose of the matrix A resp. the vector v.
A^{-1}	The inverse of the regular matrix A.
$\deg_x f, \deg_z f$	The x-degree resp. the z-degree of the polynomial f.
$\det(A)$	The determinant of the square matrix A.
$\operatorname{diag}(A_1, \ldots, A_n)$	The (block) diagonal matrix which has the (block) entries $A_1, \ldots, A_n$ on its diagonal.
$\dim_{\mathbb{F}} U$	The dimension of the $\mathbb{F}$-vectorspace U (where $\mathbb{F}$ is a field).
$\mathbb{F}[x], \mathbb{F}[z]$	The ring of all polynomials over $\mathbb{F}$ in the indeterminate x resp. z (where $\mathbb{F}$ is a field).
$\mathbb{F}(z)$	The field of all rational polynomials over $\mathbb{F}$ in the indeterminate z (where $\mathbb{F}$ is a field).
$\gcd(m, n)$	The greatest common divisor of the integers m and n (which is always a positive integer, if m or n is nonzero).
$Gl_n(\mathbb{F})$	The set of all regular (i.e. invertible) $n \times n$ matrices over the field $\mathbb{F}$.
$Gl_n(\mathbb{F}[z])$	The set of all unimodular $n \times n$-matrices over the ring $\mathbb{F}[z]$ (where $\mathbb{F}$ is a field), i.e. $G \in Gl_n(\mathbb{F}[z])$ if and only if $\operatorname{rank} G(a) = n$ for all $a \in \overline{\mathbb{F}}$ where $\overline{\mathbb{F}}$ denotes an algebraic closure of $\mathbb{F}$.
I_n	The $n \times n$ identity matrix, i.e. $I_n = \operatorname{diag}(1, \ldots, 1)$ where "1" appears n times.
$\operatorname{im}_{\mathbb{F}} A, \operatorname{im}_{\mathbb{F}[z]} A$	The set of all $\mathbb{F}$- resp. $\mathbb{F}[z]$-linear combinations of the rows of the matrix A (where $\mathbb{F}$ is a field). (The $\mathbb{F}$- resp. $\mathbb{F}[z]$-image of A.)
$\ker_{\mathbb{F}} A, \ker_{\mathbb{F}[z]} A$	The set of all vectors v with entries in $\mathbb{F}$ resp. $\mathbb{F}[z]$ such that $vA = 0$ (where $\mathbb{F}$ is a field). (The $\mathbb{F}$- resp. $\mathbb{F}[z]$-kernel of A.)
$\mathbb{N}, \mathbb{N}_0$	The set $\mathbb{N}$ is the set of all positive integers and $\mathbb{N}_0$ is $\mathbb{N} \cup \{0\}$.
$\operatorname{rank} A, \operatorname{rank}_{\mathbb{F}} A$	The rank of the matrix A over a field resp. the field $\mathbb{F}$.
$\operatorname{span}_{\mathbb{F}}(v_1, \ldots, v_n)$, $\operatorname{span}_{\mathbb{F}[z]}(v_1, \ldots, v_n)$	The set of all $\mathbb{F}$- resp. $\mathbb{F}[z]$-linear combinations of the vectors $v_1, \ldots, v_n$. (The $\mathbb{F}$- resp. $\mathbb{F}[z]$-span of $v_1, \ldots, v_n$.)
$\mathbb{Z}$	The set of all integers.

0 Introduction

There are many examples in everyday life, where some information must be stored or communicated. Examples are data storage on CD's or data transmission via the internet or via satellites. But some data on a CD might be destroyed by a scratch and also the data which pass space runs the risk of being changed or lost ("the data pass a noisy channel"). So any part of this information must be protected from being mutilated by storing resp. communicating it with redundancy ("encoding"), and it must be possible to recover the original data from the possibly changed data ("decoding").

But how can we use the idea of redundancy for both encoding and decoding our information easily and efficiently? Coding theory is one attempt to deal with this problem. Coding theory as a mathematical branch started with the article of Shannon [Sha48] in the year 1948.

We assume that a certain message source is represented by the vectors in $\mathbb{F}^k$ (the "message words"), where $\mathbb{F}$ is a finite field. The encoding procedure is described by an injective linear mapping $\mathbb{F}^k \to \mathbb{F}^n$ resp. with a full rank matrix $G \in \mathbb{F}^{k\times n}$, $n \geq k$, according to

$$(\,\cdot\,)G \;:\; \mathbb{F}^k \to \mathbb{F}^n \quad , \quad u \mapsto uG \quad .$$

The set $\operatorname{im}_{\mathbb{F}} G$ is called a block code and the elements of $\operatorname{im}_{\mathbb{F}} G$ are called code words. Let two different codewords u_1G, u_2G differ in at least d entries and assume that the codewords pass a noisy channel, where each codeword is changed in at most $\lfloor \frac{d-1}{2} \rfloor$ components. It is a familiar result from coding theory that in this case the original codewords can be recovered completely from the changed ones and therefore the original message can be recovered completely, too — at least at theory.

If we want to encode a whole sequence of message words $m_0, \ldots, m_t$ we can identify this sequence with the vector polynomial $\sum_{i=0}^{t} m_i z^i$. For the coding procedure we extend $(\,\cdot\,)G$ to the mapping

$$(\,\cdot\,)G \;:\; \mathbb{F}[z]^k \to \mathbb{F}[z]^n \quad , \quad \sum_{i=0}^{t} m_i z^i \mapsto \sum_{i=0}^{t} (m_i G) z^i \quad .$$

Now $(\,\cdot\,)G$ is an injective $\mathbb{F}[z]$-module homomorphism. Note that in this setting the z^j-th term of $(\sum_{i=0}^{t} m_i z^i)G$ only depends on m_j because G has constant entries. In this sense, the encoder $(\,\cdot\,)G$ has "no memory" and we could apply G onto the single message words $m_0, \ldots, m_t$ just as well.

But if we assume that G has polynomial entries in z, i.e. that $(\cdot)G$ is an arbitrary injective $\mathbb{F}[z]$-module homomorphism, then in general the z^j-th term of $(\sum_{i=0}^{t} m_i z^i)G$ will not be determined only by m_j (but of $m_j, m_{j-1}, \ldots$) and $(\cdot)G$ has some kind of "memory". In this sense the submodule $\mathrm{im}_{\mathbb{F}[z]}G$ has some advantage by comparison with block codes. If it satisfies some further desirable properties, it is called a convolutional code (cf. Definition 1.6).

For applications block codes resp. convolutional codes must have additional properties like easy encoding and above all they must allow easy decoding algorithms. The so called "cyclic block codes" are attractive from this point of view: They can be implemented easily and there are various practical methods for decoding them. Consequently, it would be nice to have a generalization to convolutional codes.

Yet, the theory of cyclic convolutional codes is still in the beginnings. This thesis wants to give a small contribution to the work that has already been done to shed some light on this point. We investigate the so called "minimal cyclic convolutional codes" and, as a special case, 1-dimensional cyclic convolutional codes. We are able to give to give a result about the existence of minimal cyclic convolutional codes and to determine the structure of a generator matrix of a 1-dimensional cyclic convolutional code.

A more detailed overview about the structure of this work is given now:

In Section 1 we will recall basic definitions and statements from the theory of block codes and convolutional codes. Convolutional codes are defined to be direct summands of $\mathbb{F}[z]^n$ (cf. Definition 1.6).

Section 2 provides detailed information about the so called "PQR-representation" and "KLM-representation" of submodules of $\mathbb{F}[z]^n$. The main results (Theorems 2.12, 2.13, 2.15, 2.22, 2.23, 2.24) are not new and only Theorem 2.12 will be used later on. But – and this is the reason why we treat this topic so intensively – we give purely (linear) algebraic proofs. To our knowledge, this is the first time that the proofs do not rely on system theory but only on linear algebra, matrix theory and the theory of convolutional codes.

In Section 3 we turn to the notion of cyclicity. First we sum up common results from the theory of cyclic block codes in Section 3.1. Then we introduce a generalized concept of cyclicity for submodules in Section 3.2 which is due to Piret [Pir76] and Roos [Roo79]. Based on the work of Schmale and Glüsing-Lüerßen [GS02a] we provide the necessary tools for our further investigations.

Minimal cyclic convolutional codes, the topic of this theses, will be defined to be those cyclic convolutional codes, which have no proper cyclic sub-codes. In Section 4 minimal cyclic convolutional codes are being investigated. The main result is Theorem 4.15 which gives necessary and sufficient conditions for the existence of minimal cyclic convolutional codes with nonzero complexity.

Section 5 is concerned with 1-dimensional cyclic convolutional codes, which turn out to be special examples for minimal cyclic convolutional codes. We will deepen the result about the existence of minimal cyclic convolutional codes for 1-dimensional cyclic convolutional codes in Section 5.1. Moreover, the shape of generator matrices of 1-dimensional cyclic convolutional codes is determined in Section 5.2. This allows us to present some construction methods for 1-dimensional cyclic convolutional codes resp. the investigation of special 1-dimensional cyclic convolutional codes in Section 5.3.

Finally, Section 6 reminds us of some problems, which where not solved in the present thesis (Section 6.1) and introduces a further research topic in Section 6.2.

Acknowledgements. I like to thank PD Dr. Heide Glüsing-Lüerßen and Prof. Dr. Wiland Schmale. In the lectures of Wiland Schmale I became aquainted with linear algebra and algebra. Heide Glüsing-Lüerßen woke my interest in coding theory. She attended to me carefully while I worked on this thesis and had an ear for me almost at any time.
Furthermore, I like to thank all the other people, who listened to me when I wanted to discuss my mathematical problems. In particular, these are Aiso Heinze and the people from the "KandidatInnen-Seminar" of Prof. Dr. Ulrich Knauer. Moreover, I wish to thank Onno Meyer, who read this thesis before submission and gave suggestions for improving my English.

1 Basic Information on Block Codes and Convolutional Codes

Throughout this thesis, $\mathbb{F}$ denotes a finite field. Sometimes the notation $\mathbb{F}_q$ is used to indicate that $\mathbb{F}$ has q elements. We will regard all vectors as row vectors, as it is usual in coding theory. Thus

$$\mathbb{F}[z]^n := \{(v_1, \ldots, v_n) \mid v_i \in \mathbb{F}[z] \text{ for } i = 1, \ldots, n\} \quad .$$

Consequently, images and kernels of matrices will always denote *left*-images and *left*-kernels. Moreover, the symbols given on page 5 will be used without further explanation.
Sometimes computations were made with the aid of the computer algebra system MAPLE (cf. [MAP01]). We will always indicate whether the computer was used.

In the following we give necessary preliminaries and notations which we use throughout this thesis. Most time we restrict ourselves to the presentation of definitions and results; proofs are omitted.
Section 1.1 deals with block codes. For details we refer to [MS78], [LC83] or [Bet98].
In Sections 1.2 and 1.3 we introduce basics about submodules of $\mathbb{F}[z]^n$ and convolutional codes. More information is presented in the fundamental articles [For70] and [For75] and, for instance, in the books [Pir88a], [JZ99] or in the article [McE98].

1.1 Block Codes

A k-dimensional subspace $\mathcal{C}$ of the vectorspace $\mathbb{F}^n$ (equivalently a direct summand of $\mathbb{F}^n$ with dimension k) is called a (n,k)**-block code over** $\mathbb{F}$. An element of $\mathbb{F}^k$ is called a **message word** and an element of $\mathcal{C}$ is called a **code word**.

The **Hamming weight** or simply **weight** of a vector $v \in \mathbb{F}^n$, denoted by $\mathrm{wt}(v)$, is the number of nonzero entries in v. Two different codewords of a block code $\mathcal{C}$ differ in at least $d(\mathcal{C}) := \min\{wt(v) \mid 0 \neq v \in \mathcal{C}\}$ entries, therefore $d(\mathcal{C})$ is called the **distance** of $\mathcal{C}$. It is easy to see (for example with Lemma 1.1) that we have $d(\mathcal{C}) \leq n - k + 1$ for an (n,k)-bock code $\mathcal{C}$. This bound is called the **MDS-bound**[1] (for block codes). A block code reaching this bound is called **MDS-block code**.

[1] "MDS" stands for **m**aximum **d**istance **s**eperable.

If the noisy channel which the codewords have to pass during the transmission is "not too noisy", i.e. it changes any codeword in at most $\lfloor \frac{1}{2}(d(\mathcal{C})-1) \rfloor$ entries, then any error can be corrected and the original message word can be recovered completely (see e.g. [MS78, Ch. 1, §3, Thm. 2]) — at least theoretically.

For any (n,k)-block code there exist matrices $G \in \mathbb{F}^{k\times n}$ and $H \in \mathbb{F}^{n\times(n-k)}$ (both having full rank) such that $\mathcal{C} = \operatorname{im}_{\mathbb{F}} G = \ker_{\mathbb{F}} H$. (We define $\{0_n\} =: \operatorname{im}_{\mathbb{F}} G$ for $G \in \mathbb{F}^{0\times n}$ and $\mathbb{F}^n =: \ker_{\mathbb{F}} H$ for $H \in \mathbb{F}^{n\times 0}$.) We call G a **generator matrix** and H a **parity check matrix** of $\mathcal{C}$. Generator matrix and parity check matrix are unique in the following sense: If $\hat{G}$ resp. $\hat{H}$ is another generator matrix resp. parity check matrix of $\mathcal{C}$, then $\hat{G} = UG$ for some $U \in Gl_k(\mathbb{F})$ and $\hat{H} = HV$ for some $V \in Gl_{n-k}(\mathbb{F})$.

If $\mathcal{C} \subseteq \mathbb{F}$ is an (n,k)-block code with generator matrix G and parity check matrix H, then $\mathcal{C}^\perp := \{v \in \mathbb{F}^n \mid \forall\, w \in \mathcal{C} \;:\; vw^{\mathrm{t}} = 0\} = \ker_{\mathbb{F}} G^{\mathrm{t}} = \operatorname{im}_{\mathbb{F}} H^{\mathrm{t}}$ is a uniquely determined $(n, n-k)$-block code (see e.g. [MS78, Ch. 1, §8]). It is called the **dual code** of $\mathcal{C}$ and satisfies $(\mathcal{C}^\perp)^\perp = \mathcal{C}$.

Finally, we give a result which allows us to determine the distance of a block code with the aid of a parity check matrix. This result will be used later in the proof of Theorem 5.22.

Lemma 1.1 *cf. [LC83, Theorem 3.2 and Corollaries 3.2.1, 3.2.2]*
Let $\mathcal{C}$ be an (n,k)-block code with parity check matrix $H \in \mathbb{F}^{n\times(n-k)}$. Then $d_{free}(\mathcal{C}) = d$ if and only if there exist d $\mathbb{F}$-linear dependent rows in H and any selection of $d-1$ rows of H is $\mathbb{F}$-linear independent. □

1.2 Submodules of $\mathbb{F}[z]^n$

A submodule $\mathcal{C}$ of the $\mathbb{F}[z]$-module $\mathbb{F}[z]^n$ of rank k is said to be an (n,k)-**submodule over** $\mathbb{F}$. The **weight** of a vector $v = \sum_{i\geq 0} v_i z^i \in \mathbb{F}[z]^n$ is defined as $\sum_{i\geq 0} \operatorname{wt}(v_i)$, where $\operatorname{wt}(v_i)$ is the Hamming weight of the vector $v_i \in \mathbb{F}^n$. It is also denoted by $\operatorname{wt}(v)$. The **free distance** $d_{free}(\mathcal{C}) := \min\{wt(v) \mid 0 \neq v \in \mathcal{C}\}$ of a submodule is a canonical generalisation of the free distance of a block code. As for block codes a codeword can be recovered completely if it was changed in at most $\lfloor \frac{1}{2}(d_{free}(\mathcal{C})-1) \rfloor$ components (at least in theory).

The free distance of submodules (resp. convolutional codes, see Definition 1.6) will play a minor role in this thesis. However, we will often give the free distance of the convolutional codes in our examples in order to have a more complete picture of the codes in question.

Since every (n,k)-submodule $\mathcal{C} \subseteq \mathbb{F}[z]^n$ is free, it can be written as $\mathcal{C} = \operatorname{im}_{\mathbb{F}[z]} G$ for some $G \in \mathbb{F}[z]^{k\times n}$. Such a matrix G is called a **generator matrix** of $\mathcal{C}$ and has full rank. (Here we put $\{0_n\} =: \operatorname{im}_{\mathbb{F}[z]} G$ for $G \in \mathbb{F}[z]^{0\times n}$. Furthermore, $G \in \mathbb{F}[z]^{0\times n}$ is defined to be right invertible.) A generator matrix of $\mathcal{C}$ is unique in the following sense: If G, $\hat{G}$ are two generator matrices of an (n,k)-submodule, then $\hat{G} = UG$ for some unimodular matrix $U \in \mathbb{F}[z]^{k\times k}$. Since any unimodular matrix is a product of elementary matrices (corresponding to so-called elementary operations) [2], we can also express uniqueness of a generator matrix in other terms:

Remark 1.2
Consider two generator matrices $G, \hat{G}$ of an (n,k)-submodule $\mathcal{C} \subseteq \mathbb{F}[z]^n$ and let $G_1, \ldots, G_k$ denote the rows of G. Then G can be transformed successively to $\hat{G}$ by a finite number of so called **elementary operations** *over $\mathbb{F}[z]$, these are:*

(O1) *Rescaling a row of G by a factor $\lambda \in \mathbb{F} \setminus \{0\}$;*

(O2) *exchanging two rows of G;*

(O3) *replacing G_i by $G_i + f \cdot G_j$ where $j \neq i$ and $f \in \mathbb{F}[z] \setminus \{0\}$.*

The maximal z-degree of the k-minors of a generator matrix G of an (n,k)-submodule $\mathcal{C}$ is independent of the choice of G and is called the **complexity** of $\mathcal{C}$. If δ is the complexity of $\mathcal{C}$, this submodule is also called an (n,k,δ)**-submodule**.

Rosenthal and Smarandache showed in [RS99] that the free distance of an (n,k,δ)-submodule satisfies

$$d_{free}\,\mathcal{C} \leq (n-k)(\lfloor \delta/k \rfloor + 1) + \delta + 1 \quad . \tag{1.1}$$

This bound is a generalisation for the MDS-bound in the case of block codes and is also called **MDS-bound**. A convolutional code reaching this bound is called **MDS-convolutional code**.

The z**-degree** or simply **degree** of a generator matrix $G \in \mathbb{F}[z]^{k\times n}$ of a submodule $\mathcal{C}$ is defined as the sum of the z-degrees of its rows (which are identified as elements in $\mathbb{F}^n[z]$) and it is denoted by $\deg G$. A generator matrix with minimal degree is called a **minimal basis**.

[2] For the definition of elementary (unimodular) matrices we refer to [Gan60a, Ch. VI, §1]. The statement that any unimodular matrix is a product of elementary matrices can be found in [Gan60a, Ch. VI, §2, 2., Corollary].

Forney showed how the complexity of a submodule, its minimal bases and the (row-)degrees of generator matrices are related. Before we state his fundamental result, we recall another common definition: For a matrix $M \in \mathbb{F}[z]^{l\times n}$ ($l \in \mathbb{N}$) with rows $M_1, \ldots, M_l$ and $M_i \neq 0$ for $1 \leq i \leq l$ the leading z-coefficient vector of M_i is denoted by $\mathrm{lc}_z M_i \in \mathbb{F}^n$ and the matrix

$$L(M) := \begin{bmatrix} \mathrm{lc}_z M_1 \\ \mathrm{lc}_z M_2 \\ \vdots \\ \mathrm{lc}_z M_l \end{bmatrix}$$

is called the **leading coefficient matrix of** M.

Theorem 1.3 (Theorem of Forney) *[For75, 3.]*
Let $\mathcal{C} \subseteq \mathbb{F}[z]^n$ be an (n,k,δ)-submodule and let $G \in \mathbb{F}[z]^{k\times n}$ be a generator matrix with row degrees $\nu_1 \ldots, \nu_k$. Then the following statements are equivalent:
(i) G is a minimal basis;
(ii) $\mathrm{rank}_{\mathbb{F}} L(G) = k$;
(iii) $\delta = \deg G$, *i.e.* $\delta = \nu_1 + \ldots + \nu_k$;
(iv) For all $0 \neq (u_1, \ldots, u_k) \in \mathbb{F}[z]^k$ *one has* $\deg_z uG = \max\limits_{1\leq j\leq k}(\deg_z u_j + \nu_j)$;
(v) For all $d \in \mathbb{N}_0$ *one has* $\dim_{\mathbb{F}}\{w \in \mathrm{im}_{\mathbb{F}[z]} G \mid \deg_z w < d\} = \sum\limits_{\substack{j=1 \\ \nu_j \leq d}}^{k} (d - \nu_j)$.

□

Statement (iii) of the Theorem of Forney implies that a submodule having complexity 0 has a constant generator matrix. Therefore such a submodule is said to be a block code.

It is a consequence of Theorem 1.3 (v) that the row degrees of two minimal basis of an (n,k)-submodule $\mathcal{C}$ are equal up to permutation. Therefore, the row-degrees of a minimal basis of $\mathcal{C}$ are called the **Forney indices** of $\mathcal{C}$. In Section 2 we will use the following more general result:

Lemma 1.4
Let $\mathcal{C} \subseteq \mathbb{F}[z]^n$ be an (n,k)-submodule and consider a matrix $G \in \mathbb{F}[z]^{l\times n}$ with rows $G_1, \ldots, G_l$ such that $\mathrm{im}_{\mathbb{F}[z]} G = \mathcal{C}$ (we allow $l > k$) and $G_i \neq 0$ for $1 \leq i \leq l$. Let $\mu_1 \leq \ldots \leq \mu_l$ denote the row degrees of G. Furthermore, let $\hat{G} \in \mathbb{F}[z]^{k\times n}$ be a minimal basis of $\mathcal{C}$ with row degrees $\nu_1 \leq \ldots \leq \nu_k$.

(a) If $G_{j_1}, \ldots, G_{j_k}$ is a selection of $\mathbb{F}[z]$-linear independent rows of G where $j_1 < \ldots < j_k$, then

$$\nu_1 \leq \mu_{j_1} \ , \ \ldots \ , \ \nu_k \leq \mu_{j_k} \ .$$

(b) If $\mathcal{C}$ has complexity δ, then $\delta \leq \mu_1 + \ldots + \mu_l$.

PROOF: Statement (b) is a direct consequence of (a) and Theorem 1.3 (iii). For (a), we assume that $\mu_{j_1} \leq \ldots \leq \mu_{j_s} < \nu_s$ for some $s \in \{1, \ldots, k\}$. We know that $G_{j_1}, \ldots, G_{j_k} \in \text{span}_{\mathbb{F}[z]}\{\hat{G}_1 \ldots, \hat{G}_k\}$. Theorem 1.3 (iv) implies that $G_{j_1}, \ldots, G_{j_s} \in \text{span}_{\mathbb{F}[z]}\{\hat{G}_1 \ldots, \hat{G}_{s-1}\}$. But $\text{span}_{\mathbb{F}[z]}\{\hat{G}_1 \ldots, \hat{G}_{s-1}\}$ has rank at most $s-1$ and therefore the rows $G_{j_1}, \ldots, G_{j_s}$ are $\mathbb{F}[z]$-linear dependent. The latter implies that the rows $G_{j_1}, \ldots, G_{j_k}$ are $\mathbb{F}[z]$-linear dependent. This is a contradiction, thus $\mu_{j_s} \geq \nu_s$. □

We close this brief overview with a more specialized result, which will be crucial in Section 2. It states that the elementary operations in Remark 1.2 can be specified when dealing with minimal bases:

Proposition 1.5
Let $G, \hat{G} \in \mathbb{F}[z]^{k \times n}$ be two minimal bases of the (n,k)-submodule $\mathcal{C} \subseteq \mathbb{F}[z]^n$ with Forney indices $\nu_1, \ldots, \nu_k$. Without loss of generality we assume that both the row degrees of G and of $\hat{G}$ are given by $\nu_1 \leq \ldots \leq \nu_k$. Let $G_1, \ldots, G_k$ denote the rows of G. Then G can be transformed successively to $\hat{G}$ by a finite number of operations of the type
(O1) *Rescaling a row of G by a factor $\lambda \in \mathbb{F} \setminus \{0\}$;*
(O2') *exchanging two rows of G which have the same degree;*
(O3') *replacing G_i by $G_i + z^l \cdot G_j$, where $j \neq i$, $\nu_j \leq \nu_i$ and $0 \leq l \leq \nu_i - \nu_j$.*
Note that each of these transformations preserves the property "minimal bases" (which can be seen directly).

Outline of the PROOF: We know that $\hat{G} = UG$ for some $U = (u_{i,j})_{1 \leq i,j \leq k} \in Gl_k(\mathbb{F}[z])$. Theorem 2 in [MP79] states that UG is a minimal basis if and only if $\deg_z u_{ij} \leq \nu_i - \nu_j$ for $\nu_j \leq \nu_i$ and $u_{ij} = 0$ for $\nu_j > \nu_i$. (In particular U is block-lower-triangular.) Now there exists a suitable decomposition of U into a product of elementary matrices [3] which correspond to the operations (O1), (O2') and (O3'). If we transform a minimal basis with one of the operations (O1), (O2') or (O3'), then the resulting matrix has row degrees $\nu_1 \leq \ldots \leq \nu_k$. It is a minimal basis again, as we can readily see with Theorem 1.3. □

[3] See footnote 2 on page 11.

1.3 Convolutional Codes

Block codes are subspaces of $\mathbb{F}^n$ or, equivalently, direct summands of $\mathbb{F}^n$. Therefore, there are at least two natural ways to define convolutional codes as a generalization of block codes: Convolutional codes can be defined either to be submodules of $\mathbb{F}[z]^n$ or to be direct summands of $\mathbb{F}[z]^n$. We choose the second way because it turns out that direct summands of $\mathbb{F}[z]^n$ have some useful properties.

Definition 1.6
*A direct summand $\mathcal{C}$ of $\mathbb{F}[z]^n$ (as an $\mathbb{F}[z]$-module) of rank k having complexity δ is called an (n,k)-***convolutional code*** *or an (n,k,δ)-***convolutional code***. An element of $\mathbb{F}[z]^k$ is called a* **message word** *and an element of $\mathcal{C}$ is called a* **code word**.

The follwing theorem shows that it is quite advantageous to define convolutional codes this way. Before we state it, we give two other definitions: Some (n,k)-submodules $\mathcal{C}$ (but not all of them) can be written as $\mathcal{C} = \ker_{\mathbb{F}[z]} H$ for some $H \in \mathbb{F}[z]^{n\times(n-k)}$ (cf. Theorem 1.7). If this is possible, H is called a **parity check matrix** of $\mathcal{C}$. (We say that $H \in \mathbb{F}[z]^{n\times 0}$ is a parity check matrix of the (n,n)-convolutional code $\mathbb{F}[z]^n$ and we define H to be left invertible.) If a Smith form[4] of $G \in \mathbb{F}[z]^{k\times n}$, $k \leq n$, is $[I_k \,|\, 0_{k\times(n-k)}]$, then G is called **basic**.

Theorem 1.7 *cf. [For75, 9.], [McE98, Theorem A.1]*
Let $\mathcal{C} \subseteq \mathbb{F}[z]^n$ be an (n,k)-submodule with generator matrix $G \in \mathbb{F}[z]^{k\times n}$ and let $\overline{\mathbb{F}}$ denote an algebraic closure of $\mathbb{F}$. Then the following statements are equivalent:

(i) *$\mathcal{C}$ is a convolutional code;*

(ii) *$\mathcal{C}$ has a parity check matrix, i.e. $\mathcal{C} = \ker_{\mathbb{F}[z]} H$ for some $H \in \mathbb{F}[z]^{n\times(n-k)}$;*

(iii) *G is basic, i.e. $[I_k \,|\, 0_{k\times(n-k)}]$ is a Smith-form of G;*

(iv) $\operatorname{rank} G(a) = k$ *for all $a \in \overline{\mathbb{F}}$;*

(v) *there exists $\hat{G} \in \mathbb{F}[z]^{(n-k)\times n}$ such that $\left[\begin{smallmatrix} G \\ \hat{G} \end{smallmatrix}\right] \in Gl_n(\mathbb{F}[z])$, i.e. every $\mathbb{F}[z]$-basis of $\mathcal{C}$ can be completed to an $\mathbb{F}[z]$-basis of $\mathbb{F}[z]^n$;*

(vi) *for all $v \in \mathbb{F}[z]^n$ and all polynomials $f \in \mathbb{F}[z] \setminus \{0\}$ one has*

[4] For the definition of a Smith-form we refer to [Gan60a, Chapter VI,§3]. The word "Smith-form" does not appear there. We call a "canonical diagonal matrix" as in [Gan60a, Thm. VI, §3, p.141] a Smith-form.

$$f v \in \mathcal{C} \implies v \in \mathcal{C} \ ;$$

(vii) for all submodules $\hat{\mathcal{C}}$ of rank k one has

$$\mathcal{C} \subseteq \hat{\mathcal{C}} \implies \mathcal{C} = \hat{\mathcal{C}} \ ;$$

(viii) G is right invertible over $\mathbb{F}[z]$, i.e. there exists $W \in \mathbb{F}[z]^{n\times k}$ such that

$$GW = I_k \ ;$$

(ix) $\operatorname{im}_{\mathbb{F}(z)} G \cap \mathbb{F}[z]^n = \operatorname{im}_{\mathbb{F}[z]} G$.

We will give a PROOF of this theorem (or at least some detailed hints), since the reader will possibly have difficulties to puzzle out a complete proof from the literature. We will organize the proof as follows:

$$\begin{array}{ccccc} & & & & (\text{viii}) \\ & & & & \Updownarrow \\ (\text{i}) \Leftrightarrow (\text{v}) & & \Longleftarrow & & (\text{iii}) \Leftrightarrow (\text{iv}) \\ \Downarrow & & & & \Uparrow \\ (\text{ii}) \Rightarrow (\text{vi}) & & \Rightarrow & & (\text{vii}) \\ & & \Updownarrow & & \\ & & (\text{ix}) & & \end{array}$$

Part "(i)⇔(v)" is obvious.

Part "(v)⇒(ii)" can be established as follows: Define $\left[\begin{smallmatrix} G \\ \hat{G} \end{smallmatrix}\right]^{-1} =: [\hat{H} \mid H] \in \mathbb{F}[z]^{n\times n}$ where $\hat{H}$ has k columns and H has $n-k$ columns. Now we show that the matrix $H \in \mathbb{F}[z]^{n\times(n-k)}$ is a parity check matrix of $\mathcal{C}$: Because of $[\hat{H} \mid H] \cdot \left[\begin{smallmatrix} G \\ \hat{G} \end{smallmatrix}\right] = \left[\begin{smallmatrix} G \\ \hat{G} \end{smallmatrix}\right] \cdot [\hat{H} \mid H] = I_n$ we have $GH = 0$ and $\operatorname{im}_{\mathbb{F}[z]} G \subseteq \ker_{\mathbb{F}[z]} H$. It remains to show that $\ker_{\mathbb{F}[z]} H \subseteq \operatorname{im}_{\mathbb{F}[z]} G$. Consider $v \in \mathbb{F}[z]^n$ such that $vH = 0$ and define $v[\hat{H} \mid H] =: (a, 0)$. Then on the one hand $v\big([\hat{H} \mid H] \cdot \left[\begin{smallmatrix} G \\ \hat{G} \end{smallmatrix}\right]\big) = vI_n = v$ and on the other hand $\big(v[\hat{H} \mid H]\big) \cdot \left[\begin{smallmatrix} G \\ \hat{G} \end{smallmatrix}\right] = (a, 0)\left[\begin{smallmatrix} G \\ \hat{G} \end{smallmatrix}\right] = aG$. Thus $v = aG \in \operatorname{im}_{\mathbb{F}[z]} G$.

For "(ii)⇒(vi)" we assume that H is a parity check matrix of $\mathcal{C}$. If $fv \in \ker_{\mathbb{F}[z]} H = \mathcal{C}$ for $v \in \mathbb{F}[z]^n$ and $f \in \mathbb{F}[z] \setminus \{0\}$, then $v \in \ker_{\mathbb{F}[z]} H = \mathcal{C}$, too.

Part "(vi)⇒(vii)" can be shown indirectly: Let $\hat{\mathcal{C}} \subseteq \mathbb{F}[z]^n$ be a submodule of rank k with generator matrix $\hat{G}$ and $\mathcal{C} \subseteq \hat{\mathcal{C}}$ but $\mathcal{C} \neq \hat{\mathcal{C}}$. Then there exists $v \in \hat{\mathcal{C}}$ such that $v \notin \mathcal{C}$. Since $\operatorname{im}_{\mathbb{F}(z)} G = \operatorname{im}_{\mathbb{F}(z)} \hat{G}$ there exists $u \in \mathbb{F}(z)^k$ such that $uG = v$. There exists some polynomial $f \in \mathbb{F}[z] \setminus \{0\}$ such that $fu \in \mathbb{F}[z]^k$, hence $fv \in \operatorname{im}_{\mathbb{F}[z]} G = \mathcal{C}$ but $v \notin \mathcal{C}$.

For the "(vi)⇒(ix)"-part we observe that $\operatorname{im}_{\mathbb{F}(z)} G \cap \mathbb{F}[z]^n \supseteq \mathcal{C}$ is true in any case. It remains to show $\operatorname{im}_{\mathbb{F}(z)} G \cap \mathbb{F}[z]^n \subseteq \mathcal{C}$: An element in $\operatorname{im}_{\mathbb{F}(z)} G \cap$

$\mathbb{F}[z]^n$ can be written as $\frac{1}{f}uG$ for some $f \in \mathbb{F}[z] \setminus \{0\}$ and $u \in \mathbb{F}[z]^k$. Now $f\frac{1}{f}uG = uG \in \mathcal{C}$. With (vi) we also have $\frac{1}{f}uG \in \mathcal{C}$.

Now we prove "(ix)⇒(vi)": Consider $v \in \mathbb{F}[z]^n$ and $f \in \mathbb{F}[z] \setminus \{0\}$ such that $fv \in \mathcal{C}$. Then $\frac{1}{f}fv = v \in \mathrm{im}_{\mathbb{F}(z)}G \cap \mathbb{F}[z]^n = \mathrm{im}_{\mathbb{F}[z]}G = \mathcal{C}$.

For the remaining part of the proof we use a Smith-form of G. We consider fixed unimodular matrices $U \in Gl_n(\mathbb{F}[z]), V \in Gl_k(\mathbb{F}[z])$ such that VGU is a Smith-form of G. In particular, we have

$$VGU = [\Delta \,|\, 0_{k\times(n-k)}] \quad \text{where} \quad \Delta = \mathrm{diag}\,(d_1, \ldots, d_k) \text{ and } d_i \in \mathbb{F}[z] \quad .$$

Observe that (iii) is equivalent to $\Delta \in Gl_k(\mathbb{F}[z])$.

For "(iii)⇒(viii)" we have $\Delta = I_k$ and we consider the matrix $W := U\left[\begin{smallmatrix} I_k \\ 0_{(n-k)\times k} \end{smallmatrix}\right]V \in \mathbb{F}[z]^{n\times k}$. It can be readily shown that W is a right inverse of G.

The inverse implication "(viii)⇒(iii)" can be established as follows: Let $W \in \mathbb{F}[z]^{n\times k}$ be a right inverse of G and let $M_1 \in \mathbb{F}[z]^{k\times k}$ denote the matrix consisting of the first k rows of $U^{-1}M$. We show that $\Delta \in Gl_k(\mathbb{F}[z])$. We calculate $I_k = (GU)(U^{-1}M) = (V^{-1}[\Delta \,|\, 0])(U^{-1}M) = V^{-1}\Delta M_1$. Hence we have $1 = \det(V^{-1}\Delta M_1) = \det(V^{-1})\det(\Delta)\det(M_1)$. In particular we have $\det(\Delta) \in \mathbb{F} \setminus \{0\}$, which implies that Δ is unimodular.

For "(iii)⇔(iv)" we observe that $\mathrm{rank}\, G(a) = k$ for all $a \in \overline{\mathbb{F}}$ if and only if $\mathrm{rank}\, \Delta(a) = k$ for all $a \in \overline{\mathbb{F}}$. But this is the case if and only if $\Delta \in Gl_k(\mathbb{F}[z])$.

"(vii)⇒(iii)" can be shown indirectly: Note that $[\Delta \,|\, 0_{k\times(n-k)}]U^{-1}$ is a generator matrix of $\mathcal{C}$ and that $\mathrm{im}_{\mathbb{F}[z]}[\Delta \,|\, 0_{k\times(n-k)}]U^{-1} \subseteq \mathrm{im}_{\mathbb{F}[z]}[I_k \,|\, 0_{k\times(n-k)}]U^{-1} =: \hat{\mathcal{C}}$. If $\Delta \notin Gl_k(\mathbb{F}[z])$, then $\mathcal{C} \neq \hat{\mathcal{C}}$.

Finally we prove "(iii)⇒(v)": With $\hat{G} := [0_{k\times k} \,|\, I_{n-k}]U^{-1}$ we get

$$\begin{bmatrix} G \\ \hat{G} \end{bmatrix} = \begin{bmatrix} V^{-1}[I_k \,|\, 0_{k\times(n-k)}]U^{-1} \\ [0_{k\times k} \,|\, I_{n-k}]U^{-1} \end{bmatrix} = \begin{bmatrix} V^{-1} & 0 \\ 0 & I_{n-k} \end{bmatrix} U^{-1} \in Gl_n(\mathbb{F}[z]) \quad .$$

□

Now we turn to the notion of dual codes in the context of convolutional codes. If $\mathcal{C}$ is an (n,k)-convolutional code with generator matrix $G \in \mathbb{F}[z]^{k\times n}$, then it has a parity check matrix $H \in \mathbb{F}[z]^{n\times(n-k)}$ which can be constructed as in the proof of Theorem 1.7, part "(v)⇒(ii)". In this construction a parity check matrix H consists of some columns of a unimodular matrix. This implies that H is left invertible and we conclude:

Corollary 1.8
An (n,k)-submodule $\mathcal{C} \subseteq \mathbb{F}[z]^n$ has a parity check matrix if and only if it has a left invertible parity check matrix. □

It can be readily shown that two left invertible parity check matrices $H, \hat{H} \in \mathbb{F}[z]^{n\times(n-k)}$ of an (n,k)-convolutional code $\mathcal{C}$ differ by a unimodular matrix $U \in Gl_{n-k}(\mathbb{F}[z])$. Precisely: There exists $U \in Gl_{n-k}(\mathbb{F}[z])$ such that $\hat{H} = H \cdot U$. As a consequence, the $(n, n-k)$-convolutional codes $\operatorname{im}_{\mathbb{F}[z]} H^{\mathrm{t}}$ and $\operatorname{im}_{\mathbb{F}[z]} \hat{H}^{\mathrm{t}} = \operatorname{im}_{\mathbb{F}[z]}(U^{\mathrm{t}} H^{\mathrm{t}})$ are the same. If we want to determine the complexity of this "new" code, we have to determine the minors of H. The following statement is very helpful:

Lemma 1.9 *cf. [For75, 6., Theorem 3]*
If $G \in \mathbb{F}[z]^{k\times n}$ is a generator matrix of an (n,k)-convolutional code $\mathcal{C}$ and if $H \in \mathbb{F}[z]^{n\times(n-k)}$ is a left invertible parity check matrix, then the k-minors of G and the $(n-k)$-minors of H are the same up to permutation and multiplication with constants in $\mathbb{F}\setminus\{0\}$. In particular, the convolutional codes $\mathcal{C} = \operatorname{im}_{\mathbb{F}[z]} G$ and $\operatorname{im}_{\mathbb{F}[z]} H^{\mathrm{t}}$ have the same complexity. □

The following theorem and definition summarizes and extends the recent remarks:

Theorem and Definition 1.10 *[For75, 6.]*
If $\mathcal{C} \subseteq \mathbb{F}[z]^n$ is an (n,k,δ)-convolutional code and with generator matrix G and if H is a left invertible parity check matrix of $\mathcal{C}$, then

$$C^{\perp} := \{v \in \mathbb{F}[z]^n \mid \forall\ w \in \mathcal{C}\ :\ vw^{\mathrm{t}} = 0\} = \operatorname{im}_{\mathbb{F}[z]} H^{\mathrm{t}} = \ker_{\mathbb{F}[z]} G^{\mathrm{t}}$$

is called the **dual code** *of $\mathcal{C}$. It is an $(n, n-k, \delta)$-convolutional code and satisfies $(\mathcal{C}^{\perp})^{\perp} = \mathcal{C}$.* □

2 First Order Representations of Submodules of $\mathbb{F}[z]^n$

In this section we will investigate how an (n,k,δ)-submodule $\mathcal{C} \subseteq \mathbb{F}[z]^n$ can be represented by a constant matrix triple $(P,Q,R) \in \mathbb{F}^{(\delta+k)\times\delta} \times \mathbb{F}^{(\delta+k)\times\delta} \times \mathbb{F}^{(\delta+k)\times n}$ resp. $(K,L,M) \in \mathbb{F}^{\delta\times(\delta+n-k)} \times \mathbb{F}^{\delta\times(\delta+n-k)} \times \mathbb{F}^{n\times(\delta+n-k)}$ via the equation

$$\mathcal{C} = \big(\ker_{\mathbb{F}[z]}(zP+Q)\big)\cdot R \text{ resp. } \mathcal{C} = \Big(\ker_{\mathbb{F}[z]} \begin{bmatrix} zK+L \\ M \end{bmatrix}\Big)\cdot \begin{bmatrix} 0_{\delta\times n} \\ I_n \end{bmatrix}. \quad (2.1)$$

Such representations are called **first order representations** because the indeterminate z appears only linearly.

The main target of this section is to prove three theorems – roughly speaking these are:

– Any (n,k,δ)-submodule (resp. convolutional code) can be represented as in (2.1) and the matrix triples (P,Q,R), (K,L,M) have some further "nice properties" (cf. Theorems 2.12, 2.22).

– Every matrix triple (P,Q,R) (resp. (K,L,M)) of sizes as above and having these "nice properties" defines an (n,k,δ)-submodule or even a convolutional code (cf. Theorems 2.13, 2.23).

– A representation (P,Q,R) (resp. (K,L,M)) of a submodule satisfying the "nice properties" is unique up to some matrix manipulations (cf. Theorems 2.15, 2.24).

These results, which will be specified in the following, are well known (see the book of Kuijper [Kui94, Sections 3.4, 4.1, 4.4 and 5] or the articles [RSY96], [SGR98]). Since the tree statements can be formulated in the terminology of linear algebra and matrix theory, one might want to have proofs in this terminology, too. Efforts have been made to give purely (linear) algebraic proofs, for example by Glüsing-Lüerßen. She was able to prove the first two statements in this way (cf. [Glub]), but yet, no pure (linear) algebraic proofs of all three theorems were known. We succeeded to fill this gap and for this reason we publish our results in detail in this thesis, even though we use just one result from this section (namely Theorem 2.12) later on.

We give some technicalities in Section 2.1, then we will formulate the three theorems for each representation and give the proofs in Sections 2.2 resp. 2.3.

2.1 Some Information on Matrix Pencils

In (2.1) there appear matrices of the shape $zP + Q$ where P and Q have constant entries. Such a matrix $zP + Q$ is called a **matrix pencil**. We will use a "normal form" of a matrix pencil to deal with first order representations. Before we state a main result of this section in Theorem 2.2 we introduce some shorthand expressions in the following notation:

Notation 2.1
For $\nu \in \mathbb{N}_0$ *we write*

$$L_\nu := \begin{bmatrix} z & & \\ -1 & \ddots & \\ & \ddots & z \\ & & -1 \end{bmatrix} \in \mathbb{F}[z]^{(\nu+1)\times\nu} \quad , \quad M_\nu := L_\nu^{\mathrm{t}} \in \mathbb{F}[z]^{\nu\times(\nu+1)}$$

and for $\nu_1, \ldots, \nu_s \in \mathbb{N}_0$ *we write*

$$\begin{aligned} & \mathcal{L}(\nu_1, \ldots, \nu_s) := \operatorname{diag}(L_{\nu_1}, \ldots, L_{\nu_s}) \in \mathbb{F}[z]^{(\sum_{i=1}^s \nu_i + s)\times(\sum_{i=1}^s \nu_i)} \\ \textit{and} \quad & \mathcal{M}(\nu_1, \ldots, \nu_s) := \operatorname{diag}(M_{\nu_1}, \ldots, M_{\nu_s}) \in \mathbb{F}[z]^{(\sum_{i=1}^s \nu_i)\times(\sum_{i=1}^s \nu_i + s)} \quad . \end{aligned}$$

Note: If L_0 *occurs in* $\mathcal{L}(\nu_1, \ldots, \nu_s)$ *resp. if* M_0 *occurs in* $\mathcal{M}(\nu_1, \ldots, \nu_s)$*, then it induces a zero-row in* $\mathcal{L}(\nu_1, \ldots, \nu_s)$ *resp. a zero-column in* $\mathcal{M}(\nu_1, \ldots, \nu_s)$*.*

Theorem and Definition 2.2 *[Gan60b, Chapter XII]*[5]
Let $P, Q \in \mathbb{F}^{l\times m}$*. Then there exist matrices* $T \in Gl_l(\mathbb{F})$ *and* $S \in Gl_m(\mathbb{F})$ *and uniquely determined integers* $0 \le \nu_1 \le \ldots \le \nu_s$*,* $0 \le \mu_1 \le \ldots \le \mu_t$ *and* $\nu_0, \mu_0 \in \mathbb{N}_0$ *such that*

$$T(zP + Q)S = \operatorname{diag}(\mathcal{L}(\nu_1, \ldots, \nu_s), \mathcal{M}(\mu_1, \ldots, \mu_t), zI_{\nu_0} + \hat{A}, I_{\mu_0} + z\tilde{A}) \quad (2.2)$$

where $zI_{\nu_0} + \hat{A}$ *and* $I_{\mu_0} + z\tilde{A}$ *are (regular) matrix pencils. Furthermore,* $\hat{A}$ *and* $\tilde{A}$ *are constant matrices of appropriate sizes such that* $\tilde{A}$ *is nilpotent*[6]*. It is allowed that one or more blocks do not appear in (2.2).*

[5]The normal form of $zP+Q$ given in Theorem 2.2 is not exactly the normal form as in [Gan60b]. There the blocks L_{ν_i} and M_{ν_i} have "1"-entries instead of "-1"-entries and the zero-rows resp. zero-columns are placed at the top resp. at the left side of the matrix. But of course we can deduce Theorem 2.2 from [Gan60b, Chapter XII].Observe that zero-rows resp. zero-columns are represented by matrices of the form L_0 resp. M_0.

[6]This means $\tilde{A}^r = 0$ for some $r \in \mathbb{N}$.

The pencil $T(zP+Q)S$ in (2.2) is called a **canonical form** *of $zP+Q$. In general, it is not unique (because in our setting the choice of $\hat{A}$, $\tilde{A}$ is not unique), but it is unique if the regular submatrices $zI_{\nu_0}+\hat{A}$ and $I_{\mu_0}+z\tilde{A}$ are absent. In this case we can speak of the canonical form of $zP+Q$.* □

Example 2.3
Let $\mathbb{F}=\mathbb{F}_3$. An example for a canonical form is
$\operatorname{diag}(\mathcal{L}(0,1,2),\mathcal{M}(0,1),[z+2],\begin{bmatrix}1&z\\0&1\end{bmatrix}) =$

$$\left[\begin{array}{c|cc|c|cc|c|cc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline
z & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline
0 & z & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & z & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline
0 & 0 & 0 & 0 & z & -1 & 0 & 0 & 0 \\ \hline
0 & 0 & 0 & 0 & 0 & 0 & z+2 & 0 & 0 \\ \hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & z \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{array}\right] .$$

For later purposes we must know under which conditions the blocks $\mathcal{M}(\mu_1,\ldots,\mu_t)$ and the regular block $\operatorname{diag}(zI_{\nu_0}+\hat{A}, I_{\mu_0}+z\tilde{A})$ is absent in a canonical form of $zP+Q$.

Lemma 2.4
Let $P,Q\in\mathbb{F}^{l\times m}$. Then the following statements are equivalent:

(i) The canonical form of $zP+Q$ is $\mathcal{L}(\nu_1,\ldots,\nu_s)$ for some $\nu_1\le\ldots\le\nu_s$ in $\mathbb{N}_0$;

(ii) The matrix P has full column rank and the pencil $zP+Q$ is left invertible.

PROOF: A transformation of $zP+Q$ to $T(zP+Q)S$ via $T\in Gl_l(\mathbb{F})$ and $S\in Gl_m(\mathbb{F})$ does not affect the properties given in (ii). Therefore we may assume without loss of generality that $zP+Q$ is given in a canonical form, i. e.

$$zP+Q=\operatorname{diag}(\mathcal{L}(\nu_1,\ldots,\nu_s),\mathcal{M}(\mu_1,\ldots,\mu_t),zI_{\nu_0}+\hat{A},I_{\mu_0}+z\tilde{A}) ,$$

where $\tilde{A}$ is nilpotent. Now (i)⇒(ii) is obvious (cf. Theorem 1.7 (iv)). For (ii)⇒(i) we argue as follows: If P has full column rank, there can be no block of $\mathcal{M}(\mu_1 \ldots, \mu_t)$-type, because it would induce zero-columns in P. There can be no block of the form $I_{\mu_0} + z\tilde{A}$ either, because $\tilde{A}$ is not invertible (since it is nilpotent) and this would cause that P has not full column rank. Moreover, left invertibility of $zP+Q$ implies that there is no block of the form $zI_{\nu_0} + \hat{A}$, which can be obtained as follows: Assume that the block $zI_{\nu_0} + \hat{A}$ is present (this means $\nu_0 > 0$) and let $\overline{\mathbb{F}}$ denote an algebraic closure of $\mathbb{F}$. With Theorem 1.7 (iv) it is clear that left invertibility of $zP + Q$ implies left invertibility of $zI_{\nu_0} + \hat{A}$. But $\det(zI_{\nu_0} + \hat{A})$ is a polynomial in z of degree $\nu_0 > 0$. Hence $aI_{\nu_0} + \hat{A}$ has not full rank for every $a \in \overline{\mathbb{F}}$ which in turn implies that $zI_{\nu_0} + \hat{A}$ is not left invertible, as Theorem 1.7 tells us. □

2.2 PQR-Representations of Submodules of $\mathbb{F}[z]^n$

In this section we will investigate the so called "PQR-representation" of submodules of $\mathbb{F}[z]^n$. We will deal with the set $(\ker_{\mathbb{F}[z]}(zP + Q)) \cdot R$ which will be defined in the following notation.

Notation 2.5
Consider a matrix triple $(P, Q, R) \in \mathbb{F}^{(\delta+k)\times\delta} \times \mathbb{F}^{(\delta+k)\times\delta} \times \mathbb{F}^{(\delta+k)\times n}$ *where* $k \leq n$ *and* $\delta > 0$. *We will use the following abbreviation:*

$$\begin{aligned}
\mathcal{C}_{\mathrm{P}}(P,Q,R) &:= \left\{ v \in \mathbb{F}[z]^n \;\middle|\; \exists\, \zeta \in \mathbb{F}[z]^{\delta+k} \;:\; \begin{array}{rcl} \zeta(zP+Q) & = & 0 \\ \zeta R & = & v \end{array} \right\} \\
&= \left\{ v \in \mathbb{F}[z]^n \;\middle|\; (0_\delta, v) \in \operatorname{im}_{\mathbb{F}[z]}[zP + Q \mid R] \right\} \\
&= \big(\ker_{\mathbb{F}[z]}(zP+Q)\big) \cdot R \quad .
\end{aligned}$$

Of course, any matrix triple (P, Q, R) of the sizes as in Notation 2.5 gives rise to a submodule $\mathcal{C}_{\mathrm{P}}(P, Q, R) \subseteq \mathbb{F}[z]^n$.

Definition 2.6
If a submodule $\mathcal{C} \subseteq \mathbb{F}[z]^n$ *can be written as* $\mathcal{C} = \mathcal{C}_{\mathrm{P}}(P, Q, R)$ *with* (P, Q, R) *as in Notation 2.5, then* (P, Q, R) *is called* PQR**-representation** *of* $\mathcal{C}$.

Some properties of (P, Q, R) will be of frequent use and we will introduce shorthand expressions:

Properties 2.7
Consider (P, Q, R) as in Notation 2.5. Properties of interest are:
(P1) *P has full column rank δ;*
(P2) *$[P \mid R]$ has full row rank $\delta + k$;*
(P3) *$zP + Q$ is left invertible;*
(P4) *$[zP + Q \mid R]$ is right invertible.*

It is an obvious question how two matrix triples (P, Q, R) and $(\hat{P}, \hat{Q}, \hat{R})$ of sizes as in Notation 2.5 are related, if they satisfy $\mathcal{C}_{\mathrm{P}}(P, Q, R) = \mathcal{C}_{\mathrm{P}}(\hat{P}, \hat{Q}, \hat{R})$. This question is partly answered in the following remark, which can be proved straightforwardly:

Remark 2.8
Let (P, Q, R) be a matrix triple as in Notation 2.5 and let $T \in Gl_{\delta+k}(\mathbb{F})$ and $S \in Gl_{\delta}(\mathbb{F})$. Then we have $\mathcal{C}_{\mathrm{P}}(P, Q, R) = \mathcal{C}_{\mathrm{P}}(TPS, TQS, TR)$. Furthermore, the transformation

$$(P, Q, R) \mapsto (TPS, TQS, TR)$$

does not affect any of the properties (P1), (P2), (P3) *or* (P4), *precisely: If one of these properties is true for (P, Q, R), then the transformed triple (TPS, TQS, TR) inherits this property.*
As a consequence, we can assume without loss of generality that $zP + Q$ is given in a canonical form when we deal with the set $\mathcal{C}_{\mathrm{P}}(P, Q, R)$.

In fact, under some additional conditions $\mathcal{C}_{\mathrm{P}}(P, Q, R) = \mathcal{C}_{\mathrm{P}}(\hat{P}, \hat{Q}, \hat{R})$ is true if and only if $(\hat{P}, \hat{Q}, \hat{R})$ is the result of a transformation as in Remark 2.8 and we will specify and prove this statement in Theorem 2.15.

The next theorem tells us among other things how we can produce a matrix G satisfying $\operatorname{im}_{\mathbb{F}[z]} G = \mathcal{C}_{\mathrm{P}}(P, Q, R)$ for a given matrix triple (P, Q, R). It will be a necessary tool to prove Theorem 2.12.

Theorem 2.9
Let (P, Q, R) be a matrix triple as in Notation 2.5 and let $zP+Q$ be given in a canonical form (which we can assume without loss of generality, cf. Remark 2.8) with the data as in (2.2). Then the following statements hold:
(a) The matrix $zP + Q$ equals $\mathcal{L}(\nu_1, \ldots, \nu_s) \in \mathbb{F}[z]^{(\sum_{i=1}^{s} \nu_i + s) \times \sum_{i=1}^{s} \nu_i}$ if and only if (P1) *and* (P3) *are satisfied. In this case we have $s = k$ and $\delta = \sum_{i=1}^{s} \nu_i$.*

(b) Let $zP+Q=\mathcal{L}(\nu_1,\ldots,\nu_k)$ *and define*

$$Z(\nu_1,\ldots,\nu_k) := \left[\begin{array}{c|c|c} 1 \;\; z \;\; \cdots \;\; z^{\nu_1} & & \\ \hline & \ddots & \\ \hline & & 1 \;\; z \;\; \cdots \;\; z^{\nu_k} \end{array}\right] \in \mathbb{F}[z]^{k\times(\delta+k)}$$

and $G := Z(\nu_1,\ldots,\nu_k)R \in \mathbb{F}[z]^{k\times n}$. *Then we have*

$$Z(\nu_1,\ldots,\nu_k)[zP+Q \,|\, R] = [0_{k\times\delta} \,|\, G] \quad \textit{and} \quad \operatorname{im}_{\mathbb{F}[z]} G = \mathcal{C}_{\mathrm{P}}(P,Q,R) \quad .$$

We can identify R *as the "list of coefficient vectors" of* G, *the matrix* Z *"re-collects" the rows of* G *out of* R.
In particular we have $\deg G \le \delta$ *and* $\mathcal{C}_{\mathrm{P}}(P,Q,R)$ *has rank at most* k.

(c) Let $zP+Q=\mathcal{L}(\nu_1,\ldots,\nu_k)$. *Furthermore, let* G *be defined as in (b). Let* $G_1,\ldots,G_k$ *denote the rows of* G. *Accepting possible zero-coefficients, we may assume*

$$G_i = \sum_{j=0}^{\nu_i} g_{i,j} z^j \quad \textit{where} \quad g_{ij} \in \mathbb{F}^n \quad \textit{for} \quad 1 \le i \le k \quad .$$

We define

$$L'(G) := \begin{bmatrix} g_{1,\nu_1} \\ \vdots \\ g_{k,\nu_k} \end{bmatrix} \quad .$$

(In general we will not have $L'(G)=L(G)$.*) Then we have:*

$$\begin{array}{ll} & \text{(P2) } \textit{is satisfied} \\ \iff & \operatorname{rank} L'(G) = k \\ \iff & G \textit{ is a minimal basis with row degrees } \nu_1,\ldots,\nu_k \quad . \end{array}$$

(d) Let $zP+Q=\mathcal{L}(\nu_1,\ldots,\nu_s)$ *and let* G *be defined as in (b). Then we have:*

$$G \textit{ is right invertible} \iff \text{(P4) } \textit{is satisfied} \quad .$$

In particular, we have: If $zP+Q=\mathcal{L}(\nu_1,\ldots,\nu_s)$, *then* $\mathcal{C}_{\mathrm{P}}(P,Q,R)$ *is a convolutional code if and only if* (P4) *holds.*

PROOF: (The proof uses results from Glüsing-Lüerßen, cf. [Glub].)
Part (a) is Lemma 2.4.
For part (b) we write $Z := Z(\nu_1,\ldots,\nu_k)$, for short. It is easy to check that $Z[zP+Q \,|\, R] = [0_{k\times\delta} \,|\, G]$. Thus it remains to show that G is a generator matrix of $\mathcal{C}_{\mathrm{P}}(P,Q,R)$:

Proof of $\operatorname{im}_{\mathbb{F}[z]} G \subseteq \mathcal{C}_{\mathrm{P}}(P,Q,R)$: Consider $v \in \operatorname{im}_{\mathbb{F}[z]} G$ where $v = uG$, $u \in \mathbb{F}[z]^k$. With $Z[zP+Q \mid R] = [0_{k\times\delta} \mid G]$ we have $uZ[zP+Q \mid R] = (0_\delta, v)$ and therefore $(0_\delta, v) \in \operatorname{im}_{\mathbb{F}[z]}[zP+Q \mid R]$. This implies $v \in \mathcal{C}_{\mathrm{P}}(P,Q,R)$.

Proof of $\mathcal{C}_{\mathrm{P}}(P,Q,R) \subseteq \operatorname{im}_{\mathbb{F}[z]} G$: Let v be an element of $\mathcal{C}_{\mathrm{P}}(P,Q,R)$, i.e. $0 = \zeta[zP+Q]$ and $v = \zeta R$ for some $\zeta \in \mathbb{F}[z]^{\delta+k}$. We have to show that $v \in \operatorname{im}_{\mathbb{F}[z]} G$, i.e. that there exists a vector $u \in \mathbb{F}[z]^k$ such that $uG = v$. It is sufficient to show that there exists $u \in \mathbb{F}[z]^k$ such that $uZ = \zeta$ because this implies $v = \zeta R = uZR = uG$. Towards this end, we argue as follows:
With $Z(zP+Q) = 0_{k\times\delta}$ we have $\operatorname{im}_{\mathbb{F}(z)} Z \subseteq \ker_{\mathbb{F}(z)}(zP+Q)$. It is easy to see that both $\mathbb{F}(z)$-vectorspaces have dimension k, hence they are equal. Furthermore, Theorem 1.7 "(vi)$\Leftrightarrow$(ix)" along with right invertibility of Z (cf. Theorem 1.7 (iv)) yields $\operatorname{im}_{\mathbb{F}(z)} Z \cap \mathbb{F}[z]^n = \operatorname{im}_{\mathbb{F}[z]} Z$. With $\zeta \in \ker_{\mathbb{F}(z)}(zP+Q) \cap \mathbb{F}[z]^n = \operatorname{im}_{\mathbb{F}[z]} Z$ we conclude that there exists $u \in \mathbb{F}[z]^k$ such that $uZ = \zeta$.

Now we show (c) and (d): We have $\deg G \leq \sum_{i=1}^k \nu_k = \delta$ and

$$[P \mid R] = \left[\begin{array}{ccc|c|ccc|c} & & & & & & & g_{1,0} \\ & I_{\nu_1} & & & & & & \vdots \\ & & & & & & & g_{1,\nu_1-1} \\ 0 & \cdots & 0 & & & & & g_{1,\nu_1} \\ \hline & & & \ddots & & & & \vdots \\ \hline & & & & & & & g_{k,0} \\ & & & & & I_{\nu_k} & & \vdots \\ & & & & & & & g_{k,\nu_k-1} \\ & & & & 0 & \cdots & 0 & g_{k,\nu_k} \end{array}\right], \quad L'(G) = \begin{bmatrix} g_{1,\nu_1} \\ \vdots \\ g_{k,\nu_k} \end{bmatrix}.$$

Therefore, $[P \mid R]$ has full row rank if and only if $L'(G)$ has full row rank. Observe that in this case $L(G) = L'(G)$, in particular G has full row rank.
We have $L(G) = L'(G)$ if and only if G has row degrees $\nu_1, \ldots, \nu_k$. In this case G is a minimal basis if and only if $\operatorname{rank} L(G) = \operatorname{rank} L'(G) = k$ (again we used Theorem 1.3 (ii)). This proves (c).
The matrix $[zP+Q \mid R]$ can be transformed by elementary row operations

over $\mathbb{F}[z]$ of the type (O3) (cf. Remark 1.2) into

$$\left[\begin{array}{c|c|c|c} 0\cdots 0 & & & G_1 \\ & & & g_{1,1} \\ H_{\nu_1} & & & \vdots \\ & & & g_{1,\nu_1} \\ \hline & \ddots & & \vdots \\ \hline & & 0\cdots 0 & G_k \\ & & & g_{k,1} \\ & & H_{\nu_k} & \vdots \\ & & & g_{k,\nu_k} \end{array}\right] \quad \text{where } H_{\nu_i} := \begin{bmatrix} -1 & z & & \\ & -1 & \ddots & \\ & & \ddots & z \\ & & & -1 \end{bmatrix} \in Gl_{\nu_i}(\mathbb{F}[z])\,.$$

Therefore, $[zP+Q \,|\, R]$ is right invertible if and only if G is right invertible (use Theorem 1.7 (iv)). □

We apply the results given in Theorem 2.9 (b) to an example in order to illustrate these findings:

Example 2.10
Let $\mathbb{F} = \mathbb{F}_3$. *We consider*

$$zP+Q := \mathcal{L}(0,1,2) = \begin{bmatrix} 0 & 0 & 0 \\ z & 0 & 0 \\ -1 & 0 & 0 \\ 0 & z & 0 \\ 0 & -1 & z \\ 0 & 0 & -1 \end{bmatrix} \quad \textit{and} \quad R := \left[\begin{array}{ccc} 1 & 1 & 1 \\ \hline 1 & 0 & 2 \\ 0 & 2 & 1 \\ \hline 2 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 2 & 2 \end{array}\right] .$$

Now the matrix

$$G := Z(0,1,2)R = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2z & 2+z \\ 2+z^2 & 2z^2 & 2z^2 \end{bmatrix}$$

has degree $\deg G = 0+1+2 = 3$ *and gives rise to the submodule* $\operatorname{im}_{\mathbb{F}[z]} G = \mathcal{C}_{\mathrm{P}}(P,Q,R) \subseteq \mathbb{F}[z]^3$, *which has rank at most* 3. *Lemma 1.4 (b) tells us that* $\mathcal{C}_{\mathrm{P}}(P,Q,R)$ *has complexity at most* 3.
We can calculate that G *has exactly rank* 3 *and that it is not basic, hence* $\mathcal{C}_{\mathrm{P}}(P,Q,R)$ *is a* $(3,3)$*-submodule with generator matrix* G, *but it is not a* $(3,3)$*-convolutional code. Furthermore, the leading coefficient matrix of* G *has full rank and with Theorem 1.3 we conclude that* $\mathcal{C}_{\mathrm{P}}(P,Q,R)$ *has complexity exactly* 3.

The following corollary is not necessary for the remaining part of Section 2.2, but it will be used in Section 5.3 in the proof of Theorem 5.22.

Corollary 2.11
Let $zP+Q = \mathcal{L}(\nu_1, \ldots, \nu_k)$ *and define* $Z := Z(\nu_1, \ldots, \nu_k)$ *as in Theorem 2.9 (b). Then we have* $\operatorname{im}_{\mathbb{F}[z]} Z = \ker_{\mathbb{F}[z]}[zP+Q]$.

PROOF: Theorem 1.10 along with the right invertibility of the matrix pencil $[zP+Q]^{\mathrm{t}}$ yields that $\ker_{\mathbb{F}[z]}[zP+Q]$ is a convolutional code. With Theorem 1.7 "(vi)⇔(ix)" we have $\ker_{\mathbb{F}[z]}[zP+Q] = \ker_{\mathbb{F}(z)}[zP+Q] \cap \mathbb{F}[z]^n$. We saw in the proof of Theorem 2.9 (b) that $\operatorname{im}_{\mathbb{F}[z]} Z = \ker_{\mathbb{F}(z)}(zP+Q) \cap \mathbb{F}[z]^n$ and this implies the assertion. □

Now we are able to state and to prove our main results for PQR-representations, these are the Theorems 2.12, 2.13 and 2.15:

Theorem 2.12 (PQR-Realisation Theorem I: Existence) *cf. [SGR98, Thm. 2.1] or (in system theoretical language) [Kui94, Thm. 5.1]*
Let $\mathcal{C} \subseteq \mathbb{F}[z]^n$ *be a submodule of rank* $k \leq n$ *and complexity* $\delta > 0$. *Then there exists a matrix triple*

$$(P, Q, R) \in \mathbb{F}^{(\delta+k)\times\delta} \times \mathbb{F}^{(\delta+k)\times\delta} \times \mathbb{F}^{(\delta+k)\times n}$$

satisfying the conditions (P1) – (P3) *such that* $\mathcal{C} = \mathcal{C}_{\mathrm{P}}(P, Q, R)$. *If* $\mathcal{C}$ *is even an* (n, k, δ)*-convolutional code, then* $\mathcal{C}$ *has a* PQR*-representation satisfying* (P1) – (P4).

PROOF: (The idea of the proof can be found in [Kui94].) Let $\mathcal{C} = \operatorname{im}_{\mathbb{F}[z]} G$ where $G \in \mathbb{F}[z]^{k\times n}$ is a minimal basis with rows $G_1, \ldots, G_k$ and row degrees $\nu_1 \leq \ldots \leq \nu_k$. In particular, Theorem 1.3 implies $\sum_{i=1}^{k} \nu_i = \delta$ and $\operatorname{rank} L(G) = k$. Now write

$$G_i = \sum_{j=0}^{\nu_i} g_{i,j} z^j \quad \text{for} \quad 1 \leq i \leq k \quad \text{where} \quad g_{i,j} \in \mathbb{F}^n$$

and define

$$zP+Q := \mathcal{L}(\nu_1, \ldots, \nu_k) \in \mathbb{F}[z]^{(\delta+k)\times\delta} \quad , \quad R := \left[\begin{array}{c} g_{1,0} \\ \vdots \\ g_{1,\nu_1} \\ \hline \vdots \\ \hline g_{k,0} \\ \vdots \\ g_{k,\nu_k} \end{array}\right] \in \mathbb{F}[z]^{(\delta+k)\times n} \quad .$$

Observe that $Z(\nu_1, \ldots, \nu_k)R = G$, where $Z(\nu_1, \ldots, \nu_k)$ is defined as in Theorem 2.9 (b). Now Theorem 2.9 (a), (c) and (d) imply the assertion. □

Theorem 2.13 (PQR**-Realisation Theorem II: Parameters)** *cf. [SGR98, Thm. 2.2] or (in system theoretical language) [Kui94, Thm. 4.3] Consider a matrix triple* $(P, Q, R) \in \mathbb{F}^{(\delta+k)\times\delta} \times \mathbb{F}^{(\delta+k)\times\delta} \times \mathbb{F}^{(\delta+k)\times n}$ *where* $k \leq n$ *and* $\delta > 0$. *Let* $\hat{\delta}$ *denote the complexity of the submodule* $\mathcal{C}_{\mathrm{P}}(P, Q, R) \subseteq \mathbb{F}[z]^n$. *Then the following statements hold:*

(a) The submodule $\mathcal{C}_{\mathrm{P}}(P, Q, R) \subseteq \mathbb{F}[z]^n$ *has complexity at most* δ *and rank at most* $k + (\delta - \hat{\delta})$.

(b) The submodule $\mathcal{C}_{\mathrm{P}}(P, Q, R) \subseteq \mathbb{F}[z]^n$ *has complexity exactly* δ *if and only if* (P1) – (P3) *are satisfied. In this case the submodule has rank exactly* k.

(c) The submodule $\mathcal{C}_{\mathrm{P}}(P, Q, R)$ *is an* (n, k, δ)*-convolutional code if and only if the conditions* (P1) – (P4) *are satisfied.*

Proof: (The proof uses results from Glüsing-Lüerßen, cf. [Glub].) With Remark 2.8 we can assume without loss of generality that $zP + Q$ is given in a canonical form, i.e.

$$zP + Q = \left[\begin{array}{c} \mathcal{L}(\nu_1, \ldots, \nu_s) \\ \hline \mathcal{M}(\mu_1, \ldots, \mu_t) \\ \hline zA_0 + B_0 \end{array}\right]$$

where $zA_0 + B_0$ is a regular matrix pencil (if it is present at all). We use a partition of R given by

$$R = \left[\begin{array}{c} \hat{R} \\ \tilde{R} \end{array}\right] \begin{array}{l} \} \sum_{i=1}^{s}(\nu_i + 1) \\ \} \text{"rest"} \end{array} \quad .$$

Proof of part (a): We have to investigate $\mathcal{C}_{\mathrm{P}}(P, Q, R) = \big(\ker_{\mathbb{F}[z]}(zP + Q)\big) \cdot R$. Because the matrices $\mathcal{M}(\mu_1, \ldots, \mu_t)$ and the matrix $zA_0 + B_0$ have full row rank (if they are present at all), they have trivial left kernel and we remove the corresponding block-rows from $zP + Q$ and R. Hence

$$\mathcal{C}_{\mathrm{P}}(P, Q, R) = \big(\ker_{\mathbb{F}[z]}[\mathcal{L}(\nu_1, \ldots, \nu_s), 0]\big) \cdot \hat{R} \quad .$$

Since presence or absence of zero-columns in $[\mathcal{L}(\nu_1, \ldots, \nu_s), 0]$ does not change $\ker_{\mathbb{F}[z]}[\mathcal{L}(\nu_1, \ldots, \nu_s), 0]$, we can make a further reduction-step by removing zero-columns. With $z\hat{P} + \hat{Q} = \mathcal{L}(\nu_1, \ldots, \nu_s)$ we get

$$\mathcal{C}_{\mathrm{P}}(P, Q, R) = \Big(\ker_{\mathbb{F}[z]}(z\hat{P} + \hat{Q})\Big) \cdot \hat{R} = \mathcal{C}_{\mathrm{P}}(\hat{P}, \hat{Q}, \hat{R}) \quad .$$

With Theorem 2.9 (b) applied on $(\hat{P}, \hat{Q}, \hat{R})$, we know that a generator matrix of $\mathcal{C}_{\mathrm{P}}(\hat{P}, \hat{Q}, \hat{R}) = \mathcal{C}_{\mathrm{P}}(P, Q, R)$ is given by $\hat{G} := Z(\nu_1, \ldots, \nu_s)\hat{R} \in \mathbb{F}[z]^{s \times n}$ which has degree at most $\sum_{i=1}^{s} \nu_i$. Furthermore, $\mathcal{C}_{\mathrm{P}}(P, Q, R)$ has rank at most s.
By Lemma 1.4 (b) and by comparing the sizes of $zP + Q$, $z\hat{P} + \hat{Q}$ we obtain

$$\hat{\delta} \leq \deg \hat{G} \leq \sum_{i=1}^{s} \nu_i \leq \delta \tag{2.3}$$

and

$$\delta + k \geq \sum_{i=1}^{s} (\nu_i + 1) = \sum_{i=1}^{s} \nu_i + s \geq \hat{\delta} + s \quad , \quad \text{in particular} \quad s \leq k + (\delta - \hat{\delta}) \quad .$$

This shows that the complexity $\hat{\delta}$ of $\mathcal{C}_{\mathrm{P}}(P, Q, R)$ is less or equal δ and that $\mathcal{C}_{\mathrm{P}}(P, Q, R)$ has rank less or equal $k + (\delta - \hat{\delta})$.

Proof of the "if"-part of (b): With Theorem 2.9 (a) and (c) we know that $zP + Q = \mathcal{L}(\nu_1, \ldots, \nu_k)$ and that the generator matrix $G := Z(\nu_1, \ldots, \nu_k)R \in \mathbb{F}[z]^{k \times n}$ is a minimal basis with row degrees $\nu_1, \ldots, \nu_k$. The Theorem of Forney (Theorem 1.3) implies that $\hat{\delta} = \deg G = \sum_{i=1}^{k} \nu_k = \delta$ and the submodule $\mathcal{C}_{\mathrm{P}}(P, Q, R)$ has complexity exactly δ (cf. Theorem 1.3).

Proof of the "only if"-part of (b): If one of the properties (P1) or (P3) does not hold, Theorem 2.9 (a) implies that $zP + Q$ has a block of $\mathcal{M}(\mu_1, \ldots, \mu_t)$-type or a regular block $zA_0 + B_0$. This implies

$$\sum_{i=1}^{s} \nu_i < \delta \quad .$$

With (2.3) we obtain $\hat{\delta} < \delta$, which contradicts the assumption that $\mathcal{C}_{\mathrm{P}}(P, Q, R)$ has complexity exactly δ. Hence, (P1) and (P3) must hold and thus $s = k$. Again, we construct a generator matrix $G := Z(\nu_1, \ldots, \nu_k)R$ of $\mathcal{C}_{\mathrm{P}}(P, Q, R)$ as in Theorem 2.9 (b). Recall that $\deg G \leq \delta$ by construction of G. If (P2) does not hold, then Theorem 2.9 (c) yields that G is not a minimal basis with row degrees $\nu_1, \ldots, \nu_k$, which in turn implies along with Theorem 1.3 (iii) and Lemma 1.4 (b) that $\hat{\delta} < \deg G$. Hence $\hat{\delta} < \delta$, which is the same contradiction as above.
In particular we proved that G is both a minimal basis and has the row degrees $\nu_1, \ldots, \nu_k$ if and only if $\mathcal{C}_{\mathrm{P}}(P, Q, R)$ has complexity exactly δ. In this case $\mathcal{C}_{\mathrm{P}}(P, Q, R)$ has rank exactly k which completes the proof of (b).
Part (c) follows with (b) and Theorem 2.9 (d). □

Remark 2.14
There are examples of matrix triples $(P,Q,R) \in \mathbb{F}^{(\delta+k)\times\delta} \times \mathbb{F}^{(\delta+k)\times\delta} \times \mathbb{F}^{(\delta+k)\times n}$ *satisfying* $k \leq n$ *and* $\delta > 0$, *where* $\mathcal{C}_{\mathrm{P}}(P,Q,R)$ *has complexity* $\hat{\delta} < \delta$ *and* $\operatorname{rank} \mathcal{C}_{\mathrm{P}}(P,Q,R) = k + (\delta - \hat{\delta})$. *(A quite trivial example is* $\delta + k = n$, $P = 0_{n\times\delta} = Q$ *and* $R = I_n$, *where* $\mathcal{C}_{\mathrm{P}}(P,Q,R) = \mathbb{F}[z]^n$ *has complexity* $\hat{\delta} = 0$ *and rank* $n = k + \delta - \hat{\delta}$.*) In these cases, the upper bound for the rank of the submodule* $\mathcal{C}_{\mathrm{P}}(P,Q,R)$ *in Theorem 2.13 (a) is tight. Therefore, Theorem 2.13 (a) corrects Theorem 2.2 (i) of [SGR98], which states wrongly that* $\mathcal{C}_{\mathrm{P}}(P,Q,R)$ *has rank at most* k *in any case.*

Theorem 2.15 (PQR**-Realisation Theorem III: Uniqueness)** *cf. [SGR98, Thm. 2.3] or (in system theoretical language) [Kui94, Thm. 4.29] Let* (P,Q,R) *and* $(\hat{P},\hat{Q},\hat{R})$ *be two matrix triples with the as sizes in Notation 2.5, both satisfying the conditions* (P1), (P2) *and* (P3). *Then*

$$\mathcal{C}_{\mathrm{P}}(P,Q,R) = \mathcal{C}_{\mathrm{P}}(\hat{P},\hat{Q},\hat{R}) \quad \Longleftrightarrow \quad (\hat{P},\hat{Q},\hat{R}) = (TPS, TQS, TR)$$

for some $T \in Gl_{\delta+k}(\mathbb{F})$ *and* $S \in Gl_{\delta}(\mathbb{F})$.

Proof: The proof of the "$\Leftarrow$"-part is Remark 2.8. Thus we have to prove the "$\Rightarrow$"-part: Without loss of generality we can assume that both $zP+Q$ and $z\hat{P}+\hat{Q}$ are given in a canonical form. With Theorem 2.9 (a), (b) and (c) we obtain:

$$zP + Q = \mathcal{L}(\nu_1, \ldots, \nu_k) \quad , \quad z\hat{P} + \hat{Q} = \mathcal{L}(\hat{\nu}_1, \ldots, \hat{\nu}_{\hat{k}})$$

where $\nu_1 \leq \ldots \leq \nu_k$ resp. $\hat{\nu}_1 \leq \ldots \leq \hat{\nu}_{\hat{k}}$ are the row degrees (Forney indices) of the minimal basis $G := Z(\nu_1, \ldots, \nu_k)R$ resp. $\hat{G} := Z(\hat{\nu}_1, \ldots, \hat{\nu}_{\hat{k}})\hat{R}$ of $\mathcal{C}_{\mathrm{P}}(P,Q,R)$. Since $G, \hat{G}$ have the same Forney indices, we have $k = \hat{k}$ and $\nu_i = \hat{\nu}_i$ for $1 \leq i \leq k$, which in turn implies $zP+Q = z\hat{P}+\hat{Q} = \mathcal{L}(\nu_1, \ldots, \nu_k)$,

$$G = Z(\nu_1, \ldots, \nu_k)R \ \text{ and } \ \hat{G} = Z(\nu_1, \ldots, \nu_k)\hat{R} \quad .$$

We will continue like this: Proposition 1.5 allows us to assume without loss of generality that G can be transformed to $\hat{G}$ by only one elementary operation of the type
(O1) rescaling a row of G by a factor $\lambda \in \mathbb{F} \setminus \{0\}$;
(O2') exchanging two rows of G which have the same degree;
(O3') replacing G_i by $G_i + z^l \cdot G_j$, where $j \neq i$, $\nu_j \leq \nu_i$ and $0 \leq l \leq \nu_i - \nu_j$.

We will "translate" each of these operations into constant row operations on $[zP+Q \mid R]$ to obtain $[zP'+Q' \mid \hat{R}]$ (this means transformation of $[zP+Q \mid R]$ to $T[zP+Q \mid R] = [zP'+Q' \mid \hat{R}]$ by some $T \in Gl_{\delta+k}(\mathbb{F})$). This will destroy the canonical form of $zP+Q$, but we will see that we can make some constant column operations on $zP'+Q'$ which undo the destruction of the canonical form of $zP+Q$ (this means transformation of $T(zP+Q) = zP'+Q'$ to $T(zP+Q)S = zP+Q$ by some $S \in Gl_\delta(\mathbb{F})$). This will finish the proof.

Before we start with the "translation" of the three operations, we divide R and $\hat{R}$ into blocks, precisely

$$[R \mid \hat{R}] = \left[\begin{array}{c|c} R_1 & \hat{R}_1 \\ \vdots & \vdots \\ R_k & \hat{R}_k \end{array}\right] \begin{matrix} \}\nu_1+1 \\ \vdots \\ \}\nu_k+1 \end{matrix} .$$

Let us now "translate" (O1): We assume that $\hat{G}$ can be obtained from G by multiplying the i-th row of G with $0 \neq \lambda \in \mathbb{F}$. This is the case if and only if $\hat{R}_m = R_m$ for $m \neq i$ and $\hat{R}_i = \lambda R_i$. Scaling the whole i-th block row of $[zP+Q \mid R]$ by λ we get a transformed matrix $[zP'+Q' \mid \hat{R}]$. Rescaling the i-th block column of $[zP'+Q' \mid \hat{R}]$ by λ^{-1} we obtain $[zP+Q \mid \hat{R}]$.

$$\left[\begin{array}{ccc|c} L_{\nu_1} & & & \\ & \ddots & & \\ & & L_{\nu_k} & \end{array}\right] \left[\begin{array}{c|c} & \\ R & \end{array}\right] \underset{\text{row op.}}{\overset{\text{const.}}{\sim}} \left[\begin{array}{ccc|c} L_{\nu_1} & & & \\ & \ddots & & \\ & & L_{\nu_k} & \end{array}\right] \left[\begin{array}{c|c} & \\ \hat{R} & \end{array}\right] \underset{\text{col. op.}}{\overset{\text{const.}}{\sim}} \left[\begin{array}{ccc|c} L_{\nu_1} & & & \\ & \ddots & & \\ & & L_{\nu_k} & \end{array}\right] \left[\begin{array}{c|c} & \\ \hat{R} & \end{array}\right] .$$

Now we "translate" (O2'): We assume that $\hat{G}$ can be obtained from G by exchanging the i-th and the j-th row of G, where both rows have the same degree. This is the case if and only if $\hat{R}_m = R_m$ for $i \neq m \neq j$ and $\hat{R}_j = R_i$, $\hat{R}_i = R_j$. Exchanging the i-th block row with the j-th block row of $[zP+Q \mid R]$ yields a transformed matrix $[zP'+Q' \mid \hat{R}]$. Exchanging the i-th block column with the j-th block column of $[zP'+Q' \mid \hat{R}]$ we obtain $[zP+Q \mid \hat{R}]$.

$$\left[\begin{array}{ccc|c} & L_{\nu_i} & & \\ & & L_{\nu_j} & \\ & & & \end{array}\right] \left[\begin{array}{c|c} & \\ R & \end{array}\right] \underset{\text{exchange}}{\overset{\text{row}}{\sim}} \left[\begin{array}{ccc|c} & 0 & L_{\nu_j} & \\ & & & \\ L_{\nu_i} & 0 & & \end{array}\right] \left[\begin{array}{c|c} & \\ \hat{R} & \end{array}\right] \underset{\text{exchange}}{\overset{\text{col.}}{\sim}} \left[\begin{array}{ccc|c} & L_{\nu_i} & & \\ & & L_{\nu_j} & \\ & & & \end{array}\right] \left[\begin{array}{c|c} & \\ \hat{R} & \end{array}\right] .$$

Before we "translate" (O3'), we point out to the reader that

$$G_i + z^l \cdot G_j \;=\; (1, z, \ldots, z^{\nu_i}) \cdot \left(R_i + \begin{bmatrix} 0_{l\times n} \\ R_j \\ 0_{(\nu_i-\nu_j-l)\times n} \end{bmatrix} \right) \quad .$$

Now we "translate" (O3'): We assume that $\hat{G}$ can be obtained from G by replacing G_i by $G_i + z^l \cdot G_j$, where $j \neq i$, $\nu_j \leq \nu_i$ and $0 \leq l \leq \nu_i - \nu_j$. This is the case if and only if $\hat{R}_m = R_m$ for $m \neq i$ and $\hat{R}_i = R_i + \begin{bmatrix} 0_{l\times n} \\ R_j \\ 0_{(\nu_i-\nu_j-l)\times n} \end{bmatrix}$.

Hence, operation (O3') can be interpreted as adding the j-th block row of $[zP + Q \,|\, R]$ to a suitable sub-block row of the i-th bock row, precisely: We add the j-th block row to that sub-block row of the i-th bock row, which consists of the $(l+1)$-st up to the $(l+\nu_j+1)$-st row of that block itself. We obtain $[zP' + Q' \,|\, \hat{R}]$ where

$$zP' + Q' = \begin{bmatrix} \ddots & & & \\ & \boxed{L_{\nu_j}} & & \\ & & \ddots & \\ & \boxed{\boxed{L_{\nu_j}}} & \boxed{L_{\nu_i}} & \\ & & & \ddots \end{bmatrix} \quad .$$

This matrix can be transformed to $zP + Q$ by constant column operations, because in the i-th block column of $zP' + Q'$ there exists a suitable sub-block column which we can use to delete L_{ν_j} in the $\boxed{\boxed{L_{\nu_j}}}$-block just by subtracting. □

The following example might make the last part of the proof clearer.

Example 2.16
Let $\mathbb{F}$ *be the field* $\mathbb{F}_3$*. Consider*

$$G := \begin{bmatrix} z+1 & z & z \\ z^2+1 & z+1 & z^2 \end{bmatrix} \quad \textit{and} \quad \hat{G} := \begin{bmatrix} z+1 & z & z \\ 2z^2+z+1 & z^2+z+1 & 2z^2 \end{bmatrix} \quad .$$

Both matrices are minimal bases and $\hat{G}$ *can be obtained from* G *by replacing* G_2 *with* $G_2 + z \cdot G_1$*. We choose canonical* PQR*-representations of* $\mathrm{im}_{\mathbb{F}[z]} G$ *and*

$\mathrm{im}_{\mathbb{F}[z]}\hat{G}$ *as in the proof of Theorem 2.12. These are denoted by* $[zP+Q \mid R]$ *resp.* $[z\hat{P}+\hat{Q} \mid \hat{R}]$ *and these matrices are given by*

$$\left[\begin{array}{c|cc|ccc} z & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 1 & 1 & 1 \\ \hline 0 & z & 0 & 1 & 1 & 0 \\ 0 & -1 & z & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 & 0 & 1 \end{array}\right] \quad \textit{resp.} \quad \left[\begin{array}{c|cc|ccc} z & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 1 & 1 & 1 \\ \hline 0 & z & 0 & 1 & 1 & 0 \\ 0 & -1 & z & 1 & 1 & 0 \\ 0 & 0 & -1 & 2 & 1 & 2 \end{array}\right]$$

Now we illustrate the "translation" of (O3')*:*

$$[zP+Q \mid R] \overset{\text{row op.}}{\sim} \left[\begin{array}{c|cc|ccc} z & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 1 & 1 & 1 \\ \hline 0 & z & 0 & 1 & 1 & 0 \\ z & -1 & z & 1 & 1 & 0 \\ -1 & 0 & -1 & 2 & 1 & 2 \end{array}\right] \overset{\text{col. op.}}{\sim} \left[\begin{array}{c|cc|ccc} z & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 1 & 1 & 1 \\ \hline 0 & z & 0 & 1 & 1 & 0 \\ 0 & -1 & z & 1 & 1 & 0 \\ 0 & 0 & -1 & 2 & 1 & 2 \end{array}\right].$$

2.3 KLM-Representations of Convolutional Codes

In this section we turn to the second representation which was introduced in (2.1), the so called "KLM-representation". We will see that we can use a PQR-representation of a convolutional code $\mathcal{C} \subseteq \mathbb{F}[z]^n$ to generate a KLM-representation of the dual code $\mathcal{C}^\perp$ and vice versa (cf. Theorem 2.21). This allows us to give short proofs for the main results, if these are formulated for codes (and not for submodules in general): The equality $(\mathcal{C}^\perp)^\perp = \mathcal{C}$ makes it possible to translate KLM-representations of $\mathcal{C}$ to PQR-representations of $\mathcal{C}^\perp$. We will apply our results for PQR-representations to $\mathcal{C}^\perp$ and then we will re-translate them to $\mathcal{C}$.
We will not investigate the general case, where $\mathcal{C}$ is an arbitrary submodule, because then the proofs will be more sophisticated [7] and this would go beyond the scope of this thesis.

Notation 2.17
Consider a matrix triple $(K, L, M) \in \mathbb{F}^{\delta\times(\delta+n-k)} \times \mathbb{F}^{\delta\times(\delta+n-k)} \times \mathbb{F}^{n\times(\delta+n-k)}$

[7] In this case it would also be possible to use a PQR-representation of $\mathcal{C}$ to generate a KLM-representation of the so called dual submodule $\mathcal{C}^\perp := \{v \in \mathbb{F}[z]^n \mid vw^{\mathsf{t}} = 0 \text{ for all } w \in \mathcal{C}\}$ of $\mathcal{C}$, which is always a convolutional code. However, we will not have the equality $(\mathcal{C}^\perp)^\perp = \mathcal{C}$ if $\mathcal{C}$ is not a convolutional code. So it is not possible in general to "transfer" results about $\mathcal{C}^\perp$ to $\mathcal{C}$ and vice versa.

where $k \leq n$ *and* $\delta > 0$. *We will use the following abbreviation:*

$$\begin{aligned}
\mathcal{C}_{\mathrm{K}}(K,L,M) &:= \left\{ v \in \mathbb{F}[z]^n \;\middle|\; \exists\, \zeta \in \mathbb{F}[z]^\delta \;:\zeta(zK+L)+vM=0 \right\} \\
&= \left\{ v \in \mathbb{F}[z]^n \;\middle|\; \exists\, \zeta \in \mathbb{F}[z]^\delta \;:(\zeta,v)\begin{bmatrix} zK+L \\ M \end{bmatrix} = 0 \right\} \\
&= \left\{ v \in \mathbb{F}[z]^n \;\middle|\; vM \in \operatorname{im}_{\mathbb{F}[z]}(zK+L) \right\} \\
&= \left(\ker_{\mathbb{F}[z]} \begin{bmatrix} zK+L \\ M \end{bmatrix} \right) \cdot \begin{bmatrix} 0_{\delta\times n} \\ I_n \end{bmatrix} \quad .
\end{aligned}$$

Of course, any matrix triple (K,L,M) of the sizes as in Notation 2.17 gives rise to a submodule $\mathcal{C}_{\mathrm{K}}(K,L,M) \subseteq \mathbb{F}[z]^n$.

Definition 2.18
If a submodule $\mathcal{C} \subseteq \mathbb{F}[z]^n$ *can be written as* $\mathcal{C} = \mathcal{C}_{\mathrm{K}}(K,L,M)$ *with* (K,L,M) *as in Notation 2.17, then* (K,L,M) *is called* KLM**-representation** *of* $\mathcal{C}$.

As for (P,Q,R), some properties of (K,L,M) will be of frequent use and we will introduce shorthand expressions:

Properties 2.19
Consider (K,L,M) *as in Notation 2.17. Properties of interest are:*
(K1) K *has full row rank* δ;
(K2) $\begin{bmatrix} K \\ M \end{bmatrix}$ *has full column rank* $\delta + n - k$;
(K3) $\begin{bmatrix} zK+L \\ M \end{bmatrix}$ *is left invertible;*
(K4) $zK+L$ *is right invertible.*

Remark 2.20
These properties correspond to (P1) – (P4) *in Properties 2.7, precisely: Replacing* k *by* $n-k$ *and transposing the matrices of interest,* (K1) *"equals"* (P1), (K2) *"equals"* (P2), (K3) *"equals"* (P4) *and* (K4) *"equals"* (P3).
The properties (K3) *and* (K4) *seem to be interchanged, but this is due to the following fact: We saw in Section 2.2 that* (P4) *is responsible for right invertibility of a generator matrix of the submodule* $\mathcal{C}_{\mathrm{P}}(P,Q,R)$. *For* (K,L,M) *the property* (K4) *causes right invertibility of a generator matrix of* $\mathcal{C}_{\mathrm{K}}(K,L,M)$, *cf. [RSY96, Lemma 3.2].*

As we already indicated, PQR-representations and KLM-representations are linked by dualization:

Theorem 2.21 *cf. (in system theoretical language) [Kui94, Lemma 3.26] Any* PQR*-representation* (P, Q, R) *of an* (n, k, δ)*-convolutional code* $\mathcal{C}$ *of sizes as in Notation 2.5 defines a* KLM*-representation* $(P^{\mathrm{t}}, Q^{\mathrm{t}}, R^{\mathrm{t}})$ *of* $\mathcal{C}^{\perp}$*. Any* KLM*-representation* (K, L, M) *of an* (n, k, δ)*-convolutional code* $\mathcal{C}$ *of sizes as in Notation 2.17 and satisfying* (K4) *defines a* PQR*-representation* $(K^{\mathrm{t}}, L^{\mathrm{t}}, M^{\mathrm{t}})$ *of* $\mathcal{C}^{\perp}$*.*

Proof: (The proof stems from [Glub].) We only show the first part of the theorem, the second part follows with exactly the same arguments.
First of all, note that (P3) is satisfied because of Theorem 2.13 (c). Let $\mathcal{C} = \mathcal{C}_{\mathrm{P}}(P, Q, R) = \big(\ker_{\mathbb{F}[z]}(zP + Q)\big) \cdot R$, where $zP + Q$ is left invertible (i.e. (P3)). Right invertibility of $zP^{\mathrm{t}} + Q^{\mathrm{t}}$ along with Theorem and Definition 1.10 yields

$$\big(\ker_{\mathbb{F}[z]}(zP + Q)\big)^{\perp} = \operatorname{im}_{\mathbb{F}[z]}(zP^{\mathrm{t}} + Q^{\mathrm{t}})$$

and this is used in the following chain of equivalent statements:

$$\begin{aligned} v \in \mathcal{C}^{\perp} &\iff \forall\, \zeta \in \ker_{\mathbb{F}[z]}(zP + Q) \;:\; \zeta R v^{\mathrm{t}} = 0 \\ &\iff \forall\, \zeta \in \ker_{\mathbb{F}[z]}(zP + Q) \;:\; v R^{\mathrm{t}} \zeta^{\mathrm{t}} = 0 \\ &\iff v R^{\mathrm{t}} \in \big(\ker_{\mathbb{F}[z]}(zP + Q)\big)^{\perp} = \operatorname{im}_{\mathbb{F}[z]}(zP^{\mathrm{t}} + Q^{\mathrm{t}}) \\ &\iff v \in \mathcal{C}_{\mathrm{K}}(P^{\mathrm{t}}, Q^{\mathrm{t}}, R^{\mathrm{t}}) \quad . \end{aligned}$$

□

The foregoing Theorem 2.21 makes it possible to give short proofs for the three main results of this section, which are the following Theorems 2.22, 2.23 and 2.24.

Theorem 2.22 (KLM**-Realisation Theorem for Codes I: Existence)** *cf. [RSY96, Thm. 3.1] or (in system theoretical language) [Kui94, Thm. 5.14]*
Let $\mathcal{C} \subseteq \mathbb{F}[z]^n$ *be an* (n, k, δ)*-convolutional code such that* $\delta > 0$*. Then there exists a matrix triple*

$$(K, L, M) \in \mathbb{F}^{\delta \times (\delta+n-k)} \times \mathbb{F}^{\delta \times (\delta+n-k)} \times \mathbb{F}^{n \times (\delta+n-k)}$$

satisfying the conditions (K1) – (K4) *such that* $\mathcal{C} = \mathcal{C}_{\mathrm{K}}(K, L, M)$*.*

PROOF: The dual code of $\mathcal{C}$ has a PQR-representation $(K^{\mathrm{t}}, L^{\mathrm{t}}, M^{\mathrm{t}})$, which satisfies (P1) – (P4) because of Theorem 2.13 (c). Theorem 2.21 along with Theorem and Definition 1.10 yields that (K, L, M) is a KLM-representation of $\mathcal{C}$. Obviously, the properties (K1) – (K4) hold (cf. Remark 2.20). □

Theorem 2.23 (KLM**-Realisation Theorem for Codes II: Parameters)**
cf. (in system theoretical language) [Kui94, Thm. 4.28]
Let $(K, L, M) \in \mathbb{F}^{\delta\times(\delta+n-k)} \times \mathbb{F}^{\delta\times(\delta+n-k)} \times \mathbb{F}^{n\times(\delta+n-k)}$ *be a matrix triple such that* $k \leq n$, $\delta > 0$ *and satisfying the properties* (K1) – (K4). *Then* $\mathcal{C}_{\mathrm{K}}(K, L, M)$ *is an* (n, k, δ)*-convolutional code.*

PROOF: With Theorem 2.13, the submodule $\hat{\mathcal{C}} := \mathcal{C}_{\mathrm{P}}(K^{\mathrm{t}}, L^{\mathrm{t}}, M^{\mathrm{t}})$ is an $(n, n-k, \delta)$-convolutional code because (K1) – (K4) translate into (P1) – (P4). Theorem 2.21 along with Theorem and Definition 1.10 yields that $\hat{\mathcal{C}}^{\perp} = \mathcal{C}_{\mathrm{K}}(K, L, M)$ is a (n, k, δ)-convolutional. □

Theorem 2.24 (KLM**-Realisation Theorem for Codes III: Uniqueness)**
[RSY96, Thm. 3.4] or (in system theoretical language) [Kui94, Thm. 4.35]
Let (K, L, M) *and* $(\hat{K}, \hat{L}, \hat{M})$ *be two matrix triples with the sizes as in Notation 2.17 and both satisfying the conditions* (K1) – (K4). *Then*

$$\mathcal{C}_{\mathrm{K}}(K, L, M) = \mathcal{C}_{\mathrm{K}}(\hat{K}, \hat{L}, \hat{M}) \quad \Longleftrightarrow \quad (\hat{K}, \hat{L}, \hat{M}) = (TKS, TLS, MS)$$

for some $T \in Gl_{\delta}(\mathbb{F})$ *and* $S \in Gl_{\delta+n-k}(\mathbb{F})$.

PROOF: The proof of the "$\Leftarrow$"-part can be done straightforwardly (without use of the properties (K1) – (K4)).

Proof of the "$\Rightarrow$"-part: First note that $\mathcal{C} := \mathcal{C}_{\mathrm{K}}(K, L, M) = \mathcal{C}_{\mathrm{K}}(\hat{K}, \hat{L}, \hat{M})$ is a convolutional code because of Theorem 2.23. Hence the triples $(K^{\mathrm{t}}, L^{\mathrm{t}}, M^{\mathrm{t}})$ and $(\hat{K}^{\mathrm{t}}, \hat{L}^{\mathrm{t}}, \hat{M}^{\mathrm{t}})$ both define PQR-representations of $\mathcal{C}^{\perp}$. With Theorem 2.15 there exist $T \in Gl_{\delta}(\mathbb{F})$ and $S \in Gl_{\delta+n-k}(\mathbb{F})$ such that

$$(\hat{K}^{\mathrm{t}}, \hat{L}^{\mathrm{t}}, \hat{M}^{\mathrm{t}}) = (SK^{\mathrm{t}}T, SL^{\mathrm{t}}T, SM^{\mathrm{t}})$$ □

3 Cyclicity

For the remaining part of this thesis n denotes a positive integer such that

$$\text{the characteristic of } \mathbb{F} \text{ does not divide } n. \tag{3.1}$$

This is a usual assumption in the theory of cyclic block codes and makes sure that the polynomial $x^n - 1$ factors into different prime polynomials over $\mathbb{F}$. It is also familiar to index the entries of a vector v of length n with $0, \ldots, n-1$ (i.e. $v = (v_0, \ldots, v_{n-1})$) when dealing with cyclic codes.

In Section 3.1 we will recall the concept of cyclicity for block codes and give basic results. For details concerning the theory of cyclic block codes we refer to [MS78, Chapter 7] or [LC83, Chapter 4]. In Section 3.2 we will introduce the notion of cyclic convolutional codes. Our definitions and results are based on the fundamental articles [Pir76] and [Roo79] and, in particular, on the article [GS02a].

3.1 Cyclic Block Codes

A block code $\mathcal{C} \subseteq \mathbb{F}^n$ is said to be **cyclic** if

$$(v_0, \ldots, v_{n-1}) \in \mathcal{C} \implies (v_{n-1}, v_0, \ldots, v_{n-2}) \in \mathcal{C} \tag{3.2}$$

and we say that $\mathcal{C}$ is **invariant under the cyclic shift**.

We can identify a block code $\mathcal{C} \subseteq \mathbb{F}^n$ as a subset of the ring $A := \mathbb{F}[x]/\langle\, x^n - 1 \,\rangle$ via the $\mathbb{F}$-isomorphism

$$\mathfrak{p} : \mathbb{F}^n \to A \;\;,\;\; v = (v_0, \ldots, v_{n-1}) \mapsto \mathfrak{p}(v) = \sum_{i=0}^{n-1} v_i x^i \;\;, \tag{3.3}$$

where A is displayed in the canonical way

$$A = \{ f \in \mathbb{F}[x] \mid \deg_x f < n \} \text{ with multiplication modulo } x^n - 1 \;\;.$$

The set $\mathfrak{p}(\mathcal{C})$ is called **polynomial representation** of $\mathcal{C}$. The inverse mapping of $\mathfrak{p}$ is denoted by $\mathfrak{v}$ (think of "vectorization").

One can readily show that a block code $\mathcal{C}$ is cyclic if and only if its polynomial representation is an ideal in A, or in other words

$$\mathcal{C} \text{ is cyclic if and only if } [\; a \in \mathfrak{p}(\mathcal{C}) \implies xa \in \mathfrak{p}(\mathcal{C}) \;] \;. \tag{3.4}$$

Since each ideal in A is principal, the polynomial representation of a cyclic (n,k)-block code $\mathcal{C}$ can be written as $\mathfrak{p}(\mathcal{C}) = \langle\, g \,\rangle$ for some $g \in A$. It is well known in the theory of cyclic block codes that we can choose g such that $g|(x^n - 1)$. In this case we have $\deg_x g = n - k$ and

$$\begin{bmatrix} \mathfrak{v}(g) \\ \mathfrak{v}(xg) \\ \vdots \\ \mathfrak{v}(x^{k-1}g) \end{bmatrix} \in \mathbb{F}^{k\times n} \text{ is a generator matrix of } \mathcal{C} \quad . \tag{3.5}$$

We can check cyclicity of an (n,k)-block code $\mathcal{C}$ as follows: If $v_1, \ldots, v_k$ is a basis of $\mathcal{C}$, then $\mathcal{C}$ is cyclic if and only if $x\mathfrak{p}(v_1), \ldots, x\mathfrak{p}(v_k) \in \mathfrak{p}(\mathcal{C})$.

3.2 Cyclic Submodules of $\mathbb{F}[z]^n$ and Cyclic Convolutional Codes

Before we turn to the notion of cyclicity in the context of submodules and convolutional codes, we want to point out to the reader that every submodule $\mathcal{C}$ can be identified as a subset of $A[z]$ by an extension of the mapping $\mathfrak{p}$ in (3.3) given by

$$\mathfrak{p} : \mathbb{F}[z]^n \to A[z] \quad , \quad \sum_{\nu\geq 0} z^\nu v_\nu \mapsto \sum_{\nu\geq 0} z^\nu \mathfrak{p}(v_\nu). \tag{3.6}$$

The extended mapping $\mathfrak{p}$ in (3.6) is an isomorphism of $\mathbb{F}[z]$-(left-)modules. Again, the set $\mathfrak{p}(\mathcal{C})$ is called **polynomial representation** of $\mathcal{C}$ and $\mathfrak{p}^{-1}$ is denoted by $\mathfrak{v}$.

3.2.1 A Generalized Concept of Cyclicity

The first attempt to generalize the concept of cyclicity to a convolutional code $\mathcal{C}$ by requiring (3.2) or, equivalently, by using the definition in (3.4) fails, as the following proposition tells us:

Proposition 3.1 *[Pir76, Thm. 3.12], [Roo79, Thm. 6], [GS02a, Prop. 2.7] A convolutional code $\mathcal{C} \subseteq \mathbb{F}[z]^n$ satisfying (3.2) is a block code, i.e. it has complexity* 0. □

In [Pir76] Piret suggested the following generalization of cyclicity for submodules: He called a submodule $\mathcal{C} \subseteq \mathbb{F}[z]^n$ cyclic if

$$g = \sum_{\nu \geq 0} z^\nu g_\nu \in \mathfrak{p}(\mathcal{C}) \quad \Longrightarrow \quad x * g := \sum_{\nu \geq 0} z^\nu x^{(m^\nu)} g_\nu \in \mathfrak{p}(\mathcal{C}) \quad ,$$

where n and m are coprime. This means (re-translated to $\mathcal{C}$ via $\mathfrak{v}$) that the coefficient vector $\mathfrak{v}(g_\nu)$ of $\sum_{\nu \geq 0} z^\nu \mathfrak{v}(g_\nu) \in \mathcal{C}$ undergoes m^ν cyclic shifts (instead of one) and that the resulting vector is in $\mathcal{C}$ again. The condition $\gcd(m, n) = 1$ implies that the mapping $x \mapsto x^m$ induces an $\mathbb{F}$-automorphism on the ring A. This allowed Piret (loosely speaking) to introduce an $\mathbb{F}$-algebra structure on the left $\mathbb{F}[z]$-module $A[z]$; then a submodule $\mathcal{C}$ is cyclic if and only if $\mathfrak{p}(\mathcal{C})$ is a left ideal in this $\mathbb{F}$-algebra.

Roos generalized this concept of cyclicity to arbitrary $\mathbb{F}$-automorphisms σ of A in his paper [Roo79]. We will adopt his definition of cyclicity. Since this definition depends on the choice of σ, we will use the name "σ-cyclicity":

Definition 3.2 *[Roo79]*
Consider $\sigma \in Aut_{\mathbb{F}}(A)$, where $Aut_{\mathbb{F}}(A)$ denotes the group of all $\mathbb{F}$-algebra automorphisms on A. A submodule $\mathcal{C} \subseteq \mathbb{F}[z]^n$ is called **σ-cyclic** *if*

$$g = \sum_{\nu \geq 0} z^\nu g_\nu \in \mathfrak{p}(\mathcal{C}) \quad \Longrightarrow \quad x *_\sigma g := \sum_{\nu \geq 0} z^\nu \sigma^\nu(x) g_\nu \in \mathfrak{p}(\mathcal{C}) \quad . \tag{3.7}$$

Soon we give an example of a cyclic convolutional code. But before this, we show how one can check cyclicity of a submodule $\mathcal{C}$ with an $\mathbb{F}[z]$-basis $v_1, \ldots, v_k$ in a finite number of steps. Note that it is not clear whether it is sufficient to check if $x\mathfrak{p}(v_1), \ldots, x\mathfrak{p}(v_k) \in \mathfrak{p}(\mathcal{C})$.

Lemma 3.3
Let $\mathcal{C} \subseteq \mathbb{F}[z]^n$ be an (n, k)-submodule with $\mathbb{F}[z]$-basis $v_1, \ldots, v_k$ and let $\sigma \in Aut_{\mathbb{F}}(A)$. Then $\mathcal{C}$ is σ-cyclic if and only if

$$x *_\sigma \mathfrak{p}(v_1), \ldots, x^{n-1} *_\sigma \mathfrak{p}(v_1), \ldots, x *_\sigma \mathfrak{p}(v_k), \ldots, x^{n-1} *_\sigma \mathfrak{p}(v_k) \in \mathfrak{p}(\mathcal{C}) \quad . \tag{3.8}$$

PROOF: If $\mathcal{C}$ is σ-cyclic, then (3.8) is true by the definition of σ-cyclicity. This proves the "only if"-part. For the "if"-part, we observe that $\sigma^i(x) \in A$ can be identified with a polynomial in $\mathbb{F}[x]$ with x-degree at most $n-1$ for any $i \in \mathbb{N}_0$. Hence $\sigma^i(x) *_\sigma \mathfrak{p}(v_j)$ is an $\mathbb{F}$-linear combination of the codewords $\mathfrak{p}(v_j), x *_\sigma \mathfrak{p}(v_j), \ldots, x^{n-1} *_\sigma \mathfrak{p}(v_j)$ for any $i \in \mathbb{N}_0$ and $1 \leq j \leq k$. Therefore

$\sigma^i(x) *_\sigma \mathfrak{p}(v_j)$ is in $\mathfrak{p}(\mathcal{C})$ since $\mathfrak{p}(\mathcal{C})$ is an $\mathbb{F}[z]$-left module.
Now we consider an arbitrary codeword $w := \sum_{i=1}^{k}(\sum_{j\geq 0} f_{ij}z^j)v_i$ where $f_{ij} \in \mathbb{F}$. We have to show that

$$x *_\sigma \mathfrak{p}(w) = \sum_{i=1}^{k}\sum_{j\geq 0} f_{ij}z^j \underbrace{\sigma^j(x) *_\sigma \mathfrak{p}(v_i)}_{\in \mathfrak{p}(\mathcal{C})}$$

is an element of $\mathfrak{p}(\mathcal{C})$. Again, this is true because $\mathfrak{p}(\mathcal{C})$ is an $\mathbb{F}[z]$-left module. □

Example 3.4 *cf. [GS02a, Example 2.11 (a)]*
We consider $n = 3$, $\mathbb{F} = \mathbb{F}_4 = \{0, 1, \alpha, \alpha^2\}$, $\sigma \in Aut_{\mathbb{F}}(A)$ *defined by* $\sigma(x) = \alpha^2 x$ *(cf. Remark 3.5) and the matrix*

$$G := [1 + z + z^2, \alpha + z + \alpha^2 z^2, \alpha^2 + z + \alpha z^2] \quad .$$

It is readily shown that G *is basic, hence* $\mathcal{C} := \mathrm{im}_{\mathbb{F}[z]}G$ *is a convolutional code. Furthermore,* $\mathcal{C}$ *is* σ*-cyclic: If* v *denotes the first (and only) row of* G*, we have to show that* $x *_\sigma \mathfrak{p}(v)$ *and* $x^2 *_\sigma \mathfrak{p}(v)$ *are in* $\mathfrak{p}(\mathcal{C})$ *again, as Lemma 3.3 tells us. We have*

$$g := \mathfrak{p}(v) = 1 + \alpha x + \alpha^2 x^2 + z(1 + x + x^2) + z^2(1 + \alpha^2 x + \alpha x^2) \quad .$$

We calculate

$$x *_\sigma g = \alpha^2 + x + \alpha x^2 + z\alpha^2(1 + x + x^2) + z^2(\alpha^2 + \alpha x + x^2) = \alpha^2 g \in \mathfrak{p}(\mathcal{C})$$

and therefore we have $x^2 *_\sigma g = \alpha g \in \mathfrak{p}(\mathcal{C})$.
We even showed that $\mathcal{C}$ *is the smallest* σ*-cyclic convolutional code containing the codeword* $\mathfrak{v}(g)$*. One can also show that* $d_{free}(\mathcal{C}) = 9$ *(cf. [GS02a, Example 2.11 (a)]), which is optimal among all* $(3, 1, 2)$*-submodules (recall the MDS-bound in (1.1)). Hence* $\mathcal{C}$ *is a MDS-convolutional code.*

In the generalized concept of cyclicity we use automorphisms from the group $Aut_{\mathbb{F}}(A)$. This gives rise to the question how this group looks like. In particular we may ask, if there are nontrivial automorphisms in $Aut_{\mathbb{F}}(A)$ at all. We will answer this question in Section 3.2.2. For the moment the following remark gives first information.

Remark 3.5 *[GS02a, Example 2.13]*
Note that any $\mathbb{F}$*-algebra automorphism* $\sigma \in Aut_{\mathbb{F}}(A)$ *is completely determined*

by the value of $\sigma(x) \in A$. Furthermore, the elements $1, x, \ldots, x^{n-1} \in A$ are $\mathbb{F}$-linearly independent, hence $1, \sigma(x), \ldots, \sigma(x)^{n-1} \in A$ are $\mathbb{F}$-linearly independent, too. It is not hard to show that this condition is even sufficient to define an automorphism σ, precisely: If $a \in A$ satisfies $a^n = 1$ and if $1, a, \ldots, a^{n-1}$ are $\mathbb{F}$-linearly independent, then $a := \sigma(x)$ defines an automorphism $\sigma \in Aut_{\mathbb{F}}(A)$.
For example we consider the case $\mathbb{F} = \mathbb{F}_4 = \{0, 1, \alpha, \alpha^2\}$ and use this brute force method to determine the group $Aut_{\mathbb{F}}(A)$ (there are more systematic ones, see Section 3.2.2). This yields $Aut_{\mathbb{F}}(A) = \{\sigma \mid \sigma(x) \in \{x, x^2, \alpha x, \alpha^2 x, \alpha x^2, \alpha^2 x^2\}\}$.

Now we extend "$*_\sigma$" in order to obtain a (non-commutative) ring structure on $A[z]$ and to generalize the statement "codes can be identified with ideals" in this context:

Definition and Theorem 3.6 *[GS02a, Def. 2.9, Observation 2.10 (a)]*
Let $\sigma \in Aut_{\mathbb{F}}(A)$. The product $g *_\sigma h$ of the polynomials $g = \sum_{\nu \geq 0} z^\nu g_\nu$, $h = \sum_{\mu \geq 0} z^\mu h_\mu \in A[z]$ is defined as

$$\Big(\sum_{\nu \geq 0} z^\nu g_\nu \Big) *_\sigma \Big(\sum_{\mu \geq 0} z^\mu h_\mu \Big) := \sum_{\lambda \geq 0} z^\lambda \sum_{\nu+\mu=\lambda} \sigma^\mu(g_\nu) h_\mu \quad .$$

The set $A[z]$ equipped with the usual addition and the multiplication "$*_\sigma$" is called the **Piret algebra** *(with parameters $q = |\mathbb{F}|$, n, σ) and will be denoted by $A[z; \sigma]$.*
Indeed, $A[z; \sigma]$ is an $\mathbb{F}$-algebra and it carries a canonical left $\mathbb{F}[z]$-module structure. It is non-commutative if and only if $\sigma(x) \neq x$, i.e. σ is not the identity on A. □

If we compute the product $g *_\sigma h$, then we can evaluate it distributively as usual (with respect to the rules in a non-commutative ring) and then shift all coefficients g_ν to the right side of z according to the rule

$$a *_\sigma z = z *_\sigma \sigma(a) \quad \text{for all} \quad a \in A \quad .$$

In this setting σ-cyclic submodules appear as a natural generalisation of cyclic block codes, because block codes are the (left) ideals in A and σ-cyclic submodules turn out to be left ideals in $A[z; \sigma]$:

Proposition 3.7 *[GS02a, Observation 2.10 (b)]*
Let $\sigma \in Aut_{\mathbb{F}}(A)$. A submodule $\mathcal{C} \subseteq \mathbb{F}[z]^n$ is σ-cyclic if and only if its polynomial representation $\mathfrak{p}(\mathcal{C})$ is a left ideal in $A[z; \sigma]$. □

3.2.2 More Information on the Structure of A and $Aut_{\mathbb{F}}(A)$

It will be very helpful for further investigations to display A as a cartesian product of fields as it is done in this section. We follow the lines of [GS02a, Section 3]. The notation as introduced here will also be used later on in this thesis.

The assumption (3.1) implies that the normalized prime factors $\pi_1, \ldots, \pi_r \in \mathbb{F}[x]$ of the prime factor decomposition

$$x^n - 1 = \pi_1 \cdot \ldots \cdot \pi_r \tag{3.9}$$

are pairwise different. Without loss of generality we assume that they are ordered in the following way:

$$\deg_x \pi_1 = \ldots = \deg_x \pi_{r_1} < \ldots < \deg_x \pi_{1+\sum_{i=1}^{s-1} r_i} = \ldots = \deg_x \pi_{\sum_{i=1}^{s} r_i} \tag{3.10}$$

where $r_1 + \ldots + r_s = r$. For later use we group the indices of the prime factors according to their (the prime factors) degree:

$$R^{(j)} := \{1 + \sum_{\mu=1}^{j-1} r_\mu, 2 + \sum_{\mu=1}^{j-1} r_\mu, \ldots, \sum_{\mu=1}^{j} r_\mu\} \quad , \quad 1 \leq j \leq s \quad . \tag{3.11}$$

Of course, the sets $R^{(1)}, \ldots, R^{(s)}$ define a disjoint partition of $\{1, \ldots, r\}$ and $|R^{(j)}| = r_j$ for $1 \leq j \leq s$.

Now $F[x]/\langle\, \pi_k \,\rangle$ is a finite field, precisely it is a finite Galois extension of $\mathbb{F}$ of dimension $\deg_x \pi_k$ for $1 \leq k \leq r$ and it is isomorphic to the field

$$K_k := \{f \in \mathbb{F}[x] \mid \deg_x f < \deg_x \pi_k\} \text{ with multiplication modulo } \pi_k \quad .$$

We denote by $\rho_k(a) \in K_k$ the remainder of $a \in \mathbb{F}[x]$ modulo π_k and consider the mapping

$$\rho : A \to K_1 \times \ldots \times K_r \quad , \quad a \mapsto [\rho_1(a), \ldots, \rho_r(a)] \quad , \tag{3.12}$$

which is an isomorphism of rings due to the Chinese Remainder Theorem, where the cartesian product is endowed with component-wise addition and multiplication. We can identify A with $K_1 \times \ldots \times K_r$, but to ensure that we do not mix up vectors and elements of the cartesian product we use the notation

$$[\![a_1, \ldots, a_r]\!] := \rho^{-1}([a_1, \ldots, a_r]) \quad . \tag{3.13}$$

The elements

$$\varepsilon^{(k)} := [\![0,\ldots,0,1,0,\ldots,0]\!] \text{ , where the "1" is at the } k\text{-th position , } \quad (3.14)$$

will be an important tool to investigate cyclic convolutional codes. They are the uniquely determined primitive[8] and pairwise orthogonal idempotents of A. The **k-th component** of A, which is defined as

$$K^{(k)} := \varepsilon^{(k)} A = \{\rho^{-1}(a) \mid a \in 0 \times \ldots \times 0 \times K_k \times 0 \times \ldots \times 0\} \quad , \quad (3.15)$$

is a field which is isomorphic to K_k. Hence $A = \sum_{i=1}^{r} K^{(i)}$. Any ideal of A (these "are" the cyclic block codes) can be written as

$$\sum_{i=1}^{r} U_i \quad , \quad \text{where} \quad U_i \in \{\{0\}, K^{(i)}\} \quad \text{for} \quad 1 \le i \le r \quad .$$

Finally we consider the group $Aut_{\mathbb{F}}(A)$ and give some more information on its structure. The basic idea is that any automorphism $\sigma \in Aut_{\mathbb{F}}(A)$ maps $K^{(k)}$ onto $K^{(l)}$ for some l, i.e. σ induces a permutation on the set $\{K^{(1)},\ldots,K^{(r)}\}$. Furthermore, $K^{(k)}$ and $K^{(l)}$ must have the same cardinality (this means $\deg_x \pi_k = \deg_x \pi_l$). This works (loosely speaking) the other way round, too: Any permutation of the fields $\{K^{(1)},\ldots,K^{(r)}\}$ which respects their cardinality combined with $\mathbb{F}$-linear automorphisms of the fields themselves induces an $\mathbb{F}$-automorphism of A. We specify this in the following theorem.

Theorem 3.8 *[GS02a, Theorem 3.1]*

(a) There exist (usually non-unique) $\mathbb{F}$-linear automorphisms of fields $\psi_1,\ldots,\psi_r$ such that with $\psi = (\psi_1,\ldots,\psi_r)$ the mapping

$$\psi \circ \rho \ : \ A \ \stackrel{\rho}{\to} \ K_1 \times \ldots \times K_r \ \stackrel{(\psi_1,\ldots,\psi_r)}{\to} \ L_1^{r_1} \times \ldots \times L_s^{r_s}$$

is a ring-automorphism and L_j is a finite Galois extension of $\mathbb{F}$ which is isomorphic to K_k for any $k \in R^{(j)}$.

(b) Let the data be as in (a). Let $G_j := Aut_{\mathbb{F}}(L_j)$ be the group of all $\mathbb{F}$-linear automorphisms of L_j for $1 \le j \le s$ and let $S_{r_1,\ldots,r_s}$ be the subgroup of the symmetric group S_r, which leaves all sets $R^{(1)},\ldots,R^{(s)}$ invariant. Then the group $Aut_{\mathbb{F}}(L_1^{r_1} \times \ldots \times L_s^{r_s})$ is given by

$$(G_1^{r_1} \times \ldots \times G_s^{r_s}) \circ S_{r_1,\ldots,r_s} \quad ,$$

[8] An idempotent is called *primitive* if it cannot be written as a nontrivial sum of orthogonal idempotents.

where "$\circ$" is defined as

$$((\gamma_1,\ldots,\gamma_r)\circ\beta)[a_1,\ldots,a_r] := [\gamma_1(a_{\beta^{-1}(1)}),\ldots,\gamma_r(a_{\beta^{-1}(r)})]$$

for all $[a_1,\ldots,a_r]\in L_1^{r_1}\times\ldots\times L_s^{r_s}$ *and all* $(\gamma_1,\ldots,\gamma_r)\in G_1^{r_1}\times\ldots\times G_s^{r_s}$, $\beta\in S_{r_1,\ldots,r_s}$.

(c) *By virtue of (a) the group* $Aut_{\mathbb{F}}(A)$ *and the group* $Aut_{\mathbb{F}}(L_1^{r_1}\times\ldots\times L_s^{r_s})$ *are isomorphic. An isomorphism is given by*

$$\begin{array}{rccl}\widehat{\ }: & Aut_{\mathbb{F}}(L_1^{r_1}\times\ldots\times L_s^{r_s}) & \to & Aut_{\mathbb{F}}(A)\\ & \gamma\circ\beta & \mapsto & \widehat{\gamma\circ\beta} := \rho^{-1}\circ\psi^{-1}\circ\gamma\circ\beta\circ\psi\circ\rho\end{array} .$$

The following commutative diagram shall illustrate this:

$$\begin{array}{rcl} A & \xrightarrow{\widehat{\gamma\circ\beta}} & A\\ \rho\downarrow & & \uparrow\rho^{-1}\\ K_1\times\ldots\times K_r & & K_1\times\ldots\times K_r\\ \psi=(\psi_1,\ldots,\psi_r)\downarrow & & \uparrow(\psi_1^{-1},\ldots,\psi_r^{-1})=\psi^{-1}\\ L_1^{r_1}\times\ldots\times L_s^{r_s} & \xrightarrow{\gamma\circ\beta} & L_1^{r_1}\times\ldots\times L_s^{r_s}\end{array}$$

In other words: For all $[\![a_1,\ldots,a_r]\!]\in A$ *we have*

$$\begin{array}{rl} & (\widehat{\gamma\circ\beta})([\![a_1,\ldots,a_r]\!])\\ = & [\![\psi_1^{-1}(\gamma_1(\psi_{\beta^{-1}(1)}(a_{\beta^{-1}(1)}))),\ldots,\psi_r^{-1}(\gamma_r(\psi_{\beta^{-1}(r)}(a_{\beta^{-1}(r)})))]\!] .\end{array}$$

□

As a consequence of the foregoing theorem, we can determine the cardinality of $Aut_{\mathbb{F}}(A)$:

Corollary 3.9 *[GS02a, Corollary 3.2]*
Let L_j *be as in Theorem 3.8 and put* $d_j := \deg_x \pi_i$ *where* $i\in R^{(j)}$, $1\le j\le s$. *Then*

$$|Aut_{\mathbb{F}}(A)| = d_i^{r_1}\cdots d_s^{r_s}\cdot r_1!\ldots r_s! .$$

□

The group-isomorphism which we specified in Theorem 3.8 (c) appears to be a bit cumbersome. But it has a nice property: The primitive idempotents $\varepsilon^{(1)},\ldots,\varepsilon^{(r)}$ are permuted by $\widehat{\gamma\circ\beta}\in Aut_{\mathbb{F}}(A)$ according to the permutation β:

Corollary 3.10
Let the data be given as in Theorem 3.8 and choose $\widehat{\gamma \circ \beta} \in Aut_{\mathbb{F}}(A)$. *Then we have*

$$(\widehat{\gamma \circ \beta})(\varepsilon^{(i)}) = \varepsilon^{(\beta(i))} \quad \textit{for all } 1 \leq i \leq r \ .$$

In particular, the image of $\varepsilon^{(i)}$ *does not depend on the choice of* $(\psi_1, \ldots, \psi_r)$, *it only depends on the permutation* β.

PROOF: Choose a fixed $1 \leq i \leq r$. Then $\varepsilon^{(i)} = [\![\delta_{i,j}]\!]_{1 \leq j \leq r}$, where $\delta_{i,j}$ is the Kronecker symbol[9]. Note that $\psi_j(1) = 1$ and $\gamma_j(1) = 1$ resp. $\psi_j(0) = 0$ and $\gamma_j(0) = 0$ for $1 \leq j \leq r$, since ψ_j and γ_j are isomorphisms of fields. The latter implies

$$\begin{aligned}
& (\widehat{\gamma \circ \beta})(\varepsilon^{(i)}) \\
= \ & [\![\psi_1^{-1}(\gamma_1(\psi_{\beta^{-1}(1)}(\delta_{i,\beta^{-1}(1)}))), \ldots, \psi_r^{-1}(\gamma_r(\psi_{\beta^{-1}(r)}(\delta_{i,\beta^{-1}(r)})))]\!] \\
= \ & [\![\psi_1^{-1}(\gamma_1(\delta_{i,\beta^{-1}(1)})), \ldots, \psi_r^{-1}(\gamma_r(\delta_{i,\beta^{-1}(r)}))]\!] \\
= \ & [\![\psi_1^{-1}(\delta_{i,\beta^{-1}(1)}), \ldots, \psi_r^{-1}(\delta_{i,\beta^{-1}(r)})]\!] \\
= \ & [\![\delta_{i,\beta^{-1}(1)}, \ldots, \delta_{i,\beta^{-1}(r)}]\!] = [\![\delta_{i,\beta^{-1}(j)}]\!]_{1 \leq j \leq r} = \varepsilon^{(\beta(i))} \ .
\end{aligned}$$

□

For later purposes we like to point out that the field L_1 in Theorem 3.8 is isomorphic to $\mathbb{F}$, hence $Aut_{\mathbb{F}}(L_1) = \{\mathrm{id}\}$ where "id" denotes the identity mapping. For the same reason the choice of the $\mathbb{F}$-linear isomorphisms $\psi_1, \ldots, \psi_{r_1}$ is unique: All of them must be the identity mapping. Therefore we know exactly how $\widehat{\gamma \circ \beta} \in Aut_{\mathbb{F}}(A)$ acts on the first r_1 components of A:

Remark 3.11
For any choice of $(\psi_1, \ldots, \psi_r)$ *and* $\widehat{\gamma \circ \beta} \in Aut_{\mathbb{F}}(A)$ *as in Theorem 3.8 and any* $[\![a_1, \ldots, a_{r_1}, \ldots]\!] \in A$ *we have:*

$$\begin{aligned}
& (\widehat{\gamma \circ \beta})([\![a_1, \ldots, a_{r_1}, \ldots]\!]) \\
= \ & [\![\psi_1^{-1}(\gamma_1(\psi_{\beta^{-1}(1)}(a_{\beta^{-1}(1)}))), \ldots, \psi_r^{-1}(\gamma_{r_1}(\psi_{\beta^{-1}(r)}(a_{\beta^{-1}(r)}))), \ldots]\!] \\
= \ & [\![\mathrm{id}(\mathrm{id}(\mathrm{id}(a_{\beta^{-1}(1)}))), \ldots, \mathrm{id}(\mathrm{id}(\mathrm{id}(a_{\beta^{-1}(r)}))), \ldots]\!] \\
= \ & [\![a_{\beta^{-1}(1)}, \ldots, a_{\beta^{-1}(r)}, \ldots]\!] \ .
\end{aligned}$$

In the following example we shall illustrate some of the recent results:

[9] We have $\delta_{i,j} = \begin{cases} 0 & \text{for } i \neq j \\ 1 & \text{for } i = j \end{cases}$.

Example 3.12
Again, we consider $\mathbb{F} = \mathbb{F}_4 = \{0, 1, \alpha, \alpha^2\}$ *and* $n = 3$. *The polynomial* $x^n - 1$ *splits into linear prime divisors, precisely* $x^n - 1 = (x+1)(x+\alpha)(x+\alpha^2)$. *We display* A *as*

$$\rho^{-1}\Big(\mathbb{F}[x]/\langle\, x+1 \,\rangle \;\times\; \mathbb{F}[x]/\langle\, x+\alpha \,\rangle \;\times\; \mathbb{F}[x]/\langle\, x+\alpha^2 \,\rangle \Big) \quad .$$

Now $\varepsilon^{(1)} = 1 + x + x^2$, $\varepsilon^{(2)} = 1 + \alpha^2 x + \alpha x^2$, $\varepsilon^{(3)} = 1 + \alpha x + \alpha^2 x^2$, *as one can easily check by computing* $\varepsilon^{(i)}(1)$, $\varepsilon^{(i)}(\alpha)$, $\varepsilon^{(i)}(\alpha^2)$ *for* $i = 1, 2, 3$. *We know from Remark 3.5 that* $Aut_{\mathbb{F}}(A) = \{\sigma \mid \sigma(x) \in \{x, x^2, \alpha x, \alpha^2 x, \alpha x^2, \alpha^2 x^2\}\}$.
One can readily see that

$$\begin{array}{c} [\![1, \alpha, \alpha^2]\!] = x \;,\; [\![\alpha, \alpha^2, 1]\!] = \alpha x \;,\; [\![\alpha^2, 1, \alpha]\!] = \alpha^2 x \;, \\ [\![1, \alpha^2, \alpha]\!] = x^2 \;,\; [\![\alpha, 1, \alpha^2]\!] = \alpha x^2 \;,\; [\![\alpha^2, \alpha, 1]\!] = \alpha^2 x^2. \end{array}$$

Theorem 3.8 tells us that $Aut_{\mathbb{F}}(A) \cong S_3$. *Actually, we get all automorphisms in* $Aut_{\mathbb{F}}(A)$ *if we permute the components in* $x = [\![1, \alpha, \alpha^2]\!]$.
With Remark 3.11 we know that, for instance, the automorphism σ *defined by* $\sigma(x) = \alpha x$ *satisfies* $\sigma(\varepsilon^{(1)}) = \varepsilon^{(3)}$, $\sigma(\varepsilon^{(2)}) = \varepsilon^{(1)}$ *and* $\sigma(\varepsilon^{(3)}) = \varepsilon^{(2)}$.

In the foregoing example, the fields $\mathbb{F}[x]/\langle\, \pi_i \,\rangle$ were isomorphic to $\mathbb{F}$. For a more general example where $K^{(i)} \ncong \mathbb{F}$ for some $1 < i \leq n$ we refer to [GS02a, Example 3.3 (b)].

3.2.3 Properties of Cyclic Convolutional Codes

In this section we consider a fixed automorphism $\sigma \in Aut_{\mathbb{F}}(A)$ and a fixed representation of A as a cartesian product of fields $K_1 \times \ldots \times K_r$. Let the data be as in (3.9) – (3.15).

Furthermore, we will omit the symbol " $*_\sigma$ " when multiplying polynomials in the non-commutative $\mathbb{F}$-algebra $A[z; \sigma]$, i.e.

$$gh := g *_\sigma h \quad \text{for all} \quad g, h \in A[z; \sigma] \quad . \tag{3.16}$$

Principal left ideals in $A[z; \sigma]$ will play a crucial role (cf. Proposition 3.14) and we introduce the following notation:

Notation 3.13
A principal left ideal in $A[z; \sigma]$ *generated by* $g \in A[z; \sigma]$ *is denoted by* ${}^{\bullet}\langle\, g \,\rangle$.

Proposition 3.14 *cf. [GS02a, Theorem 4.5]*
Let $\sigma \in Aut_{\mathbb{F}}(A)$. A submodule $\mathcal{C} \subseteq \mathbb{F}[z]^n$ is a σ-cyclic convolutional code if and only if it is a convolutional code and $\mathfrak{p}(\mathcal{C})$ is a principal left ideal in $A[z;\sigma]$, i.e. there exists $g \in A[z;\sigma]$ such that $\mathfrak{p}(\mathcal{C}) = {}^{\bullet}\langle g \rangle$. □

The statement (3.5), which we introduced for cyclic block codes in Section 3.1, can be generalized to σ-cyclic convolutional codes. In order to give these results in Theorem 3.18 which are due to Glüsing-Lüerßen and Schmale [GS02a], we have to go through some technicalities beforehand.

Definition 3.15 *[GS02a, Notation 4.1, Def. 4.7, Def. 4.9]*
We consider a polynomial $g \in A[z;\sigma]$. (Note that we make use of notation (3.16) in the following.)

(a) The polynomial $g^{(k)} := \varepsilon^{(k)} g \in A[z;\sigma]$ is called the **k-th component** *of g for $1 \leq k \leq r$ and the set $T_g := \{k \mid g^{(k)} \neq 0\ ,\ 1 \leq k \leq r\}$ is called the* **support** *of g. If $g = g^{(k)}$ for some $1 \leq k \leq r$ then g called a* **component**.

(b) With $g = \sum_{i=0}^{d} z^i g_i$ we have $g^{(k)} = \sum_{i=0}^{d} z^i \sigma^i(\varepsilon^{(k)}) g_i$ for $1 \leq k \leq n$.
The individual summands $\varepsilon^{(k)}(z^i g_i)$ are called the **terms** *of $g^{(k)}$. The* **(left-)leading monomial** *of $g^{(k)}$, denoted by $LM(g^{(k)})$, is the monomial $z^m \sigma^m(\varepsilon^{(k)})$ satisfying $m = \max\{0 \leq i \leq d \mid z^i \sigma^i(\varepsilon^{(k)}) g_i \neq 0\}$.*

(c) The polynomial g is called **left-reduced** *if for all $1 \leq k, l \leq r$ such that $k \neq l$ and $g^{(k)} \neq 0 \neq g^{(l)}$ no nonzero term of $g^{(k)}$ is right divisible[10] by $LM(g^{(l)})$.*
Note that g is always left-reduced if it is a component.

To get more light on this point, we give an example:

Example 3.16

(a) Once more we consider $\mathbb{F} = \mathbb{F}_4 = \{0, 1, \alpha, \alpha^2\}$ and $n = 3$. Let A be displayed as in Example 3.12 and let g, σ be given as in Example 3.4, i.e. $\sigma(x) = \alpha^2 x$. We know that $\varepsilon^{(1)} = 1 + x + x^2$, $\varepsilon^{(2)} = 1 + \alpha^2 x + \alpha x^2$, $\varepsilon^{(3)} = 1 + \alpha x + \alpha^2 x^2$, hence

$$g = \varepsilon^{(3)} + z\varepsilon^{(1)} + z^2\varepsilon^{(2)} \quad .$$

We can compute $\sigma(\varepsilon^{(1)}) = \varepsilon^{(2)}$, $\sigma(\varepsilon^{(2)}) = \varepsilon^{(3)}$ and $\sigma(\varepsilon^{(3)}) = \varepsilon^{(1)}$, hence

$$g^{(3)} = \varepsilon^{(3)}\varepsilon^{(3)} + z\sigma(\varepsilon^{(3)})\varepsilon^{(1)} + z^2\sigma^2(\varepsilon^{(3)})\varepsilon^{(2)} = g.$$

[10] Let $(R, +, \cdot)$ be a ring and $a, b \in R$. Then a is said to be right divisible by b if there exists $c \in R$ such that $cb = a$.

This implies that $g^{(1)} = g^{(2)} = 0$ *and that* g *is a component. Of course* g *is left-reduced.*

(b) *We continue with the example from part (a) but consider* $\hat{g}$ *where*

$$\hat{g} = \varepsilon^{(1)} + z\varepsilon^{(1)} + z^2\varepsilon^{(1)} \quad .$$

Now $\hat{g}^{(1)} = \varepsilon^{(1)}$, $\hat{g}^{(2)} = z^2\varepsilon^{(1)}$ *and* $\hat{g}^{(3)} = z\varepsilon^{(1)}$. *The leading monomial of* $\hat{g}^{(1)}$ *is* $LM(\hat{g}^{(1)}) = \varepsilon^{(1)}$ *and the terms of* $\hat{g}^{(2)}$ *and* $\hat{g}^{(3)}$ *are right divisible by* $\varepsilon^{(1)}$. *This shows that* $\hat{g}$ *is not reduced.*
If we consider $\mathcal{C} := {}^{\bullet}\langle\, \hat{g} \,\rangle$, *we have*

$$\mathcal{C} = {}^{\bullet}\langle\, \hat{g}^{(1)}, \hat{g}^{(2)}, \hat{g}^{(3)} \,\rangle = {}^{\bullet}\langle\, \hat{g}^{(1)}, \underbrace{\hat{g}^{(2)} - z^2\hat{g}^{(1)}}_{=0}, \underbrace{\hat{g}^{(3)} - z\hat{g}^{(1)}}_{=0} \rangle = {}^{\bullet}\langle\, \hat{g}^{(1)} \,\rangle \quad .$$

Hence $\mathcal{C}$ *can be generated by* $\hat{g}^{(1)} = \varepsilon^{(1)}$, *which is a left-reduced polynomial.*

(c) *Finally we give an example for a left-reduced polynomial which is not a component. We consider* g *as in (a), but replace* σ *with the identity. Now* $g^{(1)} = z\varepsilon^{(1)}$, $g^{(2)} = z^2\varepsilon^{(2)}$ *and* $g^{(3)} = \varepsilon^{(3)}$ *and no term of* $g^{(i)}$ *is right divisible by* $LM(g^{(j)})$ *for* $i, j = 1, 2, 3$, $i \neq j$, *therefore* g *is left-reduced.*

Theorem and Definition 3.17 *[GS02a, Cor. 4.13, Thm. 4.15 (b), Lemma 4.3 (c)]*

(a) *Let* $\mathcal{C}$ *be a* σ*-cyclic convolutional code. Then there exists a left-reduced polynomial* $g \in A[z;\sigma]$ *such that* $\mathfrak{p}(\mathcal{C}) = {}^{\bullet}\langle\, g \,\rangle$. *Any left reduced polynomial with this property is called a* **generator polynomial** *of* $\mathcal{C}$ *(resp.* $\mathfrak{p}(\mathcal{C})$*).*

(b) *If* g *is a generator polynomial of* $\mathcal{C}$ *and* $\mathfrak{p}(\mathcal{C}) = {}^{\bullet}\langle\, \hat{g} \,\rangle$ *for some* $\hat{g} \in A[z;\sigma]$ *(which is not necessarily left-reduced), then* $g = u\hat{g}$ *for some unit* $u \in A[z;\sigma]$. *If* $\hat{g}$ *is left-reduced, then* $g = u\hat{g}$ *for some unit* $u \in A$.
In particular, the support of a generator polynomial does not depend of its choice and we call T_g *the* **support** *of* $\mathcal{C}$ *(resp.* $\mathfrak{p}(\mathcal{C})$*).*

Theorem 3.18 *[GS02a, Theorem 7.8, Theorem 7.13]*
Let $\mathcal{C} \subseteq \mathbb{F}[z]^n$ *be a* σ*-cyclic submodule and let* $\mathfrak{p}(\mathcal{C}) = {}^{\bullet}\langle\, g \,\rangle$, *where* g *is left-reduced. Define* $\pi_{(g)} := \prod_{k \in T_g} \pi_k$, $\kappa_k := \deg_x \pi_k$ *and* $\kappa := \deg_x \pi_{(g)}$. *Then we have*

(a) *A left* $\mathbb{F}[z]$*-basis of* ${}^{\bullet}\langle\, g \,\rangle$ *is given by* $g, xg, \ldots, x^{\kappa-1}g$. *As a consequence, the matrix*

$$\begin{bmatrix} \mathfrak{v}(g) \\ \mathfrak{v}(xg) \\ \vdots \\ \mathfrak{v}(x^{\kappa-1}g) \end{bmatrix} \in \mathbb{F}^{\kappa \times n}$$

is a generator matrix of $\mathcal{C}$, in particular, $\mathcal{C}$ is an (n,κ)-submodule.

(b) Let $k \in T_g$. The matrix

$$G_k := \begin{bmatrix} \mathfrak{v}(g^{(k)}) \\ \mathfrak{v}(xg^{(k)}) \\ \vdots \\ \mathfrak{v}(x^{\kappa_k-1}g^{(k)}) \end{bmatrix} \in \mathbb{F}^{\kappa_k \times n}$$

is a minimal basis of the σ-cyclic (n,κ_k)-submodule $\mathfrak{v}({}^{\bullet}\langle\, g^{(k)} \,\rangle)$. If $\mathcal{C}$ is a convolutional code, then $\mathfrak{v}({}^{\bullet}\langle\, g^{(k)} \,\rangle)$ is a (σ-cyclic) convolutional code, too.

(c) If $T_g = \{k_1, \ldots, k_t\}$ and $G_{k_1}, \ldots, G_{k_t}$ are defined as in (b), then

$$\begin{bmatrix} G_{k_1} \\ \vdots \\ G_{k_t} \end{bmatrix} \in \mathbb{F}^{\kappa \times n}$$

is a minimal basis of $\mathcal{C}$. □

The foregoing theorem allows us not only to determine a generator matrix or a minimal basis of $\mathcal{C}$, but also the complexity of $\mathcal{C}$ and its Forney indices (recall Theorem 1.3). This is specified in the following corollary.

Corollary 3.19 *[GS02a, Corollary 7.15]*
Let $\mathcal{C} \subseteq \mathbb{F}[z]^n$ be a σ-cyclic convolutional code and let g be a generator polynomial. Furthermore, let $T_g = \{k_1, \ldots, k_t\}$ denote the support of $\mathcal{C}$ and write $\kappa_{k_i} := \deg_x \pi_{k_i}$ and $\delta_{k_i} := \deg_x g^{(k_i)}$ for $1 \le i \le t$. Then $\mathcal{C}$ has the Forney indices

$$\underbrace{\delta_{k_1}, \ldots, \delta_{k_1}}_{\kappa_{k_1} \text{ times}}, \ldots, \underbrace{\delta_{k_t}, \ldots, \delta_{k_t}}_{\kappa_{k_t} \text{ times}}$$

and $\mathcal{C}$ has the complexity $\delta = \sum_{i \in T_g} \kappa_i \delta_{k_i}$.
In particular, $\mathfrak{v}({}^{\bullet}\langle\, g^{(k_i)} \,\rangle)$ is a σ-cyclic (n,κ_{k_i})-convolutional code with Forney indices $\delta_{k_i}, \ldots, \delta_{k_i}$ (κ_{k_i} times) and complexity $\kappa_{k_i}\delta_{k_i}$ for $1 \le i \le t$. □

The next two propositions give more information about the difference between arbitrary σ-cyclic submodules on the one hand and σ-cyclic convolutional codes on the other hand.

We saw in Proposition 3.14 that $\mathfrak{p}(\mathcal{C})$ is a principal left ideal in $A[z;\sigma]$ if $\mathcal{C}$ is a σ-cyclic convolutional code. But not every principal left ideal in $A[z;\sigma]$

defines a σ-cyclic convolutional code. A characterization of those principal left ideals which are σ-cyclic convolutional codes as well is given in the following proposition:

Proposition 3.20 *[GS02a, Proposition 7.10]*
Let $\mathcal{C} \subseteq \mathbb{F}[z]^n$ be a submodule where $\mathfrak{p}(\mathcal{C}) = {}^{\bullet}\langle\, g \,\rangle$ is a principal left ideal in $A[z;\sigma]$ and $g \in A[z;\sigma]$ is left-reduced with z-free term g_0. Then $\mathcal{C}$ is a (σ-cyclic) convolutional code if and only if there exists some unit u in $A[z;\sigma]$ such that $g = g_0 u$. □

A necessary condition for a σ-cyclic submodule to be a σ-cyclic convolutional code is given in the next proposition and concerns the support of this code. It is used in the proof of Theorem 4.2.

Proposition 3.21 *[GS02a, Thm. 4.5, Thm. 5.1 (c)]*
Let $\mathcal{C} \subseteq \mathbb{F}[z]^n$ be a σ-cyclic convolutional code and let g be a generator polynomial with z-free term g_0. Then we have $T_g = T_{g_0}$, i.e.

$$\varepsilon^{(k)} g \neq 0 \iff \varepsilon^{(k)} g_0 \neq 0 \quad \text{for all} \quad 1 \leq k \leq r \quad .$$

Theorem 3.18 (b), (c) also provides a canonical decomposition of a σ-cyclic convolutional code into a direct sum of "small" σ-cyclic convolutional sub-codes:

Corollary 3.22
Let $\mathcal{C} \subseteq \mathbb{F}[z]^n$ be a σ-cyclic convolutional code and let g be a generator polynomial. Then we have the following decomposition of $\mathcal{C}$ and $\mathfrak{p}(\mathcal{C})$ into a direct sum of $\mathbb{F}[z]$-left modules:

$$\mathcal{C} = \bigoplus_{i \in T_g} \mathfrak{v}({}^{\bullet}\langle\, g^{(i)} \,\rangle) \quad , \quad \mathfrak{p}(\mathcal{C}) = \bigoplus_{i \in T_g} {}^{\bullet}\langle\, g^{(i)} \,\rangle$$

Furthermore, the summands in the decomposition of $\mathcal{C}$ are σ-cyclic convolutional codes themselves. □

4 Minimal Cyclic Convolutional Codes

Recall that we always assume that (3.1) holds. Furthermore, we consider a fixed automorphism $\sigma \in Aut_{\mathbb{F}}(A)$ and a fixed representation of A as a cartesian product of fields $K_1 \times \ldots \times K_r$. Let the data be given as in (3.9) – (3.15). The multiplication in the $\mathbb{F}$-algebra $A[z;\sigma]$ will be denoted as in (3.16).

We will introduce the notion of minimal σ-cyclic convolutional codes in Section 4.1. It will turn out that units in $A[z;\sigma]$ play a crucial role in the investigation of minimal σ-cyclic convolutional codes (cf. Theorem 4.6) and therefore we give some information about units in $A[z;\sigma]$ in Section 4.2. In Section 4.3 we state our main result, this is Theorem 4.15 which gives necessary and sufficient conditions for the existence of minimal cyclic convolutional codes with nonzero complexity. Consequences of this Theorem are discussed.

4.1 Minimality and First Properties

Piret and Roos investigated in [Pir76] and [Roo79] how a σ-cyclic convolutional code can be decomposed into a direct sum of "minimal" or "irreducible" σ-cyclic sub-codes. They used the following notion of minimality:

Definition 4.1
A σ-cyclic convolutional code $\mathcal{C}$ is called **minimal** *if and only if it has no nontrivial σ-cyclic convolutional sub-code, this means for any σ-cyclic convolutional code $\hat{\mathcal{C}}$ where $\{0\} \neq \hat{\mathcal{C}} \subseteq \mathcal{C}$ we have $\hat{\mathcal{C}} = \mathcal{C}$. In this case we call $\mathfrak{p}(\mathcal{C})$* **minimal**, *too.*

The following theorem is crucial for our further investigations. It states that σ-cyclic convolutional codes, whose support has cardinality 1, coincide with minimal σ-cyclic convolutional codes:

Theorem 4.2
Let $\mathcal{C}$ be a σ-cyclic convolutional code with generator polynomial $g \in A[z;\sigma]$. Then the following statements are equivalent:
(i) $\mathcal{C}$ is minimal;
(ii) g is a component, i.e. there exists $l \in \{1,\ldots,r\}$ such that $g = \varepsilon^{(l)}g = g^{(l)}$.

PROOF: The case $\mathcal{C} = \{0\}$ is trivial, hence we assume $\mathcal{C} \neq \{0\}$.
"(ii)⇒(i)": We have $0 \neq g = g^{(l)}$ and $T_g = T_{g_0} = \{l\}$ (cf. Proposition 3.21), where $g_0 \in \varepsilon^{(l)}A$ denotes the z-free term of g . Let $\{0\} \neq U \subseteq \mathcal{C}$ be a σ-cyclic sub-code of $\mathcal{C}$ with generator polynomial $0 \neq h \in A[z;\sigma]$. Since $\mathfrak{p}(U) = {}^{\bullet}\langle\, h \,\rangle \subseteq {}^{\bullet}\langle\, g \,\rangle$, we have $h = ug$ for some $u \in A[z;\sigma]$, in particular we have $h_0 = u_0 g_0 \in \varepsilon^{(l)}A$, where u_0 resp. h_0 denotes the z-free term of u resp. h. Proposition 3.21 implies that $T_h = T_{h_0}$, thus $T_h = \{l\}$ and $0 \neq h = h^{(l)}$. Theorem 3.18 (b) yields that U and $\mathcal{C}$ have the same rank as $\mathbb{F}[z]$-modules and with Theorem 1.7 (vii) we conclude $\mathcal{C} = U$.

"(i)⇒(ii)": Let $T_g = \{k_1, \ldots, k_t\}$ where $t \geq 2$. Corollary 3.22 yields that $\{0\} \neq \mathfrak{v}({}^{\bullet}\langle\, g^{(k_i)} \,\rangle) \subseteq \mathcal{C}$ is a σ-cyclic convolutional sub-code of $\mathcal{C}$ for $1 \leq i \leq t$ and it satisfies $\mathfrak{v}({}^{\bullet}\langle\, g^{(k_i)} \,\rangle) \neq \mathcal{C}$. □

A main structural result of Piret [Pir76, Theorem 3.11] and Roos [Roo79, Theorem 4 and the following text] is that every σ-cyclic convolutional code has a decomposition into a direct sum of minimal σ-cyclic sub-codes. Thus Theorem 3.18 resp. Corollary 3.22 along with Theorem 4.2 recovers this result and yields a method to construct such a decomposition, too.

Remark 4.3
The decomposition of a σ-cyclic convolutional code $\mathcal{C}$ with generator polynomial g as in Corollary 3.22, i.e. $\mathcal{C} = \bigoplus_{i \in T_g} \mathfrak{v}({}^{\bullet}\langle\, g^{(i)} \,\rangle)$, is a decomposition into a direct sum of minimal σ-cyclic convolutional codes (as $\mathbb{F}[z]$-left modules). Moreover, it is obvious that a convolutional code $\mathcal{C}$ is σ-cyclic if and only if it is a direct sum of minimal σ-cyclic convolutional codes (as $\mathbb{F}[z]$-left modules). In this sense we can think of minimal σ-cyclic convolutional codes as the "smallest components" of σ-cyclic convolutional codes.

The next remark recalls how we can determine the dimension and complexity (resp. the Forney indices) of a minimal σ-cyclic convolutional code.

Remark 4.4
Let $\mathcal{C}$ be a minimal σ-cyclic convolutional code with support $\{l\}$ and generator polynomial $g = g^{(l)}$. Define $k := \deg_x \pi_l$ and $d := \deg_z g$. Then Corollary 3.19 yields that $\mathcal{C}$ is an (n, k, kd)-submodule with Forney indices $d, \ldots, d$ (k times). In particular, the complexity of a minimal σ-cyclic convolutional code must be a multiple of its dimension.

We know that the support of a (minimal) σ-cyclic convolutional code is unique (cf. Theorem and Definition 3.17). The following remark allows us to

determine the support of a minimal σ-cyclic convolutional with the aid of a minimal basis.

Remark 4.5
Let $\mathcal{C}$ be a minimal σ-cyclic (n,k)-convolutional code with support $\{l\}$ and let G be a minimal basis of $\mathcal{C}$ with rows $G_1,\ldots,G_k$. Then one has

$$\{T_{\mathfrak{p}(G_1)},\ldots,T_{\mathfrak{p}(G_k)}\} = \{l\}$$

and the support of $\mathcal{C}$ can be determined with the knowledge of only one row of a minimal basis.

PROOF: Let $g = g^{(l)}$ be a generator polynomial of $\mathcal{C}$ and define $\hat{G}_i := \mathfrak{v}(x^{i-1}g)$ for $1 \le i \le k$. Note that $T_{\mathfrak{p}(\hat{G}_i)} = \{l\}$, since x is a unit in A. With Theorem 3.18 (c) we know that the matrix $\hat{G}$ (consisting of the rows $\hat{G}_1,\ldots,\hat{G}_k$) is a minimal basis of $\mathcal{C}$, too. Since all row degrees of G resp. $\hat{G}$ are the same, Theorem 1.3 (iv) implies that G_i must be an $\mathbb{F}$-linear combination of $\hat{G}_1,\ldots,\hat{G}_k$ for $1 \le i \le k$. Therefore we have $T_{\mathfrak{p}(G_i)} = \{l\}$. □

At this point, we know that a minimal σ-cyclic convolutional code is generated by a component and we know how to determine the dimension, the complexity and the support of the code. But we do not know how we can tell from a given polynomial $g = g^{(l)}$ if the σ-cyclic submodule $\mathfrak{v}({}^{\bullet}\langle\, g\,\rangle)$ is a convolutional code. The following theorem answers this question.

Theorem 4.6
Let $\{0\} \neq \mathcal{C}$ be a σ-cyclic submodule of $\mathbb{F}[z]^n$ and let $\mathfrak{p}(\mathcal{C}) = {}^{\bullet}\langle\, g\,\rangle$ where $g = g^{(l)} \in A[z;\sigma]$ for some $1 \le l \le r$ (in particular g is left reduced). Then the following statements are equivalent:
(i) $\mathcal{C}$ is a (σ-cyclic, minimal) convolutional code;
(ii) $g = g^{(l)}$ is the l-th component of a unit $u \in A[z;\sigma]$, i.e. we have $u^{(l)} = g$.

In particular, any component $u^{(l)}$ of a unit $u \in A[z;\sigma]$ defines a minimal σ-cyclic (n,k,dk)-convolutional code, where $k := \deg_x \pi_l$ and $d := \deg_z u^{(l)}$.

PROOF: With Proposition 3.20 we know that the submodule ${}^{\bullet}\langle\, g\,\rangle$ is a convolutional code if and only if $g = g_0 v$ for some unit $v \in A[z;\sigma]$. In this case we have

$$g = g^{(l)} = \varepsilon^{(l)} g = \varepsilon^{(l)} g_0 v = g_0^{(l)} \varepsilon^{(l)} v = g_0^{(l)} v^{(l)} \quad .$$

Now define $\hat{g}_0 := g_0^{(l)} + \sum_{i=1, i\neq l}^{r} \varepsilon^{(i)}$. Then $\hat{g}_0$ is a unit in A, since $\varepsilon^{(i)}\hat{g}_0 \neq 0$ for $1 \leq i \leq r$ (cf. Lemma 4.8) and therefore $u := \hat{g}_0 v$ is a unit in $A[z;\sigma]$ and it satisfies $g = \varepsilon^{(l)} u = u^{(l)}$. □

The above leads to the question under which conditions there exist minimal σ-cyclic convolutional codes with nonzero complexity. We will answer this question in Theorem 4.15 after we collected enough information about units in $A[z;\sigma]$ and their components. But a first step towards the solution of this problem can be introduced at this very time.

Lemma 4.7
Let $\mathcal{C}$ be a minimal σ-cyclic convolutional code with support $\{l\}$ and generator polynomial $g = g^{(l)}$. Then we have

$$\varepsilon^{(l)} g = g\varepsilon^{(l)} \iff \mathcal{C} \text{ has complexity } 0 \ .$$

In particular we have: If $\mathcal{C}$ has nonzero complexity, then $\varepsilon^{(l)} g \neq g\varepsilon^{(l)}$. This also implies $\sigma(\varepsilon^{(l)}) \neq \varepsilon^{(l)}$.

PROOF: The proof of "⇒" stems from [GS02a, Proof of Proposition 3.4]: If $\varepsilon^{(l)} g = g = g\varepsilon^{(l)}$, then ${}^{\bullet}\langle g \rangle \subseteq {}^{\bullet}\langle \varepsilon^{(l)} \rangle =: J$. Since $T_g = T_{\varepsilon^{(l)}} = \{l\}$, the convolutional codes $\mathcal{C}$ and $\mathfrak{v}(J)$ have the same dimension (cf. Theorem 3.18), they are equal by Theorem 1.7 (vii). With $\deg_z \varepsilon^{(l)} = 0$ along with Remark 4.4 we conclude that $\mathcal{C}$ has complexity $\delta = 0$.
"⇐": Remark 4.4 implies that $\deg_z g = 0$, hence $g \in A$. Of course $\varepsilon^{(l)} g = g\varepsilon^{(l)}$ is true in this case. □

In other words Lemma 4.7 states that there can only exist (minimal) σ-cyclic convolutional codes with nonzero complexity if the automorphism

$$\sigma \text{ does not act like the identity mapping on } \varepsilon^{(1)}, \ldots, \varepsilon^{(r)} \ . \tag{4.1}$$

Roos stated in [Roo79, Section VI] that the necessary condition (4.1) is sufficient, but he gave only a sketch of the proof. A complete proof is given in [GS02a, Theorem 3.4 and its proof] and that proof shows that (4.1) implies the existence of a minimal σ-cyclic (n, k, k)-convolutional code. But what about other complexities?

4.2 Units in $A[z;\sigma]$

Units in $A[z;\sigma]$ (resp. their components) play a crucial role in the investigation of minimal cyclic convolutional codes (cf. Theorem 4.6). This section gives useful information about units in $A[z;\sigma]$. First of all, we recall how a unit in A looks like:

Lemma 4.8 *[GS02a, Lemma 4.3]*
An element $a \in A$ is a unit in A if and only if $\varepsilon^{(k)}a \neq 0$ for all $1 \leq k \leq r$. $\square$

For a characterization of units in $A[z;\sigma]$, we need the following definition and remark:

Definition 4.9
Consider $l \in \{1,\ldots,r\}$ and $\sigma \in Aut_{\mathbb{F}}(A)$. Then the **$l$-order of** *$\sigma$ is defined as $o_l(\sigma) := o_l := \min\{m \in \mathbb{N} \mid \sigma^m(\varepsilon^{(l)}) = \varepsilon^{(l)}\}$.*

Remark 4.10
For $l \in \{1,\ldots,r\}$, $\sigma \in Aut_{\mathbb{F}}(A)$ and $\sigma(\varepsilon^{(l)}) = \varepsilon^{(l')}$ we have $o_l = o_{l'}$.

The next theorem states (loosely speaking) that any unit in $A[z;\sigma]$ is a finite product of special, "simple" units. In this sense, we know all units and due to Theorem 4.6 we know all minimal σ-cyclic codes.

Definition and Theorem 4.11 *(Inspired by [GS02a, Proof of Corollary 4.13])*
(a) Consider $a \in A$, $d \in \mathbb{N}_0$ and $m \in \{1\ldots,r\}$. The element $u_{(d,a,\varepsilon^{(m)})} := 1 + z^d a\varepsilon^{(m)}$ satisfies $u_{(d,a,\varepsilon^{(m)})} = u_{(d,a^{(m)},\varepsilon^{(m)})}$. Furthermore, one has:

$$u_{(d,a,\varepsilon^{(m)})} \text{ is a unit in } A[z;\sigma] \iff \begin{cases} a^{(m)} \neq -\varepsilon^{(m)} & \text{if } d = 0 \\ a^{(m)} = 0 \text{ or } o_m \nmid d & \text{if } d \geq 1 \end{cases} .$$

If $u_{(d,a,\varepsilon^{(m)})}$ is a unit, it has the inverse $u_{(d,-a,\varepsilon^{(m)})}$ and it is called an **elementary unit**.

(b) Any unit in $A[z;\sigma]$ is a finite product of elementary units. In particular, a unit $a \in A$ can be written as $a = u_{(0,a-1,\varepsilon^{(1)})} \cdot \ldots \cdot u_{(0,a-1,\varepsilon^{(r)})}$.

PROOF: (a): Obviously $u_{(d,a,\varepsilon^{(m)})} = u_{(d,a^{(m)},\varepsilon^{(m)})}$. If $d = 0$, then $u_{(d,a^{(m)},\varepsilon^{(m)})} = 1 + a^{(m)} =: \alpha \in A$ is a unit if and only if $\alpha^{(i)} \neq 0$ for $1 \leq i \leq r$ (cf. Lemma 4.8), which is the case if and only if $a^{(m)} \neq -\varepsilon^{(m)}$.
Now we consider the case $d \geq 1$. We can assume without loss of generality that $a^{(m)} \neq 0$, because otherwise $u_{(d,a,\varepsilon^{(m)})} = u_{(d,-a,\varepsilon^{(m)})} = 1$. It remains to show that

$$u_{(d,a,\varepsilon^{(m)})} \text{ is a unit in } A[z;\sigma] \iff o_m \nmid d \ .$$

We prove "$\Rightarrow$" indirectly: Write $u := u_{(d,a,\varepsilon^{(m)})}$, for short. If $o_m \,|\, d$, then $\varepsilon^{(m)}u = u^{(m)} = u\varepsilon^{(m)}$ and $\deg_z u^{(m)} = d \geq 1$. If u was a unit, then Theorem 4.6 implies that $\mathcal{C} := \mathfrak{v}(\,{}^{\bullet}\langle\, u^{(m)} \,\rangle)$ was a minimal σ-cyclic (n, k, dk)-convolutional code generated by $u^{(m)}$ with support m, where $k := \deg_x \pi_m \geq 1$. But this is in contradiction to the fact that $\mathcal{C}$ must have complexity 0 because of Lemma 4.7.
The "$\Leftarrow$"-part of the proof can be established as follows: We show that $u_{(d,a,\varepsilon^{(m)})}u_{(d,-a,\varepsilon^{(m)})} = u_{(d,-a,\varepsilon^{(m)})}u_{(d,a,\varepsilon^{(m)})} = 1$ if $o_m \not| d$. Note that $\sigma^d(\varepsilon^{(m)}) \neq \varepsilon^{(m)}$ if $o_m \not| d$, hence $\sigma^d(\varepsilon^{(m)})\varepsilon^{(m)} = 0$. With the aid of the last equation we calculate

$$\begin{aligned} u_{(d,a,\varepsilon^{(m)})}u_{(d,-a,\varepsilon^{(m)})} &= \left(1 + z^d a\varepsilon^{(m)}\right)\left(1 - z^d a\varepsilon^{(m)}\right) \\ &= 1 + z^d a\varepsilon^{(m)} - z^d a\varepsilon^{(m)} - z^d a\varepsilon^{(m)} z^d a\varepsilon^{(m)} \\ &= 1 - z^{2d}\sigma^d(a)a\sigma^d(\varepsilon^{(m)})\varepsilon^{(m)} = 1 \quad . \end{aligned}$$

The equation $u_{(d,-a,\varepsilon^{(m)})}u_{(d,a,\varepsilon^{(m)})} = 1$ can be obtained with the same arguments (replace a with $-a$).
(b): The proof of the equality $a = u_{(0,a-1,\varepsilon^{(1)})} \cdot \ldots \cdot u_{(0,a-1,\varepsilon^{(r)})}$ for a being a unit in A can be done straightforwardly. (It is even true for a being an arbitrary element of A.)
For the remaining part of the proof we use results from [GS02a]. Let $u \in A[z;\sigma]$ be a unit. First of all, note that ${}^{\bullet}\langle\, u \,\rangle = A[z;\sigma]$ and that a generator polynomial of the σ-cyclic convolutional code $A[z;\sigma]$ must be a unit $a \in A$. From [GS02a, Cor. 4.13 (a) and its proof] we know that there exist elementary units $u_1^{-1}, \ldots, u_t^{-1} \in A[z;\sigma]$ such that $a = u_t^{-1} \cdot \ldots \cdot u_1^{-1} \cdot u$. We already know that a can be written as a product of elementary units, hence $u = u_1 \cdot \ldots \cdot u_t \cdot a$ can be written as a product of elementary units. $\square$

Now we know due to the foregoing theorem how to construct (all) units in $A[z;\sigma]$. If $\sigma(\varepsilon^{(l)}) \neq \varepsilon^{(l)}$, we are even able to specify units whose l-th component has a prescribed degree. This is the subject of the following corollary:

Corollary 4.12
Consider $l \in \{1, \ldots, r\}$ and $\sigma \in Aut_{\mathbb{F}}(A)$ such that $\sigma(\varepsilon^{(l)}) \neq \varepsilon^{(l)}$. Then we have:

(a) For any $a \in A$ and $i \in \mathbb{N}_0$ the element $u_a(i) := 1 + za\sigma^i(\varepsilon^{(l)})$ is an elementary unit in $A[z;\sigma]$, its inverse is $u_{-a}(i)$.

(b) For any $d \in \mathbb{N}_0$ there exists a unit $u \in A[z;\sigma]$ such that $\deg_z u^{(l)} = d = \deg_z u$.

PROOF: Part (a) can be established as follows: We have $u_a(i) = u_{(1,a,\sigma^i(\varepsilon^{(l)}))}$. The case $\deg_z u_a(i) = 0$ is trivial. In the other case we have $\deg_z u_a(i) = 1$ and therefore $o_l \not| \deg_z u_a(i)$, because $o_l > 1$. If we put $\sigma^i(\varepsilon^{(l)}) = \varepsilon^{(l_i)}$, then Remark 4.10 implies that $o_l = o_{l_i} > 1$ and therefore $o_{l_i} \not| \deg_z u_a(i)$. Theorem 4.11 (a) implies the assertion.
Now we show part (b): Without loss of generality we may assume that $d > 0$. Define $u := u_{a_1}(1) \cdot \ldots \cdot u_{a_d}(d)$ where $a_1, \ldots, a_d$ are units in A. From part (a) we know that u is a unit in $A[z;\sigma]$, its inverse is $u_{-a_d}(d) \cdot \ldots \cdot u_{-a_1}(1)$. It has z-degree at most d. The z^d-part of u is given by

$$\begin{aligned} & \big(za_1\sigma(\varepsilon^{(l)})\big) \cdot \big(za_2\sigma^2(\varepsilon^{(l)})\big) \cdot \ldots \cdot \big(za_d\sigma^d(\varepsilon^{(l)})\big) \\ = \; & z^d \Big(\underbrace{\sigma^{d-1}(a_1)\sigma^{d-2}(a_2) \cdot \ldots \cdot \sigma(a_{d-1})a_d}_{=:\, a} \Big) \Big(\underbrace{\sigma^d(\varepsilon^{(l)}) \cdot \ldots \cdot \sigma^d(\varepsilon^{(l)})}_{= \sigma^d(\varepsilon^{(l)})} \Big) \quad . \end{aligned}$$

Now we have $u = 1 + f + z^d a\sigma^d(\varepsilon^{(l)})$, where $f = \sum_{i=1}^{d-1} z^i f_i$ is chosen appropriately. Hence

$$u^{(l)} = \varepsilon^{(l)}u = \varepsilon^{(l)}(1 + f + z^d a\sigma^d(\varepsilon^{(l)})) = \varepsilon^{(l)} + f^{(l)} + z^d a\sigma^d(\varepsilon^{(l)}) \quad .$$

Of course, a is a unit in A, since $a_1, \ldots, a_d$ are units in A. In particular we have $a\sigma^d(\varepsilon^{(l)}) \neq 0$ (cf. Lemma 4.8). We conclude that $\deg_z u^{(l)} = d = \deg_z u$. The proof is complete, but we like to mention that $\deg_z u^{(l')} < d$ for any $l' \in \{1, \ldots, r\} \setminus \{l\}$. □

As a direct consequence of Corollary 4.12, we have the following result concerning the existence of minimal σ-cyclic (n, k, dk)-convolutional codes for arbitrary $d \in \mathbb{N}_0$:

Corollary 4.13
Consider $l \in \{1, \ldots, r\}$ and $\sigma \in Aut_{\mathbb{F}}(A)$ such that $\sigma(\varepsilon^{(l)}) \neq \varepsilon^{(l)}$. Put $k := \deg_x \pi_l$. Then for any $d \in \mathbb{N}_0$ there exists a minimal σ-cyclic (n, k, dk)-convolutional code with support $\{l\}$.

PROOF: With Theorem 4.6 along with Remark 4.4 we have to show the existence of a unit u in $A[z;\sigma]$ satisfying $\deg_z u^{(l)} = d$. This was done in Corollary 4.12 (b). □

Corollary 4.12 (b) and its proof provide a construction method for minimal σ-cyclic (n, k, dk)-convolutional codes and we will use this method to generate an example:

Example 4.14

(a) We consider $\mathbb{F} = \mathbb{F}_4 = \{0, 1, \alpha, \alpha^2\}$ *and* $n = 3$ *and the representation of* A *as in Example 3.12, i.e.* $\varepsilon^{(1)} = 1 + x + x^2$, $\varepsilon^{(2)} = 1 + \alpha^2 x + \alpha x^2$ *and* $\varepsilon^{(3)} = 1 + \alpha x + \alpha^2 x^2$. *We choose the automorphism* σ *given by* $\sigma(x) = x^2$, *this implies*

$$\sigma(\varepsilon^{(1)}) = \varepsilon^{(1)} \quad , \quad \sigma(\varepsilon^{(2)}) = \varepsilon^{(3)} \quad , \quad \sigma(\varepsilon^{(3)}) = \varepsilon^{(2)} \quad .$$

Now we want to construct minimal σ*-cyclic* $(3, 1, 3)$*-convolutional codes with support* $\{2\}$. *Towards this end we put* $l = 2$ *and consider the units*

$$\begin{aligned} u &:= u_1(1) \cdot u_1(2) \cdot u_1(3) = (1 + z\varepsilon^{(3)})(1 + z\varepsilon^{(2)})(1 + z\varepsilon^{(3)}) \\ &= 1 + z\varepsilon^{(2)} + z^2(\varepsilon^{(2)} + \varepsilon^{(3)}) + z^3\varepsilon^{(3)} \quad , \\ \hat{u} &:= u_\alpha(1) \cdot u_1(2) \cdot u_1(3) = (1 + z\alpha\varepsilon^{(3)})(1 + z\varepsilon^{(2)})(1 + z\varepsilon^{(3)}) \\ &= 1 + z(\varepsilon^{(2)} + \alpha^2\varepsilon^{(3)}) + z^2(\alpha\varepsilon^{(2)} + \varepsilon^{(3)}) + z^3\alpha\varepsilon^{(3)} \quad . \end{aligned}$$

The second component of u *is* $u^{(2)} = \varepsilon^{(2)} + z^2\varepsilon^{(2)} + z^3\varepsilon^{(3)}$ *and thus* $\mathcal{C} := \mathfrak{v}({}^{\bullet}\langle\, u^{(2)} \,\rangle)$ *is a minimal* σ*-cyclic* $(3, 1, 3)$*-convolutional code. With Theorem 3.18, we have that*

$$G := [\mathfrak{v}(u^{(2)})] = [1 + z^2 + z^3, \alpha^2 + \alpha^2 z^2 + \alpha z^3, \alpha + \alpha z^2 + \alpha^2 z^3]$$

is a generator matrix of $\mathcal{C}$. *Obviously we have* $d_{free}(\mathcal{C}) \leq 9$. *Indeed we have* $d_{free}(\mathcal{C}) = 9$ *as one can verify for example with the aid of MAPLE and [GS02b]. Of course this distance is not MDS.*
The second component of $\hat{u}$ *is* $\hat{u}^{(2)} = \varepsilon^{(2)} + z\alpha^2\varepsilon^{(3)} + z^2\alpha\varepsilon^{(2)} + z^3\alpha\varepsilon^{(3)}$ *and thus* $\hat{\mathcal{C}} := \mathfrak{v}({}^{\bullet}\langle\, \hat{u}^{(2)} \,\rangle)$ *is a minimal* σ*-cyclic* $(3, 1, 3)$*-convolutional code. A generator matrix of this code is given by*

$$\hat{G} := [\mathfrak{v}(\hat{u}^{(2)})] = [1 + \alpha^2 z + \alpha z^2 + \alpha z^3, \alpha^2 + z + z^2 + \alpha^2 z^3, \alpha + \alpha z + \alpha^2 z^2 + z^3]$$

and the free distance of $\hat{\mathcal{C}}$ *is at most* 12. *MAPLE and [GS02b] yield that* $d_{free}(\mathcal{C}) = 12$, *which is MDS.*

(b) Again we deal with $\mathbb{F} = \mathbb{F}_4 = \{0, 1, \alpha, \alpha^2\}$, *but now we put* $n = 5$. *In this case* $x^n - 1 = (x + 1)(x^2 + \alpha x + 1)(x^2 + \alpha^2 x + 1)$ *and we display* A *as*

$$A = \rho^{-1}\big(\mathbb{F}[x]/\langle\, x + 1 \,\rangle \times \mathbb{F}[x]/\langle\, x^2 + \alpha x + 1 \,\rangle \times \mathbb{F}[x]/\langle\, x^2 + \alpha^2 x + 1 \,\rangle \big) \, .$$

The primitive idempotents are given by $\varepsilon^{(1)} = 1 + x + x^2 + x^3 + x^4$ *and*

$$\varepsilon^{(2)} = \alpha x + \alpha^2 x^2 + \alpha^2 x^3 + \alpha x^4 \quad , \quad \varepsilon^{(3)} = \alpha^2 x + \alpha x^2 + \alpha x^3 + \alpha^2 x^4$$

and the automorphism $\sigma \in Aut_{\mathbb{F}}(A)$ *defined by* $\sigma(x) = x^3$ *maps* $\varepsilon^{(2)}$ *onto* $\varepsilon^{(3)}$ *and vice versa. So we are able to construct* σ*-cyclic minimal* $(5, 2, 2 \cdot 3)$*-convolutional codes with support* $\{2\}$*. We put* $l = 2$ *and as in (a) we investigate the units* $u = u_1(1) \cdot u_1(2) \cdot u_1(3)$ *and* $\hat{u} = u_\alpha(1) \cdot u_1(2) \cdot u_1(3)$ *resp. their second components* $u^{(2)} = \varepsilon^{(2)} + z^2\varepsilon^{(2)} + z^3\varepsilon^{(3)}$ *and* $\hat{u}^{(2)} = \varepsilon^{(2)} + z\alpha^2\varepsilon^{(3)} + z^2\alpha\varepsilon^{(2)} + z^3\alpha\varepsilon^{(3)}$*. In this setting* $\mathcal{C} := \mathfrak{v}({}^{\bullet}\langle\, u^{(2)} \,\rangle)$ *and* $\hat{\mathcal{C}} := \mathfrak{v}({}^{\bullet}\langle\, \hat{u}^{(2)} \,\rangle)$ *are a minimal* σ*-cyclic* $(5, 2, 6)$*-convolutional codes. The MDS-bound for* $(5, 2, 6)$*-convolutional codes is* 19*, but with the generalized Griesmer bound given in [GS03, Theorem 3.4] one finds that any* $(5, 2, 6)$*-convolutional code over* $\mathbb{F}_4$ *with Forney indices* $(3, 3)$ *has free distance at most* 16*. A generator matrix of the code* $\mathcal{C}$ *is given by*

$$G := \begin{bmatrix} \mathfrak{v}(u^{(2)}) \\ \mathfrak{v}(xu^{(2)}) \end{bmatrix} = \begin{bmatrix} 0 & \alpha + \alpha z^2 + \alpha z^3 \\ \alpha + \alpha z^2 + \alpha^2 z^3 & \alpha^2 z^2 + \alpha^2 z^3 \\ \alpha^2 + \alpha^2 z^2 + \alpha z^3 & \alpha + \alpha^2 z^2 \\ \alpha^2 + \alpha^2 z^2 + \alpha z^3 & \alpha^2 + \alpha z^2 + \alpha^2 z^3 \\ \alpha + \alpha z^2 + \alpha^2 z^3 & \alpha^2 + \alpha z^3 \end{bmatrix}^{\mathrm{t}}$$

and both rows in G *have weight* 12*. With the aid of MAPLE and [GS02b] we calculated* $d_{free}(\mathcal{C}) = 10$*.*
The matrix

$$\hat{G} := \begin{bmatrix} \mathfrak{v}(\hat{u}^{(2)}) \\ \mathfrak{v}(x\hat{u}^{(2)}) \end{bmatrix} = \begin{bmatrix} 0 & \alpha + z + \alpha^2 z^2 + \alpha^2 z^3 \\ \alpha + \alpha z + \alpha^2 z^2 + z^3 & z + z^2 + z^3 \\ \alpha^2 + z + z^2 + \alpha^2 z^3 & \alpha + \alpha z + z^2 \\ \alpha^2 + z + z^2 + \alpha^2 z^3 & \alpha^2 + \alpha^2 z^2 + z^3 \\ \alpha + \alpha z + \alpha^2 z^2 + z^3 & \alpha^2 + \alpha z + \alpha^2 z^3 \end{bmatrix}^{\mathrm{t}}$$

is a generator matrix of the code $\hat{\mathcal{C}}$ *and both rows of* $\hat{G}$ *have weight* 16*. Again we used the computer to calculate* $d_{free}(\hat{\mathcal{C}}) = 14$*.*

4.3 Existence of Minimal Cyclic Convolutional Codes

In the foregoing sections we gave some information about the existence of minimal σ-cyclic convolutional codes with nonzero complexity: In Lemma 4.7 and the following comments we learned that minimal σ-cyclic convolutional codes with nonzero complexity exist if and only if (4.1) is satisfied. Corollary 4.13 gave even deeper insight: It states that (4.1) implies the existence of minimal σ-cyclic convolutional codes with support $\{l\}$ and complexity $\deg_x \pi_l \cdot d$ for any $d \in \mathbb{N}_0$. The following theorem combines these findings.

Theorem 4.15
Let $\sigma \in Aut_{\mathbb{F}}(A)$ and $l \in \{1, \ldots, r\}$ and define $k := \deg_x \pi_l$. Then the following statements are equivalent:
(i) $\sigma(\varepsilon^{(l)}) \neq \varepsilon^{(l)}$;
(ii) $\mathfrak{v}\left({}^{\bullet}\langle\, \varepsilon^{(l)} + z\sigma(\varepsilon^{(l)})\,\rangle\right)$ *is a (minimal, σ-cyclic) convolutional code;*
(iii) there exists a minimal σ-cyclic convolutional code with support $\{l\}$ and nonzero complexity;
(iv) for any $d \in \mathbb{N}_0$ there exists a minimal σ-cyclic (n, k, kd)-convolutional code with support $\{l\}$.

PROOF: "(i)⇒(ii)" was shown in [GS02a, Proof of Proposition 3.4]. But we can also develop a proof out of the present context: With Corollary 4.12 (a), we know that $\varepsilon^{(l)} + z\sigma(\varepsilon^{(l)})$ is the l-th component of the unit $u_1(1)$ and with Theorem 4.6 we conclude that $\mathfrak{v}\left({}^{\bullet}\langle\, \varepsilon^{(l)} + z\sigma(\varepsilon^{(l)})\,\rangle\right)$ is a (minimal) σ-cyclic convolutional code. "(ii)⇒(iii)" is obvious. "(iii)⇒(i)" and "(iv)⇒(i)" follow with Lemma 4.7. "(i)⇒(iv)" is a consequence of Corollary 4.13. □

As a consequence, the existence of an automorphism $\sigma \in Aut_{\mathbb{F}}(A)$ which gives rise to minimal σ-cyclic convolutional codes with nonzero complexity only depends on the the prime factor decomposition of $x^n - 1$, as the following corollary shows:

Corollary 4.16
For $k \in \{1, \ldots, n-1\}$ the following statements are equivalent:
(i) There exists $\sigma \in Aut_{\mathbb{F}}(A)$ such that there exist minimal σ-cyclic (n, k, dk)-convolutional codes for any $d \in \mathbb{N}_0$;
(ii) there exist at least two prime divisors of $x^n - 1$ over $\mathbb{F}$ of x-degree k.

The PROOF is a simple consequence of Theorem 3.8. □

With Corollary 3.22, we know that σ-cyclic convolutional codes with nonzero complexity do exist if and only if there are minimal σ-cyclic convolutional codes with nonzero complexity. For sake of clarity we repeat this result in the following remark:

Remark 4.17
A necessary and sufficient condition for the existence of a σ-cyclic convolutional code with nonzero complexity is the existence of a minimal σ-cyclic convolutional code with nonzero complexity. Hence there exist σ-cyclic convolutional codes with nonzero complexity if and only if the prime factor decomposition of $x^n - 1$ has at least two prime divisors of the same degree. This is a slightly modified version of [GS02a, Proposition 3.4].

Corollary 4.16 resp. Remark 4.17 gives rise to the question how the prime factor decomposition of $x^n - 1$ over $\mathbb{F}_q$ looks like. The theory of cyclic block codes provides a complete answer to this question, see for example [MS78, Chapter 7, §5]. A helpful tool are the so-called "cyclotomic cosets", which will be investigated now. Recall that we always assume $\gcd(n,p) = 1$ where p denotes the characteristic of $\mathbb{F} = \mathbb{F}_q$. This is equivalent to $\gcd(n,q) = 1$.

Definition and Theorem 4.18 *cf. [MS78, Chapter 7, §5]*
Let n, q be positive integers, q a prime power, where $\gcd(n,q) = 1$. For $j \in \mathbb{Z}$ we use the notation $\overline{j}$ for the remainder of j modulo n.

(a) For $i \in \{0, 1, \ldots, n-1\}$ we define

$$C_i := \{\overline{i}, \overline{iq}, \overline{iq^2}, \ldots, \overline{iq^{d_i-1}}\} \quad , \text{ where } \quad d_i := \min\{t \in \mathbb{N} \mid iq^t \equiv i \bmod n\} \quad . \tag{4.2}$$

The sets $C_0, \ldots, C_{n-1}$ are called the **cyclotomic cosets** $\bmod n$. *In particular we have $|C_i| = d_i$ for $0 \le i \le n-1$.*

(b) There exists $r \in \mathbb{N}$ and $i_1, \ldots, i_r \in \{0, 1, \ldots, n-1\}$ such that

$$\{0, 1, \ldots, n-1\} = \dot{\bigcup_{\nu=1}^{r}} C_{i_\nu} \qquad \textit{(disjoint union of sets)} \quad .$$

(c) With the data as in (b) we have: The prime factor decomposition of $x^n - 1$ over $\mathbb{F}_q$ consists of r irreducible polynomials and the degrees of these irreducible factors are (up to permutation) given by $d_{i_1}, \ldots, d_{i_r}$.

Example 4.19

(a) Consider $n = 5$, $q = 2$. Then the cyclotomic cosets $\bmod 5$ *are given by*

$$C_0 = \{0\} \ , \ C_1 = \{1, 2, 4, 3\} = C_2 = C_4 = C_3$$

and $x^5 - 1 \in \mathbb{F}_2[x]$ has one linear prime divisor (namely $x - 1$) and one of degree 4 (namely $x^4 + x^3 + x^2 + x + 1$). In particular, every σ-cyclic convolutional code has complexity 0 for any $\sigma \in Aut_{\mathbb{F}_2}(A)$.

(b) Now we consider $n = 8$, $q = 3$. The cyclotomic cosets $\bmod 8$ *are given by*

$$C_0 = \{0\},\ C_1 = \{1, 3\} = C_3,\ C_2 = \{2, 6\} = C_6,\ C_4 = \{4\},\ C_5 = \{5, 7\} = C_7$$

and $x^8 - 1 \in \mathbb{F}_3[x]$ has two linear prime divisors (these must be $x - 1$ and $x - 2$) and three of degree 2. In particular, in this case there exist automorphisms $\sigma \in Aut_{\mathbb{F}_3}(A)$ such that there exist minimal σ-cyclic $(8, 1, d)$- resp. $(8, 2, 2d)$-convolutional codes for any $d \in \mathbb{N}_0$.

Just to have a more complete picture of minimal σ-cyclic convolutional codes, we like to state that not all σ-cyclic (n,k,dk)-convolutional code are minimal:

Remark 4.20
There exist σ-cyclic (n,k,dk)-convolutional codes, which are not minimal For example, we consider a field $\mathbb{F}$ and $n \geq 3$ where $x^n - 1$ has at least three linear prime divisors (for instance $\mathbb{F} = \mathbb{F}_4$ and $n = 9$). Now $\varepsilon^{(1)}$, $\varepsilon^{(2)}$ and $\varepsilon^{(3)}$ correspond to linear prime divisors and we choose $\sigma \in Aut_{\mathbb{F}}(A)$ such that

$$\sigma(\varepsilon^{(1)}) = \varepsilon^{(2)} \quad , \quad \sigma(\varepsilon^{(2)}) = \varepsilon^{(3)} \quad , \quad \sigma(\varepsilon^{(3)}) = \varepsilon^{(1)} \quad .$$

(This is possible due to Theorem 3.8.)
The polynomial $g = (1+z\varepsilon^{(3)})(1+z\varepsilon^{(2)}) = 1+z(\varepsilon^{(2)}+\varepsilon^{(3)})$ is a unit in $A[z;\sigma]$ because it is a product of elementary units (cf. Corollary 4.12). The first and second component of g are given by $g^{(1)} = \varepsilon^{(1)} + z\varepsilon^{(2)}$ and $g^{(2)} = \varepsilon^{(2)} + z\varepsilon^{(3)}$. Hence $\mathcal{C}_i := \mathfrak{v}({}^{\bullet}\langle\, g^{(i)} \,\rangle)$ is a minimal σ-cyclic $(n,1,1)$-convolutional code with support $\{i\}$ for $i = 1,2$ (cf. Theorem 4.6).
The submodule $\mathcal{C} = \mathcal{C}_1 \oplus \mathcal{C}_2 = \mathfrak{v}({}^{\bullet}\langle\, g^{(1)}+g^{(2)} \,\rangle)$ is a σ-cyclic convolutional code because of Proposition 3.20: The z-free term of $\hat{g} := g^{(1)}+g^{(2)}$ is $\hat{g}_0 = \varepsilon^{(1)}+\varepsilon^{(2)}$ and we have $\hat{g} = \hat{g}_0 g$ and g is a unit. Note that $\hat{g}$ is left-reduced, therefore $\mathcal{C}$ is a σ-cyclic $(n,2,2)$-convolutional code, which is obviously not minimal.

We close this section with some remarks on the construction of minimal σ-cyclic convolutional codes via components of units (cf. Theorem 4.6). We intend to demonstrate that it is not clear how to determine all minimal σ-cyclic (n,k,dk)-convolutional codes resp. how to construct "good" minimal σ-cyclic (n,k,dk)-convolutional codes.

Remark 4.21

(a) *The proof of Corollary 4.12 provides a method to construct minimal σ-cyclic (n,k,dk)-convolutional codes. Here we used components of units of the form $u_{a_1}(1) \cdot \ldots \cdot u_{a_d}(d)$, where $a_1, \ldots, a_d$ are units in A. It is not clear that an arbitrary minimal σ-cyclic (n,k,dk)-convolutional code can be constructed in the same way, precisely: Let $\mathcal{C}$ be a minimal σ-cyclic (n,k,dk)-convolutional code with support $\{l\}$ with generator polynomial $g = g^{(l)}$. We know from Theorem 4.6 that we can find a unit $u \in A[z;\sigma]$ such that $g = u^{(l)}$. Is it possible to choose $u = u_{a_1}(1) \cdot \ldots \cdot u_{a_d}(d)$ for suitable units $a_1, \ldots, a_d \in A$?*

(b) Consider a fixed $d \in \mathbb{N}_0$ and $l \in \{1, \ldots, r\}$. It is not clear that all components $u^{(l)}$ having z-degree d of a unit u can be found just by investigating units having z-degree d, in other words: Do we have the equation

$$\begin{aligned} & \{u^{(l)} \mid \deg_z u^{(l)} = d,\ u \text{ is a unit in } A[z;\sigma]\} \\ = \ & \{u^{(l)} \mid \deg_z u^{(l)} = d = \deg_z u,\ u \text{ is a unit in } A[z;\sigma]\} \quad ? \end{aligned}$$

(c) What about the free distance of $\mathcal{C} := \mathfrak{v}({}^{\bullet}\langle\, u^{(l)}) \,\rangle$ where u is a unit in $A[z;\sigma]$? If u is an elementary unit, then $\operatorname{wt}(\mathfrak{v}(u^{(l)})) \leq 2n$ as one can readily show. Thus $d_{free}(\mathcal{C}) \leq 2n$ and this is in general less than the optimal bound $d_{free}(\mathcal{C}) \leq (n-k)(\lfloor \delta/k \rfloor + 1) + \delta + 1$ given in (1.1). So if we wish to find minimal σ-cyclic (n, k, dk)-convolutional codes with good free distance in this case, we must not investigate elementary units themselves but rather their products.

4.4 Automorphisms Generating the same Cyclic Convolutional Codes

This section is a digression which deals (loosely speaking) with the question how many different automorphisms in $Aut_{\mathbb{F}}(A)$ give rise to the same minimal cyclic codes with prescribed support.

In addition to the general assumptions which were made at the very beginning of Section 4 we will now consider a fixed $l \in \{1, \ldots, r\}$. Moreover, l is contained in one of the sets $R^{(1)}, \ldots, R^{(s)}$ (see (3.11)) and we will denote this set by $R^{(L)}$.

If $\mathcal{C} \subseteq \mathbb{F}[z]^n$ is a minimal σ-cyclic convolutional code with support $\{l\}$ and generator polynomial $g = g^{(l)}$, then only the action of σ on the fields $K^{(\mu)} = \varepsilon^{(\mu)} A$, $\mu \in R^{(L)}$, is of interest, precisely:

$$\mathfrak{p}(\mathcal{C}) = {}^{\bullet}\langle\, g^{(l)} \,\rangle = A[z;\sigma]\varepsilon^{(l)} g^{(l)} \subseteq \Big\{ \sum_{j \geq 0} z^j a_j \;\Big|\; a_j \in \sum_{\mu \in R^{(L)}} K^{(\mu)} \Big\} \quad .$$

This means that if two automorphisms $\sigma, \sigma' \in Aut_{\mathbb{F}}(A)$ satisfy $\sigma(h) = \sigma'(h)$ for all $h \in \sum_{\mu \in R^{(L)}} K^{(\mu)}$, a minimal σ-cyclic convolutional code with support $\{l\}$ is σ'-cyclic and vice versa. In this sense σ, σ' give rise to the same σ- resp. σ'-cyclic convolutional codes, although they may be "very" different in their action on $K^{(\mu)}$, $\mu \notin R^{(L)}$. Our aim is to give some information about the algebraic structure of automorphisms, which give rise to the same minimal σ-cyclic convolutional codes with support $\{l\}$.

We define

$$H^{(L)} := \{\sigma \in Aut_{\mathbb{F}}(A) \mid \sigma(h) = h \text{ for } h \in \textstyle\sum_{\mu \in R^{(L)}} K^{(\mu)}\} \quad .$$

Then $H^{(L)}$ is a normal subgroup of $Aut_{\mathbb{F}}(A)$ because we have $\tau^{-1} \circ \sigma \circ \tau \in H^{(L)}$ for any $\tau \in Aut_{\mathbb{F}}(A)$ and any $\sigma \in H^{(L)}$: For the proof it is sufficient to show that $h \in \sum_{\mu \in R^{(L)}} K^{(\mu)}$ implies $\tau(h) \in \sum_{\mu \in R^{(L)}} K^{(\mu)}$. But this is a consequence of Theorem 3.8 (which implies $\{K^{(\mu)} \mid \mu \in R^{(L)}\} = \{\tau(K^{(\mu)}) \mid \mu \in R^{(L)}\}$) along with the fact that τ is an $\mathbb{F}$-automorphism of A.

We conclude that

$$Aut_{\mathbb{F}}(A)/H^{(L)} = \{\sigma H^{(L)} \mid \sigma \in Aut_{\mathbb{F}}(A)\}$$

is a group again and $\sigma H^{(L)} = \{\sigma' \mid \sigma(h) = \sigma'(h) \text{ for } h \in \sum_{\mu \in R^{(L)}} K^{(\mu)}\}$ for $\sigma \in Aut_{\mathbb{F}}(A)$. Along with Theorem 3.8 we have

$$Aut_{\mathbb{F}}(A)\,/_{H^{(L)}} \cong G_L^{r_L} \circ S_{r_L} \quad ,$$

where the notation "G_L" and "$\circ$" stems from Theorem 3.8. In particular, $|Aut_{\mathbb{F}}(A)/H^{(L)}| = (\deg_x \pi_L)^{r_L} \cdot r_L!$ and there are at most $(\deg_x \pi_L)^{r_L} \cdot r_L!$ "really different" automorphisms which give rise to minimal cyclic convolutional codes with support $\{l\}$:

Theorem 4.22
Let $\mathcal{C} \subseteq \mathbb{F}[z]^n$ be a convolutional code and let $\sigma, \sigma' \in Aut_{\mathbb{F}}(A)$ satisfy $\sigma H^{(L)} = \sigma' H^{(L)}$. Then $\mathcal{C}$ is σ-cyclic and minimal with support $\{l\}$ if and only if $\mathcal{C}$ is σ'-cyclic and minimal with support $\{l\}$. □

If we have $\sigma, \sigma' \in Aut_{\mathbb{F}}(A)$ and if each minimal σ-cyclic convolutional code with support $\{l\}$ is σ'-cyclic, too, then it is not clear that this implies $\sigma H^{(L)} = \sigma' H^{(L)}$. (This would be the inverse implication of Theorem 4.22.) In the case $L = 1$ this statement is true, as we will see in Theorem 4.25. But in the case $L \geq 2$ it is not true in general: For $L \geq 2$ there exist $\sigma, \sigma' \in Aut_{\mathbb{F}}(A)$ satisfying $\sigma(\varepsilon^{(i)}) = \sigma'(\varepsilon^{(i)}) = \varepsilon^{(i)}$ for $1 \leq i \leq r$ and $\sigma H^{(L)} \neq \sigma' H^{(L)}$, cf. Theorem 3.8. Theorem 4.15 implies that any minimal σ-cyclic convolutional code with support $\{l\}$ has complexity 0, hence it is cyclic with respect to any automorphism in $Aut_{\mathbb{F}}(A)$. The latter implies that all minimal σ-cyclic convolutional codes with support $\{l\}$ are σ'-cyclic, too.

The following remark shows that we can not deduce $\sigma H^{(L)} = \sigma' H^{(L)}$ if we found (just) one minimal σ-cyclic convolutional code with support $\{l\}$ which is σ'-cyclic, too, even if it has nonzero complexity and even if $L = 1$:

Remark 4.23
In general it is not true that $\sigma, \sigma' \in Aut_{\mathbb{F}}(A)$ satisfy $\sigma H^{(L)} = \sigma' H^{(L)}$ if there exists a convolutional code which is both σ- and σ'-cyclic. A trivial counterexample are block codes, but we can also give a counterexample for minimal cyclic codes with nonzero complexity:
Consider some $n \in \mathbb{N}$ and a field $\mathbb{F}$ where there are three linear prime divisors of $x^n - 1$ corresponding to the primitive idempotent elements $\varepsilon^{(1)}$, $\varepsilon^{(2)}$ and $\varepsilon^{(3)}$. Then by Theorem 3.8 there exist automorphisms $\sigma, \sigma' \in Aut_{\mathbb{F}}(A)$ such that

$$\begin{aligned} \sigma(\varepsilon^{(1)}) = \varepsilon^{(2)} \quad &, \sigma(\varepsilon^{(2)}) = \varepsilon^{(3)} \quad , \sigma(\varepsilon^{(3)}) = \varepsilon^{(1)} \\ \sigma'(\varepsilon^{(1)}) = \varepsilon^{(3)} \quad &, \sigma'(\varepsilon^{(2)}) = \varepsilon^{(2)} \quad , \sigma'(\varepsilon^{(3)}) = \varepsilon^{(1)} \quad . \end{aligned}$$

Note that this implies $\sigma H^{(1)} \neq \sigma' H^{(1)}$. Now $u := 1 + z^5 \varepsilon^{(3)}$ is a unit both in $A[z;\sigma]$ and in $A[z;\sigma']$ (cf. Theorem 4.11) and its first component is $u^{(1)} = \varepsilon^{(1)} + z^5 \varepsilon^{(3)}$ in both $\mathbb{F}$-algebras. With Theorems 3.18 and 4.6 we conclude that $[\mathfrak{v}(\varepsilon^{(1)} + z^5 \varepsilon^{(3)})] \in \mathbb{F}[z]^{1 \times n}$ is a generator matrix of a convolutional code, which is minimal and cyclic with respect to σ and σ'.

We finish this section with an investigation of the case $L = 1$, i.e. $l \in \{1, \ldots, r_1\}$. We have $H^{(1)} = \{\sigma \in Aut_{\mathbb{F}}(A) \mid \sigma(\varepsilon^{(i)}) = \varepsilon^{(i)} \text{ for } 1 \leq i \leq r_1\}$ since $K^{(i)} \cong \mathbb{F}$ for $1 \leq i \leq r_1$. Furthermore, we have $Aut_{\mathbb{F}}(A)/H^{(1)} \cong S_{r_1}$. In this case, we can specify an isomorphism between the two groups $Aut_{\mathbb{F}}(A)/H^{(1)}$ and S_{r_1}:

Proposition 4.24
Let $\pi_i := x - b_i$ for $1 \leq i \leq r_1$. In particular we have $x = [\![b_1, \ldots, b_{r_1}, x, \ldots, x]\!]$. We know from Remark 3.11 that for any automorphism $\sigma \in Aut_{\mathbb{F}}(A)$ there exists a uniquely determined permutation $\beta_\sigma \in S_{r_1}$ such that $\sigma(x) = [\![b_{\beta_\sigma^{-1}(1)}, \ldots, b_{\beta_\sigma^{-1}(r_1)}, \ldots]\!]$. In particular we have $\sigma(\varepsilon^{(i)}) = \varepsilon^{(\beta_\sigma(i))}$ for $1 \leq i \leq r_1$ (cf. Corollary 3.10).
Along with Theorem 3.8 we obtain that the mapping

$$\beta_{(\,\cdot\,)} \;:\; Aut_{\mathbb{F}}(A) \to S_{r_1} \quad , \quad \sigma \mapsto \beta_\sigma$$

is a surjective group homomorphism and $\{\sigma \in Aut_{(A)} \mid \beta_\sigma = \mathrm{id}\} = H^{(1)}$. With the Homomorphism Theorem, we conclude that

$$Aut_{\mathbb{F}}(A)/H^{(1)} \to S_{r_1} \quad , \quad \sigma H^{(1)} \mapsto \beta_\sigma \tag{4.3}$$

is an isomorphism of the two groups. □

All minimal σ-cyclic convolutional codes with support $l \in \{1, \ldots, r_1\}$ are 1-dimensional and vice versa, as we will see in the next section in Observation 5.1. Therefore Theorem 4.22 can be reformulated and extended in this case:

Theorem 4.25
For $\sigma, \sigma' \in Aut_{\mathbb{F}}(A)$ we have $\sigma H^{(1)} = \sigma' H^{(1)}$ if and only if any σ-cyclic $(n,1)$-convolutional code is σ'-cyclic.

PROOF: The "only if"-part of the proof is Theorem 4.22. For the "if"-part we assume that $\sigma H^{(1)} \neq \sigma' H^{(1)}$ and construct a 1-dimensional code, which is cyclic with respect to σ, but not with respect to σ'. If $\sigma H^{(1)} \neq \sigma' H^{(1)}$, then there exists $l \in \{1, \ldots, r_1\}$ such that

$$\varepsilon^{(\hat{l})} = \sigma(\varepsilon^{(l)}) \neq \sigma'(\varepsilon^{(l)}) = \varepsilon^{(l')} \quad .$$

If $\varepsilon^{(\hat{l})} = \varepsilon^{(l)}$, then Theorem 4.15 yields on the one hand that any σ-cyclic $(n,1)$-convolutional code with support $\{l\}$ has complexity 0. On the other hand it tells us that there exist σ'-cyclic $(n,1)$-convolutional codes with support $\{l\}$ and with nonzero complexity. Hence the proof is complete for the case $\hat{l} = l$.
It remains to prove the case $\hat{l} \neq l$. We consider the element $u := \varepsilon^{(l)} + z\varepsilon^{(\hat{l})}$. In the algebra $A[z;\sigma]$ this is the l-th component of the elementary unit $1 + z\varepsilon^{(\hat{l})}$, hence $\mathcal{C} := \mathfrak{v}({}^{\bullet}\langle u \rangle)$ defines a σ-cyclic $(n,1)$-convolutional code.
If $\mathcal{C}$ was σ'-cyclic, too, then the left ideal $\mathcal{I} := A[z;\sigma'] \cdot u$ must be contained in $\mathfrak{p}(\mathcal{C})$. We show that this is impossible: Note that u is left reduced in $A[z;\sigma']$, since it has the two components $\varepsilon^{(l)}$ and $z\varepsilon^{(\hat{l})}$ and $l \neq \hat{l}$. Therefore $\mathfrak{v}(\mathcal{I})$ is a σ'-cyclic submodule of $\mathbb{F}[z]^n$ with "generator polynomial" u. Theorem 3.18 tells us that $\mathfrak{v}(\mathcal{I})$ has dimension 2. This means $\mathfrak{v}(\mathcal{I}) \not\subseteq \mathcal{C}$ and therefore $\mathcal{I} \not\subseteq \mathfrak{p}(\mathcal{C})$. □

5 Cyclic Convolutional Codes of Dimension 1

Recall that we always assume that (3.1) holds. Furthermore, we consider a fixed automorphism $\sigma \in Aut_{\mathbb{F}}(A)$ and a fixed representation of A as a cartesian product of fields $K_1 \times \ldots \times K_r$. Again we make use of the framework as in (3.9) – (3.15) and the multiplication in the $\mathbb{F}$-algebra $A[z;\sigma]$ will be denoted as in (3.16).

This section deals with 1-dimensional σ-cyclic convolutional codes, because these are examples for minimal σ-cyclic convolutional codes, as the following observation tells us:

Observation 5.1
Let $\{0\} \neq \mathcal{C} \subseteq \mathbb{F}[z]^n$ be a σ-cyclic (n,k)-submodule and let $\mathfrak{p}(\mathcal{C}) = {}^{\bullet}\langle\, g \,\rangle$, where g is left-reduced. Recall Theorem 3.18 (a), which states that $g, x\mathfrak{v}(g), \ldots, x^{k-1}\mathfrak{v}(g)$ is a left $\mathbb{F}[z]$-basis of $\mathcal{C}$ and that $k = \deg_x \pi_{(g)}$, where $\pi_{(g)} := \prod_{k \in T_g} \pi_k$. This implies

$$\mathcal{C} \text{ is 1-dimensional} \iff g = g^{(l)} \text{ for some } l \in \{1, \ldots, r_1\} = R^{(1)} \quad .$$

In particular we have that $\mathcal{C}$ is a 1-dimensional σ-cyclic convolutional code if and only if $\mathcal{C}$ is a minimal σ-cyclic convolutional code with support $\{l\}$ for some $l \in \{1, \ldots, r_1\}$.

We will deepen the results about minimal σ-cyclic convolutional codes from the previous section for 1-dimensional cyclic convolutional codes in Section 5.1. In Section 5.2 we determine the shape of generator matrices of 1-dimensional cyclic convolutional codes. This result is used to suggest some methods for constructing 1-dimensional cyclic convolutional codes resp. the investigation of special 1-dimensional cyclic convolutional codes in Section 5.3.

5.1 Existence of 1-dimensional Cyclic Convolutional Codes

With Observation 5.1, we can reformulate Theorem 4.15 and Remark 4.17 for 1-dimensional cyclic convolutional codes:

Theorem 5.2
Let $\sigma \in Aut_{\mathbb{F}}(A)$ and $l \in \{1, \ldots, r_1\}$. Then the following statements are equivalent:
(i) $\sigma(\varepsilon^{(l)}) \neq \varepsilon^{(l)}$;
(ii) $[\mathfrak{v}\left(\varepsilon^{(l)} + z\sigma(\varepsilon^{(l)})\right)]$ *is basic;*
(iii) there exists a σ-cyclic $(n, 1, \delta)$-convolutional code for some $\delta \in \mathbb{N}$;
(iv) there exists a σ-cyclic $(n, 1, \delta)$-convolutional code for any $\delta \in \mathbb{N}_0$. □

Corollary 5.3
The following statements are equivalent:
(i) There exists $\sigma \in Aut_{\mathbb{F}}(A)$ such that there exists a σ-cyclic $(n, 1, \delta)$-convolutional code for some $\delta \in \mathbb{N}$;
(ii) there exist at least two linear prime divisors of $x^n - 1$ over $\mathbb{F}$. □

The problem, for which n, q the polynomial $x^n - 1$ has at least two linear prime divisors over $\mathbb{F}_q$, can be solved completely and we will develop an answer now. Recall Theorem 4.18 and the notation C_i and d_i from (4.2). We can re-formulate the question for which n, q the polynomial $x^n - 1$ has at least two linear prime divisors over $\mathbb{F}_q$ by asking: For which n, q does there exist a cyclotomic coset C_i where $d_i = 1$ and $i \in \{1, \ldots, n-1\}$? But $d_i = 1$ if and only if $iq \equiv i \bmod n$, which is equivalent to $n \,|\, i(q-1)$.

Lemma 5.4
For $n, q \in \mathbb{N}$, $n, q \geq 2$, the following statements are equivalent:
(i) There exists $i \in \{1, \ldots, n-1\}$ such that $n \,|\, i(q-1)$;
(ii) $\gcd(n, q-1) \neq 1$.

Proof. (i)⇒(ii): If $\gcd(n, q-1) = 1$, then $n \,|\, i(q-1)$ is possible only if $n \,|\, i$. This is impossible because of $i < n$.
(ii)⇒(i): Define $f := \gcd(n, q-1) > 1$ and $i := \frac{n}{f}$. Then $i \in \{1, \ldots, n-1\}$ and $i(q-1) = n\frac{q-1}{f}$, hence $n \,|\, i(q-1)$. □

Now the following theorem is obvious:

Theorem 5.5
Let $n \in \mathbb{N}$, let q be a prime power and consider $\mathbb{F} := \mathbb{F}_q$. (As always we assume $\gcd(n, q) = 1$*.) The following statements are equivalent:*
(i) The polynomial $x^n - 1$ has at least two linear prime divisors over $\mathbb{F}_q$;

(ii) $\gcd(n, q-1) \neq 1$;

(iii) *there exists* $\sigma \in Aut_{\mathbb{F}}(A)$, *such that there exists a* σ*-cyclic* $(n, 1, \delta)$*-convolutional code for any* $\delta \in \mathbb{N}_0$. □

We illustrate this by investigating some special fields in the following corollary.

Corollary 5.6

(a) *There exists no* σ*-cyclic* $(n, 1, \delta)$*-convolutional code over* $\mathbb{F}_2$ *for any* n *(satisfying (3.1)) and* $\sigma \in Aut_{\mathbb{F}_2}(A)$ *if we have* $\delta \geq 1$. *In other words: All* 1*-dimensional* σ*-cyclic convolutional codes over* $\mathbb{F}_2$ *have complexity 0 and are block codes in this sense.*
Observe that $\mathbb{F}_2$ *is an important field for applications and that the non-existence of* 1*-dimensional* σ*-cyclic convolutional codes over* $\mathbb{F}_2$ *with nonzero complexity is a pity from this point of view.*

(b) *Over* $\mathbb{F}_3$ *there exist* σ*-cyclic* $(n, 1, \delta)$*-convolutional codes for* $\delta \geq 1$ *if and only if* $2 \,|\, n$.

(c) *Over* $\mathbb{F}_4$ *there exist* σ*-cyclic* $(n, 1, \delta)$*-convolutional codes for* $\delta \geq 1$ *if and only if* $3 \,|\, n$.

(d) *If* $n \in \mathbb{N}$ *is even and* $\mathbb{F}$ *is a finite field where* $char(\mathbb{F}) \neq 2$, *then* n *and* $q-1$ *are both multiples of* 2. *Hence there exists* $\sigma \in Aut_{\mathbb{F}}(A)$ *such that there exists a* σ*-cyclic* $(n, 1, \delta)$*-convolutional code for any* $\delta \in \mathbb{N}_0$.

5.2 Generator Matrices of 1-dimensional Cyclic Convolutional Codes

We intend to develop how generator matrices of 1-dimensional σ-cyclic convolutional codes look like. For this purpose, we present a proposition which provides a first link between a generator polynomial and a generator matrix of a 1-dimensional σ-cyclic convolutional code.
Throughout this section let σ denote a fixed automorphism in $Aut_{\mathbb{F}}(A)$.

Proposition 5.7

(a) *A generator matrix of an* $(n, 1)$*-submodule of* $\mathbb{F}[z]^n$ *is unique up to multiplication with a constant factor in* $\mathbb{F} \setminus \{0\}$. *In particular, a generator polynomial* g *of a* σ*-cyclic* $(n, 1)$*-convolutional code* $\mathcal{C}$ *is unique up to multiplication with a constant factor in* $\mathbb{F} \setminus \{0\}$ *and* $\mathfrak{v}(g)$ *is a minimal basis of* $\mathcal{C}$.

(b) Let v be an $\mathbb{F}[z]$-basis of the $(n,1)$-submodule $\mathcal{C}$ and let $g := \mathfrak{p}(v) \in A[z;\sigma]$ satisfy $g = g^{(l)}$ for some $l \in \{1,\dots,r_1\}$. Then $\mathcal{C}$ is σ-cyclic and we have $\mathfrak{p}(\mathcal{C}) = {}^{\bullet}\langle\, g \,\rangle$.

(c) If $\mathcal{C}$ is an $(n,1)$-convolutional code with $\mathbb{F}[z]$-basis v, then

$$\mathcal{C} \text{ is } \sigma\text{-cyclic} \iff \mathfrak{p}(v) = \mathfrak{p}(v)^{(l)} \text{ for some } l \in \{1,\dots,r_1\} \quad .$$

PROOF: The first statement of part (a) is obvious. The second one follows from the first along with Theorem 3.18 (a).

Part (b) can be proved as follows. If $g = g^{(l)}$ for some $l \in \{1,\dots,r_1\}$, then g is left-reduced and it is an $\mathbb{F}[z]$-basis of ${}^{\bullet}\langle\, g \,\rangle$ by Theorem 3.18 (a). This means v is an $\mathbb{F}[z]$-basis both of $\mathcal{C}$ and of the σ-cyclic submodule $\hat{\mathcal{C}} := \mathfrak{v}({}^{\bullet}\langle\, g \,\rangle)$, hence $\mathcal{C} = \hat{\mathcal{C}}$. In particular $\mathcal{C}$ is σ-cyclic.

The "$\Leftarrow$"-part of (c) is a special case of (b). For the "$\Rightarrow$"-part we write $\mathfrak{p}(\mathcal{C}) = {}^{\bullet}\langle\, \hat{g} \,\rangle$, where $\hat{g} = \hat{g}^{(l)} \in A[z;\sigma]$ for some $l \in \{1,\dots,r_1\}$ (cf. Observation 5.1). Notice that $\hat{g}$ is left-reduced. Again Theorem 3.18 (a) implies that $\mathfrak{v}(\hat{g})$ is an $\mathbb{F}[z]$-basis of $\mathcal{C}$ and along with part (a) we have $v = \alpha\mathfrak{v}(\hat{g})$ for some $\alpha \in \mathbb{F} \setminus \{0\}$. Hence $\mathfrak{p}(v) = \alpha\hat{g}$ and $\mathfrak{p}(v) = \mathfrak{p}(v)^{(l)}$. □

Parts (b) and (c) of Proposition 5.7 tell us that a 1-dimensional submodule $\mathcal{C} \subseteq \mathbb{F}[z]^n$ with generator matrix $G = [v] \in \mathbb{F}[z]^{1\times n}$ is a σ-cyclic convolutional code if and only if G is basic and $\mathfrak{p}(v) = \mathfrak{p}(v)^{(l)}$ for some $l \in \{1,\dots,r_1\}$. This means that the set of all generator matrices of σ-cyclic $(n,1)$-convolutional codes with support $\{l\}$, $l \in \{1,\dots,r_1\}$, is given by

$$\mathcal{G}_l := \left\{\ [\mathfrak{v}(g^{(l)})] \;\middle|\; g \in A[z;\sigma]\ ,\ [\mathfrak{v}(g^{(l)})] \text{ is basic } \right\} \quad . \tag{5.1}$$

Obviously we can learn something about generator matrices of 1-dimensional σ-cyclic convolutional if we investigate the components of a polynomial in $A[z;\sigma]$. If $g = \sum_{i=0}^{\delta} z^i g_i \in A[z;\sigma]$ is such a polynomial and $l \in \{1,\dots,r_1\}$, then

$$g^{(l)} = \varepsilon^{(l)} g = \sum_{i=0}^{\delta} z^i \sigma^i(\varepsilon^{(l)}) g_i \quad .$$

Notice that $\sigma^i(\varepsilon^{(l)}) = \varepsilon^{(l')}$ for some $l' \in \{1,\dots,r_1\}$, hence we have $\sigma^i(\varepsilon^{(l)})g_i \in \varepsilon^{(l')}A \cong \mathbb{F}$. Therefore we can pretend that g_i are elements in $\mathbb{F}$, precisely: For $l \in \{1,\dots,r_1\}$ we have

$$\left\{\ g^{(l)} \;\middle|\; g \in A[z;\sigma]\ \right\} = \left\{\ \sum_{i=0}^{\delta} a_i z^i \sigma^i(\varepsilon^{(l)}) \;\middle|\; \delta \in \mathbb{N}_0\ ,\ a_0,\dots,a_\delta \in \mathbb{F}\ \right\} \quad .$$

Consequently the set $\mathcal{G}_l$ defined in (5.1) is the set of all basic matrices of the form

$$[\sum_{i=0}^{\delta} a_i z^i \mathfrak{v}(\sigma^i(\varepsilon^{(l)}))] \quad , \quad \text{where} \quad \delta \in \mathbb{N}_0 \ , \ a_0, \ldots, a_\delta \in \mathbb{F} \quad . \tag{5.2}$$

Fortunately, we can determine exactly how $\mathfrak{v}(\varepsilon^{(1)}), \ldots, \mathfrak{v}(\varepsilon^{(r_1)})$ look like (and therefore we know how $\mathfrak{v}(\sigma^i(\varepsilon^{(l)}))$ looks like). Details are given now:

Notation 5.8
(a) We write $\pi_l = x - b_l$ for $1 \leq l \leq r_1$.
(b) For a fixed $l \in \{1, \ldots, r_1\}$ we use the notation

$$\varepsilon^{(l_i)} := \sigma^i(\varepsilon^{(l)}) \quad \textit{and} \quad b_{\sigma^i} := b_{l_i} \quad \textit{for any } i \in \mathbb{N}_0 \quad .$$

Note that $b_{\sigma^i} = b_{\sigma^{i \bmod o_l}}$ for any $i \in \mathbb{N}_0$. (For the definition of o_l recall Definition 4.9.)

Lemma 5.9
For $1 \leq l \leq r_1$ and b_l as in Notation 5.8 (a) we have $\varepsilon^{(l)} = n^{-1} \sum_{i=0}^{n-1} b_l^{n-i} x^i$, in particular

$$\mathfrak{v}(\varepsilon^{(l)}) = n^{-1}[1, b_l^{n-1}, b_l^{n-2}, \ldots, b_l^2, b_l] \quad .$$

PROOF: Define $p_l := \frac{x^n - 1}{x - b_l} = \prod_{i=1, i \neq l}^{r} \pi_i$. It is easy to see that p_l equals $\varepsilon^{(l)}$ up to a constant in $\mathbb{F} \setminus \{0\}$, precisely

$$\varepsilon^{(l)} = \big(p_l(b_l)\big)^{-1} p_l \quad .$$

Note that in this setting $b_l^n = 1$ and therefore $b_l^{-n} = 1$. Dividing $x^n - 1$ by $\pi_l = x - b_l$ we calculate $p_l = \sum_{i=0}^{n-1} b_l^{n-(i+1)} x^i$. This yields $p_l(b_l) = n b_l^{n-1} = n b_l^{-1}$ and $\varepsilon^{(l)} = n^{-1} b_l p_l = n^{-1} \sum_{i=0}^{n-1} b_l^{n-i} x^i$. □

Again, a matrix $G \in \mathbb{F}[z]^{1 \times n}$ is a generator matrix of a σ-cyclic $(n, 1)$-convolutional code $\mathcal{C}$ with support $\{l\} \subseteq \{1, \ldots, r_1\}$ if and only if $G \in \mathcal{G}_l$ (see (5.1) and (5.2)). Now we know exactly how the elements of $\mathcal{G}_l$ look like (cf. Lemma 5.9), indeed $G \in \mathcal{G}_l$ if and only if G is basic and

$$\begin{aligned} G &= [\sum_{i=0}^{\delta} a_i z^i \mathfrak{v}(\sigma^i(\varepsilon^{(l)}))] = [\sum_{i=0}^{\delta} a_i z^i \mathfrak{v}(\varepsilon^{(l_i)})] \\ &= \sum_{i=0}^{\delta} a_i z^i \cdot n^{-1} \cdot [1, b_{\sigma^i}^{n-1}, b_{\sigma^i}^{n-2}, \ldots, b_{\sigma^i}^2, b_{\sigma^i}] \\ &= n^{-1} \cdot \sum_{i=0}^{\delta} a_i z^i [1, b_{\sigma^i}^{n-1}, b_{\sigma^i}^{n-2}, \ldots, b_{\sigma^i}^2, b_{\sigma^i}] \quad \text{for some } a_0, \ldots, a_\delta \in \mathbb{F} \quad , \end{aligned}$$

where we made use of Notation 5.8 (b). Rescaling the matrix G with n we can assume without loss of generality that

$$G = \sum_{i=0}^{\delta} a_i z^i [1, b_{\sigma^i}^{n-1}, b_{\sigma^i}^{n-2}, \ldots, b_{\sigma^i}^2, b_{\sigma^i}] \text{ for some } a_0, \ldots, a_\delta \in \mathbb{F} \ .$$

We know from Proposition 5.7 (a) that G is a minimal basis. Hence, the code $\operatorname{im}_{\mathbb{F}[z]} G$ has complexity exactly δ if and only if $a_\delta \neq 0$.

We summarize these statements in the following corollary.

Corollary 5.10
Let $\mathcal{C}$ be an $(n, 1, \delta)$-submodule. Then the following statements are equivalent:
(i) $\mathcal{C}$ is a σ-cyclic $(n, 1, \delta)$-convolutional code;
(ii) there exists some $l \in \{1, \ldots, r_1\}$ and $a_0, \ldots, a_\delta \in \mathbb{F}$ satisfying $a_0 \neq 0 \neq a_\delta$ such that

$$\sum_{i=0}^{\delta} a_i z^i [1, b_{\sigma^i}^{n-1}, b_{\sigma^i}^{n-2}, \ldots, b_{\sigma^i}^2, b_{\sigma^i}] \text{ is a basic generator matrix of } \mathcal{C} \ ,$$

where we made use of Notation 5.8 (b).
In particular, a σ-cyclic $(n, 1, \delta)$-convolutional code $\mathcal{C}$ having complexity $\delta > 0$ satisfies $2n \leq d_{free}(\mathcal{C}) \leq |\{i \mid a_i \neq 0\}| \cdot n \leq (\delta + 1) \cdot n$. If $\mathcal{C}$ has complexity 0 then $d_{free}(\mathcal{C}) = n$.

PROOF: Most parts of the proof are clear from the foregoing investigations. It remains to show that for some σ-cyclic $(n, 1, \delta)$-convolutional code $\mathcal{C}$ the formula $2n \leq d_{free}(\mathcal{C}) \leq |\{i \mid a_i \neq 0\}| \cdot n$ is true in the case $\delta > 0$ and that $d_{free}(\mathcal{C}) = n$ if $\delta = 0$.
Let $G := \sum_{i=0}^{\delta} a_i z^i [1, b_{\sigma^i}^{n-1}, b_{\sigma^i}^{n-2}, \ldots, b_{\sigma^i}^2, b_{\sigma^i}]$ be a generator matrix of $\mathcal{C}$. If $\delta > 0$, then every entry in G has a nonzero constant term and a nonzero coefficient of z^δ. A nonzero codeword in $\mathcal{C}$ can be written as fG for some $f \in \mathbb{F}[z] \setminus \{0\}$ and we have $\deg_z fG > 0$. Again, every entry in fG has a nonzero constant term and the z-leading coefficient is nonzero, too. Hence every entry of fG has weight at least 2 and we conclude $2n \leq d_{free}(\mathcal{C})$. The upper bound for $d_{free}(\mathcal{C})$ can be obtained with $\operatorname{wt}(G) = |\{i \mid a_i \neq 0\}| \cdot n$. The case $\delta = 0$ follows with the same arguments, but now the constant term and the z^δ-term of every entry in G are the same. □

The two bounds for $d_{free}(\mathcal{C})$ which were given in Corollary 5.10 are tight in general. We give examples in the next remark.

Remark 5.11
Observe that the upper bound $d_{free}(\mathcal{C}) \leq (\delta+1)\cdot n$ *is the MDS-bound for* $(n,1,\delta)$*-codes (cf. (1.1)). Consequently all* σ*-cyclic* $(n,1,0)$*-convolutional codes are MDS block codes.*
Both the lower and the upper bound $2n \leq d_{free}(\mathcal{C}) \leq |\{i \mid a_i \neq 0\}|\cdot n$ *for* σ*-cyclic* $(n,1,\delta)$*-convolutional codes having complexity* $\delta > 0$ *are tight, the simplest example is the case* $\delta = 1$*. Then* $2n \leq d_{free}(\mathcal{C}) \leq 2n$ *and we conclude that every* σ*-cyclic* $(n,1,\delta)$*-convolutional code having complexity* $\delta = 0$ *or* $\delta = 1$ *is an MDS-convolutional code.*
There are also examples for both bounds being tight for $\delta \geq 2$*: For the upper bound recall Example 3.4, which introduced a* σ*-cyclic* $(3,1,2)$*-convolutional code over* $\mathbb{F}_4$ *with optimal free distance* 9*. For the lower bound we consider some* $\sigma \in Aut_{\mathbb{F}}(A)$ *such that* $\sigma(\varepsilon^{(1)}) = \varepsilon^{(2)}$ *and* $\sigma(\varepsilon^{(2)}) = \varepsilon^{(1)}$*. We define an* $(n,1,3)$*-convolutional code* $\mathcal{C}$ *via the generator polynomial* $g = g^{(1)} = \varepsilon^{(1)} + z^3\varepsilon^{(2)}$*, which is the first component of the elementary unit* $1+z^3\varepsilon^{(2)}$ *(see Theorems 4.6 and 4.11). Now* $[\mathfrak{v}(g)] = [1+z^3, b_{\sigma^0}^{n-1} + z^3 b_{\sigma^3}^{n-1}, \ldots, b_{\sigma^0} + z^3 b_{\sigma^3}]$ *is a generator matrix of* $\mathcal{C}$ *which has weight* $2n$*, hence* $d_{free}(\mathcal{C}) = 2n$*.*

We close this section by a corollary which tells us how one can check whether a given basic matrix $G \in \mathbb{F}[z]^{1\times n}$ is a generator matrix of a σ-cyclic $(n,1)$-convolutional code.

Corollary 5.12
We know from Remark 4.5 that we can determine the support of a minimal σ*-cyclic convolutional code* $\mathcal{C} \subseteq \mathbb{F}[z]^n$ *with the knowledge of one row of a minimal basis of* $\mathcal{C}$*. In the* 1*-dimensional case every generator matrix* $G = [v] \in \mathbb{F}[z]^{1\times n}$ *is a minimal basis, hence* $T_{\mathfrak{p}(v)}$ *is the support of* $\mathcal{C}$*. But we know even more in the* 1*-dimensional case, because the shape of* G *must be as described in Corollary 5.10 (ii).*
If G *is a generator matrix of an* $(n,1,\delta)$*-convolutional code we can write*

$$G = \sum_{i=0}^{\delta} a_i [1, g_{i,1}, \ldots, g_{i,n-1}] z^i$$

with suitable $a_i, g_{i,j} \in \mathbb{F}$*,* $0 \leq i \leq \delta$*,* $1 \leq j \leq n-1$*. If we want to check whether* $\mathcal{C}$ *is cyclic, we can first investigate* $g_{1,1}$*. If* $g_{1,1} = b_l^{-1}$ *for some* $l \in \{1, \ldots, r_1\}$*, then* $\mathcal{C}$ *might be cyclic with support* $\{l\}$*.*
Using Notation 5.8 (b) and choosing some $\sigma \in Aut_{\mathbb{F}}(A)$*, we have*

$$\mathcal{C} \text{ is } \sigma\text{-cyclic} \iff \forall\ 0 \leq i \leq \delta \text{ where } a_i \neq 0 \ \ \forall\ 1 \leq j \leq n-1 : \ g_{i,j} = b_{\sigma^i}^{n-j} \ .$$

By the way: We do not have to specify a certain $\sigma \in Aut_{\mathbb{F}}(A)$*; we can start with a permutation* $\beta \in S_{r_1}$ *and use the Proposition 4.24 resp. the mapping in (4.3). The code* $\mathcal{C}$ *is cyclic with respect to any automorphism* σ *satisfying* $\beta_\sigma = \beta$ *if and only if*

$$\forall\ 0 \leq i \leq \delta \text{ where } a_i \neq 0\ \ \forall\ 1 \leq j \leq n-1 :\ \ g_{i,j} = b^{n-j}_{\beta^i(l)}\ \ .$$

5.3 Some Attempts of Constructing 1-dimensional Cyclic Convolutional Codes

Of course we can try to construct 1-dimensional σ-cyclic convolutional codes (or minimal σ-cyclic convolutional codes in general) via components of units (cf. Theorem 4.6). In the foregoing section we learned how generator matrices of 1-dimensional σ-cyclic convolutional codes look like. This provides another method to construct these codes: We have to find or define matrices which have the "right shape" and which are basic.

In Section 5.3.1 we present a method to construct 1-dimensional σ-cyclic convolutional codes which uses irreducible polynomials over $\mathbb{F}[z]$. In Section 5.3.2 we investigate a special choice of a generator matrix of the "right shape" and try to find out under which condition this matrix is basic. Some results will be used in Section 5.3.3, where we introduce a method to construct 1-dimensional σ-cyclic convolutional codes with a good free distance.

Throughout this section we consider a fixed $l \in \{1, \ldots, r_1\}$ and a fixed $\sigma \in Aut_{\mathbb{F}}(A)$ such that $\sigma(\varepsilon^{(l)}) \neq \varepsilon^{(l)}$. This means $o_l \geq 2$ (for the definition of o_l recall Definition 4.9) and makes sure that we can find σ-cyclic $(n, 1, \delta)$-convolutional codes for any $\delta \in \mathbb{N}_0$ (cf. Theorem 5.2). We will also make use of Notation 5.8.

5.3.1 Construction via Irreducible Polynomials

We are able to describe a generator matrix of a σ-cyclic $(n, 1, \delta)$-convolutional code with the aid of an irreducible polynomial in $\mathbb{F}[z]$ of degree $\delta > 0$:

Lemma 5.13
Choose an irreducible polynomial $f = \sum_{i=0}^{\delta} a_i z^i \in \mathbb{F}[z]$ *with degree* $\delta > 0$*. (This is always possible, cf. [LN97, Corollary 2.11].) Then*

$$G := \sum_{i=0}^{\delta} a_i z^i [1, b^{n-1}_{\sigma^i}, b^{n-2}_{\sigma^i}, \ldots, b_{\sigma^i}]$$

is a (basic) generator matrix of a σ-cyclic $(n, 1, \delta)$-convolutional code with $2n \leq d_{free}(\mathcal{C}) \leq n \cdot |\{i \mid a_i \neq 0\}|$.

PROOF: Let f_1 resp. f_2 denote the first resp. the second entry of G. We show that f_1 and f_2 are coprime. This implies that G is basic. We have

$$\begin{aligned} b_{\sigma^0}^{n-1} \cdot f_1 &= a_0 b_{\sigma^0}^{n-1} + a_1 b_{\sigma^0}^{n-1} z + \ldots + a_\delta b_{\sigma^0}^{n-1} z^\delta \quad , \\ f_2 &= a_0 b_{\sigma^0}^{n-1} + a_1 b_{\sigma^1}^{n-1} z + \ldots + a_\delta b_{\sigma^\delta}^{n-1} z^\delta \quad . \end{aligned}$$

Now $b_{\sigma^0}^{n-1} \cdot f_1$ and f_2 have the same z-free term and a different z-coefficient. Since f_1 was chosen irreducible, f_2 must be coprime to f_1. For the lower and the upper bound of $d_{free}(\mathcal{C})$ recall Corollary 5.10. □

It is perhaps an advantage of this construction that we do not have to deal with concrete units. But on the other hand it is not clear if the constructed codes have good properties. For example, if we want to have a chance to obtain an MDS convolutional code via the construction method in Lemma 5.13, we have to find irreducible polynomials $f = \sum_{i=0}^{\delta} a_i z^i \in \mathbb{F}[z]$ with $a_i \neq 0$, $0 \leq i \leq \delta$. Even if we succeeded to find such an f, it can happen that the correspondinng code is not MDS (see Example 5.14).
One may also argue that it is not so easy in general to find irreducible polynomials of prescribed degree over $\mathbb{F}$. Fortunately, MAPLE is able to manage this task and the following example uses irreducible polynomials generated by MAPLE via the command `Randprime`.

Example 5.14

(a) We consider $\mathbb{F} = \mathbb{F}_7$ and $n = 4$. The polynomial $x^n - 1$ has the prime factor decomposition $x^n - 1 = (x+1)(x+6)(x^2+1)$. We display A as

$$A = \rho^{-1}\big(\mathbb{F}[x]/\langle x+1 \rangle \times \mathbb{F}[x]/\langle x+6 \rangle \times \mathbb{F}[x]/\langle x^2+1 \rangle \big) \quad .$$

We choose an automorphism $\sigma \in Aut_{\mathbb{F}}(A)$ which maps $\varepsilon^{(1)} = 2 + 5x + 2x^2 + 5x^3$ onto $\varepsilon^{(2)} = 2 + 2x + 2x^2 + 2x^3$ and vice versa. MAPLE supplies us with the irreducible polynomial $f := 5 + 3z + 3z^2 + 4z^3 + 6z^4 + z^5$. Put $G_0 := [\mathfrak{v}(\varepsilon^{(1)})] = [2, 5, 2, 5]$ and $G_1 := [\mathfrak{v}(\varepsilon^{(2)})] = [2, 2, 2, 2]$. Then the matrix

$$\begin{aligned} G &:= 5G_0 + 3zG_1 + 3z^2G_0 + 4z^3G_1 + 6z^4G_0 + z^5G_1 \\ &= 2 \cdot \begin{bmatrix} 5 + 3z + 3z^2 + 4z^3 + 6z^4 + z^5 \\ 2 + 3z + 4z^2 + 4z^3 + z^4 + z^5 \\ 5 + 3z + 3z^2 + 4z^3 + 6z^4 + z^5 \\ 2 + 3z + 4z^2 + 4z^3 + z^4 + z^5 \end{bmatrix}^{\mathrm{t}} \end{aligned}$$

is a generator matrix of a σ-cyclic $(4,1,5)$-convolutional code $\mathcal{C}$. We have $d_{free}(\mathcal{C}) \leq 6 \cdot 4 = 24$. With MAPLE and [GS02b] we calculated[11] *$d_{free}(\mathcal{C}) = 20$, hence the code is not an MDS convolutional code.*

(b) Once more we choose $\mathbb{F} = \mathbb{F}_4 = \{0, 1, \alpha, \alpha^2\}$, $n = 3$ and display A as in Example 3.12, i.e. $\mathfrak{v}(\varepsilon^{(1)}) = [1,1,1]$, $\mathfrak{v}(\varepsilon^{(2)}) = [1, \alpha^2, \alpha]$ and $\mathfrak{v}(\varepsilon^{(3)}) = [1, \alpha, \alpha^2]$. The automorphism σ defined by $\sigma(x) = \alpha^2 x$ satisfies $\sigma(\varepsilon^{(1)}) = \varepsilon^{(2)}$, $\sigma(\varepsilon^{(2)}) = \varepsilon^{(3)}$ and $\sigma(\varepsilon^{(3)}) = \varepsilon^{(1)}$. An irreducible polynomial of degree 4 over $\mathbb{F}_4$ is $f := 1 + \alpha z + z^3 + z^4$ and another one is $\hat{f} := \alpha + \alpha^2 z + z^2 + \alpha^2 z^3 + z^4$. The matrix

$$\begin{aligned} G &:= [\mathfrak{v}(\varepsilon^{(1)} + \alpha z \sigma(\varepsilon^{(1)}) + z^3 \sigma^3(\varepsilon^{(1)}) + z^4 \sigma^4(\varepsilon^{(1)}))] \\ &= [1,1,1] + \alpha z [1, \alpha^2, \alpha] + z^3 [1,1,1] + z^4 [1, \alpha^2, \alpha] \end{aligned}$$

is a generator matrix of a σ-cyclic $(3,1,4)$-convolutional code $\mathcal{C}$ with support $\{1\}$ and free distance $d_{free}(\mathcal{C}) \leq 4 \cdot 3 = 12$. The code $\mathcal{C}$ is, of course, not an MDS convolutional code.
The matrix

$$\begin{aligned} \hat{G} &:= [\mathfrak{v}(\alpha\varepsilon^{(1)} + \alpha^2 z \sigma(\varepsilon^{(1)}) + z^2 \sigma^2(\varepsilon^{(1)}) + \alpha^2 z^3 \sigma^3(\varepsilon^{(1)}) + z^4 \sigma^4(\varepsilon^{(1)}) \\ &= \alpha[1,1,1] + \alpha^2 z [1, \alpha^2, \alpha] + z^2 [1, \alpha, \alpha^2] + \alpha^2 z^3 [1,1,1] + z^4 [1, \alpha^2, \alpha] \end{aligned}$$

is a generator matrix of a σ-cyclic $(3,1,4)$-convolutional code $\hat{\mathcal{C}}$ with support $\{1\}$ and free distance $d_{free}(\mathcal{C}) \leq 5 \cdot 3 = 15$. The code $\mathcal{C}$ might be MDS. MAPLE and [GS02b] yield $d_{free}(\mathcal{C}) = 12$ and $d_{free}(\hat{\mathcal{C}}) = 14$, in particular $\hat{\mathcal{C}}$ is not MDS. Note that the generalized Griesmer bound given in [GS03] for $(3,1,4)$-convolutional codes over $\mathbb{F}_4$ is 14, thus $\hat{\mathcal{C}}$ is optimal.

We wish to mention that a generator matrix of a σ-cyclic $(n,1,\delta)$-convolutional code does not have one irreducible entry in general. Our very first Example 3.4 may illustrate this: There

$$G = [(z+\alpha)(z+\alpha^2), \alpha^2(z+1)(z+\alpha^2), \alpha(z+1)(z+\alpha)]$$

is a generator matrix of a $(3,1,2)$-convolutional code over $\mathbb{F}_4 = \{0, 1, \alpha, \alpha^2\}$ which is cyclic with respect to σ given by $\sigma(x) = \alpha^2 x$.

[11] Since the first and third resp. the second and fourth entry in $G =: [g_1, g_2, g_3, g_4]$ are the same, we tested $\hat{G} := [g_1, g_2]$ and found out that $\operatorname{im}_{\mathbb{F}[z]} \hat{G}$ has free distance 10.

5.3.2 Investigation of the Special Matrix $G_l^{(\delta)}$

A generator matrix G of a σ-cyclic $(n,1,\delta)$-convolutional code must have the shape as in Corollary 5.10 (ii), i.e. there exists some $l \in \{1,\dots,r_1\}$ such that

$$G = \sum_{i=0}^{\delta} a_i z^i [1, b_{\sigma^i}^{n-1}, b_{\sigma^i}^{n-2}, \dots, b_{\sigma^i}^2, b_{\sigma^i}]$$

for some $a_0,\dots,a_\delta \in \mathbb{F}$ where $a_0 \neq 0 \neq a_\delta$.

What happens if we choose $a_0 = \dots = a_\delta = 1$? We know from Corollary 5.7 that for $l \in \{1,\dots,r_1\}$ the matrix

$$G_l^{(\delta)} := \sum_{i=0}^{\delta} z^i [1, b_{\sigma^i}^{n-1}, b_{\sigma^i}^{n-2}, \dots, b_{\sigma^i}^2, b_{\sigma^i}] = n \cdot \mathfrak{v}\Big(\sum_{i=0}^{\delta} z^i \sigma^i(\varepsilon^{(l)})\Big) \tag{5.3}$$

is a generator matrix of a σ-cyclic $(n,1,\delta)$-submodule. But for which δ is this matrix basic and generates thereby a convolutional code? This section will give some answers to that question. We start with an example from [GSS02]:

Example 5.15 *cf. [GSS02, Example 3.4]*
We consider $n = 3$ and $\mathbb{F} = \mathbb{F}_4$. Let A be displayed as in Example 3.12, in particular we have $\mathfrak{v}(\varepsilon^{(1)}) = [1,1,1]$, $\mathfrak{v}(\varepsilon^{(2)}) = [1,\alpha^2,\alpha]$, $\mathfrak{v}(\varepsilon^{(3)}) = [1,\alpha,\alpha^2]$. We choose $\sigma(x) = \alpha^2 x$, this means $\sigma(\varepsilon^{(1)}) = \varepsilon^{(2)}$, $\sigma(\varepsilon^{(2)}) = \varepsilon^{(3)}$ and $\sigma(\varepsilon^{(3)}) = \varepsilon^{(1)}$ (cf. Example 3.16). Under these conditions we have

$$G_3^{(\delta)} = \sum_{i=0}^{\delta} z^i G_{i \bmod 3} \quad , \text{ where } G_0 = [1,\alpha,\alpha^2],\ G_1 = [1,1,1],\ G_2 = [1,\alpha^2,\alpha].$$

Glüsing-Lüerßen, Schmale and Striha claimed in [GSS02, Example 3.4] that $G_3^{(\delta)}$ is not basic for $\delta > 1$ and $\delta \equiv 2 \bmod 3$ (because in this case all entries in $G_3^{(\delta)}$ have a common divisor) and that $G_3^{(\delta)}$ is basic in all other cases. Furthermore, they computed

$$d_{free}\,(\mathrm{im}_{\mathbb{F}[z]} G_3^{(2)}) = 9\ ,\ d_{free}\,(\mathrm{im}_{\mathbb{F}[z]} G_3^{(3)}) = 12\ ,$$
$$d_{free}\,(\mathrm{im}_{\mathbb{F}[z]} G_3^{(4)}) = 13\ ,\ d_{free}\,(\mathrm{im}_{\mathbb{F}[z]} G_3^{(6)}) = 15$$

and pointed out that these free distances are quite good (in the first two cases even MDS).

We think that Example 5.15 justifies a further investigation of matrices as described in (5.3). It can be generalised immediately for the case $\delta > o_l - 1$, $\delta \equiv -1 \bmod o_l$:

Proposition 5.16
Let $\delta > o_l - 1$ and $\delta \equiv -1 \bmod o_l$. Then the matrix $G_l^{(\delta)}$ is not basic.

PROOF: First of all, recall that $b_{\sigma^i} = b_{\sigma^{i \bmod o_l}}$ for any $i \in \mathbb{N}_0$ (cf. Notation 5.8). If $\delta > o_l - 1$ and $\delta \equiv -1 \bmod o_l$, then $(1 + z^{o_l} + z^{2o_l} + \ldots + z^{\delta-(o_l-1)})$ is a common factor of the entries in $G_l^{(\delta)} = \sum_{i=0}^{\delta} z^i [1, b_{\sigma^i}^{n-1}, \ldots, b_{\sigma^i}]$, precisely $G_l^{(\delta)} = (1+z^{o_l}+z^{2o_l}+\ldots+z^{\delta-(o_l-1)})[\sum_{i=0}^{o_l-1} z^i, \sum_{i=0}^{o_l-1} b_{\sigma^i}^{n-1} z^i, \ldots, \sum_{i=0}^{o_l-1} b_{\sigma^i} z^i]$. □

There is a second case which is not hard to handle: If $\delta \leq o_l - 1$, then the matrix $G_l^{(\delta)}$ is basic (under the general assumptions made at the beginning of Section 5.3):

Proposition 5.17
For $\delta \leq o_l - 1$ the matrix $G_l^{(\delta)}$ is basic, hence $G_l^{(\delta)}$ is generator matrix of an $(n, 1, \delta)$-convolutional code.

PROOF: A canonical PQR-representation (P, Q, R) of $\operatorname{im}_{\mathbb{F}[z]} G_l^{(\delta)}$ is given by

$$[zP+Q \mid R] := \left[\begin{array}{ccc|cccc} z & & & 1 & b_{\sigma^0}^{n-1} & b_{\sigma^0}^{n-2} & \ldots & b_{\sigma^0} \\ -1 & \ddots & & 1 & b_{\sigma^1}^{n-1} & b_{\sigma^1}^{n-2} & \ldots & b_{\sigma^1} \\ & \ddots & z & \vdots & \vdots & \vdots & & \vdots \\ & & -1 & 1 & b_{\sigma^\delta}^{n-1} & b_{\sigma^\delta}^{n-2} & \ldots & b_{\sigma^\delta} \end{array}\right] \in \mathbb{F}[z]^{(\delta+1)\times(\delta+n)} .$$

(We refer to Section 2.2, in particular to Theorem 2.12 and its proof.) The matrix $G_l^{(\delta)}$ is basic if and only if it is right invertible (cf. Theorem 1.7), and this is the case if and only if $[zP+Q \mid R]$ is right invertible (cf. Theorem 2.13 (c)). Observe that, of course, $o_l \leq n$. Since we required $\delta \leq o_l - 1 \leq n - 1$, the elements $b_{\sigma^0}, \ldots, b_{\sigma^\delta} \in \mathbb{F} \setminus \{0\}$ are pairwise different and the first $\delta + 1$ columns of R show a Vandermonde-structure[12] because $b_{\sigma^\nu}^{n-\mu} = (b_{\sigma^\nu}^{-1})^\mu$. (We

[12]For $a_1, \ldots, a_m \in \mathbb{F}$ we have

$$\det \begin{bmatrix} 1 & a_1 & \ldots & a_1^{m-1} \\ 1 & a_2 & \ldots & a_2^{m-1} \\ \vdots & \vdots & & \vdots \\ 1 & a_m & \ldots & a_m^{m-1} \end{bmatrix} = \prod_{i<j} (a_j - a_i) \quad ,$$

can also argue that the last $\delta+1$ columns of R show a Vandermonde-structure up to multiplication of the rows of R with nonzero constants.) In particular, R has full rank. The latter implies that $[zP+Q \,|\, R]$ is right invertible (cf. Theorem 1.7 (iv)). □

In Propositions 5.16 and 5.17, we dealt on the one hand with the case $\delta > o_l - 1$, $\delta \equiv -1 \bmod o_l$ and on the other hand with the case $\delta \leq o_l - 1$. In the following we give some results for the remaining case $\delta > o_l - 1$ and $\delta \not\equiv -1 \bmod o_l$, but they are not as complete as in the other cases.

For the remaining part of this section we consider $G_l^{(\delta)}$ as in (5.3) where $\delta > o_l - 1$ and $\delta \not\equiv -1 \bmod o_l$.

Let f_1, f_2 denote the first two entries in $G_l^{(\delta)}$, i.e. $f_1 = \sum_{i=0}^{\delta} z^i$ and $f_1 = \sum_{i=0}^{\delta} b_{\sigma^i}^{-1} z^i$. Inspired by [GSS02, Example 3.4] (see Example 5.15) we tested coprimeness of f_1 and f_2 in many MAPLE-examples. (Coprimeness of f_1 and f_2 implies that $G_l^{(\delta)}$ is basic and that the submodule $\mathcal{C} := \mathrm{im}_{\mathbb{F}[z]} G$ is a convolutional code.) In all our examples f_1 and f_2 were coprime (but we suspect that this is not true in general).
We will develop a criterion for coprimeness of f_1 and f_2, which will allow us to solve some special cases, namely $o_l = 2$ ($\mathbb{F}$ arbitrary) and $o_l = 3$ where $\mathbb{F} = \mathbb{F}_4$ (cf. Proposition 5.18). For this purpose, write

$$\begin{aligned} f_1 &= \Big(\sum_{i=0}^{o_l-1} z^i\Big)(1 + z^{o_l} + \ldots + z^{ro_l}) + z^{(r+1)o_l}(1 + z + \ldots + z^m) \quad , \\ f_2 &= \Big(\sum_{i=0}^{o_l-1} b_{\sigma^i}^{-1} z^i\Big)(1 + z^{o_l} + \ldots + z^{ro_l}) + z^{(r+1)o_l}(b_{\sigma^0}^{-1} + b_{\sigma^1}^{-1} z + \ldots + b_{\sigma^m}^{-1} z^m) \end{aligned}$$

with some $r \in \mathbb{N}_0$ and $0 \leq m \leq o_l - 2$. This is always possible via division with remainder and since $b_{\sigma^i} = b_{\sigma^{i \bmod o_l}}$ for any $i \in \mathbb{N}_0$ (cf. Notation 5.8).
Any common divisor of f_1 and f_2 must be a common divisor of

$$\begin{aligned} & \Big(\sum_{i=0}^{o_l-1} b_{\sigma^i}^{-1} z^i\Big) f_1 - \Big(\sum_{i=0}^{o_l-1} z^i\Big) f_2 \\ = \; & z^{(r+1)o_l}\Big(\sum_{i=0}^{o_l-1} b_{\sigma^i}^{-1} z^i \sum_{i=0}^{m} z^i - \sum_{i=0}^{o_l-1} z^i \sum_{i=0}^{m} b_{\sigma^i}^{-1} z^i\Big) \end{aligned}$$

in particular this determinant (called the **Vandermonde determinant**) is nonzero if and only if $a_1, \ldots, a_m$ are pairwise different. This is a well known result in linear algebra. For reference see e.g. [Lan86, VII, §2, p. 208].

$$
\begin{aligned}
&= z^{(r+1)o_l}\Big(\Big(\sum_{i=0}^{m} b_{\sigma^i}^{-1} z^i + \sum_{i=m+1}^{o_l-1} b_{\sigma^i}^{-1} z^i\Big)\sum_{i=0}^{m} z^i - \Big(\sum_{i=0}^{m} z^i + \sum_{i=m+1}^{o_l-1} z^i\Big)\sum_{i=0}^{m} b_{\sigma^i}^{-1} z^i\Big) \\
&= z^{(r+1)o_l}\Big(\sum_{i=m+1}^{o_l-1} b_{\sigma^i}^{-1} z^i \sum_{i=0}^{m} z^i - \sum_{i=m+1}^{o_l-1} z^i \sum_{i=0}^{m} b_{\sigma^i}^{-1} z^i\Big) \\
&= z^{(r+1)o_l+(m+1)}\Big(\sum_{i=0}^{o_l-1-(m+1)} b_{\sigma^{i+m+1}}^{-1} z^i \sum_{i=0}^{m} z^i - \sum_{i=0}^{o_l-1-(m+1)} z^i \sum_{i=0}^{m} b_{\sigma^i}^{-1} z^i\Big) \quad .
\end{aligned}
$$

Since z does neither divide f_1 nor f_2, any common divisor of f_1 and f_2 must be a common divisor of

$$
g_{[m]} := \sum_{i=0}^{o_l-m-2} b_{\sigma^{i+m+1}}^{-1} z^i \sum_{i=0}^{m} z^i - \sum_{i=0}^{o_l-m-2} z^i \sum_{i=0}^{m} b_{\sigma^i}^{-1} z^i \quad ,
$$

which is a polynomial of z-degree at most $o_l - 2$. This criterion is used in the following proposition to investigate the cases $o_l = 2$ and $o_l = 3$:

Proposition 5.18

(a) If $o_l = 2$, then $G_l^{(\delta)}$ is basic if and only if $\delta = 1$ or δ is even.

(b) If $o_l = 3$ and $\mathbb{F} = \mathbb{F}_4$, then $G_l^{(\delta)}$ is basic if and only if $\delta = 2$ or $\delta \not\equiv 2 \bmod 3$. (This is a generalization of [GSS02, Example 3.4 (a)].)

PROOF: For part (a) we argue as follows: Along with the Propositions 5.16 and 5.17 we have that $G_l^{(\delta)}$ is basic for $\delta \in \{0, 1\}$ and that it is not basic if $\delta > 1$ and δ is odd. It remains to investigate the case where $\delta > 1$ is even, this is the case $\delta > o_l - 1$ and $\delta \not\equiv -1 \bmod o_l$. Hence we can apply the criterion from above to check, if the first two entries f_1 and f_2 of $G_l^{(\delta)}$ are coprime: The only way to choose $0 \le m \le o_l - 2$ is $m = 0$. Now $g_{[0]} = b_{\sigma^1}^{-1} - b_{\sigma^0}^{-1} \in \mathbb{F} \setminus \{0\}$ and we conclude that f_1 and f_2 are coprime. This implies the assertion.
For part (b) we use Propositions 5.16 and 5.17 again. They yield, that $G_l^{(\delta)}$ is basic for $\delta \in \{0, 1, 2\}$ and that it not basic if $\delta > 2$ and $\delta \equiv 2 \bmod 3$. The remaining case is $\delta > 2$ and $\delta \not\equiv 2 \bmod 3$ and we will use the criterion from above tho show, that the first two entries f_1 and f_2 of $G_l^{(\delta)}$ are coprime. This will finish the proof.
In the case $\delta > 2$ and $\delta \not\equiv 2 \bmod 3$ the choice of m according to $0 \le m \le o_l - 2$ yields the two cases $m = 0$ and $m = 1$. We have $g_{[0]} = (b_{\sigma^1}^{-1} - b_{\sigma^0}^{-1}) + (b_{\sigma^2}^{-1} - b_{\sigma^0}^{-1})z$ and $g_{[1]} = (b_{\sigma^2}^{-1} - b_{\sigma^0}^{-1}) + (b_{\sigma^2}^{-1} - b_{\sigma^1}^{-1})z$. Both $g_{[0]}$ and $g_{[1]}$ are linear polynomials, hence f_1 and f_2 are coprime if they have no common root together with $g_{[0]}$ resp. $g_{[1]}$. This is in particular true if we can show that $1, \alpha, \alpha^2$ are no common

roots of f_1 and f_2, where $\mathbb{F}_4 = \{0, 1, \alpha, \alpha^2\}$. As for the common roots, we argue as follows:
For $\zeta \in \mathbb{F} \setminus \{0\}$ we have $\zeta^3 = 1$. If we put $h_1(\zeta) = 1 + \zeta + \zeta^2$, $h_2(\zeta) := b_{\sigma^0}^{-1} + b_{\sigma^1}^{-1}\zeta + b_{\sigma^2}^{-1}\zeta^2$ we get

$$\begin{aligned} f_1(\zeta) &= (r+1)h_1(\zeta) + \begin{cases} 1 & \text{for } m = 0 \\ 1+\zeta & \text{for } m = 1 \end{cases} \\ f_2(\zeta) &= (r+1)h_2(\zeta) + \begin{cases} b_{\sigma^0}^{-1} & \text{for } m = 0 \\ b_{\sigma^0}^{-1} + b_{\sigma^1}^{-1}\zeta & \text{for } m = 1 \end{cases} . \end{aligned}$$

If r is odd, then

$$f_1(\zeta) = \begin{cases} 1 & \text{for } m = 0 \\ 1+\zeta & \text{for } m = 1 \end{cases}, \; f_2(\zeta) = \begin{cases} b_{\sigma^0}^{-1} & \text{for } m = 0 \\ b_{\sigma^0}^{-1} + b_{\sigma^1}^{-1}\zeta & \text{for } m = 1 \end{cases} .$$

In this case $f_1(\zeta) = 0$ implies $m = 1$ and $\zeta = 1$. But then $f_2(\zeta) = f_2(1) = b_{\sigma^0}^{-1} + b_{\sigma^1}^{-1} \neq 0$ since $b_{\sigma^0} \neq b_{\sigma^1}$.
If r is even, then

$$f_1(\zeta) = \begin{cases} \zeta + \zeta^2 & \text{for } m = 0 \\ \zeta^2 & \text{for } m = 1 \end{cases}, \; f_2(\zeta) = \begin{cases} b_{\sigma^1}^{-1}\zeta + b_{\sigma^2}^{-1}\zeta^2 & \text{for } m = 0 \\ b_{\sigma^2}^{-1}\zeta^2 & \text{for } m = 1 \end{cases} .$$

In this case $f_1(\zeta) = 0$ implies $m = 0$ and $\zeta = 1$. But then $f_2(\zeta) = f_2(1) = b_{\sigma^1}^{-1} + b_{\sigma^2}^{-1} \neq 0$ because $b_{\sigma^1} \neq b_{\sigma^2}$.
This shows that $(f_1(\zeta), f_2(\zeta)) \neq (0, 0)$ for any $\zeta \in \{1, \alpha, \alpha^2\}$, hence f_1 and f_2 have no common roots in $\mathbb{F}_4$. □

Although we have no complete solution for the case $\delta > o_l - 1$ and $\delta \not\equiv -1 \bmod o_l$, we have a complete solution for $\mathbb{F} \in \{\mathbb{F}_3, \mathbb{F}_4\}$ (for $\mathbb{F} = \mathbb{F}_2$ recall Corollary 5.6 (a)): In the case $\mathbb{F} = \mathbb{F}_3$ the polynomial $x^n - 1$ has at most two linear prime divisors and $o_l \geq 2$ implies $o_l = 2$. In the case $\mathbb{F} = \mathbb{F}_4$ the polynomial $x^n - 1$ has at most three linear prime divisors, hence $R^{(1)} = \{1, \ldots, r_1\} \subseteq \{1, 2, 3\}$. For $l \in R^{(1)}$ the l-order of an automorphism $\sigma \in Aut_{\mathbb{F}}(A)$, which gives rise to 1-dimensional σ-cyclic convolutional codes with nonzero complexity and support $\{l\}$, is 2 or 3. As a consequence we know exactly under which conditions $G_l^{(\delta)}$ is basic, cf. Proposition 5.18:

Corollary 5.19
(a) Consider $\mathbb{F} = \mathbb{F}_3$. Then $o_l = 2$ and $G_l^{(\delta)}$ is basic if and only if $\delta = 1$ or δ is even.
(b) Consider $\mathbb{F} = \mathbb{F}_4$. Then $o_l = 2$ or $o_l = 3$ and we have: If $o_l = 2$ then $G_l^{(\delta)}$ is basic if and only if $\delta = 1$ or δ is even. If $o_l = 3$ then $G_l^{(\delta)}$ is basic if and only if $\delta = 2$ or $\delta \not\equiv 2 \bmod 3$. □

5.3.3 Construction of Cyclic $(n,1,\delta)$-Convolutional Codes with Good Free Distance

In this section we will deal with $G_l^{(\delta)}$ as in (5.3) where $\delta \leq o_l - 1$. We know from Proposition 5.17 that $\mathcal{C} := \operatorname{im}_{\mathbb{F}[z]} G_l^{(\delta)}$ is a σ-cyclic $(n,1,\delta)$-convolutional code and that it satisfies

$$2n \leq d_{free}(\mathcal{C}) \leq n(\delta+1) \quad ,$$

as we saw in Corollary 5.10. For $\delta = 1$ the upper and the lower bound coincide, but if $\delta > 1$ we may wish to find better lower bounds for $d_{free}(\mathcal{C})$ than $2n$.

Roughly speaking, we intend to consider a special field $\mathbb{F}$ and to construct a special automorphism σ such that the formula

$$\big(n - \frac{\delta}{2}\big)(\delta+1) \leq d_{free}(\mathcal{C}) \leq n(\delta+1) \quad . \tag{5.4}$$

holds true. The lower bound $(n - \frac{\delta}{2})(\delta+1)$ is better than $2n$ for $n \geq 4$ and $\delta > 1$:

Lemma 5.20
For $n, \delta \in \mathbb{N}$ satisfying $n \geq 4$ and $n-1 \geq \delta \geq 2$ we have $2n < (n - \frac{\delta}{2})(\delta+1)$.

PROOF: After some manipulations we get that $2n < (n - \frac{\delta}{2})(\delta+1)$ is equivalent to $0 > \delta^2 + \delta(1-2n) + 2n$. We investigate the polynomial $f(D) := D^2 + D(1-2n) + 2n$, which has the two roots

$$\delta_+(n) := n - \frac{1}{2} + \sqrt{n^2 - 3n + \frac{1}{4}} \quad \text{and} \quad \delta_-(n) := n - \frac{1}{2} - \sqrt{n^2 - 3n + \frac{1}{4}}$$

and we have to show that $\delta \in \{d \mid f(d) < 0\}$ for $n \geq 4$. One can readily show that $n^2 - 3n + \frac{1}{4} = (n - \frac{3}{2} + \sqrt{2})(n - \frac{3}{2} - \sqrt{2})$ is positive if and only if $n \geq 3$ (because $n \in \mathbb{N}$), hence $\{d \mid f(d) < 0\} \neq \emptyset$ if and only if $n \geq 3$.
Now we assume $n \geq 3$. In this case $\{d \mid f(d) < 0\} = \{d \mid \delta_-(n) < d < \delta_+(n)\}$. Note that $\delta \leq n-1 < n - \frac{1}{2} < \delta_+(n)$. Furthermore, $\delta_-(3) = 2$ and one can show that $\delta_-(n)$ is decreasing for $n \geq 3$, hence $\delta_-(n) < 2$ for $n \geq 4$. Therefore $\delta_-(n) < \delta$ if $n \geq 4$. This implies the assertion. □

If a σ-cyclic $(n,1,\delta)$-convolutional code $\mathcal{C}$ satisfies (5.4), then the foregoing Lemma tells us that its free distance is better than $2n$ if $n \geq 4$ and δ is not too big. However, so far it is not clear whether $d_{free}(\mathcal{C})$ is "much" bigger than $2n$, i.e. whether $\mathcal{C}$ has "good" free distance. If n is much bigger than δ, then the next remark will allow us to say that $\mathcal{C}$ has good free distance:

Remark 5.21

(a) In the formula (5.4), the ratio of the lower and the upper bound is given by $(n - \frac{\delta}{2})/n = 1 - \frac{\delta}{2n}$. *For fixed* δ *and "big"* n *this ratio is nearly* 1. *In this sense, codes satisfying (5.4) are asymptotically MDS-convolutional codes for fixed* δ *and* n *going to infinity.*

(a) The difference between the lower and upper bound in (5.4) is $\frac{1}{2}\delta(\delta+1)$, *which is independent from* n. *So, again, a code satisfying (5.4) is not far from being an MDS-code if* δ *is fixed and* n *is "big".*

In the following we will deal with a finite field $\mathbb{F}$ where $x^n - 1$ splits into linear prime divisors over $\mathbb{F}$. In this case we know (e.g. from [MS78, Ch. 7, §5]) that there exists an n-th primitive root of unity α in $\mathbb{F}$, i.e. $\alpha^n = 1$ and $\alpha^m \neq 1$ for $0 < m < n$. Furthermore, the prime factor decomposition of $x^n - 1$ is given by

$$x^n - 1 = \prod_{i=0}^{n-1}(x - \alpha^i) \quad .$$

With this knowledge, we can construct a special automorphism $\sigma \in Aut_{\mathbb{F}}(A)$ and a special σ-cyclic $(n, 1, \delta)$-code which satisfies (5.4).

Theorem 5.22

Let $\alpha \in \mathbb{F}$ *be an* n*-th primitive root of unity. This implies* $x^n - 1 = \prod_{i=0}^{n-1}(x - \alpha^i)$. *Furthermore, let* $\varepsilon^{(i)}$ *be the primitive idempotent element in* A *corresponding to the prime divisor* $(x - \alpha^i)$. *Then we have:*

(a) If $\sigma \in Aut_{\mathbb{F}}(A)$ *is the (uniquely determined) automorphism which satisfies*

$$\sigma(\varepsilon^{(1)}) = \varepsilon^{(2)} \ , \ \sigma(\varepsilon^{(2)}) = \varepsilon^{(3)} \ , \ \ldots \ , \ \sigma(\varepsilon^{(n-1)}) = \varepsilon^{(n)} \ , \ \sigma(\varepsilon^{(n)}) = \varepsilon^{(1)}$$

and if we put $\beta := \alpha^{-1}$, *then we have* $n\mathfrak{v}(\varepsilon^{(i)}) = [1, \beta^i, \beta^{2i}, \ldots, \beta^{(n-1)i}]$ *for* $1 \leq i \leq n$ *and*

$$G_1^{(\delta)} = \sum_{i=0}^{\delta} z^i [1, \beta^{i+1}, \beta^{2(i+1)}, \ldots, \beta^{(n-1)(i+1)}]$$

for any $\delta \in \mathbb{N}_0$, *where* $G_1^{(\delta)}$ *is defined as in (5.3).*

(b) If $n \geq 4$, σ *was chosen as in (a) and if* $2 \leq \delta \leq n-1$, *then the* σ*-cyclic* $(n, 1, \delta)$*-convolutional code* $\mathcal{C} := \mathrm{im}_{\mathbb{F}[z]} G_l^{(\delta)}$ *satisfies (5.4), precisely*

$$2n < \big(n - \frac{\delta}{2}\big)(\delta + 1) \leq d_{free}(\mathcal{C}) \leq n(\delta + 1) \quad .$$

PROOF: Part (a): With Lemma 5.9 along with $\alpha^n = 1 = \beta^n$ we obtain $n\mathfrak{v}(\varepsilon^{(i)}) = [1, \beta^i, \beta^{2i}, \ldots, \beta^{(n-1)i}]$ for $1 \leq i \leq n$. If we put $l = 1$ and if we use Notation 5.8, then the automorphism σ satisfies

$$b_{\sigma^i} = \alpha^{i+1} \quad , \quad b_{\sigma^i}^{-1} = \beta^{i+1} \quad .$$

This shows that $G_1^{(\delta)} = \sum_{i=0}^{\delta} z^i [1, \beta^{i+1}, \beta^{2(i+1)}, \ldots, \beta^{(n-1)(i+1)}]$.

For part (b) we recall Lemma 5.20, which proves that $2n < \left(n - \frac{\delta}{2}\right)(\delta+1)$. It remains to show that $\left(n - \frac{\delta}{2}\right)(\delta+1) \leq d_{free}(\mathcal{C})$.
We need PQR-representations again. In analogy to the proof of Proposition 5.17 and with Theorem 2.12 and its proof we obtain that a canonical PQR-representation (P, Q, R) of $\operatorname{im}_{\mathbb{F}[z]} G_1^{(\delta)}$ is given by

$$[zP + Q \mid R] := \left[\begin{array}{ccc|cccc} z & & & 1 & \beta & \beta^2 & \ldots & \beta^{n-1} \\ -1 & \ddots & & 1 & \beta^2 & \beta^4 & \ldots & \beta^{2(n-1)} \\ & \ddots & z & \vdots & \vdots & \vdots & & \vdots \\ & & -1 & 1 & \beta^{\delta+1} & \beta^{(\delta+1)2} & \ldots & \beta^{(\delta+1)(n-1)} \end{array}\right] \in \mathbb{F}[z]^{(\delta+1)\times(\delta+n)} \ .$$

We have to show that $\operatorname{wt}(v) \geq (n - \frac{\delta}{2})(\delta+1)$ for $0 \neq v \in \mathcal{C}$. Towards this end we define for $d \in \{0, 1, \ldots, \delta\}$ the matrices

$$R_d := \left[\begin{array}{ccccc} 1 & \beta & \beta^2 & \ldots & \beta^{n-1} \\ 1 & \beta^2 & \beta^4 & \ldots & \beta^{2(n-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \beta^{d+1} & \beta^{(d+1)2} & \ldots & \beta^{(d+1)(n-1)} \end{array}\right] \in \mathbb{F}^{(d+1)\times n} \quad . \tag{5.5}$$

Note, that the first $d+1$ columns of R_d show a Vandermonde-structure[13]. Hence the matrix R_d has full row rank for $0 \leq d \leq \delta$ since $1, \beta, \ldots, \beta^d \in \mathbb{F} \setminus \{0\}$ are pairwise different.
With Corollary 2.11 we have $\mathcal{C} = \left(\operatorname{im}_{\mathbb{F}[z]}[1, z, \ldots, z^\delta]\right) \cdot R_\delta$. Hence $0 \neq v \in \mathcal{C}$ can be written as ζR_δ for some $0 \neq \zeta \in \operatorname{im}_{\mathbb{F}[z]}[1, z, \ldots, z^\delta]$ and we have to show that $\operatorname{wt}(\zeta R_\delta) \geq (n - \frac{\delta}{2})(\delta+1)$.

We write $\zeta = (\sum_{i\geq 0} z^i u_i)[1, z, \ldots, z^\delta] = \sum_{i\geq 0} \zeta_i z^i$ where $u_i \in \mathbb{F}$, $\zeta_i \in \mathbb{F}^{\delta+1}$. Because of right invertibility of $[1, z, \ldots, z^\delta]$ along with Theorem 1.7 (vi) we may assume without loss of generality that $\zeta_0 \neq 0$, which means $u_0 \neq 0$. It is easy to see that $\zeta_i = [u_i, u_{i-1}, \ldots, u_1, u_0, 0, \ldots, 0] \neq 0$ for $0 \leq i \leq \delta$ and therefore $0 \neq \zeta_i R_\delta \in \operatorname{im}_{\mathbb{F}} R_i$ for $0 \leq i \leq \delta$. We will show later that the block code $\operatorname{im}_{\mathbb{F}} R_i$ has distance $n - i$, hence $\operatorname{wt}(\zeta_i R) \geq n - i$ for $0 \leq i \leq \delta$.

[13] See footnote 12 on page 77.

This implies

$$\mathrm{wt}\,(\zeta R) = \sum_{i\geq 0} \mathrm{wt}\,(\zeta_i R) \geq \sum_{i=0}^{\delta} \underbrace{\mathrm{wt}\,(\zeta_i R)}_{\geq n-i} \geq \sum_{i=0}^{\delta} (n-i) = n(\delta+1) - \frac{\delta(\delta+1)}{2}$$

which yields $\mathrm{wt}\,(\zeta R) \geq (n-\frac{\delta}{2})(\delta+1)$. This yields the lower bound of $d_{free}\,(\mathcal{C})$.

It remains to show that the block code $\mathrm{im}_{\mathbb{F}} R_i$ has distance $n-i$ for $0 \leq i \leq \delta$. This means, we have to show that $\mathrm{im}_{\mathbb{F}} R_i$ is an MDS-block code. Since a block code is MDS if and only if its dual code is MDS [14], we can also show that $\ker_{\mathbb{F}} R_i^{\mathrm{t}}$ has free distance $i+2$. Towards this end we show that any selection of $i+1$ columns of R_i is $\mathbb{F}$-linear independent. Then Lemma 1.1 yields the desired result.

We consider the columns $j_0+1, \ldots, j_i+1$ of R_i, the corresponding matrix is given by

$$\begin{bmatrix} \beta^{j_0} & \beta^{j_1} & \beta^{j_2} & \ldots & \beta^{j_i} \\ \beta^{2j_0} & \beta^{2j_1} & \beta^{2j_2} & \ldots & \beta^{2j_i} \\ \vdots & \vdots & \vdots & & \vdots \\ \beta^{(i+1)j_0} & \beta^{(i+1)j_1} & \beta^{(i+1)j_2} & \ldots & \beta^{(i+1)j_i} \end{bmatrix} \in \mathbb{F}^{(i+1)\times(i+1)} \quad .$$

Multiplying the $(\nu+1)$-th column with $(\beta^{j_\nu})^{-1} \in \mathbb{F} \setminus \{0\}$ for $0 \leq \nu \leq i$ does not effect the rank of the matrix and we get the transformed matrix

$$\begin{bmatrix} 1 & 1 & 1 & \ldots & 1 \\ \beta^{j_0} & \beta^{j_1} & \beta^{j_2} & \ldots & \beta^{j_i} \\ (\beta^{j_0})^2 & (\beta^{j_1})^2 & (\beta^{j_2})^2 & \ldots & (\beta^{j_i})^2 \\ \vdots & \vdots & \vdots & & \vdots \\ (\beta^{j_0})^i & (\beta^{j_1})^i & (\beta^{j_2})^i & \ldots & (\beta^{j_i})^i \end{bmatrix} \in \mathbb{F}^{(i+1)\times(i+1)} \quad . \tag{5.6}$$

The elements $\beta^{j_0}, \ldots, \beta^{j_i}$ are pairwise different since $j_0, \ldots, j_i < n$ and $\beta = \alpha^{-1}$ is an n-th primitive root of unity as well. With the Vandermonde-structure [15] of the matrix in (5.6), we know that this matrix has full rank. The latter implies that any selection of $i+1$ columns of R_i is $\mathbb{F}$-linear independent. □

[14] This is a well known fact from the theory of block codes. One can readily show it with the Lemmata 1.1 and 1.9.

[15] See footnote 12 on page 77.

Remark 5.23
It is worth mentioning that the matrices R_i in (5.5) have two nice properties: They generate the MDS-block codes $\operatorname{im}_{\mathbb{F}} R_i$ and they are parity check matrices of BCH-codes (cf. [MS78, Ch. 7, §6]). In particular, the block codes $\operatorname{im}_{\mathbb{F}} R_i$ are cyclic because they are dual to BCH-codes.

Before we finish this section with some comments on the construction of convolutional codes, we give an example where we construct a σ-cyclic $(n, 1, \delta)$-convolutional code according to the method given in Theorem 5.22.

Example 5.24
We consider $n = 5$ and $\mathbb{F} = \mathbb{F}_{2^4} = \{0, 1, \omega, \omega^2, \ldots, \omega^{14}\}$, where $\omega^4+\omega+1 = 0$. In this case $x^n - 1$ factors into linear prime divisors. The element $\alpha := \omega^3$ is a 5-th primitive root of unity, hence

$$x^5 - 1 = (x + \alpha)(x + \alpha^2)(x + \alpha^3)(x + \alpha^4)(x + \alpha^5) \quad .$$

The inverse of α is given by $\beta := \omega^{12} = 1 + \omega + \omega^2 + \omega^3$ and we display A such that $\varepsilon^{(i)}$ corresponds to the prime factor $x + \alpha^i$. This implies $\mathfrak{v}(\varepsilon^{(i)}) = [1, \beta^i, \beta^{i2}, \beta^{i3}, \beta^{i4}]$ for $1 \le i \le 5$. We calculate

$$\begin{aligned}
\mathfrak{v}(\varepsilon^{(1)}) &= [1, \omega^{12}, \omega^{24}, \omega^{36}, \omega^{48}] = [1, \omega^2 + \omega^3 + \omega + 1, \omega + \omega^3, \omega^3 + \omega^2, \omega^3] \quad , \\
\mathfrak{v}(\varepsilon^{(2)}) &= [1, \omega^{24}, \omega^{48}, \omega^{72}, \omega^{96}] = [1, \omega + \omega^3, \omega^3, \omega^2 + \omega^3 + \omega + 1, \omega^3 + \omega^2] \quad , \\
\mathfrak{v}(\varepsilon^{(3)}) &= [1, \omega^{36}, \omega^{76}, \omega^{108}, \omega^{144}] = [1, \omega^3 + \omega^2, \omega^2 + \omega^3 + \omega + 1, \omega^3, \omega + \omega^3] \quad , \\
\mathfrak{v}(\varepsilon^{(4)}) &= [1, \omega^{48}, \omega^{92}, \omega^{144}, \omega^{192}] = [1, \omega^3, \omega^3 + \omega^2, \omega + \omega^3, \omega^2 + \omega^3 + \omega + 1] \quad , \\
\mathfrak{v}(\varepsilon^{(5)}) &= [1, \omega^{60}, \omega^{120}, \omega^{180}, \omega^{240}] = [1, 1, 1, 1, 1] \quad .
\end{aligned}$$

Choose σ as in Theorem 5.22 (a). Now we have

$$\begin{aligned}
G_1^{(0)} &= \mathfrak{v}(\varepsilon^{(1)}) \quad , \\
G_1^{(1)} &= \mathfrak{v}(\varepsilon^{(1)}) + z\mathfrak{v}(\varepsilon^{(2)}) \quad , \\
G_1^{(2)} &= \mathfrak{v}(\varepsilon^{(1)}) + z\mathfrak{v}(\varepsilon^{(2)}) + z^2\mathfrak{v}(\varepsilon^{(3)}) \quad , \\
G_1^{(3)} &= \mathfrak{v}(\varepsilon^{(1)}) + z\mathfrak{v}(\varepsilon^{(2)}) + z^2\mathfrak{v}(\varepsilon^{(3)}) + z^3\mathfrak{v}(\varepsilon^{(4)}) \quad , \\
G_1^{(4)} &= \mathfrak{v}(\varepsilon^{(1)}) + z\mathfrak{v}(\varepsilon^{(2)}) + z^2\mathfrak{v}(\varepsilon^{(3)}) + z^3\mathfrak{v}(\varepsilon^{(4)}) + z^4\mathfrak{v}(\varepsilon^{(5)}) \quad .
\end{aligned}$$

From Theorem 5.22 we know that $\mathcal{C}_i := \operatorname{im}_{\mathbb{F}[z]} G_1^{(i)}$ is a σ-cyclic $(5, 1, i)$-convolutional code satisfying $(5-\frac{i}{2})(i+1) \le d_{free}(\mathcal{C}_i) \le 5(i+1)$ for $0 \le i \le 4$. With Corollary 5.10, we know that $d_{free}(\mathcal{C}_0) = n = 5$ and $d_{free}(\mathcal{C}_1) = 2n = 10$. We used MAPLE and [GS02b] to calculate $d_{free}(\mathcal{C}_2) = 15$, $d_{free}(\mathcal{C}_3) = 20$ and $d_{free}(\mathcal{C}_4) = 25$. So all the codes in question are MDS-codes.

There are some methods to construct a (not necessarily cyclic) convolutional code $\mathcal{C}$, which guarantee that $d_{free}(\mathcal{C})$ has quite a good lower bound d_{low}. For most of these methods, the construction of such a code uses a block code $\hat{\mathcal{C}}$ with distance exactly d_{low} (examples are given in [Jus73] or [Pir88b]). In any case we may ask why we should use $\mathcal{C}$ to encode messages but rather $\hat{\mathcal{C}}$ — the structure of the code $\hat{\mathcal{C}}$ is simpler than that one of $\mathcal{C}$ and its free distance is in general (by construction) quite good.
We want to illustrate this by sketching the method in [Pir88b]: Here Piret constructs an $(n, k, n-k)$-convolutional code $\mathcal{C} \subseteq \mathbb{F}[z]^n$ over $\mathbb{F}_{2^m}$ with free distance at least $2n - 2k + 1 =: d_{low}$. Towards this end, he specifies a parity check matrix $H \in \mathbb{F}[z]^{n\times(n-k)}$ of $\mathcal{C}$ of the form $H = H_0 + zH_1$, where $H_0, H_1 \in \mathbb{F}^{n\times(n-k)}$ show a strong structure (similar to that one of the matrices defined in (5.5)). Piret assumes that $(n-k) \leq n/2$. For the proof of $d_{free}(\mathcal{C}) \geq d_{low}$ he uses that in his construction $[H_1, H_2] \in \mathbb{F}^{n\times(2n-2k)}$ is a parity check matrix of an $(n, 2k-n)$-MDS-block code $\hat{\mathcal{C}}$, i.e. $d_{free}(\hat{\mathcal{C}}) = d_{low}$. Why not using the MDS-block code $\hat{\mathcal{C}}$ for the encoding procedure instead of the convolutional code $\mathcal{C}$?
Theorem 5.22 provides a method to construct a σ-cyclic $(n, 1, \delta)$-convolutional code $\mathcal{C}$ with free distance at least $d_{low} := (n - \frac{\delta}{2})(\delta + 1)$. We would like to point out that in contrast to the references mentioned above we do not use a block code (with distance d_{low}) as the starting point of our construction.

6 Some Open Problems

In this final section we will point out some problems which where not solved in the present thesis (Section 6.1). After that we introduce a further research topic in Section 6.2 which deals with the so called "equivalence" of convolutional codes.

6.1 Some Questions which Arose in the Foregoing Sections

Uniqueness of a decomposition? In Corollary 3.22 resp. Remark 4.3, we learned how we can construct a decomposition of a σ-cyclic convolutional code into a direct sum of minimal σ-cyclic convolutional sub-codes (as left $\mathbb{F}[z]$-modules). Such decompositions are not unique. For example, we consider $\mathbb{F} = \mathbb{F}_4 = \{0, 1, \alpha, \alpha^2\}$, $n = 3$ and the automorphism σ defined by $\sigma(x) = \alpha^2 x$. One can show that the σ-cyclic (block) code

$$\mathcal{C} := \operatorname{im}_{\mathbb{F}[z]} \begin{bmatrix} 1 & 1 & 1 \\ 1 & \alpha^2 & \alpha \end{bmatrix} = \operatorname{im}_{\mathbb{F}[z]}[1,1,1] \oplus \operatorname{im}_{\mathbb{F}[z]}[1,\alpha^2,\alpha]$$

has the following decomposition into minimal σ-cyclic convolutional sub-codes, too:

$$\mathcal{C} = \operatorname{im}_{\mathbb{F}[z]}[1+z, 1+\alpha^2 z, 1+\alpha z] \oplus \operatorname{im}_{\mathbb{F}[z]}[1,\alpha^2,\alpha] \quad .$$

One would possibly prefer the first decomposition, because in the second one z appears, pretending that $\mathcal{C}$ has nonzero complexity. So different decompositions have different quality.
Consider a σ-cyclic (n,k,δ)-convolutional code $\mathcal{C}$. Then the sum of the complexities of the direct summands in the decomposition of $\mathcal{C}$ as in Remark 4.3 is equal to δ, we say "the decompositions respects the complexity of $\mathcal{C}$". This is due to the fact that in this decomposition every summand is the $\mathbb{F}[z]$-span of some rows of a minimal basis of $\mathcal{C}$. We expect (loosely speaking) that a decomposition, which respects the complexity of $\mathcal{C}$, is linked with a minimal basis and vice versa. If this is true: Can we find something like uniqueness of a decomposition into a direct sum of minimal σ-cyclic convolutional sub-codes which respects the complexity of $\mathcal{C}$ (up to certain operations with the summands)? Is this analogous to Proposition 1.5?

Determination of units? With Theorem 4.6, we know that we can determine all minimal σ-cyclic convolutional codes with the knowledge of all

units in $A[z;\sigma]$. Theorem 4.11 (b) tells us that we can construct all units via finite products of elementary units. But if we aim to construct all components of units with prescribed degree d, it is not clear how we can find these by building products of elementary units without further rules or knowledge of the algebraic structure of the units (cf. Remark 4.21). So we may ask: Can we find more information about the algebraic structure of units? How "good" is the description of units given in Theorem 4.11 (b) and do there exist more powerful (or "beautiful") ones?

Free distance of a σ-cyclic convolutional code? We have only few lower and upper bounds for the free distance of convolutional codes. The computer packages which we applied to determine the free distances of the codes in our examples did not use the cyclic structure of the codes in question. It is not clear if there is any link between the free distance and cyclicity in general. If so: Are there any estimates for the free distance of a σ-cyclic code in terms of a reduced generator polynomial and of the automorphism σ?
In this connection we may ask: Do the construction methods of 1-dimensional σ-cyclic convolutional codes given in Sections 5.3.1 and 5.3.2 yield codes with good free distance?
For the codes constructed in Section 5.3.3 (cf. Theorem 5.22) we have quite a good lower bound, but in all examples (cf. Example 5.24) the codes were MDS-convolutional codes. Are there examples where the lower bound given in Theorem 5.22 is tight?

More information about $G_l^{(\delta)}$? In Section 5.3.2 we introduced the matrix $G_l^{(\delta)}$. We found out that $G_l^{(\delta)}$ is not basic for $\delta > o_l - 1$, $\delta \equiv -1 \bmod o_l$ and that it is basic for $\delta < o_l - 1$ (provided that $\sigma(\varepsilon^{(l)}) \neq \varepsilon^{(l)}$). But we were not able determine in general whether $G_l^{(\delta)}$ is basic or not in the remaining case $\delta > o_l - 1$ and $\delta \not\equiv -1 \bmod o_l$. What is the answer to that problem?
Of course we can also ask if we can determine the free distance of im $_{\mathbb{F}[z]}G_l^{(\delta)}$.

6.2 The Concept of Equivalence of Codes – A Case Study

Imagine that we have two convolutional codes $\mathcal{C}, \hat{\mathcal{C}} \subseteq \mathbb{F}[z]^n$ and a linear bijective mapping

$$\varphi \ : \ \mathcal{C} \to \hat{\mathcal{C}} \quad \text{satisfying} \quad \mathrm{wt}\,(v) = \mathrm{wt}\,(\varphi(v)) \quad \text{for all} \quad v \in \mathcal{C}. \tag{6.1}$$

Two such codes may be different, but they behave "equal" concerning the weight of their codewords. For theoretical investigations, it does not matter

if we use $\mathcal{C}$ or $\hat{C}$ for the encoding procedure. However, in applications one of the codes might be preferred (for example, because it can be implemented more easily). From this point of view, it is worthwhile to know if two codes satisfy (6.1).

In the following we will use a stronger[16] definition of equivalence of codes leaving the weight distribution invariant.

Definition 6.1 *cf. [Bet98, p. 26f]*
Two (n,k)-submodules $\mathcal{C},\mathcal{C}' \subseteq \mathbb{F}[z]^n$ are called **equivalent** *if there exist $\alpha_1,\ldots,\alpha_n \in \mathbb{F}\setminus\{0\}$ and a permutation $\pi \in S_n$ such that*

$$\mathcal{C}' = \{(\alpha_1 v_{\pi(1)},\ldots,\alpha_n v_{\pi(n)}) \mid (v_1,\ldots,v_n) \in \mathcal{C}\} \quad .$$

We can express this equivalently via a generator matrix G of $\mathcal{C}$, because $\mathcal{C}$ and $\mathcal{C}'$ are equivalent if and only if there exist $\alpha_1,\ldots,\alpha_n \in \mathbb{F}\setminus\{0\}$ and a permutation matrix $P \in \mathbb{F}^{n\times n}$ such that $\mathcal{C}' = \operatorname{im}_{\mathbb{F}[z]}(G\cdot P\cdot \operatorname{diag}(\alpha_1,\ldots,\alpha_n))$.

Of course, we have an equivalence relation "$\sim$" on the set of all (n,k)-convolutional codes defined by $[\mathcal{C}\sim\hat{\mathcal{C}} :\Leftrightarrow \mathcal{C}$ and $\hat{\mathcal{C}}$ are equivalent$]$. If we apply the concept of equivalence to 1-dimensional submodules of $\mathbb{F}[z]^n$, we get a very simple description because generator matrices of 1-dimensional submodules are unique up to multiplication with a nonzero element in $\mathbb{F}$ (cf. Proposition 5.7 (a)):

Remark 6.2
Let $\mathcal{C}\subseteq\mathbb{F}[z]^n$ be an $(n,1)$-submodule with generator matrix $G=[g_1,\ldots,g_n]\in\mathbb{F}[z]^{1\times n}$. An $(n,1)$-submodule $\mathcal{C}'\subseteq\mathbb{F}[z]^n$ is equivalent to $\mathcal{C}$ if and only if there exist $\alpha_1,\ldots,\alpha_n\in\mathbb{F}\setminus\{0\}$ and a permutation $\pi\in S_n$ such that

$$G' = [\alpha_1 g_{\pi(1)},\ldots,\alpha_n g_{\pi(n)}] \quad \textit{is a generator matrix of } \mathcal{C}' \quad .$$

We are interested in the question how the set of all cyclic $(n,1,\delta)$-convolutional codes (cyclic with respect to possibly different automorphisms) over a field $\mathbb{F}$ decomposes into equivalence classes $\mathfrak{C}_1,\ldots,\mathfrak{C}_\mu$. Are there certain patterns? And if we consider the subset of all cyclic $(n,1,\delta)$-convolutional codes with a prescribed support, do the equivalence classes have a similar structure as $\mathfrak{C}_1,\ldots,\mathfrak{C}_\mu$?
We will examine these questions in one specific example.

[16] So far it is unknown if the definition of equivalence given in Definition 6.1 coincides with the definition given in (6.1).

A case study. We consider $n = 3$ and $\mathbb{F} = \mathbb{F}_4 = \{0, 1, \alpha, \alpha^2\}$. We have $x^3 - 1 = (x+1)(x+\alpha)(x+\alpha^2)$ and we display A as

$$\rho^{-1}\Big(\mathbb{F}[x]/\langle x+1 \rangle \times \mathbb{F}[x]/\langle x+\alpha \rangle \times \mathbb{F}[x]/\langle x+\alpha^2 \rangle \Big) \ .$$

Our target is to find all cyclic $(3, 1, \delta)$-convolutional codes for $\delta = 0, 1, 2, 3$. (Here we mean cyclicity with respect to any automorphism $\sigma \in Aut_{\mathbb{F}_4}(A)$, i.e. $\sigma(x) \in \{x, x^2, \alpha x, \alpha^2 x, \alpha x^2, \alpha^2 x^2\}$, see Remark 3.5.) Then we want to determine the equivalence classes both of all these codes and of those which have the same support.

We know from Corollary 5.10 that an $(n, 1, \delta)$-submodule $\mathcal{C} \subseteq \mathbb{F}[z]^n$ is a σ-cyclic convolutional code if and only if it has a basic generator matrix of the shape

$$\sum_{i=0}^{\delta} a_i z^i [1, b_{\sigma^i}^2, b_{\sigma^i}] \ , \text{ where } b_{\sigma^i} \in \{1, \alpha, \alpha^2\},\ a_i \in \mathbb{F} \text{ and } a_0 = 1,\ a_\delta \neq 0. \quad (6.2)$$

The condition $a_0 = 1$ is a normalization that makes sure that two different matrices having the shape as in (6.2) give rise to different $(3, 1, \delta)$-submodules.

If the matrix in (6.2) is basic, then the convolutional code im $_{\mathbb{F}[z]}G$ has support $\{1\}$ if $b_{\sigma^0} = 1$, it has support $\{2\}$ if $b_{\sigma^0} = \alpha^2$ and it has support $\{3\}$ if $b_{\sigma^0} = \alpha$ (cf. Corollary 5.12).

• The case $\delta = 0$: Obviously only the three matrices

$$[1, 1, 1] \ , \ [1, \alpha, \alpha^2] \text{ and } [1, \alpha^2, \alpha]$$

are of interest and of course all three matrices are basic and cyclic with respect to any automorphism $\sigma \in Aut_{\mathbb{F}}(A)$. Each of these matrices can be transformed to any other by rescaling the columns with nonzero constants. Hence we have three different cyclic $(3, 1, 0)$-convolutional codes (resp. block codes), one for each support, and they are all equivalent.

• The case $\delta = 1$: A matrix which generates a cyclic $(3, 1, 1)$-convolutional can be written as

$$[1 + az, b_{\sigma^0}^2 + ab_{\sigma^1}^2 z, b_{\sigma^0} + ab_{\sigma^1} z] \ , \text{ where } a \neq 0 \text{ and } b_{\sigma^0} \neq b_{\sigma^1} \ . \quad (6.3)$$

There are three possibilities to choose a and two possibilities to chose b_{σ^1} for fixed b_{σ^0}. Hence there are six different possibilities to write down such a matrix for each support and therefore there are 18 different matrices. We

can readily show that the three entries in a matrix as in (6.3) have pairwise different roots, therefore any two entries of such a matrix are coprime. The latter implies that the entries are given by $z+1$, $z+\alpha$, $z+\alpha^2$ (up to permutation and multiplication with nonzero constants). We conclude that there exist six different cyclic $(3,1,1)$-convolutional codes for each support, altogether 18 different cyclic $(3,1,1)$-convolutional codes, and they are all equivalent.

• The case $\delta = 2$ and $\delta = 3$: Here we used MAPLE to generate all possible matrices of the shape as in (6.2) for each support and the pairwise different ones were put into three lists, one for each support. Then the matrices in the lists were checked if they where basic. Here are the preliminary results of these calculations:

There are 42 different cyclic $(3,1,2)$-convolutional codes for each support, hence there are 126 cyclic $(3,1,2)$-convolutional codes over $\mathbb{F}_4$. There are 144 different cyclic $(3,1,3)$-convolutional codes for each support, hence there are 432 cyclic $(3,1,2)$-convolutional codes over $\mathbb{F}_4$.

We wrote a simple algorithm to check equivalence of the matrices in a given list L: The algorithm took the first entry of the list L (a representative of the first equivalence class), generated the list L' of all matrices which are equivalent to the first representative. Then each entry of L was taken and the algorithm checked, if it appeared in L'. If so, this entry was removed from L and added to a new list, the first equivalence class. After that, the process started again with the first entry of the new list L (a representative of the second equivalence class) and so forth. We got the following results:

In the case $\delta = 2$ the algorithm found seven equivalence classes for each support, each consisting of six elements. We listed all $3 \cdot 7 = 21$ representatives of these equivalence classes and checked this new list on equivalence: Again, the algorithm found seven equivalence classes and each of the new equivalence classes had exactly three members. Hence the 126 cyclic $(3,1,2)$-convolutional codes over $\mathbb{F}_4$ fall into seven equivalence classes with $3 \cdot 6 = 18$ elements each.

In the case $\delta = 3$ the algorithm found 24 equivalence classes for each support and – again – each consisted of six elements. We listed all $3 \cdot 24 = 72$ representatives of these equivalence classes and checked this new list on equivalence: The algorithm found 24 equivalence classes and, again, each of the new equivalence classes had exactly three members. Hence the 432 cyclic $(3,1,3)$-convolutional codes over $\mathbb{F}_4$ fall into 24 equivalence classes with $3 \cdot 6 = 18$ elements each.

These are our results so far. We wish to conclude with some questions about these findings:

Is it by mere coincidence that all equivalence classes for prescribed support have six elements in the cases $\delta = 1, 2, 3$?

Why do all equivalence classes have the same cardinality for fixed δ?

In the case study, one might suppose that the equivalence classes for the cyclic $(3, 1, \delta)$-convolutional codes with support $\{l\}$ determine the equivalence classes of all cyclic $(3, 1, \delta)$-convolutional codes with support $\{l'\} \neq \{l\}$. (Or loosely speaking: Maybe the knowledge of the equivalence classes for the cyclic $(3, 1, \delta)$-convolutional codes with support $\{1\}$ is "sufficient" to know everything about the equivalence classes for the supports $\{2\}$ and $\{3\}$.)
We can generalize this conjecture to cyclic codes in general: Do the equivalence classes of the set of all σ-cyclic $(n, 1, \delta)$-convolutional codes (or even (n, k, δ)-convolutional codes) with support $\{l\} \subseteq R^{(L)}$ determine all the other σ-cyclic $(n, 1, \delta)$-convolutional codes (or even (n, k, δ)-convolutional codes) with support $\{l'\} \subseteq R^{(L)} \setminus \{l\}$?

References

[Bet98] Anton Betten et. al. *Codierungstheorie: Konstruktion und Anwendung linearer Codes*. Springer-Verlag, Berlin, 1998.

[For70] G. David Forney, Jr. Convolutional codes I: Algebraic structure. *IEEE Trans. Inform. Theory*, 16:720–738, 1970. (See also corrections in *IEEE Trans. Inf. Theory*, Vol. 17, 1971, p. 360).

[For75] G. David Forney, Jr. Minimal bases of rational vector spaces, with applications to multivariable linear systems. *SIAM J. Control*, 13(3):493–520, 1975. (*SIAM J. Contr.* was passed into *SIAM J. Contr. & Opt.* after 1975).

[Gan60a] F. R. Gantmacher. *The Theory of Matrices*, volume 1. Chelsea Publishing Company, New York, 1960.

[Gan60b] F. R. Gantmacher. *The Theory of Matrices*, volume 2. Chelsea Publishing Company, New York, 1960.

[Glua] Heide Gluesing-Luerssen. Einführung in die Codierungstheorie. Unpublished manuscript. Carl von Ossietzky University Oldenburg, Germany, winter term 2000/2001.

[Glub] Heide Gluesing-Luerssen. Private communications and notes. 2002.

[GS02a] Heide Gluesing-Luerssen and Wiland Schmale. On cyclic convolutional codes. Submitted. The paper is available both as preprint (Otto-von-Guericke-University Magdeburg, Germany, Preprint Nr. 5, 2003) and at `http://front.math.ucdavis.edu/` with ID-number RA/0211040, 2002.

[GS02b] Heide Gluesing-Luerssen and Wiland Schmale. Procedures programmed with MAPLE which provide commands concerning cyclic convolutional coding and the determination of free distances. Carl von Ossietzy University Oldenburg, 2002.

[GS03] Heide Gluesing-Luerssen and Wiland Schmale. Distance bounds for convolutional codes and some optimal codes. Preprint. Available at `http://front.math.ucdavis.edu/` with ID-number RA/0305135, 2003.

[GSS02] Heide Gluesing-Luerssen, Wiland Schmale, and Melissa Striha. Some small cyclic convolutional codes. In *Electronic Proceedings of the 15th International Symposium on Mathematical Theory of Networks and Systems*, Notre Dame, IN (USA), CD-Rom, 2002. (8 pages).

[Jus73] Jørn Justesen. New convolutional code constructions and a class of asymptotically good time-varying codes. *IEEE Trans. Inform. Theory*, 19(2):220–225, 1973.

[JZ99] R. Johannesson and K. S. Zigangirov. *Fundamentals of Convolutional Coding*. IEEE Press, New York, 1999.

[Kui94] Margreet Kuijper. *First-Order Representations of Linear Systems* . Birkhäuser, Boston, 1994.

[Lan86] Serge Lang. *Introduction to Linear Algebra*. Springer-Verlag, New York, 2nd edition, 1986.

[LC83] Shu Lin and Daniel J. Costello, Jr. *Error Control Coding: Fundamentals and Applications* . Prentice-Hall, Inc., Englewood Cliffs, 2nd edition, 1983.

[LN97] Rudolf Lidl and Harald Niederreiter. *Finite Fields*. Encyclopedia of mathematics and its applications; 20. Cambridge University Press, Cambridge, 2nd edition, 1997.

[MAP01] MAPLE 7.0. Computer-Software. Waterloo Maple Inc., 2001.

[McE98] Robert J. McEliece. The algebraic theory of convolutional codes. In *Handbook of Coding Theory* , volume 1, pages 1065–1138. Elsevier, Amsterdam, 1998.

[MP79] H. F. Münzner and D. Prätzel-Wolters. Minimal bases of polynomial modules, structural indices and Brunovsky-transformations. *Int. J. Contr.*, 30:291–318, 1979.

[MS78] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error-Correcting Codes*. North-Holland, 1978.

[Pir76] Philippe Piret. Structure and constructions of cyclic convolutional codes. *IEEE Trans. Inform. Theory*, 22:147–155, 1976.

[Pir88a] Philippe Piret. *Convolutional Codes; An Algebraic Approach* . MIT Press, Cambridge, MA, 1988.

[Pir88b] Philippe Piret. A convolutional equivalent to reed-solomon codes. *Philips J. Res.*, 43:441–458, 1988.

[Rom92] Steven Roman. *Coding and Information Theory*. Springer-Verlag, New York, 1992.

[Roo79] Cornelis Roos. On the structure of convolutional and cyclic convolutional codes. *IEEE Trans. Inform. Theory*, 25:676–683, 1979.

[RS99] Joachim Rosenthal and Roxana Smarandache. Maximum distance separable convolutional codes. *Appl. Algebra Engrg. Comm. Comput.*, 10:15–32, 1999.

[RSY96] Joachim Rosenthal, J. M. Schumacher, and Eric V. York. On behaviors and convolutional codes. *IEEE Trans. Inform. Theory*, 42:1881–1891, 1996.

[Sch] Wiland Schmale. Einführung in die Algebra. Unpublished manuscript. Carl von Ossietzky University Oldenburg, Germany, summer term 2000.

[SGR98] Roxana Smarandache, Heide Gluesing-Luerssen, and Joachim Rosenthal. Generalized first order descriptions and canonical forms for convolutional codes. In A. Beghi, L. Finesso, and G. Picci, editors, *Mathematical Theory of Networks and Systems* , pages 1091–1094. Proceedings of the MTNS-98 Symposium held in Padova, Italy, 1998.

[Sha48] Claude Elwood Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27:379–423, 623–656, 1948. Available at `http://cm.bell-labs.com/cm/ms/what/shannonday/paper.html`.

[Str] Melissa Striha. Struktur und Konstruktion zyklischer Faltungscodes. Diploma thesis. Fachbereich Mathematik of the Carl von Ossietzky University Oldenburg, Germany, 2002.

Index

Hiermit versichere ich, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel und Quellen benutzt habe.

Oldenburg, den 30. Juni 2003 (Barbara Langfeld)