On the Algebraic Parameters of Convolutional Codes with Cyclic Structure

Heide Gluesing-Luerssen* and Barbara Langfeld†

October 5, 2004

Abstract

In this paper convolutional codes with cyclic structure will be investigated. These codes can be understood as left principal ideals in a suitable skew-polynomial ring. It has been shown in [4] that only certain combinations of the algebraic parameters (field size, length, dimension, and Forney indices) can occur for such cyclic codes. We will investigate whether all these combinations can indeed be realized by a suitable cyclic code and, if so, how to construct such a code. A complete characterization and construction will be given for minimal cyclic codes. It is derived from a detailed investigation of the units in the skew-polynomial ring.

Keywords: Algebraic convolutional coding theory, cyclic convolutional codes, skew-polynomial rings, Forney indices.

MSC (2000): 94B10, 94B15, 16S36

1 Introduction

The two most important classes of codes used in practice are block codes and convolutional codes. While both classes play an equally important role in engineering practice, the theory of convolutional codes is much younger and not nearly as developed as the theory of block codes. The foundation of the mathematical theory of convolutional codes was laid only in the seventies of the last century by the articles of Forney, see e. g. [1]. It led to quite some mathematical investigation in that decade among which are basically two groups of papers.

The first group [11, 7, 8] deals with the construction of convolutional codes with large distance, mainly by using cyclic block codes and resorting to the weight-retaining property for bridging the gap between cosets of polynomials in the block code case and vector polynomials in the convolutional case. These ideas were resumed later on again in [20], leading to the construction of MDS convolutional codes.

*University of Groningen, Department of Mathematics, P. O. Box 800, 9700 AV Groningen, The Netherlands; gluesing@math.rug.nl

†Kombinatorische Geometrie (M9), Zentrum Mathematik, Technische Universität München, Boltzmannstr. 3, 85747 Garching bei München, Germany; langfeld@ma.tum.de

The second group of papers [13, 14, 16] initiated a completely different approach. In the paper [14] it was investigated for the first time as to how cyclic structure has to be understood for a convolutional code itself. The first crucial fact being found was that cyclic structure in the classical sense (i. e. invariance under the cyclic shift) is not an appropriate concept for convolutional codes. Precisely, it was shown in [14] that each convolutional code that is invariant under the cyclic shift has complexity zero, hence is a block code. This insight has led Piret to a different, much more complex notion of cyclicity, which then was further generalized by Roos [16]. In the simplest form this structure can be understood as a sort of graded shift in the coefficients of the polynomial codewords. The precise notion will be given in Section 2. At this point we only want to mention that cyclic convolutional codes (CCC's, for short) of length n over the field $\mathbb{F}$ can be understood as certain left ideals in a skew-polynomial ring $A[z;\sigma]$, where $A = \mathbb{F}[x]/_{\langle x^n - 1\rangle}$, the variable z represents the delay operator, and σ determines the non-commutative structure. Both Piret and Roos gave several examples of convolutional codes that are cyclic in this new sense. They also computed (or estimated) the distances of their codes, and they turned out to be very good.

Although these papers initiated an algebraic theory of CCC's, they did not come very far and the topic came to a halt. Only recently it has been resumed in [4]. Therein an algebraic theory of CCC's, fully in terms of ideals in the skew-polynomial ring, has been established. It leads to a nice, yet nontrivial, generalization of the algebraic theory of cyclic block codes. The translation from ideals into polynomial vectors is achieved by suitable circulant matrices. In particular, CCC's are principal left ideals (thus have a generator polynomial), they are also left annihilators of right ideals (thus have a parity check polynomial), the parameters can be computed in terms of these polynomials, and the dual of a CCC is cyclic again. Moreover, in [3] plenty of examples of CCC's are given, and their distances are all optimal in the sense that they attain the Griesmer bound, see (1.3). All this indicates that the notion of cyclicity as introduced by Piret and Roos is the appropriate one for convolutional codes not only when it comes to the algebraic theory, but also for constructing good codes.

We also wish to mention a very recent approach for constructing good convolutional codes. Indeed, in [15] methods from algebraic geometry are used in order to obtain so called Goppa convolutional codes. Several examples of codes attaining the generalized Singleton bound (1.2) have been derived.

In this paper we will continue the algebraic theory of CCC's as it was set up in [4]. Consequently, we will work with the skew-polynomial ring $A[z;\sigma]$ and identify CCC's as certain left ideals therein. The aim of this paper is an existence result for CCC's with prescribed algebraic parameters (field size, length, dimension, and Forney indices). To be more precise, we first observe that, as a consequence of the results in [4], only certain combinations of the algebraic parameters can occur for CCC's; see also Theorem 2.8(3) below. Then we seek to investigate whether all these combinations do really occur. The key role for this aim is played by so called minimal CCC's; these are cyclic codes without proper cyclic subcodes. They form the building blocks of all cyclic codes in the sense that each CCC is the direct sum of minimal CCC's and the Forney indices of the code are given by the union of the Forney indices of each component. Minimal codes have a very simple ideal theoretic description in terms of their generator polynomial, see Proposition 3.3. Moreover, for these codes all Forney indices are the same, hence these codes are compact in the sense of [12, Cor. 4.3]. This makes these codes also very important from a coding point of view since compact codes are in general good candidates for having a large distance (for instance codes attaining the generalized Singleton bound are always compact, see [19]). We will show that under a certain necessary and sufficient condition any arbitrarily chosen Forney index can be realized by suitable minimal CCC's and we will show how to construct such codes. This result will then be further exploited for investigating non-minimal codes with prescribed Forney indices. As we will show in a forthcoming paper all this may serve as a preliminary step in order to construct CCC's with large distance.

The outline of the paper is as follows. The end of the introduction is devoted to the basic notions of convolutional coding theory. Thereafter in Section 2 we will introduce cyclicity for convolutional codes along with the algebraic machinery and the main results from [4] as needed for our purposes. In Section 3 we will turn to minimal CCC's. Their investigation amounts basically to a detailed study of the units in the skew polynomial ring $A[z;\sigma]$. This will lead us to the existence of minimal codes with prescribed Forney indices under a certain necessary and sufficient condition. Finally, in Section 4 we will turn to certain direct sums of minimal codes. These direct sums are specific in the sense that the generator polynomials of the minimal components are pairwise orthogonal, resulting in an easy handling of the direct sum. The existence result from Section 3 will be extended to these codes.

We will end the introduction with repeating the basic notions of convolutional coding theory. Convolutional codes are certain submodules of $\mathbb{F}[z]^n$, where $\mathbb{F}$ is a finite field. Before presenting the definition we wish to recall that each submodule $\mathcal{S}$ of $\mathbb{F}[z]^n$ is free and therefore can be written as

$$\mathcal{S} = \operatorname{im} G := \big\{uG \,\big|\, u \in \mathbb{F}[z]^k\big\}$$

where k is the rank of $\mathcal{S}$ and $G \in \mathbb{F}[z]^{k\times n}$ is a matrix containing a basis of $\mathcal{S}$. Any such matrix G is called a *generator matrix* of the module $\mathcal{S}$. It is unique up to left multiplication by a unimodular matrix, that is, for any pair of matrices $G,\, G' \in \mathbb{F}[z]^{k\times n}$ having full row rank the identity $\operatorname{im} G = \operatorname{im} G'$ is equivalent to $G' = VG$ for some matrix $V \in Gl_k(\mathbb{F}[z])$. This makes the following notions well-defined.

Definition 1.1 Let $\mathbb{F}$ be any finite field and let $G \in \mathbb{F}[z]^{k\times n}$ be a matrix of rank k.

(a) The number $\delta := \delta(G) := \max\{\deg\gamma \mid \gamma \text{ is a } k\text{-minor of } G\}$ is called the *complexity* of the submodule $\operatorname{im} G$ or of the matrix G.

(b) The submodule $\mathcal{C} := \operatorname{im} G \subseteq \mathbb{F}[z]^n$ of rank k is called a *convolutional code over* $\mathbb{F}$ *with (algebraic) parameters* (n, k, δ) if it has complexity δ and the matrix G is right invertible, i. e., if there exists some matrix $\tilde{G} \in \mathbb{F}[z]^{n\times k}$ such that $G\tilde{G} = I_k$. In this case the parameter n is called the *length* of the code.

Notice that the algebraic parameters $(n,\, k,\, \delta)$ do not contain any information about the error-correcting properties of the code. The complexity is also known as the *overall constraint length* [6, p. 55], [1, p. 721] or the *degree* [12, Def. 3.5] of the code. It is an important parameter describing the size of the code and of the encoding process. In the coding literature a right invertible matrix is often called *basic* [1, p. 730] or *delay-free and non-catastrophic*, see [12, p.1102]. Often in coding literature convolutional codes are defined as subspaces of the vector space $\mathbb{F}((z))^n$ of vector valued Laurent series over $\mathbb{F}$, see for instance [12] and [1]. However, as long as one restricts to right invertible generator matrices it makes no difference with respect to code properties and code constructions whether one works in the context of infinite message and codeword sequences (Laurent series) or finite ones (polynomials). Only for decoding it becomes important whether or not one may assume the sent codeword to be finite. The issue whether convolutional coding theory should be based on finite or infinite message sequences has first been raised and discussed in detail in [18, 17].

Since every right invertible matrix $G \in \mathbb{F}[z]^{k\times n}$ can be completed to a unimodular matrix (e.g. by using the Smith normal form), one has the following properties.

Remark 1.2 (a) The convolutional codes over $\mathbb{F}$ of length n are the direct summands of the module $\mathbb{F}[z]^n$.

(b) Each convolutional code $\mathcal{C} \subseteq \mathbb{F}[z]^n$ has a parity check matrix, that is, there exists a matrix $H \in \mathbb{F}[z]^{n\times(n-\mathrm{rk}\,\mathcal{C})}$ such that $\mathcal{C} = \ker H := \{v \in \mathbb{F}[z]^n \mid vH = 0\}$.

Part (b) can be considered as one of the main reasons for restricting to direct summands rather than arbitrary submodules for convolutional codes. A parity check matrix is an important tool for data transmission since it is helpful for checking whether or not the received data are erroneous.

The following property of convolutional codes will be needed later on.

Lemma 1.3 *Let $\mathcal{C}$, $\hat{\mathcal{C}} \subseteq \mathbb{F}[z]^n$ be two submodules having the same rank and satisfying $\hat{\mathcal{C}} \subseteq \mathcal{C}$. Furthermore, let $\hat{\mathcal{C}}$ be a convolutional code. Then $\hat{\mathcal{C}} = \mathcal{C}$.*

Proof: Let $\mathcal{C} = \mathrm{im}\, G$ and $\hat{\mathcal{C}} = \mathrm{im}\, \hat{G}$ where $G, \hat{G} \in \mathbb{F}[z]^{k\times n}$ and $\hat{G}$ is right invertible. The assumption $\hat{\mathcal{C}} \subseteq \mathcal{C}$ implies the existence of some matrix $U \in \mathbb{F}[z]^{k\times k}$ such that $\hat{G} = UG$. Using a right inverse of $\hat{G}$ shows $U \in Gl_k(\mathbb{F}[z])$ and the assertion follows. □

It is well-known that each submodule of $\mathbb{F}[z]^n$ has a minimal generator matrix in the sense of the next definition [1, Thm. 5] or [2, p. 495]. In the same paper [2, Sec. 4] it has been shown how to derive such a matrix from a given generator matrix in a constructive way.

Definition 1.4 (1) For $v = \sum_{j=0}^{N} v_j z^j \in \mathbb{F}[z]^n$ where $v_j \in \mathbb{F}^n$ and $v_N \neq 0$ let $\deg v =: N$ be the *degree* of v. Moreover, put $\deg 0 = -\infty$.

(2) Let $G \in \mathbb{F}[z]^{k\times n}$ be a matrix with rank k and complexity δ and let $\nu_1, \ldots, \nu_k$ be the degrees of the rows of G. We say that G is *minimal* if $\delta = \sum_{i=1}^{k} \nu_i$. In this case the row degrees of G are uniquely determined by the submodule $\mathcal{S} := \mathrm{im}\, G$. They are called the *Forney indices* of $\mathcal{S}$.

The notion “minimal” stems from the fact that for an arbitrary generator matrix G one has $\delta \leq \sum_{i=1}^{k} \nu_i$. Thus, in a minimal generator matrix the rows degrees have been reduced to their minimal values.

From the above it follows that a convolutional code with parameters (n, k, δ) has a constant generator matrix if and only if $\delta = 0$. In that case the code can be regarded as an (n, k)-block code.

The most important concept for a code is its distance. It measures the error-correcting capability, hence the quality, of the code. The definition of the distance of a convolutional code is straightforward. For a constant vector $w = (w_1, \ldots, w_n) \in \mathbb{F}^n$ we define, just like in block code theory, its *(Hamming) weight* as $\mathrm{wt}(w) = \#\{i \mid w_i \neq 0\}$. For a polynomial vector

$v=\sum_{j=0}^{N} v_j z^j \in \mathbb{F}[z]^n$, where $v_j \in \mathbb{F}^n$, the *weight* is defined as $\mathrm{wt}(v)=\sum_{j=0}^{N}\mathrm{wt}(v_j)$. Then the *(free) distance* of a code $\mathcal{C}\subseteq\mathbb{F}[z]^n$ with generator matrix $G\in\mathbb{F}[z]^{k\times n}$ is given as

$$\mathrm{dist}(\mathcal{C}) := \min\{\mathrm{wt}(v) \mid v\in\mathcal{C},\ v\neq 0\} = \min\big\{\mathrm{wt}(uG)\,\big|\, u\in\mathbb{F}[z]^k,\ u\neq 0\big\}. \tag{1.1}$$

In coding theoretic terms this notion is based on counting only the number of errors during data transmission, but not their magnitude; for more details about the distance of convolutional codes see for instance [6, Sec. 3.1]. Although we will not present any theoretical results concerning the distance of a CCC we will show several examples of codes which do have optimal distance. In all these cases the distances have been computed with routines written in Maple and then compared to some suitable bound known from the literature. One of these bounds is the *generalized Singleton bound* [19] stating that the distance d of a code with parameters (n,k,δ) over any field satisfies

$$d\leq S(n,k,\delta) := (n-k)\Big(\Big\lfloor\frac{\delta}{k}\Big\rfloor+1\Big)+\delta+1. \tag{1.2}$$

A code $\mathcal{C}$ with $\mathrm{dist}(\mathcal{C})=S(n,k,\delta)$ is called an MDS code; see [19]. The *Griesmer bound* also takes the field size into account. It states that each code over a field with q elements and with parameters (n,k,δ) and largest Forney index m has distance d bounded by

$$d\leq \max\Big\{d'\in\{1,\ldots,S(n,k,\delta)\}\,\Big|\,\sum_{l=0}^{k(m+i)-\delta-1}\Big\lceil\frac{d'}{q^l}\Big\rceil\leq n(m+i)\text{ for all } i\in\hat{\mathbb{N}}\Big\}, \tag{1.3}$$

see [6, 3.22] for $q=2$ and [3, Thm. 3.4] for general field size. At the end of Sections 3 and 4 we will present several codes where the distance attains this maximum value.

2 The Piret algebra and the Notion of Cyclicity

In this section we will introduce the notion of cyclicity for convolutional codes. This will require quite some algebraic machinery, but we will restrict ourselves to introducing the notions that are absolutely necessary for the rest of the paper. First we will recall from [14] that the classical notion of invariance under cyclic shift will always lead to complexity zero, hence is much to restrictive for convolutional codes. We will then introduce a more general notion of cyclicity, taken from [14, 16, 4]. In order to do so, we define the skew-polynomial ring $A[z;\sigma]$. It is isomorphic to $\mathbb{F}[z]^n$ as left $\mathbb{F}[z]$-module, and we will declare a code in $\mathbb{F}[z]^n$ cyclic if it corresponds to a left ideal in $A[z;\sigma]$. We will briefly discuss some features of $A[z;\sigma]$ and summarize the main results about cyclic codes, as obtained in [4], in Theorem 2.8. Moreover, we will introduce the notion of unmixed polynomials in $A[z;\sigma]$.

Just like for cyclic block codes we assume from now on that the length n and the field size $|\mathbb{F}|$ are coprime. Recall that a block code $\mathcal{C}\subseteq\mathbb{F}^n$ is called cyclic if it is invariant under the cyclic shift, i. e.,

$$(v_0,\ldots,v_{n-1})\in\mathcal{C}\Longrightarrow(v_{n-1},v_0,\ldots,v_{n-2})\in\mathcal{C} \tag{2.1}$$

for all $(v_0,\ldots,v_{n-1})\in\mathbb{F}^n$. It is well-known that this is the case if and only if $\mathcal{C}$ is an ideal in the quotient ring

$$A:=\mathbb{F}[x]/_{\langle x^n-1\rangle}=\Big\{\sum_{i=0}^{n-1}f_ix^i \bmod (x^n-1)\,\Big|\, f_0,\ldots,f_{n-1}\in\mathbb{F}\Big\}, \tag{2.2}$$

which we canonically identify with $\mathbb{F}^n$ via the mapping

$$\mathfrak{p}:\mathbb{F}^n\longrightarrow A,\quad (v_0,\ldots,v_{n-1})\longmapsto\sum_{i=0}^{n-1}v_ix^i \bmod (x^n-1).$$

Recall that the cyclic shift in $\mathbb{F}^n$ translates into multiplication by x in A, i. e.,

$$\mathfrak{p}(v_{n-1},v_0,\ldots,v_{n-2})=x\mathfrak{p}(v_0,\ldots,v_{n-1}) \tag{2.3}$$

for all $(v_0,\ldots,v_{n-1})\in\mathbb{F}^n$.

In order to extend the situation of cyclic block codes to the convolutional setting, we have to replace the vector space $\mathbb{F}^n$ by the free module $\mathbb{F}[z]^n=\{\sum_{j=0}^N z^jv_j\mid N\in\mathbb{N}_0,\,v_j\in\mathbb{F}^n\}$ and, consequently, the ring A by the polynomial ring $A[z]=\{\sum_{j=0}^N z^ja_j\mid N\in\mathbb{N}_0,\,a_j\in A\}$. Then we can extend the mapping $\mathfrak{p}$ above coefficient-wise to these polynomials. Precisely, $\mathfrak{p}\big(\sum_{j=0}^N z^jv_j\big)=\sum_{j=0}^N z^j\mathfrak{p}(v_j)$, where, of course, $v_j\in\mathbb{F}^n$ and thus $\mathfrak{p}(v_j)\in A$ for all j. This mapping is an isomorphism of $\mathbb{F}[z]$-modules. Again by construction, the cyclic shift in $\mathbb{F}[z]^n$ corresponds to multiplication by x in $A[z]$, that is, we have (2.3) for all $(v_0,\ldots,v_{n-1})\in\mathbb{F}[z]^n$. At this point it sounds quite natural to call a convolutional code $\mathcal{C}\subseteq\mathbb{F}[z]^n$ cyclic if it is invariant under the cyclic shift, i. e., if (2.1) holds true for all $(v_0,\ldots,v_{n-1})\in\mathbb{F}[z]^n$. This, however, does not result in any codes other than block codes. Indeed, in [14, Thm. 3.12] and [16, Thm. 6] it has been shown that any convolutional code $\mathcal{C}\subseteq\mathbb{F}[z]^n$ satisfying (2.1) for all $(v_0,\ldots,v_{n-1})\in\mathbb{F}[z]^n$ has complexity zero, thus is a block code. For an elementary proof see also [4, Prop. 2.7].

This result has led Piret [14] to suggesting a different notion of cyclicity for convolutional codes. We will present this notion in the slightly more general version introduced by Roos [16]. In order to do so, notice that $\mathbb{F}$ can be regarded in a natural way as a subfield of the ring A. As a consequence, A is an $\mathbb{F}$-algebra. In the sequel the automorphism group $\mathrm{Aut}_{\mathbb{F}}(A)$ of the $\mathbb{F}$-algebra A will play an important role. Details of how to determine this group can be found in [4, Sec. 3]

The main idea of Piret and Roos was to impose a new ring structure on $A[z]$ and to declare a code cyclic if it is a left ideal with respect to that ring structure. The new structure is non-commutative and based on an (arbitrarily chosen) automorphism on A. In detail this looks as follows.

Definition 2.1 Let $\sigma\in\mathrm{Aut}_{\mathbb{F}}(A)$.

(1) On the set $A[z]$ we define addition as usual while multiplication is defined via the rule

$$az=z\sigma(a)\text{ for all }a\in A \tag{2.4}$$

along with classical multiplication for the coefficients in the quotient ring A as well as associativity and distributivity. This turns $A[z]$ into a skew-polynomial ring, denoted by $A[z;\sigma]$. We call $A[z;\sigma]$ a *Piret algebra.*

(2) Consider the mapping

$$\mathfrak{p}:\mathbb{F}[z]^n \to A[z;\sigma],\quad \sum_{j=0}^{N} z^j(v_{j,0},\ldots,v_{j,n-1}) \longrightarrow \sum_{j=0}^{N} z^j \sum_{i=0}^{n-1} v_{j,i}x^i.$$

A submodule $\mathcal{S}\subseteq\mathbb{F}[z]^n$ is said to be *σ-cyclic* if $\mathfrak{p}(\mathcal{S})$ is a left ideal in $A[z;\sigma]$. A convolutional code $\mathcal{C}\subseteq\mathbb{F}[z]^n$ is said to be *σ-cyclic* (or a σ-CCC) if $\mathcal{C}$ is a direct summand of $\mathbb{F}[z]^n$ and a σ-cyclic submodule.

Notice the following two facts. Firstly, cyclic block codes (in the classical sense of (2.1)) are σ-cyclic for all automorphisms σ. Secondly, unless σ is the identity, the indeterminate z does not commute with its coefficients. Consequently, it becomes important to distinguish between left and right coefficients of z. Of course, the coefficients can be moved to either side since σ is invertible. In the sequel we will always use the representation via right coefficients since that is the one needed for the mapping $\mathfrak{p}$ in part (2) above. Since multiplication inside A remains the same as before, A is a commutative subring of $A[z;\sigma]$. Moreover, since $\sigma|_{\mathbb{F}}=\mathrm{id}_{\mathbb{F}}$, the classical polynomial ring $\mathbb{F}[z]$ is a commutative subring of $A[z;\sigma]$, too. As a consequence, $A[z;\sigma]$ is a left and right $\mathbb{F}[z]$-module. One can show that the mapping $\mathfrak{p}:\mathbb{F}[z]^n\to A[z;\sigma]$ is an isomorphism of left $\mathbb{F}[z]$-modules (but not of right $\mathbb{F}[z]$-modules). We will denote the inverse as

$$\mathfrak{p}^{-1}=\mathfrak{v}. \tag{2.5}$$

In the special case where $\sigma=\mathrm{id}_A$, the ring $A[z;\sigma]$ is the classical commutative polynomial ring and we know from the result mentioned above that no σ-cyclic convolutional codes with nonzero complexity exist. In [4, Prop 3.4] all automorphisms σ allowing no σ-CCC with positive complexity have been characterizied. We will come to this point in detail later on. The reader might also have noticed that both module structures on $A[z;\sigma]$, over the ring $A[z;\sigma]$ as well as over $\mathbb{F}[z]$, are used in the definition of a σ-CCC. As one might expect, this will turn out to be expressible in simpler terms (cf. Remark 2.10 at the end of this secion).

Example 2.2 Let us consider the case where $\mathbb{F}=\mathbb{F}_2$ and $n=7$. Thus $A=\mathbb{F}_2[x]/_{\langle x^7-1\rangle}$. In this case $\mathrm{Aut}_{\mathbb{F}}(A)$ contains 18 automorphisms (see also [16, p. 680, Table II]), one of which is defined via $\sigma(x)=x^5$. We choose this automorphism for the following computations. Consider the polynomial

$$g:=1+x^2+x^3+x^4+z(x+x^2+x^3+x^5)+z^2(1+x+x^4+x^6)\in A[z;\sigma] \tag{2.6}$$

and denote by ${}^{\bullet}\langle\, g\,\rangle:=\{fg\mid f\in A[z;\sigma]\}$ the left ideal generated by g in $A[z;\sigma]$. Moreover, put $\mathcal{C}:=\mathfrak{v}({}^{\bullet}\langle\, g\,\rangle)\subseteq\mathbb{F}[z]^7$. We will show now that $\mathcal{C}$ is a direct summand of $\mathbb{F}[z]^7$, hence $\mathcal{C}$ is a σ-cyclic convolutional code. In order to do so we first notice

$${}^{\bullet}\langle\, g\,\rangle=\mathrm{span}_{\mathbb{F}[z]}\{g,\,xg,\ldots,x^6g\}$$

and therefore, using the isomorphism $\mathfrak{v}$ from (2.5),

$$\mathcal{C}=\big\{uM\,\big|\,u\in\mathbb{F}[z]^7\big\}\text{ where }M=\begin{bmatrix}\mathfrak{v}(g)\\ \mathfrak{v}(xg)\\ \vdots\\ \mathfrak{v}(x^6g)\end{bmatrix}\in\mathbb{F}[z]^{7\times7}.$$

Thus we have to compute $x^i g$ for $i = 1, \ldots, 6$. Using the multiplication rule in (2.4) we obtain

$$\begin{aligned}
xg &= x + x^3 + x^4 + x^5 + z(1 + x + x^3 + x^6) + z^2(x + x^3 + x^4 + x^5),\\
x^2 g &= x^2 + x^4 + x^5 + x^6 + z(x + x^4 + x^5 + x^6) + z^2(1 + x + x^2 + x^5),\\
x^3 g &= 1 + x^3 + x^5 + x^6 + z(x^2 + x^3 + x^4 + x^6) + z^2(1 + x^3 + x^5 + x^6) = g + x^2 g.
\end{aligned}$$

Since $x^3 g$ is in the $\mathbb{F}$-span of the previous elements, we obtain ${}^{\bullet}\!\langle\, g\,\rangle = \operatorname{span}_{\mathbb{F}[z]}\big\{g, xg, x^2 g\big\}$ and, since $\mathfrak{v}$ is an isomorphism, $\mathcal{C} = \big\{uG \,\big|\, u \in \mathbb{F}[z]^3\big\}$, where

$$G = \begin{bmatrix} \mathfrak{v}(g) \\ \mathfrak{v}(xg) \\ \mathfrak{v}(x^2 g) \end{bmatrix} = \begin{bmatrix} 1+z^2 & z+z^2 & 1+z & 1+z & 1+z^2 & z & z^2 \\ z & 1+z+z^2 & 0 & 1+z+z^2 & 1+z^2 & 1+z^2 & z \\ z^2 & z+z^2 & 1+z^2 & 0 & 1+z & 1+z+z^2 & 1+z \end{bmatrix}.$$

One can easily check that the matrix G is right invertible and minimal (see Definition 1.4). Hence $\mathcal{C} \subseteq \mathbb{F}[z]^7$ is indeed a CCC. It is worth mentioning that $\operatorname{dist}(\mathcal{C}) = 12$ (derived via Maple routines), and this is the optimum value for any convolutional code over $\mathbb{F}_2$ with parameters $(7, 3, 6)$ by virtue of the Griesmer bound (1.3).

For our investigations in the next sections we will not only need the main results on CCC's as derived in [4], but also part of the machinery. In the rest of this section we will introduce the concepts and results that are absolutely necessary for the subsequent sections.

The main tool for describing the left ideals in $A[z;\sigma]$ is the fact that A is a direct product of fields. Since we need the details of this fact, we will first elaborate on this. By coprimeness of the length n and the field size $|\mathbb{F}|$, the polynomial $x^n - 1$ is square free, say

$$x^n - 1 = \pi_1 \cdot \ldots \cdot \pi_r, \tag{2.7}$$

where $\pi_1, \ldots, \pi_r \in \mathbb{F}[x]$ are irreducible, monic, and pairwise different. We will also assume that the polynomials are ordered according to

$$\deg_x \pi_1 \leq \ldots \leq \deg_x \pi_r.$$

The Chinese Remainder Theorem provides us with an isomorphism of rings

$$\psi : A \longrightarrow K_1 \times \ldots \times K_r, \quad a \longmapsto \big(a \bmod \pi_1, \ldots, a \bmod \pi_r\big) \tag{2.8}$$

where $K_k = \mathbb{F}[x]/_{\langle \pi_k \rangle}$. Notice that $K_k \cong K_l$ if and only if $\deg_x \pi_k = \deg_x \pi_l$. Even though we will not present elements in A explicitly in the form $\big(a \bmod \pi_1, \ldots, a \bmod \pi_r\big)$ it is quite helpful for computations to have this representation, along with componentwise operations, in mind. The elements

$$\varepsilon^{(k)} := \psi^{-1}\big(0, \ldots, 0, 1, 0, \ldots, 0\big) \text{ for } k = 1, \ldots, r$$

(where 0 and 1 have to be understood as the elements $0 \bmod \pi_l$ and $1 \bmod \pi_l$ in K_l) are particularly important since they form the uniquely determined set of primitive idempotents in A. Furthermore, the idempotents are pairwise orthogonal, thus $\varepsilon^{(k)}\varepsilon^{(l)} = 0$ for $k \neq l$. Observe that for any $a \in A$ the products $\varepsilon^{(l)}a$ single out the various components of a. Precisely, $\psi(\varepsilon^{(l)}a) = (0, \ldots, 0, a \bmod \pi_l, 0, \ldots, 0)$ for any $l = 1, \ldots, r$. Hence $\varepsilon^{(1)}a + \ldots + \varepsilon^{(r)}a$

might serve as a decomposition of $a \in A$ just like the one in (2.8). In the sequel we will use this representation rather than the one from (2.8). Moreover we have

$$a \in A \text{ is a unit in } A \Longleftrightarrow \varepsilon^{(l)}a \neq 0 \text{ for all } l = 1, \ldots, r. \tag{2.9}$$

Let us now study the effect of a given automorphism $\sigma \in \mathrm{Aut}_{\mathbb{F}}(A)$ on the components. It is straightforward to see that

$$\sigma(\varepsilon^{(k)}) = \varepsilon^{(l)} \text{ for some } l \text{ such that } \deg_x \pi_k = \deg_x \pi_l. \tag{2.10}$$

Thus σ induces a permutation on the primitive idempotents. This gives rise to the following definition. We will use the notation S_r for the symmetric group on r symbols.

Definition 2.3 Let $\sigma \in \mathrm{Aut}_{\mathbb{F}}(A)$. Define the permutation $\Pi_\sigma \in S_r$ via $\Pi_\sigma(k) = l$, where l is such that $\sigma(\varepsilon^{(k)}) = \varepsilon^{(l)}$ for all $k = 1 \ldots, r$. We call Π_σ the permutation induced by σ. Furthermore, define the equivalence relation $\equiv_\sigma$ on the index set $\{1, \ldots, r\}$ via $k \equiv_\sigma l$ if there exists some $i \in \mathbb{N}_0$ such that $\sigma^i(\varepsilon^{(k)}) = \varepsilon^{(l)}$.

Of course, the permutation Π_σ simply reflects the permutation induced by σ on the set $\{\varepsilon^{(1)}, \ldots, \varepsilon^{(r)}\}$, that is, $\sigma(\varepsilon^{(k)}) = \varepsilon^{(\Pi_\sigma(k))}$. It is worth noticing that in general not the whole permutation group S_r can be realized by induced permutations. This can be seen immediately from (2.10). Notice also, that by definition $k \equiv_\sigma l$ if and only if k and l belong to the same cycle of the permutation Π_σ. Thus, by (2.10), $k \equiv_\sigma l$ implies $\deg_x \pi_k = \deg_x \pi_l$ for all $k, l \in \{1, \ldots, r\}$. Moreover, the automorphisms in $\mathrm{Aut}_{\mathbb{F}}(A)$ are in general not uniquely determined by their induced permutation, see also [4, Sec. 3].

Having this description of the ring A and its automorphisms available we will now fix some $\sigma \in \mathrm{Aut}_{\mathbb{F}}(A)$ and turn to the Piret algebra $A[z;\sigma]$. We will give some basic properties. Details can be found in [4].

Using $1 = \varepsilon^{(1)} + \ldots + \varepsilon^{(r)}$ we can write each polynomial $f \in A[z;\sigma]$ in the form

$$f = f^{(1)} + \ldots + f^{(r)}, \text{ where } f^{(k)} := \varepsilon^{(k)} f.$$

We call $f^{(k)}$ the *k-th component* of f. Furthermore, the set $T_f := \{k \in \{1, \ldots, r\} \mid f^{(k)} \neq 0\}$ is called the *support* of f. Each $f \in A[z;\sigma]$ can be written as an A-linear combination of the elements

$$z^\mu \varepsilon^{(l)}, \ \mu \geq 0, \ l = 1, \ldots, r. \tag{2.11}$$

We call these elements the *monomials* of $A[z;\sigma]$. From (2.4) and Definition 2.3 it is clear that the monomials of the k-th component $f^{(k)}$ of f are not in $\varepsilon^{(k)}A$ but rather move around the fields $\varepsilon^{(l)}A$ where $l \equiv_\sigma k$.

Example 2.4 Consider again the situation of Example 2.2 where $\mathbb{F} = \mathbb{F}_2$, $n = 7$ and $\sigma(x) = x^5$. The polynomial $x^7 - 1$ decomposes into $x^7 - 1 = \pi_1\pi_2\pi_3$ where $\pi_1 = x+1$, $\pi_2 = x^3+x+1$, and $\pi_3 = x^3 + x^2 + 1$. Furthermore, one has the primitive idempotents

$$\varepsilon^{(1)} = 1 + x + x^2 + x^3 + x^4 + x^5 + x^6, \ \varepsilon^{(2)} = 1 + x + x^2 + x^4, \ \varepsilon^{(3)} = 1 + x^3 + x^5 + x^6,$$

which can easily be checked by verifying $(\varepsilon^{(k)} \bmod \pi_i) = \delta_{ik}$ for $i, k = 1, 2, 3$. Moreover, $\sigma(\varepsilon^{(1)}) = \varepsilon^{(1)}$, $\sigma(\varepsilon^{(2)}) = \varepsilon^{(3)}$, $\sigma(\varepsilon^{(3)}) = \varepsilon^{(2)}$. In other words, σ induces the permutation

$\Pi_\sigma = (1)(2,3)$. It can be shown straightforwardly that the polynomial g given in (2.6) satisfies $\varepsilon^{(1)}g = \varepsilon^{(2)}g = 0$ and $\varepsilon^{(3)}g = g$, thus $g = g^{(3)}$. Furthermore, $g = \varepsilon^{(3)}(1 + x + x^2) + z\varepsilon^{(2)}x + z^2\varepsilon^{(3)}x$, showing how the coefficients switch between $\varepsilon^{(3)}A$ and $\varepsilon^{(2)}A$.

At this point we want to introduce the following notions which will come in very handy later on.

Definition 2.5 Let $g \in A[z;\sigma]$.

(a) g is called a *component polynomial* if $g = g^{(k)}$ for some $k = 1, \ldots, r$, i. e., if $|T_g| \leq 1$.

(b) g is called *unmixed* if $k \not\equiv_\sigma l$ for all $k, l \in T_g$ where $k \neq l$.

A component polynomial is always unmixed. As for the notion of unmixedness observe that each component $g^{(k)}$ of an arbitrarily given polynomial $g \in A[z;\sigma]$ satisfies

$$g^{(k)} \in \operatorname{span}_A\{z^\mu \varepsilon^{(k')} \mid \mu \geq 0,\, k' \equiv_\sigma k\}.$$

Therefore, in an unmixed polynomial g no nonzero terms of different components are left A-multiples of the same monomial $z^\mu\varepsilon^{(l)}$. Roughly speaking, the components do not overlap.

Example 2.6 With the data as in Example 2.4 the polynomial $f := \varepsilon^{(1)} + \varepsilon^{(2)} + z\varepsilon^{(3)}$ is unmixed since $\varepsilon^{(1)}f = \varepsilon^{(1)}$, $\varepsilon^{(2)}f = \varepsilon^{(2)} + z\varepsilon^{(3)}$, and $\varepsilon^{(3)}f = 0$. Thus $T_f = \{1, 2\}$ and we have $1 \not\equiv_\sigma 2$. The polynomial $f' := \varepsilon^{(2)} + z\varepsilon^{(2)}$ is not unmixed since $T_{f'} = \{2, 3\}$ and $2 \equiv_\sigma 3$.

Unmixed polynomials form a very important special case of so called reduced polynomials as they have been defined in [4, Def. 4.9(b)]. In that paper a Gröbner-type theory has been established for the Piret algebra $A[z;\sigma]$. It is based on the monomials given in (2.11) and leads to a reduction algorithm and unique reduced generating sets for left ideals just like for commutative polynomials in several variables. Since we will not need the notion of reducedness directly, we do not want to repeat the (straightforward) definition here but rather want the reader to have the following in mind. It is a direct consequence of the given definitions.

Remark 2.7 Unmixed polynomials as well as constant polynomials are reduced in the sense of [4, Def. 4.9(b)].

In the sequel we will have to make use of the following results from [4].

Theorem 2.8 *Fix $\sigma \in \operatorname{Aut}_{\mathbb{F}}(A)$.*

(1) Let $\mathcal{C} \subseteq \mathbb{F}[z]^n$ be a σ-CCC. Then there exists a reduced polynomial $g \in A[z;\sigma]$ such that

$$\mathfrak{p}(\mathcal{C}) = {}^{\bullet}\!\langle\, g \,\rangle := \{fg \mid f \in A[z;\sigma]\}.$$

In particular, the left ideal $\mathfrak{p}(\mathcal{C})$ is principal. Moreover, the polynomial g is unique up to left multiplication by units in A, and the support of g satisfies $T_g = T_{g_0}$, where g_0 denotes the constant part of g.

(2) Let $g \in A[z;\sigma]$ be a reduced polynomial. Then $\mathfrak{v}(\overset{\bullet}{\langle}\, g\,\rangle) \subseteq \mathbb{F}[z]^n$ is a direct summand of $\mathbb{F}[z]^n$ (thus a σ-cyclic convolutional code) if and only if

$$g = \sum_{l \in T_g} u^{(l)} \text{ for some unit } u \in A[z;\sigma] \tag{2.12}$$

That is, the reduced generator polynomials of σ-CCC's consist of some of the components of suitable units.

(3) Let $g \in A[z;\sigma]$ be a reduced polynomial with support T_g and let $\mathcal{S} = \mathfrak{v}(\overset{\bullet}{\langle}\, g\,\rangle) \subseteq \mathbb{F}[z]^n$. For $l \in T_g$ let $\deg_x \pi_l = \kappa_l$, where π_l is as in (2.7), and put $\kappa := \sum_{l \in T_g} \kappa_l$. Then the matrix

$$G := \big[\mathfrak{v}\big(x^i g^{(l)}\big)\big]_{l \in T_g,\, i=0,\ldots,\kappa_l - 1} \in \mathbb{F}[z]^{\kappa \times n} \tag{2.13}$$

is a minimal generator matrix of the submodule $\mathcal{S}$. As a consequence, $\mathcal{S}$ is a submodule of rank κ and complexity $\delta := \sum_{l \in T_g} \kappa_l \deg_z g^{(l)}$. The Forney indices are given by the numbers $\deg_z g^{(l)}$, $l \in T_g$, each one counted κ_l times.

Proof: Part (1) is [4, Thm. 4.5, Cor. 4.13(b)] while part (3) is in Thm. 7.13 of the same paper. As for part (2) we obtain from [4, Prop .7.10] that $\mathfrak{v}(\overset{\bullet}{\langle}\, g\,\rangle)$ is a direct summand if and only if $gv = g_0$ for some unit $v \in A[z;\sigma]$. Now put $u = \big(g_0 + \sum_{l \notin T_{g_0}} \varepsilon^{(l)}\big)v^{-1}$. Then u is a unit since the first factor is a unit due to (2.9). It is easy to see that g satisfies (2.12). □

It is worth mentioning that $A[z;\sigma]$ is not a left principal ideal ring. Part (1) above only states that left ideals associated to direct summands in $\mathbb{F}[z]^n$ are principal. Indeed, there exist left ideals that are not principal [4, Exa. 4.6(a)]. Moreover, we want to emphasize that, according to (3), the algebraic parameters of σ-cyclic convolutional codes can occur only in certain combinations. In particular, the Forney indices appear, in general, with higher multiplicities, depending on the degrees of the prime factors π_l. In the next sections we will investigate this situation in more detail.

We illustrate the results above by continuing the examples given earlier.

Example 2.9 Let us return once more to Example 2.2 and its continuation in Example 2.4. In that case the polynomial $g = g^{(3)}$ is a component polynomial and thus reduced by Remark 2.7. As shown explicitly in Example 2.2 it generates a left ideal corresponding to a code of rank 3 and complexity 6. This is compliant with what has been stated in Theorem 2.8(3). Furthermore, one can show that the polynomial $u = 1+x+x^2+z(1+x+x^2+x^6)+z^2(1+x+x^4+x^6)$ is a unit in $A[z;\sigma]$ with inverse $u^{-1} = 1+x^2+x^3+x^6+z(x+x^2)+z^2(1+x^2+x^5+x^6)$, and that $g = u^{(3)}$, illustrating Theorem 2.8(2). We do not discuss how to obtain the unit u from the given reduced polynomial g since that needs more detailed results from [4].

We close this section with an additional comment concerning the two different module structures used in Definition 2.1(2) of σ-CCC's. As it turns out from the previous results that definition can be expressed in simpler terms. Since this will not be needed throughout the paper, we will only sketch the reformulation in the following remark.

Remark 2.10 Let I be a left ideal in $A[z;\sigma]$. Then I is a direct summand of the left $\mathbb{F}[z]$-module $A[z;\sigma]$ if and only if I is a direct summand of the ring $A[z;\sigma]$. Hence, the set of

σ-CCC's is the same as the set of direct summands of the ring $A[z;\sigma]$. In particular, a σ-CCC has a direct complement which is σ-cyclic, too. All this can be derived by using Theorem 2.8(2). Indeed, one can easily show that a direct complement of ${}^{\bullet}\!\langle\, g\,\rangle$, where g is as in (2.12), is given by the left ideal generated by $g' := \sum_{l\not\in T_g} u^{(l)}$. In this context it is also worth recalling that in every ring R with 1 a left ideal $\mathcal{I}$ that is a direct summand is left principal and even has an idempotent generator. We wish to emphasize that reduced generator polynomials, as guaranteed by Theorem 2.8(1), are in general not idempotent. But the above shows how idempotent generators can easily be obtained from the reduced generator. Indeed, we have that $g + g' = u$ is a unit in $A[z;\sigma]$. Thus $1 = u^{-1}g + u^{-1}g'$ and $u^{-1}gu^{-1}g' = u^{-1}g - u^{-1}gu^{-1}g \in {}^{\bullet}\!\langle\, g\,\rangle \cap {}^{\bullet}\!\langle\, g'\,\rangle = \{0\}$. From this it follows that both terms $u^{-1}g$ and $u^{-1}g'$ are idempotent generators of the respective left ideal. In general, these idempotent generators have much higher degree than the reduced ones. At any rate, as Theorem 2.8(3) shows, the reduced generators are the more useful ones when it comes to the associated module in $\mathbb{F}[z]^n$.

3 Minimal Cyclic Codes

As before let $\mathbb{F}$ be a finite field such that n and $|\mathbb{F}|$ are coprime and let $\sigma \in \mathrm{Aut}_{\mathbb{F}}(A)$ be a fixed automorphism, where A is as in (2.2). In this section we will investigate the building blocks of σ-CCC's, i. e., the minimal CCC's. We will derive necessary and sufficient conditions for the automorphism σ to allow for σ-cyclic codes with arbitrarily prescribed Forney indices.

As we saw in Theorem 2.8(1) each σ-cyclic convolutional code $\mathcal{C} \subseteq \mathbb{F}[z]^n$ corresponds to a principal left ideal in $A[z;\sigma]$ which is generated by a reduced polynomial. Moreover, since according to the same result the reduced polynomial is unique up to constant units, the following definition is well-posed.

Definition 3.1 Let $g \in A[z;\sigma]$ be a reduced polynomial. Then its support T_g is called the *support of the left ideal* ${}^{\bullet}\!\langle\, g\,\rangle$ and also the *support of the submodule* $\mathfrak{v}({}^{\bullet}\!\langle\, g\,\rangle)$.

Furthermore, part (3) of Theorem 2.8 shows that each σ-cyclic convolutional code can be presented as the direct sum of σ-cyclic codes with component polynomials as generator polynomials. Indeed, we consider first the case where $g = g^{(l)}$ for some l. Then, using the theorem and applying the isomorphism $\mathfrak{p}$ to the identity $S = \mathrm{im}\, G$ we derive ${}^{\bullet}\!\langle\, g\,\rangle = \mathfrak{p}\big(\mathrm{im}\,[\mathfrak{v}(x^i g)]_{i=0,\ldots,\kappa_l}\big)$. From this and the full rank of the matrix G in (2.13) it follows that in the general case, where $g = \sum_{l\in T_g} g^{(l)}$ is reduced, Equation (2.13) translates into the direct sum

$${}^{\bullet}\!\langle\, g\,\rangle = \bigoplus_{l\in T_g} {}^{\bullet}\!\langle\, g^{(l)}\,\rangle \tag{3.1}$$

of left ideals in $A[z;\sigma]$. All this leads to the following definition.

Definition 3.2 Let $\{0\} \neq \mathcal{C} \subseteq \mathbb{F}[z]^n$ be a σ-cyclic convolutional code with reduced generator polynomial $g \in A[z;\sigma]$. Then $\mathcal{C}$ is called *minimal* if g is a component polynomial, i. e., if $g = \varepsilon^{(l)}g$ for some $l = 1,\ldots,r$ or, alternatively, if $|T_g| = 1$.

Thus, by Equation (3.1) each σ-cyclic convolutional code is a direct sum of minimal σ-cyclic codes. As can be seen by some examples the decomposition into a direct sum of minimal codes is *not* unique. However, we will not need this property and thus omit an example. The notion "minimal" (which is not related to minimal generator matrices) is justified by the following result.

Proposition 3.3 *Let $\mathcal{C} \subseteq \mathbb{F}[z]^n$ be a σ-cyclic convolutional code with generator polynomial $g \in A[z;\sigma]$. Then the following are equivalent.*

(i) $\mathcal{C}$ is minimal,

(ii) $\mathcal{C} \neq \{0\}$ and $\mathcal{C}$ does not contain any proper σ-cyclic subcodes. Precisely, if $\hat{\mathcal{C}}$ is a σ-cyclic convolutional code and $\{0\} \neq \hat{\mathcal{C}} \subseteq \mathcal{C}$, then $\hat{\mathcal{C}} = \mathcal{C}$.

(iii) There exists a unit $u \in A[z;\sigma]$ such that $g = u^{(l)}$ for some index l.

Proof: (i) $\Rightarrow$ (ii): By assumption $0 \neq g = g^{(l)}$ for some index l. Let $\{0\} \neq \hat{\mathcal{C}}$ be a σ-cyclic convolutional code with reduced generator polynomial $h \neq 0$ and let $\hat{\mathcal{C}} \subseteq \mathcal{C}$. Then ${}^{\bullet}\!\langle\, h \,\rangle \subseteq {}^{\bullet}\!\langle\, g \,\rangle$, thus $h = fg$ for some $f \in A[z;\sigma]$. This implies the identity $h_0 = f_0 g_0$ for the constant terms of the polynomials. From Theorem 2.8(1) we know $g_0 = g_0^{(l)} \neq 0$, hence $h_0 = f_0 \varepsilon^{(l)} g_0 = h_0^{(l)}$. Using again Theorem 2.8(1) we deduce $T_h = T_{h_0} = \{l\}$. Thus $h = h^{(l)}$ and by Theorem 2.8(3) the codes $\hat{\mathcal{C}}$ and $\mathcal{C}$ have the same rank. From Lemma 1.3 we conclude $\hat{\mathcal{C}} = \mathcal{C}$.

(ii) $\Rightarrow$ (i): follows directly from Theorem 2.8(3) or Equation (3.1) since each component of the generator polynomial results in a σ-cyclic subcode of $\mathcal{C}$.

The equivalence (i) $\Leftrightarrow$ (iii) is clear with Theorem 2.8(2). $\square$

In the sequel we will show which algebraic parameters (n, k, δ) a minimal σ-cyclic convolutional code can attain. From Theorem 2.8(3) and Proposition 3.3 we have the following situation.

Remark 3.4 (a) Any component $u^{(l)}$ of a unit $u \in A[z;\sigma]$ defines a minimal σ-cyclic code $\mathfrak{v}({}^{\bullet}\!\langle\, u^{(l)} \,\rangle)$ with parameters (n, k, dk) where $k = \deg_x \pi_l$ and $d = \deg_z u^{(l)}$.

(b) Any minimal σ-cyclic code in $\mathbb{F}[z]^n$ with support $\{l\}$ has parameters (n, k, dk) and Forney index d counted k times, where $k = \deg_x \pi_l$ and d is the degree of the l-th component of a unit in $A[z;\sigma]$.

Hence the question raised above amounts to investigating as to which degrees can occur for a given component of a unit in $A[z;\sigma]$. The case where the complexity is zero is, of course, known from block code theory. Indeed, for each $k \in \{\deg_x \pi_1, \ldots, \deg_x \pi_r\}$ there exists a cyclic block code with parameters (n, k), hence a σ-cyclic convolutional code with parameters $(n, k, 0)$ for any automorphism σ. This follows also immediately from Remark 3.4(a). The existence of σ-cyclic convolutional codes with nonzero complexity, however, implies certain relations between the parameters and the automorphism. Indeed, we have

Lemma 3.5 *Let $\mathcal{C} \subseteq \mathbb{F}[z]^n$ be a minimal σ-cyclic code with generator polynomial $g = g^{(l)}$. Then $\mathcal{C}$ has complexity zero if and only if $g = g\varepsilon^{(l)}$. Furthermore, if $\mathcal{C}$ has nonzero complexity then $\sigma\big(\varepsilon^{(l)}\big) \neq \varepsilon^{(l)}$.*

Proof: First of all, the polynomial g is reduced by Remark 2.7, thus we may apply Theorem 2.8. If $\mathcal{C}$ has complexity zero, then, by Theorem 2.8(3), the polynomial g has degree zero, thus $g \in A$. But then $g = \varepsilon^{(l)}g = g\varepsilon^{(l)}$ follows from commutativity of A. Conversely, $g = \varepsilon^{(l)}g = g\varepsilon^{(l)}$ implies ${}^{\bullet}\!\langle\, g\,\rangle \subseteq {}^{\bullet}\!\langle\, \varepsilon^{(l)}\,\rangle$ and thus $\mathcal{C} \subseteq \mathfrak{v}({}^{\bullet}\!\langle\, \varepsilon^{(l)}\,\rangle)$. Both submodules are direct summands and by virtue of Theorem 2.8(3) they have the same rank. Thus, Lemma 1.3 implies $\mathcal{C} = \mathfrak{v}({}^{\bullet}\!\langle\, \varepsilon^{(l)}\,\rangle)$ and therefore has complexity zero (again by Theorem 2.8(3)). As for the last assertion notice that if $\sigma(\varepsilon^{(l)}) = \varepsilon^{(l)}$, the very definition of multiplication in the Piret algebra implies that $\varepsilon^{(l)}$ is in the center of $A[z;\sigma]$. Hence in this case $g = \varepsilon^{(l)}g = g\varepsilon^{(l)}$ and the code has complexity zero by the first part of the lemma. □

As a consequence we have that for fixed parameters n and $|\mathbb{F}|$ a given automorphism $\sigma \in \mathrm{Aut}_{\mathbb{F}}(A)$ admits (minimal) σ-cyclic convolutional codes of positive complexity only if the induced permutation $\Pi_\sigma \in S_r$ (see Definition 2.3) is nontrivial. According to Equation (2.10) this in turn is possible only if $x^n - 1$ has (at least) two prime factors of the same degree. Recall that one easily obtains the degrees of the prime factors of $x^n - 1$ by computing the cyclotomic cosets modulo n over $\mathbb{F}$; see [10, Ch. 7, § 5]. With different methods it has been shown in [16, Sec. VI] and in [4, Prop. 3.4] that the condition $\Pi_\sigma \neq \mathrm{id}$ is not only necessary but also sufficient for the existence of σ-cyclic codes with positive complexity. Our goal is to prove even more. We will show that for any $\sigma \in \mathrm{Aut}_{\mathbb{F}}(A)$ and any $l \in \{1, \ldots, r\}$ such that $\sigma(\varepsilon^{(l)}) \neq \varepsilon^{(l)}$ and for any $d \in \mathbb{N}$ there exists a minimal σ-cyclic code with parameters (n, k, kd) where $k = \deg_x \pi_l$. To this aim we need the following notion.

Definition 3.6 Let $\sigma \in \mathrm{Aut}_{\mathbb{F}}(A)$ and $l \in \{1, \ldots, r\}$. Then the l-order of σ is defined as $o_l(\sigma) := \min\{m \in \mathbb{N} \mid \sigma^m(\varepsilon^{(l)}) = \varepsilon^{(l)}\}$.

Using the permutation $\Pi_\sigma \in S_r$ the l-order can also be expressed as $o_l(\sigma) = \min\{m \in \mathbb{N} \mid \Pi_\sigma^m(l) = l\}$. In other words, the l-order of σ is the length of the cycle of Π_σ containing l; therefore

$$l \equiv_\sigma l' \Longrightarrow o_l(\sigma) = o_{l'}(\sigma). \tag{3.2}$$

With the following lemma we will establish the existence of units in $A[z;\sigma]$ with a particularly simple form. They will suffice to show the existence of the desired minimal σ-cyclic codes. We will also obtain that each unit in $A[z;\sigma]$ can be expressed as a finite product of these simple units. In this sense we can construct, at least theoretically, all units of $A[z;\sigma]$ and thus, by Theorem 2.8(2), all σ-cyclic convolutional codes.

Lemma 3.7 *Let $\sigma \in \mathrm{Aut}_{\mathbb{F}}(A)$ with l-order $o_l := o_l(\sigma)$ where $l \in \{1, \ldots, r\}$.*

(a) Let $a \in A$ and $d \in \mathbb{N}_0$. Put $u_{d,a,l} := 1 + z^d a\varepsilon^{(l)} \in A[z;\sigma]$. Then

$$\big[u_{d,a,l} \text{ is a unit in } A[z;\sigma] \Longleftrightarrow a^{(l)} = 0 \text{ or } o_l \nmid d\big] \quad \text{if } d > 0,$$

and

$$\big[u_{d,a,l} \text{ is a unit in } A[z;\sigma] \iff \quad a^{(l)} \neq -\varepsilon^{(l)} \quad \big] \quad \text{if } d = 0.$$

If $u_{d,a,l}$ is a unit in $A[z;\sigma]$, then its inverse is given by $u_{d,-a,l}$. In this case we call $u_{d,a,l}$ an elementary unit.

(b) Any unit in $A[z;\sigma]$ can be written as a finite product of elementary units.

Proof: (a) If $d = 0$ then $u_{d,a,l} = 1 + a^{(l)}$ and the assertion follows from (2.9). Thus let $d > 0$. We may assume $a^{(l)} \neq 0$ for otherwise the assertion is trivial.
"$\Rightarrow$" Write $u := u_{d,a,l}$, for short. Since u is a unit, we know from Remark 3.4(a) that $\mathfrak{v}({}^{\bullet}\!\langle\, u^{(l)}\,\rangle)$ is a minimal σ-cyclic convolutional code and its complexity is given by $\deg_x \pi_l \deg_z u^{(l)}$. If $o_l \mid d$ then $\varepsilon^{(l)} z^d = z^d \varepsilon^{(l)}$, and thus $u^{(l)} = \varepsilon^{(l)} u = u\varepsilon^{(l)} = \varepsilon^{(l)} + z^d a^{(l)}$, hence $\deg_z u^{(l)} = d > 0$. But on the other side Lemma 3.5 implies that the complexity of $\mathfrak{v}({}^{\bullet}\!\langle\, u^{(l)}\,\rangle)$ is zero, a contradiction.
"$\Leftarrow$" Let $o_l \nmid d$. Then $\sigma^d(\varepsilon^{(l)}) \neq \varepsilon^{(l)}$ and thus $\sigma^d(\varepsilon^{(l)})\varepsilon^{(l)} = 0$. But then

$$u_{d,a,l} u_{d,-a,l} = (1 + z^d a\varepsilon^{(l)})(1 - z^d a\varepsilon^{(l)}) = 1,$$

and likewise $u_{d,-a,l} u_{d,a,l} = 1$, completing the proof of (a).
(b) Let $u \in A[z;\sigma]$ be a unit. Then ${}^{\bullet}\!\langle\, u\,\rangle = A[z;\sigma]$ and thus the constant polynomial $1 \in A$ is a reduced generator polynomial of ${}^{\bullet}\!\langle\, u\,\rangle$, see Remark 2.7. In [4, Cor. 4.13(a) and its proof] it has been shown that the reduction of a single polynomial in $A[z;\sigma]$ can be described by left multiplication with suitable elementary units. In other words, there exist elementary units $u_1, \ldots, u_t \in A[z;\sigma]$ such that $1 = u_t \cdot \ldots \cdot u_1 u$. This proves the assertion. □

It should be noticed that from a coding theoretic point of view the elementary units are not desirable for code construction if d is big. Indeed, since the coefficients of $z, z^2, \ldots, z^{d-1}$ are zero, the same is true for the coefficients of any component $u^{(l)}$ and thus the code $\mathfrak{v}({}^{\bullet}\!\langle\, u^{(l)}\,\rangle)$ has small distance. This argument, of course, does not apply if $d = 1$, and we will proceed with that more specific case. These units are not only candidates for the construction of good codes but, as we will see next, will lead us to the existence of the desired minimal σ-cyclic codes. To this end we will now construct units whose l-th component have a prescribed degree.

Corollary 3.8 *Let $\sigma \in \mathrm{Aut}_{\mathbb{F}}(A)$ and $l \in \{1, \ldots, r\}$ such that $\sigma(\varepsilon^{(l)}) \neq \varepsilon^{(l)}$. Then*

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

(2) For any $d \in \mathbb{N}_0$ and any units $a_1, \ldots, a_d$ in A the polynomial $u := u_{a_1}(1) \cdot \ldots \cdot u_{a_d}(d)$ is a unit in $A[z;\sigma]$ and satisfies $\deg_z u^{(l)} = d = \deg_z u$.

Proof: (1) If $\deg_z u_a(i) = 0$ the assertion is trivial. Thus let us assume $\deg_z u_a(i) = 1$. Note that, with the notation of Lemma 3.7, $u_a(i) = u_{1,a,l'}$ where l' is such that $\sigma^i(\varepsilon^{(l)}) = \varepsilon^{(l')}$. From (3.2) we obtain $o_l(\sigma) = o_{l'}(\sigma)$ and by assumption this number is bigger than 1. Thus $o_{l'}(\sigma) \nmid \deg_z u_a(i)$ and Lemma 3.7(a) implies the assertion.
(2) Without loss of generality let $d > 0$. Let $u := u_{a_1}(1) \cdot \ldots \cdot u_{a_d}(d)$ where $a_1, \ldots, a_d$ are units in A. Part (a) implies that u is a unit in $A[z;\sigma]$ and it obviously satisfies $\deg_z u \leq d$. In order to show $\deg_z u = d$ we compute the z^d-term of u. It is given by

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

where $a := \sigma^{d-1}(a_1)\sigma^{d-2}(a_2) \cdot \ldots \cdot \sigma(a_{d-1})a_d$. Since $a_1, \ldots, a_d$ are units in A, the same is true for a. Thus $a\sigma^d(\varepsilon^{(l)}) \neq 0$ and we have $\deg_z u = d$. Finally, $\deg_z u^{(l)} = d$ since $\varepsilon^{(l)} z^d a\sigma^d(\varepsilon^{(l)}) = z^d a\sigma^d(\varepsilon^{(l)}) \neq 0$. □

We would like to mention that for the unit u thus constructed $\deg_z u^{(l')} < d$ whenever $l' \neq l$. This can easily be seen from the above.

The following theorem combines our findings about the existence of minimal σ-cyclic convolutional codes. The proof follows from Corollary 3.8(2) along with Remark 3.4(a) and from Lemma 3.5.

Theorem 3.9 *Let $\sigma \in \mathrm{Aut}_{\mathbb{F}}(A)$ and $l \in \{1,\ldots,r\}$. Put $k := \deg_x \pi_l$ where π_l is as in (2.7). Then the following are equivalent:*

(i) $\sigma(\varepsilon^{(l)}) \neq \varepsilon^{(l)}$.

(ii) For any $d \in \mathbb{N}_0$ one can construct a minimal σ-cyclic convolutional code with parameters (n,k,dk) and support $\{l\}$. The Forney indices of the code are all equal to d.

(iii) There exists a σ-cyclic convolutional code with nonzero complexity and support $\{l\}$.

Notice that the considerations so far do not lead to any insight about the quality of a minimal σ-cyclic convolutional code, that is, about the distance. In a forthcoming paper we will use the units of Corollary 3.8 in order to construct CCC's with parameters $(n,1,\delta)$, where $\delta \leq n-1$, that are all MDS, i. e., the distances of these codes attain the generalized Singleton bound given in (1.2). In the rest of this section we will restrict ourselves to presenting examples of several 2-dimensional codes using the ideas above. They are all optimal (i. e., their distances attain the Griesmer bound) suggesting that this construction is worth being investigated with respect to distance properties. As for the general situation, we wish to add that the codes constructed in Theorem 3.9(ii) are *compact*, which in this case (rank k dividing the complexity) means that the Forney indices are all the same [12, Cor. 4.3]. In general, compact codes are better candidates for good codes; for instance, codes attaining the generalized Singleton bound (1.2) are always compact [19, Proof of Thm. 2.2].

Example 3.10 We begin with the case $n=3$ and $\mathbb{F} := \mathbb{F}_4 = \{0,1,\alpha,\alpha^2\}$ where $\alpha^2+\alpha+1=0$. Thus $A = \mathbb{F}[x]/_{\langle x^3-1\rangle}$ and we have the prime factor decomposition $x^3-1=\pi_1\pi_2\pi_3$ where $\pi_1 = x+1$, $\pi_2 = x+\alpha$, and $\pi_3 = x+\alpha^2$. The corresponding primitive idempotents are

$$\varepsilon^{(1)} = x^2+x+1,\ \varepsilon^{(2)} = \alpha x^2+\alpha^2 x+1,\ \varepsilon^{(3)} = \alpha^2 x^2+\alpha x+1$$

as can readily be seen by verifying $(\varepsilon^{(i)} \bmod \pi_j) = \delta_{ij}$ for $i,\, j = 1,2,3$. We will use the automorphism $\sigma \in \mathrm{Aut}_{\mathbb{F}}(A)$ defined by $\sigma(x) = x^2$. One easily checks $\sigma(\varepsilon^{(2)}) = \varepsilon^{(3)}$ and vice versa. Hence $\Pi_\sigma = (1)(2,3)$. We will construct minimal σ-cyclic codes with support $\{2\}$ by using the construction of units in Corollary 3.8 for $l=2$. Choose the units

$$v_1 = u_1(1),\ v_2 = u_\alpha(2),\ v_3 = u_{\alpha^2}(3),\ v_4 = u_\alpha(4),\ v_5 = u_{\alpha^2}(5),\ v_6 = u_\alpha(6) \in A[z;\sigma]$$

and put $g_\delta := \varepsilon^{(2)}(v_1 \cdot \ldots \cdot v_\delta)$ for $\delta = 1,\ldots,6$. From Corollary 3.8(2) we obtain $\deg_z g_\delta = \delta$ and thus $\mathcal{C}_\delta := \mathfrak{v}({}^\bullet\langle g_\delta\rangle)$ is a σ-cyclic code with parameters $(3,1,\delta)$ over $\mathbb{F}_4$. Using Maple we computed the distances of these codes which turn out to be very good in each case. Indeed, the respective distances are

$$\mathrm{dist}(\mathcal{C}_1)=6,\ \mathrm{dist}(\mathcal{C}_2)=9,\ \mathrm{dist}(\mathcal{C}_3)=12,\ \mathrm{dist}(\mathcal{C}_4)=14,\ \mathrm{dist}(\mathcal{C}_5)=16,\ \mathrm{dist}(\mathcal{C}_6)=18.$$

For $\delta = 1, \ldots, 5$ the distances attain the Griesmer bound (1.3), hence these codes are optimal (for $\delta = 1, 2, 3$ this is even the generalized Singleton bound (1.2)). For $\delta = 6$ the computed distance is just one less than the Griesmer bound, which in this case is 19. It should be added that, as to our knowledge, it is unknown whether there exists any code over $\mathbb{F}_4$ with algebraic parameters $(3, 1, 6)$ and distance 19. Recall from Theorem 2.8(3) that $G_\delta := \mathfrak{v}(g_\delta)$ is a generator matrix of $\mathcal{C}_\delta$. These matrices are given explicitly by

$$\begin{aligned}
G_1 &= [z+1,\ \alpha z+\alpha^2,\ \alpha^2 z+\alpha],\\
G_2 &= [\alpha z^2+z+1,\ z^2+\alpha z+\alpha^2,\ \alpha^2 z^2+\alpha^2 z+\alpha],\\
G_3 &= [z^3+\alpha z^2+\alpha z+1,\ \alpha z^3+z^2+\alpha^2 z+\alpha^2,\ \alpha^2 z^3+\alpha^2 z^2+z+\alpha],\\
G_4 &= [\alpha z^4+z^3+z^2+\alpha z+1,\ z^4+\alpha z^3+\alpha^2 z^2+\alpha^2 z+\alpha^2,\ \alpha^2 z^4+\alpha^2 z^3+\alpha z^2+z+\alpha],\\
G_5 &= [z^5+\alpha z^4+\alpha z^3+z^2+z+1,\ \alpha z^5+z^4+\alpha^2 z^3+\alpha^2 z^2+\alpha z+\alpha^2,\\
&\qquad\qquad \alpha^2 z^5+\alpha^2 z^4+z^3+\alpha z^2+\alpha^2 z+\alpha],\\
G_6 &= [\alpha z^6+z^5+z^4+\alpha z^3+\alpha^2 z^2+z+1,\ z^6+\alpha z^5+\alpha^2 z^4+\alpha^2 z^3+\alpha z^2+\alpha z+\alpha^2,\\
&\qquad\qquad \alpha^2 z^6+\alpha^2 z^5+\alpha z^4+z^3+z^2+\alpha^2 z+\alpha].
\end{aligned}$$

Example 3.11 Now we consider the case $n = 5$ and $\mathbb{F} = \mathbb{F}_4 = \{0, 1, \alpha, \alpha^2\}$. In this case $x^5 - 1 = \pi_1\pi_2\pi_3$ where $\pi_1 = x + 1$, $\pi_2 = x^2 + \alpha x + 1$, and $\pi_3 = x^2 + \alpha^2 x + 1$ and the corresponding primitive idempotents are

$$\varepsilon^{(1)} = x^4+x^3+x^2+x+1,\ \varepsilon^{(2)} = \alpha x^4+\alpha^2 x^3+\alpha^2 x^2+\alpha x,\ \varepsilon^{(3)} = \alpha^2 x^4+\alpha x^3+\alpha x^2+\alpha^2 x.$$

We choose the automorphism defined via $\sigma(x) = x^2$. Again it is easily seen that $\sigma(\varepsilon^{(2)}) = \varepsilon^{(3)}$ and vice versa. We will use Corollary 3.8 for $l = 2$ in order to construct minimal σ-cyclic codes with support $\{2\}$. We define

$$v_1 := u_1(1),\ v_2 := u_\alpha(2),\ v_3 := u_{\alpha^2}(3)$$

and put $g_m := \varepsilon^{(2)} v_1 \cdot \ldots \cdot v_m$ for $m = 1, 2, 3$. Then we know $\deg_z g_m = m$ and that $\mathcal{C}_m := \mathfrak{v}({}^{\bullet}\!\langle\, g_m \rangle)$ is a σ-cyclic code over $\mathbb{F}_4$ with parameters $(5, 2, 2m)$ for $m = 1, 2, 3$. Again we computed the distances and they are optimal in each case. Indeed, $\mathrm{dist}(\mathcal{C}_1) = 8$, $\mathrm{dist}(\mathcal{C}_2) = 12$, and $\mathrm{dist}(\mathcal{C}_3) = 16$, which in each case is the Griesmer bound (1.3) for codes over $\mathbb{F}_4$ with parameters $(5, 2, 2m)$. Theorem 2.8(3) implies that the generator matrix of $\mathcal{C}_m$ is made up by the two rows $\mathfrak{v}(g_m)$ and $\mathfrak{v}(x g_m)$. They are given by

$$G_1 = \begin{bmatrix} 0 & \alpha+\alpha^2 z & \alpha^2+\alpha z & \alpha^2+\alpha z & \alpha+\alpha^2 z \\ \alpha+\alpha z & \alpha^2 z & \alpha & \alpha^2+\alpha^2 z & \alpha^2+\alpha z \end{bmatrix},$$

$$G_2 = \begin{bmatrix} 0 & \alpha+\alpha^2 z+\alpha^2 z^2 & \alpha^2+\alpha z+z^2 & \alpha^2+\alpha z+z^2 & \alpha+\alpha^2 z+\alpha^2 z^2 \\ \alpha+\alpha z+\alpha^2 z^2 & \alpha^2 z+z^2 & \alpha+z^2 & \alpha^2+\alpha^2 z+\alpha^2 z^2 & \alpha^2+\alpha z \end{bmatrix},$$

$$G_3 = \begin{bmatrix} 0 & \alpha+z+\alpha^2 z^2+\alpha^2 z^3 & \alpha^2+\alpha^2 z+z^2+\alpha z^3 & \alpha^2+\alpha^2 z+z^2+\alpha z^3 & \alpha+z+\alpha^2 z^2+\alpha^2 z^3 \\ \alpha+\alpha^2 z+\alpha^2 z^2+\alpha z^3 & z+z^2+\alpha z^3 & \alpha+z^2+\alpha^2 z^3 & \alpha^2+z+\alpha^2 z^2 & \alpha^2+\alpha^2 z+\alpha^2 z^3 \end{bmatrix}.$$

Remark 3.12 In [3, Table II] some other sequences of codes over $\mathbb{F}_4$ with parameters $(3,1,\delta)$ for $\delta=1,\ldots,5$ and $(5,2,2m), m=1,2,3$ have been presented. They have the same distances as the ones given in the previous two examples, hence are also optimal. It is worth pointing out that those codes and the ones presented here are *not* monomially equivalent, where we call two codes $\operatorname{im} G$ and $\operatorname{im} G'$ *monomially equivalent* if $G=G'PD$ where $P\in Gl_n(\mathbb{F})$ is a permutation matrix and $D\in Gl_n(\mathbb{F})$ is a nonsingular diagonal matrix; see also [5, p.24] for monomial equivalence in the block code case. In other words, codes are monomially equivalent if they only differ by a permutation and a rescaling of the entries of the codewords. Monomially equivalent codes have, of course, the same algebraic parameters and the same distance. From a coding point of view they have the same properties and can therefore be identified. From this point of view, the two families of codes obtained in the examples above are significantly different from those constructed in [3].

4 Orthogonal Sums of Minimal Cyclic Codes

In this section we will extend the existence result from Theorem 3.9 to certain non minimal σ-cyclic codes. In fact we will generalize that theorem to codes with unmixed generator polynomials. To this end we will make use of the fact that component polynomials corresponding to disjoint cycles of Π_σ are orthogonal. Precisely

$$f,\, g\in A[z;\sigma],\ k\not\equiv_\sigma l \Longrightarrow f^{(k)}g^{(l)}=g^{(l)}f^{(k)}=0. \tag{4.1}$$

Again, let $\mathbb{F}$ be a finite field such that $|\mathbb{F}|$ and n are coprime and let $\sigma\in\operatorname{Aut}_{\mathbb{F}}(A)$ be a fixed automorphism. We will make heavy use of the prime factor decomposition (2.7) and the notations introduced in Definition 2.3.

Lemma 4.1 *Let $l_1,\ldots,l_t\in\{1,\ldots,r\}$ be such that $l_i\not\equiv_\sigma l_j$ for $i\neq j$. Furthermore, put $I:=\{1,\ldots,r\}\backslash\{l\mid l\equiv_\sigma l_i$ for some $i=1,\ldots,t\}$.*

(1) Let $u\in A[z;\sigma]$ be a unit with inverse $u^{-1}=\bar{u}$. Then

$$\sum_{j\equiv_\sigma l_i} u^{(j)}\sum_{j\equiv_\sigma l_i}\bar{u}^{(j)}=\sum_{j\equiv_\sigma l_i}\varepsilon^{(j)} \text{ for each fixed } i=1,\ldots,t$$

and

$$\sum_{j\in I} u^{(j)}\sum_{j\in I}\bar{u}^{(j)}=\sum_{j\in I}\varepsilon^{(j)}.$$

(2) For $i=1,\ldots,t$ let $u_i\in A[z;\sigma]$ be a unit with inverse $u_i^{-1}=\bar{u}_i$ and let $u\in A[z;\sigma]$ be a unit with inverse $u^{-1}=\bar{u}$. Then the element $w:=\sum_{i=1}^t\sum_{j\equiv_\sigma l_i}u_i^{(j)}+\sum_{j\in I}u^{(j)}$ is a unit with inverse $w^{-1}=\sum_{i=1}^t\sum_{j\equiv_\sigma l_i}\bar{u}_i^{(j)}+\sum_{j\in I}\bar{u}^{(j)}$.

Proof: (1) The implication in (4.1) yields

$$u\bar{u}=\sum_{i=1}^t\Big(\sum_{j\equiv_\sigma l_i}u^{(j)}\sum_{j\equiv_\sigma l_i}\bar{u}^{(j)}\Big)+\sum_{j\in I}u^{(j)}\sum_{j\in I}\bar{u}^{(j)}=1=\sum_{j=1}^r\varepsilon^{(j)}.$$

From this the assertion follows immediately since the coefficients of each of the first t summands are contained in $\sum_{j\equiv_\sigma l_i}\varepsilon^{(j)}A$ while those of the second sum are in $\sum_{j\in I}\varepsilon^{(j)}A$ and these two sets have only the zero element in common.
(2) follows from (1) along the same line of arguments. □

All this leads to the existence of units with prescribed degrees for pairwise orthogonal components (in the sense of (4.1)).

Theorem 4.2 *Let $l_1,\dots,l_t \in \{1,\dots,r\}$ be such that $l_i \not\equiv_\sigma l_j$ for $i \neq j$. Furthermore assume $o_{l_i}(\sigma) > 1$, that is, $\sigma(\varepsilon^{(l_i)}) \neq \varepsilon^{(l_i)}$, for all $i = 1,\dots,t$. Then for all $d_1,\dots,d_t \in \mathbb{N}_0$ there exists a unit $w \in A[z;\sigma]$ such that $g := \sum_{i=1}^t w^{(l_i)}$ is unmixed and $\deg_z g^{(l_i)} = d_i$ for $i = 1,\dots,t$.*

Proof: From Corollary 3.8(2) we know that for each $i = 1,\dots,t$ there exists a unit u_i such that $\deg_z u_i^{(l_i)} = d_i$. Put $w := \sum_{i=1}^t \sum_{j\equiv_\sigma l_i} u_i^{(j)} + \sum_{i\in I} u_1^{(i)}$, where again $I = \{1,\dots,r\}\backslash\{l \mid l \equiv_\sigma l_i$ for some $i = 1,\dots,t\}$. Then Lemma 4.1(2) yields the desired results. □

Using Theorem 2.8(3) we obtain immediately the existence of orthogonal sums of minimal cyclic codes with prescribed Forney indices.

Corollary 4.3 *Let $l_1,\dots,l_t \in \{1,\dots,r\}$ be such that $l_i \not\equiv_\sigma l_j$ for $i \neq j$ and such that $o_{l_i}(\sigma) > 1$ for all $i = 1,\dots,t$. Put $k_i := \deg_x \pi_{l_i}$ where π_j is as in (2.7). Then for all $d_1,\dots,d_t \in \mathbb{N}_0$ there exists a σ-cyclic code $\mathcal{C} \subseteq \mathbb{F}[z]^n$ with parameters (n,k,δ) where*

$$k = \sum_{i=1}^t k_i \text{ and } \delta = \sum_{i=1}^t k_i d_i.$$

The support is given by $\{l_1,\dots,l_t\}$.

Note that according to Theorem 2.8(3), any σ-cyclic code with support $\{l_1,\dots,l_t\}$ has to have parameters of the type above.

The arguments above may be used to construct non-minimal codes with given algebraic parameters and an unmixed generator polynomial directly out of minimal codes. We formulate the result in terms of direct summands in $\mathbb{F}[z]^n$.

Theorem 4.4 *For $i = 1,\dots,t$ let $\mathcal{C}_i \subseteq \mathbb{F}[z]^n$ be a minimal σ-cyclic code with support $\{l_i\}$ and complexity δ_i and assume $l_i \not\equiv_\sigma l_j$ for $i \neq j$. Then $\mathcal{C} := \sum_{i=1}^t \mathcal{C}_i \subseteq \mathbb{F}[z]^n$ is a σ-cyclic code, too. Its rank is given by $\operatorname{rank}\mathcal{C} = \sum_{i=1}^t \operatorname{rank}\mathcal{C}_i = \sum_{i=1}^t \deg_x \pi_{l_i}$, and its complexity is $\delta(\mathcal{C}) = \delta_1 + \ldots + \delta_t$. Furthermore, $\mathcal{C} = \bigoplus_{i=1}^t \mathcal{C}_i$ and its Forney indices are given by the union of the Forney indices of the codes $\mathcal{C}_1,\dots,\mathcal{C}_t$.*

Proof: For all $i = 1,\dots,t$ let $\mathcal{C}_i = \mathfrak{v}(\overset{\bullet}{\langle} g_i \rangle)$ where $g_i = u_i^{(l_i)}$ for some unit $u_i \in A[z;\sigma]$. Put $g := g_1 + \ldots + g_t$ and $\mathcal{C} := \mathfrak{v}(\overset{\bullet}{\langle} g \rangle)$. Then the polynomial g is unmixed and by Lemma 4.1(2) $g = \sum_{i=1}^t w^{(l_i)}$ for some suitable unit $w \in A[z;\sigma]$. Hence, by Theorem 2.8(2), the submodule $\mathcal{C} = \mathfrak{v}(\overset{\bullet}{\langle} g \rangle)$ is a direct summand, and by part (3) of that theorem it is the direct sum of $\mathcal{C}_1,\dots,\mathcal{C}_t$ and has the desired rank, complexity, and Forney indices. □

We wish to illustrate the above by an example, indicating that this construction does indeed lead to good codes.

Example 4.5 Let $n = 7$ and $\mathbb{F} = \mathbb{F}_8 = \{0, 1, \alpha, \alpha^2, \ldots, \alpha^6\}$ where $\alpha^3 + \alpha + 1 = 0$. Then $x^7 - 1 = \prod_{i=0}^{6} \pi_i$, where $\pi_i = x - \alpha^i$. Since all fields $\mathbb{F}[x]/_{\langle \pi_i \rangle}$ are isomorphic to $\mathbb{F}_8$, the automorphisms on $A = \mathbb{F}[x]/_{\langle x^7 - 1 \rangle}$ are fully determined by the their induced permutation Π_σ. In other words, $\mathrm{Aut}_{\mathbb{F}}(A) \cong S_7$. We choose the automorphism σ corresponding to the permutation $\Pi_\sigma = (1,2)(3,4,5)(6)(7)$. Moreover, we take the polynomials

$$g_1 = \varepsilon^{(1)} + z\varepsilon^{(2)} + z^2\varepsilon^{(1)}\alpha \text{ and } g_2 = \varepsilon^{(3)} + z\varepsilon^{(4)}\alpha + z^2\varepsilon^{(5)}\alpha^2.$$

Then $g_1 = \varepsilon^{(1)}g_1$ and $g_2 = \varepsilon^{(3)}g_2$. Since both polynomials, being components, are reduced, Theorem 2.8(3) tells us that ${}^{\bullet}\!\langle\, g_1 \,\rangle$ and ${}^{\bullet}\!\langle\, g_2 \,\rangle$ are submodules of rank 1 and complexity 2 each. It can be checked via some tedious but straightforward calculation that the associated matrices $\mathfrak{v}(g_i)$ are right invertible (they are the rows of the matrix below), thus $\mathfrak{v}({}^{\bullet}\!\langle\, g_1 \,\rangle)$ and $\mathfrak{v}({}^{\bullet}\!\langle\, g_2 \,\rangle)$ are both direct summands of $\mathbb{F}_8[z]^7$. Hence they are σ-cyclic codes over $\mathbb{F}_8$ with parameters $(7, 1, 2)$ each. Since $1 \not\equiv_\sigma 3$ the polynomial $g = g_1 + g_2$ is unmixed and ${}^{\bullet}\!\langle\, g \,\rangle$ is a direct summand according to Theorem 4.4. A minimal generator matrix of the code $\mathfrak{v}({}^{\bullet}\!\langle\, g \,\rangle) \subseteq \mathbb{F}_8[z]^7$ is given by

$$\begin{bmatrix} 1{+}z{+}\alpha z^2 & 1{+}\alpha^6 z{+}\alpha z^2 & 1{+}\alpha^5 z{+}\alpha z^2 & 1{+}\alpha^4 z{+}\alpha z^2 & 1{+}\alpha^3 z{+}\alpha z^2 & 1{+}\alpha^2 z{+}\alpha z^2 & 1{+}\alpha z{+}\alpha z^2 \\ 1{+}\alpha z{+}\alpha^2 z^2 & \alpha^5{+}\alpha^5 z{+}\alpha^5 z^2 & \alpha^3{+}\alpha^2 z{+}\alpha z^2 & \alpha{+}\alpha^6 z{+}\alpha^4 z^2 & \alpha^6{+}\alpha^3 z{+}z^2 & \alpha^4{+}z{+}\alpha^3 z^2 & \alpha^2{+}\alpha^4 z{+}\alpha^6 z^2 \end{bmatrix}.$$

The first and second row generate the codes $\mathfrak{v}({}^{\bullet}\!\langle\, g_1 \,\rangle)$ and $\mathfrak{v}({}^{\bullet}\!\langle\, g_2 \,\rangle)$. Again, all codes involved are optimal with respect to their distance. Both the codes $\mathfrak{v}({}^{\bullet}\!\langle\, g_i \,\rangle)$, $i = 1, 2$, have distance 21, which is the generalized Singleton bound (1.2). Hence these codes are MDS codes in the sense of [19]. The code $\mathfrak{v}({}^{\bullet}\!\langle\, g \,\rangle)$ has distance 18, which is the optimum value for codes over $\mathbb{F}_8$ with parameters $(7, 2, 4)$ due to the Griesmer bound (1.3).

Finally we wish to add that the existence result of Corollary 4.3 does not hold without the restriction $l_i \not\equiv_\sigma l_j$ for $i \neq j$. More precisely, in general it is *not* possible to arbitrarily prescribe the degrees of the components of a reduced, but not unmixed, polynomial. For instance, one can show that in the situation of Example 3.11 no σ-CCC with algebraic parameters $(5, 4, 2)$ exists. In this case one has to make detailed use of the definition of reducedness as given in [4].

5 Open Problems

We close the paper with some problems open to future research. As indicated at the end of the last section, in the general situation it remains open as to which Forney indices (and complexity) a σ-cyclic code can attain. But from a coding theoretic point of view an investigation of σ-cyclic codes with respect to their distance is much more important. More precisely, it needs to be investigated whether one can relate the distance of a CCC to some properties of the reduced generator polynomial (or any other suitable generating polynomial of the associated left ideal). As a starting point one might begin with minimal codes. In particular we think it is worth to investigate the construction of minimal codes via units as described in Corollary 3.8(2). As indicated earlier, in a forthcoming paper we will present a construction of 1-dimensional (thus minimal) cyclic MDS codes using the ideas of Section 3. Furthermore, it is also unclear which automorphisms should be chosen for obtaining good codes. Finally, the class of all CCC's of a given length should be investigated with respect to monomial equivalence in the sense given in Remark 3.12. First ideas can be found in [9]. They indicate that one may restrict to certain automorphisms in order to cover all equivalence classes. A detailed positive result would considerably reduce the amount of data to be investigated for the search of good CCC's.

Acknowledgement

We wish to thank the anonymous reviewers for their helpful comments. They improved the exposition considerably.

References

[1] G. D. Forney Jr. Convolutional codes I: Algebraic structure. *IEEE Trans. Inform. Theory*, IT-16:720–738, 1970. (see also corrections in *IEEE Trans. Inf. Theory*, vol. 17,1971, p. 360).

[2] G. D. Forney Jr. Minimal bases of rational vector spaces, with applications to multivariable linear systems. *SIAM J. on Contr.*, 13:493–520, 1975.

[3] H. Gluesing-Luerssen and W. Schmale. Distance bounds for convolutional codes and some optimal codes. Preprint 2003. Submitted. Available at http://front.math.ucdavis.edu/ with ID-number RA/0305135.

[4] H. Gluesing-Luerssen and W. Schmale. On cyclic convolutional codes. *Acta Applicandae Mathematicae*, 82:183–237, 2004.

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

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

[7] J. Justesen. New convolutional code constructions and a class of asymptotically good time-varying codes. *IEEE Trans. Inform. Theory*, IT-19:220–225, 1973.

[8] J. Justesen. Algebraic construction of rate $1/\nu$ convolutional codes. *IEEE Trans. Inform. Theory*, IT-21:577–580, 1975.

[9] B. Langfeld. Minimal cyclic convolutional codes. Diploma Thesis at the University of Oldenburg (Germany). Available at http://www-m9.ma.tum.de/dm/homepages/langfeld/thesis.pdf, 2003.

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

[11] J. L. Massey, D. J. Costello, and J. Justesen. Polynomial weights and code constructions. *IEEE Trans. Inform. Theory*, IT-19:101–110, 1973.

[12] R. J. McEliece. The algebraic theory of convolutional codes. In V. Pless and W. Huffman, editors, *Handbook of Coding Theory, Vol. 1*, pages 1065–1138. Elsevier, Amsterdam, 1998.

[13] P. Piret. On a class of alternating cyclic convolutional codes. *IEEE Trans. Inform. Theory*, IT-12:64–69, 1975.

[14] P. Piret. Structure and constructions of cyclic convolutional codes. *IEEE Trans. Inform. Theory*, IT-22:147–155, 1976.

[15] J. M. M. Porras, J. A. D. Pérez, J. I. I. Curto, and G. S. Sotelo. Convolutional Goppa codes. Preprint 2003. Available at http://front.math.ucdavis.edu/ with ID-number OC/0310149.

[16] C. Roos. On the structure of convolutional and cyclic convolutional codes. *IEEE Trans. Inform. Theory*, IT-25:676–683, 1979.

[17] J. Rosenthal. Connections between linear systems and convolutional codes. In B. Marcus and J. Rosenthal, editors, *Codes, Systems, and Graphical Models*, pages 39–66. Springer, Berlin, 2001.

[18] J. Rosenthal, J. M. Schumacher, and E. V. York. On behaviors and convolutional codes. *IEEE Trans. Inform. Theory*, IT-42:1881–1891, 1996.

[19] J. Rosenthal and R. Smarandache. Maximum distance separable convolutional codes. *Appl. Algebra Engrg. Comm. Comput.*, 10:15–32, 1999.

[20] R. Smarandache, H. Gluesing-Luerssen, and J. Rosenthal. Constructions of MDS-convolutional codes. *IEEE Trans. Inform. Theory*, IT-47:2045–2049, 2001.