On Behaviors and Convolutional Codes

Joachim Rosenthal *Senior Member, IEEE*, J.M. Schumacher *Member, IEEE*,
and Eric V. York *Student Member,IEEE*

Abstract— **It is well known that a convolutional code is essentially a linear system defined over a finite field. In this paper we elaborate on this connection. We will define convolutional codes as the dual of a complete linear behavior in the sense of Willems. Using ideas from systems theory we describe a set of generalized first order descriptions for convolutional codes. As an application of these ideas, we present a new algebraic construction for convolutional codes.**

Index-Terms: Convolutional codes, behaviors, duality, first order representations, code constructions.

I. Introduction

In this paper we take a detailed look at convolutional codes from the perspective of linear systems theory with an emphasis on duality relations and on the different representations of these codes. Using these representations, we present a construction of convolutional codes with distance lower bounded by the complexity of the encoder.

Throughout the relatively short history of the theory of convolutional codes, there have been several authors that have made the link between convolutional codes and linear systems theory. Among the first authors to do this were Massey and Sain. They published a series of papers [20], [21], [33], containing a systems theoretic analysis of convolutional codes and encoders. After this, Omura in [24] considered Viterbi decoding and its relationship to dynamic programming and later applications of control theory to optimal receiver design for convolutional codes [25]. In several landmark papers, [4], [5] Forney started to lay the foundation for the algebraic structure of convolutional codes.

Since these papers were written, there have been significant advances in the theory of linear systems. One notable advance has been the behavioral approach to linear systems of Willems, championed in the papers [39], [40], [41]. This point of view generated a renewed interest in the interplay between systems theory and convolutional coding theory and we would like to mention in particular the recent papers by Fornasini and Valcher [3], [37] and the recent papers [7], [16], [17] by Forney, Loeliger, Mittelholzer and Trott. Actually, as will be discussed in Section 2, some basic results of the behavioral theory of codes were developed by L. Staiger [34], [35] already in 1982.

In this paper we will introduce convolutional codes as submodules of the free module $\mathbb{F}^n[z]$. In doing so convolutional codes become dual to linear time invariant complete behaviors in the sense of Willems. In Section 2 we develop this viewpoint and we show how it fits into the current theory.

In Section 3 we will show that convolutional codes have some canonical first order representations as they are known to exist for time invariant linear complete behaviors. We also provide an algorithm to compute first order representations.

In Section 4 we use the representations from Section 3 to construct a class of convolutional codes whose free distance is lower bounded by the complexity + 1 of the encoder. Some of the results presented in Sections 2 and 3 of this paper appeared in abbreviated form in [42].

Supported in part by NSF grant DMS-9400965. The research for this paper was carried out in part while Joachim Rosenthal and Eric V. York were visitors at CWI, Amsterdam, The Netherlands. Joachim Rosenthal and Eric V. York are with the Department of Mathematics, University of Notre Dame, Notre Dame, IN, USA 46556-5683.

J.M. Schumacher is with CWI, P.O. Box 94079, 1090 GB Amsterdam and Department of Economics, Tilburg University, Tilburg, The Netherlands

II. The duality between codes and behaviors

In this section we shall be concerned with a behavioral interpretation of convolutional codes. In a series of papers (see for instance [38], [39], [40], [41]), J. C. Willems has advocated viewing a dynamical system primarily as a collection of trajectories, without necessarily having in mind some specification for instance by means of differential equations, transfer functions, or some other device. In the same way one can view a code as a collection of sequences without necessarily having in mind a particular method to describe this collection, such as for instance an encoding device or a syndrome former. A collection of trajectories is called a *behavior* by Willems, and his definition of this concept as given below is wide enough to include codes as a special case.

Definition 2.1: A *dynamical system* Σ is a triple

$$\Sigma = (T, W, \mathcal{B})$$

where $T \subseteq \mathbb{R}$ is the *time axis*, W is a set called the *signal alphabet*, and $\mathcal{B} \subseteq W^T$ is called the *behavior*. The elements of $\mathcal{B}$ are called the *trajectories* of the system.

The advantage of taking collections of trajectories as a starting point is that it becomes possible to discuss properties of dynamical systems without reference to some specific representation. For an illustration of this, consider the notion of 'free distance' of a convolutional code. This concept depends only on the collection of code sequences, not on the specific device that is used to generate those sequences. To give concrete algorithms for the design of codes with good distance properties one of course has to work with finite representations of codes, but for this purpose one may choose any representation form that is convenient for the problem at hand, and one is not tied for instance to representations in terms of encoding devices. We shall give an

example of such an approach in Section 4 below. In the final stages of a code design, one may want to construct representations that can be used for various purposes such as encoding and syndrome forming; for this one needs the theory of transformations between representations, some aspects of which will be addressed in Section 3.

As noted above, the definition of behaviors as given by Willems includes convolutional codes. Depending on the precise definition that one wants to use for convolutional codes, one may think of convolutional codes as similar to discrete-time behaviors, the main difference being that in the coding context one works over a finite field rather than over the real field as is the standard in discrete-time system theory. This seems to be the dominant point of view so far in the emerging literature on the behavioral approach to codes (see for instance [2], [3], [6], [16], [37]). Actually Forney and Trott in [6, p. 1491] explicitly state that they "treat the terms "code" and "system" as synonyms". However, there are some indications that the relation between codes and behaviors may for a number of purposes better be viewed as one of *duality* rather than as one of inclusion. We shall work out this point of view below.

A. A duality relation between codes and behaviors

Let $\mathbb{F} := \mathbb{F}_q$ be the Galois field with q elements. It is generally accepted to define a linear block code as a linear subspace of the vector space $\mathbb{F}^n$. The situation for convolutional codes is not so clear and there seems to be no universal agreement on how to define a convolutional code. Although it is natural in the behavioral framework to define a linear convolutional code as a linear subspace of a space of sequences of vectors over $\mathbb{F}$, this still leaves open the question whether the time axis should be $\mathbb{Z}$ or $\mathbb{Z}_+$, and whether the sequences should have finite, left-bounded, or infinite support. It is possible to avoid making these choices in an abstract setting, and of course it is part of the behavioral program to do just that, but there is a need to be specific once one starts to work with concrete representations. From an algebraic point of view, it is perhaps easiest to work with left-bounded sequences defined on $\mathbb{Z}$ since these may be identified with formal Laurent series and thus form a field; this approach is classical (see for instance [4]). In work that emphasizes connections to automata theory, Staiger [34], [35] uses $\mathbb{Z}_+$ as a time axis and allows infinite support. Fornasini and Valcher [2], [37] study 2D convolutional codes; their 'time axis' is $\mathbb{Z}^2$ and they consider mostly (doubly indexed) sequences with finite support.

In the approach we shall take, it will be crucial to consider sequences of finite support. Whether these sequences are defined on $\mathbb{Z}$ or on $\mathbb{Z}_+$ is much less essential, and we shall use both settings. The entire discussion below can be extended to the level of nD codes as in [2], [37] (cf. also the work of Rocha [30] and of Oberst [23] for nD behavioral theory), but for simplicity we shall remain within the 1D framework.

In anticipation of the duality that will be discussed below, and because we would like to use the results from [12], [31], we state the following definitions for the case in which the time axis is $\mathbb{Z}_+$.

Definition 2.2: A *finite-support linear convolutional code* is a right shift invariant subspace of $\mathbb{F}^n[z]$.

Definition 2.3: An *infinite-support discrete-time behavior* is a left shift invariant subspace of $\mathbb{F}^n[[z]]$.

For the case where the time axis is $\mathbb{Z}$, the definitions are the same except that right shift invariance for codes and left shift invariance for behaviors is replaced by shift invariance for both. So in this case the distinction between the two definitions is just in the finiteness requirements.

In this paper we shall refer to finite-support linear convolutional codes simply as *codes* and infinite-support discrete-time behaviors will be called just *behaviors*. Of course this terminology may be viewed as restrictive both with respect to codes and with respect to behaviors. We believe though that left shift invariance is a natural property if one thinks of behaviors as processes for which the state at time 0 is defined by some unspecified past, whereas right shift invariance is more natural if the state at time 0 must be zero. The first framework suggests itself in the study of physical phenomena, and is used as a standard in the work of Willems; the latter framework appears to be relevant in the context of coding theory where it is usually required that sender and receiver both start from the zero state.

The duality relations that we shall discuss are based on a bilinear form that is defined between the space of polynomials in a variable z, indicated by $\mathbb{F}[z]$, and the space of formal power series in z, indicated by $\mathbb{F}[[z]]$. The form is defined (in a vector version) as follows:

$$\begin{array}{rccc} (\,,\,): & \mathbb{F}^n[[z]] \times \mathbb{F}^n[z] & \longrightarrow & \mathbb{F} \\ & (w, v) & \mapsto & \sum_{i=0}^{\infty} \langle w_i, v_i \rangle \end{array} \tag{2.1}$$

where $\langle\,,\,\rangle$ represents the standard dot product on $\mathbb{F}^n$. The above definition applies to systems over $\mathbb{Z}_+$; the analogous definition for systems over $\mathbb{Z}$ uses Laurent polynomials and bi-infinite sequences, and has the summation extending from $-\infty$ to ∞. Note that the infinite sum is indeed well defined since at most finitely many terms are nonzero. The bilinear form above was apparently first used by Macaulay in 1916 [18, Section IV]. Macaulay used $\mathbb{F} = \mathcal{C}$ and vector space dimension $n = 1$, but he allowed an arbitrary number of variables; so in today's terminology, he considered the scalar complex nD case.

The bilinear form above brings with it a number of standard constructions and remarks. We shall give these for the $\mathbb{Z}_+$ case; analogous statements hold for the case of two-sided sequences. Note that the bilinear form $(\,,\,)$ is nonsingular in the sense that $(w, v) = 0$ for all $v \in \mathbb{F}^n[z]$ implies that $w = 0$, and $(w, v) = 0$ for all $w \in \mathbb{F}^n[[z]]$ implies that $v = 0$. For any subset $\mathcal{C}$ of $\mathbb{F}^n[z]$ one defines the annihilator ("inverse system" in Macaulay's terminology)

$$\mathcal{C}^\perp = \{w \in \mathbb{F}^n[[z]] \mid (w, v) = 0, \forall v \in \mathcal{C}\} \tag{2.2}$$

and the annihilator of a subset $\mathcal{B}$ of $\mathbb{F}^n[[z]]$ is

$$\mathcal{B}^\perp = \{v \in \mathbb{F}^n[z] \mid (w, v) = 0, \forall w \in \mathcal{B}\}. \tag{2.3}$$

We use σ to denote the left shift operator on $\mathbb{F}^n[[z]]$ (defined by $\sigma \sum_0^\infty w_i z^i = \sum_0^\infty w_{i+1} z^i$), and z to denote the right shift operator on $\mathbb{F}^n[z]$. It is easy to check that, if $w \in \mathbb{F}^n[[z]]$ and $v \in \mathbb{F}^n[z]$, one has $(w, zv) = (\sigma w, v)$. Also, if G is a matrix over $\mathbb{F}$ of size $n \times k$ and $w \in \mathbb{F}^n[[z]]$, $\ell \in \mathbb{F}^k[z]$, then $(w, G\ell) = (G^t w, \ell)$. By bilinearity, it follows that for every polynomial matrix $G(s) \in \mathbb{F}^{n\times k}[s]$ we have

$$(w, G(z)\ell) \;=\; (G^t(\sigma)w, \ell).$$

One also easily verifies that, if $\mathcal{B}$ is a left shift invariant subspace of $\mathbb{F}^n[[z]]$, then $\mathcal{B}^\perp$ is a right shift invariant subspace of $\mathbb{F}^n[z]$, and conversely, if $\mathcal{C}$ is a right shift invariant subspace of $\mathbb{F}^n[z]$, then $\mathcal{C}^\perp$ is a left shift invariant subspace of $\mathbb{F}^n[[z]]$.

A behavior $\mathcal{B}$ is said to have a *kernel representation* if there exists a polynomial matrix $P(s)$ such that

$$\mathcal{B} \;=\; \{w \in \mathbb{F}^n[[z]] \mid P(\sigma)w = 0\}.$$

One may think of the rows of the matrix $P(s)$ as representing the 'laws' governing the behavior $\mathcal{B}$. Not every behavior has a kernel representation, and a characterization of the behaviors that do have such a representation is in fact one of the main results in the behavioral theory. To state the result one needs the following definition [39], in which $|_N$ denotes restriction to $0, \ldots, N$.

Definition 2.4: A behavior $\mathcal{B}$ is said to be *complete* if $w \in \mathbb{F}^n[[z]]$ belongs to $\mathcal{B}$ whenever $w|_N$ belongs to $\mathcal{B}|_N$ for all N.

In other words, a behavior $\mathcal{B}$ is complete if membership can be decided on the basis of finite windows. The result referred to above is the following.

Theorem 2.5: [39, Theorem 5] A behavior $\mathcal{B}$ has a kernel representation if and only if it is complete.

An extension of this important result to nD systems was given by Oberst in [23, p. 62].

It is a classical result from algebra that codes as we defined them above always have image representations, in the sense that for each code $\mathcal{C}$ there exists a polynomial matrix $G(s)$ such that

$$\mathcal{C} \;=\; \{v \in \mathbb{F}^n[z] \mid \exists \ell(z) \in \mathbb{F}^k[z] : v(z) = G(z)\ell(z)\}.$$

Obviously the matrix $G(s)$ can be interpreted as an encoder. The result from algebra that is used here is the fact that the free module $\mathbb{F}^n[z]$ is Noetherian [13, Thm. VI.2.1], which means that every submodule is finitely generated; note here that the definition of a code as given above might be rephrased by saying that a code is a submodule of $\mathbb{F}^n[z]$. The same theory also shows that a generator matrix may always be chosen to have full column rank, and that two polynomial matrices $G(s)$ and $G'(s)$ of full column rank generate the same code if and only if there exists a unimodular matrix $U(s)$ (i.e. a polynomial matrix with constant nonzero determinant) such that $G'(s) = G(s)U(s)$. Note that in this setting a finer structure is obtained than in the usual setting (see for instance [26]) in which a convolutional code is understood as a subspace of the space of rational vectors, and two generator matrices are equivalent whenever they are related by a nonsingular rational transformation.

The following theorem formally establishes that there is a duality relation (in Macaulay's sense) between *codes* on the one hand and *complete behaviors* on the other. The analogous result for sequences over $\mathbb{Z}$ rather than $\mathbb{Z}_+$ has been given by Nieuwenhuis and Willems [22, Prop. 2] and the 2D case was discussed by Valcher and Fornasini [37]. A proof of the theorem below is provided in the appendix. This proof is more algebraic in nature than the ones given in [22] and [37], which depend on functional-analytic methods, and should therefore be more amenable to generalization to cases where $\mathbb{F}$ is not a field.

Theorem 2.6: If $\mathcal{C} \subseteq \mathbb{F}^n[z]$ is a convolutional code with generator matrix $G(s)$, then $\mathcal{C}^\perp$ is a linear, left shift invariant and complete behavior with kernel representation $P(s) = G^t(s)$. Conversely, if $\mathcal{B} \subseteq \mathbb{F}^n[[z]]$ is a linear, left shift invariant and complete behavior with kernel representation $P(s)$, then $\mathcal{B}^\perp$ is a convolutional code with generator matrix $G(s) = P^t(s)$.

Dual codes were used already in the early seventies by Forney [4], [5]. Working with $\mathbb{Z}$ as a time axis and using sequences with left-bounded support, Forney looked at codes as subspaces of finite-dimensional vector spaces over the field $\mathbb{F}(s)$ of rational functions and so was able to use the duality theory of finite-dimensional vector spaces. This context has a symmetry in it in the sense that the annihilator of a code is then again an object of the same type (i.e. a shift invariant linear space of sequences on $\mathbb{Z}$ with left-bounded support); below we shall emphasize a point of view in which codes and their annihilators are regarded as objects of a different nature. In extensive work, Oberst [23] has developed a duality theory for linear behaviors on an abstract level. He defines behaviors by kernel representations and shows that these are dual in a natural way to certain quotient modules. In the case of discrete-time systems, the duals can be formed by taking quotients with respect to the Macaulay annihilator of the given behavior. The framework used by Oberst is general enough however to include also continuous-time systems, and remains even meaningful in some cases in which there is no time axis in the usual sense of the word, such as the one that is covered by the classical Pontryagin duality in which the 'signals' are elements of the circle group. The theorem above can be constructed as a special case of Oberst's results. Theorem 2.6 can also be deduced from the main result by Kaplan [11] (compare also with [7, Theorem 2.2]). Our proof however is elementary and does not rely on either [11] or [23].

B. Controllability and observability

In this subsection we discuss the duality between controllability and observability. If codes and behaviors are viewed as duals, then the dualization of a notion of controllability for behaviors is expected to lead to a notion of observability for codes. We shall work mostly over $\mathbb{Z}$ now since we rely on a number of definitions and results from the literature which have been stated for that case;

the analogous theory over $\mathbb{Z}_+$ does not seem to be equally well-developed.

The following notion of controllability for behaviors is due to Willems [40]. For a sequence $w = \sum_{-\infty}^{\infty} w_i z^i$, we use the symbol w^+ to denote the 'right half' $\sum_{0}^{\infty} w_i z^i$ and the symbol w^- to denote the 'left half' $\sum_{-\infty}^{0} w_i z^i$.

Definition 2.7: A behavior $\mathcal{B}$ defined on $\mathbb{Z}$ is said to be *controllable* if for every w and w' in $\mathcal{B}$ there exists a $w'' \in \mathcal{B}$ and integers N, M such that $(z^N w'')^- = w^-$ and $(z^M w'')^+ = w'^+$.

In words, the above definition says that a left part of a trajectory and a right part of another trajectory may always be connected via some intermediate string to form a new trajectory in $\mathcal{B}$.

Willems does not give a definition for *observability* of behaviors, although he does define such a notion for certain *representations*. If we consider what kind of controllability concept might be defined for codes, it may be noted that the definition above can be made to apply for codes as well (by embedding $\mathbb{F}^n[z]$ in $\mathbb{F}^n[[z]]$). The result is easily seen to be trivial: codes are always controllable in the above sense. This may be viewed as a mirroring, through duality, of the lack of an observability concept for behaviors.

The quickest way to find the dualization of controllability is to use the algebraic characterization of controllability in terms of kernel representations, as given by Willems [40, Prop. 4.3]. It should be noted that kernel representations for behaviors over $\mathbb{Z}$ may be given by matrices whose entries are polynomial in s and s^{-1}. A polynomial matrix $P(s, s^{-1})$ of size $k \times n$, with $k \leq n$, is said to be *left prime* over the ring $\mathbb{F}[s, s^{-1}]$ if its $k \times k$ minors are not all zero and have no nontrivial common factors (where factors of the form s^k, $k \in \mathbb{Z}$, are counted as trivial). Right primeness is defined analogously; obviously $P(s, s^{-1})$ is left prime if and only if $P^t(s, s^{-1})$ is right prime.

Proposition 2.8: A complete behavior is controllable if and only if it has a left prime kernel representation.

This would suggest to define a code to be observable if it has a right prime generator matrix. Right prime encoders are well-known (see for instance [21]) as *non-catastrophic encoders*; see also the discussion in [26, Ch. 2]). To define observability of codes in this way would however not be in the true behavioral spirit since the definition would then rely on a particular representation. Fornasini and Valcher [3] have recently presented a number of equivalent characterizations of observability which avoid this and therefore could be used as behavioral definitions. Rephrasing their results for the 1D case, observability can for instance be defined as follows.

Definition 2.9: (cf. [3, Prop. 2.1]) A code $\mathcal{C}$ is *observable* if there exists an integer N such that, whenever the supports of v and v' are separated by a distance of at least N and $v + v' \in \mathcal{C}$, then also $v \in \mathcal{C}$ and $v' \in \mathcal{C}$.

In other words, observability means that one can be sure that a message has been completed once a sufficiently long string of zeros has been received. An important property of observable codes is that they allow kernel representations, in the sense that there exists a polynomial matrix $H(s)$ (a *syndrome former*) with the property that $v \in \mathcal{C}$ if and only if $H(z)v(z) = 0$; this is the dual of the fact that controllable behaviors have an image representation [41, Prop. 4.3]. For the case in which the time axis is $\mathbb{Z}_+$, observability can be defined in the same way as above. Codes on $\mathbb{Z}_+$ are naturally associated with matrices over $\mathbb{F}[s]$ rather than $\mathbb{F}[s, s^{-1}]$, and the standard concept of left or right primeness for polynomial matrices requires that the greatest common divisor of the appropriate minors should be a constant. The characterization that we find for observability is however the same as in the $\mathbb{Z}$ case.

Proposition 2.10: A code $\mathcal{C}$ (on $\mathbb{Z}_+$) is observable if and only if it has a generator matrix $G(s)$ that is right prime when considered as a matrix over $\mathbb{F}[s, s^{-1}]$.

The proof of this proposition is in the appendix. The proof also shows that membership of an observable code on $\mathbb{Z}_+$ cannot in general be decided by a syndrome former alone; an additional finite test is needed. However if the *predictable delay* property [26, p. 44] is added, then this additional test can be dispensed with and one has a so-called *basic* encoder [26, p. 53]. It follows from a result by Massey and Sain [21] that the property in the above proposition is equivalent to the existence of a feed-forward inverse with delay.

C. Completion

On $\mathbb{F}^n[[z]]$ one can introduce the topology of pointwise convergence, with the understanding that on $\mathbb{F}^n$ the discrete topology is used (i. e. the topology induced by the Hamming distance). As noted by Willems [39], a behavior is complete if and only if it is closed in this topology. For a code $\mathcal{C}$ on $\mathbb{Z}_+$, we denote by $\bar{\mathcal{C}}$ its *completion*, i. e. its closure with respect to the topology of pointwise convergence. More explicitly, we have

$$\bar{\mathcal{C}} \;=\; \{v \in \mathbb{F}^n[[z]] \mid v|_N \in \mathcal{C}|_N \text{ for all } N\}. \tag{2.4}$$

It follows from the work of Staiger [35] that the completion of a code can be given in terms of the generator matrix as follows.

Proposition 2.11: The completion of a code $\mathcal{C}$ with generator matrix $G(s)$ of size $n \times k$ is given by

$$\bar{\mathcal{C}} \;=\; \{v(z) \in \mathbb{F}^n[[z]] \mid \exists \ell(z) \in \mathbb{F}^k[[z]] : v(z) = G(z)\ell(z)\}. \tag{2.5}$$

The suggestion presents itself to call any subset of $\mathbb{F}^n[[z]]$ that arises in this way an *infinite-input convolutional code*. This would again not be a definition in behavioral style. Actually already before the behavioral theory was developed within systems theory, an intrinsic definition of infinite-input convolutional codes was found by Staiger [35]. In this work Staiger uses the following definition.

Definition 2.12: Let $A \subseteq \mathbb{F}^n[[z]]$. A sequence $v \in A$ is said to be *remergable with respect to* A if for every $t \in \mathbb{Z}_+$ there exists a sequence of vectors $\{v'_{t+1}, v'_{t+2}, \cdots, v'_{t+N}\}$ such that

$$(v_0, v_1, v_2, \ldots, v_t, v'_{t+1}, v'_{t+2}, \cdots, v'_{t+N}, 0, 0, \ldots) \in A.$$

The set $A \subseteq \mathbb{F}^n[[z]]$ is said to be *remergable* if every element in A is remergable with respect to A.
One easily sees that remergability, under the extra condition of right shift invariance, is equivalent to Willems' notion of controllability as given in Def. 2.7 (with the obvious modifications to cover the case of trajectories on $\mathbb{Z}_+$). Staiger now proves the following result.

Theorem 2.13: [35] A right shift invariant subspace $\bar{\mathcal{C}}$ of $\mathbb{F}^n[[z]]$ is of the form (2.5) if and only if it is closed (in the topology of pointwise convergence) and remergable.
One should compare this to the results by Willems (see for instance [41, Prop. 4.3]) about the relation between controllability and the existence of image representations for behaviors. Staiger's result suggests to *define* an infinite-input linear convolutional code as a remergable closed right shift invariant subspace of $\mathbb{F}^n[[z]]$.

For codes over $\mathbb{Z}$ rather than over $\mathbb{Z}_+$, the completion leads to a shift invariant subspace of $\mathbb{F}^n[[z, z^{-1}]] \cong (\mathbb{F}^n)^{\mathbb{Z}}$; so in this case the completion is a behavior. For such codes there are two ways to relate a behavior to a code, namely by duality and by completion.

III. First-order representations of convolutional codes

One of the advantages of having a duality relation between codes and behaviors is that it becomes possible to transfer the whole theory of representations and transformations from the context of behaviors to the context of codes. See in particular the book by Kuijper [12] for the most comprehensive account so far of the first-order representation theory for linear behaviors. In this section we provide a few examples of representation results, and in the next section we shall use first-order representations for the construction of convolutional codes. The first result below states that each code has a generalized first-order representation. A code is said to have rate $\frac{k}{n}$ if its full rank generator matrices have size $n \times k$, and the *complexity* of a code is the highest degree of the full size minors of any full rank generator matrix (we skip here the behavioral definitions of these terms). In the following, the term *pencil* will refer to linear polynomials or equations in one variable with matrix coefficients.

Theorem 3.1 (Realization Theorem I, Existence)
Assume $\mathcal{C} \subseteq \mathbb{F}^n[z]$ is a rate $\frac{k}{n}$ convolutional code of complexity c. Then there exist $(c+n-k) \times c$ matrices K, L and a $(c+n-k) \times n$ matrix M (all defined over $\mathbb{F}$) such that the code $\mathcal{C}$ is described by

$$\begin{aligned}\mathcal{C} = \{v(z) \in \mathbb{F}^n[z] \mid \exists x(z) \in \mathbb{F}^c[z] : \\ zKx(z) + Lx(z) + Mv(z) = 0\}. \end{aligned} \tag{3.1}$$

Moreover the following minimality conditions will be satisfied:

1. K has full column rank
2. $[K \;\; M]$ has full row rank
3. $[zK + L \mid M]$ is left prime.

Proof: Write $\mathcal{C} = \mathcal{B}^\perp$, where $\mathcal{B}$ is a linear time-invariant complete behavior. By the representation theory of linear complete behaviors (cf. [12, Ch. 5]), we know that $\mathcal{B}$ can be represented in the pencil form $\sigma G\zeta = F\zeta$, $w = H\zeta$. So v belongs to $\mathcal{C}$ if and only if $H^t v$ belongs to $(\ker(\sigma G - F))^\perp = \mathrm{im}(zG^t - F^t)$, or in other words, if there exists an $x \in \mathbb{F}[z]$ such that $zG^t x - F^t x - H^t v = 0$. Now define $K = G^t$, $L = -F^t$, $M = -H^t$. The minimality properties follow immediately from the corresponding properties for pencil representations of behaviors [12, Thm. 4.3]. ∎

One also has the following property.

Lemma 3.2: Suppose that a code $\mathcal{C}$ with generator matrix $G(z)$ of full column rank is represented by (3.1), where (K, L, M) is a minimal triple. Then for all $z_0 \in \overline{\mathbb{F}}$, $G(z_0)$ has full column rank if and only if $z_0 K + L$ has full column rank.

Proof: See [29, Thm. 11]. ∎

Lemma 3.2 implies that a generator matrix $G(z)$ is right prime if and only if the pencil $zK + L$ is right prime over the ring $\mathbb{F}[z, z^{-1}]$. By Proposition 2.10 we therefore have:

Corollary 3.3: A code $\mathcal{C}$ represented by the minimal triple (K, L, M) is observable if and only if the pencil $zK + L$ is right prime when considered as a matrix over $\mathbb{F}[z, z^{-1}]$.

The next result describes the extent to which minimal first-order realizations are unique. The proof is obtained by dualizing [12, Thm. 4.34].

Theorem 3.4 (Realization Theorem II, Uniqueness)
The matrices K, L, M that were introduced in Theorem 3.1 are unique in the following way: if $(\tilde{K}, \tilde{L}, \tilde{M})$ is a second triple of matrices describing the code $\mathcal{C}$ through

$$\mathcal{C} = \{v(z) \mid \exists x(z) \in \mathbb{F}^c[z] : z\tilde{K}x(z) + \tilde{L}x(z) + \tilde{M}v(z) = 0\}$$

and if $(\tilde{K}, \tilde{L}, \tilde{M})$ satisfies the minimality conditions of Theorem 3.1 then there exist unique invertible matrices T and S such that

$$(\tilde{K}, \tilde{L}, \tilde{M}) = (TKS^{-1}, TLS^{-1}, TM). \tag{3.2}$$

The set of triples (K, L, M) satisfying the minimality conditions of Thm. 3.1, modulo the equivalence action of (3.2), can be studied from a geometrical viewpoint. In particular it has been shown that the categorical quotient forms a projective variety and we refer to [27], [28] for details.

As a consequence of Theorem 3.1 and 3.4 we can work either with minimal generalized first-order representations of a given size or with polynomial encoder matrices of a fixed rate and a fixed complexity. Algorithms are available to go from one description to the other. If a code $\mathcal{C}$ is described by a triple of matrices (K, L, M) which satisfies the minimality conditions 1 and 2 of Theorem 3.1 one can compute a generator matrix $G(z)$ through the computation of a minimal basis of $\ker[zK + L \;\; | \;\; M]$. The converse transformation can be done by an algorithm that we briefly outline below (cf. [29], [31] for more details).

Assume $G(z)$ has column indices $\nu_1 \geq \cdots \geq \nu_k$ and complexity $c := \sum_{i=0}^{k} \nu_i$. Let

$$\begin{aligned} X(z) &= \mathrm{diag}(X_1(z), \ldots X_k(z)), \\ X_i(z) &= [1 \;\; z \;\; \cdots \;\; z^{\nu_i - 1}]^t \quad (i = 1, \ldots, k). \end{aligned} \tag{3.3}$$

The matrix $X(z)$ has dimensions $c \times k$, $X(z)$ is right prime and has the property that for every polynomial vector

$$f(z) = (f_1(z), \ldots, f_k(z)) \in \mathbb{F}^k[z], \quad \deg f_i(z) \leq \nu_i - 1$$

there exists a unique vector $v \in \mathbb{F}^c$ such that $vX(z) = f(z)$. Because of those properties the matrix $X(z)$ was called in [29], [31] a 'basis matrix of size ν'.

Let $f(z) \in \mathbb{F}^k[z]$ with $\deg f_i(z) \leq \nu_i$ and let $[f(z)]$ denote the $c+k$ scalar vector obtained from $f(z)$ by identifying each $f_i(z)$ with the $1 \times (\nu_i + 1)$ row vector corresponding to the coefficients of $f_i(z)$. In this way we can view $f(z)$ as a vector in $\mathbb{F}^{c+k}$. Now consider the map

$$\begin{array}{rccc} \Phi: & \mathbb{F}^{2c+n} & \longrightarrow & \mathbb{F}^{c+k} \\ & v & \longmapsto & v \begin{bmatrix} zX(z) \\ X(z) \\ G(z) \end{bmatrix}. \end{array} \tag{3.4}$$

Since $X(z)$ is of full rank one verifies that there are $c+n-k$ linearly independent constant vectors in the left kernel of this matrix, i.e. there is a full rank matrix $(K \mid L \mid M)$ of size $(c+n-k) \times (2c+n)$ such that $zKX(z) + LX(z) + MG(z) = 0$. This matrix corresponds to a minimal first-order representation of $G(z)$. We illustrate the procedure by a simple example.

Example 3.5: Consider the rate $\frac{k}{n} = \frac{2}{3}$ code over $\mathbb{F}_2$ given by the generator matrix

$$G(z) := \begin{pmatrix} z^2 & z+1 \\ z^2+z+1 & 1 \\ 1 & z \end{pmatrix}.$$

The column indices are given by $\nu := [2, 1]$, the complexity is $c = 3$ and a basis matrix is

$$X(z) := \begin{pmatrix} 1 & 0 \\ z & 0 \\ 0 & 1 \end{pmatrix}.$$

The scalar matrix corresponding to $[zX(z)^t, X(z)^t, G(z)^t]$ is given by

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

which describes the map $\Phi : \mathbb{F}^9 \longrightarrow \mathbb{F}^5$. The kernel of Φ is given by

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

hence,

$$K := \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad L := \begin{pmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix},$$

$$M := \begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

is a minimal first order representation.

IV. An algebraic construction of convolutional codes

A key problem in convolutional coding theory has been to find a method for effectively characterizing the free distance d_f of a given convolutional code. Very much related to this problem is the task of designing codes of a given rate and complexity with good free distance. At present, perhaps the most effective technique for doing this has been to make an exhaustive search of the class of codes determined by a fixed rate and complexity,and compute the free distance of encoders in this class, until one with maximal or *near* maximal free distance is found. Obviously, this technique has its limitations.

Several methods have been investigated for constructing convolutional codes. Perhaps the most popular technique is to relate the generators of a convolutional code to the generators of some corresponding cyclic or quasi cyclic code and show that the distance of the cyclic code is a lower bound for the free distance (see e.g., [9], [14], [19], [36]). One can also restrict to the class of rate $\frac{1}{n}$ convolutional codes over $\mathbb{F}_q$ and develop very effective techniques for code constructions in this setting [1], [10]. Yet another way is restrict the search for good codes to the sub-class of convolutional codes having a nontrivial automorphism group. This technique is thoroughly investigated in [26]. In this section we present an algebraic construction technique based on first-order representations of codes. This technique is more general than the above constructions. It is also very similar to existing block code constructions in that we make direct use of the *parity check* matrix for the convolutional code.

Consider a convolutional code $\mathcal{C} \subset \mathbb{F}^n[z]$ and let $H(z)$ be a syndrome former. If $v(z)$ is a codeword of degree at most γ then the weight of $v(z)$ can be characterized in the following way. If $v(z) = v_0 + v_1 z + \ldots + v_\gamma z^\gamma$ and $H(z) = H_0 + H_1 z + \ldots + H_m z^m$, then the relation $H(z)v(z) = 0$ is equivalently described by a so called 'sliding block matrix' of size $(m + \gamma + 2) \times (\gamma + 1)$, (see e.g. [15]). The minimal dependence relation of the sliding block matrix describes the minimal weight of the code words having degree at most γ and in this way one achieves a bound on the free distance. Unfortunately it does not seem to be easy to construct a parity check matrix $H(z)$ which results in a sliding block matrix that has a good distance property for all values of γ. This is certainly one reason why there have been no algebraic code constructions using this matrix.

Below we shall use first-order representations to construct a sliding block matrix that is more manageable for

code constructions. For this let

$$\Upsilon := \begin{pmatrix} L & & & \\ K & L & & \\ & K & \ddots & \\ & & \ddots & L \\ & & & K \end{pmatrix}, \quad v := \begin{pmatrix} v_0 \\ v_1 \\ \vdots \\ v_\gamma \end{pmatrix},$$

and

$$\Omega := \begin{pmatrix} M & & & & \\ & M & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & M \end{pmatrix}, \quad x := \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{\gamma-1} \end{pmatrix}.$$

Consider a convolutional code $\mathcal{C}$ represented by a first order description of the form (3.1). The (K, L, M) representation (3.1) is then equivalent to the linear constraint

$$\begin{pmatrix} \Upsilon & \Omega \end{pmatrix} \begin{pmatrix} x \\ v \end{pmatrix} = 0. \tag{4.1}$$

Let $\mathcal{C}_\gamma$ be the set of code words of degree at most γ, and let colsp Υ be the space spanned by the columns of Υ. Then $\mathcal{C}_\gamma = \{v \mid \Omega v \in \text{ colsp } \Upsilon\}$. For any matrix Ψ such that colsp $\Upsilon = \ker \Psi$ we get a representation that no longer involves the state vector:

$$\mathcal{C}_\gamma = \ker \Psi \Omega. \tag{4.2}$$

One particular way to carry out this elimination is as follows. Note that any triple (K, L, M) that satisfies minimality conditions 1 and 2 of Thm. 3.1 can be written, after a suitable similarity transformation and a permutation of the components of the code vector if needed, in the following form:

$$K = \begin{bmatrix} -I \\ 0 \end{bmatrix}, \quad L = \begin{bmatrix} A \\ C \end{bmatrix}, \quad M = \begin{bmatrix} 0 & B \\ -I & D \end{bmatrix}. \tag{4.3}$$

We shall indicate the partitioning of the code vector v_t in the above by $v_t = \left[\begin{smallmatrix} y_t \\ u_t \end{smallmatrix}\right]$.

Remark 4.1: In the partitioning of the matrices K, L, M we used matrices A, B, C, D. It is possible to describe the dynamics of the code words in $\mathcal{C}_\gamma$ by the linear input/state/output system

$$\begin{aligned} x_{t-1} = Ax_t + Bu_t,\ y_t = Cx_t + Du_t, \\ 0 \le t \le \gamma,\ x_\gamma = 0,\ x_{-1} = 0. \end{aligned} \tag{4.4}$$

The time evolution is from the 'future' to the 'past'; of course it would be possible to reverse the time axis in the description to get a more familiar-looking form. In either case the representation above is different from a state space representation of a convolutional code often considered in the coding literature. In the coding literature (see e.g. Massey and Sain [20, Theorem 1]), the image representation $v(z) = G(z)\ell(z)$ is usually described through state space equations where the *input* $\ell(z)$ *drives the output* $v(z)$. In contrast to this, system (4.3) is a state space description where k *components* $u(z)$ *of the codeword* $v(z)$ *drive the remaining* $n-k$ *components* $y(z)$ *of* $v(z)$. We would like to point out that the corresponding systematic encoder may not be a polynomial encoder, and to obtain a polynomial encoder, one would have to choose a non-systematic one.

Now we eliminate the state vector x. Our first step is to substitute the partitions defined by (4.3) into (4.1) and perform elementary row operations to obtain an equation of the form:

$$\left(\begin{array}{c|c} 0 & P \\ \hline -I & * \end{array}\right) \begin{pmatrix} x \\ v \end{pmatrix} = 0. \tag{4.5}$$

Then, after permuting columns, $Pv = 0$ can be expressed as:

$$\left(\begin{array}{c|cccc} \mathbf{0} & B & AB & A^2B & \cdots & A^\gamma B \\ \hline & D & CB & CAB & \cdots & CA^{\gamma-1}B \\ & & D & CB & \cdots & CA^{\gamma-2}B \\ \mathbf{-I} & & & \ddots & & \vdots \\ & & & & D & CB \\ & & & & & D \end{array}\right) \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_\gamma \\ \hline u_0 \\ u_1 \\ \vdots \\ u_\gamma \end{pmatrix} = 0, \tag{4.6}$$

where $\mathbf{I}$ is the $\gamma(n-k) \times \gamma(n-k)$ identity matrix and $\mathbf{0}$ is the $c \times \gamma(n-k)$ all zero matrix. Note that the matrix appearing above takes the place of the usual sliding block matrix; also note that the structure of this matrix is rather different from the sliding block form.

By making particular choices of the parameters A, B, C, and D we can now attempt to find convolutional codes with good distance properties. Here we propose the following. Let $c, n, k \in \mathbb{Z}_+$ with $n > k$. Let $r := \max\{n-k, k\}$ and $i := \lceil \frac{c}{n-k} \rceil$. Choose a primitive α of the field $\mathbb{F}_q$, where $q \ge cri$, and define

$$A := \begin{pmatrix} \alpha^r & 0 & \cdots & 0 \\ 0 & \alpha^{2r} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \alpha^{cr} \end{pmatrix},$$

$$B := \begin{pmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^{k-1} \\ 1 & \alpha^2 & \alpha^4 & \cdots & \alpha^{2(k-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \alpha^c & \alpha^{2c} & \cdots & \alpha^{c(k-1)} \end{pmatrix},$$

$$C := \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \alpha & \alpha^2 & \cdots & \alpha^c \\ \alpha^2 & \alpha^4 & \cdots & \alpha^{2c} \\ \vdots & \vdots & & \vdots \\ \alpha^{n-k-1} & \alpha^{2(n-k-1)} & \cdots & \alpha^{c(n-k-1)} \end{pmatrix},$$

and

$$D := \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \alpha & \alpha^2 & \cdots & \alpha^k \\ \vdots & \vdots & \vdots & \vdots \\ \alpha^{(n-k-1)} & \alpha^{2(n-k-1)} & \cdots & \alpha^{k(n-k-1)} \end{pmatrix}.$$

Lemma 4.2: The triple (K, L, M) defined by the matrices (A, B, C, D) and (4.3) satisfies the minimality conditions of Theorem 3.1. Moreover the convolutional code $\mathcal{C}$ defined in this way is observable.

Proof: First we will show that the triple K, L, M defined by the matrices (A, B, C, D) satisfies the minimality conditions 1-3 of Theorem 3.1.

Conditions 1 and 2 are readily verified. Condition 3 is satisfied if and only if $[zI-A \mid -B]$ is left prime and by the well known Hautus test [8] (which works over finite fields!) this is the case as soon as the 'controllability matrix'

$$R := [B \; AB \; \ldots \; A^{c-1}B]$$

has rank c. Since R is a Vandermonde matrix by construction this is the case and K, L, M is a minimal representation of a rate $\frac{k}{n}$ convolutional code $\mathcal{C}$ of complexity c.

It remains to be shown that $\mathcal{C}$ is observable. By Corollary 3.3, it is enough to show that that $zK + L$ is right prime, i.e. that $\begin{bmatrix} zI-A \\ -C \end{bmatrix}$ is right prime. Applying once more the Hautus test [8] with the 'observability matrix'

$$\mathcal{O} := \left[C^t \; (CA)^t \; \ldots \; (CA^{c-1})^t\right]^t$$

readily shows the claim. ■

Theorem 4.3: Let $\epsilon = \max\{n-2k+1, 0\}$. The code $\mathcal{C}$ as defined above has rate $\frac{k}{n}$, complexity c and free distance

$$d_f \geq c + 1 + \epsilon.$$

Remark 4.4: Although the following proof is technical, the idea is rather simple. Any code word $v(z)$ in the code defined by the matrices A, B, C and D has a well defined degree γ. If γ is small (less than ci), then our choice of A and B ensure that the weight of $v(z)$ is bigger than $c + 1$. If γ is large (bigger than ci), then our choice of A and C ensure that the weight of $v(z)$ is bigger than $c + 1$, (i.e. that $v(z)$ is not too *sparse*).

Proof: That the code has the specified rate and complexity can be directly determined from the sizes of the matrices used for its representation.

Let $v(z) = v_0 + v_1 z + \ldots + v_\gamma z^\gamma \in \mathcal{C}$, where $v_0 \neq 0, v_\gamma \neq 0$. As above we partition v as $v_t = \begin{pmatrix} y_t \\ u_t \end{pmatrix}$. Our aim is to show that the weight of $v(z)$ is at least $c + 1 + \epsilon$. For this we consider two cases: 1) $\gamma \leq ci - 1$, and 2) $\gamma > ci - 1$, where as above i denotes $\lceil \frac{c}{n-k} \rceil$.

Case 1. Suppose that $\gamma \leq ci-1$. The vector $(u_0, u_1, \cdots u_\gamma)^t$ belongs to the kernel of

$$R := [B \;\; AB \;\; \cdots \;\; A^{\gamma-1}B \;\; A^\gamma B] \tag{4.7}$$

By construction, R is a Vandermonde matrix of size $c \times (\gamma + 1)k$. Our choice of A and B implies that $(\gamma + 1)k \geq c$, therefore,wt $(u_0, u_1, \cdots u_\gamma) \geq c + 1$. Since $u_\gamma \neq 0$, it follows that $\begin{bmatrix} u_\gamma \\ y_\gamma \end{bmatrix}$ is in the rate $\frac{k}{n}$ block code with parity check matrix $[I \;\; -D]$. By our choice of D, this block code is an MDS code, which implies that wt $(y_\gamma) \geq \epsilon$; hence, wt $(v(z)) \geq c + 1 + \epsilon$.

Case 2. Suppose $\gamma > ci-1$. If wt $(u_{\gamma-ci+1}, \cdots, u_{\gamma-1}, u_\gamma) \geq c + 1$ we are done, so assume wt $(u_{\gamma-ci+1}, \cdots, u_{\gamma-1}, u_\gamma) = b < c + 1$. It now follows that there are at most b non-zero u_j's in the interval $\gamma \geq j \geq \gamma - ci + 1$, which implies that there are at least $c - b$ disjoint subsequences of length i of the form $u_t, u_{t+1}, \cdots u_{t+i-1}$ containing only zero vectors and lying completely in the interval $[\gamma - ci + 1, \gamma - 1]$. Let $u_t, u_{t+1}, \cdots u_{t+i-1}$ be one such subsequence and let $\mathbf{y}$ be the transpose of $(y_t, y_{t+1}, \ldots, y_{t+i-1})$. From (4.6), we obtain the following system of equations for the corresponding $\mathbf{y}$:

$$\mathbf{y} = \begin{pmatrix} CA^{i-1}B & CA^iB & \cdots & CA^{\gamma-t-1}B \\ CA^{i-2}B & CA^{i-1}B & \cdots & CA^{\gamma-t-2}B \\ \vdots & \vdots & \vdots & \vdots \\ CB & CAB & \cdots & CA^{\gamma-t-i}B \end{pmatrix} \begin{pmatrix} u_{t+i} \\ u_{t+i+1} \\ \vdots \\ u_{\gamma-1} \\ u_\gamma \end{pmatrix},$$

which is equivalent to

$$\mathbf{y} = \begin{pmatrix} CA^{i-1} \\ \vdots \\ CA \\ C \end{pmatrix} \left(Bu_{t+i} + ABu_{t+i+1} + \ldots + A^{\gamma-t-i}Bu_\gamma \right). \tag{4.8}$$

The expression $Bu_{t+i} + ABu_{t+i+1} + \cdots + A^{\gamma-t-i}Bu_\gamma$ must be nonzero, since if it were zero, this would imply that $(u_{t+i}, \cdots, u_\gamma, 0, \cdots, 0)'$ is in the kernel of R as defined in (4.7); hence it would follow that wt $((u_{t+i}, \cdots, u_\gamma)) \geq c+1$, which contradicts our assumptions. Since the observability matrix appearing in (4.8) is of size $i(n-k) \times c$ where $i(n-k) \geq c$, and of full rank, there must be at least one nonzero output y_j for $t \leq j \leq t + i - 1$. Since we have at least $c - b$ such subsequences, we must have at least $c - b$ nonzero outputs from $y_{\gamma-ci+1}$ to $y_{\gamma-1}$. As shown in case 1, we have wt $(y_\gamma) \geq \epsilon$, hence wt $((v_0, v_1, \cdots v_{ci-1})) \geq c-b+b+\epsilon$. Since $v_0 \neq 0$, we obtain wt $(v(z)) \geq c + 1 + \epsilon$. ■

Since the class of convolutional codes constructed in Thm. 4.3 is observable, their free distance does not change when one considers the codes completion $\bar{\mathcal{C}}$ as defined in (2.4). Therefore, Thm. 4.3 can been seen as a construction theorem for infinite input convolutional codes as well. In fact, one could start by discussing input/state/output representations for infinite input convolutional codes and their properties, then derive (4.6) from this viewpoint.

Example 4.5: Let $n = 3, k = 2$ and $c = 4$. Then $c^2k = 32$ so we can choose any $q \geq 32$. For simplicity, we choose $q = 37$. Next, let $\alpha = 2$, where 2 is a generator for the group of units of $\mathbb{F}_{37}$. The corresponding A, B, C and D

matrices are:

$$A = \begin{pmatrix} 4 & 0 & 0 & 0 \\ 0 & 16 & 0 & 0 \\ 0 & 0 & 27 & 0 \\ 0 & 0 & 0 & 34 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 2 \\ 1 & 4 \\ 1 & 8 \\ 1 & 16 \end{pmatrix},$$

$$C = \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix}, \quad D = \begin{pmatrix} 1 & 1 \end{pmatrix}.$$

A computation of a minimal basis of $\ker [zK + L \mid M]$ (compare with Section 3) results in a generator matrix

$$G(z) := \begin{pmatrix} 3\,z^2 + 4\,z + 9 & 2\,z^2 + 17\,z + 13 \\ 18\,z^2 + 26\,z & 26\,z + 1 \\ 29\,z^2 + 29\,z + 9 & 34\,z^2 + 14\,z + 14 \end{pmatrix}.$$

The designed distance is 5, however one can easily show that the actual distance is greater than or equal to 6.

Setting $q = p^m, m \in \mathbb{Z}_+$ one can construct subfield codes, i.e., codes over $\mathbb{F}_p$. This is done in a manner quite similar to the classical BCH construction. The main difference is that the parity check matrix (4.6) needs to be extended in a way that preserves the factorization. That this can be done, as well as the types of codes this technique yields, is presented in [32].

V. Conclusions

In this paper we studied convolutional codes from a module theoretic point of view and we related our framework to systems theory. We showed that the class of linear behaviors having a kernel representation can be considered as dual to the class of convolutional codes. In our development we stressed matrix representations of convolutional codes as opposed to the traditional graph representations.

Using such matrix representations we were able to represent the class of convolutional codes in ways not considered in the literature previously. In a final section we were able to derive an algebraic construction of convolutional codes where the resulting codes have free distance lower bounded by the complexity + 1.

VI. Appendix

In this section we provide proofs for a number of results in the main text.

Proof of Theorem 2.6: Let $\mathcal{C}$ be a convolutional code with generator matrix $G(z)$. An element $w \in \mathbb{F}^n[[z]]$ belongs to $\mathcal{C}^\perp$ if and only if $(w, G(z)\ell) = 0$ for all $\ell \in \mathbb{F}^k[z]$. This is equivalent to $(G^t(\sigma)w, \ell) = 0$ for all $\ell \in \mathbb{F}^k[z]$, which in turn is equivalent to $G^t(\sigma)w = 0$.

For the second part of the proof, let $\mathcal{B}$ be a linear, left shift invariant and complete behavior with kernel representation $P(s)$. Assume $P(s)$ is of size $k \times n$ and let $\mathcal{C}(P^t)$ be the convolutional code generated by $P^t(s)$. Take $v \in \mathcal{C}(P^t)$ so that $v = P^t(z)\ell$ for some $\ell \in \mathbb{F}^k[z]$. For any $w \in \mathcal{B}$, we then have $(w, v) = (w, P^t(z)\ell) = (P(\sigma)w, \ell) = 0$. So it follows that $\mathcal{C}(P^t) \subseteq \mathcal{B}^\perp$ and the rest of the proof will be devoted to the reverse inclusion.

First assume that the matrix $P(s)$ is left prime, i.e., there is a matrix $\hat{P}(s)$ such that

$$U(s) := \begin{bmatrix} P(s) \\ \hat{P}(s) \end{bmatrix}$$

is unimodular, i.e., has a polynomial inverse. Write

$$U^{-1}(s) =: [T(s) \mid \hat{T}(s)]$$

where the partitioning is conformable to that of $U(s)$. We claim that $P(\sigma)w = 0$ for $w \in \mathbb{F}^n[[z]]$ if and only $w = \hat{T}(\sigma)w'$ for some $w' \in \mathbb{F}^{n-k}[[z]]$. Indeed, if $w = \hat{T}(\sigma)w'$ then we can also write $w = [T(\sigma) \mid \hat{T}(\sigma)]\left[\begin{smallmatrix} 0 \\ w' \end{smallmatrix}\right]$ which implies $U(\sigma)w = \left[\begin{smallmatrix} 0 \\ w' \end{smallmatrix}\right]$ and so $P(\sigma)w = 0$. Conversely if $P(\sigma)w = 0$ then $w = \hat{T}(\sigma)w'$ for $w' = \hat{P}(\sigma)w$. Now take $v \in \mathcal{B}^\perp$. It follows that $(\hat{T}(\sigma)w', v) = 0$ for all $w' \in \mathbb{F}^{n-k}[[z]]$, so that $\hat{T}^t(z)v = 0$. Define v' by $v' = T^t(z)v$, then

$$v = [P^t(z) \mid \hat{P}^t(z)] \begin{bmatrix} T^t(z) \\ \hat{T}^t(z) \end{bmatrix} v = [P^t(z) \mid \hat{P}^t(z)] \begin{bmatrix} v' \\ 0 \end{bmatrix} = P^t(z)v'$$

so that $v \in \mathcal{C}(P^t)$.

Now consider a general kernel representation $P(s)$. We may assume without loss of generality that $P(s)$ has full row rank. We may then write $P(s) = T(s)Q(s)$ where $T(s)$ is a square and nonsingular polynomial matrix, and $Q(s)$ is of size $k \times n$ and left prime. (This follows by an application of the Smith form, which is valid over a general Euclidean domain and so in particular for matrices over $\mathbb{F}[s]$.) From the above it already follows that

$$\mathcal{C}(P^t) \subset \mathcal{B}^\perp \subset \mathcal{C}(Q^t).$$

To prove that actually $\mathcal{B}^\perp = \mathcal{C}(P^t)$, it suffices to show that the quotient spaces $\mathcal{C}(Q^t)/\mathcal{B}^\perp$ and $\mathcal{C}(Q^t)/\mathcal{C}(P^t)$ are both finite-dimensional vector spaces and that the dimensions of these spaces agree. This is what we shall do now.

As is well-known, the behavior $\mathcal{B}(T)$ determined by the nonsingular matrix $T(s)$ is a finite-dimensional vector space over $\mathbb{F}$ with dimension $r := \deg \det T(s)$. Also the mapping $Q(\sigma)$ from $\mathbb{F}^n[[z]]$ to $\mathbb{F}^k[[z]]$ is surjective so we can find elements $w_1, \ldots, w_r \in \mathbb{F}^n[[z]]$ such that the elements $\tilde{w}_i$ defined by $\tilde{w}_i = Q(\sigma)w_i$ form a basis for $\mathcal{B}(T)$. Then $\mathcal{B}(P)$ is spanned by $\mathcal{B}(Q)$ together with the elements w_i, and so $v \in \mathcal{B}^\perp$ for $v \in \mathbb{F}^n[z]$ if and only if $v \in \mathcal{C}(Q^t)$ and $(w_i, v) = 0$ for all $i = 1, \ldots, r$. To show that these extra restrictions are independent, assume that $(\sum_{i=1}^r \alpha_i w_i, v) = 0$ for some $\alpha_i \in \mathbb{F}$ and for all $v \in \mathcal{C}(Q^t)$. It then follows that for all $\ell \in \mathbb{F}^k[z]$ we have

$$(\sum_{i=1}^r \alpha_i \tilde{w}_i, \ell) = (\sum_{i=1}^r \alpha_i Q(\sigma)w_i, \ell) = (\sum_{i=1}^r \alpha_i w_i, Q^t(z)\ell) = 0$$

so that $\sum_{i=1}^{r} \alpha_i \tilde{w}_i = 0$ and hence all α_i are zero because the $\tilde{w}_i$ are independent. It follows that the quotient space $\mathcal{C}(Q^t)/\mathcal{B}^\perp$ is a finite-dimensional vector space with dimension $r = \deg \det T(s)$.

To complete the proof, we note that it is a standard fact from polynomial module theory that the quotient module $\mathcal{C}(Q^t)/\mathcal{C}(P^t)$ is finite-dimensional as a vector space over $\mathbb{F}$ with dimension given by $\deg \det T(s)$. ■

For the proof of Prop. 2.10, we need the following lemma which states that "nontrivial 1/1 codes are never observable".

Lemma 6.1: The 1/1 code generated by a scalar polynomial $p(z)$ is not observable unless $p(z) = p_m z^m$ for some $p_m \in \mathbb{F}$ and some $m \in \mathbb{Z}_+$.

Proof: Let $p(z)$ be of the form $p_m z^m + p_{m+1} z^{m+1} + \cdots$, where $p_m \neq 0$. Suppose that there exists an integer N as in the definition of observability. We can solve the equation $z^m = p(z)r(z) + z^{m+N} q(z)$ in the polynomial unknowns $r(z)$ and $q(z)$ by successively solving the equations

$$\begin{aligned} 1 &= p_m r_0 \\ 0 &= p_{m+1} r_0 + p_m r_1 \\ &\vdots \\ 0 &= p_{m+N-1} r_0 + \cdots + p_m r_{N-1}. \end{aligned}$$

It follows that $z^m - z^{m+N} q(z)$ belongs to the code $\mathcal{C}$; however z^m does not, unless $p(z)$ is of the form indicated in the statement of the lemma. ■

Proof of Proposition 2.10: Suppose first that the right primeness condition holds. Then, after post-multiplication by an $\mathbb{F}[z]$-unimodular matrix if necessary, we may assume that $G(z) = R(z)T(z)$ where $R(z)$ has a left polynomial inverse and $T(z)$ is a diagonal matrix with diagonal entries of the form z^j. We can find a matrix $R'(z)$ such that $[R'(z) \mid R(z)]$ is unimodular. Define $H(z)$ and $H'(z)$ by

$$\begin{bmatrix} H(z) \\ H'(z) \end{bmatrix} = [R'(z) \mid R(z)]^{-1}$$

so that in particular $H'(z)R(z) = I$ and $H(z)R(z) = 0$. A polynomial $w(z)$ belongs to $\mathcal{C}(G)$ (the code generated by G) if and only if $H(z)w(z) = 0$ and $H'(z)w(z) \in \mathcal{C}(T)$. Now let N be an integer that is larger than the degree of $H(z)$ and the degree of $H'(z)$, and suppose that $v + v' \in \mathcal{C}$, with supports separated by a distance of at least N. It then follows from $H(z)(v(z) + v'(z)) = 0$ that both $H(z)v(z) = 0$ and $H(z)v'(z) = 0$. Moreover, we must have either $H'(z)v(z) = 0$ or $H'(z)v'(z) = 0$, and in both cases it follows from $H'(z)(v(z)+v'(z)) \in \mathcal{C}(T)$ that $H'(z)v(z) \in \mathcal{C}(T)$ as well as $H'(z)v'(z) \in \mathcal{C}(T)$.

For the converse part of the proof, suppose now that the full size minors of $G(z)$ have a common divisor that is not of the form $p_m z^m$ for some m. We can then write $G(z) = R(z)T(z)$ where $T(z)$ is diagonal and at least one of the diagonal elements is not of the form $p_m z^m$. It then follows from the preceding lemma that $T(z)$ generates a non-observable code, so for all integers N there exist polynomial vectors $v(z)$ and $v'(z)$ whose supports are separated by a distance at least N and whose sum belongs to $\mathcal{C}(T)$, but that do not themselves belong to $\mathcal{C}(T)$. By considering $R(z)v(z)$ and $R(z)v'(z)$, one sees that the same property holds for $\mathcal{C}(G)$ (note that it follows from $v(z) \notin \mathcal{C}(T)$ that $R(z)v(z) \notin \mathcal{C}(G)$, because $R(z)$ has full column rank). ■

Acknowledgment

We would like to acknowledge the comments provided by the anonymous referees and in particular those of the guest editor Dr. G. David Forney. Their careful reading of earlier versions of this manuscript greatly improved the quality of the final version.

References

[1] K. A. S. Abdel-Ghaffar. Some convolutional codes whose free distances are maximal. *IEEE Trans. Inform. Theory*, IT-35(1):188–191, 1989.

[2] E. Fornasini and M.E. Valcher. Algebraic aspects of 2D convolutional codes. *IEEE Trans. Inform. Theory*, IT-40(4):1068–1082, 1994.

[3] E. Fornasini and M.E. Valcher. Observability and extendability of finite support nD behaviors. In *Proc. of the 34th IEEE Conference on Decision and Control*, pages 3277–3282, New Orleans, Louisiana, 1995.

[4] G. D. Forney. Convolutional codes I: Algebraic structure. *IEEE Trans. Inform. Theory*, IT-16(5):720–738, 1970.

[5] G. D. Forney. Structural analysis of convolutional codes via dual codes. *IEEE Trans. Inform. Theory*, IT-19(5):512–518, 1973.

[6] G. D. Forney and M. D. Trott. The dynamics of group codes: State space, trellis diagrams and canonical encoders. *IEEE Trans. Inform. Theory*, IT-39:1491–1513, 1993.

[7] G. D. Forney and M. D. Trott. Controllability, observability, and duality in behavioral group systems. In *Proc. of the 34th IEEE Conference on Decision and Control*, pages 3259–3264, New Orleans, Louisiana, 1995.

[8] M. L. J. Hautus. Controllability and observability condition for linear autonomous systems. *Ned. Akad. Wetenschappen, Proc. Ser. A*, 72:443–448, 1969.

[9] J. Justesen. New convolutional code constructions and a class of asymptotically good time-varying codes. *IEEE Trans. Inform. Theory*, IT-19(2):220–225, 1973.

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

[11] S. Kaplan. Extension of the pontryagin duality I: Infinite products. *Duke Math J.*, 15:649–658, 1948.

[12] M. Kuijper. *First-Order Representations of Linear Systems*. Birkhäuser, Boston, 1994.

[13] S. Lang. *Algebra*. Addison-Wesley, Reading, Mass., 1965.

[14] Y. Levy and D.J. Costello Jr. An algebraic approach to constructing convolutional codes from quasi-cyclic codes. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 14:189–198, 1993.

[15] S. Lin and D. Costello. *Error Control Coding: Fundamentals and Applications*. Prentice-Hall, Englewood Cliffs, NJ, 1983.

[16] H. A. Loeliger, G. D. Forney, T. Mittelholzer, and M. D. Trott. Minimality and observability of group systems. *Linear Algebra Appl.*, 205/206:937–963, 1994.

[17] H. A. Loeliger and T. Mittelholzer. Convolution codes over groups. Preprint, April 1992.

[18] F.S. Macaulay. *The Algebraic Theory of Modular Systems*. Cambridge University Press, Cambridge, 1916.

[19] J. L. Massey, D.J. Costello, and J. Justesen. Polynomial weights and code constructions. *IEEE Trans. Inform. Theory*, IT-19(1):101–110, 1973.

[20] J. L. Massey and M. K. Sain. Codes, automata, and continuous systems: Explicit interconnections. *IEEE Trans. Automat. Contr.*, AC-12(6):644–650, 1967.

[21] J. L. Massey and M. K. Sain. Inverse of linear sequential circuits. *IEEE Trans. on Computers*, C-17(4):330–337, 1968.

[22] J.W. Nieuwenhuis and J. C. Willems. Duality for linear time invariant finite dimensional systems. In J.L. Lions A. Bensoussan, editor, *Analysis and Optimization of Systems*, Lect. Notes Contr. Inf. Sci. 111, pages 13–21. Springer, Berlin, 1988.

[23] U. Oberst. Multidimensional constant linear systems. *Acta Appl. Math*, 20:1–175, 1990.

[24] J. K. Omura. On the Viterbi decoding algorithm. *IEEE Trans. Inform. Theory*, IT-15:177–179, 1969.

[25] J. K. Omura. Optimal receiver design for convolutional codes and channels with memory via control theoretical concepts. *Information Sciences*, 3:243–266, 1971.

[26] Ph. Piret. *Convolutional Codes, an Algebraic Approach.* MIT Press, Cambridge, MA, 1988.

[27] M. S. Ravi and J. Rosenthal. A smooth compactification of the space of transfer functions with fixed McMillan degree. *Acta Appl. Math*, 34:329–352, 1994.

[28] M. S. Ravi and J. Rosenthal. A general realization theory for higher order linear differential equations. *Systems & Control Letters*, 25(5):351–360, 1995.

[29] M. S. Ravi, J. Rosenthal, and J. M. Schumacher. A realization theory for homogeneous AR-systems, an algorithmic approach. In *Proc. IFAC Conference on System Structure and Control*, pages 183–188, Nantes, France, 1995.

[30] M.P.M Rocha. *Structure and Representation of 2D Systems.* PhD thesis, University of Groningen, 1990.

[31] J. Rosenthal and J. M. Schumacher. Realization by inspection. CWI Report BS-R9534, December 1995. Submitted to *IEEE Trans. Automat. Contr.*

[32] J. Rosenthal and E.V. York. BCH convolutional codes. In Preparation.

[33] M.K. Sain and J.L. Massey. Invertibility of linear time-invariant dynamical systems. *IEEE Trans. Automat. Contr.*, AC-14:141–149, 1969.

[34] L. Staiger. Towards the structure of convolutional codes. *Wiss. Ztschr. Friedrich-Schiller-Univ. Jena, Math.-Nat. R.*, 31:647–650, 1982.

[35] L. Staiger. Subspaces of $GF(q)^\omega$ and convolutional codes. *Information and Control*, 59:148–183, 1983.

[36] R.M. Tanner. Convolutional codes from quasi-cyclic codes: A link between the theories of block and convolutional codes. Computer Research Laboratory, Technical Report, USC-CRL-87-21, November 1987.

[37] M.E. Valcher and E. Fornasini. On 2D finite support convolutional codes: an algebraic approach. *Multidim. Sys. and Sign. Proc.*, 5:231–243, 1994.

[38] J. C. Willems. System theoretic models for the analysis of physical systems. *Ricerche di Automatica*, 10:71–106, 1979.

[39] J. C. Willems. From time series to linear system. Part I: Finite dimensional linear time invariant systems. *Automatica*, 22:561–580, 1986.

[40] J. C. Willems. Models for dynamics. *Dynamics Reported*, 2:171–269, 1989.

[41] J. C. Willems. Paradigms and puzzles in the theory of dynamical systems. *IEEE Trans. Automat. Control*, AC-36(3):259–294, 1991.

[42] E.V. York, J. Rosenthal, and J. M. Schumacher. On the relationship between algebraic systems theory and coding theory: Representations of codes. In *Proc. of the 34th IEEE Conference on Decision and Control*, pages 3271–3276, New Orleans, Louisiana, 1995.