

Families of Convolutional Codes over Finite Fields: A Survey

M. Isabel García-Planas^(✉)

Dept. de Matemàtiques, Universitat Politècnica de Catalunya, Barcelona, Spain
`maria.isabel.garcia@upc.edu`

Abstract. The goal of this work is to give explicit interconnections between control theory and coding. It is well-known the existence of a closed relation between linear systems over finite fields and convolutional codes that allow to understand some properties of convolutional codes and to construct them. The connection between convolutional codes and linear systems permit to consider control as well as analyze observability of convolutional codes under linear systems point of view.

An accurate look at the algebraic structure of convolutional codes using techniques of linear systems theory as well a study of input-state-output representation control systems. A particular property considered in control systems theory called output-controllability property is analyzed and used for solve the decoding process of this kind of codes.

1 Introduction

At the origin, coding theory has been devoted mainly to information theory. In coding theory had, in fact, emerged from the need for better communication and better computer data storage. Concretely, convolutional codes are used on many occasions to transfer data with high demands on speed. To this end, we require potent codes of high rates. These codes are frequently implemented in composite with a hard-decision code, particularly Reed Solomon. Before turbo codes, such constructions were the most efficient, coming closest to the Shannon limit.

The convolutional codes are an alternative to the block codes because of their simplicity of generation with a little shift register. The main difference between them is the introduction of the concept of memory, that is, the coding at any given time will not depend only on the word to be coded, also on those previously coded. These codes have a great advantage over those of blocks in channels with high noise (high probability of error). Wireless communications or satellite communications stand out among their uses.

Convolutional codes were introduced by Elias [3] which suggests using a polynomial matrix $G(z)$ in the encoding process and allow the generation of the code line without using a previous buffer. G.D. Forney in [4] explained that the term “convolutional” is used because the output sequences can be regarded as the convolution of the input sequence with the sequences in the encoder.

There is a considerable amount of literature on the theory of convolutional codes over finite fields, see [1, 3, 5, 9–11, 13–17] or [21], for example. In particular,

in [16] the author find an overview of the different approaches to the subject of convolutional code. In this work we use the definition of convolutional code as submodule of $\mathbb{F}[z]^n$ being interesting in order to obtain a realization as linear system. First order and input-state-output representations can be found in [18, 19, 22, 23].

2 Convolutional Codes over Finite Fields

Let \mathbb{F}_q be the finite field of $q = p^r$ elements, the set of the input alphabet channel. In the sequel, and if the confusion is not possible, we denote \mathbb{F}_q simply as \mathbb{F} .

Definition 1. *A rate (n, k) convolutional code \mathcal{C} , over a finite field \mathbb{F} is a finitely generated $\mathbb{F}[z]$ -submodule of $\mathbb{F}^n[z]$ of rank k .*

A convolutional code \mathcal{C} can be expressed in a matrix form (called generator matrix) as follows.

$$\begin{aligned} G(z) : \mathbb{F}[z]^\ell &\longrightarrow \mathbb{F}[z]^n \\ u(z) &\longrightarrow v(z) = G(z)u(z) \end{aligned}$$

of order $n \times \ell$, $\ell \geq k$, whose columns collect a system of generators of the finitely generated submodule representing the code, that is to say $\mathcal{C} = \text{Im } G(z)$.

Note that $\mathbb{F}[z]$ is a principal ideal domain and then a convolutional code \mathcal{C} has a well-defined rank k and there exists a full-rank matrix $G(z)$ (of rank k) such that $\mathcal{C} = \text{colsp}_{\mathbb{F}[z]} G(z)$.

So, it is possible to refine the definition of generator matrix considering the notion of *encoder*, (see [23], for more details).

Definition 2. *An encoder to \mathcal{C} is a matrix*

$$\begin{aligned} G(z) : \mathbb{F}[z]^k &\longrightarrow \mathbb{F}[z]^n \\ u(z) &\longrightarrow v(z) = G(z)u(z) \end{aligned}$$

such that $\text{Im } G(z) = \mathcal{C}$ and $G(z)$ is injective.

If we assume that $G(z)$ is a $n \times k$ matrix with entries in $\mathbb{F}[z]$, the set

$$\mathcal{C} = \{v(z) \in \mathbb{F}^n[z] \mid \exists u(z) \in \mathbb{F}^k[z] \text{ such that } v(z) = G(z)u(z)\}$$

defines a submodule of $\mathbb{F}^n[z]$. Note that $\text{Im } (G)$ is a finitely generated submodule.

The above definition implies that a $n \times k$ polynomial matrix is an encoder of \mathcal{C} if its columns form a basis of the free module \mathcal{C} . In particular, an encoder is a generator matrix which $l = k$ and $G(z)$ is injective.

We denote by ν_i the maximum of all degrees of each of the polynomials of each column and we can assume that $\nu_1 \geq \nu_2 \geq \dots \geq \nu_k$ up to realignment. The number ν_1 is called the memory of the code and the collection of numbers ν_i are known as Forney's indices.

Remember that in convolutional codes, the coding of a word varies according to the words transmitted previously. And just the memory of the code ν_1 corresponds to the number of previous words on which the encoding depends. Notice that if $\nu_1 = 0$ the convolutional code is a block code.

Moreover, there exists another parameter related with convolutional codes and their encoders; that is, the complexity of both objects. The relation between these complexities is the key of the definition of a minimal encoder.

Definition 3. (a) *The complexity of the encoder (also called constraint length)*

$$\text{is } c = \sum_{i=0}^k \nu_i.$$

(b) *The degree or complexity of a convolutional code \mathcal{C} is the highest degree of the full size minors of any encoder, and it is denoted by $\delta(\mathcal{C})$.*

We ask if these two numbers ever coincide, the answer is “in general no”, and for the case where they coincide we have the following definition.

Definition 4. *Let $\mathcal{C} \subset \mathbb{F}[z]^n$ be a (n, k) -convolutional code. An encoder matrix $G(z)$ of \mathcal{C} is called minimal if and only if the complexity of the encoder coincides with the complexity of the code. That is to say $c = \delta(\mathcal{C})$*

It is well known that if we apply a basis change in $\mathbb{F}[z]^k$, it does not change the path of the map $G(z)$. Then, we have the following results relating minimal encoders:

Lemma 1. *Let $G(z)$ be an $n \times k$ polynomial matrix of rank k defining a convolutional code $\mathcal{C} = \text{colsp}_{\mathbb{F}[z]} G(z)$. Let $\hat{G}(z)$ be an $n \times k$ polynomial matrix of rank k over $\mathbb{F}[z]$. The following statements are verified:*

1. *$G(z)$ and $\hat{G}(z)$ define the same behaviour if and only if there exists a $k \times k$ unimodular matrix $U(z)$ such that $\hat{G}(z) = G(z)U(z)$*
2. *There exists an unimodular matrix $U(z)$ such that $\hat{G}(z) = G(z)U(z)$ is a minimal encoder.*
3. *If $G(z)$ and $\hat{G}(z)$ are minimal encoders of \mathcal{C} then they have the same column degrees.*

Definition 5. *The column degrees $(\kappa_1, \dots, \kappa_k)$ of any minimal encoder $\hat{G}(z)$ of \mathcal{C} are known as the Kronecker or controllability indices of the code. We can reorder them if it is necessary such that $\kappa_1 \geq \dots \geq \kappa_k$. The invariant $\delta = \sum_{i=1}^k \kappa_i$ is the degree of complexity of the code \mathcal{C} .*

(In some coding literature, δ is called the complexity of the code).

Note that the controllability indices of a convolutional code are unique and invariant of the code. If we consider a minimal encoder of a convolutional code then the controllability indices and Forney's indices are equal, and in this case, $\kappa_1 = \nu_1$ is the memory of the encoder.

We give some notions about observable convolutional codes that are useful in the following Chapter.

Definition 6. Let $G(z)$ be an encoder of a (n, k) convolutional code \mathcal{C} over \mathbb{F} . A syndrome former for the code \mathcal{C} is a homomorphism of modules given by

$$\psi : \mathbb{F}[z]^n \rightarrow \mathbb{F}[z]^{n-k}$$

with the property that $\text{Im } G(z) \subseteq \text{Ker } \psi$.

Definition 7. Let $G(z)$ be an encoder of a (n, k) convolutional code \mathcal{C} over \mathbb{F} . The convolutional code \mathcal{C} is observable if and only if $G(z)$ is right-prime, i.e. all $k \times k$ -minors are non-zero and they have non trivial common factors (z^ℓ , $\ell \in \mathbb{N}$ are trivial).

Proposition 1. Let $G(z)$ be an encoder of a (n, k) convolutional code \mathcal{C} over \mathbb{F} . The convolutional code \mathcal{C} is observable if and only if there exists an encoder $G(z)$ and a syndrome Former ψ such that the following sequence is exact

$$0 \rightarrow \mathbb{F}[z]^k \xrightarrow{G(z)} \mathbb{F}[z]^n \xrightarrow{\psi} \mathbb{F}[z]^{n-k} \rightarrow 0$$

in other words, if a convolutional code \mathcal{C} is observable there exists a polynomial matrix $H(z)$ (a syndrome former) with the property that $v \in \mathcal{C}$ if and only if $H(z)v(z) = 0$.

The representation of a code among relatively different representations by means of a polynomial matrix is not unique, but we have the following proposition.

Proposition 2. Two $n \times k$ rational encoders $G_1(z)$, and $G_2(z)$ define the same convolutional code, if and only if there exists a $k \times k$ unimodular matrix $U(z)$ such that $G_1(z)U(z) = G_2(z)$.

Remember that a polynomial matrix $P(z) \in \mathbb{F}[z]$ is unimodular if there exists another matrix $Q(z)$ such that $P(z)Q(z) = I$.

After a suitable permutation of the rows, we can assume that the generator matrix $G(z)$ is in the form

$$G(z) = \begin{pmatrix} P(z) \\ Q(z) \end{pmatrix} \quad (1)$$

with right coprime polynomial factors (block of polynomials) $P(z) \in \mathbb{F}_{(n-k) \times k}$ and $Q(z) \in \mathbb{F}_{k \times k}$, respectively.

It is possible to consider the equivalent rational encoder where $Q(z) \neq 0$

$$\begin{pmatrix} P(z) \\ Q(z) \end{pmatrix} Q^{-1}(z) = \begin{pmatrix} P(z)Q^{-1}(z) \\ I \end{pmatrix}. \quad (2)$$

In the convolutional codes the Hamming distance can be defined as in block codes, the number of symbols in which two encoded bit sequences differ.

In convolutional codes the free distance $d_{\text{free}}(\mathcal{C})$ of a code \mathcal{C} is defined as the minimum Hamming distance between two encoded bit sequences. This depends on the number of errors that the code is able to correct. As in block codes, the Hamming distance is calculated by comparing the outputs with the null input.

In a more formal form

Definition 8.

$$d_{\text{free}}(\mathcal{C}) = \min \{wt(v(z)) \mid v(z) \in \mathcal{C} \text{ with } v(z) \neq 0\}.$$

where the weight $wt(v(z))$ of $v(z) = v_0 + v_1z + \dots + v_lz^l \in \mathbb{F}^n q[z]$ (with $l \geq 0$) is defined as the sum of the Hamming weights of all their coefficients, that is,

$$wt(v(z)) = \sum_{i=0}^l wt(v_i).$$

and Hamming weight $wt(v)$ of a vector $v_i \in \mathbb{F}^n$, is the number of its nonzero components.

The importance of free distance is because it determines the corrective capacity of the code.

3 Convolutional Codes and Linear Systems

In this section, we recall the systems theory tools by introducing the input-state-output representation; then, we will talk about convolutional codes using the linear systems theory; and also introduce the realization for the transition between codes and linear systems.

A discrete linear time-invariant system is described by the equations

$$\begin{cases} x(t+1) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases} \quad (3)$$

where $A \in M_\delta(\mathbb{F})$, $B \in M_{\delta \times k}(\mathbb{F})$, $C \in M_{p \times \delta}(\mathbb{F})$, $D \in M_{p \times k}(\mathbb{F})$ (in our particular setup $p = n - k$) are constant matrices over the field \mathbb{F} , and $u(t) \in \mathbb{F}^k$, $x(t) \in \mathbb{F}^\delta$, $y(t) \in \mathbb{F}^p$ are the input, state and output vectors, respectively.

We will denote a system simply as the quadruple of matrices (A, B, C, D) .

With initial condition $x(0) = 0$, a solution of the Eq. (3) can be obtained by making use of the Z -transform. Let $u(z)$, $x(z)$, $y(z)$ be the Z -transforms of the variables u , x , y of a time-invariant linear system. Then by applying the Z -transform to the equations of the system we obtain

$$\begin{cases} zx(z) = Ax(z) + Bu(z) \\ y(z) = Cx(z) + Du(z) \end{cases} \quad (4)$$

and as a result we have

$$y(z) = (C(zI_\delta - A)^{-1}B + D)u(z), \quad (5)$$

called the transfer function of the system, and the rational matrix

$$C(zI_\delta - A)^{-1}B + D = \frac{1}{\det(zIA)} C \text{adj}(zIA) B + D,$$

where $\text{adj}(M)$ represents the adjoint matrix of M , is called the transfer matrix, (notice that the transfer matrix will always be a rational matrix).

The values $z_0 \in \mathbb{F}$ (where \mathbb{F} denotes the algebraic closure of the field \mathbb{F}) such that $\det(z_0 I_\delta - A) = 0$ are called eigenvalues of A and the set of all eigenvalues is called spectrum of A and is denoted by $\text{Spec}(A)$.

The bridge between linear systems theory and convolutional codes is given by a duality between codes and sets input/state/output representations that are controllable state space systems.

Given a convolutional code, with a specific encoding matrix $G(z)$, we can find four matrices (A, B, C, D) of adequate sizes, corresponding to the size of the encoder, defining the system

$$\left. \begin{aligned} x(t+1) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \\ v(t) &= \begin{pmatrix} y(t) \\ u(t) \end{pmatrix} \\ x(0) &= 0 \end{aligned} \right\}. \quad (6)$$

where $x(t)$ is called state vector, $u(t)$ information vector, $y(t)$ parity vector and $v(t)$ the code vector or codeword. The linear system (A, B, C, D) associated to the encoder $G(z)$ is called a realization of $G(z)$. We are interested in minimal realizations.

In terms of the input-state-output representation of a convolutional code, we have the following characterization of the free distance.

Definition 9.

$$d_{\text{free}}(\mathcal{C}) = \min \left\{ \sum_{t=0}^{\infty} wt(u_t) + \sum_{t=0}^{\infty} wt(y_t) \right\}$$

Where the minimum is considered over all non-null code words.

Due to algebraic reasons, we assume throughout the paper that the code words are of finite weight.

Another well-studied concept in convolutional codes theory is that of column distances. The j th column distance of the code \mathcal{C} is defined as the following manner

Definition 10.

$$d_j = \min \left\{ \sum_{t=0}^j wt(u_t) + \sum_{t=0}^j wt(y_t) \right\},$$

where the minimum is taken over all trajectories (u_t, y_t) of the system (6) with initial vector $u_0 \neq 0$.

It is clear that

$$d_0 \leq d_1 \leq d_2 \leq \dots$$

and hence there exists an integer r such that $d_r = d_{r+j}$ for all $j \geq 0$. This largest possible column distance is of central importance in coding theory.

Proposition 3.

$$d_{\text{free}} = \lim_{j \rightarrow \infty} d_j$$

Codes with a large free distance and the largest possible column distances are very desirable.

3.1 Realization

Linear systems for convolutional codes represent a mechanism to work on every little sub-piece of the encoding process. If we try to understand the physical control process, that goes along with the coding, the state of our encoding machine is modified by both the dynamics matrix and the input matrix.

Now, we present a method to obtain a realization.

Let $G(z)$ be a matrix generator of (n, k) convolutional code, in which we consider that is in the form $\begin{pmatrix} P(z) \\ Q(z) \end{pmatrix}$ with $Q(z)$ invertible and the degree δ of the polynomial $\det Q(z)$ being maximal among all minors of order k .

We decompose $P(z)Q(z)^{-1}$ into a polynomial matrix and a strictly proper matrix.

Let $p(z) = z^\delta + a_{\delta-1}z^{\delta-1} + \dots + a_1z + a_0$ the monic polynomial deduced from $\det Q(z)$. So, the matrix $P(z)Q(z)^{-1}$ is written in the form

$$\begin{pmatrix} d_{11} + \frac{q_{11}(z)}{p(z)} & \dots & d_{1k} + \frac{q_{1k}(z)}{p(z)} \\ \vdots & & \vdots \\ d_{n-k1} + \frac{q_{n-k1}(z)}{p(z)} & \dots & d_{n-kk} + \frac{q_{n-kk}(z)}{p(z)} \end{pmatrix}$$

$$q_{ij} = c_0^{ij} + c_1^{ij}z + \dots + c_{\delta-1}^{ij}z^{\delta-1}$$

(by construction $d_{ij} \in \mathbb{F}$ and degree $q_{ij} < \delta$).

First of all and for simplicity, we analyze the case where $k = 1$.

We consider the following matrices

$$D = \begin{pmatrix} d_{11} \\ \vdots \\ d_{n-k1} \end{pmatrix} \in M_{(n-k) \times 1}(\mathbb{F}).$$

$$A = \begin{pmatrix} -a_{\delta-1} & -a_{\delta-2} & \dots & -a_1 & -a_0 \\ 1 & 0 & \dots & 0 & 0 \\ & \ddots & & & \\ 0 & 0 & \dots & 1 & 0 \end{pmatrix} \in M_\delta(\mathbb{F})$$

$$B = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \in M_{\delta \times 1}(\mathbb{F})$$

$$C = \begin{pmatrix} c_{\delta-1}^{11} & \cdots & c_0^{11} \\ \vdots & & \vdots \\ c_{\delta-1}^{n-k1} & \cdots & c_0^{n-k1} \end{pmatrix} \in M_{(n-k) \times \delta}.$$

A simple calculation shows that $C(zI_\delta - A)^{-1}B + D = P(z)Q(z)^{-1}$.

Example 1. We consider the following code

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

over the field \mathbb{F}_4 ,

$$\begin{aligned} G(z) &= \begin{pmatrix} 1 + z + z^2 \\ \alpha + z + \alpha^2 z^2 \\ \alpha^2 + z + \alpha z^2 \end{pmatrix} = \begin{pmatrix} \frac{1 + z + z^2}{\alpha^2 + z + \alpha z^2} \\ \frac{\alpha + z + \alpha^2 z^2}{\alpha^2 + z + \alpha z^2} \\ 1 \end{pmatrix} \alpha^2 + z + \alpha z^2 \\ &= \begin{pmatrix} \frac{1 + z + z^2}{\alpha^2 + z + \alpha z^2} \\ \frac{\alpha + z + \alpha^2 z^2}{\alpha^2 + z + \alpha z^2} \\ 1 \end{pmatrix} = \begin{pmatrix} \alpha + 1 + \frac{1 + \alpha + \alpha z}{\alpha^2 + z + \alpha z^2} \\ \alpha + \frac{(1 + \alpha) + (1 + \alpha)z}{\alpha^2 + z + \alpha z^2} \\ 1 \end{pmatrix}; \\ P(z)Q(z)^{-1} &= \begin{pmatrix} 1 + \alpha + \frac{\alpha + z}{\alpha + (\alpha + 1)z + z^2} \\ \alpha + \frac{\alpha + \alpha z}{\alpha + (\alpha + 1)z + z^2} \end{pmatrix}. \end{aligned}$$

Following as before we obtain the following realization (A, B, C, D) of the convolutional code where

$$\begin{aligned} D &= \begin{pmatrix} \alpha + 1 \\ \alpha \end{pmatrix}, B = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \\ q_{11} &= \alpha + z = c_0^{11} + c_1^{11}z \\ q_{21} &= \alpha + \alpha z = c_0^{21} + c_1^{21}z \\ C &= \begin{pmatrix} c_1^{11} & c_0^{11} \\ c_1^{21} & c_0^{21} \end{pmatrix} = \begin{pmatrix} 1 & \alpha \\ \alpha & \alpha \end{pmatrix} \\ p(z) &= a_0 + a_1 z + z^2 = \alpha + (1 + \alpha)z + z^2 \\ A &= \begin{pmatrix} -a_1 & -a_0 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 + \alpha & \alpha \\ 1 & 0 \end{pmatrix} \end{aligned}$$

A similar result holds for $k > 1$ case, the single input state-space models that correspond to the individual transfer functions from each input to each output, could be stacked into one large $k > 1$ state-space model.

Example 2. Let $G(z)$ be the following encoder matrix

$$G(z) = \begin{pmatrix} 1+z & 1 \\ z & 1+z \\ 1+z+z^2 & 0 \\ 0 & 1+z+z^2 \end{pmatrix} = \begin{pmatrix} P(z) \\ Q(z) \end{pmatrix}$$

So,

$$C(zI - A)^{-1}B + D = P(z)Q(z)^{-1} = \begin{pmatrix} \frac{1+z}{1+z+z^2} & \frac{1}{1+z+z^2} \\ \frac{z}{1+z+z^2} & \frac{1+z}{1+z+z^2} \end{pmatrix}$$

In this case $D = 0$ and $A = \begin{pmatrix} -1 & 0 & -1 & 0 \\ 0 & -1 & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$, $B = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$, $C = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix}$.

An important concept in realization theory is the minimality.

Definition 11. A realization (A, B, C, D) of a transfer matrix $G(z)$ is said to be minimal if no other realization of $G(z)$ has smaller dimension.

In order to know the minimality of the realization we have the following result

Theorem 1. Let (A, B, C, D) be a realization of $G(z)$. The following statements are equivalent:

- (1) (A, B, C, D) is minimal.
- (2) The poles of $G(z)$ are the eigenvalues of A

Theorem 2. Given a transfer matrix $G(z)$, all the minimal realizations of $G(z)$ are algebraically equivalent.

The equivalence is in the following sense.

Definition 12. Two systems (A, B, C, D) and (A', B', C', D') are equivalent if and only if there exist an invertible matrix P such that

$$\begin{pmatrix} A' & B' \\ C' & D' \end{pmatrix} = \begin{pmatrix} P & \\ & I_p \end{pmatrix} \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} P^{-1} & \\ & I_k \end{pmatrix}$$

Notice that this equivalence relation preserve the transfer matrix associate to the system:

$$\begin{aligned} C'(zI - A')^{-1}B' + D' &= CP^{-1}(zI - PAP^{-1})^{-1}PB + D \\ &= CP^{-1}(P(zI - A)P^{-1})^{-1}PB + D = CP^{-1}P(zI - A)^{-1}P^{-1}PB + D \\ &= C(zI - A)^{-1}B + D \end{aligned}$$

3.2 Control Properties of Convolutional Codes

We review the standard conditions about reachability (controllability from the origin) over the input-state-output representation of a convolutional code \mathcal{C} over \mathbb{F} . First, we recall some results.

Definition 13. Let (A, B, C, D) be matrices over \mathbb{F} describing a linear system as in (3). The controllability (reachability) matrix was defined by

$$\mathcal{C}(A, B) = (B \ AB \ \dots \ A^{\delta-2}B \ A^{\delta-1}B) \quad (7)$$

It is well-known that, a linear system (A, B, C, D) over a field \mathbb{F} is reachable if its controllability matrix has full row rank; that is, $\text{rank } \Phi_\delta(A, B) = \delta$. Or, equivalently, the Hautus test is verified.

$$\text{rank } \mathcal{C}(A, B) = \delta \text{ if and only if } \text{rank } (z_0 I + A \mid B) = \delta, \forall z_0 \in \overline{\mathbb{F}}$$

Remark 1. The controllability depends only on the state equation of the system.

Remark 2. Note that by construction, realization constructed is controllable.

Duality between convolutional codes and reachable state space realization is useful to construct observable convolutional codes: an input-state-output realization is always a reachable dynamical linear system. If it is also observable, then the following results allow us to get an associated observable convolutional code.

Rosenthal and York in [19] show that, starting from a minimal representation of a convolutional code, then this code is non-catastrophic if and only if the pair (A, C) is observable.

In terms of linear systems, let (A, B, C, D) be matrices over \mathbb{F} describing a system. The observability matrix is defined by

$$\mathcal{O}(A, C) = \begin{pmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{\delta-1} \end{pmatrix} \quad (8)$$

Lemma 2. The system (A, B, C, D) is observable if and only $\text{rank } \mathcal{O}(A, C) = \delta$ or equivalently, by the Hautus Test, $\forall z_0 \in \overline{\mathbb{F}}$,

$$\text{rank} \begin{pmatrix} -z_0 I + A \\ C \end{pmatrix} = \delta$$

There are multiple realizations (A, B, C, D) for a given linear system. In particular, δ , the size of matrix A is not constant in the set of all realizations. Since δ is always a positive integer, it must reach a minimum value for certain realization. This minimum value of δ is called the McMillan degree of the system.

A realization (A, B, C, D) for which δ is equal to the degree of McMillan, we say that is a minimal realization. It is well known that the minimality property of a realization is related to the concepts of controllability and observability in the following sense.

Theorem 3 ([2]). *The realization (A, B, C, D) of a linear system is minimal if and only if (A, B) is a controllable pair and (A, C) is an observable pair.*

It is important to note that while in linear systems theory, a realization is minimal if and only if the pair (A, B) is controllable and the pair (A, C) is observable, for input-state-output representation of a convolutional code we do not have the same result. In fact, it is enough that the pair (A, B) be controllable so that the representation is minimal.

Related to the decodification of the encoders is the output-observability property.

Output-observability represents the possibility of an internal state, to be only defined by a finite set of outputs, for a finite number of steps. There are some literature about this topic, as for example [6–8].

Definition 14. *A system (A, B, C, D) is said to be output observable if the state sequence $x(0), \dots, x(\ell)$ is uniquely determined by the knowledge of the output sequence $y(0), \dots, y(\ell)$ for a finite number of steps $\ell \in \mathbb{N}$.*

Observe that $x(1), \dots, x(\ell)$ are determined by the knowledge of $x(0)$ and $u(0), \dots, u(\ell - 1)$ and the elements $x(0), u(0), \dots$, and $u(\ell)$ can be obtained solving the following system of matrix equations.

$$\begin{cases} y(0) = Cx(0) + Du(0) \\ y(1) = Cx(1) + Du(1) \\ \quad = CAx(0) + CBu(0) + Du(1) \\ \vdots \\ y(\ell) = Cx(\ell) + Du(\ell) \\ \quad = CA^\ell x(0) + CA^{\ell-1}Bu(0) + \dots + CBu(\ell-1) + Du(\ell). \end{cases} \quad (9)$$

Calling $T_\ell(A, B, C, D)$ (that we simply write T_ℓ if no confusion is possible) the matrix

$$T_\ell = \begin{pmatrix} C & D & & \\ CA & CB & D & \\ CA^2 & CAB & CB & D \\ \vdots & & & \ddots & \ddots \\ CA^\ell & CA^{\ell-1}B & CA^{\ell-2}B & \dots & CB & D \end{pmatrix}. \quad (10)$$

We have the following.

Proposition 4. *A system (A, B, C, D) is output observable if and only if the matrix T_ℓ has full row rank for all $\ell \in \mathbb{N}$.*

Remark 3. If the number of rows is bigger than the number of columns, there are values of $y(0), \dots, y(\ell)$, for which $(y(0), \dots, y(\ell))$ is not a parity vector.

Corollary 1. *A necessary condition for output-observability of the system (A, B, C, D) is that the matrix $(C \ D)$ has full row rank.*

Therefore, we assume that the number of rows is less than or equal to the number of columns. It is well known that in this case and for each ℓ , the systems (9) have solution for all $y(0), \dots, y(\ell)$, if and only if the systems have full rank.

Fixing the initial state $x(s) = 0$, the output-observability matrix allows us to describe a sequence of trajectories $\{v_s, \dots, v_{s+\ell}\}$ in the following manner.

Theorem 4. *Let (A, B, C, D) be a representation of a convolutional code. Suppose that the initial state of the system is $x(s) = 0$, then*

$$\{v_s, \dots, v_{s+\ell}\} = \text{Ker } T_\ell,$$

where

$$T_\ell = \begin{pmatrix} D & -I & & & \\ CB & 0 & D & -I & \\ CAB & 0 & CB & 0 & D & -I \\ \vdots & & \ddots & \ddots & & \\ CA^{\ell-1}B & 0 & CA^{\ell-2}B & 0 & \dots & CB & 0 & D & -I \end{pmatrix}$$

The output observability matrix is related with the syndrome former matrix used by Rosenthal and York [20], solving the decoding problem.

Let (A, B, C, D) be a realization of a convolutional code.

From the system

$$\begin{pmatrix} C & D \\ CA & CB & D \\ CA^2 & CAB & CB & D \\ \vdots & & \ddots & \ddots \\ CA^\ell & CA^{\ell-1}B & CA^{\ell-2}B & \dots & CB & D \end{pmatrix} \begin{pmatrix} x(s) \\ u(s) \\ \vdots \\ u(s+\ell) \end{pmatrix} = \begin{pmatrix} y(s) \\ y(s+1) \\ \vdots \\ y(s+\ell) \end{pmatrix} \quad (11)$$

we can deduce the syndrome former matrix for the given code.

Proposition 5. *Suppose that $\ell \geq \delta$. By making elementary transformations to matrix Eq. (11) we can deduce the syndrome former matrix for the convolutional code.*

Proof. The system (11) can be rewritten as

$$\begin{pmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^\ell \end{pmatrix} x(s) = \begin{pmatrix} D & & & I & & \\ CB & D & & I & & \\ CAB & CB & D & & I & \\ \vdots & & & \ddots & \ddots & \\ CA^{\ell-1}B & CA^{\ell-2}B & \dots & CB & D & I \end{pmatrix} \begin{pmatrix} -u(s) \\ -u(s+1) \\ \vdots \\ -u(s+\ell) \\ y(s) \\ y(s+1) \\ \vdots \\ y(s+\ell) \end{pmatrix} \quad (12)$$

and making row elementary transformations, we obtain

$$\left(\frac{\mathcal{O}(A, B)}{0} \right) (x(s)) = \left(\frac{M_1 | M_2}{M_3 | M_4} \right) \begin{pmatrix} -u(s) \\ \vdots \\ -u(s+\ell) \\ y(s) \\ \vdots \\ y(s+\ell) \end{pmatrix} \quad (13)$$

Then, $(M_3 | M_4)$ is the syndrome former matrix.

Example 3. In \mathbb{F}_2 , we consider the system (A, b, C, D) with

$$A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad C = (1 \ 0), \quad D = (1).$$

Then, the system (12) for this particular case is

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix} (x(s)) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} -u(0) \\ -u(1) \\ -u(2) \\ -u(3) \\ -u(4) \\ -u(5) \\ -u(6) \\ y(0) \\ y(1) \\ y(2) \\ y(3) \\ y(4) \\ y(5) \\ y(6) \end{pmatrix}$$

Now, taking

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

The system is reduced to

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} (x(s)) = \left(\begin{array}{cccccc|cccccc} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 1 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 1 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 \\ -1 & 1 & 0 & 1 & 1 & 0 & -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 & 1 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 1 \\ -1 & 1 & 0 & 1 & 0 & 1 & 1 & -1 & 0 & 0 & 0 & 0 & 1 \end{array} \right) \begin{pmatrix} -u(0) \\ -u(1) \\ -u(2) \\ -u(3) \\ -u(4) \\ -u(5) \\ -u(6) \\ y(0) \\ y(1) \\ y(2) \\ y(3) \\ y(4) \\ y(5) \\ y(6) \end{pmatrix}$$

So, the syndrome former matrix is

$$\left(\begin{array}{cccccc|cccccc} -1 & 1 & 1 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 1 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 \\ -1 & 1 & 0 & 1 & 1 & 0 & -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 & 1 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 1 \\ -1 & 1 & 0 & 1 & 0 & 1 & 1 & -1 & 0 & 0 & 0 & 0 & 1 \end{array} \right).$$

On the other hand, the following L -order block Toeplitz submatrix of the output-observability matrix

$$\mathfrak{T}_L = \begin{pmatrix} D & & & & \\ CB & D & & & \\ CAB & CB & D & & \\ \vdots & & & \ddots & \ddots \\ CA^{L-1}B & CA^{L-2}B & \dots & CB & D \end{pmatrix}. \quad (14)$$

allows us to obtain a characterization of the convolutional codes with maximum distance profile, in terms of its input-state-output representation.

Remember that (see [12]), an (n, k) -code \mathcal{C} , with column distances d_j and free distance d_{free} , has a maximum distance profile if

$$d_j = (n - k)(j + 1) + 1 \text{ for } j = 0, \dots, L = \left\lfloor \frac{\delta}{k} \right\rfloor + \left\lfloor \frac{\delta}{n - k} \right\rfloor$$

Maximum distance profile convolutional codes are characterized by the property that two trajectories which start in the same state and proceed to a different state will have the maximum possible distance from each other relative to any other convolutional code of the same rate and degree.

Theorem 5 ([12]). *The matrices (A, B, C, D) generate a (n, k) -code with of maximum distance profile, (in terms of the input-state-output representation), if and only if the matrix \mathfrak{T}_L , verifies that any minor that is not trivially zero, is non-zero.*

Minor not trivially zero is understood in the following sense. We consider In this definition, we think of the nonzero entries of the block Toeplitz matrix \mathfrak{T}_L as indeterminates of the polynomial ring $R = \mathbb{F}_q[x_{1,1}, \dots, x_{1,(L+1)k}, \dots, x_{(L+1)p,1}, \dots, x_{(L+1)p,(L+1)k}]$. Specifically, if the entry (i, j) of the matrix is nonzero, we set it equal to $x_{i,j}$; otherwise, we leave it zero. So, a minor of \mathfrak{T}_L is called trivially zero if it is zero viewed as an element of the ring R .

Example 4. In \mathbb{F}_4 , the convolutional code (A, B, C, D) with

$$A = \begin{pmatrix} \alpha & \\ & 1 \end{pmatrix}, B = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, C = (\alpha + 1 \ \alpha), D = (\alpha + 1)$$

where $\delta = 2$, $k = 1$, $p = 1$ then $L = 1$ and

$$\mathfrak{T}_L = \begin{pmatrix} \alpha + 1 & \\ & 1 & \alpha + 1 \end{pmatrix}$$

So, the convolutional code has maximum distance profile.

4 Families of Convolutional Codes over Finite Fields

We are interested in convolutional codes where the matrices (A, B, C, D) or one of them, are not entirely defined having in certain positions parameters that can take any value from the field. So we can consider this parametric code as a family of convolutional codes.

These families of codes may be of interest when attempting to protect or hide certain information.

Anyway, we can not place parameters anywhere if we want to maintain certain properties of the code. In particular the structure of the matrix A , in this case and taking into account that the equivalence relation given in Definition 12 preserves this structure of matrices, we can consider classes of systems, and as representative of each class we find a system in which the matrix A is in some reduced form.

Example 5. In \mathbb{F}_5 , we consider the following family of systems $(A(a), B(a), C(a), D(a))$ with

$$A(a) = \begin{pmatrix} 1 & a & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{pmatrix}, B(a) = \begin{pmatrix} 1 & 2 \\ 1 & 1 \\ 1 & 0 \end{pmatrix}, C(a) = \begin{pmatrix} 1 & 1 & 1 \\ 2 & 0 & 1 \end{pmatrix}, D(a) = \begin{pmatrix} 1 & 2 \\ 2 & 0 \end{pmatrix}$$

Taking the family of invertible matrices $P = \begin{pmatrix} 1 & a & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$, this family is equivalent to $(A_1(a), B_1(a), C_1(a), D_1(a))$ with

$$A_1(a) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{pmatrix}, \quad B_1(a) = \begin{pmatrix} 1-a & 2-a \\ 1 & 1 \\ 1 & 0 \end{pmatrix}, \quad C_1(a) = \begin{pmatrix} 1 & a+1 & 1 \\ 2 & 2a & 1 \end{pmatrix}, \quad D_1(a) = \begin{pmatrix} 1 & 2 \\ 2 & 0 \end{pmatrix}$$

So, for each $a \in \mathbb{F}_5$ we have a different system but all matrices $A(a)$ have the same structure. Obviously is not the same for the family $(\bar{A}(a), \bar{B}(a), \bar{C}(a), \bar{D}(a))$ with

$$\bar{A}(a) = \begin{pmatrix} 1+a & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{pmatrix}, \quad \bar{B}(a) = \begin{pmatrix} 1 & 2 \\ 1 & 1 \\ 1 & 0 \end{pmatrix}, \quad \bar{C}(a) = \begin{pmatrix} 1 & 1 & 1 \\ 2 & 0 & 1 \end{pmatrix}, \quad \bar{D}(a) = \begin{pmatrix} 1 & 2 \\ 2 & 0 \end{pmatrix}$$

where the matrix A in each member of the family has a different structure.

In \mathbb{F}_q there are exactly $q^{\delta^2} \times q^{k\delta} \times q^{p\delta} \times q^{pk}$ different systems. In particular, if the matrix A is in such a way that in its reduced form is diagonal, we have

$$\frac{(\delta + q - 1)!}{\delta!(q - 1)!} \times q^{k\delta} \times q^{p\delta} \times q^{pk}.$$

Taking into account that the cardinal of the set of invertible matrices $Gl(\delta, \mathbb{F}_q)$ is $\prod_{k=0}^{\delta-1} (q^\delta - q^k)$, it is possible to count the number of elements of each equivalent class and the number of classes.

For that, it suffices to define an action of the linear group over the set of systems $\mathcal{M} = \{(A, B, C, D)\}$:

$$\begin{aligned} \varphi : Gl(\delta, \mathbb{F}_q) \times \mathcal{M} &\longrightarrow \mathcal{M} \\ (P, (A, B, C, D)) &\longrightarrow (P^{-1}AP, P^{-1}B, CP, D) \end{aligned}$$

Then, after to compute the stabilizer $\mathcal{S}_{(A, B, C, D)}$ of a fixed point $(A, B, C, D) \in \mathcal{M}$ defined as $\mathcal{S}_{(A, B, C, D)} = \{P \in Gl(\delta, \mathbb{F}_q) \mid \alpha(P, (A, B, C, D)) = (A, B, C, D) = \{P \in Gl(\delta, \mathbb{F}_q) \mid AP - PA = 0, PB - B = 0, CP - C = 0\}$

and now, it is easy to prove that there is a bijection between the set of equivalent systems to (A, B, C, D) and the quotient group $Gl(\delta, \mathbb{F}_q)/\mathcal{S}_{(A, B, C, D)}$.

Given a convolutional code (A, B, C, D) , we are interested in to perturb it in order to improve their behaviour and control properties. That is, to find the values of the parameters for which our code has the appropriate or expected properties.

Example 6. In \mathbb{F}_4 , let (A, B, C, D) be a convolutional code with

$$A = \begin{pmatrix} \alpha & \alpha+1 \\ \alpha+1 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} \alpha+1 \\ \alpha \end{pmatrix}, \quad C = (\alpha+1 \ 1), \quad D = (1)$$

This code is no controllable because of:

$$\text{rank} \begin{pmatrix} z - \alpha & -(\alpha + 1) & \alpha + 1 \\ -(\alpha + 1) & z - 1 & \alpha \end{pmatrix} = 1 \text{ for } z = \alpha + 1$$

And not observable because of:

$$\text{rank} \begin{pmatrix} z - \alpha & -(\alpha + 1) \\ -(\alpha + 1) & z - 1 \\ \alpha + 1 & 1 \end{pmatrix} = 1 \text{ for } z = 0$$

Considering the family of convolutional codes $(A(a), B(a), C(a), D(a))$ be a convolutional code with

$$A = \begin{pmatrix} \alpha + a & \alpha + 1 \\ \alpha + 1 & 1 \end{pmatrix}, B = \begin{pmatrix} \alpha + 1 \\ \alpha \end{pmatrix}, C = (\alpha + 1 \ 1), D = (1)$$

The codes of the family are controllable and observable if and only if $a \neq 0$.

References

1. de Castro, N.: Feedback classification of linear systems and convolutional codes. applications in cybernetics, coding theory and cryptography. Ph.D. Universidad de León (2016)
2. Chen, C.T.: Introduction to Linear System Theory. Holt, Rinehart and Winston Inc., New York (1970)
3. Elias, P.: Coding for noisy channels. IRE Conv. Rec. **4**, 37–46 (1955)
4. Forney, G.D.: Convolutional codes: algebraic structure. IEEE Trans. Inf. Theor. **16**(6), 720–738 (1970)
5. Fragouli, C., Wesel, R.D.: Convolutional codes and matrix control theory. In: Proceedings of the 7th International Conference on Advances in Communications and Control, Athens, Greece (1999)
6. Garcia-Planas, M.I., Domínguez-Garcia, J.L.: A general approach for computing residues of partial-fraction expansion of transfer matrices. WSEAS Trans. Math. **12**(7), 647–756 (2013)
7. Garcia-Planas, M.I., Domínguez-Garcia, J.L.: Alternative tests for functional and pointwise output-controllability of linear time-invariant systems. Syst. Control Lett. **62**(5), 382–387 (2013)
8. García-Planas, M.I., Tarragona, S.: Output observability of time-invariant singular linear systems. In: PHYSCON 2011, León, Spain (2011)
9. García-Planas, M.I., Soudi, E.M., Um, L.E.: Convolutional codes under linear systems point of view. Analysis of output-controllability. Wseas Trans. Math. **11**(4), 324–333 (2010)
10. Gluesing-Luerssen, H.: On the weight distribution of convolutional codes. Linear Algebra Appl. **408**, 298–326 (2005)
11. Gluesing-Luerssen, H., Helmke, U., Iglesias Curto, J.I.: Algebraic decoding for doubly cyclic convolutional codes. Adv. Math. Commun. **4**, 83–99 (2010)
12. Hutchinsona, R., Rosenthal, J., Smarandacheb, R.: Convolutional codes with maximum distance prole. Syst. Control Lett. **54**(1), 53–63 (2005)

13. Kuijper, M., Pinto, R.: On minimality of convolutional ring encoders. *IEEE Trans. Inf. Theor.* **55**(11), 4890–4897 (2009)
14. Rosenthal, J.: Some interesting problems in systems theory which are of fundamental importance in coding theory. In: *Proceedings of the 36th IEEE Conference on Decision and Control* (1997)
15. Rosenthal, J.: An algebraic decoding algorithm for convolutional codes. In: *Dynamical Systems, Control, Coding, Computer Vision; New Trends, Interfaces, and Interplay*, Birkhäuser, Basel, pp. 343–360 (1999)
16. Rosenthal J.: Connections between linear systems and convolutional codes. In: Marcus B., Rosenthal J. (eds.) *Codes, Systems, and Graphical Models. The IMA Volumes in Mathematics and its Applications*, vol 123. Springer, New York (2001)
17. Rosenthal, J., Smarandache, R.: Maximum distance separable convolutional codes. *Appl. Algebra Eng. Commun. Comput.* **10**(1), 15–32 (1999)
18. Rosenthal, J., Schumacher, J.M., York, E.V.: On behaviors and convolutional codes. *IEEE Trans. Inf. Theor.* **42**(6), 1881–1891 (1996)
19. Rosenthal, J., York, E.V.: BCH convolutional codes. *IEEE Trans. Inform. Theor.* **45**(6), 1833–1844 (1999)
20. Rosenthal, J., York, E.V.: On behaviors and convolutional codes. *IEEE Trans. Inform. Theor.* **42**(6), 1881–1891 (1996)
21. Smarandache, R., Gluesing-Luerssen, H., Rosenthal, J.: Constructions of MDS-convolutional codes. *IEEE Trans. Inf. Theor.* **47**(5), 2045–2049 (2002)
22. Um, L.E.: A contribution to the theory of convolutional codes from systems theory point of view. Ph.D. dissertation. Université Mohammed V. Rabat (2015)
23. York, E.V.: Algebraic description and construction of error correcting codes, a systems theory point of view, Ph.D. dissertation, Univ. Notre Dame (1997)

Codes, Cryptology and Information Security
Second International Conference, C2SI 2017, Rabat,
Morocco, April 10–12, 2017, Proceedings - In Honor of
Claude Carlet
El Hajji, S.; Nitaj, A.; Souidi, E.M. (Eds.)
2017, XII, 384 p. 46 illus., Softcover
ISBN: 978-3-319-55588-1