**ESAIM: Control, Optimisation and Calculus of Variations** June 2002, Vol. 8, 703–713
URL: `http://www.emath.fr/cocv/` DOI: 10.1051/cocv:2002035

# ON THE STRUCTURE OF LINEAR RECURRENT ERROR-CONTROL CODES

Michel Fliess[1, 2]

**Abstract**. We are extending to linear recurrent codes, *i.e.*, to time-varying convolutional codes, most of the classic structural properties of fixed convolutional codes. We are also proposing a new connection between fixed convolutional codes and linear block codes. These results are obtained thanks to a module-theoretic framework which has been previously developed for linear control.

**Mathematics Subject Classification.** 93C65, 94B05, 94B10, 12H05, 13A02, 13C05.

Received December 2, 2001.

**À la mémoire du professeur Jacques-Louis LIONS**

## 1. Introduction

This paper is devoted to various aspects of convolutional codes which are with linear block codes the most popular class of error-control codes (see, *e.g.* [2,4,30,31]). It is based on the well known ties between convolutional codes and linear automatic control (see, *e.g.* [21–23, 25–27, 33, 34, 38]).

Our aim is twofold:

- we are extending to linear recurrent codes, *i.e.*, to time-varying convolutional codes, most of the classic structural properties of fixed, *i.e.*, time-invariant, convolutional codes (see, *e.g.* [6, 26, 34, 37]). Although Shannon's channel coding theorem has been extended to time-varying convolutional codes (see [42]) and not to fixed ones, those time-varying codes were much less utilized in practice than the time-invariant ones (see, nevertheless [26]);
- the connection between fixed convolutional codes and special types of linear block codes, like cyclic codes, which has been the subject of many investigations (see, *e.g.* [31, 37, 40] and the references therein), is here approached from a new perspective. Our main theorem states that for an arbitrary block code there exists a convolutional code such that all its *frames* (see Sect. 4.2 for a precise definition) are isomorphic to this block code. This leads to families of convolutional codes and to a feedback decoding procedure, which seem to be novel.

The relationship between those two rather independent subjects is a module-theoretic approach to linear control [9, 11, 12, 15, 18–20] [3], which has been quite useful in practice [16–19, 36] [4]. We are utilising some elementary

*Keywords and phrases:* Convolutional codes, linear recurrent codes, block codes, transducers, encoders, feedback decoding, linear systems, controllability, observability, input-output inversion, modules.

[1] Centre de Mathématiques et Leurs Applications, École Normale Supérieure de Cachan, 61 avenue du Président Wilson, 94235 Cachan, France; e-mail: `fliess@cmla.ens-cachan.fr`

[2] Laboratoire GAGE, École polytechnique, 91128 Palaiseau, France.

[3] One should also cite some recent works of different natures on the connection between coding and control (see, *e.g.* [7, 24, 32, 38–40]).

[4] See also [13] and the references therein.

notions of difference algebra [5], homological algebra [41], and non-commutative algebra [29,35], which is most natural in the time-varying case (see, *e.g.* [9,11,12,15]).

In the first part we define, following [26], *transducers*, *i.e.*, input-output systems, and study their main properties: state-variable representation, controllability, observability, transfer matrices, input-output inversion. In particular, an *encoder* is a right invertible transducer. The second part is devoted to codes. A code, here, is an equivalence class between encoders having the same output. We derive syndrome formers, dual codes, parity check matrices, polynomial and basic encoders, and Forney's theory in a manner which is often very short thanks to our algebraic setting. We end with the connection with block codes.

The following topics will be developed in future publications:

- constructions of cyclic-like convolutional codes, *i.e.*, convolutional codes which thanks to the results of Section 4 will also benefit from the properties of some types of cyclic codes;
- turbo-codes [1]. They are often given by two convolutional encoders in parallel with an interleaver, and are known to be related to time-varying convolutional codes;
- non-linear tree codes, which correspond to non-linear encoders, *i.e.*, to right invertible non-linear input-output systems [8] (see Sect. 2.6.2);
- cryptography is already known to be related to error-control codes (see, *e.g.* [45]). Encrypters will be associated to invertible square input-output systems (see Sect. 2.6.2).

## 2. Linear recurrent transducers

### 2.1. Algebraic preliminaries

#### 2.1.1. *Difference fields*

A *difference field* [5] is a commutative field $F$, equipped with a *transformation* $\delta : F \to F$, *i.e.*, a monomorphism. Here $\delta$ should be understood as the *delay operator* of one unit of time. A *constant* is an element $c \in F$, such that $c\delta = c$ (mappings are written on the right). The *subfield of constants* of $F$ is the subfield of all constant elements of $F$. A *field of constants* is a difference field which coincide with its subfield of constants. The *inversive closure* $F^{\maltese}$ [5] of $F$, which is unique up to isomorphism, is the smallest difference overfield of $F$ such that $\delta$ is an automorphism. The difference field $F$ is said to be *inversive* if, and only if, $F = F^{\maltese}$.

**Example 2.1.** Let $\mathbb{F}(t)$ be the field of rational functions in the indeterminate $t$ over the field $\mathbb{F}$, a finite field for instance. With the $\mathbb{F}$-automorphism $\delta : \mathbb{F}(t) \to \mathbb{F}(t)$, $t \mapsto t - 1$, $\mathbb{F}(t)$ becomes an inversive difference field, where the subfield of constants is $\mathbb{F}$.

#### 2.1.2. *A principal right ideal ring*

The set of polynomials of the form

$$\sum_{\text{finie}} \delta^s a_s \tag{2.1}$$

$a_s \in F$, is a *principal right ideal ring* $F[\delta]$. It is commutative if, and only if, $F$ is a field of constants.

### 2.2. Input-output system

A *linear system* is a finitely generated right $F[\delta]$-module, where $F$ is an inversive difference field[5]. A *linear recurrent transducer*, or a *time-varying convolutional transducer*, or a *linear input-output system*, $\mathcal{T}$ is a system with the following properties:

- there is an *input*, *i.e.*, a finite subset $\boldsymbol{u} = (u_1, \ldots, u_k)$ of $\mathcal{T}$, such that the quotient module $\mathcal{T}/\mathrm{span}_{F[\delta]}(\boldsymbol{u})$ is torsion. The input will be assumed to be *independent*, *i.e.*, the module $\mathrm{span}_{F[\delta]}(\boldsymbol{u})$ is free, of rank $k$;

[5] This assumption on $F$ being inversive will simplify several further developments. It does not seem to bring any limitation from a practical viewpoint (see, *e.g.* [26]).

- there is an *output*, *i.e.*, a finite subset $\boldsymbol{y} = (y_1, \ldots, y_n)$ of $\mathcal{T}$;
- $\mathcal{T}$ is *causal* (cf. [11]), or *nonanticipative*, *i.e.*, the *semi-linear* [3] mapping $\delta : \mathcal{T}/\mathrm{span}_{F[\delta]}(\boldsymbol{u}) \to \mathcal{T}/\mathrm{span}_{F[\delta]}(\boldsymbol{u})$ is injective.

**Example 2.2.** The transducer $y\delta = u$, *i.e.*, $y(t-1) = u(t)$, where $k = n = 1$, should obviously be viewed as non-causal. It is also non-causal in our abstract setting. As a matter of fact the quotient module $\mathcal{T}/\mathrm{span}_{F[\delta]}(u)$ is a 1-dimensional $F$-vector space spanned by an element corresponding to $u(t+1)$, which is mapped to 0 by $\delta$.

When $F$ is a field of constants, a linear recurrent transducer is called a *fixed*, or *time-invariant*, *convolutional transducer*.

### 2.3. State-variable representation

When viewed as a $F$-vector space, the finitely generated torsion module $\mathcal{T}/\mathrm{span}_{F[\delta]}(\boldsymbol{u})$ is of finite dimension, $m$. Take a basis $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_m)$. The next lemma is clear.

**Lemma 2.3.** *$\boldsymbol{\xi}\delta$ is also a basis.*

**Corollary 2.4.** *$\boldsymbol{\xi} = \boldsymbol{\xi}\delta\ A$, $A \in F^{m\times m}$, $\det(A) \neq 0$.*

Take in $\mathcal{T}$ a $m$-tuple $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_m)$ the image of which in $\mathcal{T}/\mathrm{span}_{F[\delta]}(\boldsymbol{u})$ is $\boldsymbol{\xi}$. Then Corollary 2.4 yields a *generalised state-variable representation* of the transducer $\mathcal{T}$

$$\boldsymbol{\eta} = \boldsymbol{\eta}\delta\ A + \sum_{\mu=0}^{\nu} \boldsymbol{u}\delta^{\mu}\ \bar{B}_\mu \tag{2.2}$$

$$\boldsymbol{y} = \boldsymbol{\xi}\ \bar{C} + \sum_{\text{finite}} \boldsymbol{u}\delta^{\iota}\ \bar{D}_\iota \tag{2.3}$$

$\bar{B}_\mu \in F^{k\times m}$, $\bar{C} \in F^{m\times n}$, $\bar{D}_\iota \in F^{k\times n}$. Let $\boldsymbol{\xi}'$ be another basis of $\mathcal{T}/\mathrm{span}_{F[\delta]}(\boldsymbol{u})$. Thus $\boldsymbol{\xi}' = \boldsymbol{\xi}P$, $P \in F^{m\times m}$, $\det(P) \neq 0$. Take a $m$-tuple $\boldsymbol{\eta}' = (\eta'_1, \ldots, \eta'_m)$ in $\mathcal{T}$ the image of which in $\mathcal{T}/\mathrm{span}_{F[\delta]}(\boldsymbol{u})$ is $\boldsymbol{\xi}'$. Then

$$\boldsymbol{\eta}' = \boldsymbol{\eta} + \sum_{\text{finite}} \boldsymbol{u}\delta^{\iota}\ Q_\iota \tag{2.4}$$

$Q \in F^{k\times m}$. Note that (2.4) is input-dependent. If, in (2.2), $\nu \geq 2$ and $\bar{B}_\nu \neq 0$, set

$$\boldsymbol{\eta} = \tilde{\boldsymbol{\eta}} - \boldsymbol{u}\delta^{\nu-1}\ (\bar{B}_\nu A^{-1}\delta^{-1}).$$

It yields

$$\tilde{\boldsymbol{\eta}} = \tilde{\boldsymbol{\eta}}\delta\ A + \sum_{\mu=0}^{\nu-1} \boldsymbol{u}\delta^{\mu}\ \tilde{B}_\mu.$$

If $\bar{B}_0 \neq 0$, setting

$$\tilde{\boldsymbol{\eta}} = \overline{\boldsymbol{\eta}} + \boldsymbol{u}\ \bar{B}_0$$

yields

$$\overline{\boldsymbol{\eta}} = \overline{\boldsymbol{\eta}}\delta + \sum_{\mu=1}^{\nu-1} \boldsymbol{u}\delta^{\mu}\ \overline{B}_\mu.$$

We have proved the following theorem which is a time-varying generalisation of [11].

**Theorem 2.5.** *A causal linear recurrent transducer may be given the* Kalman state-variable representation

$$\boldsymbol{x} = \boldsymbol{x}\delta\ A + \boldsymbol{u}\delta\ B \tag{2.5}$$

$$\boldsymbol{y} = \boldsymbol{x}\ C + \sum_{\text{finite}} \boldsymbol{u}\delta^{\iota}\ D_{\iota} \tag{2.6}$$

*where* $\boldsymbol{x} = (x_1, \ldots, x_m)$, $m = \dim_F(\mathcal{T}/\mathrm{span}_{F[\delta]}(\boldsymbol{u}))$, $A \in F^{m\times m}$, $\det A \neq 0$, $B \in F^{k\times m}$, $C \in F^{m\times n}$, $D_\iota \in F^{k\times m}$.

**Remark 2.6.** Setting $\boldsymbol{x} = \bar{\boldsymbol{x}} - \boldsymbol{u}\ (BA^{-1}\delta^{-1})$ yields $\bar{\boldsymbol{x}} = \bar{\boldsymbol{x}}\delta\ A + \boldsymbol{u}\ (BA^{-1}\delta^{-1})$ which might also be interesting in some applications.

## 2.4. Controllability and observability

### 2.4.1. *Controllability*

The transducer $\mathcal{T}$ is called *controllable* if, and only if, the module $\mathcal{T}$ free. The next result is an extension to (2.5) of the classic Kalman controllability criterion (compare with [43]):

**Proposition 2.7.** *The transducer $\mathcal{T}$ is controllable if, and only if, the matrix*

$$\left(B, B\delta A, \ldots, B(\delta A)^{m-1}\right)$$

*is of rank $m$.*

*Proof.* It is easy to check that $\mathrm{rk}(B, B\delta A, \ldots, B(\delta A)^{m-1}) < m$ is equivalent to the existence of a nontrivial torsion submodule of $\mathcal{T}$. □

### 2.4.2. *Observability*

The transducer $\mathcal{T}$ is called *observable* if, and only if, the modules $\mathcal{T}$ and $\mathrm{span}_{F[\delta]}(\boldsymbol{u}, \boldsymbol{y})$ coincide. The next result is an extension to (2.5, 2.6) of the classic Kalman observability criterion (compare with [43]):

**Proposition 2.8.** *The transducer $\mathcal{T}$ is observable if, and only if, the matrix*

$$\left({}^tC, {}^tC\delta\ {}^tA^{-1}, \ldots, {}^tC(\delta\ {}^tA^{-1})^{m-1}\right)$$

*where ${}^t\bullet$ indicates the transpose matrix, is of rank $m$.*

*Proof.* Utilize $\boldsymbol{x}\delta = \boldsymbol{x}\ A^{-1} - \boldsymbol{u}\delta\ BA^{-1}$ for expressing $\boldsymbol{y}\delta^{\iota}$, $\iota = 1, \ldots, m-1$, as $F$-linear combinations of the components of $\boldsymbol{x}$ and $\boldsymbol{u}\delta^{\kappa}$, $\kappa \geq 0$. □

**Remark 2.9.** By utilizing the inverse $A\delta^{-1}$ of $\delta A^{-1}$, Proposition 2.8 becomes

$$\mathrm{rk}\left({}^tC, {}^tC{}^tA\delta^{-1}, \ldots, {}^tC({}^tA\delta^{-1})^{m-1}\right) = m.$$

## 2.5. Transfer matrices

### 2.5.1. *Definition*

Let $F(\delta)$ be the quotient field of $F[\delta]$ which is a right Ore ring. The right $F(\delta)$-vector space $\hat{\mathcal{T}} = \mathcal{T} \otimes_{F[\delta]} F(\delta)$ is called the *transfer vector space* of $\mathcal{T}$ [12]. The $F[\delta]$-linear mapping $\mathcal{T} \to \hat{\mathcal{T}}$, $\tau \mapsto \hat{\tau} = \tau \otimes 1$, is the *(formal) Laplace transform* [12]. Its kernel is the torsion submodule of $T$. It is thus injective if, and only if, the module $\mathcal{T}$ is free. As $\boldsymbol{u}$ is independent, $\hat{\boldsymbol{u}} = (\hat{u}_1, \ldots, \hat{u}_k)$ is a basis of $\hat{\mathcal{T}}$. It yields

$$\hat{\boldsymbol{y}} = (\hat{y}_1, \ldots, \hat{y}_n) = \hat{\boldsymbol{u}}\ G \tag{2.7}$$

where $G \in F(\delta)^{m\times n}$ is the *rational transfer matrix*, or the *rational generating matrix*, of the transducer (compare with [28]). When $k = n = 1$, $G$ is called a *rational transfer*, or *generating function.*

**Remark 2.10.** Note that the dimension of $\hat{\mathcal{T}}$ is equal to the rank of $\mathcal{T}$.

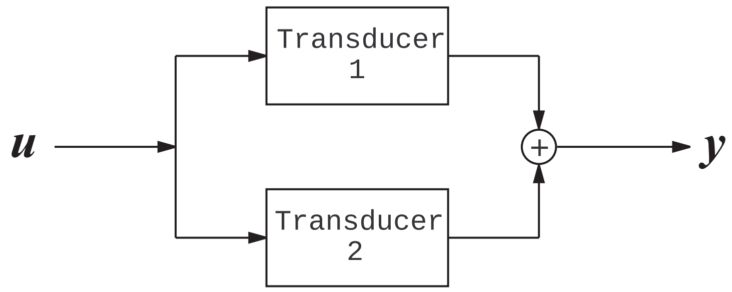
Any element of $F(\delta)$ may be written as a Laurent series $\sum_{\nu\geq\nu_0} \delta^\nu a_\nu$, $a_\nu \in F$, $\nu_0 \in \mathbb{Z}$. It is said to be *causal* if, and only if, $\nu_0 \geq 0$. The matrix $G$ is said to be *causal* if, and only if, all its entries are causal.

**Theorem 2.11.** *Any causal linear recurrent transducer possesses a rational causal transfer matrix. Conversely, any rational causal matrix is the transfer matrix of a causal linear recurrent transducer, which is controllable and observable.*

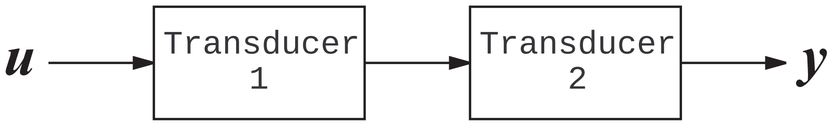
*Proof.* The first part is an immediate consequence of the definition of causality in Section 2.2 and of the input-output relation (2.7). For the second part, utilize the right coprime factorization $G = ND^{-1}$, $N \in F[\delta]^{k\times n}$, $D \in F[\delta]^{n\times n}$, where $D$ is invertible (see [12]). The transfer matrix of the transducer $\boldsymbol{y}D = \boldsymbol{u}N$, which is both controllable and observable (see [12]), is $G$. □

2.5.2. *Interconnection*

Let $h_\upsilon : \Sigma \to \mathcal{S}_\upsilon$, $\upsilon \in \Upsilon$, be a morphism of systems, *i.e.*, of finitely generated right $F[\delta]$-modules. The corresponding fibered sum is a *system interconnection* (*cf.* [14]). The parallel interconnection




and the series interconnection




are particular instances of system interconnections. The proof of the following result is straightforward.

**Proposition 2.12.** *The transfer matrix of the parallel (resp. series) interconnection of linear recurrent transducers is the sum (resp. product) of the transfer matrices.*

**Remark 2.13.** Interconnections as simple as those in Proposition 2.12 may lead to a lost of controllability or observability[6] which is not readable *via* transfer matrices [14].

## 2.6. **Input-output inversion**

2.6.1. *General results*

The *output rank* [8] of the transducer $\mathcal{T}$ is $\varrho = \mathrm{rk}(\mathrm{span}_{F[\delta]}(\boldsymbol{y}))$. Obviously $0 \leq \varrho \leq \min(k, n)$. The transducer $\mathcal{T}$ is said to be *right invertible* (resp. *left invertible*) if, and only if, $\varrho = k$ (resp. $\varrho = n$).

[6]The continuous-time examples and the results in [14] (see also the references therein) may trivially be adapted to our discrete-time context.

**Proposition 2.14.** *$\mathcal{T}$ is right invertible, if and only if, the quotient module $\mathcal{T}/\text{span}_{F[\delta]}(\boldsymbol{y})$ is torsion.*

*Proof.* $\text{rk}(\mathcal{T}/\text{span}_{F[\delta]}(\boldsymbol{y})) = \text{rk}(\mathcal{T}) - \varrho$. Since $\mathcal{T}/\text{span}_{F[\delta]}(\boldsymbol{u})$ is torsion, $\text{rk}(\mathcal{T}) = \text{rk}(\text{span}_{F[\delta]}(\boldsymbol{u}) = k$. Thus $\text{rk}(\mathcal{T}/\text{span}_{F[\delta]}(\boldsymbol{y})) = 0$ if, and only if, $\varrho = k$. □

In a more down to earth language, Lemma 2.14 means that $\boldsymbol{u}$ may be obtained from $\boldsymbol{y}$ thanks to difference equations. The example $y = u\delta$, where $k = n = 1$, shows that the right inverse transducer is not generally causal. Left invertibility means that the components of $\boldsymbol{y}$ are $F[\delta]$-linearly independent.

The next proposition is an immediate consequence of Remark 2.10.

**Proposition 2.15.** *The linear recurrent transducer $\mathcal{T}$ is right (resp. left) invertible if, and only if, its transfer matrix is right (resp. left) invertible.*

**Corollary 2.16.** *If the linear recurrent transducer $\mathcal{T}$ is right (resp. left) invertible, then $n \geq k$ (resp. $n \leq k$).*

If $k = n$, the transducer is said to be *square*. Then right and left invertibilities coincide. An *invertible* square transducer is right and left invertible.

2.6.2. *Encoders*

A linear recurrent transducer, which is right invertible, is called a *linear recurrent encoder*, or a *(time-varying) convolutional encoder*. If $F$ is a field of constants, it is called a *(fixed) convolutional encoder*[7]. A square encoder is called a *linear recurrent encrypter*.

## 2.7. Some useful constructions

2.7.1. *Blocking*

For any integer $\Omega > 1$, $F[\delta^\Omega] \subset F[\delta]$. Thus any right $F[\delta]$-module $\boldsymbol{M}$ may also be viewed as a right $F[\delta^\Omega]$-module $\boldsymbol{M}_\Omega$ called the $\Omega^{\text{th}}$-*blocking*, or $\Omega^{\text{th}}$-*interleaving*, module.

**Lemma 2.17.** $\text{rk}(\boldsymbol{M}_\Omega) = \Omega\ \text{rk}(\boldsymbol{M})$.

*Proof.* If $\xi_1, \ldots, \xi_\ell$ are $F[\delta]$-linearly independent elements in $\boldsymbol{M}$, then $\xi_1, \xi_1\delta, \ldots, \xi_1\delta^{\Omega-1}, \ldots, \xi_\ell, \xi_\ell\delta, \ldots, \xi_\ell\delta^{\Omega-1}$ are $F[\delta^\Omega]$-linearly independent. □

The $\Omega^{\text{th}}$-blocking transducer, or $\Omega^{\text{th}}$-interleaving transducer, $\mathcal{T}_\Omega$ of $\mathcal{T}$ is the linear recurrent transducer defined by

- its module is the $\Omega^{\text{th}}$-blocking module $\mathcal{T}_\Omega$;
- its input and output are respectively $(\boldsymbol{u}, \boldsymbol{u}\delta, \ldots, \boldsymbol{u}\delta^{\Omega-1})$ and $(\boldsymbol{y}, \boldsymbol{y}\delta, \ldots, \boldsymbol{y}\delta^{\Omega-1})$.

The next result is clear:

**Proposition 2.18.** *If $\mathcal{T}$ is controllable (resp. observable, right invertible, left invertible), then $\mathcal{T}_\Omega$ is also controllable (resp. observable, right invertible, left invertible).*

2.7.2. *Puncturing*

*Puncturing* a linear recurrent transducer $\mathcal{T}$ means taking a linear recurrent transducer $\mathcal{T}_P$ defined by the same module, the same input and by an output which is a subset of $\boldsymbol{y}$. The next result is clear:

**Proposition 2.19.** *If $\mathcal{T}$ is controllable (resp. left invertible), then $\mathcal{T}_P$ is also controllable (resp. left invertible). If $\mathcal{T}$ is observable (resp. right invertible), then $\mathcal{T}_P$ is not necessarily observable (resp. right invertible).*

[7]Even if $F$ is a finite field, there exists several definitions of convolutional encoders in the existing literature.

## 3. Some properties of linear recurrent codes

### 3.1. Equivalence of encoders and codes

3.1.1. *Equivalence*

Two linear recurrent encoders with inputs $\boldsymbol{u} = (u_1, \dots, u_k)$, $\boldsymbol{u}' = (u_1, \dots, u'_{k'})$ and outputs $\boldsymbol{y} = (y_1, \dots, y_n)$, $\boldsymbol{y}' = (y'_1, \dots, y'_{n'})$ are said to be *equivalent* if, and only if, the following conditions are satisfied:

(1) $n = n'$;

(2) there exists $\sigma \in \mathrm{S}_n$, where $\mathrm{S}_n$ is the symmetric group over $\{1, \dots, n\}$, such that the mapping $y_\iota \mapsto y'_{\iota\sigma}$, $\iota = 1, \dots, n$, defines an isomorphism between the modules $\mathrm{span}_{F[\delta]}(\boldsymbol{y})$ and $\mathrm{span}_{F[\delta]}(\boldsymbol{y}')$.

**Proposition 3.1.** *The inputs of two equivalent linear recurrent encoders possess the same number of components.*

*Proof.* Let $\varrho$ and $\varrho'$ be the output ranks of the encoders $\mathcal{T}$ and $\mathcal{T}'$. The right invertibility of $\mathcal{T}$ and $\mathcal{T}'$ implies $\varrho = k$ and $\varrho' = k'$. The equivalence of $\mathcal{T}$ and $\mathcal{T}'$ implies $\varrho = \varrho'$. ☐

3.1.2. *Codes*

A *linear recurrent code*, or a *(time-varying) convolutional code* is an equivalence class between linear recurrent encoders. From Proposition 3.1, we know already two integers $k$, $n$, $0 < k \leq n$ which are attached to the code, which is therefore called a $(n, k)$ linear recurrent code. Its *rate* is $\frac{k}{n}$. By a slight abuse of language, $\mathrm{span}_{F[\delta]}(\boldsymbol{y})$ is sometimes called a linear recurrent code, or a (time-varying) convolutional code. When $F$ is a finite field of constants, a linear recurrent code is called a *(fixed) convolutional code.* A code is said to be *free*, or *controllable*, if, and only if, the module $\mathrm{span}_{F[\delta]}(\boldsymbol{y})$ is free.

### 3.2. Syndrome formers

Let $\mathcal{F}_n$ be the free right $F[\delta]$-module, with basis $\bar{y}_1, \dots, \bar{y}_n$. The mapping $\bar{y}_\iota \mapsto y_\iota$, $\iota = 1, \dots, n$, defines an epimorphism $\mathcal{F}_n \to \mathrm{span}_{F[\delta]}(\boldsymbol{y})$ and the short exact sequence

$$0 \to \mathcal{F}_{n-k} \to \mathcal{F}_n \to \mathrm{span}_{F[\delta]}(\boldsymbol{y}) \to 0 \tag{3.1}$$

where $\mathcal{F}_{n-k}$ a free right $F[\delta]$-module of rank $n-k$. A *syndrome former* of the code is a presentation matrix of $\mathrm{span}_{F[\delta]}(\boldsymbol{y})$, which corresponds here to the monomorphism $\mathcal{F}_{n-k} \to \mathcal{F}_n$.

The sequence (3.1) splits, *i.e.*, $\mathcal{F}_n \simeq \mathcal{F}_{n-k} \oplus \mathrm{span}_{F[\delta]}(\boldsymbol{y})$, if, and only if, the code is free.

### 3.3. Some properties of free codes

From now on and until the end of the paper codes are assumed to be free[8]. When $F$ is a finite field of constants, a (fixed) convolutional code may be defined as a certain $F[\delta]$-submodule of the $F[\delta]$-module $\mathcal{L} = \{\sum_{\upsilon \geq 0} \delta^\upsilon a_{1\upsilon}, \dots, \sum_{\upsilon \geq 0} \delta^\upsilon a_{n\upsilon}\}$ of $n$-tuple of formal power series. The relationship with our approach[9] is given the $F[\delta]$-module $\mathrm{Hom}\left(\mathrm{span}_{F[\delta]}(\boldsymbol{y}), \mathcal{L}\right)$ of $F[\delta]$-module morphisms $\Phi = (\phi_1, \dots, \phi_n) : \mathrm{span}_{F[\delta]}(\boldsymbol{y}) \to \mathcal{L}$, $(y_1, \dots, y_n) \mapsto (y_1\phi, \dots, y_n\phi)$ (compare with [38]).

3.3.1. *Dual codes and parity check matrices*

The image of $\mathcal{F}_{n-k}$ in $\mathcal{F}_n$ is called the *dual code.* A syndrome former of the dual code is called a *parity check matrix* of the code.

**Remark 3.2.** When $F$ is a finite field of constants, the dual code of a convolutional code is usually defined as for block codes *via* an orthogonality relation. We leave to the reader to construct explicitly the relationship with our definition.

[8]When $F$ is a finite field of constants, a (fixed) convolutional code is often defined as a vector subspace of $F(\delta)^{1\times n}$ (see, *e.g.* [26, 34]). With respect to this transfer matrix setting the freeness may always be assumed.

[9]This is more generally the relationship (see [10]) between our module-theoretic setting and Willems' *behavioral approach* [44].

3.3.2. *Polynomial and basic encoders*

A controllable and observable encoder $\mathcal{E}$ is said to be *polynomial* if, and only if, $\boldsymbol{u}$ is a basis of the free module $\mathcal{E}$. The next property is an immediate consequence of Theorem 2.11:

**Proposition 3.3.** *A controllable and observable encoder is polynomial if, and only if, the entries of its transfer matrix are polynomial, i.e., belong to* $F[\delta]$.

The polynomial encoder $\mathcal{E}$ is said to be *basic* if, and only if, $\mathcal{E} = \mathrm{span}_{F[\delta]}(\boldsymbol{y})$. By taking for $\boldsymbol{u}$ any basis of the free module $\mathrm{span}_{F[\delta]}(\boldsymbol{y})$ we obtain the

**Proposition 3.4.** *Any free code admits a basic encoder.*

3.3.3. *Systematic encoders*

**Proposition 3.5.** *Any free code admits a* systematic *encoder, i.e., an encoder where* $k$ *components of the output are identical to the* $k$ *components of the input.*

*Proof.* The result is clear if $k = n$; $\boldsymbol{y}$ is a basis of $\mathrm{span}_{F[\delta]}(\boldsymbol{y})$ and can be taken as an input. Assume that the result holds for $n = n_0 \geq k$. Take $n = n_0 + 1$. Since the components of $\boldsymbol{y}$ are $F[\delta]$-linearly dependent we may write

$$y_1\gamma_1 + \cdots + y_{n_0+1}\gamma_{n_0+1} = 0 \tag{3.2}$$

where $\gamma_1, \ldots, \gamma_{n_0+1} \in F[\delta]$ are right coprime. At least one of the coefficients $\gamma_\iota$, $\iota = 1, \ldots, n_0 + 1$, $\gamma_{n_0+1}$ for instance, when expressed as a sum (2.1), is such that $a_0 \neq 0$. Apply the assumption to the code spanned by $y_1, \ldots, y_{n_0}$ and utilise the causal relation $y_{n_0+1} = -(y_1\gamma_1 + \cdots + y_{n_0}\gamma_{n_0})\gamma_{n_0+1}^{-1}$. □

3.3.4. *Non-catastrophic encoders*

The ring of Laurent polynomials $F[\delta, \delta^{-1}]$ is the localized ring of $F[\delta]$ by the multiplicative monoid $\{\delta^s \mid s \geq 0\}$, which satisfies the right Ore condition. The corresponding localized right $F[\delta, \delta^{-1}]$-module $\mathcal{E} \otimes_{F[\delta]} F[\delta, \delta^{-1}]$ of $\mathrm{span}_{F[\delta]}(\boldsymbol{u})$ is free, if $\mathcal{E}$ is controllable. The canonical mapping $\mathcal{E} \to \mathcal{E} \otimes_{F[\delta]} F[\delta, \delta^{-1}]$, $\upsilon \mapsto \upsilon \otimes 1$, being injective, $\mathcal{E}$ may be considered as a subset of $\mathcal{E} \otimes_{F[\delta]} F[\delta, \delta^{-1}]$. A controllable encoder is said to be *non-catastrophic* if, and only if, $\boldsymbol{u}$ belongs to $\mathrm{span}_{F[\delta]}(\boldsymbol{y}) \otimes_{F[\delta]} F[\delta, \delta^{-1}]$. The next result is an immediate consequence of Proposition 3.4.

**Proposition 3.6.** *Any free code admits a non-catastrophic encoder.*

## 3.4. Forney's theorem

3.4.1. *An important filtration*

Define a *filtration* of $F[\delta]$ by setting $\mathbf{F}_\alpha = \{P\delta^\alpha\}$, $\alpha \geq 0$, $P \in F[\delta]$. Thus $F[\delta] = \mathbf{F}_0 \supset \mathbf{F}_1 \supset \ldots$. The corresponding filtration for the free module $\mathrm{span}_{F[\delta]}(\boldsymbol{y})$ is obtained by setting $\mathbf{C}_\alpha = \mathrm{span}_{F[\delta]}(\boldsymbol{y})\mathbf{F}_\alpha$. Thus $\mathrm{span}_{F[\delta]}(\boldsymbol{y}) = \mathbf{C}_0 \supset \mathbf{C}_1 \supset \ldots$. Any element $x \in \mathrm{span}_{F[\delta]}(\boldsymbol{y})$ may be written uniquely as a finite sum

$$x = \sum_{\alpha=\nu}^{\mu} \xi_\alpha \delta^\alpha \tag{3.3}$$

where $\xi_\alpha\delta^\alpha$ is *homogeneous*, of *weight* $\alpha$ ($\xi_0$ is homogeneous of weight 0). The element $x$ is said to be of *order* $\nu$ (resp. *degree* $\mu$) if, and only if, $\xi_\nu \neq 0$ (resp. $\xi_\mu \neq 0$). It is homogeneous if, and only if, $\nu = \mu$. The next results are clear.

**Lemma 3.7.** *The semi-linear linear mapping* $\delta^\ell : \boldsymbol{C}_\alpha \to \boldsymbol{C}_{\alpha+\ell}$, $\ell > 0$, *is bijective.*

**Corollary 3.8.** *For any homogeneous element* $x_{\alpha+\ell}$ *of order* $\alpha + \ell$ *there exists a homogeneous element* $x_\alpha$ *of order* $\alpha$ *such that* $x_\alpha\delta^\ell = x_{\alpha+\ell}$.

**Lemma 3.9.** *Homogeneous elements of order $\nu$ are $F[\delta]$-linearly independent if, and only if, they are $F$-linearly independent.*

**Corollary 3.10.** *The $F$-vector space $\boldsymbol{C}_\alpha/\boldsymbol{C}_{\alpha+1}$ is of dimension $k$.*

3.4.2. *The result*

Let $\varepsilon_1$ be the highest degree of the components of $\boldsymbol{y}$, when written as in (3.3). Let $V_1$ be the $\varpi_1$-dimensional $F$-vector space spanned by the corresponding homogeneous elements. Choose according to Corollary 3.8 homogeneous elements $u_1, \ldots, u_{\varpi_1}$, of degree 0, such that $V_1 = \mathrm{span}(u_1\delta^{\varepsilon_1}, \ldots, u_{\varpi_1}\delta^{\varepsilon_1})$. Let $\varepsilon_2 < \varepsilon_1$ be the first integer such that $u_1\delta^{\varepsilon_2}, \ldots, u_{\varpi_1}\delta^{\varepsilon_2}$ does not span the $F$-vector space spanned by the homogeneous components of order $\varepsilon_2$ in $\boldsymbol{y}$. Complete then $u_1, \ldots, u_{\varpi_1}$ as above. We obtain a basis $\boldsymbol{u} = (u_1, \ldots, u_m)$ and a corresponding polynomial transfer matrix with lines of degrees[10] $e_1 \leq e_2 \leq \cdots \leq e_k$.

We must show that the above basic encoder is *minimal*, *i.e.*, that the degrees $f_1 \leq f_2 \leq \cdots \leq f_k$ of the lines of any polynomial generating matrix verify $e_\iota \leq f_\iota$, $\iota = 1, \ldots, k$. The next lemma, which is obvious, demonstrates that this result holds true if $k = 1$.

**Lemma 3.11.** *Take a free $F[\delta]$-module $M$ of rank 1. Two bases $b$ and $b'$ are related by $b = b'\gamma$, $\gamma \in F$, $\gamma \neq 0$. Let $N \supseteq M$ be another free $F[\delta]$-module of rank 1. Then, for any basis $c$ of $N$, $b = b\pi$, $\pi \in F[\delta]$.*

By considering the quotient module $\mathrm{span}_{F[\delta]}(\boldsymbol{y})/\mathrm{span}_{F[\delta]}(u_1)$, which is free of rank $k-1$, we obtain the minimality for any $k \geq 2$, assuming that it holds true for $k-1$.

We have proved:

**Theorem 3.12.** *For any free linear recurrent code, there exists a basic encoder, called* minimal, *such that the degrees of the lines of its transfer matrix are $e_1 \leq e_2 \leq \cdots \leq e_k$. The degrees $f_1 \leq f_2 \leq \cdots \leq f_k$ of the lines of a transfer matrix of any equivalent polynomial encoder verify $e_\kappa \leq f_\kappa$, $\kappa = 1, \ldots, k$.*

A corresponding input is called a *Forney input*.

## 4. A connection between convolutional and block codes

From now on $F$ is a finite field $\mathbb{F}_q$ of constants. We will therefore be working with free (fixed) linear convolutional codes.

### 4.1. Sliding block codes

A *sliding presentation* of a free $(n, k)$ linear convolutional code is given by a submodule $\mathrm{C}$ of rank $k$ of a free $\mathbb{F}_q[\delta]$-module $\mathrm{E}$ of rank $n$ such that the quotient module $\mathrm{E}/\mathrm{C}$ is free[11]. The *sliding (linear) block code of order* $\Omega$ of a given sliding presentation is given by the $\mathbb{F}_q$-vector subspace $\mathrm{C}/\mathrm{C}\delta^\Omega$ of the $\mathbb{F}_q$-vector space $\mathrm{E}/\mathrm{E}\delta^\Omega$. It is obviously a $(n\Omega, k\Omega)$ block code.

**Theorem 4.1.** *For integers $n, k, \Omega$, $1 \leq k < n$, $\Omega \geq 1$, there exists a free $(n, k)$ convolutional code with a sliding presentation such that its sliding block code of order $\Omega$ is an arbitrary $(n\Omega, k\Omega)$ block code.*

*Proof.* Take an arbitrary $(n\Omega, k\Omega)$ block code defined by a $k\Omega$-dimensional subspace $U$ of a $n\Omega$-dimensional of a $\mathbb{F}_q$-vector space $Y$. For any integer $\nu \geq 0$, set $Y\delta^{\nu\Omega} = \{y\delta^{\nu\Omega} \mid y \in E\}$. Define the free $\mathbb{F}_q[\delta^\Omega]$-modules $\mathrm{U} = \bigoplus_{\nu \geq 0} U\delta^{\nu\Omega} \subset \mathrm{Y} = \bigoplus_{\nu \geq 0} Y\delta^{\nu\Omega}$. Consider now the free $\mathbb{F}_q[\delta]$-modules $\mathbb{F}_q[\delta] \bigotimes_{\mathbb{F}_q[\delta^\Omega]} \mathrm{U} \subset \mathbb{F}_q[\delta] \bigotimes_{\mathbb{F}_q[\delta^\Omega]} \mathrm{Y}$.

Any basis $\underline{\boldsymbol{u}} = (\underline{u}_1, \ldots, \underline{u}_{k\Omega})$ of $U$ may be viewed as a basis of $\mathrm{U}$. By considering a systematic presentation of the block code, we may complete some basis $\underline{\boldsymbol{u}}$ as a basis $\underline{\boldsymbol{y}} = (\underline{y}_1, \ldots, \underline{y}_{n\Omega})$ of $Y$; $\underline{\boldsymbol{y}}$ may also be viewed as a basis of $\mathrm{Y}$. Take a partition of $\underline{\boldsymbol{u}}$ consisting of $k$ disjoint sets of $\Omega$ elements. Complete it as a partition of $\underline{\boldsymbol{y}}$ of $n$ disjoint sets of $\Omega$ elements. For the subsets $\{(z_1^\iota, \ldots, z_\Omega^\iota) \mid \iota = 1, \ldots, n$ of the partition, define the submodule $\mathrm{P} = \mathrm{span}_{\mathbb{F}_q[\delta]}(\{z_\kappa^\iota\delta - z_{\kappa+1}^\iota \mid \iota = 1, \ldots, n; \kappa = 1, \ldots, \Omega - 1\})$ of $\mathbb{F}_q[\delta] \bigotimes_{\mathbb{F}_q[\delta^\Omega]} \mathrm{U} \subset \mathbb{F}_q[\delta] \bigotimes_{\mathbb{F}_q[\delta^\Omega]} \mathrm{Y}$.

[10] The degree of a line is the maximum degree of its entries.

[11] This is an immediate consequence of the splitting property of the short exact sequence (3.1).

**Lemma 4.2.** *The quotient $\mathbb{F}_q[\delta]$-module $E = \mathbb{F}_q[\delta] \bigotimes_{\mathbb{F}_q[\delta^\Omega]} Y/P$ is free of rank n. The canonical image C of $\mathbb{F}_q[\delta] \bigotimes_{\mathbb{F}_q[\delta^\Omega]} U$ into $\mathbb{F}_q[\delta] \bigotimes_{\mathbb{F}_q[\delta^\Omega]} Y/P$ is free of rank k.*

*Proof.* A basis of E is $\{z_1^\iota \mid \iota = 1, \ldots, n\}$. □

The solution is given by the sliding presentation C ⊂ E. □

## 4.2. Sketch of a feedback decoding procedure

The $\mathbb{F}_q[\delta, \delta^{-1}]$-modules $\mathbb{F}_q[\delta, \delta^{-1}] \otimes_{\mathbb{F}_q[\delta]}$ C and $\mathbb{F}_q[\delta, \delta^{-1}] \otimes_{\mathbb{F}_q[\delta]}$ E may be seen as $\mathbb{F}_q[\delta]$-modules $\bar{\mathrm{C}}$ and $\bar{\mathrm{E}}$. For any integer $\alpha \geq 0$, the $(\alpha+1)^{\text{th}}$ frame[12] is the block code is given by the $\mathbb{F}_q$-vector subspace $\bar{\mathrm{C}}\delta^{(-\alpha-\Omega)}/\bar{\mathrm{C}}\delta^{-\alpha}$ of the $\mathbb{F}_q$-vector space $\bar{\mathrm{E}}\delta^{(-\alpha-\Omega)}/\bar{\mathrm{E}}\delta^{-\alpha}$. It is clear that all those frames are isomorphic to the sliding block code of order $\Omega$.

A decoding procedure of any frame will of course take advantage of the nature of the sliding block code. Comparing the results for the $\beta^{\text{th}}$ and $(\beta+\ell)^{\text{th}}$, $\beta, \ell \geq 1$, permits some checking if $\ell < \Omega$. This *feedback* type decoding of the convolutional code may be enriched by some *concatenations* (see, *e.g.* [2,4,30]) of the frames.

## References


[1] C. Berrou and A. Glavieux, Near-optimum error-correcting coding and decoding: Turbo-codes. *IEEE Trans. Communicat.* **44** (1996) 1261-1271.
[2] R.E. Blahut, *Theory and Practice of Error Control Codes.* Addison-Wesley (1983).
[3] N. Bourbaki, *Algèbre*, Chap. 2. Hermann (1970).
[4] G. Cohen, J.-L. Dornstetter and P. Godlewski, *Codes correcteurs d'erreurs.* Masson (1992).
[5] R.M. Cohn, *Difference Algebra.* Interscience (1965).
[6] A. Dholakia, *Introduction to Convolutional Codes with Applications.* Kluwer (1994).
[7] F. Fagnani and S. Zampieri, System-theoretic properties of convolutional codes over rings. *IEEE Trans. Inform. Theory* **47** (2001) 2256-2274.
[8] M. Fliess, Automatique en temps discret et algèbre aux différences. *Forum Math.* **2** (1990) 213-232.
[9] M. Fliess, Some basic structural properties of generalized linear systems. *Systems Control Lett.* **15** (1990) 391-396.
[10] M. Fliess, A remark on Willems' trajectory characterization of linear controllability. *Systems Control Lett.* **19** (1992) 43-45.
[11] M. Fliess, Reversible linear and nonlinear discrete-time dynamics. *IEEE Trans. Automat. Control* **37** (1992) 1144-1153.
[12] M. Fliess, Une interprétation algébrique de la transformation de Laplace et des matrices de transfert. *Linear Algebra Appl.* **203-204** (1994) 429-442.
[13] M. Fliess, Variations sur la notion de contrôlabilité, in *Journée Soc. Math. France.* Paris (2000) 47-86.
[14] M. Fliess and H. Bourlès, Discussing some examples of linear system interconnections. *Systems Control Lett.* **27** (1996) 1-7.
[15] M. Fliess, J. Lévine, P. Martin and P. Rouchon, Flatness and defect of non-linear systems: Introductory theory and applications. *Internat. J. Control* **61** (1995) 1327-1361.
[16] M. Fliess and R. Marquez, Continuous-time linear predictive control and flatness: A module-theoretic setting with examples. *Internat. J. Control* **73** (2000) 606-623.
[17] M. Fliess and R. Marquez, Une approche intrinsèque de la commande prédictive linéaire discrète. *APII J. Europ. Syst. Automat.* **35** (2001) 127-147.
[18] M. Fliess, R. Marquez, E. Delaleau and H. Sira–Ramírez, Correcteurs proportionnels-intégraux généralisés. *ESAIM: COCV* **7** (2002) 23-41.
[19] M. Fliess, R. Marquez and H. Mounier, An extension of predictive control, PID regulators and Smith predictors to some linear delay systems. *Internat. J. Control* (to appear).
[20] M. Fliess and H. Mounier, Controllability and observability of linear delay systems: An algebraic approach. *ESAIM: COCV* **3** (1998) 301-314.
[21] G.D. Forney Jr., Convolutional codes I: Algebraic structure. *IEEE Trans. Inform. Theory* **16** (1970) 720-738.
[22] G.D. Forney Jr., Minimal bases of rational vector spaces, with applications to multivariable linear systems. *SIAM J. Control* **13** (1975) 493-520.
[23] G.D. Forney Jr., Algebraic structure of convolutional codes and algebraic system theory, in *Mathematical System Theory – The Influence of R.E. Kalman*, edited by A.C. Antoulas. Springer (1991) 527-557.


[12]This mathematical definition of frames, where the previous codewords are subtracted, differs from the usual meaning in the literature (see, *e.g.* [2]). Note however that this substraction of previous codewords is asserted in [2] to be of utmost importance in feedback decoding.

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

[25] G.D. Forney Jr., B. Marcus, N.T. Sindhushayana and M. Trott, A multilingual dictionary: System theory, coding theory, symbolic dynamics and automata theory, in *Different Aspects of Coding Theory. Proc. Symp. Appl. Math.* **50**; *Amer. Math. Soc.* (1995) 109-138.

[26] R. Johannesson and K.Sh. Zigangirov, *Fundamentals of Convolutional Coding*. IEEE Press (1999).

[27] T. Kailath, *Linear Systems*. Prentice-Hall (1979).

[28] E.W. Kamen, P.P. Khargonekar and K.R. Poola, A transfer-function approach to linear time-varying discrete-time systems. *SIAM J. Control Optim.* **23** (1985) 550-565.

[29] T.Y. Lam, *Lectures on Rings and Modules*. Springer (1999).

[30] S. Lin and D.J. Costello Jr., *Error Control Coding: Fundamentals and Applications*. Prentice-Hall (1983).

[31] J.H. van Lint, *Introduction to Coding Theory*, 3$^{\text{rd}}$ Edition. Springer (1999).

[32] H.-A. Loeliger, G.D. Forney Jr., T. Mittelholzer and M.D. Trott, Minimality and observability of group systems. *Linear Algebra Appl.* **205-206** (1994) 937-963.

[33] J.L. Massey and M.K. Sain, Codes, automata and contnuous systems: Explicit interconnections. *IEEE Trans. Automat. Control* **12** (1967) 644-650.

[34] R.J. McEliece, The algebraic theory of convolutional codes, in *Handbook of Coding Theory*, Vol. 1, edited by V. Pless and W.C. Huffman. Elsevier (1998) 1065-1138.

[35] J.C. McConnel and J.C. Robson, *Noncommutative Noetherian Rings*. Wiley (1987).

[36] H. Mounier, P. Rouchon and J. Rudolph, Some examples of linear systems with delays. *APII J. Europ. Syst. Automat.* **31** (1997) 911-925.

[37] P. Piret, *Convolutional Codes, an Algebraic Approach*. MIT Press (1988).

[38] J. Rosenthal, Connections between linear systems and convolutional codes, in *Codes, Systems and Graphical Models*, edited by B. Marcus and J. Rosenthal. Springer (2000) 39-66.

[39] J. Rosenthal, J.M. Schumacher and E.V. York, On behaviors and convolutional codes. *IEEE Trans. Informat. Theory* **42** (1996) 1881-1891.

[40] J. Rosenthal and E.V. York, BCH convolutional codes. *IEEE Trans. Inform. Theory* **45** (1999) 1833-1844.

[41] J. Rotman, *An Introduction to Homological Algebra*. Academic Press (1979).

[42] A.J. Viterbi and J.K. Omura, *Principles of Digital Communication and Coding*. McGraw-Hill (1979).

[43] L. Weiss, Controllability, realization and stability of discrete-time systems. *SIAM J. Control* **10** (1972) 230-251.

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

[45] G. Zémor, *Cours de cryptographie*. Cassini (2000).