A Study on Convolutional Codes. Classification, New Families and Decoding

José Ignacio Iglesias Curto

Tesis Doctoral dirigida por
José María Muñoz Porras
José Ángel Domínguez Pérez

Departamento de Matemáticas
Universidad de Salamanca

A Study on Convolutional Codes. Classification, New Families and Decoding

Tesis presentada por

José Ignacio Iglesias Curto

para optar al título de

Doctor por la Universidad de Salamanca

José Ignacio Iglesias Curto, Doctorando

José María Muñoz Porras, Director

José Ángel Domínguez Pérez, Codirector

Departamento de Matemáticas
Universidad de Salamanca

Agradecimientos

Mi primer agradecimiento es para mis padres, José y Petra, y para mis hermanos, Pedro y M^{a} Eugenia, que en todos estos años me han cuidado y apoyado y de cuyos esfuerzos me he servido para desarrollar mi formación y poder dedicarme a la elaboración de este trabajo. Y para el resto de mi familia, que han seguido muy de cerca mi esfuerzo en el desarrollo del mismo.

Éste no hubiera sido posible sin la dirección, consejo y orientación de José María Muñoz Porras, a quien debo agradecer la atención, la ayuda y la confianza prestadas en todo este tiempo.

También quiero agradecer a Francisco José Plaza, José Ángel Domínguez, Gloria Serrano y Esteban Gómez su colaboración, sus correcciones y sus observaciones, que me han ayudado enormemente. Así como al resto de profesores y personal del Departamento de Matemáticas de la Universidad de Salamanca, que directa o indirectamente han facilitado mi trabajo y creado un entorno distendido donde poder llevarlo a cabo.

A part of this work was carried out as a guest in other research groups. For this reason I want to thank Gerard van der Geer and specially Joachim Rosenthal and Uwe Helmke, who hosted me in their groups and took care of me both personally and professionally. Ich bedanke mich auch bei meinen Kollegen und Freunden am Mathematischen Institut der Universität Würzburg für die freundliche Unterstutzung, die Zusammenarbeit und die nette Zeit.

Besides, I want to thank the advice and help of H. Wiemer and P. Fuhrmann who paid attention to my questions and provided me with valuable ideas.

In all this time it was quite worthful the help and understanding of all the friends who have shared with me the experience of writing a PhD thesis, as well as the concerns but also the gratification of research life. I thank them for their support and attention and I wish to those who still didn't finish a prompt success.

Mi agradecimiento también se dirige por supuesto a todos mis amigos, que sin excepción se han interesado constantemente por la marcha de mi trabajo. And I thank most sincerely the friends in Würzburg, a colorful group who have been for me, in every sense, a family.

Finally, I want to thank the Spanish Ministry of Education, the CTS Marie Curie Training Network and the german DAAD service who provided me with financial support to carry out this work.

Contents

A mis padres y
a mis hermanos

Chapter 1

Introduction to Coding Theory

1.1 Motivation and Development of Coding Theory

Information is one of the most valuable goods of our time. However, the physical means used to transmit and store information are never perfect and they are subject to errors that might result in loss of important data. Error correcting codes are a key element in the transmission and storage of digital information. The fact that a scratched CD can still be used or that power-limited devices in spacecrafts allow safe communication over large distances is due to the use of codes which enable to correct the errors and erasures that may happen in noisy channels and physical devices.

In the past 30 years, research in areas related to information has developed a wide range of sophisticated mathematical techniques that allow the implementation of robust and time optimal coding and decoding schemes in current and future communication technologies. From the CRC codes used in the communications between components of a computer, to the concatenation of block and convolutional codes used for deep-space transmissions, including also bar codes, the ISBN code for books, and the ones used for credit cards or identity cards, there is a wide range of different classes of codes. The code with the correction capacities that best fit the reliability of the physical devices is used in each instance of information processing. One of these classes of codes are Turbo Codes (which combine two convolutional codes), used in CDMA2000 1x digital cellular technology and its variation for internet access 1xEV-DO. Another well-known class are Reed-Solomon codes, which have applications in the storage of data on CDs and DVDs, in cellular technologies and in protocols for digital TV and radio [RR97].

To reach secure transmission of information, the communication process is formally described in mathematical terms. For every different code, there is an encoding map from the set of information messages to the set of all the codewords. This map adds to the information some redundancy, which will allow to detect and correct the errors that might happen. Once the codeword is sent, if the received message is not any of the codewords then an error is detected. This error is assumed to be the

smallest possible. The decoding process assigns to any received message a codeword having maximum probability of being the sent one.

The origin of coding theory dates back to the works of Claude Shannon and Richard Hamming in the Bell Laboratories. Shannon developed a communication theory and in his 1948 seminal paper "A Mathematical Theory of Communication" [Sha48] he showed that it is always possible to encode a message so that it can be sent with maximal reliability and minimal redundancy. However the proof was not constructive and the actual codes with those capacities were not given. On his side, Hamming published [Ham50] the first well known code construction, which was already cited in Shannon's paper. This code allowed the correction of one error and added significantly less redundancy than the repetition code.

In 1949, Marcel Golay proposed Golay codes for forward error correction [Gol49], and in 1954 Irving S. Reed [Ree54] and D.E. Muller [Mul54] introduced the well-known Reed-Muller codes.

All these families of codes were block codes. The conceptual jump to convolutional codes was made by Peter Elias in 1955 [Eli55].

Two other important families of block codes appeared in the following years: BCH codes discovered independently by Alexis Hocquenghem in 1959 [Hoc59] and by Raj Chandra Bose and Dwijendra Kumar Ray-Chaudhuri in 1960 [BC60b, BC60a], as well as Irving S. Reed and Gustave Solomon's famous Reed-Solomon codes [RS60] also in 1960. Although these codes were theoretically very powerful, there was no decoding algorithm which enabled their practical use. However in 1967 Elwyn Berlekamp invented an algorithm to find the minimal polynomial of a linearly recurrent sequence [Ber67] and short after James L. Massey observed its connection with decoding of linear block codes [Mas69], giving origin to the celebrated Berlekamp-Massey decoding algorithm and its use for the technical application of BCH and Reed-Solomon codes.

In the meanwhile, in 1962 Robert G. Gallager proposed the family of low-density parity-check (LDPC) codes which would be forgotten for about 30 years due to technical limitations, but which would arise again as one of the most powerful families of codes.

Also convolutional coding developed, with the invention by Andrew Viterbi in 1967 of the Viterbi algorithm, [Vit67], which made decoding of general convolutional codes of low degree practicable.

Some years later, in 1977, V. D. Goppa presented a surprising new construction of codes [Gop77]. It was surprising on the one hand because this family beat the Gilbert-Varshamov asymptotic bound on the minimum distance, which by that time it was thought it couldn't be improved. On the other hand, its construction was based on algebraic geometric tools, which allowed to compute the dimensions and the minimum distances, making an unexpected link between the most pure and applied mathematics. Goppa codes were the first of the many Algebraic Geometric code constructions that followed them.

In 1993 Claude Berrou, Alain Glavieux and Punya Thitimajshima introduced

Turbo codes [GBT93], a family of codes with a performance very close to the channel capacity.

From 1996 on, when David J.C. MacKay and Radford M. Neal, recovered Gallager's low-density parity-check codes [MN96], new constructions of this kind of codes attained closer and closer performances to the channel capacity. As a result, the task of finding the codes predicted by Shannon is considered achieved for any practical purpose.

Decoding of linear block codes experienced an enormous improvement with the introduction in 1997 of the first algorithm for List Decoding by Madhu Sudan [Sud97]. This decoding scheme allows as output more than one possible codeword but permits instead decoding more errors than half the minimum distance of the code.

At present, coding theory is a very active area of research. Still many tasks related to the constructions of asymptotically good families of codes, new mathematical constructions of block and convolutional codes, practical decoding algorithms,... and relationships of coding theory with other branches of mathematics as algebra, algebraic geometry, linear systems theory, complexity theory, cryptography, and even theoretical computer science, attract the attention and efforts of many researchers.

1.2 Block Codes

To formalize the concept of code and the encoding and decoding processes, we consider a finite set of symbols $\mathbb{F}_q$, called alphabet, with q elements. The information to be processed and the codewords will be expressed with symbols from this alphabet. $\mathbb{F}_q$ has the structure of a (finite) field (in particular the size of the alphabet q is the power of a prime).

Definition 1.1. A *linear block code* of length n and dimension k is a k-dimensional subspace $\mathcal{C}$ of $\mathbb{F}_q^n$.

The length of the code fixes the length of the data streams sent through the channel, and the dimension measures the amount of information, without redundancy, that each of these streams has.

Encoding is described by means of an encoding map, an injective linear map

$$g : \mathbb{F}_q^k \longrightarrow \mathbb{F}_q^n$$

with image space $\mathcal{C}$. $\mathbb{F}_q^k$ is the set of information words, and each element of the code $\mathcal{C}$ is a codeword.

Definition 1.2. A *generator matrix* of the code is a matrix representation of the encoding map.

Remark 1.3. To represent a morphism by a matrix, a basis of the image is taken and its vectors placed as the columns of the matrix. However it is an usual criterion in coding theory to place the vectors that generate the image (the code) as rows of

the generator matrix. We will follow this criterion although is not rare to find also column-wise generator matrices.

Remark 1.4. Although the smaller family of nonlinear block codes, where the encoding map is nonlinear, provides some examples of good codes, we will focus on linear block codes, which are the most extensively studied ones. Linearity in all elements does not only help in the analysis of the code, but also provides easy implementations of the encoding and decoding algorithms.

The third key parameter of a block code is its minimum distance.

Definition 1.5. Let $x, y \in \mathbb{F}^n$. The *Hamming weight* of x is the number of nonzero components of x, $w(x) = \sharp\{i|x_i \neq 0\}$. The *Hamming distance* between x and y is the number of components in which x and y differ, $d(x,y) = \sharp\{i|x_i \neq y_i\}$. The *minimum distance* of a code $\mathcal{C}$ is the minimum Hamming distance between any two different codewords, $d(\mathcal{C}) = \min_{x,y\in\mathcal{C}}\{d(x,y)\}$.

Because of linearity $d(x,y) = w(x-y)$ and thus $d(\mathcal{C}) = \min_{x,y\in\mathcal{C}}\{w(x-y)\} = \min_{c\in\mathcal{C}}\{w(c)\}$. Endowed with the Hamming distance, $\mathbb{F}^n$ is a metric space.

A block code over $\mathbb{F}_q$ with length n, dimension k and minimum distance d is often referred to as a $[n,k,d]_q$-code.

The minimum distance quantifies the number of errors that the code can detect or correct. It is well-known that a code with minimum distance d can detect all errors of weight at most $d-1$ and correct all errors of weight at most $\lfloor \frac{d-1}{2} \rfloor$. Therefore the minimum distance of a code characterizes its error correcting capacity.

However, in contrast to other parameters of the code, the minimum distance of a code is in general very difficult to compute and one has to be often satisfied with bounds. Well known examples of such bounds are

$$\sum_{k=0}^{d-1} \binom{n}{k} (q-1)^k \geq q^{n-k} \qquad d \leq \left\lfloor \frac{nq^{k-1}(q-1)}{q^k-1} \right\rfloor \qquad \sum_{l=0}^{k-1} \left\lceil \frac{d}{q^l} \right\rceil \leq n$$

Gilbert-Varshamov bound — Plotkin bound — Griesmer bound

$$\sum_{k=0}^{\frac{d-1}{2}} \binom{n}{k} (q-1)^k \leq q^n \qquad d \leq n-k+1$$

Hamming or sphere packing bound — Singleton bound

The Singleton bound is particularly interesting and those codes that attain this bound form the class of codes known as *Maximum Distance Separable (MDS) codes*. MDS codes have many nice and useful properties:

- Any k positions form an information set (the corresponding columns of the generator matrix are linearly independent), i. e., if k components are fixed, the information vector related to a codeword has as entries the elements in these components.

- Any d positions support a codeword of minimum weight.
- The weight distribution of MDS codes (the number of codewords of each weight) is completely determined.

Let us consider the exact sequence

$$0 \longrightarrow \mathbb{F}_q^k \xrightarrow{g} \mathbb{F}_q^n \xrightarrow{h^T} \mathbb{F}_q^{n-k} \longrightarrow 0, \tag{1.1}$$

then, the code $\mathcal{C} = Img$ can be also characterized as $\mathcal{C} = Kerh^T$. If H^T is a matrix representing h^T (with the criterion in Remark 1.3), then $GH^T = 0$ and

$$\mathcal{C} = \{x \in \mathbb{F}_q^n | xH^T = 0\}.$$

H^T is called a *parity check matrix* or just *check matrix* of the code.

Any vector $v \in \mathbb{F}_q^n$ can be written as $v = c + e$, $c \in \mathcal{C}$, and

$$vH^T = (c + e)H^T = cH^T + eH^T = eH^T.$$

eH^T is called the *syndrome* of v. Every element of the coset $e + \mathcal{C}$ has the same syndrome. If $e \neq 0$, i. e. $v \notin \mathcal{C}$, the syndrome decoding method consists on finding the vector with minimum weight in the coset $e + \mathcal{C}$. In practice, unless the number of correctable patterns is low enough, this scheme cannot be used.

Let us now consider duals in the sequence (1.1),

$$0 \longrightarrow \mathbb{F}_q^{n-k} \xrightarrow{h} \mathbb{F}_q^n \xrightarrow{g^T} \mathbb{F}_q^k \longrightarrow 0, \tag{1.2}$$

h defines a code $\mathcal{C}'$, the generator matrix of which is precisely H. In particular, for every $c \in \mathcal{C}$, $c' \in \mathcal{C}'$, the scalar product $c \cdot c' = 0$.

Definition 1.6. The *dual code* of $\mathcal{C}$, denoted $\mathcal{C}^\perp$ is the vector subspace

$$\mathcal{C}^\perp = \{x \in \mathbb{F}_q^n | x \cdot c = 0 \text{ for all } c \in \mathcal{C}\}$$

The term dual makes sense, as $(\mathcal{C}^\perp)^\perp = \mathcal{C}$. On the other side, the check matrices of $\mathcal{C}^\perp$ are the transpose of the generator matrices of $\mathcal{C}$.

Although the mathematical definition of a code is rather simple, it is not so easy to construct good codes, i. e. those with minimum distance approaching or attaining the bounds mentioned before. On the other side, the design of decoding algorithms represents itself a whole subject of particular interest. These are two of the main problems in coding theory.

For a more detailed introduction and further description of block codes many references as for example the classical [MS77, McE77, vL82] are available.

1.3 Introduction to Convolutional Coding

Convolutional codes were considered for the first time by Elias in [Eli55]. After some development, the theory was formalized first by Forney [For70] and then by Piret [Pir88] and McEliece [McE98].

For a better understanding of the underlying idea we will view the step from block codes to convolutional codes as presented in [McE98, Ros01]. Let us consider a whole transmission process, where a set of information words $\{u_0, u_1, \ldots, u_t\}$ are encoded to the codewords $\{c_0, c_1, \ldots, c_t\}$ which are transmitted sequentially. We can then describe the whole process with only one expression,

$$\begin{aligned} \mathbb{F}_q^k[z] &\longrightarrow \mathbb{F}_q^n[z] \\ \sum_i u_i z^i &\longmapsto \sum_i u_i z^i G = \sum_i c_i z^i \end{aligned}$$

where both the information and the encoded symbols are written as the vector polynomials $u(z)$, $c(z)$. Their terms of degree i are the vectors at time step i.

Elias' idea was to replace the scalar matrix G with a polynomial matrix $G(z)$, leading to an injective module homomorphism

$$\begin{aligned} \mathbb{F}_q^k[z] &\longrightarrow \mathbb{F}_q^n[z] \\ u(z) &\longmapsto u(z)G(z) = c(z) \end{aligned} \quad . \tag{1.3}$$

In $c(z)$, thought as a vector polynomial, the vector coefficient c_i is now dependant not only on u_i but also on $u_{i-1}, u_{i-2}, \ldots$ This is the main difference between convolutional and block codes.

With this in mind we can give a formal definition of convolutional codes.

Definition 1.7. A *convolutional code* of length n and dimension k is a k-rank submodule of $\mathbb{F}_q^n[z]$. The matrix $G(z)$ is a *generator matrix* of the code and represents the encoding map (1.3), the image of which is the code.

As explained in [For70], the term "convolutional" is used because the output sequences can be regarded as the convolution of the input sequences with the sequences in the encoder.

Remark 1.8. In the literature several definitions of convolutional codes can be found. It's not a matter of meaning but of the setting. Sometimes, $\mathbb{Z}$ instead $\mathbb{Z}_+$ is used as time axis, and therefore $\mathbb{F}[z, z^{-1}]$, the ring of Laurent polynomials, instead of $\mathbb{F}[z]$, is the ring considered. In a more general case, the generator matrix $G(z)$ can have not just polynomial but rational components, leading to different definitions. A usual one develops the theory in an analogous way to the block codes case considering the field of rational functions on z, $\mathbb{F}_q(z)$, as the ground field. Then, a convolutional code is defined as a k-dimensional subspace of $\mathbb{F}_q^n(z)$. Another approach is to consider infinite transmission sequences, in which case the ground field is $\mathbb{F}_q((z))$

and a code is defined as a subspace of $\mathbb{F}_q^n((z))$ with a basis of vectors in $\mathbb{F}_q(z)$. Both definitions are shown to be equivalent in [Ros01].

However we think that the most insightful approach is that of codes defined as submodules over the ring $\mathbb{F}[z]$ since it describes finite processes starting in time 0, which is usually the application context of convolutional codes. Further, this definition has the advantage of finiteness, what makes its algebraic study simpler. This presents though the problem that essentially identical codes may correspond to different submodules, all of which are however included in the same maximal one. Only the maximal submodules, corresponding to the so called *observable codes* [RSY96], have desirable properties and are in general studied. However, it is always possible to obtain the observable code containing a non-observable one. For this reason we prefer to consider all of them as the same code (although given by different submodules), and we contemplate the following definition.

Definition 1.9. A *convolutional code* of length n and dimension k generated by a $k \times n$ polynomial matrix $G(z)$ is the maximal, with respect to the inclusion, k-rank submodule contained in the $\mathbb{F}(z)$-subspace generated by the rows of $G(z)$.

As well as in the block codes case, every convolutional code has different generator matrices. Two matrices will be called *equivalent* if they generate the same code. Equivalence can be characterized in the following way, which is also valid for the rational setting.

Lemma 1.10 ([Ros01])**.** *Two $k \times n$ generator matrices $G(z)$, $G'(z)$ are equivalent if and only if there is a $k \times k$ invertible rational matrix $R(z)$ such that $G'(z) = R(z)G(z)$.*

As a result, even when the setting is that over the field of rational functions, it is always possible to work with polynomial generator matrices.

Theorem 1.11 ([For70] Theorem 3)**.** *Every generator matrix with entries in $\mathbb{F}_q^n(z)$ has an equivalent polynomial matrix.*

If codes are defined as submodules, it is possible to restrict the notion of equivalence to the matrices that generate the maximal submodule associated with the code. To this respect, there is an habitual characterization.

Lemma 1.12 ([Ros01])**.** *Two $k \times n$ polynomial matrices $G(z)$, $G'(z)$ are equivalent if and only if there is a $k \times k$ unimodular polynomial matrix $U(z)$ such that $G'(z) = U(z)G(z)$.*

Apart from the length and dimension of a convolutional code, the third analogous parameter to those of block codes is the free distance. For that we define, also in an analogous way, the Hamming weight of a convolutional codeword.

Definition 1.13. The *Hamming weight* of a component of a convolutional codeword is the number of its nonzero coefficients, $w(p(z)) = w(\sum p_i z^i) = \sharp\{i | p_i \neq 0\}$. The *Hamming weight* of a convolutional codeword $c \in \mathcal{C}$ is the sum of the weights

of its components, $w(c) = \sum w(c_i)$. Given two convolutional codewords $c, c' \in \mathcal{C}$ the *Hamming distance* between c and c' is the Hamming weight of their difference, $d(c, c') = w(c - c')$. The *free distance* of a convolutional code is the minimum Hamming distance between any two different codewords, or equivalently, the minimum Hamming weight of any codeword, $d_{free}(\mathcal{C}) = \min_{c,c' \in \mathcal{C}} \{d(c, c')\} = \min_{c \in \mathcal{C}} \{w(c)\}$.

According to these definitions, in the polynomial setting, every codeword has a finite weight. In the rational setting the concept of a codeword with finite weight needs to be included.

In convolutional coding, some additional parameters referring to the distance between words characterize the code.

Definition 1.14. The *j-th column distance* of the code $\mathcal{C}$ is the minimum weight of the codewords truncated at degree j,

$$d_j^c = min\{\sum_{i=0}^{j} wt(c_i) \,|\, c(z) = \sum_{i=0} c_i z^i \in \mathcal{C}, c_0 \neq 0\}.$$

Clearly, $d_0^c \leq d_1^c \leq d_2^c \leq \ldots \leq d_{free}$, and there exists an integer r such that $d_r^c = d_{r+j}^c = d_{free}$ for all $j \geq 0$ [JZ99]. r is exactly the minimum degree of any codeword with minimum weight.

Proposition 1.15 ([HRS05])**.** *For every $j \in \mathbb{N}_0$, we have $d_j^c \leq (n-k)(j+1)+1$.*

Definition 1.16. The *j-th row distance* d_j^r of the code $\mathcal{C}$ is the minimum weight of the codewords encoding a non-zero information word of degree j,

$$d_j^r = min\{wt(u(z)G(z)) \,|\, u(z) = \sum_{i=0}^{j} u_i z^i \in \mathbb{F}[z]^k, u(z) \neq 0\}.$$

Clearly, $d_0^r \geq d_1^r \geq d_2^r \geq \ldots \geq d_{free}$, and we have [JZ99]

$$d_0^c \leq d_1^c \leq \ldots \leq d_\infty^c = d_{free} = d_\infty^r \leq \ldots \leq d_1^r \leq d_0^r.$$

The calculus of the free distance is even harder than that of the minimum distance of block codes, and also bounds for it have to be considered. Several of these bounds will be presented later, after some further needed concepts are introduced.

Considering the analogous to our setting of the exact sequences (1.1, 1.2) we can define the check matrix and the dual code of a convolutional code.

Definition 1.17. A $n-k \times n$ polynomial matrix H is a *check matrix* of the code $\mathcal{C}$ if $cH^T = 0 \Leftrightarrow c \in \mathcal{C}$.

Definition 1.18. The *dual code* of a convolutional code $\mathcal{C}$ is the module

$$\mathcal{C}^\perp = \{x \in \mathbb{F}_q^n[z] | x \cdot c = 0 \text{ for all } c \in \mathcal{C}\}.$$

As for block codes, a check matrix of a convolutional code generates its dual code.

Once the analogous concepts to those in block coding have been presented, we proceed with those which arise specifically in convolutional coding.

Definition 1.19. The *i-th row degree* of a polynomial matrix is the maximum degree of the polynomials in the i-th row.

In block coding all equivalent generator matrices are suitable for our purposes. However, in convolutional coding, there are some preferred over the others, and even some "catastrophic" ones. We consider now the following definitions of "desirable" generator matrices following [McE98].

Definition 1.20. A *basic generator matrix* of a code is a matrix in which the highest degree of its $k \times k$ minors is minimum among all its equivalent matrices. A *reduced generator matrix* of a code (termed *minimal basic encoder* in [Pir88, JZ99]) is a row reduced matrix. A matrix which is both basic and reduced is called a *canonical generator matrix*.

Every code has at least one basic encoder, and several algorithms to find one are known [Pir88, McE98]. There is a well known characterization of basic encoders.

Theorem 1.21 ([McE98], Theorem A.1)**.** *A generator matrix $G(z)$ of a code C is basic if and only if any of these equivalent conditions is fulfilled:*

1. *The invariant factors of $G(z)$ are all 1.*
2. *The gcd of the $k \times k$ minors of $G(z)$ is 1.*
3. *For any $\alpha \in \overline{\mathbb{F}_q}$, the algebraic closure of $\mathbb{F}_q$, $G(\alpha)$ has maximum rank.*
4. *$G(z)$ has a right polynomial inverse $H(z)$, i. e., $G(z)H(z) = Id_k$.*
5. *$G(z)$ maps only polynomials into polynomials, i. e., a finite support output implies a finite support input.*
6. *We can add a submatrix $L(z)$ to complete $G(z)$ to a $n \times n$ unimodular matrix $\binom{G(z)}{L(z)}$.*

Remark 1.22. Condition 5 has a special meaning. It could be the case, if the transmission error has weight bigger than the correction capacity of the code, that the decoded codeword $v' \in \mathcal{C}$ is not the same one as the sent one, $v \in \mathcal{C}$. This is known as a decoding error. This error, the difference between both codewords $f = v - v'$, is also (because of linearity) a codeword, and in particular has finite support. If $G(z)$ allows to encode an infinite support input into a finite support output, then a polynomial codeword can be decoded into an infinite length input. In this case, a decoding error could result in an infinite error in the information word decoded, which is a catastrophic behavior that should be avoided. Those generator matrices which encode infinite inputs into finite outputs are called *catastrophic generator matrices*. Condition 5 means that a basic encoder is not catastrophic, and in particular, for every code there is at least one non-catastrophic encoder.

Reduced matrices are also characterized by two interesting and useful properties.

Theorem 1.23 ([McE98], Theorem A.2)**.** *A generator matrix $G(z)$ of a code C is reduced if and only if any of these equivalent conditions is fulfilled:*

1. *The matrix $\bar{G}$ with entries the coefficients of the highest degree terms on each row of G has full row-rank k.*
2. *The "predictable degree property": For any k-dimensional polynomial vector $u(z) = (u_1(z), \ldots, u_k(z)$*

$$deg(u(z)G(z)) = \max_{1 \leq i \leq k} (degu_i(z) + degg_i(z))$$

being $g_i(z)$ the i-th row of G.

The following concepts, although important, appear in the literature with different terms which we collect here.

Definition 1.24. The set of row degrees of a canonical encoder are called the *constraint lengths* [For70] or more usually the *Forney indices* [McE98] of the code and they are, up to ordering, invariants of the code [For70] . Their sum, denoted by δ, is called the *overall constraint length* [For70], the *degree* [McE98], or the *complexity* [Pir88] of the code. The highest Forney index is called the *memory* of the code.

The Forney indices of a convolutional code can be interpreted in terms of algebraic geometry (Grothendieck indices of the quotient sheaf associated with the code as a submodule) and linear systems theory (Kronecker indices of the transfer function of the code as a linear system).

The complexity of the code appears in the systems literature as the McMillan degree of the corresponding linear system. Roughly speaking, it measures the influence of past inputs in the present output of the encoder. In fact, convolutional codes of complexity zero are precisely block codes. In a similar way, the memory accounts for the number of past inputs on which each encoding step depends.

Remark 1.25. Another invariant of the code, up to ordering, is the set of $k \times k$ minors of any basic generator matrix of the code (they don't change under product by an unimodular matrix).

Although the sets of Forney indices of a code $\{e_1, \ldots, e_k\}$ and those of its dual $\{f_1, \ldots, f_{n-k}\}$ are in general different, their overall constraint lengths are equal.

We can now give some bounds on the free distance of a convolutional code,

$$d_{free} \leq S(n, k, \delta) = (n - k)\left(\left\lfloor \frac{\delta}{k} \right\rfloor + 1\right) + \delta + 1$$

generalized Singleton bound [RS99]

$$d_{free} \leq \min_{i \in \hat{\mathbb{N}}} \left\lfloor \frac{n(m+i)q^{k(m+i)-\delta-1}(q-1)}{q^{k(m+i)-\delta}-1} \right\rfloor$$

Heller bound

$$d_{free} \leq max\{d' \in \{1, \ldots, S(n,k,\delta)\} / \sum_{l=0}^{k(m+i)-\delta-1} \left\lceil \frac{d'}{q^l} \right\rceil \leq n(m+i) \forall i \in \hat{\mathbb{N}}\}$$

Griesmer bound

where

$$\hat{\mathbb{N}} = \begin{cases} \mathbb{N} \equiv \{1, 2, \ldots\} & \text{if } km = \delta \\ \mathbb{N}_0 \equiv \{0, 1, 2, \ldots\} & \text{if } km > \delta \end{cases}.$$

Analogously to the block case, those codes that attain the generalized Singleton bound are known as *MDS convolutional codes*. It has been proven in [RS99] that the Forney indices of MDS convolutional codes can only take the values $\lfloor \delta/k \rfloor$ and $\lceil \delta/k \rceil$.

Heller and Griesmer bounds were developed in [GLS03] by applying Plotkin and Griesmer bounds on the minimum distance to certain block codes that appear as subsets of the convolutional code.

A particularly interesting approach to convolutional codes comes from the fact that by definition they are equivalent to linear dynamical systems over a finite field. As a result, not only the algebraic techniques used to study block codes, but also tools from systems theory, have been successfully used to construct families of convolutional codes, to better understand their properties and to develop decoding algorithms. This relationship will be detailed in Chapter 4.

For a deeper and further insight on convolutional codes, classical references are [Pir88, McE98, JZ99].

1.4 Applications of Coding Theory

From its origin, coding theory has had mainly a practical dedicated purpose, as a fundamental part of information theory.

However, very soon after some development of the theory and the first good families of codes being known, it was clear that the relationship between codes and other branches of mathematics, that had provided some of the code constructions, was useful not necessarily only in one direction. Applications from the coding knowledge were developed to solve problems from other areas as different as algebraic geometry, complexity theory or computer science.

An early natural application of codes was in the setting of fault-tolerant computation. The problem arises when in a natural model of fault-tolerant computation a

boolean function has to be computed by a circuit, the gates of which have nonzero probability of error (giving a wrong output). To solve it in order to construct a circuit that even in the presence of errors has a high enough probability of correct computation, error correcting codes were suggested, and in particular LDPC codes were applied.

In a similar way, codes can be used to design fuzzy extractors of cryptographic keys. Reliable keys are expected to be unique, non-trivially long, and easy to remember, all of which are usually contradictory requests. To overcome this problem, biometric properties (retina-scan, fingerprint, ...) may be used to define a private key fulfilling these conditions. However, a cryptographic key must be also accurately reproduced, while measure of biometric properties is often subject to reading errors. Error correction codes may be used to solve problems derived from the extraction of cryptographic keys from any keying material that, unlike classical keys, is neither precisely reproducible nor uniformly distributed.

The appearance of Goppa codes opened a way to apply coding theory in algebraic geometry. In general, for the construction of algebraic geometric codes is important to count on algebraic curves with as many rational points as possible. The interest arisen around these codes stimulated also research on the number of rational points of algebraic curves with a fixed genus as well as on the asymptotic values of the ratio of the number of rational points to the genus. coding theory methods were successfully applied, giving better results for this ratio over small fields than the Hasse-Weil theorem.

Many of the applications of coding theory to cryptography and complexity theory are based in two particular classes of codes. The first one is that of *locally testable codes*. These are families of codes that are highly robust against errors and that are provided with sublinear time probabilistic algorithms for error detection. The error detection algorithm probes the received vector only at a small number of components. This property is essential for their use in probabilistically checkable proofs (PCP) or in software verification. Local testability allows approximate tests of large objects by considering only a very small number of probes. This yields much faster algorithms for approximate-testing of the corresponding property (i.e., to be a codeword, a valid proof or a correct software).

The second kind of codes that we refer to are *locally decodable codes*. These codes are provided with sublinear time probabilistic error correction algorithms, that with an input of a small number of components from the received vector give as output a particular component of the corresponding information word. Formally, a $[n, k]$ linear code $\mathcal{C}$ is (q, δ, p)-locally decodable if there is a decoder D for $\mathcal{C}$ such that

- given a vector $v \in \mathbb{F}^n$ and $i \in \{1, 2, \ldots, k\}$, the decoder reads q components of v uniformly at random and outputs a single component $D(v, i)$.
- give any message $u \in \mathbb{F}^k$ and its encoding c, for all vectors v that agree with c on at least δn components, then $Pr[D(y, i) = u_i] \geq p$ for all $i \in \{1, 2, \ldots, k\}$.

There is a wide range of particular applications of codes in the mentioned areas. We will enumerate some of them.

In cryptography error correcting codes are used in the cryptanalysis of certain block ciphers, in the search of smooth integers, for efficient traitor tracing, for private information retrieval and for generation of pseudorandom bits. These are in turn applied in the construction of cryptographic primitives, which are needed for different algorithms such as public-key cryptosystems, pseudorandom function generators, pseudorandom permutation generators, digital signature schemes, bit commitment protocols and zero-knowledge interactive proof systems.

The main applications to complexity theory come from the connections between error correcting codes and key combinatorial objects in complexity theory: hash functions, randomness extractors, pseudorandom generators and expander graphs. Some of these applications are average-case complexity, program testing, probabilistically checkable proofs (PCP), hardness amplification of boolean functions and hardcore predicates.

In Computer Science codes have been a valuable tool to solve problems like giving lower bounds on the complexity of algorithms, in particular algorithms for the matrix product, the generation random numbers and the verification of software.

There are also algorithmic application related to guessing secrets and a number of them for communication complexity (how many messages should two or more parts exchange for a particular aim).

For more on applications of coding theory interesting references are [Sud00, Dou03, Tre04].

We present now in more detail some examples of applications of coding theory in order to show different uses of codes.

1.4.1 McEliece Public Key Cryptosystem

Recall that a public key cryptosystem is a system which provides encryption for everyone, due to the use of a public encryption key, while only the allowed receiver can decrypt the message, by using a private decryption key. Both public and private keys are related, but is not possible in practice to obtain the private key just knowing the public one.

As explained before, syndrome decoding of linear block codes is not a practical decoding scheme. However, there are families of codes with very fast decoding algorithms based on their construction. The idea of McEliece [McE78] was to take one of these codes and "disguise" it. Then, although everyone knows that encryption is done through a block code, only the allowed receiver knows which is the actual code and which fast decoding algorithm to use, while an enemy can only use for decryption a general decoding scheme, i. e. syndrome decoding, which in practice is not possible.

The private key consists of the matrices (S, G, P), where S is a random, invertible $k \times k$ matrix, P is a random $n \times n$ permutation matrix, and G is the $k \times n$ generator matrix of a code that corrects up to t errors.

The public key is the pair $(\tilde{G}, t')$, where $\tilde{G}$ is the $k \times n$ matrix product of the three private matrices, $\tilde{G} = SGP$, and $t' \leq t$ is the number of errors that a sender

of a message is allowed to add to his message.

For the encryption the plain text is split up into blocks of k bits. Each block is encoded with $\tilde{G}$ and a random error vector of size n with at most t' entries is added to the codeword, resulting in the cipher block

$$c = m\tilde{G} + e$$

which will be sent.

For the decryption, the receiver multiplies the cipher text with the inverse of the permutation matrix,

$$c' = cP^{-1} = m\tilde{G}P^{-1} + eP^{-1} = mSG + eP^{-1}.$$

c' has to be decoded respect to G. This is possible as the error contained in c', eP^{-1}, has at most the $t' \leq t$ intentional errors. By means of the fast decoding algorithm, c' can be quickly decoded into mS. To get the plain text message the receiver will then multiply it with the inverse of S

$$m = mSS^{-1}.$$

In order to use this scheme to decipher a message, the inverse matrices of P and S have to be known. An unauthorized third part which does not have this information will face the problem to decode with respect to a general linear code. With an average choice of $t \geq 50$ and $n \geq 2^{10}$, this is a very difficult problem.

McEliece suggested using Goppa Codes, but any linear code with a fast decoding algorithm can be used.

1.4.2 Hardcore Predicates for One-Way Permutations

A basic tool in classical and modern cryptography are hardcore predicates of given one-way functions.

A *one-way function* $f : \{0,1\}^k \to \{0,1\}^k$ is a function easy to compute but hard on average to invert, i. e., computationally it cannot be inverted in polynomial time with correctness probability $Pr(correct) >> 0$. A function $P : \{0,1\}^k \to \{0,1\}^m$ is a *hard predicate* for f if it is hard on average to compute $P(x)$ given $f(x)$.

Hard predicates can be extracted from the Discrete Log function, however, for a general one-way function it doesn't seem possible. For this reason, a slight modification was introduced allowing P to depend also on an auxiliary random string r. In this context, P is a *hardcore predicate* for f if it is hard on average to compute $P(x, r)$ given $f(x)$ and r.

With the help of error-correcting codes, a hardcore predicate can be obtained for any one-way function.

Let $\mathcal{C} : \{0,1\}^k \to \{0,1\}^n$ be an efficiently list-decodable binary code, and $f : \{0,1\}^k \to \{0,1\}^k$ a one-way permutation. Consider $P : \{0,1\}^k \times [n] \to \{0,1\}$ a predicate with $P(x, j) = \mathcal{C}(x)_j$ and the function $f_0 : \{0,1\}^k \times [n] \to \{0,1\}^k \times [n]$ given by $f_0(x, j) = (f(x), j)$. P is a hardcore predicate for f_0: if given $f(x)$ some

algorithm could recover correctly in polynomial time a $\frac{1}{2} + \epsilon$ fraction (i. e., better than just dropping a coin) of the values of $P(x, i)$, $1 \leq i \leq n$, by list-decodability of $\mathcal{C}$ one could get a small list of vectors including x, and by checking one by one $f(x)$ the actual x could be found. However, this is contradictory with the hardness of inverting f.

We briefly present now two scenarios where hardcore predicates are used.

Pseudorandom generators

It is often hard to find big enough sequences of random numbers to be used in practice. However, for technical purposes, it is enough to have sequences of pseudorandom numbers computationally indistinguishable from random, i. e., that no algorithm can distinguish from random in polynomial time. For that purpose methods that stretch small sequences of truly random numbers into larger sequences of pseudorandom numbers have been created.

Consider $x, r \in \{0,1\}^k$, a one-way function f and a hardcore predicate for it P. As $P(x, r)$ is hard to compute given $f(x), r$, then $(f(x), r, P(x, r))$ is computationally indistinguishable from a random $(2k+1)$-bit string. Then

$$\begin{aligned} G: \quad \{0,1\}^{2k} &\longrightarrow \{0,1\}^{2k+1} \\ (x, r) &\mapsto (f(x), r, P(x, r)) \end{aligned}$$

is a pseudorandom generator. It is known that given a pseudorandom generator which stretches a sequence in one bit, then one can give a pseudorandom generator which stretches a sequence in any polynomial expansion factor.

The usual hardcore predicate for this purpose is $P(x, r) = x \cdot r$, which as a hardcore predicate defined by a code corresponds to the Hadamard code.

Probabilistic Public-Key Encryption

There is a particular class of one-way functions that has a prominent role in cryptography. A *one-way trapdoor function* is a one-way function which can be correctly inverted in polynomial time given some extra trapdoor information.

One-way trapdoor functions are in the basis of public-key cryptosystems, where the valuation of the function on a particular value has the role of the public key and the trapdoor information that allows the inversion has the role of the private key.

The use of hardcore predicates for encryption allows to add security to the system by adding a random factor before encrypting with a deterministic algorithm, and getting rid of it by decryption.

The scheme for the cryptosystem would be the following: Let m be a message to be encrypted, r a random string, f a one-way trapdoor function and P a hardcore predicate for f. Then the cipher of m is obtained by

$$c = (f(r), m + P(r)).$$

Decryption of a cipher (e, x) is done in the following way

$$m = x - P(f^{-1}(e)).$$

1.4.3 Secret Sharing

The problem of secret sharing consists of dividing a secret to distribute it to n parts so that no small number of them can recover the secret, but a big enough number of parts can combine their shares and get the secret.

More formally, a (l, m)-secret sharing scheme allows to divide a secret s into n shares $s_1, \ldots, s_n$, which are distributed to n parts $P_1, \ldots, P_n$ so that l or less parts cannot recover the secret combining their shares and m or more parts can always recover the secret.

A (l, m)-secret sharing scheme can be implemented using a code in the following way:

Let us represent the secret as $s \in \mathbb{F}_q$, and let $\mathcal{C}$ be a $[n+1, k, d]_q$-code, with $m \geq n - d + 2$ and $l \leq d^\perp - 2$, being $d^\perp$ the minimum distance of the dual code $\mathcal{C}^\perp$.

Choose any codeword $c = (c_0, c_1, \ldots, c_n)$ being $c_0 = s$. The shares to be given to the parts P_i will be $s_i = c_i$.

To recover the secret we need to have enough shares to recover the rest by decoding the "erasures" of the codeword. For that, the number of erasures (including the secret, c_0) cannot be bigger than the minimum distance, i. e., $n + 1 - m < d$, then it is enough $m \geq n - d + 2$.

To make sure that the secret cannot be recovered with l shares, recall that to recover an unknown component of the codeword from l known ones, it is enough to have a word in the dual code of weight $l+1$ with support in the l known components and in the one to be guessed, as the scalar product of both vectors (in $\mathcal{C}$ and in $\mathcal{C}^\perp$) is 0. Therefore, we need to impose that there is no codeword in $\mathcal{C}^\perp$ of weight $\leq l+1$, i. e. $d^\perp \geq l + 2$.

This construction allows an extension to the case where some parts are traitors and give an erroneous share. As long as we have enough shares and the traitors are not many, the codeword can be correctly decoded (and the traitors located).

1.5 Aims of This Work

This work is mainly devoted to the study of convolutional codes from different points of view. We address some of the main problems in coding theory: the construction and analysis of codes, in particular by means of a classification of convolutional codes as points of an algebraic variety, and the design of a decoding algorithm.

Both the algebraic and the systemstheoretic approaches to convolutional coding theory have proved to be useful. For this reason we consider the interplay of coding theory with algebraic geometry and systems theory and we use techniques from both areas to get our results.

For the classification of convolutional codes in Chapter 2 we will consider the structure of these codes as $K[z]$-modules to represent them as certain sheaves over the projective line. Then we consider the well-known representation of quotient sheaves as points of a grassmannian variety. However this representation does not

identify straightforwardly codes and geometric points. We will explain how to overcome this problem and we will determine the varieties representing convolutional codes with a certain set of parameters. This classification gives us a knowledge of convolutional codes that will be applied to derive bounds on the free distance and to construct optimal convolutional codes from well-known block codes.

Algebraic geometry will be used also to generalize the well-known family of Goppa codes to the convolutional setting in Chapter 3. In this work we study the construction and some properties of convolutional Goppa codes defined over elliptic curves. Many examples this class of convolutional Goppa codes have optimal free distance. In the same direction, we will also transfer to the context of convolutional codes another family of general algebraic geometric codes.

The relationship between linear systems and convolutional codes will be detailed in Chapter 4. In particular, this relationship will be used to state the problem of convolutional decoding in linear systems terms. This will result in a decoding algorithm for general convolutional codes.

Chapter 2

Classification of Convolutional Codes

2.1 Introduction

Convolutional codes have a structure of $\mathbb{F}[z]$-module. This makes its analysis much more complex than that of block codes. To study the properties of convolutional codes it is desirable to be able to classify them as points of some variety. We investigate here the relationship of convolutional codes with certain $\mathbb{F}$-vector subspaces, which will depend on some of their parameters. This will allow to identify each convolutional code as a point of a particular grassmannian variety. In addition, the subvariety of each grassmannian that is made up of the points representing convolutional codes will be determined. This classification of convolutional codes sheds light on their structure and turns out to be helpful to give bounds on their free distance and to define convolutional codes with good parameters.

2.2 Algebraic Geometric Preliminaries

The classification presented here is made in algebraic geometric terms. We now include some definitions, characterizations and well-known results that will be needed, with a particular focus on their implications in projective line, which is the scheme to be considered.

Let $(X, \mathcal{O}_X)$ be a scheme over a perfect field K (in particular finite fields, in which codes are usually defined, are perfect fields).

Definition 2.1. Given a $\mathcal{O}_X$-module M, the *localizations sheaf* of M, $\widetilde{M}$, is the sheaf associated with the presheaf such that for every open subset $U \subset X$, $\widetilde{M}(U) = M_U$ is the localization of M by the multiplicative system given by U.

Definition 2.2. A sheaf of $\mathcal{O}_X$-modules $\mathcal{F}$ on X is a *quasicoherent sheaf* if it is locally equal to the sheaf of localizations of a module, i. e., there is an open covering $\{U_i\}_i$ of X, and a family of $\mathcal{O}_X(U_i)$-modules $\{M_i\}_i$ such that $\mathcal{F}|_{U_i} \simeq \widetilde{M_i}$ (where $\widetilde{M_i}$

stands for the localizations sheaf of the module M_i). The sheaf is a *coherent sheaf* if in addition the modules M_i are finite generated.

The class of coherent sheaves is a more general class than those of locally free sheaves, invertible sheaves or sheaves of sections of vector bundles, but its share elements still some characteristic properties. In particular, the kernels and cokernels of morphisms of coherent sheaves are still coherent sheaves. In fact, this is the smallest class of sheaves that contains the structure sheaf and that for a given short exact sequence of sheaves, if two of them belong to the class, also the third one. For this reason it has an intrinsic interest to study the properties of coherent sheaves.

Remark 2.3. ([Har77, Ex II.5.9]) In particular, in $\mathbb{P}^1$, given the equivalence relationship between coherent sheaves $M \sim M' \Leftrightarrow \exists\, d \in \mathbb{N}$ such that $\underset{n\geq d}{\oplus} M_n \simeq \underset{n\geq d}{\oplus} M'_n$, there is an equivalence of categories

$$\begin{array}{ccc} \left\{ \begin{array}{l} \text{coherent sheaves} \\ \text{on } \mathbb{P}^1 \end{array} \right\} & \longleftrightarrow & \left\{ \begin{array}{l} \text{classes modulo } \sim \text{ of graduate} \\ K[x_0,x_1]\text{-modules finite generated} \end{array} \right\} \\ \mathcal{F} & \longmapsto & \underset{m}{\oplus} H^0(\mathbb{P}^1, \mathcal{F}(r)) \\ \widetilde{M} & \longleftarrow\!\shortmid & M \end{array} \tag{2.1}$$

where (for any scheme X) $\mathcal{F}(r) \equiv \mathcal{F} \otimes \mathcal{O}_X(r) \equiv \mathcal{F} \otimes \mathcal{O}_X(1)^{\otimes r}$, and $\mathcal{O}_X(1)$ is the invertible sheaf that maps every open subset to the same module as $\mathcal{O}_X$ but with the degree of every element increased by 1.

It is well-known that $H^0(\mathbb{P}^1, \mathcal{O}_{\mathbb{P}^1}(r))$ is the K-vector space of homogeneous polynomials of degree r in x_0, x_1 (the homogeneous coordinates of $\mathbb{P}^1$), $K[x_0, x_1][r]$. If we take the affine line $\mathbb{A}^1$, which is the open subset complementary to the point $P_\infty = (0;1)$, with affine coordinate $z = x_1/x_0$, then $H^0(\mathbb{A}^1, \mathcal{O}_{\mathbb{A}^1}(r))$ is the space of polynomials in z of degree $\leq r$. For a coherent sheaf $\mathcal{F}$, $H^0(\mathbb{A}^1, \mathcal{F}|_{\mathbb{A}^1}(r))$ is the module of the sections of $\mathcal{F}$ which are regular in $\mathbb{A}^1$ with a pole of order $\leq r$ at the infinite point P_∞. Unless otherwise stated, we will consider the standard ordered bases $\{x_0^r, x_0^{r-1}x_1, \ldots x_1^r\}$ for $H^0(\mathbb{P}^1, \mathcal{O}_{\mathbb{P}^1}(r))$ and $\{1, z, \ldots, z^r\}$ for $H^0(\mathbb{A}^1, \mathcal{O}_{\mathbb{A}^1}(r))$.

Definition 2.4. A sheaf $\mathcal{F}$ is *generated by its global sections* if there exists a family $\{s_i\}_{i\in I} \subset H^0(X, \mathcal{F})$ of global sections such that in every point $x \in X$ the images of all s_i in the fiber of $\mathcal{F}$, $\mathcal{F}_x$, generate $\mathcal{F}_x$ as a $\mathcal{O}_x$-module. As a consequence $\mathcal{F}$ is generated by its global sections if and only if $\mathcal{F}$ is a quotient sheaf of a free sheaf $\underset{i\in I}{\oplus} \mathcal{O}_X$. In particular the sheaves morphism

$$H^0(X, \mathcal{F}) \otimes \mathcal{O}_X \longrightarrow \mathcal{F}$$

is an epimorphism.

Theorem 2.5. (Serre, [Har77]) *Let X be a projective Noetherian scheme and $\mathcal{F}$ a coherent sheaf. Then there exists a $n_0 \in \mathbb{N}$ such that for every $n \geq n_0$, $\mathcal{F}(n)$ is generated by (a finite number of) its global sections.*

In particular, given an ample sheaf $\mathcal{L}$ for every $i > 0$ and for every $n > n_0$, $H^i(X, \mathcal{F} \otimes \mathcal{L}^n) = 0$.

Definition 2.6. Given the ample sheaf $\mathcal{O}_X(1)$ and a coherent sheaf $\mathcal{F}$, there is a map

$$\begin{aligned} P_{\mathcal{F}} : \mathbb{Z} &\longrightarrow \mathbb{Z} \\ r &\longmapsto \chi(\mathcal{F}(r)) = \sum_{i=0}^{dimX} (-1)^i dim H^i(X, \mathcal{F}(r)) \end{aligned}$$

being χ the Euler characteristic. $P_{\mathcal{F}}$ is a polynomial in r with rational coefficients [Har77, I Th 7.5, III Ex 5.2], called the *Hilbert polynomial* of the sheaf $\mathcal{F}$.

For the sake of clarity, we will also use the notation $P(\mathcal{F}, r)$.

Remark 2.7. In $\mathbb{P}^1$ it is known that for every coherent sheaf $\mathcal{F}$, $H^i(\mathbb{P}^1, \mathcal{F}(r)) = 0$ $\forall i > 1$. Then,

$$P_{\mathcal{F}}(r) = dim H^0(\mathbb{P}^1, \mathcal{F}(r)) - dim H^1(\mathbb{P}^1, \mathcal{F}(r))$$

and as we have seen there exists $n_1 \in \mathbb{N}$ such that $H^1(\mathbb{P}^1, \mathcal{F}(n)) = 0$ for every $n \geq n_1$.

For our classification not just coherent sheaves over a scheme X will be considered, but families of sheaves parameterized by a scheme S. We will take a base change in order to shift from working with a coherent sheaf $\mathcal{F}$ over the scheme

$$X \longrightarrow Spec\ k$$

to consider a family of sheaves $\mathcal{F}_S = \{\mathcal{F}_s\}_S$ over the S-scheme

$$X \times S \xrightarrow{f} S.$$

This family will be coherent and flat over S.

First the property of being flat will be characterized, to show that as a consequence some properties are locally stable on the fibers.

Definition 2.8. Recall the definition of flat modules and sheaves [Har77]. Given a morphism of schemes $f : T \longrightarrow S$, a quasicoherent sheaf of $\mathcal{O}_T$-modules $\mathcal{F}$ is *flat over* S if for every $t \in T$ the fiber $\mathcal{F}_t$ is a flat $\mathcal{O}_{S,f(t)}$-module (via the pull-back of f).

Proposition 2.9. [Har77]

1. *The property of being flat is stable with respect to changes on the base scheme.*
2. *Let $f : T \longrightarrow S$ be a relatively projective morphism (which will be our case) and $\mathcal{F}$ a coherent sheaf on T, then the following are equivalent:*
   - $\mathcal{F}$ is flat over S.
   - $f_*\mathcal{F}(r)$ is locally free in Y for every big enough $r \in \mathbb{N}$.
   - $P_{\mathcal{F}_{f(\alpha)}}(r)$ is locally constant in $f(\alpha) \in S$, i. e., the Hilbert polynomial of a fiber is independent of the base point and in particular the dimensions of the fibers are locally constant.

We proceed now with a brief introduction to the notion of the *quotient scheme*, which will be the key tool to classify convolutional codes.

The quotient scheme parameterizes all the coherent sheaves with a fixed Hilbert polynomial that are a quotient of a free sheaf of rank n.

It is often the case, like in ours, that it is easier to define the functor of points of a scheme X, $Hom(\cdot, X)$, than the scheme itself. For this reason, the quotient functor will be defined first, and then it will be shown to be representable (i. e., it is naturally isomorphic to the functor of points of a scheme).

Definition 2.10. Given a coherent sheaf $\mathcal{F}$ over X and a scheme S, a *quotient sheaf* of $\mathcal{F}_S = \mathcal{F} \otimes \mathcal{O}_S$ is a coherent sheaf Q on $X \times S$ flat over S, with a surjective morphism of sheaves $q : \mathcal{F} \otimes \mathcal{O}_S \longrightarrow Q$. Two quotient sheaves Q, Q' are equivalent if there exists an isomorphism $f : Q \longrightarrow Q'$ such that $q' = f \circ q$.

Definition 2.11. Given a coherent sheaf $\mathcal{F}$ over X and a polynomial $P(z) \in \mathbb{Q}[z]$, the *quotient functor* $\mathcal{Q}uot^P_{\mathcal{F}}$ is a contravariant functor from the category of K-schemes to the category of sets mapping every K-scheme to the set of equivalence classes of quotient sheaves of $\mathcal{F}$ in the following way

$$\begin{aligned} \mathcal{C}_{K\text{-schemes}} &\rightsquigarrow \mathcal{C}_{\text{sets}} \\ S &\longmapsto \left\{ \begin{array}{l} \text{classes of } S\text{-flat coherent sheaves } Q, \text{ being a quotient of} \\ \mathcal{F}_S, \text{ with Hilbert polynomial } P_Q(z) = P(z) \end{array} \right\}. \end{aligned}$$

The image of a morphism of schemes $q : T \longrightarrow S$ is the pull-back of the morphism $id \times q : X \times T \longrightarrow X \times S$ which maps classes of S-quotients to classes of T-quotients.

It is known [AK80, Nit05] that given a coherent sheaf $\mathcal{F}$ and a polynomial $P(r)$ there exists $r_0 \in \mathbb{N}$ such that for every quotient sheaf $q : \mathcal{F}_S \longrightarrow Q$ with Hilbert polynomial $P(r)$, being $K_Q = Ker q$, the sheaves $K_Q(r), \mathcal{F}_S(r), Q(r)$ are generated by their global sections for every $r \geq r_0$.

Theorem 2.12. *$\mathcal{Q}uot^P_{\mathcal{F}}$ is a functor represented by a subscheme of the grassmannian $Grass(H^0(\mathcal{F}(r)), P(r))$ for every $r \geq r_0$.*

Proof. The proof can be found in [Gro60, AK80, Nit05]. □

The *quotient scheme*, which we will denote both as $Quot^P_{\mathcal{F}}$ and as $Quot(\mathcal{F}, P)$, represents the quotient functor $\mathcal{Q}uot^P_{\mathcal{F}}$, so that given a scheme S, $\mathcal{Q}uot^P_{\mathcal{F}}(S) = Hom(S, Quot^P_{\mathcal{F}}) = \{\text{flat sheaves quotients of } \mathcal{F} \text{ parameterized by } S\}$. In particular, if the base field K is considered, any sheaf over $Spec(K)$ is flat, and $\mathcal{Q}uot^P_{\mathcal{F}}(K) = Hom(Spec(K), Quot^P_{\mathcal{F}}) = \{\text{quotient sheaves } \mathcal{F} \text{ with Hilbert polynomial } P\}$. Hence, every rational point from the scheme $Quot^P_{\mathcal{F}}$ represents a quotient sheaf of $\mathcal{F}$ with Hilbert polynomial P.

2.3 Kronecker-Hermite Canonical Form

The description of convolutional codes it is done according to a number of parameters such as its length, dimension and free distance. In addition, also the Forney indices, the memory and the degree of the code are considered. These, although obtained from a generator matrix of the code, are known to be invariants. For instance, the Forney indices are obtained from a canonical generator matrix (Definition 1.20) and they are the smallest row degrees of any generator matrix of the code. However, from a reduced matrix, still some elementary row operations can be made in order to reduce the degrees of their elements. In [FH01] the so-called Kronecker-Hermite canonical form and the modified Kronecker-Hermite canonical form of a polynomial matrix are presented, as a tool in the parametrization of conditioned invariant subspaces.

In our classification canonical matrices play an important role, although not all the matrices that describe a convolutional code have this property. For this reason, it is essential to count on a method to obtain a canonical matrix from any generator matrix of a code.

Before proceeding with our classification, we present the Kronecker-Hermite and the modified Kronecker-Hermite canonical forms of a polynomial matrix. In particular, a polynomial matrix in each of these forms is canonical, and every convolutional code has exactly one generator matrix in each of these forms. We reproduce a result from [FH01] which shows how to obtain the modified Kronecker-Hermite canonical matrix of a code from a reduced generator matrix.

Definition 2.13. Let $G(z) = (g_{ij})$ be a reduced polynomial matrix of maximum rank with Forney indices $\nu_1, \ldots, \nu_k$. Let us denote the rows of G by $g_1, \ldots, g_k$. $G(z)$ is in *Kronecker-Hermite canonical form* if there exists a uniquely determined set of pivot indices $1 \leq j_1 < \ldots < j_k \leq n$ such that

1.- g_{ij_i} is a monic polynomial with $deg(g_{ij_i}) = deg(g_i) = \nu_i$.

2.- $deg(g_{lj_i}) < \nu_i \ \forall \ 1 \leq l \leq k, l \neq i$.

3.- $deg(g_{il}) < \nu_i \ \forall \ l > j_i$ (and $deg(g_{il}) \leq \nu_i \ \forall \ l < j_i$).

$G(z)$ is in *modified Kronecker-Hermite canonical form* if it satisfies the conditions to be in Kronecker-Hermite canonical form with the exception that the Forney indices are in increasing order, $\nu_1 \leq \ldots \leq \nu_k$, and for the pivot indices the order $j_r < j_s$ is only required if $\nu_r = \nu_s$ ($deg(g_r) = deg(g_s)$).

Remarks 2.14.

1. The modified Kronecker-Hermite canonical form is obtained from the Kronecker-Hermite canonical form by a permutation of rows. In both cases each pivot entry, g_{ij_i} is a monic polynomial of degree ν_i while the rest of the polynomials in the same column have a lower degree, and those in the same row have lower or equal degree which is strictly lower if they are on the right of g_{i,j_i}.

2. The highest coefficient row matrix of G is in reverse row reduced echelon form.

3. In particular, every matrix in Kronecker-Hermite canonical form or in modified Kronecker-Hermite canonical form is basic and hence canonical in the sense of Definition 1.20.

Example 2.15. [FH01, Example 6.1] Two polynomial matrices with entries of degree

$$\begin{pmatrix} 2 & \boxed{2} & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 5 & 1 & 5 & \boxed{5} & 4 & 4 & 4 & 1 & 4 & 4 & 4 \\ 5 & 1 & 5 & 4 & 5 & 5 & \boxed{5} & 1 & 4 & 4 & 4 \\ 2 & 1 & 2 & 2 & 2 & 2 & 2 & \boxed{2} & 1 & 1 & 1 \\ 5 & 1 & 5 & 4 & 5 & 5 & 4 & 1 & 5 & \boxed{5} & 4 \end{pmatrix} \quad \begin{pmatrix} 2 & \boxed{2} & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 1 & 2 & 2 & 2 & 2 & 2 & \boxed{2} & 1 & 1 & 1 \\ 5 & 1 & 5 & \boxed{5} & 4 & 4 & 4 & 1 & 4 & 4 & 4 \\ 5 & 1 & 5 & 4 & 5 & 5 & \boxed{5} & 1 & 4 & 4 & 4 \\ 5 & 1 & 5 & 4 & 5 & 5 & 4 & 1 & 5 & \boxed{5} & 4 \end{pmatrix}$$

where the boxed elements correspond to the pivot entries g_{ij_i}, are in Kronecker-Hermite canonical form and modified Kronecker-Hermite canonical form respectively.

The following theorem, corresponding to a part of [FH01, Theorem 6.1], gives the method to obtain the modified Kronecker-Hermite canonical matrix of a convolutional code.

Theorem 2.16. *Every full row rank polynomial matrix G can be reduced to unique a modified Kronecker-Hermite canonical form by the product with a unimodular matrix (elemental row operations).*

Proof. Let us assume without loss of generality that G is in row proper form with s different Forney indices $\nu_1 < \ldots < \nu_s$, and k_i rows with degree ν_i. Let us call $G_i(z)$ the $k_i \times n$ submatrix of G with all its rows of degree ν_i. The proof goes by induction on the number s of different Forney indices.

If $s = 1$, all the Forney indices are equal. The highest coefficient row matrix can be reduced to reverse row reduced echelon form by left multiplication by constant elementary matrices. The resulting matrix is in modified Kronecker-Hermite canonical form (also in Kronecker-Hermite canonical form) and this form is uniquely determined.

Let us assume now as induction hypothesis that the matrix $G^{(i-1)} = (G_1^{\perp}, \ldots, G_{i-1}^{\perp})^{\perp}$ is in modified Kronecker-Hermite canonical form. In particular each G_r, $r < i$, is in Kronecker-Hermite canonical form with pivot indices $\{j_1^r, \ldots, j_{k_r}^r\} \subset \{1, \ldots, n\}$ and $deg(g_{mj_v^r}) < \nu_r \ \forall\, m \leq k_1 + \ldots + k_{i-1}$ $(m \neq k_1 + \ldots + k_{r-1} + v)$. The submatrix of $G^{(i-1)}$ made up with the columns $\bigcup_{m=1}^{i-1} \{j_1^m, \ldots, j_{k_m}^m\}$ is by the induction hypothesis nonsingular, column and row proper and its highest coefficient row matrix is a permutation matrix.

We can then reduce the submatrix of G_i formed with the columns $\bigcup_{m=1}^{i-1} \{j_1^m, \ldots, j_{k_m}^m\}$ with respect to the submatrix of $G^{(i-1)}$ corresponding to the same columns, and we note that this reduction doesn't increase the degrees of the rows of G_i nor affect the

rows of $G^{(i-1)}$. Thus, G_i is still row proper with all its row indices equal to ν_i, and its entries in the columns j_r^m, $m \leq i-1$, $r \leq k_m$, have degree $< \nu_m$. Therefore, the entries with degree ν_i cannot be in the columns $\bigcup_{m=1}^{i-1} \{j_1^m, \ldots, j_{k_m}^m\}$. As in the case for $s = 1$, the highest coefficient row matrix of G_i can be reduced by left multiplication by constant elementary matrices to reverse row reduced echelon form, with pivot indices $j_1^i, \ldots, j_{k_i}^i$. This multiplication keeps below ν_m the degrees of the entries in the columns $j_1^m, \ldots, j_{k_m}^m$, $m \leq i-1$, while the entries of $G^{(i-1)}$, and in particular those in the columns $j_1^i, \ldots, j_{k_i}^i$, have degree $< \nu_i$.

As a result, the polynomial matrix $(G_1^\perp, \ldots, G_{i-1}^\perp, G_i^\perp)^\perp$ has been reduced to modified Kronecker-Hermite canonical form.

From this construction we observe also that the submatrix of a matrix in modified Kronecker-Hermite canonical form given by the columns of the pivot indices is both column and row proper, its highest coefficient row matrix is a permutation matrix, and its determinant has degree δ. □

Corollary 2.17. *Every class of polynomial matrices modulo left multiplication by unimodular matrices has a unique representative in modified Kronecker-Hermite canonical form.*

Proof. Consider two different polynomial matrices M, M' from the same class modulo multiplication by unimodular matrices, i. e., $U^*M = M'$. By the theorem, each can be reduced to a unique modified Kronecker-Hermite canonical form via multiplication by an unimodular matrix, i. e., there exists exactly one unimodular matrix U and one unimodular matrix U' such that $UM = H$, $U'M' = H'$ with H and H' in modified Kronecker-Hermite canonical form.

Then, we would have that $H' = U'M' = U'U^*M$ and as the product of both unimodular matrices is also an unimodular matrix $\overline{U}$, we would have, $H' = \overline{U}M$. By the theorem, M can be reduced to a unique matrix in modified Kronecker-Hermite canonical form, thus it must be $H = H'$. □

The previous results show how to transform any generator matrix of a convolutional code to a matrix of a specific form, which among all matrices of this form is unique for each code.

2.4 Classification of Convolutional Codes

Our aim is to identify the algebraic structure of convolutional codes that share the same parameters by representing them as points of a variety. We will represent first generator matrices and then convolutional codes as quotients of certain sheaves. In our construction the key parameters will be the length, the dimension, the degree and the memory of the convolutional code.

2.4.1 Convolutional Codes as Quotient Sheaves

By analogy with Theorem 1.23 let us consider the following definition.

Definition 2.18. A polynomial matrix is *column reduced* if the constant matrix consisting on its column leading coefficients has maximum rank.

Matrices in modified Kronecker-Hermite canonical form are a particular case of column reduced matrices.

Lemma 2.19. *Every convolutional code can be associated with a point in a quotient scheme given by a class of sheaves without torsion in the affine line.*

Proof. Given a convolutional code $\mathcal{C}$ with memory m let us take a basic column reduced generator matrix $\overline{G}$ with polynomial entries of degree $\leq m$, which defines an injective $K[z]$-linear map

$$\overline{\phi_G} : K[z]^k \hookrightarrow K[z]^n. \tag{2.2}$$

As $\overline{G}$ is basic $Coker\overline{\phi_G}$ is a free module, or equivalently $\overline{\phi_G}$ has a retraction (there exists a polynomial right inverse matrix for $\overline{G}$ –Theorem 1.21). Hence, there is an exact sequence

$$0 \to K[z]^k \overset{\overline{\phi_G}}{\to} K[z]^n \to \overline{L} \to 0 \tag{2.3}$$

with $\overline{L}$ a free module with rank $n-k$. $\mathcal{C}$ must be associated with a sheaf that belongs to an exact sequence of sheaves such that when taking sections over $\mathbb{A}^1 = \mathbb{P}^1 - \{\infty\} = SpecK[z] \subset ProjK[x_0, x_1]$ the sequence (2.3) is obtained.

When homogenizing (2.3) and shifting the degrees so that the morphisms are of graduated modules, we get the exact sequence

$$0 \to K[x_0, x_1]^k \overset{\phi_G}{\to} \overset{n}{\underset{i=1}{\oplus}} K[x_0, x_1][m_i] \to Coker\phi_G \to 0 \tag{2.4}$$

being the matrix representation of ϕ_G

$$G = \begin{pmatrix} g_{11} & \cdots & g_{1n} \\ \vdots & & \vdots \\ g_{k1} & \cdots & g_{kn} \end{pmatrix}$$

where the elements g_{ij} are homogeneous polynomials on x_0, x_1 of degree m_j. If we take the corresponding polynomials in affine coordinates we get the matrix $\overline{G}$.

The sequence (2.4) is an exact sequence of graduated $K[x_0, x_1]$-modules that as shown in Remark 2.3 corresponds to a sequence of coherent sheaves on $\mathbb{P}^1$

$$0 \longrightarrow \mathcal{O}_{\mathbb{P}^1}^k \xrightarrow{\widetilde{\phi_G}} \overset{n}{\underset{i=1}{\oplus}} \mathcal{O}_{\mathbb{P}^1}(m_i) \longrightarrow Q \to 0 \tag{2.5}$$

and when taking sections on $\mathbb{A}^1$ it results the sequence (2.3). In this way, the sheaf Q corresponds to the convolutional code generated by $\overline{G}$.

Note that all column reduced generator matrices of a code with the same sequence of column degrees are in the same class modulo multiplication by elements of the general linear group Gl_K (the product of a column reduced generator matrix by a polynomial matrix would increase some column degrees). Then, all the quotient sheaves of $\overset{n}{\underset{i=1}{\oplus}} \mathcal{O}_{\mathbb{P}^1}(m_i)$ representing column reduced generator matrices of the same convolutional code are equivalent. Hence, the code is represented by a unique point in the corresponding quotient scheme.

In addition, as $\Gamma(\mathbb{A}^1, Q) = \overline{L}$ is a free module, its torsion $T(Q)$, can only be in the point at infinite, p_∞. The fact that then, all sheaves equivalent to Q have torsion only in the infinity means that for the corresponding polynomial matrices, the property of being basic is invariant under the action of Gl_K, as we already knew.

Let us now determine the quotient scheme in which the code is represented.

The Hilbert polynomial of Q is

$$\begin{aligned} P(Q,r) &= P(\overset{n}{\underset{i=1}{\oplus}} \mathcal{O}_{\mathbb{P}^1}(m_i), r) - P(\mathcal{O}^k_{\mathbb{P}^1}, r) = \\ &= \sum_{i=1}^{n}(m_i + 1 + r) - k(r+1) = \\ &= (n-k)(r+1) + \sum_{i=1}^{n} m_i = (n-k)(r+1) + \deg Q \end{aligned}$$

Recall that the degree of the sheaf Q is precisely

$$\deg Q = \deg det Q = \deg det(\overset{n}{\underset{i=1}{\oplus}} \mathcal{O}_{\mathbb{P}^1}(m_i)) - \deg det \mathcal{O}^k_{\mathbb{P}^1} = \sum_{i=1}^{n} m_i - 0 = \sum_{i=1}^{n} m_i \ .$$

Then, the convolutional code is represented by a sheaf Q, with $sup(T(Q)) \subset \{P_\infty\}$, which belongs to the quotient scheme defined by the sheaf $\overset{n}{\underset{i=1}{\oplus}} \mathcal{O}_{\mathbb{P}^1}(m_i)$ and the polynomial $P(r) = (n-k)(r+1) + grQ$,

$$Q \quad \in Quot^{P(r)}_{\oplus \mathcal{O}_{\mathbb{P}}(m_i)} \ .$$

□

Remark 2.20. The relationship between column reduced generator matrices of a code having the same column degrees and the relationship between equivalent quotient sheaves is the same, given by multiplication by elements of $Gl_K(k)$. However, it is possible in general to have generator matrices with different sequences of column degrees, which would result in quotients of different sheaves. Further, if we considered no column reduced generator matrices, it could be possible to have non-equivalent quotients of the same sheaf corresponding to generator matrices of the same code. Both cases don't occur when all the Forney indices are the same. In general, if we don't assume all the Forney indices equal, a slightly different correspondence between codes and sheaves has to be established, as explained later on in this chapter.

Theorem 2.21. *Every convolutional code of type $[n,k]$ that has a column reduced generator matrix with column degrees $\{m_1,\ldots,m_n\}$ is represented by a point in the grassmannian $Grass(H^0(\overset{n}{\underset{i=1}{\oplus}}\mathcal{O}_{\mathbb{P}^1}(m_i)), n+\theta-k)$ with $\theta=\sum_{i=1}^n m_i$. This point is a block code of type $[n+\theta,k,d_0^r]$.*

Proof. As seen before, every quotient scheme can be considered a subscheme of several grassmannians.

Let us check which one is the smallest grassmannian containing the quotient scheme $Quot^{P(r)}_{\oplus\mathcal{O}_{\mathbb{P}}(m_i)}$ and how is the point representing each code.

By Theorem 2.12 the smallest grassmannian will be given by the minimal r_0 such that for every quotient sheaf $\overset{n}{\underset{i=1}{\oplus}}\mathcal{O}_{\mathbb{P}^1}(m_i)\longrightarrow Q$ with Hilbert polynomial $P(r)$, $\mathcal{O}^k_{\mathbb{P}^1}(r_0)$, $\overset{n}{\underset{i=1}{\oplus}}\mathcal{O}_{\mathbb{P}^1}(m_i+r_0)$ and $Q(r_0)$ are generated by their global sections.

For that it will be enough to take $r_0=0$. Then we have

$$Quot(\overset{n}{\underset{i=1}{\oplus}}\mathcal{O}_{\mathbb{P}^1}(m_i),P(r))\xhookrightarrow{\phi_{\{m_i\}_1^n}} Grass(H^0(\overset{n}{\underset{i=1}{\oplus}}\mathcal{O}_{\mathbb{P}^1}(m_i)),P(0)) \tag{2.6}$$

which maps the quotient sheaf Q given by the sequence

$$0\longrightarrow\mathcal{O}^k_{\mathbb{P}^1}\xrightarrow{\widetilde{\phi_G}}\oplus\mathcal{O}_{\mathbb{P}^1}(m_i)\longrightarrow Q\longrightarrow 0$$

to the element of the grassmannian obtained by taking global sections,

$$0\longrightarrow H^0(\mathbb{P}^1,\mathcal{O}^k_{\mathbb{P}^1})\xrightarrow{\psi_G}H^0(\mathbb{P}^1,\oplus\mathcal{O}_{\mathbb{P}^1}(m_i))\longrightarrow H^0(\mathbb{P}^1,Q)\longrightarrow 0.$$

As seen before, $\widetilde{\phi_G}$ can be obtained from the morphism

$$\overline{\phi_G}:K[z]^k\longrightarrow K[z]^n,$$

by considering the morphism of graduated $K[x_0,x_1]$-modules associated with $\overline{\phi_G}$ and the corresponding morphism of sheaves. In a similar manner $\overline{\phi_G}$ can be recovered from $\widetilde{\phi_G}$ by taking sections on the affine line. $Im\overline{\phi_G}$ is a submodule that defines a convolutional code $\mathcal{C}$. Further, the class of generator matrices of $\mathcal{C}$ with the same column degrees $\{m_i\}_1^n$ corresponds to the class of quotient sheaves represented by Q, which has Hilbert polynomial $P(r)=(n-k)(r+1)+\theta$. Hence, the code $\mathcal{C}$ is represented by $\phi_{\{m_i\}_1^n}(Q)$ as a point of the grassmannian $Grass(H^0(\overset{n}{\underset{i=1}{\oplus}}\mathcal{O}_{\mathbb{P}^1}(m_i)),n+\theta-k)$.

Let us examine now how is the K-subspace given by the point that represents the convolutional code. We have that

$$H^0(\mathbb{P}^1,\mathcal{O}^k_{\mathbb{P}^1})\simeq K^k$$
$$H^0(\mathbb{P}^1,\oplus\mathcal{O}_{\mathbb{P}^1}(m_i))\simeq\underset{i}{\oplus}<x_0^{m_i},x_0^{m_i-1}x_1,\ldots,x_1^{m_i}>\simeq\underset{i}{\oplus}K^{(m_i+1)}=K^{n+\theta}$$

then the map ψ_G, which corresponds to $\overline{\phi_G}$ via $\phi_{\{m_i\}_1^n}$ defined in (2.6) and characterizes $\phi_{\{m_i\}_1^n}(Q)$, is in fact

$$\psi_G : K^k \longrightarrow K^{n+\theta}.$$

Considering the standard bases that we have fixed, if $\overline{G}$ is matrix representation of $\overline{\phi_G}$ with polynomial entries $g^{(ij)}(z) = \sum g_k^{(ij)} z^k$, then $Im\psi_G$ is the row space of the matrix

$$BG = \begin{pmatrix} g_0^{(11)} & g_1^{(11)} & \dots & g_{m_1}^{(11)} & \dots & \dots & g_0^{(1n)} & g_1^{(1n)} & \dots & g_{m_n}^{(1n)} \\ \vdots & & & & & & & & & \vdots \\ g_0^{(k1)} & g_1^{(k1)} & \dots & g_{m_1}^{(k1)} & \dots & \dots & g_0^{(kn)} & g_1^{(kn)} & \dots & g_{m_n}^{(kn)} \end{pmatrix}.$$

$Im\psi_G$ can be regarded as a block code of type $[n+\theta, k]$ and the equivalence of generator matrices of this block code corresponds to the relationship between the sheaves of the associated point in the quotient scheme.

It is immediate to check that the minimum distance of this block code is the 0-th row distance, d_0^r, of the convolutional code $\mathcal{C}$. □

To know exactly which points of a grassmannian represent in the previous way a convolutional code, we need first to describe how is the image of the morphism $\phi_{\{m_i\}_1^n}$. For that it will be needed the following auxiliary matrix.

Definition 2.22. Given a sequence of natural numbers $\{m_i\}_1^n$, with $\sum m_i = \theta$, and a full row rank matrix BG with entries in K and dimensions $k \times n'$, where $n' = n+\theta$, let us take the following partition in blocks of BG:

$$BG = (BG^{(1)}|BG^{(2)}|\dots|BG^{(n)}),$$

so that the block $BG^{(i)}$ has $m_i + 1$ columns. Let us call $g_j^{(i)}$ the j-th column of $BG^{(i)}$, then we define the matrix $\widehat{BG}$ as

$$\left(\begin{array}{cccccc|cc|cccccc} g_0^{(1)} & g_1^{(1)} & g_2^{(1)} & \dots & g_{m_1}^{(1)} & 0 & \dots & \dots & g_0^{(n)} & g_1^{(n)} & g_2^{(n)} & \dots & g_{m_n}^{(n)} & 0 \\ 0 & g_0^{(1)} & g_1^{(1)} & \dots & g_{m_1-1}^{(1)} & g_{m_1}^{(1)} & \dots & \dots & 0 & g_0^{(n)} & g_1^{(n)} & \dots & g_{m_n-1}^{(n)} & g_{m_n}^{(n)} \end{array}\right)$$

being "0" a 0-column vector of length k.

Lemma 2.23. *The points of $Grass(k, H^0(\bigoplus_{i=1}^n \mathcal{O}_{\mathbb{P}^1}(m_i)))$ in the image of $\phi_{\{m_i\}_1^n}$ are the subspaces spanned by the rows of a matrix BG such that the associated matrix $\widehat{BG}$ has maximum rank.*

Proof. First, notice that by fixing the ambient space, $H^0(\mathbb{P}^1, \bigoplus_{i=1}^n \mathcal{O}_{\mathbb{P}^1}(m_i))$, and the dimension k, in particular we fix the numbers $\theta = \sum_{i=1}^n m_i$ and $P(0) = n + \theta - k$, which means fixing the Hilbert polynomial of the quotient sheaves that can be mapped to this grassmannian. All of them have the same Hilbert polynomial, $P(r) = (n-k)(r+1) + \theta$.

Let us consider an element from $Grass(k, H^0(\overset{n}{\underset{i=1}{\oplus}} \mathcal{O}_{\mathbb{P}^1}(m_i)))$ given by the morphism $\psi_G : H^0(\mathcal{O}_{\mathbb{P}}^k) \longrightarrow H^0(\overset{n}{\underset{i=1}{\oplus}} \mathcal{O}_{\mathbb{P}}(m_i))$. There is a commutative diagram

$$\begin{array}{ccc} H^0(\mathcal{O}_{\mathbb{P}}^k) \otimes \mathcal{O}_{\mathbb{P}} & \xhookrightarrow{\psi_G \otimes Id_{\mathcal{O}_{\mathbb{P}}}} & H^0(\overset{n}{\underset{i=1}{\oplus}} \mathcal{O}_{\mathbb{P}}(m_i)) \otimes \mathcal{O}_{\mathbb{P}} \\ \downarrow \wr & & \downarrow \psi \\ \mathcal{O}_{\mathbb{P}}^k & \xrightarrow{\widetilde{\phi_G}} & \overset{n}{\underset{i=1}{\oplus}} \mathcal{O}_{\mathbb{P}}(m_i) \end{array} \tag{2.7}$$

where $\psi_G \otimes Id_{\mathcal{O}_{\mathbb{P}}}$ and ψ are an injective and a surjective morphism respectively. Then the point of the grassmannian defined by ψ_G belongs to the image of $\phi_{\{m_i\}_1^n}$ if the morphism $\widetilde{\phi_G}$ is injective, i. e., the intersection $Im(\psi_G \otimes Id_{\mathcal{O}_{\mathbb{P}}}) \cap Ker\psi$ is equal to (0).

For every $m \in \mathbb{N}$ we have the exact sequence

$$0 \to \mathcal{O}_{\mathbb{P}}(-1)^m \xrightarrow{\phi_m} H^0(\mathcal{O}_{\mathbb{P}}(m)) \otimes \mathcal{O}_{\mathbb{P}} \xrightarrow{\psi_m} \mathcal{O}_{\mathbb{P}}(m) \to 0. \tag{2.8}$$

ϕ_m and ψ_m are determined by two morphisms of $K[x_0, x_1]$-modules which, with the previously chosen standard basis, are represented respectively by the matrices

$$A_m = \begin{pmatrix} x_1 & -x_0 & 0 & \dots & \\ 0 & x_1 & -x_0 & 0 & \dots \\ & & \dots & & \\ 0 & \dots & 0 & x_1 & -x_0 \end{pmatrix} \quad B_m = \begin{pmatrix} x_0^m \\ x_0^{m-1}x_1 \\ \vdots \\ x_1^m \end{pmatrix}.$$

By taking the direct sum for all m_i we have

$$0 \to \overset{n}{\underset{i=1}{\oplus}} \mathcal{O}_{\mathbb{P}}(-1)^{m_i} \xrightarrow{\phi} H^0(\overset{n}{\underset{i=1}{\oplus}} \mathcal{O}_{\mathbb{P}}(m_i)) \otimes \mathcal{O}_{\mathbb{P}} \xrightarrow{\psi} \overset{n}{\underset{i=1}{\oplus}} \mathcal{O}_{\mathbb{P}}(m_i) \to 0 \tag{2.9}$$

where the matrices characterizing ϕ and ψ are

$$A = \begin{pmatrix} A_{m_1} & & & \\ & A_{m_2} & & \\ & & \ddots & \\ & & & A_{m_n} \end{pmatrix} \quad B = \begin{pmatrix} B_{m_1} & & & \\ & B_{m_2} & & \\ & & \ddots & \\ & & & B_{m_n} \end{pmatrix}.$$

By the exactness of the sequence we have $Ker\psi = Im\phi$, which will be of use in our proof.

To characterize the images of $\psi_G \otimes Id_{\mathcal{O}_{\mathbb{P}}}$ and ϕ, we consider the tensor product

of the previous morphisms by $\mathcal{O}_{\mathbb{P}}(1)$. Then, we have the diagram

$$\begin{array}{ccc} & & \bigoplus_{i=1}^{n} \mathcal{O}_{\mathbb{P}}^{m_i} \\ & & \downarrow \phi \\ H^0(\mathcal{O}_{\mathbb{P}}^k) \otimes \mathcal{O}_{\mathbb{P}}(1) & \xrightarrow{BG \otimes Id_{\mathcal{O}_{\mathbb{P}}(1)}} & H^0(\bigoplus_{i=1}^{n} \mathcal{O}_{\mathbb{P}}(m_i)) \otimes \mathcal{O}_{\mathbb{P}}(1) \ . \\ \downarrow \wr & & \downarrow \psi \\ \mathcal{O}_{\mathbb{P}}^k(1) & \xrightarrow{\widetilde{G}} & \bigoplus_{i=1}^{n} \mathcal{O}_{\mathbb{P}}(m_i+1) \end{array}$$

The image of ϕ is determined by the rows of A. In particular, the image of a global section $(\alpha_1^1, \alpha_2^1, \ldots, \alpha_{m_1}^1, \ldots, \alpha_1^n, \ldots, \alpha_{m_n}^n)$ of $\bigoplus_{i=1}^{n} \mathcal{O}_{\mathbb{P}}^{m_i}$, with $\alpha_j^i \in K$, is of the form

$$(\alpha_1^1 x_1, \alpha_2^1 x_1 - \alpha_1^1 x_0, \ldots, \alpha_{m_1}^1 x_1 - \alpha_{m_1-1}^1 x_0, \alpha_{m_1}^1 x_0, \ldots\ldots, \alpha_1^n x_1, \ldots, \alpha_{m_n}^n x_0). \quad (2.10)$$

On the other side, if ψ_G is represented by the matrix

$$BG = \begin{pmatrix} g_0^{(1)} & g_1^{(1)} \ldots & g_{m_1-1}^{(1)} & g_{m_1}^{(1)} & \ldots & g_0^{(n)} & g_1^{(n)} & \ldots & g_{m_n-1}^{(n)} & g_{m_n}^{(n)} \end{pmatrix}$$

where each $g_j^{(i)}$ is a column of BG, then the image by $\psi_G \otimes Id_{\mathcal{O}_{\mathbb{P}}}$ of the global section $u_1 x_1 + u_0 x_0$ of $H^0(\mathcal{O}_{\mathbb{P}}^k) \otimes \mathcal{O}_{\mathbb{P}}(1)$, with $u_1, u_0 \in K^k$, is

$$(u_0^\top g_0^{(1)} x_0 + u_1^\top g_0^{(1)} x_1, \ldots, u_0^\top g_{m_1}^{(1)} x_0 + u_1^\top g_{m_1}^{(1)} x_1, \ldots\ldots, u_0^\top g_{m_n}^{(n)} x_0 + u_1^\top g_{m_n}^{(n)} x_1). \quad (2.11)$$

By comparing each term in the vectors (2.10) and (2.11) we have that $\psi_G \otimes Id_{\mathcal{O}_{\mathbb{P}}}(u_1 x_1 + u_0 x_0)$ is in the image of ϕ if and only if for all $1 \leq j \leq n$

$$\begin{array}{l} u_0^\top g_0^{(j)} = 0 \\ u_1^\top g_i^{(j)} = -u_0^\top g_{i+1}^{(j)} = \alpha_{i+1}^{(j)} \quad \forall\ 0 \leq i \leq m_j - 1 \\ u_1^\top g_{m_j}^{(j)} = 0 \end{array} ,$$

which can be written in matrix form as $(u_0^\top, u_1^\top)\widehat{BG} = 0$. As a result, the intersection $Im(\psi_G \otimes Id_{\mathcal{O}_{\mathbb{P}}}) \cap Im\phi$ is the image by $\psi_G \otimes Id_{\mathcal{O}_{\mathbb{P}}}$ of the sections $u_1 x_1 + u_0 x_0$ such that $(u_0^\top, u_1^\top)\widehat{BG} = 0$. Then, the condition $Im\phi \cap Im(\psi_G \otimes Id_{\mathcal{O}_{\mathbb{P}}}) = (0)$ is equivalent to the fact that there does not exist a nonzero linear combination of the rows of $\widehat{BG}$, i. e., this matrix has maximum row rank.

Further, it is clear that if the condition is satisfied for one of the representing matrices of the subspace defined by the point, it is satisfied for all of them. □

We are interested in the points of $Im\phi_{\{m_i\}_1^n}$ which have as counterimage quotient sheaves over $\mathbb{P}_K^1$ with torsion only in P_∞, i. e., such that when taking sections on $\mathbb{A}_K^1$ the morphism defining them is represented by a basic polynomial matrix.

Lemma 2.24. *The set of points of* $Grass(k, H^0(\overset{n}{\underset{i=1}{\oplus}} \mathcal{O}_{\mathbb{P}^1}(m_i)))$ *representing basic polynomial matrices is an open subset.*

Proof. Recall that a condition for a polynomial matrix $M(z)$ to be basic is that $gcd\{k \times k - \text{minors of } M(z)\} = 1$, i. e., the ideal generated by its $k \times k$-minors is the whole ring.

For each constant matrix M representing a point $P \in Grass(k, H^0(\overset{n}{\underset{i=1}{\oplus}} \mathcal{O}_{\mathbb{P}^1}(m_i)))$ consider the polynomial matrix $M(z)$ given by multiplying M on the right by the matrix

$$\begin{pmatrix} 1\, z \ldots z^{m_1} & 0 & 0 \\ & \ddots & \\ 0 & 0 & 1\, z \ldots z^{m_n} \end{pmatrix}^{\top} \tag{2.12}$$

Let us denote the grassmannian by $\mathbb{G}$ and consider the scheme $\mathbb{G} \times \mathbb{P}^1$, and its projection $\pi : \mathbb{G} \times \mathbb{P}^1 \longrightarrow \mathbb{G}$, which is a proper morphism.

Let $\mathcal{I}$ be the ideal sheaf over $\mathbb{G} \times \mathbb{P}^1$ such that $\mathcal{I}_P$ is the ideal generated by the $k \times k -$ minors of $M(z)$ (note that they don't depend on the matrix representation M of P). Let us consider the projection

$$(\mathcal{I})_0 \xrightarrow{\pi} \mathbb{G}.$$

$Im\pi$ is a closed subset of $\mathbb{G}$ since π is a proper morphism.

Then, $P \in \mathbb{G}$ represents a basic polynomial matrix if and only if $\mathcal{I}_P$ is the whole ring, if and only if $(\mathcal{I}_P)_0$ has no zeros, i. e.,

$$\pi^{-1}(P) = (\mathcal{I}_P)_0 \begin{cases} = \emptyset, & \text{if } P \text{ is basic} \\ \neq \emptyset, & \text{if } P \text{ is not basic} \end{cases}.$$

Therefore, the basic polynomial matrices are represented by the points in the set $\{P \in \mathbb{G} | \pi^{-1}(P) = \emptyset\} \subset \mathbb{G}$, which is the complementary of $Im\pi$ and hence, open. □

The two previous lemmas determine the points in $Grass(k, H^0(\overset{n}{\underset{i=1}{\oplus}} \mathcal{O}_{\mathbb{P}^1}(m_i)))$ that represent basic generator matrices of $[n, k]$ convolutional codes with column degrees $m_1, \ldots, m_n$. According to them we have the following result.

Theorem 2.25. *The set of points of* $Grass(k, H^0(\overset{n}{\underset{i=1}{\oplus}} \mathcal{O}_{\mathbb{P}^1}(m_i)))$ *representing convolutional codes of length* n *and dimension* k *is an open subset.*

Proof. As we have seen, a point of $Grass(k, H^0(\overset{n}{\underset{i=1}{\oplus}} \mathcal{O}_{\mathbb{P}^1}(m_i)))$ determined by a constant matrix M is associated with the convolutional code generated by the polynomial matrix $M(z)$ resulting from right multiplication of M by the matrix (2.12) if $M(z)$ is column reduced, basic and has maximum rank.

In Lemmas 2.24 and 2.23 it is proven that the second and third conditions, respectively, define open subsets of $Grass(k, H^0(\overset{n}{\underset{i=1}{\oplus}} \mathcal{O}_{\mathbb{P}^1}(m_i)))$.

To examine the meaning of the first condition notice that the matrix consisting on the column leading coefficients of $M(z)$ is precisely the matrix made up with the columns $\{\sum_i m_i + i\}_{i=1}^n$ of M (the last column of each block when M is partitioned in blocks of length $m_i + 1$). Then, $M(z)$ is column reduced if and only if this submatrix of M has maximum rank. This condition defines also an open subset in $Grass(k, H^0(\overset{n}{\underset{i=1}{\oplus}} \mathcal{O}_{\mathbb{P}^1}(m_i)))$.

The intersection of these three open subsets are the points of the grassmannian that define convolutional codes. □

2.4.2 Classification of Sheaves versus Classification of Codes

The previous construction characterizes as quotient sheaves basic generator matrices of a code with a fixed sequence of column degrees. However, different generator matrices of the same code may correspond to quotients of different sheaves.

This is due to the fact that we are dealing with objects, sheaves on the one side and submodules on the other, having different equivalences. In the case of sheaves this equivalence is given by an isomorphism, the matrix representations of which have constant coefficients. In the case of submodules, as we have seen, equivalence is given in terms of an unimodular matrix. In fact non-isomorphic sheaves may result in equivalent submodules.

Recall that the way to obtain the corresponding $K[z]$-submodule of a sheaf over $\mathbb{P}^1_K$ is by taking sections on the affine line $\mathbb{A}^1_K$. Then, it is possible to find non-isomorphic sheaves, which coincide on the affine part, i. e. which have the same sections on $\mathbb{A}^1_K$, but don't coincide globally, i. e. they don't have the same global sections.

Consider an injective morphism ϕ of coherent sheaves of the same rank k

$$0 \longrightarrow \mathcal{F}' \xrightarrow{\phi} \mathcal{F} \longrightarrow \mathcal{F}'' \longrightarrow 0$$

such that $\mathcal{F}''$ is centered in the infinite point, P_∞, i. e. $\mathcal{F}''_P = 0 \ \forall P \neq P_\infty$. Then, when taking sections on the affine line we have

$$0 \longrightarrow H^0(\mathbb{A}^1_K, \mathcal{F}') \xrightarrow{\phi_{\mathbb{A}}} H^0(\mathbb{A}^1_K, \mathcal{F}) \longrightarrow \underbrace{H^0(\mathbb{A}^1_K, \mathcal{F}'')}_{0} \longrightarrow \dots$$

and hence $\phi_{\mathbb{A}}$ is an isomorphism of $\mathcal{O}_{\mathbb{A}}$-modules, i. e., $\phi_{\mathbb{A}}$ is an injective map with an inverse, i. e., the matrix of $\phi_{\mathbb{A}}$ is invertible, i. e., its determinant belongs to K. This is exactly the algebraic interpretation of an *unimodular matrix*.

However, if we take global sections we get

$$0 \longrightarrow H^0(\mathbb{P}^1_K, \mathcal{F}') \xrightarrow{\phi_{\mathbb{P}}} H^0(\mathbb{P}^1_K, \mathcal{F}) \longrightarrow \underbrace{\mathcal{F}''_{P_\infty}}_{\neq 0} \longrightarrow \dots$$

which means that there is no isomorphism between the sheaves $\mathcal{F}$ and $\mathcal{F}'$, and as a result also not between the corresponding block codes.

Let us illustrate it with an example.

Example 2.26. Let us consider the field $K = \mathbb{F}_5$, the finite field with five elements, and let us denote by R the ring $\mathbb{F}_5[z]$ and by $\bar{R}$ the ring $\mathbb{F}_5[x_0, x_1]$. Consider the coherent sheaves $\mathcal{F}$, $\mathcal{F}'$, $\mathcal{F}''$ corresponding to the $\bar{R}$-modules generated by

$$\begin{array}{rcl} \langle (3x_0, 3x_0, 4x_0), (4x_0 + 2x_1, x_0, 2x_1) \rangle & \subset & \bar{R}[1] \oplus \bar{R} \oplus \bar{R}[1] \\ \langle (3x_0, 3x_0, 4x_0), (4x_0, x_0 + 3x_1, x_1) \rangle & \subset & \bar{R} \oplus \bar{R}[1] \oplus \bar{R}[1] \\ \langle (3x_0, 3x_0, 4x_0), (4x_0 + 3x_1, x_0 + x_1, 0) \rangle & \subset & \bar{R}[1] \oplus \bar{R}[1] \oplus \bar{R} \end{array} .$$

Then, $H^0(\mathbb{A}^1, \mathcal{F})$, $H^0(\mathbb{A}^1, \mathcal{F}')$ and $H^0(\mathbb{A}^1, \mathcal{F}'')$ are respectively the R-modules generated by the rows of the matrices

$$\begin{pmatrix} 3 & 3 & 4 \\ 4+2z & 1 & 2z \end{pmatrix} \quad \begin{pmatrix} 3 & 3 & 4 \\ 4 & 1+3z & z \end{pmatrix} \quad \begin{pmatrix} 3 & 3 & 4 \\ 4+3z & 1+z & 0 \end{pmatrix}$$

and they are submodules of

$$R[1] \oplus R \oplus R[1] \quad , \quad R \oplus R[1] \oplus R[1] \quad , \quad R[1] \oplus R[1] \oplus R$$

In fact these submodules *generate the same submodule of* $R[1]^{\oplus 3}$. However $H^0(\mathbb{P}^1, \mathcal{F})$, $H^0(\mathbb{P}^1, \mathcal{F}')$ and $H^0(\mathbb{P}^1, \mathcal{F}'')$ are the subvector spaces of $\mathbb{F}_5^5$ generated by the matrices

$$\begin{pmatrix} 3 & 0 & 3 & 4 & 0 \\ 4 & 2 & 1 & 0 & 2 \end{pmatrix} \quad \begin{pmatrix} 3 & 3 & 0 & 4 & 0 \\ 4 & 1 & 3 & 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 3 & 0 & 3 & 0 & 4 \\ 4 & 3 & 1 & 1 & 0 \end{pmatrix}$$

and *these subvector spaces are different.*

Therefore, this fundamental fact when relating the concepts of convolutional codes and sheaves, which however we couldn't find explicitly mentioned in the literature, will be a key element of our classification.

2.4.3 Classification of Convolutional Codes

In order to use quotient sheaves in the classification of convolutional codes we need to identify the different classes of sheaves which by taking sections on $\mathbb{A}^1_K$ result in the same submodule. We will consider their inclusion on a "bigger" sheaf, which will contain all the sheaves corresponding to the same convolutional code. This allows to make a further step by characterizing each code as a unique point of a grassmannian.

Theorem 2.27. *Given a convolutional code $\mathcal{C}$ of type $[n, k, \delta]$ and memory ν_k there is a unique sequence of integers $\{n_i\}_{i=1}^n$, which will be called the* minimal column indices *of the code, bounded by ν_k, such that the code (by means of all its canonical generator matrices) is represented by a unique point of the grassmannian $Grass(k(\nu_k + 1) - \delta, H^0(\bigoplus_{i=1}^{n} \mathcal{O}_{\mathbb{P}^1}(n_i)))$.*

Proof. As in the representation of generator matrices with certain column degrees, we will first represent polynomial generator matrices as quotients of a certain sheaf, and then via the inclusion of the *Quot* scheme in a grassmannian, the generator

matrices of a code will be represented as points of this variety. However, in this case the column degrees are not fixed and all generator matrices of a convolutional code (possibly with different column degrees) must be represented by an unique quotient sheaf and hence by an unique point of a grassmannian.

The proof follows this outline: first we determine the minimal sheaf which contains all generator matrices of a given convolutional code and we take the corresponding quotient sheaf. In particular, this sheaf will determine the *minimal column indices* of the code. Then we consider the representation of the quotient sheaves as points of a grassmannian, and we give a description of the K-vector subspace representing a convolutional code in terms of one of its generator matrices.

Let us consider two polynomial basic generator matrices G_1, G_2 of the same code with row degrees, which without loss of generality we will assume ordered, $\nu_1 \leq \ldots \leq \nu_k$, the Forney indices of the code. This means in particular that G_1, G_2 are canonical matrices. Then there is an unimodular polynomial matrix

$$U = \begin{pmatrix} p_{11} & \cdots & p_{1k} \\ \vdots & & \vdots \\ p_{k1} & \cdots & p_{kk} \end{pmatrix}$$

with $degp_{ij} \leq \nu_i - \nu_j$ for $\nu_i \geq \nu_j$ and $p_{ij} = 0$ otherwise, such that $G_2 = UG_1$. G_1 and G_2 may have in general different sets of column indices.

As we have seen, G_1 and G_2 correspond to sheaves morphisms

$$G_1 \rightsquigarrow \mathcal{O}^k \xrightarrow{\phi_1} \bigoplus_{i=1}^{n} \mathcal{O}(m_i')$$
$$G_2 \rightsquigarrow \mathcal{O}^k \xrightarrow{\phi_2} \bigoplus_{i=1}^{n} \mathcal{O}(m_i'')$$

defining the corresponding quotient sheaves.

U (as a matrix with homogenized components) corresponds to the sheaves morphism

$$\mathcal{O}^k \xhookrightarrow{\phi_U} \bigoplus_{i=1}^{k} \mathcal{O}(\nu_k - \nu_i) \hookrightarrow \mathcal{O}(N)^k$$

with $N = \nu_k - \nu_1$, which is fixed for all the canonical generator matrices of a code. Then, we have a commutative diagram

$$\begin{array}{ccc} \mathcal{O}^k & \xrightarrow{\phi_1} & \bigoplus_{i=1}^{n} \mathcal{O}(m_i') \\ \downarrow & & \downarrow \\ \mathcal{O}(N)^k & \xrightarrow{\phi_1 \otimes Id_N} & \bigoplus_{i=1}^{n} \mathcal{O}(m_i' + N) \\ \uparrow \phi_U & & \uparrow j_U \\ \mathcal{O}^k & \xrightarrow{\phi_2} & \bigoplus_{i=1}^{n} \mathcal{O}(m_i'') \end{array}$$

where $j_U = \overset{n}{\underset{i=1}{\oplus}} j_U^i$, with $j_U^i : \mathcal{O}(m_i'') \longrightarrow \mathcal{O}(m_i' + N)$, is an immersion that increases by $m_i' + N - m_i''$ the degree in the component i. j_U depends on U and makes the diagram commutative.

Moreover, as G_1, G_2 have the same row indices, then $m_i', m_j'' \leq \nu_k$ for all $i, j \leq n$. In particular from the previous diagram it follows that both $Im\phi_1$ and $Im\phi_2$ are included in the two sheaves $\overset{n}{\underset{i=1}{\oplus}} \mathcal{O}(n_i')$ and $\overset{n}{\underset{i=1}{\oplus}} \mathcal{O}(n_i'')$, where we define $n_i' = min\{m_i' + N, \nu_k\}$, $n_i'' = min\{m_i'' + N, \nu_k\}$ for all i. We can take then the minimum set of indices $\{n_i\}$ such that $Im\phi' \subset \overset{n}{\underset{i=1}{\oplus}} \mathcal{O}(n_i)$ for every ϕ' representing a canonical generator matrix of the code, and $n_i \leq n_i'$ $\forall i$ for every collection of indices $\{n_i'\}$ that verify that condition. These indices are defined to be the *minimal column indices* of the code. In fact, each n_i is the maximum polynomial degree on the i-th column of any canonical generator matrix of the code. Then, we have

$$\begin{array}{ccc} \mathcal{O}^k & \xrightarrow{\phi_1} & \overset{n}{\underset{i=1}{\oplus}} \mathcal{O}(m_i') \\ \downarrow & & \downarrow \\ \overset{k}{\underset{i=1}{\oplus}} \mathcal{O}(\nu_k - \nu_i) & \xrightarrow{\widehat{\phi_1}} & \overset{n}{\underset{i=1}{\oplus}} \mathcal{O}(n_i) \end{array} .$$

We will see next that an injective morphism of sheaves ϕ'' defined by another canonical polynomial matrix generates the same convolutional code as ϕ_1 if and only if there is an inclusion of $Im\phi''$ in the sheaf $Im(\widehat{\phi_1}) \subset \overset{n}{\underset{i=1}{\oplus}} \mathcal{O}(n_i)$. Hence we will be able to represent the convolutional code as the quotient sheaf of $\overset{n}{\underset{i=1}{\oplus}} \mathcal{O}(n_i)$ defined by $\widehat{\phi_1}$.

Indeed, an injective morphism $\phi'' : \mathcal{O}^k \longrightarrow \overset{n}{\underset{i=1}{\oplus}} \mathcal{O}(m_i'')$, defines the same convolutional code as ϕ_1 if there exists an injective morphism ϕ_U with cokernel centered in P_∞ such that

$$\begin{array}{ccc} \overset{k}{\underset{i=1}{\oplus}} \mathcal{O}(\nu_k - \nu_i) & \xrightarrow{\widehat{\phi_1}} & \overset{n}{\underset{i=1}{\oplus}} \mathcal{O}(n_i) \\ \uparrow{\scriptstyle \phi_U} & & \uparrow{\scriptstyle j_U} \\ \mathcal{O}^k & \xrightarrow{\phi''} & \overset{n}{\underset{i=1}{\oplus}} \mathcal{O}(m_i'') \end{array} \tag{2.13}$$

is a commutative diagram. Then for any ϕ'' defined by a canonical generator matrix of the code, the image of $\mathcal{O}^k \hookrightarrow \overset{n}{\underset{i=1}{\oplus}} \mathcal{O}(m_i'') \hookrightarrow \overset{n}{\underset{i=1}{\oplus}} \mathcal{O}(n_i)$ is a k-rank subsheaf of $Im(\widehat{\phi_1})$.

On the other side, if $\phi'' : \mathcal{O}^k \hookrightarrow \overset{n}{\underset{i=1}{\oplus}} \mathcal{O}(m_i'')$ is an injective morphism such that there is an inclusion $j : Im\phi'' \hookrightarrow Im(\widehat{\phi_1})$, then we can define an injective morphism

$\phi_U : \mathcal{O}^k \hookrightarrow \bigoplus_{i=1}^{k} \mathcal{O}(\nu_k - \nu_i)$ so that the corresponding diagram (2.13) is commutative.

The sheaf $Im(\widehat{\phi_1})$ doesn't depend on ϕ_1. Recall that given two morphisms of sheaves ϕ_1, ϕ_2 defining the convolutional code, they are determined by two equivalent generator matrices G_1, G_2 with the same row degrees. Our aim is to prove that $Im(\widehat{\phi_1}) = Im(\widehat{\phi_2})$.

Let us call $\widetilde{G}_1$, $\widetilde{G}_2$ the homogeneous polynomial matrices representing the morphisms of graduated modules corresponding to $\widehat{\phi_1}$, $\widehat{\phi_2}$, and let $g_i^{(j)}$ the vector on the i-th row of $\widetilde{G}_j$. $g_i^{(1)}$, $g_i^{(2)}$ are homogeneous polynomial vectors of degree ν_i. Then, $Im(\widehat{\phi_i})$ is generated by

$$K[x_0, x_1][\nu_k - \nu_1]g_1^{(i)} \oplus K[x_0, x_1][\nu_k - \nu_2]g_2^{(i)} \oplus \ldots \oplus K[x_0, x_1]g_k^{(i)}$$

Let us assume that the row degrees of the matrices $\widetilde{G}_1$, $\widetilde{G}_2$ take s different values $\mu_1 = \nu_1 < \mu_2 < \ldots < \mu_s = \nu_k$, so that l_r rows have row degree μ_r, and let us denote $g_{\mu_r,1}^{(i)}, \ldots, g_{\mu_r,l_r}^{(i)}$ the l_r rows of $\widetilde{G}_j$ with degree μ_r. Then $Im(\widehat{\phi_i})$ is generated by

$$K[x_0, x_1][\mu_s - \mu_1]^{l_1} \begin{pmatrix} g_{\mu_1,1}^{(i)} \\ \vdots \\ g_{\mu_1,l_1}^{(i)} \end{pmatrix} \oplus \ldots\ldots \oplus K[x_0, x_1]^{l_s} \begin{pmatrix} g_{\mu_s,1}^{(i)} \\ \vdots \\ g_{\mu_s,l_s}^{(i)} \end{pmatrix} .$$

We prove that for all $j \leq s$

$$K[x_0, x_1][\mu_s - \mu_j]^{l_j} \begin{pmatrix} g_{\mu_j,1}^{(2)} \\ \vdots \\ g_{\mu_j,l_j}^{(2)} \end{pmatrix} \subseteq K[x_0, x_1][\mu_s - \mu_1]^{l_1} \begin{pmatrix} g_{\mu_1,1}^{(1)} \\ \vdots \\ g_{\mu_1,l_1}^{(1)} \end{pmatrix} \oplus \ldots \oplus K[x_0, x_1][\mu_s - \mu_j]^{l_j} \begin{pmatrix} g_{\mu_j,1}^{(1)} \\ \vdots \\ g_{\mu_j,l_j}^{(1)} \end{pmatrix} . \tag{2.14}$$

Considering that G_1 and G_2 are equivalent and as $\tilde{G}_1$, $\tilde{G}_2$ have the same row degrees, there exists an unimodular matrix $\tilde{U}(x_0, x_1) = (p_{ij}(x_0, x_1))$, with p_{ij} a homogeneous polynomial of degree $\nu_i - \nu_j$ for $\nu_i \geq \nu_j$ and 0 otherwise, such that $\tilde{G}_2 = \tilde{U}\tilde{G}_1$.

$\tilde{G}_2 = \tilde{U}\tilde{G}_1$ means in particular that there exist matrices of maximum rank $M_i \in \mathbb{M}_{l_j \times l_{j-i}}(K[x_0, x_1][\mu_j - \mu_{j-i}])$ for all $i < j$ such that

$$\begin{pmatrix} g_{\mu_j,1}^{(2)} \\ \vdots \\ g_{\mu_j,l_j}^{(2)} \end{pmatrix} = M_0 \begin{pmatrix} g_{\mu_j,1}^{(1)} \\ \vdots \\ g_{\mu_j,l_j}^{(1)} \end{pmatrix} + M_1 \begin{pmatrix} g_{\mu_{j-1},1}^{(1)} \\ \vdots \\ g_{\mu_{j-1},l_{j-1}}^{(1)} \end{pmatrix} + \ldots + M_{j-1} \begin{pmatrix} g_{\mu_1,1}^{(1)} \\ \vdots \\ g_{\mu_1,l_1}^{(1)} \end{pmatrix}$$

and taking into account that $K[x_0, x_1][\mu_s - \mu_j]^{l_j} M_i \subset K[x_0, x_1][\mu_s - \mu_{j-i}]^{l_{j-i}}$ for all $i < j$ we get our result.

As (2.14) holds for all $j \leq s$ then $Im(\widehat{\phi_2}) \subseteq Im(\widehat{\phi_1})$. With the same argument we get $Im(\widehat{\phi_1}) \subseteq Im(\widehat{\phi_2})$ and hence $Im(\widehat{\phi_1}) = Im(\widehat{\phi_2})$.

Then, the morphism

$$\bigoplus_{i=1}^{k} \mathcal{O}(\nu_k - \nu_i) \xrightarrow{\widehat{\phi_1}} \bigoplus_{i=1}^{n} \mathcal{O}(n_i) \tag{2.15}$$

makes it possible to characterize every canonical generator matrix of the code, and hence the code itself, by a quotient sheaf Q' of $\overset{n}{\underset{i=1}{\oplus}}\mathcal{O}(n_i)$ with Hilbert polynomial

$$\begin{aligned} P(Q',r) &= P(\overset{n}{\underset{i=1}{\oplus}}\mathcal{O}(n_i),r) - P(\overset{k}{\underset{i=1}{\oplus}}\mathcal{O}(\nu_k-\nu_i),r) = \\ &= \sum_{i=1}^{n}(n_i+1+r) - \sum_{i=1}^{k}(\nu_k-\nu_i+1+r) = \\ &= (n-k)(r+1) + \sum_{i=1}^{n} n_i + \delta - k\nu_k \end{aligned} .$$

We can describe now how the quotient sheaves of $\overset{n}{\underset{i=1}{\oplus}}\mathcal{O}(n_i)$, in particular those associated in the previous way with a convolutional code, are represented as points of a grassmannian. The inclusion

$$\begin{array}{ccc} \mathcal{O}^k & \xrightarrow{\phi_1} & \overset{n}{\underset{i=1}{\oplus}}\mathcal{O}(m_i) \\ \downarrow & & \downarrow \\ \overset{k}{\underset{i=1}{\oplus}}\mathcal{O}(\nu_k-\nu_i) & \xrightarrow{\widehat{\phi_1}} & \overset{n}{\underset{i=1}{\oplus}}\mathcal{O}(n_i) \end{array}$$

where both vertical morphisms are the natural inclusions, gives rise to an inclusion

$$\begin{array}{ccc} Quot(\overset{n}{\underset{i=1}{\oplus}}\mathcal{O}_{\mathbb{P}^1}(m_i), P_Q(r)) & & \\ \downarrow & \searrow^{\widetilde{\phi}_{\{m_i\}}} & \\ Quot(\overset{n}{\underset{i=1}{\oplus}}\mathcal{O}_{\mathbb{P}^1}(n_i), P_{Q'}(r)) & \xrightarrow{\phi_{\{n_i\}_1^n}} & Grass(k(\nu_k+1)-\delta, H^0(\overset{n}{\underset{i=1}{\oplus}}\mathcal{O}_{\mathbb{P}^1}(n_i))) \end{array} \tag{2.16}$$

as $dim H^0(\mathbb{P}^1, \overset{n}{\underset{i=1}{\oplus}}\mathcal{O}(n_i)) - P'_Q(0) = k(\nu_k+1)-\delta$. Recall that $P_Q(r) = (n-k)(r+1)+\sum m_i$ is the Hilbert polynomial of the quotient sheaf $Q = Coker\phi_1$. $\widetilde{\phi}_{\{m_i\}}(Q)$ will be the subspace of $H^0(\overset{n}{\underset{i=1}{\oplus}}\mathcal{O}_{\mathbb{P}^1}(n_i))$ defined by $\phi_{\{n_i\}}(Q')$, with $Q' = Coker(\widehat{\phi_1})$, which by the inclusion of the quotient schemes is the only point in $Im\phi_{\{n_i\}}$ that thought as a subspace contains the k-dimensional subspace of $H^0(\overset{n}{\underset{i=1}{\oplus}}\mathcal{O}_{\mathbb{P}^1}(m_i)) \subset H^0(\overset{n}{\underset{i=1}{\oplus}}\mathcal{O}_{\mathbb{P}^1}(n_i))$ defined by $\phi_{\{m_i\}}(Q)$ as in (2.6).

Let us describe now, considering our fixed standard bases, the subspace of $H^0(\overset{n}{\underset{i=1}{\oplus}}\mathcal{O}_{\mathbb{P}^1}(n_i))$ corresponding to the sheaves morphism $\widehat{\phi_1} : \overset{k}{\underset{i=1}{\oplus}}\mathcal{O}(\nu_k-\nu_i) \longrightarrow \overset{n}{\underset{i=1}{\oplus}}\mathcal{O}(n_i)$.

If $Q' = Coker(\widehat{\phi_1})$, then $\phi_{\{n_i\}}(Q')$ is the subspace defined by the sequence

$$0 \to H^0(\mathbb{P}^1, \overset{k}{\underset{i=1}{\oplus}}\mathcal{O}(\nu_k-\nu_i)) \to H^0(\mathbb{P}^1, \overset{n}{\underset{i=1}{\oplus}}\mathcal{O}(n_i)) \to H^0(\mathbb{P}^1, Q') \to 0 .$$

$\phi_{\{n_i\}}(Q')$ is determined by ϕ_1, which can be represented by the polynomial matrix

$$G_1 = \begin{pmatrix} g^{(11)} & \dots & g^{(1n)} \\ \vdots & & \vdots \\ g^{(k1)} & \dots & g^{(kn)} \end{pmatrix}$$

with $g^{(ij)} = \sum g_k^{(ij)} x_0^{m_j-k} x_1^k$ an homogeneous polynomial of degree m_j. Accordingly, the corresponding matrix representing $\widehat{\phi_1}$ is $\widetilde{G}_1 = (\widetilde{g}^{(ij)})$ where $\widetilde{g}^{(ij)} = \sum g_k^{(ij)} x_0^{\nu_i-k} x_1^k$. Note that in fact $g_k^{(ij)} = 0 \; \forall \, k > min\{\nu_i, m_j\}$. Then, for each $1 \leq i \leq k$, the image of the morphism

$$H^0(\mathbb{P}^1, \mathcal{O}(\nu_k - \nu_i)) \simeq K^{\nu_k-\nu_i+1} \longrightarrow H^0(\mathbb{P}^1, \overset{n}{\underset{i=1}{\oplus}} \mathcal{O}(n_i)) \simeq K^{\sum n_i+n}$$

is the rowspan of the constant matrix $(\mathrm{g}^{(i1)} \dots \mathrm{g}^{(in)})$, where

$$\mathrm{g}^{(ij)} = \begin{pmatrix} g_0^{(ij)} & \dots & \dots & g_{m_j}^{(ij)} & 0 & \dots & \dots 0 \\ 0 \dots & g_0^{(ij)} & \dots & \dots & g_{m_j}^{(ij)} & 0 & \dots 0 \\ & \ddots & & & \ddots & & \\ 0 \dots & \dots 0 & g_0^{(ij)} & \dots & \dots & g_{m_j}^{(ij)} & 0 \dots 0 \end{pmatrix}$$

and has dimensions $\nu_k - \nu_i + 1 \times n_i + 1$.

Then, the morphism

$$H^0(\mathbb{P}^1, \overset{k}{\underset{i=1}{\oplus}} \mathcal{O}(\nu_k - \nu_i)) \longrightarrow H^0(\mathbb{P}^1, \overset{n}{\underset{i=1}{\oplus}} \mathcal{O}(n_i))$$

is represented by the matrix

$$\begin{pmatrix} \mathrm{g}^{(11)} & \dots & \mathrm{g}^{(1n)} \\ \vdots & & \vdots \\ \mathrm{g}^{(k1)} & \dots & \mathrm{g}^{(kn)} \end{pmatrix} = \tag{2.17}$$

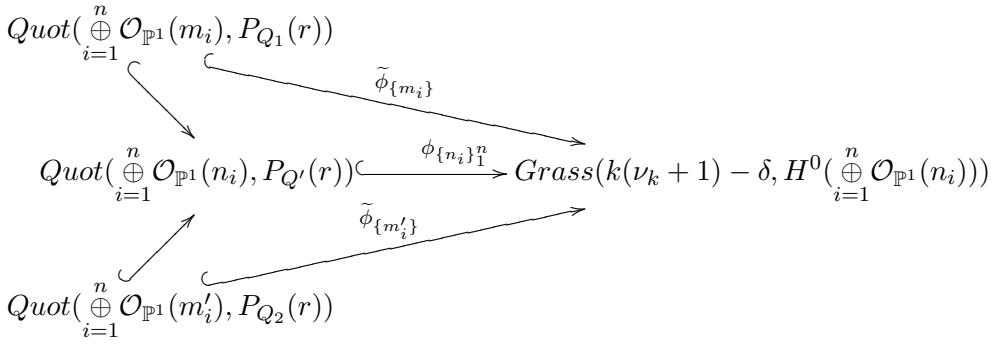
$$\begin{array}{l} \nu_k - \nu_i + 1 \left\{ \right. \\ \dots \\ \nu_k - \nu_i + 1 \left\{ \right. \\ \dots \\ \nu_k - \nu_k + 1 \left\{ \right. \end{array} \left(\begin{array}{ccccccc|cc|ccccccc} g_0^{(11)} & \dots & \dots & g_{m_1}^{(11)} & 0 & \dots & \dots 0 & \dots & \dots & g_0^{(n1)} & \dots & \dots & g_{m_n}^{(n1)} & 0 & \dots & \dots 0 \\ 0 \dots & g_0^{(11)} & \dots & \dots & g_{m_1}^{(11)} & 0 \dots & \dots 0 & \dots & \dots & 0 \dots & g_0^{(n1)} & \dots & \dots & g_{m_n}^{(n1)} & 0 \dots & \dots 0 \\ 0 \dots & \dots 0 & g_0^{(11)} & \dots & \dots & g_{m_1}^{(11)} & 0 \dots 0 & \dots & \dots & 0 \dots & \dots 0 & g_0^{(n1)} & \dots & \dots & g_{m_n}^{(n1)} & 0 \dots 0 \\ \hline \dots & & & & & & & & & & & & & & & \\ \hline g_0^{(k1)} & g_1^{(k1)} & \dots & g_{m_1}^{(k1)} & 0 & \dots & 0 & \dots & \dots & g_0^{(kn)} & g_1^{(kn)} & \dots & g_{m_n}^{(kn)} & 0 & \dots & 0 \end{array} \right),$$

the rows of which generate the image point $\phi_{\{n_i\}}(Q') \in Gr(k(\nu_k+1)-\delta, H^0(\mathbb{P}^1, \overset{n}{\underset{i=1}{\oplus}} \mathcal{O}(n_i)))$ as a vector subspace. Notice that by the definition of the indices n_i (there is a canonical generator matrix of the convolutional code with i-th column degree equal to n_i) in each of the submatrices

$$\begin{pmatrix} \mathrm{g}^{(1i)} \\ \vdots \\ \mathrm{g}^{(ki)} \end{pmatrix}$$

the rightmost column is different from the 0-column.

As a conclusion, given any two canonical generator matrices G_1, G_2 of the same convolutional code with row degrees $\nu_1 \leq \ldots \leq \nu_k$ and column degrees $\{m_i\}_1^n$ and $\{m'_i\}_1^n$ respectively, represented by the quotient sheaves Q_1, Q_2, we have



as well as

$$\phi_{\{m_i\}}(Q_1) \hookrightarrow \widetilde{\phi}_{\{m_i\}}(Q_1)\ , \qquad \phi_{\{m'_i\}}(Q_2) \hookrightarrow \widetilde{\phi}_{\{m'_i\}}(Q_2)\ ,$$

and therefore

$$\phi_{\{m_i\}}(Q_1), \phi_{\{m'_i\}}(Q_2) \subset \widetilde{\phi}_{\{m_i\}}(Q_1) = \widetilde{\phi}_{\{m'_i\}}(Q_2) = \phi_{\{n_i\}}(Q').$$

As a consequence, both matrices G_1, G_2 are represented by the same point in $Quot(\bigoplus_{i=1}^{n}\mathcal{O}_{\mathbb{P}^1}(n_i), P_{Q'}(r))$ and hence in $Grass(k(\nu_k+1)-\delta, H^0(\bigoplus_{i=1}^{n}\mathcal{O}_{\mathbb{P}^1}(n_i)))$, and this point is generated as a vector subspace by the rows of the matrix (2.17). □

Remarks 2.28.

1. There is a correspondence between the equivalence of canonical generator matrices of convolutional codes and the equivalence of quotient sheaves of $\bigoplus_{i=1}^{n}\mathcal{O}_{\mathbb{P}^1}(n_i)$. Notice that the equivalence is not given in terms of the quotient sheaves that correspond to the generator matrices of a code, but in terms of the sheaves that contain those associated with these generator matrices.

2. The words of the block code generated by the matrix (2.17) correspond to the codewords of the convolutional code with maximum degree ν_k. This set of convolutional codewords doesn't depend on the generator matrix of the code. This proves in another way that the subspace which represents the convolutional code as a point of the grassmannian doesn't depend on the the generator matrix, or equivalently on the morphism $\mathcal{O}^k \longrightarrow \bigoplus_{i=1}^{n}\mathcal{O}(m_i)$, chosen to carry out the previous construction.

3. In the case where $\nu_i = \nu_k\ \forall i \leq k$, all Forney indices are equal, the equivalence between canonical generator matrices is given by constant matrices which means that both constructions from Theorem 2.21 and Theorem 2.27 are the same one.

4. Notice that $n_j < \nu_k$ if and only if for some canonical matrix of the code its j-th column degree is $m_j < \nu_1$.

Theorem 2.29. *Every convolutional code of type $[n,k]$ and memory m is represented by a point in $Gr(\lambda, K^l)$ for some λ, where $l = n(m+1)$. The codes with degree δ are represented precisely by the subspaces of K^l with dimension $\lambda = k(m+1) - \delta$.*

Proof. By the previous theorem it is known that a $[n,k,\delta]$ code with memory m and minimal column indices $\{n_i\}_i$ can be represented by a $k(m+1)-\delta$-subspace of $H^0(\overset{n}{\underset{i=1}{\oplus}} \mathcal{O}_{\mathbb{P}^1}(n_i))$. Further, all $[n,k,\delta]$ codes with *memory* $m = \nu_k$, can be represented as points of the same grassmannian by considering the inclusions

$$Gr(\lambda, H^0(\mathbb{P}^1, \overset{n}{\underset{i=1}{\oplus}} \mathcal{O}(n_i))) \hookrightarrow Gr(\lambda, H^0(\mathbb{P}^1, \overset{n}{\underset{i=1}{\oplus}} \mathcal{O}(m))). \tag{2.18}$$

Therefore, we can classify all $[n,k,\delta]$-convolutional codes with a fixed memory as points of the same grassmannian. □

Note that to fix the length n of the codes that we want to classify as subspaces of K^l is equivalent to fix the isomorphism $K^l = K^{n(m+1)} \simeq H^0(\mathbb{P}^1, \overset{n}{\underset{i=1}{\oplus}} \mathcal{O}(m))$, where $m = l/n - 1$.

We want now to determine which points from $Gr(\lambda, H^0(\mathbb{P}^1, \overset{n}{\underset{i=1}{\oplus}} \mathcal{O}(m)))$ correspond to convolutional codes.

Notice first that the points in the image of nontrivial inclusions of the form (2.18) are subspaces of $H^0(\mathbb{P}^1, \overset{n}{\underset{i=1}{\oplus}} \mathcal{O}(n_i)))$ with some $n_i < m$, i. e., they are the subspaces spanned by a generator matrix that if partitioned in n blocks of $m+1$ columns has a 0-column in the rightmost position of the i-th block for some i. In fact, the number of 0-columns in the rightmost positions of each block determines the sequence $\{n_i\}_1^n$. Then to check if a point of the grassmannian determines a convolutional code, it is enough to give conditions on $Gr(\lambda, H^0(\mathbb{P}^1, \overset{n}{\underset{i=1}{\oplus}} \mathcal{O}(n_i)))$.

Let us first check which points are in the image of $\phi_{\{n_i\}_1^n}$.

Recall the definition of a matrix $\widehat{M}$ associated with a matrix M given by Definition 2.22.

Lemma 2.30. *A point of $Gr(\lambda, H^0(\mathbb{P}^1, \overset{n}{\underset{i=1}{\oplus}} \mathcal{O}(n_i)))$ which as a subspace is generated by the rows of a matrix M represents a rank $n-k$ quotient sheaf of $\overset{n}{\underset{i=1}{\oplus}} \mathcal{O}(n_i)$ with $k \leq \lambda$ if and only if the rank of $\widehat{M}$ is $\lambda + k$.*

Proof. Let us call the lower λ rows of $\widehat{M}$ the "shifted" rows corresponding to the upper ones.

The matrix M represents the subspace associated to a rank $n-k$ quotient sheaf if and only if the rows of M can be grouped in k row-blocks such that after a number

of row linear operations M has the shape (2.17). Notice that the rows of a row-block and those from the "shifted" row-block coincide except the first row from the original block and the last one from the "shifted" block. Then, in a nonzero minor of $\widehat{M}$ there can be at most all the rows from a row-block of M and the last row from the shifted block. This would give a nonzero minor of order at most $\lambda + k$.

On the other side, there exists a nonzero $\lambda + k$-minor within the submatrix of $\widehat{M}$ consisting on the rows of M and the last row of each shifted row-block. This is due to the facts that M has maximum rank and that given the sequence $\{n_i\}_1^n$ the partition of the rows of M so that it has the shape (2.17) up to linear row operations is unique. Indeed, every linear dependency of the rows of this submatrix of $\widehat{M}$ involving only rows of M or only shifted rows would mean that M doesn't have maximum rank. Now, every linear dependency involving both rows of M and shifted rows would mean that it is possible to take a linear combination on the rows of M to get one of the shifted rows. Then, by performing this linear operations in M we can substitute a row from M by a shifted row from a different row-block (all the rows from a row-block together with the shifted row of the last one cannot be linearly dependent). In that case, it would be possible in M to add an extra row to the bottom of one of the row-blocks, and the row partition wouldn't be unique. □

Note that for $k < \lambda$ the set of points satisfying the condition on the previous lemma is a closed subset which will be denoted Z_k. For $k = \lambda$, the result is equivalent to Lemma 2.23, and the set of points that it defines is an open subset which will be denoted U_k.

The condition for a point from $Gr(\lambda, H^0(\mathbb{P}^1, \bigoplus_{i=1}^{n} \mathcal{O}(n_i)))$ represented by a matrix M to correspond to a basic polynomial matrix is the one given in Lemma 2.24, which defines an open subset of this grassmannian. On the other side, the condition that this point is not contained in a smaller grassmannian of the form $Gr(\lambda, H^0(\mathbb{P}^1, \bigoplus_{i=1}^{n} \mathcal{O}(n_i')))$ means that the corresponding polynomial matrix is reduced, which according to the characterization in Theorem 1.23 defines an open subset. The intersection of both subsets, i. e., those points associated to canonical polynomial matrices, is an open subset which will be denoted U_C.

Then, we can classify the convolutional codes of length n, dimension k, degree δ and memory m in the following way.

Theorem 2.31. *The convolutional codes of length n, dimension k, memory m and degree $\delta < km$ are represented by an open subset of a closed subset of the grassmannian $Grass(k(m+1) - \delta, H^0(\bigoplus_{i=1}^{n} \mathcal{O}_{\mathbb{P}^1}(m)))$ given by $U_C \cap Z_k$.*

The convolutional codes of type $[n, k, \delta; m]$ which have all their Forney indices equal, i. e. $\delta = km$, are represented by the open subset of the grassmannian $Gr(k, H^0(\mathbb{P}^1, \bigoplus_{i=1}^{n} \mathcal{O}(m)))$ given by $U_C \cap U_k$.

Proof. The construction carried along Theorem 2.27 allows to identify a convolutional code of type $[n, k, \delta]$ and memory m with a point of the grassmannian

$Grass(k(m+1)-\delta, H^0(\overset{n}{\underset{i=1}{\oplus}} \mathcal{O}_{\mathbb{P}^1}(m)))$. However, not all the points of this grassmannian define convolutional codes. For that they have to be in the image by $\phi_{\{n_i\}_1^n}$ of a rank $n-k$ quotient sheaf without torsion in the affine line. Further, they cannot be in the image of any morphism $\phi_{\{n_i'\}_1^n}$ with some $n_i' < n_i$, in order that this representation is unique.

The second condition is equivalent to the fact that the module given by the sections of the sheaf in $\mathbb{A}$ is generated by a basic polynomial matrix, whereas the third condition means that this polynomial matrix is reduced. Then, the second and third conditions mean that the point must lie in the subset U_C. When $\delta < km$, the first condition means that the point must lie in the subset Z_k. As a result, the points representing convolutional codes of type $[n, k, \delta]$ and memory m are those in $U_C \cap Z_k$.

In the case where $\delta = km$, the construction carried along Lemma 2.19 and Theorem 2.21 identifies a $[n, k, \delta]$ convolutional code with a point of $Gr(k, H^0(\mathbb{P}^1, \overset{n}{\underset{i=1}{\oplus}} \mathcal{O}(m)))$. In this case, the set of points of the grassmannian associated with a rank $n-k$ sheaf is the open subset U_k. As seen before, the set of points of the grassmannian associated with a sheaf without torsion in the affine line that represents uniquely a convolutional code is the open subset U_C. Then, only the points in $U_C \cap U_k$ represent $[n, k, \delta]$ convolutional codes. □

Remark 2.32. A different classification of convolutional codes in terms of a quotient sheaf has been given in [RR94]. The main difference is that the basic indices for that classification are the row degrees.

Specifically, instead of the sequence (2.5) this one is considered

$$0 \longrightarrow \overset{k}{\underset{i=1}{\oplus}} \mathcal{O}_{\mathbb{P}^1}(-\nu_i) \longrightarrow \mathcal{O}_{\mathbb{P}^1}^n \longrightarrow Q \to 0$$

where ν_i are the row degrees of the code.

In terms of the encoding process, this means that the message words are considered to have polynomial coordinates with different degrees in order to give encoded words with the same degree in all coordinates.

As a result, the set of convolutional codes of type $[n, k, \delta]$ is identified with a closed subset of the grassmannian $Gr(k(\delta+1)-\delta, K^{n(\delta+1)})$.

The classification presented here seems more natural from the coding point of view in the sense that no constraint is given for the message words, while the difference of degrees in the polynomial components of the codewords will give information about the code. This, in addition, allows to introduce the set of *minimal column indices* as an invariant to describe the code.

On the other side, our classification process makes explicit the difference between the equivalence of $K[z]$-submodules and that of the K-vector subspaces that can be associated with them.

Further, by considering also the *memory* of the code as a classification parameter, it is possible to represent convolutional codes in "smaller" grassmannians.

2.5 Considerations about the Free Distance

Several bounds on the free distance of a convolutional code are known, such as the Heller bound, the Griesmer bound and the generalized Singleton bound. They are generalizations for convolutional codes of bounds on the minimun distance of block codes, such as the Plotkin bound, the Griesmer bound and the Singleton bound respectively [RS99, GLS03].

If we consider a canonical generator matrix with column degrees $\{m_i\}$, such that $\theta = \sum m_i$, we may derive in a similar way to [GLS03, Th 3.4] generalizations of those bounds for the free distance. For that, although in the rest of the chapter we have worked with a general field K, we consider here a finite field $\mathbb{F}_q$, since the size of the field is a parameter in the expressions of some bounds.

Theorem 2.33. *Given a convolutional code $\mathcal{C}$ of type $[n,k]_q$ which has a generator matrix with column degrees $\{m_i\}_{i=1}^n$, with $\sum m_i = \theta$, its free distance d_{free} verifies*

$$d_{free} \leq \min_{i \in \mathbb{N}_0} \left\lfloor \frac{(n(i+1)+\theta)q^{k(i+1)-1}(q-1)}{q^{k(i+1)}-1} \right\rfloor$$

(componentwise Heller bound)

$$d_{free} \leq S(n,k,\theta) := n - k + \theta + 1$$

(componentwise generalized Singleton bound)

$$d_{free} \leq max\{d' \in \{1,\ldots,S(n,k,\theta)\} | \sum_{l=0}^{k(i+1)-1} \left\lceil \frac{d'}{q^l} \right\rceil \leq n(i+1)+\theta \ \forall i \in \mathbb{N}_0\}$$

(componentwise Griesmer bound)

Proof. The strategy will be to upper-bound the free distance of the code with that of the subcodes made up by the encoding of information words of a maximum fixed degree.

Let the code $\mathcal{C}$ be generated by a polynomial matrix G with column degrees $m_1,\ldots,m_n$ and $\sum m_i = \theta$. For each integer $i \geq 0$, let us consider the subspace of messages $U_i = \{(u_1,\ldots,u_k) \in \mathbb{F}_q[z]^k/_{z^{i+1}}\}$, i. e., the polynomial k-vectors of maximum degree i, and let us define $\mathcal{C}_i = \{uG | u \in U_i\} \subset \mathcal{C}$. $\mathcal{C}_i$ is a $\mathbb{F}_q$-vector space with the same dimension as U_i, $dim_{\mathbb{F}_q}\mathcal{C}_i = dim_{\mathbb{F}_q}U_i = k(i+1)$.

On the other side, we have that given a codeword $c = (c_1,\ldots,c_n) \in \mathcal{C}_i$, then $deg(c_j) \leq m_j + i$.

Then, by taking the coefficients up to degree $m_j + i$ in the j-th component of the codewords in $\mathcal{C}_i$ we can identify this convolutional subcode of $\mathcal{C}$ with a block code of length $\sum m_j + i + 1 = n(i+1) + \theta$ and the same dimension as $\mathcal{C}_i$, i. e., $k(i+1)$. The free distance of $\mathcal{C}_i$ is the same as the minimum distance of this block code. By applying the Plotkin bound, the Griesmer bound and the Singleton bound to the parameters of the block code we get the corresponding bounds on $d_{free}(\mathcal{C}_i)$. As $\mathcal{C}_i \subseteq \mathcal{C}$, and therefore $d_{free}(\mathcal{C}) \leq d_{free}(\mathcal{C}_i)\ \forall i \geq 0$, by considering the minimum values of these bounds so that they hold for all $i \geq 0$ we obtain the bounds on the free distance of the whole code. □

These bounds have a rather simpler form than those in [GLS03], and by comparison one can check that they are sharper when both

$$n + \theta < nm \quad k + \delta < km$$

hold.

The classification presented in the previous section can be used also to derive generalizations of distance bounds for convolutional codes.

Theorem 2.34. *Let $\mathcal{C}$ be a convolutional code of type $[n, k, \delta; m]_q$ with minimal column indices (mci) $n_1, \ldots, n_n$, then its free distance d_{free} verifies*

$$d_{free} \leq \min_{i \in \mathbb{N}_0} \left\lfloor \frac{(\sum n_j + n(i+1))q^{k(m+i+1)-\delta-1}(q-1)}{q^{k(m+i+1)-\delta}-1} \right\rfloor$$

(mci Heller bound)

$$d_{free} \leq S(n, k, \{n_i\}) := \sum n_j + n - k(m+1) - \delta + 1$$

(mci generalized Singleton bound)

$$d_{free} \leq max\{d' \in \{1, \ldots, S(n, k, \{n_i\})\} | \sum_{l=0}^{k(m+i+1)-\delta-1} \left\lceil \frac{d'}{q^l} \right\rceil \leq \sum_{j=1}^{n} n_j + n(i+1) \ \forall i \in \mathbb{N}_0\}.$$

(mci Griesmer bound)

Proof. From the classification in the previous section it is known that a $[n, k, \delta]$ convolutional code with memory m is associated with the block code which as a subspace corresponds to the point that represents the convolutional code in $Grass(k(m+1) - \delta, H^0(\bigoplus_{i=1}^{n} \mathcal{O}_{\mathbb{P}^1}(n_i)))$. The minimum distance of the block code upper-bounds the free distance of the convolutional code. This block code has length $n' = \sum n_i + n$ and dimension $k' = k(m+1) - \delta$. Then, as we did in the previous theorem we can bound the minimum distance of block codes with these parameters to get bounds for the minimum distance of the block code and hence, we get bounds for the free distance of the convolutional code:

$$\begin{aligned}
d_{free} &\leq \left\lfloor \frac{(\sum n_i + n)q^{k(m+1)-\delta-1}(q-1)}{q^{k(m+1)-\delta}-1} \right\rfloor, && \text{after applying Plotkin bound} \\
\sum_{l=0}^{k(m+1)-\delta-1} \left\lceil \frac{d_{free}}{q^l} \right\rceil &\leq \sum n_i + n, && \text{after applying Griesmer bound} \\
d_{free} &\leq \sum n_j + n - k(m+1) - \delta + 1, && \text{after applying Singleton bound}
\end{aligned}$$

Notice that the codewords of the $[\sum n_j + n, k(m+1) - \delta]$ block code correspond to the polynomial codewords of the $[n, k]$ convolutional code of maximum degree m. In general, for the polynomial codewords of maximum degree $m + i$, $i \geq 0$, we may consider the tensor product of the morphism (2.15) by $\mathcal{O}(i)$, which in the same way as in Theorem 2.27 results in a correspondence of the convolutional code

$\mathcal{C}$ with a block code of type $[\sum n_j + n(i+1), k(m+i+1) - \delta]$, $i \in \mathbb{N}_0$, the minimum distance of which also upper-bounds the free distance of the convolutional code. Then, by applying the Plotkin bound, the Griesmer bound and the Singleton bound respectively, and by considering the minimum values that hold for all $i \geq 0$, we get the three bounds on the free distance of the convolutional code. □

Corollary 2.35. *A MDS convolutional code of type $[n, k, \delta; m]$ must have minimal column indices $\{n_i\}_1^n$ satisfying $\sum n_i \geq (m-1)n + k$ if $\delta < km$, or $n_i = m$ for all $i \leq n$ if $\delta = km$.*

Proof. For an MDS convolutional code the *mci generalized Singleton bound* from the previous theorem cannot be strictly sharper than the generalized Singleton bound. Bearing this in mind, by comparison of both bounds we get that for a MDS convolutional code it must hold

$$\sum n_i - km \geq (n-k) \left\lfloor \frac{\delta}{k} \right\rfloor$$

from which our result follows. □

In addition, our classification of convolutional codes makes it possible to use in a different way the bounds on the minimum distance of a particular class of block codes in order to bound the free distance of convolutional codes.

Recall that we identify any canonical generator matrix of a convolutional code with a subspace of K^l spanned by a generator matrix that can be written in the form (2.17).

Theorem 2.36. *Let $G(z)$ be a canonical generator matrix of a convolutional code $\mathcal{C}$ with Forney indices $\nu_1, \ldots, \nu_k$, and*

$$G(z) \cdot (1, z^{n_1+1}, z^{n_1+n_2+2}, \ldots, z^{\sum_1^{n-1} n_i + n - 1})^\top = (p_1(z), \ldots, p_k(z))^\top.$$

Let $p_i(z)$ generate a cyclic code of type $[n, k, d] = [deg(p_i) + \nu_k - \nu_i + 1, \nu_k - \nu_i + 1, d_i]$. Then,

$$d_{free}(\mathcal{C}) \leq min\{d_i\}.$$

If $g_{ij}(z)$ is the (i,j)-th entry of $G(z)$ and it generates a cyclic code of type $[n, k, d] = [deg(g_{ij}) + \nu_k - \nu_i + 1, \nu_k - \nu_i + 1, d_{ij}]$, then

$$d_i \geq \sum_j d_{ij}.$$

Proof. Let us consider the generator matrix of the block code associated with $\mathcal{C}$,

$$\begin{pmatrix} \mathrm{g}^{(11)} & \cdots & \mathrm{g}^{(1n)} \\ \vdots & & \vdots \\ \mathrm{g}^{(k1)} & \cdots & \mathrm{g}^{(kn)} \end{pmatrix} \tag{2.19}$$

Note from the explicit expression in (2.17) that every row-block submatrix $(\mathrm{g}^{(i1)}, \ldots, \mathrm{g}^{(in)})$ is in fact the generator matrix, up to the last 0-columns, of a cyclic code. In fact, if $G(z)$ is the generator matrix of $\mathcal{C}$ the entries of which have as coefficients the elements of (2.19), and $g^{(i*)}$ is the i-th row of $G(z)$, a generating polynomial of the cyclic code defined by $(\mathrm{g}^{(i1)}, \ldots, \mathrm{g}^{(in)})$ is

$$p_i(z) = g^{(i*)} \cdot (1, z^{n_1+1}, z^{n_1+n_2+2}, \ldots, z^{\sum^{n-1} n_i+n-1})^\top$$

(which is the *generator polynomial* of the cyclic code if and only if it is monic). As a result, the minimum distance of the cyclic code generated by $p_i(z)$ upper-bounds the minimum distance of the block code generated by (2.19), and hence the free distance of the convolutional code.

On the other side, notice that each of the blocks $\mathrm{g}^{(ij)}$ is in fact the generator matrix (up to the last 0-columns) of a cyclic code generated precisely by the polynomial in the (i, j)-th component of $G(z)$. Then, the minimum distance of the cyclic code generated by $p_i(z)$ can be lower-bounded by the sum of the minimum distances of the cyclic codes generated by the polynomial entries in the i-th row of $G(z)$. □

2.6 Some Optimal Convolutional Codes Obtained from Their Related Block Codes

The representation of convolutional codes as K-vector subspaces led us to investigate the relationship between the minimum distance and the free distance of related codes. This gave as a result some bounds on the free distance of convolutional codes based on well-known bounds on the minimum distance of block codes. Then, the natural question arises wether it is possible to exploit this relationship to derive convolutional codes with an optimal free distance from related block codes which are known to have optimal minimum distance. We present here a number of cases, considering different optimal block codes, in which this is possible.

Hamming Codes

Let us take the *Hamming code* $\mathcal{H}_2(3)$ over $\mathbb{F}_2$ generated by

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

Then, after different partitions we have the convolutional codes,

Partition $\{n_i\}$	{0,1,1,1}	{1,0,1,1}	{1,1,1,0}
$G(z)$	$\begin{pmatrix} 0 & 0 & 1+z & 1+z \\ 0 & 1+z & 0 & 1+z \\ 1 & z & z & z \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 1+z & 1+z \\ z & 1 & 0 & 1+z \\ 1 & 1 & z & z \end{pmatrix}$	$\begin{pmatrix} 0 & z & 1+z & 1 \\ z & 1 & z & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$
$[n, k, \delta, d_{free}]$	[4,3,1,2]	[4,3,1,2]	[4,3,2,4]

The third one is MDS and the other two reach the Griesmer bound for their parameters.

Adding a parity check bit to the previous Hamming code we obtain the extended $[8, 4]$ Hamming code generated by the matrix

$$\begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \sim \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

By considering the partition given by the column indices $\{0, 0, 0, 1, 0, 0, 0\}$ we have the convolutional code generated by

$$\begin{pmatrix} 0 & 0 & 0 & 1+z & 1 & 1 & 0 \\ 1 & 1 & 0 & z & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{pmatrix} \sim \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1+z & 0 & 0 & 0 \end{pmatrix}$$

which has parameters $[n, k, \delta] = [7, 4, 1]$ and $d_{free} = 4$, and hence, reaches the Griesmer bound.

Let us take now the subcode generated by

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

and the sequence of column indices $\{1, 1, 1, 1\}$. Then this code represents the convolutional code generated by

$$\begin{pmatrix} 0 & z & 1+z & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

which has parameters $[n, k, \delta] = [4, 2, 1]$ and $d_{free} = 4$, i. e., it is a MDS convolutional code.

We consider now another *Hamming code*, $\mathcal{H}_3(3)$, over $\mathbb{F}_3$, generated by the matrix

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 & 2 \\ 1 & 0 & 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 \end{pmatrix}.$$

The partition defined by the indices $\{1, 0, 1, 0, 1, 0, 1, 0, 0\}$ results in the convolutional code generated by the matrix

$$\begin{pmatrix} 0 & 0 & z & 1 & 1+z & 1 & 1+z & 1 & 1 \\ z & 1 & 1 & 0 & z & 1 & 1+2z & 2 & 2 \\ 1 & 1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 \end{pmatrix}$$

with parameters $[n, k, \delta] = [9, 3, 2]$ and free distance $d_{free} = 9$. Thus it is a MDS code.

We may consider different partitions leading to convolutional codes with different degrees. However not all of them have optimal free distance. In particular when there is a big difference between the column degrees the value of the free distance stays far apart from the optimal value, as we have seen in the previous section. As a small illustration we present a number of possible partitions and the parameters of the convolutional code obtained, together with the value of the Griesmer bound for those parameters

$\{n_i\}$	(n, k, δ)	d_{free}	Griesmer
{1,0,1,0,1,0,0,0,0,0}	(10,3,2)	9	10
{1,1,1,1,1,1,0}	(7,3,3)	7	9
{2,2,2,1,1}	(5,3,5)	7	9
{2,2,2,2,0}	(5,3,6)	7	9
{3,3,3,0}	(4,3,9)	5	10

Reed-Solomon Codes

Let us consider now the *Reed-Solomon code* over a finite field $\mathbb{F}_q$ ($q \geq 7$) generated by

$$\begin{pmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 & \alpha_5 & \alpha_6 \\ \alpha_1^2 & \alpha_2^2 & \alpha_3^2 & \alpha_4^2 & \alpha_5^2 & \alpha_6^2 \end{pmatrix}.$$

Let us take the partition given by the column indices $\{n_i\}_i = \{1, 1, 1\}$. Then we get the convolutional code generated by the matrix

$$\begin{pmatrix} \alpha_1 z + \alpha_2 & \alpha_3 z + \alpha_4 & \alpha_5 z + \alpha_6 \\ \alpha_1^2 z + \alpha_2^2 & \alpha_3^2 z + \alpha_4^2 & \alpha_5^2 z + \alpha_6^2 \end{pmatrix}$$

which having parameters $[3, 2, 2]_q$ and $d_{free} = 5$ is also MDS.

Other Optimal Codes

Let us consider the block code over $\mathbb{F}_4 = \mathbb{F}_2[x]/_{x^2+x+1} \simeq F_2(\alpha)$ generated by

$$\begin{pmatrix} \alpha & \alpha^2 & 1 & \alpha & \alpha^2 & 1 \\ \alpha^2 & \alpha & 1 & \alpha^2 & \alpha & 1 \end{pmatrix}.$$

This code has minimum distance $d = 4$. Let us consider the partition given by $\{n_i\}_i = \{1, 1, 1\}$. The corresponding convolutional code is generated by the matrix

$$\begin{pmatrix} \alpha z + \alpha^2 & z + \alpha & \alpha^2 z + 1 \\ \alpha^2 z + \alpha & z + \alpha^2 & \alpha z + 1 \end{pmatrix}.$$

This code has parameters $[3, 2, 2]_4$ and free distance $d_{free} = 4$, and as a result it reaches the Griesmer bound for those parameters.

The variety of block codes that result in optimal convolutional codes leads us to consider this a quite valuable method to derive new families of convolutional codes. However, as we have seen, the precise conditions under which the resulting code has optimal free distance are still not clear and we don't have much information except that given by the conditions on the column distances derived in the previous section. This remains therefore as an appealing topic for future research.

Chapter 3

Convolutional Goppa Codes Associated with Elliptic Curves

3.1 Goppa Codes

The origin of algebraic geometric codes are Goppa codes, presented by V.D. Goppa in the late seventies [Gop77, Gop81]. The construction of these codes was the seed of a fruitful link between coding theory and algebraic geometry which resulted in many other code constructions but also in the study of related open questions, which led to breakthrough results. One of these questions is concerned with the number of rational points in a curve, in particular when the base field is finite. Tsfasman, Vladut and Zink proved the existence of curves with many rational points. As a result, it was also proven the existence of families of codes which beat the Gilbert-Varshamov bound.

For the new codes also decoding algorithms had to be devised which, by taking advantage of the algebraic properties of their construction, could provide them with practical usability. The first ones looked for an error locator polynomial, i. e., a polynomial with zeros at the points corresponding with the error positions, and made use of one or several divisors related to those defining the particular code. Afterwards another scheme, known as *majority vote for unknown syndromes*, appeared, and even the Berlekamp-Massey algorithm was adapted. For a review on decoding of algebraic geometric codes [HP95] is recommended.

3.1.1 Geometric Construction of Goppa Codes

Before proceeding with the construction of Goppa codes we briefly review the geometric elements in which it is based.

Let X be a projective curve of genus g over a finite field $\mathbb{F}_q$.[1] Let us denote $\mathbb{F}_q(X)$ its field of rational functions.

[1]The constructions presented in this chapter are carried out over finite fields. Hence K will not denote a general field as in Chapter 2 but, following the usual notation in algebraic geometry, the canonical divisor of a curve.

A *rational point* of X is a point with coordinates in the base field $\mathbb{F}_q$. The degree of a point is defined by $deg_{\mathbb{F}_q}(P) = n$ if $\mathbb{F}_{q^n}$ is the smallest extension of $\mathbb{F}_q$ in which P is rational. For every point $P \in X$ there is a *valuation* v_P such that, for every $f \in \mathbb{F}_q(X)$, $v_P(f)$ is the order of the zero or minus the order of the pole of f at P.

A *divisor* is a formal sum $\sum_{P \in X} n_P P$ such that all coefficients $n_P \in \mathbb{Z}$ and $n_P \neq 0$ only for finitely many of them. The sum of divisors is defined coefficientwise and they have a partial order given by comparison of their coefficients. The *support* of a divisor is the set or points P such that $n_P \neq 0$. A divisor is an *effective divisor* if all its coefficients are nonnegative. The *degree* of a divisor $\sum n_P P$ is $\sum n_P \deg(P)$. Every rational function f defines a divisor $(f) = \sum v_P(f)P$, called *principal divisor* of the function. The degree of every principal divisor is 0. There is an equivalence relationship of divisors defined as $G \sim H$ if and only if $G - H$ is a principal divisor. For every divisor G there is an invertible sheaf $\mathcal{O}_X(G)$ such that for every open subset U

$$\mathcal{O}_X(G)(U) = \{f \in \mathbb{F}_q(X) | D_{|U} + (f)_{|U} \geq 0\},$$

and $\mathcal{O}_X(G) \otimes \mathcal{O}_X(H) = \mathcal{O}_X(G+H)$. In particular, taking global sections, G defines the finite dimensional vector space

$$L(G) = H^0(X, \mathcal{O}_X(G)) = \{f \in \mathbb{F}_q(X) \text{ such that } (f) + G \geq 0\}$$

with dimension, denoted $l(G)$ or $h^0(\mathcal{O}_X(G))$, nonzero if and only if $\deg G \geq 0$. The dimension of its cohomology group $H^1(X, \mathcal{O}_X(G))$, denoted $h^1(\mathcal{O}_X(G))$, is nonzero if and only if $\deg G \leq 2g - 2$.

These elements are used to define the family of Goppa codes in the following way. Let X be a geometrically irreducible, nonsingular projective curve of genus g over a finite field $\mathbb{F}_q$. Let us consider n different rational points of X, $P_1, \ldots, P_n$, and D the divisor $D = P_1 + \cdots + P_n$. D is an effective divisor.

Then we have the exact sequence of sheaves

$$0 \to \mathcal{O}_X(-D) \to \mathcal{O}_X \to Q \to 0 \tag{3.1}$$

where Q is a sheaf with support only at the points P_i, i. e., $Q_P = 0$ for any other point $P \in X$ different from the points P_i. For any open subset U containing all points P_i, $Q(U) \simeq \mathcal{O}_{P_1}/\mathfrak{m}_{P_1} \times \cdots \times \mathcal{O}_{P_n}/\mathfrak{m}_{P_n}$, where $\mathcal{O}_{P_i}$ is the local ring at the point P_i with maximal ideal $\mathfrak{m}_{P_i}$. As the points are rational their residue field, $\mathcal{O}_{P_i}/\mathfrak{m}_{P_i}$, is $\mathbb{F}_q$ and $Q(U) \simeq \mathbb{F}_q \times \overset{n}{\cdots} \times \mathbb{F}_q$.

Recall that for every point P_i, if t_i is a local parameter at P_i, i. e., t_i has a single zero at P_i, there are exact sequences

$$\begin{array}{c} 0 \to \mathfrak{m}_{P_i} \to \mathcal{O}_{P_i} \to \mathcal{O}_{P_i}/\mathfrak{m}_{P_i} \simeq \mathbb{F}_q \to 0 \\ s(t_i) \mapsto s(P_i) \end{array} . \tag{3.2}$$

We consider another divisor, G, with a disjoint support from D. By tensoring (3.1) by $\mathcal{O}_X(G)$ we get

$$0 \to \mathcal{O}_X(G - D) \to \mathcal{O}_X(G) \to Q \to 0.$$

We take global sections on the previous sequence and we get an exact sequence of cohomology

$$\begin{aligned} 0 \to H^0(X, \mathcal{O}_X(G-D)) \to H^0(X, \mathcal{O}_X(G)) \xrightarrow{\alpha} \mathbb{F}_q^n \to \\ \to H^1(X, \mathcal{O}_X(G-D)) \to H^1(X, \mathcal{O}_X(G)) \to 0 \end{aligned} . \tag{3.3}$$

If we impose $\deg(G) < n = \deg(D)$, we have

$$0 \to H^0(X, \mathcal{O}_X(G)) \xrightarrow{\alpha} \mathbb{F}_q^n \to H^1(X, \mathcal{O}_X(G-D)) \to \dots$$

and considering (3.2) we have the injective evaluation map

$$\begin{aligned} \alpha : L(G) &\longrightarrow \mathbb{F}_q^n \\ f &\longmapsto (f(P_1), \dots, f(P_n)) \end{aligned} .$$

Definition 3.1. The *Goppa code* $\mathcal{C}(D, G)$ defined by D and G is the image of α.

This construction allows to use a number algebraic geometric tools, like the Riemann-Roch theorem.

Theorem 3.2 (Riemann-Roch)**.** *Let D be a divisor on a curve X of genus g, then*

$$l(D) - h^1(\mathcal{O}_X(D)) = deg(D) + 1 - g.$$

□

A direct consequence of Riemann-Roch theorem is the following result.

Proposition 3.3. *The Goppa code $\mathcal{C}(D, G)$ defined over a curve of genus g has parameters $[n, k, d]$ with*

$$k \geq \deg(G) + 1 - g, \quad d \geq n - \deg(G).$$

Moreover, if $\deg(G) > 2g - 2$ the dimension is exactly $k = \deg(G) + 1 - g$.

Proof. The dimension of the code is precisely $l(G)$ and by the Riemann-Roch theorem

$$k = l(G) = h^1(\mathcal{O}_X(G)) + deg(G) - g + 1.$$

Further, if $\deg(G) > 2g - 2$, then $h^1(\mathcal{O}_X(G)) = 0$.

On the other side, let $f \in L(G)$ be such that $\alpha(f)$ is a codeword with minimum weight d. Then, there are $n - d$ points from the support of D, $P_{i_1}, \dots, P_{i_{n-d}}$ such that $f(P_{i_j}) = 0 \ \forall j$. As the supports of D and G are disjoint, this means that $f \in L(G - P_{i_1} - \dots - P_{i_{n-d}})$ and then it must be $deg(G) - n + d \geq 0$. □

From the inequalities in Proposition 3.3 it can be obtained

$$d \geq n - k - g + 1.$$

When compared with the Singleton bound, this means that in particular for $g = 0$ every Goppa code $\mathcal{C}(D, G)$ is MDS.

3.1.2 The Dual Construction

The dual codes are defined by means of the residues of certain differentials. A differential over a curve X is an expression of the form fdg where $f, g \in \mathbb{F}_q(X)$ and $d : \mathbb{F}_q(X) \longrightarrow \mathbb{F}_q(X)$ is a derivation, i. e., d satisfies the Leibniz rule, $d(uv) = udv + vdu$. The space of differentials over X is denoted Ω_X. A differential ω has a zero, respectively a pole, of degree n_P at the point P if for $\omega = fdu$ being u a local parameter at P, f has a zero respectively a pole of degree n_P at P. As rational functions, every differential ω defines a divisor $(\omega) = \sum v_P(\omega)P$, called *canonical divisor* of the differential. Every canonical divisor has degree $2g-2$ and all of them are equivalent. Generically, every divisor of the equivalence class is called *canonical divisor*, and denoted as K.

A divisor G defines also a vector space of differentials by

$$\Omega(G) = \{\omega \in \Omega_X | (\omega) - G \geq 0 \text{ or } \omega = 0\}.$$

The dimension of $\Omega(G)$, denoted $i(G)$, is called the *index of speciality* of G, and in particular $i(0) = g$, the genus of X.

By Serre duality ([Har77, III,7.7]) there is an invertible sheaf ω_X such that for every invertible sheaf $\mathcal{L}$ on X there is a canonical isomorphism of $\mathbb{F}_q$-vector spaces $H^1(X, \mathcal{L})^* \simeq H^0(X, \omega_X \otimes \mathcal{L}^{-1})$. ω_X is called the *dualizing sheaf* of X and $H^0(X, \omega_X) = \Omega(0)$, the global regular differentials over X.

Then, given divisors D, G one has

$$\begin{aligned} H^1(X, \mathcal{O}_X(D))^* &\simeq H^0(X, \omega_X \otimes \mathcal{O}_X(-D)) \simeq \Omega(D) \\ H^1(X, \mathcal{O}_X(G-D))^* &\simeq \Omega(G-D) \end{aligned} \quad .$$

Let us consider now the exact sequence (3.3) assuming $deg(G) > 2g-2$, then we have

$$\ldots \to H^0(X, \mathcal{O}_X(G)) \xrightarrow{\alpha} \mathbb{F}_q^n \to H^1(X, \mathcal{O}_X(G-D)) \to 0.$$

By taking duals we get

$$0 \to H^1(X, \mathcal{O}_X(G-D))^* \xrightarrow{\beta} (\mathbb{F}_q^n)^* \xrightarrow{\alpha^t} H^0(X, \mathcal{O}_X(G))^* \to \ldots \tag{3.4}$$

with β the injective map

$$\begin{aligned} \beta : \Omega(G-D) &\longrightarrow \mathbb{F}_q^n \\ \omega &\longmapsto (\mathrm{res}_{P_1}(\omega), \ldots, \mathrm{res}_{P_n}(\omega)) \end{aligned}$$

Definition 3.4. The *dual Goppa code* $\mathcal{C}^*(D, G)$ defined by D and G is the image of β.

Proposition 3.5. *The code $\mathcal{C}^*(D, G)$ has parameters $[n, k, d]$ with*

$$k \geq n - \deg(G) - 1 + g, \quad d \geq \deg(G) - 2g + 2 \, .$$

Moreover, if $\deg(G) < n$ then $k = n - \deg(G) - 1 + g$.

Proof. These statements are a direct consequence of Riemann-Roch theorem and Serre duality [HP95]. □

Remark 3.6. Though the use of term *dual* in definition 3.4 may be considered to be based in the construction via duality, by the residues theorem it can be proven that this code is in fact the dual code (in the coding sense) of the one given by Definition 3.1 [HvLP98]. Alternatively, this duality can be checked in the geometric construction presented before, as exactness of the sequence (3.4) means that $\alpha^t \circ \beta = 0$. In addition, both constructions define the same family of codes, i. e., the code given by Definition 3.1 can be defined in terms of Definition 3.4 by choosing suitable divisors: for every set of n rational points $P_1, \ldots, P_n$ there exists a differential form ω with simple poles at the points $\{P_i\}_i$ and $\text{res}_{P_i}(\omega) = 1 \ \forall \ i$. Then $\mathcal{C}(D, G) = \mathcal{C}^*(D, (\omega) + D - G)$.

Example 3.7 (Generalized Reed-Solomon codes)**.** Let $\alpha, x \in \mathbb{F}_q^n$ where all the components of $\alpha = (\alpha_1, \ldots, \alpha_n)$ are different and $x = (x_1, \ldots, x_n)$ has nonzero components. The *Generalized Reed-Solomon code* defined by α, x is

$$GRS(\alpha, x) = \{(p(\alpha_1)x_1, \ldots, p(\alpha_n)x_n) | p(z) \in \mathbb{F}_q[z], \deg(p) < k\}.$$

Classical Reed-Solomon codes are those with $x = (1, \ldots, 1)$.

Let $f(z) \in \mathbb{F}_q[z]$ with $f(\alpha_i) = x_i \ \forall \ i$ and let us consider curve $X = \mathbb{P}^1$, the projective line. Let us take the points $P_i = (\alpha_i; 1)$, $i \leq n$, $P_\infty = (1; 0)$ and the divisors $D = P_1 + \ldots + P_n$ and $G = (k-1)P_\infty - (f)$. Then $GRS(\alpha, x) = \mathcal{C}(D, G)$. In particular, Reed-Solomon codes are a subfamily of Goppa codes.

Example 3.8 (Classical Goppa codes)**.** Let $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{F}_q^n$ with different components and $p(z) \in \mathbb{F}_{q^m}[z]$ such that $p(\alpha_i) \neq 0$ for all i. The *classical Goppa code* defined by $\alpha, p(z)$ is

$$\Gamma(\alpha, p(z)) = \{c \in \mathbb{F}_q^n | \sum \frac{c_i}{z - \alpha_i} \equiv 0 (\text{mod } p(z))\}.$$

Let us consider now $X = \mathbb{P}^1$, let P_i, P_∞, D be as in Example 3.7. Let us take the divisor of $p(z)$, $(p) = (p)_0 - (p)_\infty$, with both $(p)_0, (p)_\infty$ effective divisors, and $G = (p)_0 - P_\infty$. Then,

$$c \in \Gamma(\alpha, p(z)) \Leftrightarrow \sum \frac{c_i}{z - \alpha_i} dz \in \Omega(G - D)$$

that is to say, $\Gamma(\alpha, p(z)) = \mathcal{C}^*(D, G)$.

3.2 Convolutional Goppa Codes

As convolutional codes are a generalization of linear block codes, one might wonder wether algebraic geometric tools, and in particular similar tools as the ones used by

Goppa, could be also used to construct families of convolutional codes with certain good properties.

A first attempt to define convolutional Goppa codes was made in [MDS04], where instead of a curve, a family of curves parameterized by the affine line was considered. Instead of points, disjoint sections of the projection of this family over the affine line were taken, and instead of divisors on a curve, a Cartier divisor and an invertible sheaf. A construction analogous to the classical one led to a family of convolutional codes "of Goppa type".

After that, a more general construction with simpler geometric tools has been given in [MDIS06]. We will briefly present this construction. Note however that in order to achieve simplicity the resulting objects defined as convolutional codes are $\mathbb{F}_q(z)$-vector spaces. For the construction of convolutional Goppa codes as submodules we refer to [MDS04].

3.2.1 General Construction

Let $\mathbb{F}_q$ be a finite field and $\mathbb{F}_q(z)$ the field of rational functions on one variable. Let X be a smooth projective curve over $\mathbb{F}_q(z)$ of genus g and let us assume that $\mathbb{F}_q(z)$ is algebraically closed in the field of rational functions of X. Both Riemann-Roch and the Residues theorems still hold under this hypothesis [Har77].

Let us take n different $\mathbb{F}_q(z)$-rational points $P_1, \ldots, P_n$ and the divisor $D = P_1 + \cdots + P_n$, with its associated invertible sheaf $\mathcal{O}_X(D)$. We have then the exact sequence of sheaves (3.1),

$$0 \to \mathcal{O}_X(-D) \to \mathcal{O}_X \to Q \to 0\,,$$

where Q is a sheaf with support only at the points P_i.

Let G be another divisor on X with support disjoint from D. By tensoring the exact sequence (3.1) by the associated invertible sheaf $\mathcal{O}_X(G)$, we have

$$0 \to \mathcal{O}_X(G-D) \to \mathcal{O}_X(G) \to Q \to 0\,.$$

and by taking global sections we get the sequence

$$\begin{aligned} 0 \to H^0(X, \mathcal{O}_X(G-D)) \to H^0(X, \mathcal{O}_X(G)) \xrightarrow{\alpha} H^0(X, Q) \to \\ \to H^1(X, \mathcal{O}_X(G-D)) \to H^1(X, \mathcal{O}_X(G)) \to 0 \end{aligned}\,.$$

If we impose $\deg(G) < n = \deg(D)$, we have an injective $\mathbb{F}_q(z)$-linear map

$$\begin{aligned} 0 \longrightarrow L(G) &\xrightarrow{\alpha} \mathbb{F}_q(z) \times \overset{n}{\cdots} \times \mathbb{F}_q(z) \longrightarrow \cdots \\ s &\longmapsto (s(P_1), \ldots, s(P_n)) \end{aligned}$$

Definition 3.9. The *convolutional Goppa code* $\mathcal{C}(D, G)$ defined by the divisors D and G is the image of $\alpha\colon L(G) \to \mathbb{F}_q(z)^n$. Given a subspace $S \subseteq L(G)$, the convolutional Goppa code $\mathcal{C}(D, S)$ defined by D and S is the image of $\alpha_{|S}$.

We can use the Riemann-Roch theorem to calculate the dimension of a convolutional Goppa code.

Proposition 3.10 ([MDIS06])**.** *$\mathcal{C}(D,G)$ is a convolutional code of length $n=deg(D)$ and dimension $k \geq deg(G)+1-g$. If $deg(G) > 2g-2$ then $k = deg(G)+1-g$.*

The geometric tools to characterize the free distance of convolutional Goppa codes are much more sophisticated than the analogous ones in the block case, involving jets, osculating planes and an interpretation of the points P_i as sections over the affine line. Then, the calculus of the free distance could be interpreted as a problem of Enumerative Geometry over finite fields [MDIS06].

3.2.2 Dual Convolutional Goppa Codes

Similarly to the block case, we can define the dual code of $\mathcal{C}(D,G)$. We will develop here a different but equivalent construction for it.

Let us consider $\mathcal{C}(D,G)$, the convolutional Goppa code defined by the divisors $D = P_1 + \cdots + P_n$ and G over the curve X, and let K be the canonical divisor. By tensoring the sequence (3.1) by the invertible sheaf $\mathcal{O}_X(K+D-G)$, we have

$$0 \to \mathcal{O}_X(K-G) \to \mathcal{O}_X(K+D-G) \to Q \to 0\,. \tag{3.5}$$

If we take global sections, we get

$$\begin{aligned} 0 \to H^0(X,\mathcal{O}_X(K-G)) \to H^0(X,\mathcal{O}_X(K+D-G)) \xrightarrow{\beta} H^0(X,Q) \to \\ \to H^1(X,\mathcal{O}_X(K-G)) \to H^1(X,\mathcal{O}_X(K+D-G)) \to 0\,. \end{aligned}$$

Then, by imposing $deg(G) > 2g-2$ we obtain the exact sequence

$$0 \to H^0(X,\mathcal{O}_X(K+D-G)) \xrightarrow{\beta} \mathbb{F}_q(z)^n \to H^1(X,\mathcal{O}_X(K-G)) \to \dots \tag{3.6}$$

which via Serre duality (taking $\mathbb{F}_q(z)$ as the ground field instead of $\mathbb{F}_q$) coincides with (3.4). Then we have an injective $\mathbb{F}_q(z)$-linear map

$$\begin{array}{ccccccc} 0 & \longrightarrow & \Omega(G-D) & \xrightarrow{\beta} & \mathbb{F}_q(z) \times \overset{n}{\dots} \times \mathbb{F}_q(z) & \longrightarrow & \cdots \\ & & \eta & \longmapsto & (\mathrm{res}_{P_1}(\eta), \dots, \mathrm{res}_{P_n}(\eta)) & & \end{array}$$

Definition 3.11. The *dual convolutional Goppa code* $\mathcal{C}^*(D,G)$ defined by the divisors D and G is the image of β.

As mentioned before, the residues theorem holds in our setting and hence it can be used to check that in fact $\mathcal{C}^*(D,G)$ is the dual convolutional code of $\mathcal{C}(D,G)$.

Proposition 3.12 ([MDIS06])**.** *The code $\mathcal{C}^*(D,G)$ is a convolutional code of length $n = deg(D)$ and dimension $k \geq n - deg(G) - 1 + g$. If $deg(G) < n$ then $k = n - deg(G) - 1 + g$.*

3.2.3 Convolutional Goppa Codes over the Projective Line

As an instance of a family of Goppa convolutional codes, the following construction is presented in [MDIS06].

Let $X = \mathbb{P}^1_{\mathbb{F}_q(z)}$ be the projective line over $\mathbb{F}_q(z)$, and let t be the affine coordinate. Let $P_0 = (1;0)$ be the point at the origin and $P_\infty = (0;1)$ the point at infinity. Consider $P_1, \ldots, P_n$ n different rational points of $\mathbb{P}^1$, $P_i \neq P_0, P_\infty$, with $t(P_i) = x_i$ ($P_i = (1;x_i)$). Let us define the divisors $D = P_1 + \cdots + P_n$ and $G = rP_\infty - sP_0$, with $0 \leq s \leq r < n$. Then a basis of $L(G)$ is given by $\{t^s, t^{s+1}, \ldots, t^r\}$.

Since $g = 0$, the evaluation map

$$\begin{aligned} \alpha : L(G) &\longrightarrow \mathbb{F}_q(z)^n \\ t^i &\longmapsto (x_1^i, \ldots, x_n^i) \end{aligned}$$

is injective, and $\mathcal{I}\mathrm{m}\,\alpha$ defines a convolutional Goppa code $\mathcal{C}(D,G)$ of length n and dimension $k = r - s + 1$.

Example 3.13. Let us consider the field $\mathbb{F}_5(z)$ and the points of the projective line $\mathbb{P}^1_{F_5(z)}$

$$P_1 = (1; z+1), P_2 = (1; 2z+3), P_3 = (1; 4z+4), P_4 = (1; 3z+2).$$

Consider the divisors $D = P_1 + P_2 + P_3 + P_4$ and $G = 2P_\infty - P_0$, then $L(G) = \langle t, t^2 \rangle$ and the convolutional Goppa code $\mathcal{C}(D,G)$ is generated by the matrix

$$G = \begin{pmatrix} z+1 & 2z+3 & 4z+4 & 3z+2 \\ (z+1)^2 & (2z+3)^2 & (4z+4)^2 & (3z+2)^2 \end{pmatrix}.$$

$\mathcal{C}(D,G)$ has parameters $[n, k, \delta, d_{free}] = [4, 2, 3, 8]$.

Its dual, $\mathcal{C}^*(D,G)$, obtained by taking the residues of the rational differential forms in $\Omega(G - D)$ is generated by the matrix

$$H = \begin{pmatrix} 4(z+4) & 3(z+1) & (z+4) & 2(z+1) \\ (z+4)^2 & (z+1)^2 & (z+4)^2 & (z+1)^2 \end{pmatrix}$$

and it also has parameters $[n, k, \delta, d_{free}] = [4, 2, 3, 8]$.

Both $\mathcal{C}(D,G)$ and $\mathcal{C}^*(D,G)$ are MDS convolutional codes.

3.3 Convolutional Goppa Codes over Elliptic Curves

Let $X \subset \mathbb{P}^2_{\mathbb{F}_q(z)}$ be a plane elliptic curve over $\mathbb{F}_q(z)$. Without loss of generality we will assume that X has a rational point of order at least 4 (so that there are enough rational points to define a convolutional code). Then, in an affine plane containing this point, X can be written in Tate Normal form [Hus87]

$$y^2 + axy + by = x^3 + bx^2 \tag{3.7}$$

being x, y the affine coordinates in this plane and $a, b \in \mathbb{F}_q(z)$. Let P_∞ be the point at infinity, $P_0 = (0,0)$ and $P_1, \ldots, P_n$ n different rational points of X, with $P_i = (x_i, y_i)$ and $x_i, y_i \in \mathbb{F}_q(z)$. Consider the divisors $D = P_1 + \cdots + P_n$ and $G = rP_\infty$, $r < n$.

Recall that the divisors of the functions x, y are

$$(x) = P_0 + Q - 2P_\infty \, , \, (y) = 2P_0 + Q' - 3P_\infty$$

where Q, Q' are two rational points different from P_0, P_∞.

Then, a basis of $L(G)$ is given by $\{1, x, y, \ldots, x^i y^j, \ldots\}$, with $2i + 3j \leq r$ (and to avoid linear dependencies $j = 0, 1$).

Since $r < n$ the evaluation map

$$\begin{aligned} \alpha : L(G) &\longrightarrow \mathbb{F}_q(z)^n \\ x^i y^j &\longmapsto (x_1^i y_1^j, \ldots, x_n^i y_n^j) \end{aligned}$$

is an injective morphism and $Im\alpha$ defines the convolutional Goppa code $\mathcal{C}(D, G)$ with length n. As $g = 1$ and $deg(G) > 2g-2$ the code has dimension $k = r = deg(G)$.

Let us consider now the case with $G = rP_\infty - sP_0$, where $0 < r - s < n$. A basis of $L(G)$ is $\{x^a y^b, \ldots, x^c y^d\}$ with $a + 2b = s$, $2c + 3d = r$ (and $b, d = 0, 1$). The code $\mathcal{C}(D, G)$ has length n and dimension $r - s$.

The corresponding generator matrix of the code $\mathcal{C}(D, G)$ is

$$G = \begin{pmatrix} x_1^a y_1^b & x_2^a y_2^b & \ldots & x_n^a y_n^b \\ x_1^{a+1} y_1^b & x_2^{a+1} y_2^b & \ldots & x_n^{a+1} y_n^b \\ \vdots & \vdots & \ddots & \vdots \\ x_1^c y_1^d & x_2^c y_2^d & \ldots & x_n^c y_n^d \end{pmatrix}.$$

Example 3.14. Let us consider the elliptic curve with Tate Normal form

$$y^2 + zxy + y = x^3 + x^2$$

over a field $F_2(z)$.

We consider the divisor $D = P_1 + P_2 + P_3 + P_4$ with

$$\begin{array}{cc} P_1 = (1 + z, z) & P_2 = (1 + z, 1 + z^2) \\ P_3 = (\frac{1+z^3}{z^2}, \frac{1+z^3+z^4+z^5}{z^3}) & P_4 = (\frac{1+z^3}{z^2}, \frac{1+z^2+z^4}{z^3}) \end{array}$$

and the divisor $G = 3P_\infty - P_0$, and $L(G) = \langle\, x,\, y \rangle$. Then the convolutional Goppa Code defined by D and G is generated by the matrix

$$\begin{pmatrix} z^2 & z^2 & 1 + z + z^2 & 1 + z + z^2 \\ 1 + z & 1 + z^2 + z^3 & 1 + z + z^3 & 0 \end{pmatrix}$$

$\mathcal{C}(D, G)$ has parameters $[n, k, \delta, m, d_{free}] = [4, 2, 5, 3, 8]$. The free distance of the code attains the Griesmer bound.

Example 3.15. We consider now the elliptic curve

$$y^2 + zxy + 2z^2y = x^3 + 2z^2x^2$$

over $\mathbb{F}_5(z)$, and the divisor $D = P_1 + P_2$, with support at the points

$$P_1 = (3z^2, 3z^2 + 2z^3) \quad P_2 = (\tfrac{2z^2+3z^3+4z^4}{1+3z+z^2}, \tfrac{2z^4+2z^5+3z^6}{4+3z+2z^2+z^3})$$

We take the divisor $G = 2P_\infty - P_0$, and we have $L(G) = \langle x \rangle$.

A canonical generator matrix of the code $\mathcal{C}(D, G)$ is

$$\left(2 + z + 2z^2, 3 + 2z + z^2\right) .$$

The code has parameters $[n, k, \delta, d_{free}] = [2, 1, 2, 6]$, and hence it is MDS.

Example 3.16. Now we take the curve

$$y^2 + (1 + z + z^2)xy + (z^2 + z^3)y = x^3 + (z^2 + z^3)x^2$$

over $\mathbb{F}_q(z)$, with $q \neq 2^m$. We consider the divisor $D = P_1 + P_2 + P_3$ where

$$\begin{aligned} P_1 &= (0, -z^3 - z^2) \\ P_2 &= (z^2 - z, -z^4 - 2z^3 + z^2) \\ P_3 &= (-z^2 - z, -z^3 + z) \end{aligned}$$

Let us take $G = 2P_\infty$, then $L(G) = \langle 1, x \rangle$ and a generator matrix for the convolutional Goppa code $\mathcal{C}(D, G)$ is

$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 - z & 1 + z \end{pmatrix}.$$

$\mathcal{C}(D, G)$, has parameters $[n, k, \delta, m, d_{free}] = [3, 2, 1, 1, 3]$. Then $\mathcal{C}(D, G)$ is an MDS convolutional code.

3.4 Some Optimal Convolutional Goppa Codes over Elliptic Curves

The previous examples illustrate how convolutional Goppa codes of different parameters are constructed from their defining elements, i. e., the elliptic curve and the divisors D and G. They also show the existence of optimal convolutional codes of this class.

In this section we present a wide collection of convolutional Goppa codes defined over elliptic curves. The description of these codes is done by means of their generating elements, and they are ordered in different tables according to their parameters. To characterize the elliptic curve over which each code is defined, let a, b be the parameters on its Tate Normal form (3.7). On this curve we take the points $\{P_i\}_{i=1}^{7} = \{2P, 3P, -3P, 4P, -4P, 5P, -5P\}$, where $P = (0,0) = P_0$ is a rational point which belongs to the curve, and nP, $n \in \mathbb{Z}$ is obtained by the addition law defined over every elliptic curve. Together with them, the basis or the bases (when in the same curve and with the same divisor D there is more than one code for those parameters) of the space of functions that define each code are provided.

We want to point out that although plenty of codes with degree 0, i. e., block codes, can be obtained with this construction only those with $\delta \geq 1$ are presented.

3.4.1 Codes with Dimension 1

The following tables show different $[n, 1, \delta]_p$ convolutional codes with optimal free distance, meaning that it reaches either the Griesmer bound or as in most cases the generalized Singleton bound. Each is defined over the elliptic curve with equation in Tate Normal form determined by the parameters (a, b). In some cases one of these parameters is a function of other parameters. Then, the values that each of these parameters take are also presented at the bottom of the table. Note that for $n \geq 3$ any puncturing gives as a result another code with optimal distance.

$[n, k, \delta]$=[2,1,1] d_{free}=4

p	(a,b)	$L(G)$	i,$D = \sum P_i$
$p \geq 3$	$(z, \alpha_1 z - \alpha_1 z^2)$	$\{x\}$	1, 4
3	$(\alpha_3 z, \alpha_3 z + 2z^2)$	$\{x\}$	1, 4
3	$(\alpha_3 z, 1 + \alpha_3 z + z^2)$	$\{x\}$	1, 4
5	$(2z, 1 + z^2)$	$\{x\}$	1, 4
5	$(2z, 3 + 3z^2)$	$\{x\}$	1, 4
5	$(z, 1 + z + 3z^2)$	$\{x\}$	1, 4
7	$(2z, 4 + 5z^2)$	$\{x\}$	1, 4
7	$(2z, 5 + z^2)$	$\{x\}$	1, 4
7	$(z, 1 + z + 5z^2)$	$\{x\}$	1, 4
11	$(2z, 4 + 6z^2)$	$\{x\}$	1, 4
11	$(z, 1 + 4z + 6z^2)$	$\{x\}$	1, 4
11	$(z, 1 + 5z + 5z^2)$	$\{x\}$	1, 4
13	$(2z, 5 + 6z^2)$	$\{x\}$	1, 4
13	$(z, 1 + 6z + 6z^2)$	$\{x\}$	1, 4
13	$(2z, 4 + 2z + 6z^2)$	$\{x\}$	1, 4

$\alpha_1 \neq -1^*$, $\alpha_i \neq 0$

* Para $\alpha_1 = -1$ se tiene $\delta = 0$.

$[n,k,\delta]$=[2,1,2] d_{free}=5

p	(a,b)	$L(G)$	i,$D=\sum P_i$
2	(z, z^2)	$\{x\}$	1, 4
2	$(1+z, 1)$	$\{x\}$	2, 6
2	$(1+z, 1)$	$\{y\}$, $\{xy\}$	4, 5
			d_{free}=6
3	$(1+z, 1)$	$\{y\}$,$\{xy\}$	4, 5
3	$(1+z, 1)$	$\{x\}$	2, 6
3	$(1+z, 2z+z^2)$	$\{x\}$	2, 6
5	$(0, \alpha_5+\beta z)$	$\{y\}$, $\{xy\}$	4, 5
5	$(0, \alpha_5+\beta z)$	$\{x\}$	2, 6
5	$(z, 1+2z+2z^2)$	$\{x^2\}$	2, 6
7	$(0, \alpha_7+\beta z)$	$\{y\}$, $\{xy\}$	4, 5
7	$(0, \alpha_7+\beta z)$	$\{x\}$	2, 6
7	$(z, 1+2z+4z^2)$	$\{x\}$	2, 6
11	$(0, \alpha_{11}+\beta z)$	$\{y\}$, $\{xy\}$	4, 5
11	$(0, \alpha_{11}+\beta z)$	$\{x\}$	2, 6
13	$(0, \alpha_{13}+\beta z)$	$\{y\}$, $\{xy\}$	4, 5
13	$(0, \alpha_{13}+\beta z)$	$\{x\}$	2, 6
13	(z, α)	$\{x\}$	2, 6
13	$(2z, \alpha'_{13}-2\alpha'_{13}z)$	$\{y\}$, $\{xy\}$	6, 7
5,7,11	(z, α')	$\{x\}$	2, 6
$p\geq 5$	$(2z, 1)$	$\{x\}$	2, 6
$p\geq 5$	$(z, 2z^2)$	$\{x\}$	1, 4
$p\geq 7$	$(z, 1-2^{-1}z-2^{-1}z^2)$	$\{x^2\}$	2, 6

$\alpha, \beta = 1, \ldots, 6$, $\alpha_5 \leq 2$, $\alpha_7 = 1, 4$, $\alpha_{11} = 1, \ldots, 8$,
$\alpha_{13} = 2, \ldots, 9$, $\alpha' = 1, \ldots, p-3$

$[n,k,\delta]$=[2,1,3] d_{free}=6

p	(a,b)	$L(G)$	i,$D=\sum P_i$
2	$(1+z, 1)$	$\{y\}$	3, 6
			d_{free}=8
7,13	$(0, 1+\beta z)$	$\{y\}$	3, 6
11,13	$(0, 2+\beta z)$	$\{y\}$	3, 6
7,13	$(0, 3+\beta z)$	$\{y\}$	3, 6
11,13	$(0, 5+\beta z)$	$\{y\}$	3, 6
11,13	$(0, 6+\beta z)$	$\{y\}$	3, 6
5,11,13	$(\alpha_1 z, 1)$	$\{y\}$	3, 6
7,11,13	$(\alpha_2 z, \beta_2)$	$\{y\}$	3, 6

$\beta = 1, \ldots, 6$, $\alpha_i = 1, 2$, $\beta_2 = 2, 4$

Some of these codes are actually sMDS (as presented in the next subsection)

$[n,k,\delta]$=[2,1,4] d_{free}=10

p	(a,b)	$L(G)$	i,$D=\sum P_i$
11	$(0,1+2z+6z^2)$	$\{y\}$, $\{xy\}$	4, 5
11	$(0,1+4z+2z^2)$	$\{y\}$, $\{xy\}$	4, 5
13	$(0,1+z+z^2)$	$\{y\}$, $\{xy\}$	4, 5
13	$(0,1+2z+4z^2)$	$\{y\}$, $\{xy\}$	4, 5
13	$(0,1+3z+4z^2)$	$\{y\}$, $\{xy\}$	4, 5
13	$(0,1+5z+z^2)$	$\{y\}$, $\{xy\}$	4, 5
13	$(0,1+6z+3z^2)$	$\{y\}$, $\{xy\}$	4, 5
13	$(0,3+z+4z^2)$	$\{y\}$, $\{xy\}$	4, 5
13	$(0,3+z+6z^2)$	$\{y\}$, $\{xy\}$	4, 5
13	$(0,3+2z+3z^2)$	$\{y\}$, $\{xy\}$	4, 5
13	$(0,3+3z+2z^2)$	$\{y\}$, $\{xy\}$	4, 5
13	$(0,3+2z+3z^2)$	$\{x\}$	2, 6
13	$(0,3+z+4z^2)$	$\{x\}$	2, 6
11,13	$(0,3+\alpha z)$	$\{x^2\}$	2, 6
11,13	$(0,6+\alpha z)$	$\{x^2\}$	2, 6
11,13	$(z,2z+2z^2)$	$\{x^2\}$	1, 4
11,13	$(z,4z+5z^2)$	$\{x^2\}$	1, 4

$\alpha = 1,\ldots,6$

$[n,k,\delta]$=[2,1,5] d_{free}=12

p	(a,b)	$L(G)$	i,$D=\sum P_i$
11	$(0,2+\alpha z)$	$\{xy\}$	3, 6
13	$(0,3+\alpha z)$	$\{xy\}$	3, 6
13	$(0,6+\alpha z)$	$\{xy\}$	3, 6
11	$(4z^2,5+2z+6z^2)$	$\{y\}$, $\{xy\}$	6, 7
11	$(5z^2,5+4z+2z^2)$	$\{y\}$, $\{xy\}$	6, 7
13	$(z^2,4+3z+6z^2)$	$\{y\}$, $\{xy\}$	6, 7
13	$(3z^2,4+z+5z^2)$	$\{y\}$, $\{xy\}$	6, 7

$\alpha = 1,\ldots,6$

$[n,k,\delta]$=[2,1,6] d_{free}=14

p	(a,b)	$L(G)$	i,$D=\sum P_i$
11	$(z^2,4+3z+4z^2)$	$\{y\}$, $\{xy\}$	6, 7
11	$(3z^2,4+4z+z^2)$	$\{y\}$, $\{xy\}$	6, 7
11	$(4z^2,4+5z+5z^2)$	$\{y\}$, $\{xy\}$	6, 7
11	$(4z^2,4+6z+5z^2)$	$\{y\}$, $\{xy\}$	6, 7
13	$(0,2+z+z^2)$	$\{y\}$	3, 6
13	$(0,2+2z+4z^2)$	$\{y\}$	3, 6
13	$(0,2+4z+3z^2)$	$\{y\}$	3, 6
13	$(0,3+3z+3z^2)$	$\{y\}$	3, 6
13	$(0,3+4z+z^2)$	$\{y\}$	3, 6
13	$(0,3+5z+4z^2)$	$\{y\}$	3, 6
13	$(z+6z^2,6)$	$\{y\}$	3, 6

$[n,k,\delta]$=[2,1,7] d_{free}=15

p	(a,b)	$L(G)$	i,$D=\sum P_i$
7	$(z+2z^2, 1+4z+3z^2)$	$\{y\}$, $\{xy\}$	6, 7
7	$(2z+z^2, 1+z+5z^2)$	$\{y\}$, $\{xy\}$	6, 7
			d_{free}=16
13	$(3z^2, 1+2z+2z^2)$	$\{y\}$, $\{xy\}$	6, 7
13	$(z^2, 1+6z+5z^2)$	$\{y\}$, $\{xy\}$	6, 7

$[n,k,\delta]$=[3,1,1] d_{free}=6

p	(a,b)	$L(G)$	i,$D=\sum P_i$
$p\geq 5$	$(z, 2-5z+3z^2)$	$\{y\}$, $\{xy\}$	1, 4, 5
5	$(z, 1+2z+2z^2)$	$\{y\}$, $\{xy\}$	1, 4, 5
5	$(2z, 1+4z+3z^2)$	$\{y\}$, $\{xy\}$	1, 4, 5
5	$(2z, 2+2z^2)$	$\{y\}$, $\{xy\}$	1, 4, 5
7	$(z, 1+4z+2z^2)$	$\{y\}$, $\{xy\}$	1, 4, 5
7	$(z, 3+4z^2)$	$\{y\}$, $\{xy\}$	1, 4, 5
7	$(2z, 4+3z+6z^2)$	$\{y\}$, $\{xy\}$	1, 4, 5
11	$(z, 3+4z+4z^2)$	$\{y\}$, $\{xy\}$	1, 4, 5
11	$(z, 4+2z+5z^2)$	$\{y\}$, $\{xy\}$	1, 4, 5
11	$(2z, 5+2z^2)$	$\{y\}$, $\{xy\}$	1, 4, 5
13	$(2z, 6+2z^2)$	$\{y\}$, $\{xy\}$	1, 4, 5

$[n,k,\delta]$=[3,1,2] d_{free}=8

p	(a,b)	$L(G)$	i,$D=\sum P_i$
2	(z, z^2)	$\{x\}$	1, 4, 5
			d_{free}=9
7	$(z, 1+2z+4z^2)$	$\{y\}$	1, 4, 5
7	$(2z, 1+4z+2z^2)$	$\{y\}$	1, 4, 5
11	$(z, 1+4z+6z^2)$	$\{y\}$	1, 4, 5
11	$(2z, 1+2z+3z^2)$	$\{y\}$	1, 4, 5
13	$(z, 1+6z+6z^2)$	$\{y\}$	1, 4, 5
13	$(2z, 1+2z+5z^2)$	$\{y\}$	1, 4, 5

$[n,k,\delta]$=[3,1,3] d_{free}=12

p	(a,b)	$L(G)$	i,$D=\sum P_i$
5	$(z, 2+2z+z^2)$	$\{xy\}$	1, 4, 5
5	$(2z, 2+4z+4z^2)$	$\{xy\}$	1, 4, 5
7	$(z, 1+z+5z^2)$	$\{xy\}$	1, 4, 5
7	$(z, 1+2z+4z^2)$	$\{xy\}$	1, 4, 5
7	$(z, 1+5z+z^2)$	$\{xy\}$	1, 4, 5
7	$(z, 2+5z^2)$	$\{xy\}$	1, 4, 5
11	$(z, 1+4z+6z^2)$	$\{xy\}$	1, 4, 5
11	$(z, 1+6z+4z^2)$	$\{xy\}$	1, 4, 5
13	$(z, 2+5z+6z^2)$	$\{xy\}$	1, 4, 5
13	$(z, 2+6z+5z^2)$	$\{xy\}$	1, 4, 5

$[n,k,\delta]$=[3,1,4] d_{free}=15

p	(a,b)	$L(G)$	i,$D=\sum P_i$
7,11,13	$(z, 2+3z^2)$	$\{y\}$	1, 4, 5
7,11,13	$(z, 2+3z+z^2)$	$\{y\}$	1, 4, 5
7,11,13	$(z, 4+3z+3z^2)$	$\{y\}$	1, 4, 5
7,11,13	$(z, 4+3z+5z^2)$	$\{y\}$	1, 4, 5

$[n,k,\delta]$=[3,1,6] d_{free}=21

p	(a,b)	$L(G)$	i,$D=\sum P_i$
11	$(z, 1+3z^2)$	$\{xy\}$	1, 4, 5
11	$(z, 1+2z+3z^2)$	$\{xy\}$	1, 4, 5
11	$(z, 1+3z+5z^2)$	$\{xy\}$	1, 4, 5
13	$(z, 1+4z^2)$	$\{xy\}$	1, 4, 5
13	$(z, 1+5z^2)$	$\{xy\}$	1, 4, 5
13	$(z, 1+3z+z^2)$	$\{xy\}$	1, 4, 5

$[n,k,\delta]$=[4,1,2] $d_{free}=12$

p	(a,b)	$L(G)$	i,$D=\sum P_i$
5	$(z, 1+2z+2z^2)$	$\{y\}$	2, 3, 6, 7

$[n,k,\delta]$=[4,1,3] $d_{free}=16$

p	(a,b)	$L(G)$	i,$D=\sum P_i$
7	$(z, 1+4z+2z^2)$	$\{xy\}$	2, 3, 6, 7
7	$(z, 2+2z+3z^2)$	$\{xy\}$	2, 3, 6, 7
11	$(z, 2+6z+3z^2)$	$\{xy\}$	2, 3, 6, 7
11	$(z, 4+2z+5z^2)$	$\{xy\}$	2, 3, 6, 7
13	$(z, 3+6z+4z^2)$	$\{xy\}$	2, 3, 6, 7
13	$(z, 4+4z+5z^2)$	$\{xy\}$	2, 3, 6, 7

$[n,k,\delta]$=[4,1,4] $d_{free}=20$

p	(a,b)	$L(G)$	i,$D=\sum P_i$
11	$(2z, 4+z+4z^2)$	$\{y\}$	2, 3, 6, 7

$[n,k,\delta]$=[4,1,6] $d_{free}=28$

p	(a,b)	$L(G)$	i,$D=\sum P_i$
11	$(z, 1+4z+6z^2)$	$\{xy\}$	2, 3, 6, 7
11	$(3z, 2+4z^2)$	$\{xy\}$	2, 3, 6, 7
11	$(4z, 2+z^2)$	$\{xy\}$	2, 3, 6, 7
13	$(z, 4+3z+6z^2)$	$\{xy\}$	2, 3, 6, 7
13	$(z, 5+5z+3z^2)$	$\{xy\}$	2, 3, 6, 7
13	$(z, 5+6z+2z^2)$	$\{xy\}$	2, 3, 6, 7

$[n,k,\delta]$=[4,1,8] $d_{free}=36$

p	(a,b)	$L(G)$	i,$D=\sum P_i$
11	$(3z, 6+4z^2)$	$\{y\}$	2, 3, 6, 7
11	$(4z, 6+z^2)$	$\{y\}$	2, 3, 6, 7
13	$(z, 3+5z+2z^2)$	$\{y\}$	2, 3, 6, 7
13	$(z, 5+2z+4z^2)$	$\{y\}$	2, 3, 6, 7

3.4.2 Codes with Dimension > 1

The following tables present different optimal $[n, k, \delta]$ convolutional codes with $k > 2$. In order to give a sample impression, in the following two tables also the generator matrix, in modified Kronecker-Hermite canonical form, is given. Hence, each entry represents just one code.

$[n, k, \delta; m] = [3, 2, 1; 1]$ $\qquad$ $d_{free} = 3$

p	(a,b)	$L(G)$	i,$D = \sum P_i$	$G(z)$
5	$(1+z, z+3z^2)$	$\{x, y\}$	1, 4, 5	$\begin{pmatrix} 3 & 1 & 1 \\ z & 4+z & 0 \end{pmatrix}$
5	$(1+z, z+3z^2)$	$\{x^2, xy\}$	1, 4, 5	$\begin{pmatrix} 4 & 1 & 1 \\ 3z & 4+z & 0 \end{pmatrix}$
11	$(1+z, z+3z^2)$	$\{x, y\}$	1, 4, 5	$\begin{pmatrix} 3 & 2 & 2 \\ 6z & 5+7z & 0 \end{pmatrix}$
11	$(1+z, z+3z^2)$	$\{x^2, xy\}$	1, 4, 5	$\begin{pmatrix} 3 & 5 & 5 \\ z & 3+2z & 0 \end{pmatrix}$
13	$(1+z, z+3z^2)$	$\{x, y\}$	1, 4, 5	$\begin{pmatrix} 4 & 1 & 1 \\ 7z & 6+12z & 0 \end{pmatrix}$
13	$(1+z, z+3z^2)$	$\{x^2, xy\}$	1, 4, 5	$\begin{pmatrix} 1 & 9 & 9 \\ 12z & 10+7z & 0 \end{pmatrix}$
17	$(1+z, z+3z^2)$	$\{x, y\}$	1, 4, 5	$\begin{pmatrix} 7 & 13 & 13 \\ 16z & 1+z & 0 \end{pmatrix}$
17	$(1+z, z+3z^2)$	$\{x^2, xy\}$	1, 4, 5	$\begin{pmatrix} 15 & 16 & 16 \\ 10z & 13+13z & 0 \end{pmatrix}$
7	$(1+3z, 3z+3z^2)$	$\{x, y\}$	1, 4, 5	$\begin{pmatrix} 1 & 2 & 2 \\ 2z & 4+5z & 0 \end{pmatrix}$
7	$(1+3z, 3z+3z^2)$	$\{x^2, xy\}$	1, 4, 5	$\begin{pmatrix} 2 & 1 & 1 \\ 2z & 1+3z & 0 \end{pmatrix}$
11	$(1+3z, 3z+3z^2)$	$\{x, y\}$	1, 4, 5	$\begin{pmatrix} 5 & 2 & 2 \\ 2z & 3+z & 0 \end{pmatrix}$
11	$(1+3z, 3z+3z^2)$	$\{x^2, xy\}$	1, 4, 5	$\begin{pmatrix} 1 & 5 & 5 \\ 3z & 4+5z & 0 \end{pmatrix}$
13	$(1+3z, 3z+3z^2)$	$\{x, y\}$	1, 4, 5	$\begin{pmatrix} 4 & 3 & 3 \\ 10z & 1+11z & 0 \end{pmatrix}$
13	$(1+3z, 3z+3z^2)$	$\{x^2, xy\}$	1, 4, 5	$\begin{pmatrix} 1 & 3 & 3 \\ 2z & 6+z & 0 \end{pmatrix}$
7	$(1+z, z+2z^2)$	$\{x, y\}$	1, 4, 5	$\begin{pmatrix} 3 & 1 & 1 \\ z & 6+6z & 0 \end{pmatrix}$
7	$(1+z, z+2z^2)$	$\{x^2, xy\}$	1, 4, 5	$\begin{pmatrix} 1 & 4 & 4 \\ 6z & 5+5z & 0 \end{pmatrix}$
11	$(1+z, z+2z^2)$	$\{x, y\}$	1, 4, 5	$\begin{pmatrix} 3 & 5 & 5 \\ 3z & 8+z & 0 \end{pmatrix}$
11	$(1+z, z+2z^2)$	$\{x^2, xy\}$	1, 4, 5	$\begin{pmatrix} 3 & 1 & 1 \\ 5z & 10+4z & 0 \end{pmatrix}$
13	$(1+z, z+2z^2)$	$\{x, y\}$	1, 4, 5	$\begin{pmatrix} 5 & 3 & 3 \\ z & 12+z & 0 \end{pmatrix}$
13	$(1+z, z+2z^2)$	$\{x^2, xy\}$	1, 4, 5	$\begin{pmatrix} 4 & 3 & 3 \\ 7z & 1+12z & 0 \end{pmatrix}$
17	$(1+z, z+2z^2)$	$\{x, y\}$	1, 4, 5	$\begin{pmatrix} 9 & 16 & 16 \\ 3z & 14+7z & 0 \end{pmatrix}$
17	$(1+z, z+2z^2)$	$\{x^2, xy\}$	1, 4, 5	$\begin{pmatrix} 13 & 1 & 1 \\ z & 2+z & 0 \end{pmatrix}$

$[n, k, \delta; m] = [3, 2, 1; 1]$ $\qquad$ $d_{free} = 3$

p	(a,b)	$L(G)$	i,$D = \sum P_i$	$G(z)$
7	$(1+2z, 2z+z^2)$	$\{x, y\}$	1, 4, 5	$\begin{pmatrix} 3 & 1 & 1 \\ z & 3+6\,z & 0 \end{pmatrix}$
7	$(1+2z, 2z+z^2)$	$\{x^2, xy\}$	1, 4, 5	$\begin{pmatrix} 2 & 1 & 1 \\ z & 1+2\,z & 0 \end{pmatrix}$
11	$(1+2z, 2z+z^2)$	$\{x, y\}$	1, 4, 5	$\begin{pmatrix} 7 & 1 & 1 \\ z & 5+3\,z & 0 \end{pmatrix}$
11	$(1+2z, 2z+z^2)$	$\{x^2, xy\}$	1, 4, 5	$\begin{pmatrix} 5 & 1 & 1 \\ z & 7+2\,z & 0 \end{pmatrix}$
13	$(1+2z, 2z+z^2)$	$\{x, y\}$	1, 4, 5	$\begin{pmatrix} 9 & 1 & 1 \\ z & 6+z & 0 \end{pmatrix}$
13	$(1+2z, 2z+z^2)$	$\{x^2, xy\}$	1, 4, 5	$\begin{pmatrix} 3 & 1 & 1 \\ z & 5+3\,z & 0 \end{pmatrix}$
17	$(1+2z, 2z+z^2)$	$\{x, y\}$	1, 4, 5	$\begin{pmatrix} 13 & 1 & 1 \\ 7\,z & 5+3\,z & 0 \end{pmatrix}$
17	$(1+2z, 2z+z^2)$	$\{x^2, xy\}$	1, 4, 5	$\begin{pmatrix} 16 & 1 & 1 \\ 2\,z & 13+z & 0 \end{pmatrix}$
5	$(1+2z, 2z+2z^2)$	$\{x, y\}$	1, 4, 5	$\begin{pmatrix} 3 & 1 & 1 \\ z & 2+z & 0 \end{pmatrix}$
5	$(1+2z, 2z+2z^2)$	$\{x^2, xy\}$	1, 4, 5	$\begin{pmatrix} 4 & 1 & 1 \\ 3\,z & 2+z & 0 \end{pmatrix}$
7	$(1+2z, 2z+2z^2)$	$\{x, y\}$	1, 4, 5	$\begin{pmatrix} 3 & 2 & 2 \\ 5\,z & 1+6\,z & 0 \end{pmatrix}$
7	$(1+2z, 2z+2z^2)$	$\{x^2, xy\}$	1, 4, 5	$\begin{pmatrix} 1 & 2 & 2 \\ 3\,z & 6+z & 0 \end{pmatrix}$
11	$(1+2z, 2z+2z^2)$	$\{x, y\}$	1, 4, 5	$\begin{pmatrix} 1 & 5 & 5 \\ z & 5+3\,z & 0 \end{pmatrix}$
11	$(1+2z, 2z+2z^2)$	$\{x^2, xy\}$	1, 4, 5	$\begin{pmatrix} 1 & 3 & 3 \\ z & 3+4\,z & 0 \end{pmatrix}$
13	$(1+2z, 2z+2z^2)$	$\{x, y\}$	1, 4, 5	$\begin{pmatrix} 7 & 3 & 3 \\ 2\,z & 12+3\,z & 0 \end{pmatrix}$
13	$(1+2z, 2z+2z^2)$	$\{x^2, xy\}$	1, 4, 5	$\begin{pmatrix} 10 & 9 & 9 \\ 3\,z & 4+z & 0 \end{pmatrix}$
17	$(1+2z, 2z+2z^2)$	$\{x, y\}$	1, 4, 5	$\begin{pmatrix} 1 & 8 & 8 \\ 7\,z & 5+z & 0 \end{pmatrix}$
17	$(1+2z, 2z+2z^2)$	$\{x^2, xy\}$	1, 4, 5	$\begin{pmatrix} 4 & 1 & 1 \\ 7\,z & 6+8\,z & 0 \end{pmatrix}$
5	$(1-z, -z+z^2)$	$\{x, y\}$	1, 4, 5	$\begin{pmatrix} 4 & 1 & 1 \\ z & 1+z & 0 \end{pmatrix}$
5	$(1-z, -z+z^2)$	$\{x^2, xy\}$	1, 4, 5	$\begin{pmatrix} 1 & 1 & 1 \\ 2\,z & 3+3\,z & 0 \end{pmatrix}$
7	$(1-z, -z+z^2)$	$\{x, y\}$	1, 4, 5	$\begin{pmatrix} 6 & 1 & 1 \\ 6\,z & 6+5\,z & 0 \end{pmatrix}$
7	$(1-z, -z+z^2)$	$\{x^2, xy\}$	1, 4, 5	$\begin{pmatrix} 1 & 1 & 1 \\ 2\,z & 5+3\,z & 0 \end{pmatrix}$
11	$(1-z, -z+z^2)$	$\{x, y\}$	1, 4, 5	$\begin{pmatrix} 10 & 1 & 1 \\ 10\,z & 10+7\,z & 0 \end{pmatrix}$
11	$(1-z, -z+z^2)$	$\{x^2, xy\}$	1, 4, 5	$\begin{pmatrix} 1 & 1 & 1 \\ 2\,z & 9+3\,z & 0 \end{pmatrix}$
13	$(1-z, -z+z^2)$	$\{x, y\}$	1, 4, 5	$\begin{pmatrix} 12 & 1 & 1 \\ 3\,z & 3+2\,z & 0 \end{pmatrix}$
13	$(1-z, -z+z^2)$	$\{x^2, xy\}$	1, 4, 5	$\begin{pmatrix} 1 & 1 & 1 \\ 2\,z & 11+3\,z & 0 \end{pmatrix}$
17	$(1-z, -z+z^2)$	$\{x, y\}$	1, 4, 5	$\begin{pmatrix} 16 & 1 & 1 \\ 8\,z & 8+5\,z & 0 \end{pmatrix}$
17	$(1-z, -z+z^2)$	$\{x^2, xy\}$	1, 4, 5	$\begin{pmatrix} 1 & 1 & 1 \\ 2\,z & 15+3\,z & 0 \end{pmatrix}$

$[n, k, \delta; m] = [3, 2, 3; 2]$ $d_{free} = 6$

p	(a,b)	$L(G)$	i,$D = \sum P_i$
7	$(z, 2 + 5z^2)$	$\{x, y\}$	1, 4, 5
7	$(2z, 2 + 6z^2)$	$\{x, y\}$	1, 4, 5
7	$(3 + 3z, 3z + z^2)$	$\{x, y\}$	1, 4, 5
11	$(5z, 1 + z + 3z^2)$	$\{x, y\}$	1, 4, 5
11	$(2 + 3z, 2 + 4z^2)$	$\{x, y\}$	1, 4, 5
11	$(3 + 4z, z + 2z^2)$	$\{x, y\}$	1, 4, 5
11	$(3 + 4z, 2z + 4z)$	$\{x, y\}$	1, 4, 5
13	$(5z, 2 + 2z^2)$	$\{x, y\}$	1, 4, 5

$[n, k, \delta; m] = [4, 2, 1; 1]$ $d_{free} = 4$

p	(a,b)	$L(G)$	i,$D = \sum P_i$
$p \geq 3$	$(z, -2 + 2z)$	$\{1, x\}$	0, 1, 2, 4
$p \geq 3$	$(z, -2 + 3z - z^2)$	$\{1, x\}$	0, 1, 2, 4
$p \geq 3$	$(z, -2 + 3z - z^2)$	$\{1, x\}$	1, 2, 4, 6
11	$(4 + 4z, 5 + 2z + 6z^2)$	$\{1, x\}$	1, 2, 4, 6

$[n, k, \delta; m] = [4, 2, 3; 2]$ $d_{free} = 8$

p	(a,b)	$L(G)$	i,$D = \sum P_i$
7	$(z, 3 + 4z^2)$	$\{x, y\}$	2, 3, 6, 7
7	$(2z, 3 + 2z^2)$	$\{x, y\}$	2, 3, 6, 7
11	$(z, 5 + 6z^2)$	$\{x, y\}$	2, 3, 6, 7
11	$(2z, 5 + 2z^2)$	$\{x, y\}$	2, 3, 6, 7
11	$(z, 5 + 6z^2)$	$\{x, y\}$	2, 3, 6, 7
13	$(z, 2 + 6z + 3z^2)$	$\{x, y\}$	2, 3, 6, 7
13	$(z, 3 + 6z + 4z^2)$	$\{x, y\}$	2, 3, 6, 7
13	$(z, 4 + 4z + 5z^2)$	$\{x, y\}$	2, 3, 6, 7

$[n, k, \delta; m] = [4, 2, 5; 3]$ $d_{free} = 8$

p	(a,b)	$L(G)$	i,$D = \sum P_i$
2	$(z, 1)$	$\{x, y\}$	2, 3, 6, 7
			d_{free}=12
13	$(z, 4 + 4z + 5z^2)$	$\{x, y\}$	2, 3, 6, 7

$[n, k, \delta; m] = [4, 3, 2; 1]$ $d_{free} = 4$

p	(a,b)	$L(G)$	i,$D = \sum P_i$
7	$(2 + \alpha z, 6 - \alpha z)$	$\{1, x, y\}$	1, 3, 6, 7
7	$(5 + \alpha z, 3 - \alpha z)$	$\{1, x, y\}$	1, 3, 6, 7
11	$(6 + \alpha z, 6 - \alpha z)$	$\{1, x, y\}$	1, 3, 6, 7

$\alpha \neq 0$

$[n, k, \delta; m] = [5, 2, 1; 1]$ $d_{free} = 5$

p	(a,b)	$L(G)$	i,$D = \sum P_i$
$p \geq 3$	$(z, -2 + 3z - z^2)$	$\{1, x\}$	0, 1, 2, 4, 6
11	$(4 + 4z, 5 + 2z + 6z^2)$	$\{1, x\}$	0, 1, 2, 4, 6

3.4.3 Strongly MDS Convolutional Codes

The set of *strongly MDS* convolutional codes is a particularly interesting subset of MDS convolutional codes. They are characterized by the property that their free distance is attained by the earliest column distance possible. This can be interpreted in the sense that to decode a strongly MDS convolutional code, the smallest possible number of vector coefficients of the received word are needed in each step. This property is very convenient to develop iterative decoding algorithms, as the one in [GLRS03], with an error decoding capability per time interval similar to MDS block codes of a large length.

The family of convolutional Goppa codes defined over elliptic curves contains also some codes which are strongly MDS. Some of them are presented in the following tables.

We would like to stress two interesting facts. In [GLRS03] several examples of strongly MDS convolutional codes are presented, which are obtained by different methods. However all of them have in common that the length of the code and the characteristic of the base field have to be coprime. For some of the codes shown here this condition doesn't need to be fulfilled. Secondly, the already mentioned decoding algorithm for this kind of codes that is proposed in the same paper, has the drawback of needing a general syndrome decoding algorithm. The variety of examples presented below suggests that convolutional Goppa codes over elliptic curves are a promising way to obtain strongly MDS codes with an algebraic structure that would allow to use in practice that decoding scheme.

$[n, k, \delta]$=[2,1,1] $d_{free} = 4$

p	(a,b)	$L(G)$	i,$D = \sum P_i$
5	$(z, 1 + 2z + 2z^2)$	$\{x\}$	2, 6
5	$(z, 2 + 3z^2)$	$\{x\}$	2, 6
7	$(z, 1 + 4z + 2z^2)$	$\{x\}$	2, 6
7	$(z, 2 + 2z + 3z^2)$	$\{x\}$	2, 6
11	$(z, 2 + 6z + 3z^2)$	$\{x\}$	2, 6

$[n, k, \delta]$=[2,1,2] $d_{free} = 6$

p	(a,b)	$L(G)$	i,$D = \sum P_i$
11	$(0, 2 + \alpha z)$	$\{y\}$, $\{xy\}$	4, 5
13	$(0, 3 + \alpha z)$	$\{y\}$, $\{xy\}$	4, 5
13	$(0, 5 + \alpha z)$	$\{y\}$, $\{xy\}$	4, 5
13	$(0, 6 + \alpha z)$	$\{y\}$, $\{xy\}$	4, 5
11,13	$(z, 3)$	$\{y\}$, $\{xy\}$	4, 5
11,13	$(2z, 3)$	$\{y\}$, $\{xy\}$	4, 5
13	$(z, 5)$	$\{y\}$, $\{xy\}$	4, 5
13	$(2z, 5)$	$\{y\}$, $\{xy\}$	4, 5
13	$(2z, 5 + 3z)$	$\{y\}$, $\{xy\}$	6, 7

$\alpha = 1, \ldots, 6$

$[n,k,\delta;m]=[3,2,1;1]$ $d_{free}=3$

p	(a,b)	$L(G)$	i,$D=\sum P_i$
3	$(z^2,1+z)$	$\{1,x\}$	1, 2, 6
3	$(z^2,1+2z)$	$\{1,x\}$	1, 2, 6
3	$(z,1+2z)$	$\{1,x\}$	1, 2, 6
5	$(2+\alpha z,3+3\alpha z)$	$\{1,x\}$	1, 2, 6
7	$(2+\alpha z,\alpha_7+\alpha\alpha_7 z)$	$\{1,x\}$	1, 2, 6
11	$(2+\alpha z,\alpha_{11}+\alpha\alpha_{11} z)$	$\{1,x\}$	1, 2, 6
13	$(2+\alpha z,\alpha_{13}+\alpha\alpha_{13} z)$	$\{1,x\}$	1, 2, 6
7,11,13	$(2+\beta_1 z,\beta_2+\beta_3 z+\beta_4 z^2)$	$\{1,x\}$	0, 2, 6
$p\geq 5$	$(z,1-3z+2z^2)$	$\{x,y\}$,$\{x^2,xy\}$	1, 4, 5
$p\geq 5^{(*)}$	$(z,2-5z+3z^2)$	$\{x,y\}$,$\{x^2,xy\}$	1, 4, 5

$\alpha\geq 1$ α_7=2,4,6, $\alpha_{11}\geq 2$, $\alpha_{13}\neq 3,4$

$\beta_3=(2\beta_2-1)\beta_1$, $\beta_4=(\beta_2-1)\beta_1^2$, $\beta_1\geq 1$, $\beta_2\geq 2$, $\beta_i\neq 0\ \forall i$

$^{(*)}$ Except for $(p,L(G))=(17,\{x,y\})$

$[n,k,\delta;m]=[4,2,1;1]$ $d_{free}=4$

p	(a,b)	$L(G)$	i,$D=\sum P_i$
5	$(z,1+4z)$	$\{1,x\}$	0, 1, 2, 4
7	$(z,1+6z)$	$\{1,x\}$	0, 1, 2, 4
7	$(z,2+5z)$	$\{1,x\}$	0, 1, 2, 4
7	$(z,3+4z)$	$\{1,x\}$	0, 1, 2, 4
7	$(z,4+3z)$	$\{1,x\}$	0, 1, 2, 4
11	$(z,5+6z)$	$\{1,x\}$	0, 1, 2, 4
11	$(z,6+5z)$	$\{1,x\}$	0, 1, 2, 4
5,11	$(2+\alpha z,\alpha_5+\alpha\alpha_5 z)$	$\{1,x\}$	0, 1, 2, 4
7	$(2+\alpha z,\alpha_7+\alpha\alpha_7 z)$	$\{1,x\}$	0, 1, 2, 4
13	$(2+\alpha z,\alpha_{13}+\alpha\alpha_{13} z)$	$\{1,x\}$	0, 1, 2, 4

$\alpha\geq 1$ $\alpha_5\geq 2$ α_7=2,4,6, $\alpha_{13}\neq 4$

3.5 AG Convolutional Codes

3.5.1 AG Block Codes

The breakthrough of Goppa codes attracted the attention of coding theorists to algebraic geometry. As a result, new constructions of block codes arose based on geometric elements such as points, curves, surfaces and functions defined over them, and algebraic geometric tools were used to calculate the parameters of these codes.

Some notorious examples have been given by Xing, Niederreiter and Lam in [NXL99, XNL99, NLX99] and afterwards in [Pre01]. With respect to the first ones, Özbudak and Stichtenoth showed in [OS99] the relationships between those constructions and they proved that one of them, termed *Generalized AG codes*, is a generalization of the others and also of the classical Goppa codes.

We will present here these Generalized AG codes as they appear in [XNL99, OS99], and we will propose, as we have done for Goppa codes, an analogous family

of convolutional codes.

3.5.2 Generalized AG Codes

Let X be a smooth projective curve of genus g over $\mathbb{F}_q$ and let us take s different points $P_1, \ldots, P_s$ from X of degree $degP_i = k_i$. Let us consider a divisor G without support in $\{P_1, \ldots, P_s\}$. For each $i = 1, \ldots, s$ let $\pi_i : \mathcal{O}_{P_i}/_{\mathfrak{m}_{P_i}} \to \mathcal{C}_i$ be an $\mathbb{F}_q$-linear isomorphism from the residue field of the point P_i, $\mathcal{O}_{P_i}/_{\mathfrak{m}_{P_i}} \simeq \mathbb{F}_q^{k_i}$, onto a linear $[n_i, k_i, d_i]$ block code $\mathcal{C}_i \subseteq \mathbb{F}_q^{n_i}$. Let $n = \sum n_i$ and consider the linear map

$$\begin{aligned} \pi : L(G) &\longrightarrow \mathbb{F}_q^n \\ f &\longmapsto (\pi_1(f(P_1)), \ldots, \pi_s(f(P_s))) \end{aligned}$$

Definition 3.17. The *generalized AG code* defined by $P_1, \ldots, P_s$, the divisor G and the codes $\mathcal{C}_1, \ldots, \mathcal{C}_s$, $\mathcal{C}(P_1, \ldots, P_s; G; \mathcal{C}_1, \ldots, \mathcal{C}_s)$, is the image of π.

Definition 3.18. Let

$$I = \left\{ S \subseteq \{1, \ldots, s\} \mid \sum_{i \in S} k_i \leq degG \right\}.$$

The *designed minimum distance* δ of the generalized AG code $\mathcal{C}$ is

$$\delta := min \left\{ \sum_{i \notin S} d_i \mid S \in I \right\}.$$

Proposition 3.19 ([OS99])**.** *Assume* $deg(G) < \sum_{i=1}^{s} k_i$. *Then the generalized AG code* $\mathcal{C}$ *is an* $[n, k, d]$ *code with*

$$k = l(G) \geq deg(G) + 1 - g, \quad d \geq \delta.$$

3.5.3 AG Convolutional Codes

Considering a similar construction we will be able to use convolutional codes of small length to produce longer codes with optimal distance, even when the ones used are not optimal.

Let X be a smooth projective curve over $\mathbb{F}_q(z)$ of genus g, and let us take s different points from X $P_1, \ldots, P_s$ of degree $degP_i = k_i$ and a divisor G with $suppG \cap \{P_1, \ldots, P_s\} = \emptyset$. Let us consider s convolutional codes $\mathcal{C}_1, \ldots, \mathcal{C}_s$ (which as algebraic objects are $\mathbb{F}_q(z)$-vector spaces), with $\mathcal{C}_i$ a code of type $[n_i, k_i]$ and free distance d_i $\forall i = 1, \ldots, s$, and s $\mathbb{F}_q(z)$-linear isomorphisms $\pi_i : \mathcal{O}_{P_i}/_{\mathfrak{m}_{P_i}} \to \mathcal{C}_i$.

Definition 3.20. The image of the linear map

$$\begin{aligned} \pi : L(G) &\longrightarrow \mathbb{F}_q(z)^n \\ f &\longmapsto (\pi_1(f(P_1)), \ldots, \pi_s(f(P_s))) \end{aligned}$$

with $n = \sum n_i$ is the *AG convolutional code* $\mathcal{C}(P_1, \ldots, P_s; G; \mathcal{C}_1, \ldots, \mathcal{C}_s)$ defined by $P_1, \ldots, P_s$, the divisor G and the convolutional codes $\mathcal{C}_1, \ldots, \mathcal{C}_s$.

Definition 3.21. Let

$$I = \left\{ S \subseteq \{1, \ldots, s\} \mid \sum_{i \in S} k_i \leq deg(G) \right\}.$$

The *designed free distance* δ_{free} of the generalized AG code $\mathcal{C}$ is

$$\delta_{\text{free}} := min \left\{ \sum_{i \notin S} d_i \mid S \in I \right\}.$$

Proposition 3.22. *If* $deg(G) < \sum_{i=1}^{s} k_i$, *then the map* π *is injective and the AG convolutional code* $\mathcal{C}$ *is an* $[n, k, d_{free}]$ *code with*

$$k = l(G) \geq deg(G) + 1 - g, \quad d_{free} \geq \delta_{free}.$$

Proof. The proof for the block case applies also here. □

Example 3.23. Let X be the projective line over $\mathbb{F}_5(z)$, α a root of the polynomial $x^2 + x + 1$ and the points $P_1 = (1; \alpha)$, $P_2 = (1; \alpha + 1)$, $deg(P_i) = 2$, with $\mathcal{O}_{P_1}/\mathfrak{m}_{P_1} = \mathcal{O}_{P_2}/\mathfrak{m}_{P_2} = \mathbb{F}_5(z)[x]/_{x^2+x+1} \simeq \mathbb{F}_5(z)(\alpha)$. Let $\mathcal{C}_1$, $\mathcal{C}_2$ be the codes from Example 3.13, generated by

$$G = \begin{pmatrix} z+1 & 2z+3 & 4z+4 & 3z+2 \\ (z+1)^2 & (2z+3)^2 & (4z+4)^2 & (3z+2)^2 \end{pmatrix}$$

$$H = \begin{pmatrix} 4(z+4) & 3(z+1) & (z+4) & 2(z+1) \\ (z+4)^2 & (z+1)^2 & (z+4)^2 & (z+1)^2 \end{pmatrix}.$$

Then, $d_1 = d_2 = 8$. Let us call $r_1^G, r_2^G, r_1^H, r_2^H$ the vectors in the first or second row of the matrices G or H respectively, and consider the morphisms

$$\begin{aligned} \pi_1 : \mathbb{F}_5(z)(\alpha) &\longrightarrow \mathcal{C}_1 \ , \quad \pi_2 : \mathbb{F}_5(z)(\alpha) \longrightarrow \mathcal{C}_2 \\ 1 &\longmapsto r_1^G \qquad\qquad 1 \longmapsto r_1^H \\ \alpha &\longmapsto r_2^G \qquad\qquad \alpha \longmapsto r_2^H - r_1^H \end{aligned}$$

We must choose a divisor G with degree < 4. We take $G = P_\infty$ and $L(G) = \langle 1, t \rangle$. As $deg(G) = 1$, the designed distance is $\delta_{free} = 16$.

Then, we have

$$\begin{array}{ccccc} L(G) & \to & (f(P_1), f(P_2)) & \longrightarrow & (\pi_1(f((P_1)), \pi_2(f(P_2))) \\ 1 & \mapsto & (1,1) & \mapsto & (z+1, 2z+3, 4z+4, 3z+2, 4(z+4), 3(z+1), (z+4), 2(z+1)) \\ t & \mapsto & (\alpha, \alpha+1) & \mapsto & ((z+1)^2, (2z+3)^2, (4z+4)^2, (3z+2)^2, (z+4)^2, (z+1)^2, (z+4)^2, (z+1)^2) \end{array}$$

and it results the code generated by

$$\begin{pmatrix} z+1 & 2z+3 & 4z+4 & 3z+2 & 4(z+4) & 3(z+1) & (z+4) & 2(z+1) \\ (z+1)^2 & (2z+3)^2 & (4z+4)^2 & (3z+2)^2 & (z+4)^2 & (z+1)^2 & (z+4)^2 & (z+1)^2 \end{pmatrix}$$

with parameters $[n, k, \delta] = [8, 2, 3]$ and for these parameters the designed distance reaches the generalized Singleton bound. Thus, the code is MDS by construction.

Alternatively, we may consider the divisor $G = 2P_\infty - P_0$ and $L(G) = \langle t, t^2 \rangle$. Again $deg(G) = 1$ and the designed distance is $\delta_{free} = 16$. We have

$$\begin{array}{ccccc} L(G) & \to & (f(P_1), f(P_2)) & \longrightarrow & (\pi_1(P_1), \pi_2(P_2)) \\ t & \mapsto & (\alpha, \alpha+1) & \mapsto & ((z+1)^2, (2z+3)^2, (4z+4)^2, (3z+2)^2, (z+4)^2, (z+1)^2, (z+4)^2, (z+1)^2) \\ t^2 & \mapsto & (\alpha+1, \alpha) & \mapsto & (z^2+3z+2, 4z^2+4z+2, z^2+z, 4z^2+1, z^2+4z, z^2+4z+3, z^2+2z+2, z^2+4) \end{array}$$

and as a result we obtain the code generated by

$$\begin{pmatrix} (z+1)^2 & (2z+3)^2 & (4z+4)^2 & (3z+2)^2 & (z+4)^2 & (z+1)^2 & (z+4)^2 & (z+1)^2 \\ z^2+3z+2 & 4z^2+4z+2 & z^2+z & 4z^2+1 & z^2+4z & z^2+4z+3 & z^2+2z+2 & z^2+4 \end{pmatrix}$$

with parameters $[n, k, \delta] = [8, 2, 3]$. Therefore this code is also MDS by construction.

Recall that the "composing" codes $\mathcal{C}_i$ don't need to be different. Then, the codes generated by the matrices G and H can be taken several times, and by choosing appropriate points P_i and isomorphisms π_i, the two constructions of this example can be used to give two optimal families of $[4r, 2, 3]$ MDS convolutional codes.

Remark 3.24. The knowledge on the words of minimum weight of the "composing" codes $\mathcal{C}_i$ can be used to improve the free distance of the resulting AG convolutional code by choosing the isomorphisms π_i so that the evaluation on the P_i of no function can be projected into codewords of minimum d_{free} of each code $\mathcal{C}_i$.

In our convolutional context, it is possible to improve the designed free distance of Definition 3.21 to get a bigger designed free distance in some particular cases.

Proposition 3.25. *Let $\mathcal{C}(P_1, \ldots, P_s; G; \mathcal{C}_1, \ldots, \mathcal{C}_s)$ be the AG convolutional code defined by the rational points $P_1, \ldots, P_s$, the divisor G and the convolutional codes $\mathcal{C}_1, \ldots, C_s$. Let $\{f_1, \ldots, f_k\}$ be a basis of $L(G)$ such that $\{\pi(f_1), \ldots, \pi(f_k)\}$ generate a free $\mathbb{F}_q[z]$-submodule of $\mathbb{F}_q[z]^n$, and for each $i \leq s$ let $d'_i \geq d_i$ be the minimum Hamming weight of the polynomial n_i-vectors from the $\mathbb{F}_q[z]$-submodule generated by $\{\pi_i(f_1(P_i)), \ldots, \pi_i(f_k(P_i))\}$, which is contained in $\mathcal{C}_i$. Let I be defined as before. Then, the convolutional code $\langle \pi(f_1), \ldots, \pi(f_k) \rangle_{\mathbb{F}_q[z]}$ has designed free distance*

$$\delta'_{free} := min \left\{ \sum_{i \notin S} d'_i \mid S \in I \right\}$$

and its free distance is $d_{free} \geq \delta'_{free}$.

Proof. The arguments for the previous analogous results apply also here considering the free distances of the submodules $\langle \pi_i(f_1(P_i)), \ldots, \pi_i(f_k(P_i))\rangle_{\mathbb{F}_q[z]}$ since $\langle \pi(f_1), \ldots, \pi(f_k)\rangle_{\mathbb{F}_q[z]}$ is a free submodule. □

Example 3.26. Let us consider the projective line over $\mathbb{F}_7(z)$, α a root of the polynomial x^2-x-1 and the points $P_1 = (1; \alpha z+3\alpha)$, $P_2 = (1; \alpha z+\alpha)$, $deg(P_i) = 2$, with $\mathcal{O}_{P_i}/_{\mathfrak{m}_{P_i}} = \mathbb{F}_7(z)[x]/_{x^2-x-1} \simeq \mathbb{F}_7(z)(\alpha)$ for $i = 1, 2$. Let $\mathcal{C}_1$, $\mathcal{C}_2$ be the codes generated respectively by

$$G = \begin{pmatrix} 3 & 1 & 1 \\ 6+z & 5+6z & 2 \end{pmatrix} \quad H = \begin{pmatrix} 2 & 1 & 1 \\ 2+z & 2+2z & 1 \end{pmatrix}$$

both of which have $d_{free} = 3$. Consider the morphisms

$$\begin{array}{ccc} \pi_1 : \mathbb{F}_7(z)(\alpha) \longrightarrow \mathcal{C}_1 \ , & & \pi_2 : \mathbb{F}_7(z)(\alpha) \longrightarrow \mathcal{C}_2 \\ \alpha \longmapsto r_1^G & & \alpha \longmapsto r_1^H \\ 1 \longmapsto r_2^G & & 1 \longmapsto r_2^H \end{array}$$

being $r_1^G, r_2^G, r_1^H, r_2^H$ the vectors in the first and second rows of the corresponding generator matrix. Let us take the divisor $G = P_\infty$, then $L(G) = \langle 1, t\rangle$. We have the morphism π

$$\begin{array}{ccccc} L(G) & \to & (f(P_1), f(P_2)) & \longrightarrow & (\pi_1(f((P_1)), \pi_2(f(P_2))) \\ 1 & \mapsto & (1,1) & \mapsto & (6+z, 5+6z, 2, 2+z, 2+2z, 1) \\ t & \mapsto & (\alpha z+3\alpha, \alpha z+\alpha) & \mapsto & (2+3z, 3+z, 3+z, 2+2z, 1+z, 1+z) \end{array}$$

and $Im\pi$ is a free submodule.

The minimum Hamming weight of the polynomial vectors in the submodules $\langle (6+z, 5+6z, 2), (2+3z, 3+z, 3+z)\rangle_{\mathbb{F}_7[z]}, \langle (2+z, 2+2z, 1), (2+2z, 1+z, 1+z)\rangle_{\mathbb{F}_7[z]}$ is in both cases 5.

The code defined by $Im\pi$ is generated by the matrix

$$\begin{pmatrix} 6+z & 5+6z & 2 & 2+z & 2+2z & 1 \\ 2+3z & 3+z & 3+z & 2+2z & 1+z & 1+z \end{pmatrix}.$$

We have $d_1' = d_2' = 5$ and, as $deg(G) = 1$, the designed distance of the code, of type $[6, 2, 2; 1]$, is $\delta_{free} = d_1' + d_2' = 10$ and hence by construction it reaches the Griesmer bound.

This modified construction allows to use not just codes but submodules that may have a good free distance. In this way codes with free distance not reaching any bound can be used to construct codes with an optimal free distance.

Chapter 4

Linear Systems and Convolutional Codes

4.1 Brief Introduction to Linear Systems

A system is a model of an isolated fragment of the Nature with a dynamic behavior that can be observed and studied. This behavior is the response of the system to an external stimulation, and this response may not be always the same, but rather depend also on the current circumstances of the system. In our model, there are variables which represent the external stimulation of the system, its present circumstances, and the response given by the system: the input variable u, the state variable x and the output variable y, respectively. All of them are functions of a time variable t.

The properties of the system under study are usually, but not always, simple enough so that it can be linearly modeled. A linear system can be then represented in terms of the time variable t as

$$\begin{aligned} \dot{x}(t) &= A(t)x(t) + B(t)u(t) \\ y(t) &= C(t)x(t) + D(t)u(t) \end{aligned} .$$

Depending on wether the system has a single or multiple input and a single or multiple output (these cases are usually considered separately) u, y can be single variables or variable vectors. A, B, C, D are matrices of rational functions on t of the appropriate dimensions. In the case where the matrices have constant entries the system is called a *time-invariant system*.

In the previous model, time is considered to be a continuous magnitude. However, some systems can be better modeled if the time variable is considered discrete, $t \in \mathbb{Z}$.

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

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t \\ y_t &= Cx_t + Du_t \end{aligned} \tag{4.1}$$

where $A \in K^{\delta\times\delta}$, $B \in K^{\delta\times k}$, $C \in K^{p\times\delta}$, $D \in K^{p\times k}$ are constant matrices over the field K, and $u_t \in K^k$, $x_t \in K^\delta$, $y_t \in K^p$ are the input, state and output vectors, respectively.

The quadruple (A, B, C, D) is called a *realization* of the system (4.1).

The realization of a system is not unique, and in particular, the size δ of the state vector may differ from some realizations to others. A realization with a minimum δ is called a *minimal realization* and the value of δ is the *McMillan degree* of the system.

A primary tool used in classical control theory to study the properties of a Single-Input-Single-Output system is the *transfer function*. This function relates directly the inputs and the outputs of a system. For Multiple-Input-Multiple-Output systems a matrix transfer function can be always obtained by means of the Laplace transform. Let $U(z)$, $X(z)$, $Y(z)$ be the Laplace transforms of the variables u, x, y of a time invariant linear system. Then by applying the Laplace transform to the equations of the system we have

$$\begin{aligned} zX(z) &= AX(z) + BU(z) \\ Y(z) &= CX(z) + DU(z) \end{aligned}$$

and as a result $Y(z) = T(z)U(z)$, where

$$T(z) = C(zId - A)^{-1}B + D$$

is the transfer function of the system.

We introduce now the fundamental notions of controllability and observability. Observability means the possibility of identifying the internal state of a system from measurements of the outputs. Controllability means instead the possibility of steering the system from any initial state to any final one by means of a control signal in the input. Let us formalize these concepts.

Definition 4.1. A linear system with a realization (A, B, C, D) is a *controllable system* if and only if the *controllability matrix* of the system

$$\begin{pmatrix} B & AB & A^2B & \dots & A^{\delta-1}B \end{pmatrix}$$

has full rank δ.

It can be shown that a system is controllable if for any couple of internal states there is a finite sequence of inputs which drives the system from one state to the other. This means that if δ columns of the controllability matrix are linearly independent then every state of the system is reachable via a proper finite sequence of inputs.

The input given to control a system is called the *control signal* or just the *control*.

Definition 4.2. A linear system with a realization (A, B, C, D) is an *observable system* if and only if the *observability matrix* of the system

$$\begin{pmatrix} C \\ CA \\ CA^2 \\ \cdots \\ CA^{\delta-1} \end{pmatrix}$$

has full rank δ.

If a system is observable any internal state can be determined from a finite sequence of outputs. This means that if δ rows of the observability matrix are linearly independent, then each state can be determined through linear combinations of the output variables y_t.

Controllability and observability are dual aspects of the same problem. These notions can be used to characterize minimal realizations.

Theorem 4.3. *A realization (A, B, C, D) of a linear system is minimal if and only if it is both controllable and observable.*

A module designed to estimate the state of a system from measurements of the outputs is called a *state observer* or simply an *observer* of the system. To control the system it is used a module called *controller*. Roughly speaking, a controller manipulates the inputs of the system to force it to have a certain behavior. There are different ways to control a system.

One of these control techniques is *optimal control*. Optimal control consists of obtaining a control law so that a system is optimal with respect to some criterion. This criterion is often modeled as the minimum of a cost functional, which usually has the form of the integral (or sum) over time of some function, plus a fixed cost that depends on the state in which the system starts (or ends up):

$$J = m(x_0) + \sum_{t=0}^{T} F(u, x, y, t)$$

A few suggested references on basic linear systems theory and on control theory are [AM71, AM89, Son98]

4.2 Convolutional Codes as Linear Systems

The relationship between linear systems theory and coding theory, in particular convolutional coding, has been studied for long. This relationship appears already in the early papers by Massey and Sain [MS67, MS68, SM69] and further research strengthened the links between both mathematical areas. As a consequence, different definitions, characterizations and results from linear systems theory have found their

successful counterpart in convolutional coding theory, see e.g. [RSY96, McE98, GL05].

This connection with systems theory has thus helped to better understand the properties of convolutional codes. In fact, the concepts of controllability and observability of linear systems can be translated into the context of convolutional codes [RSY96, GL05], leading to a correspondence between observability and non-catastrophicity.

Similarly, the convolutional decoding process can be interpreted in terms of systems theory in at least two ways [Ros]: as a tracking problem and as a filtering problem.

The interpretation of convolutional codes as linear systems provided also the tools to derive a few constructions of families of codes and decoding algorithms. Examples of it can be found in [RSY96, RS97, SGLR, Ros99, RS99, RY99, Ros01, GLRS03, HRS05, GL05].

A natural way to represent a convolutional code $\mathcal{C}$ over a finite field $\mathbb{F}_q$ as a discrete linear time-invariant system over $\mathbb{F}_q$ is to consider the information words as inputs of the system and the codewords, that result from the encoding process, as the output. Since the output is an element of $\mathbb{F}^n[z]$, the initial state of the system is zero and the final state after a finite time will be also zero. The system would be then defined by the equations

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t \\ y_t &= Cx_t + Du_t \\ x_0 &= 0 \end{aligned}$$

Here the vectors u_t represent the information words while the output vectors y_t represent the codewords. The transfer function of the system is

$$C(zId - A)^{-1}B + D = G(z^{-1})$$

being $G(z)$ is a generator matrix of the code.

Remark 4.4. Note that we have to change our usual criterion (Remark 1.3) and use a vertical representation of generator matrices in order to fit with the standard notation in linear systems. For the same reason, we will keep this representation criterion in the rest of the chapter.

This representation has a weak spot: A must be nilpotent (otherwise finite weight information words could be encoded into infinite weight information words). Then, convolutional codes would be represented just by a very restricted class of linear systems.

In contrast, there is a different approach, given in terms of duality between codes and certain objects from linear system theory, which allows convolutional codes to be represented by a more general set of linear systems. Let us identify an infinite

sequence $v_0, v_1, v_2, \ldots$ of vectors $v_i \in \mathbb{F}^n$ with the power series $v(z) = \sum v_i z^i \in \mathbb{F}^n[[z]]$, and consider the bilinear form

$$\begin{aligned} \mathbb{F}^n[[z]] \times \mathbb{F}^n[z] &\longrightarrow \mathbb{F} \\ (v(z), c(z)) &\longmapsto \sum_{i=0}^{\infty} \langle v_i, c_i \rangle \end{aligned}$$

This bilinear form induces a duality between submodules of $\mathbb{F}^n[z]$ and linear left shift invariant complete behaviors of $\mathbb{F}^n[[z]]$ as explained in [RSY96, Ros01].

This duality results in the following realization as a linear system of the submodule that defines the code. Given the $n \times k$-generator matrix $G(z)$ of a code with complexity δ, consider the partition

$$G(z) = \begin{pmatrix} P(z) \\ Q(z) \end{pmatrix} \tag{4.2}$$

where $P(z)$, $Q(z)$ have dimensions $n - k \times k$ and $k \times k$ respectively and up to reordering of rows we may suppose that $detQ(z)$ has degree δ. In the same way, given a codeword $\sum c_i z^i$, we consider the partition of each vector coefficient into

$$c_i = (y_i, u_i)^T.$$

Then we have then a controllable [Ros01, Th 5.3] state space representation

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t \\ y_t &= Cx_t + Du_t \\ x_0 &= 0 \end{aligned}$$

where both the inputs and the outputs of the system are part of the codewords. The transfer function of the linear system is

$$C(zId - A)^{-1}B + D = P(z)Q(z)^{-1}.$$

In addition, if this representation is observable, the encoder $G(z)$ is non-catastrophic.

4.2.1 Decoding of Convolutional Codes from a Systems Point of View

The task of decoding a received message which was encoded using a convolutional code, can be interpreted in terms of linear systems by making use of the previous system representation of the code in the following way: Given a received word $v(z) = (y'(z), u'(z))$ (partitioned as before), find the codeword that minimizes the error

$$\min_{c(z) \in \mathcal{C}} d(c(z), v(z)) = min \left(\sum_{k=0}^{T} d(y_k, y'_k) + d(u_k, u'_k) \right)$$

This process can be interpreted in terms of systems theory as a tracking problem or as a filtering problem [Ros]. In a tracking problem the system is controlled so that its output follows a certain sequence. This desired output of the system is called the *reference*. In the case of decoding, the decoder should track the received message by the most probable codeword sent. In a filtering problem the system is intended to remove an undesired component of a sequence and or to amplify the desired one. In our case, the decoder is requested to filter the noise sequence introduced by the channel in the transmitted encoded sequence.

Classically control problems are studied over the fields of real or complex numbers. To make use of this interpretation of decoding in terms of well-known control problems it is necessary to develop the same theory over finite fields. However, the tools applied to solve these problems make strong use of a quadratic norm, which in the case of finite fields cannot be defined. In vector spaces over finite fields there is no Euclidean metric, and the one given by the Hamming distance is not induced by a positive definite bilinear form. Therefore, the straightforward use of standard techniques from control theory is not possible.

On the other side, it is possible to use an ambient $\mathbb{R}$-vector space by considering a transmission over a Gaussian channel. In that case the received elements are points of a Euclidean space, and a Euclidean metric can be used to minimize the error. However, codewords are just $\mathbb{F}$-linear (not $\mathbb{R}$-linear) elements.

4.3 Tracking Problems over Finite Fields

We study a tracking problem over finite fields in order to apply it in a decoding method for convolutional codes.

Optimal control problems are a broad area of research in systems theory, and they have been considered from multiple points of view and for a wide range of system types. The solutions given make use of different strategies as for example Riccati equations [AM89].

One of the common characteristics of the different problems posed and the tools used to solve them is that they are considered over the fields of real or complex numbers, since the systems related to such problems deal with magnitudes given in terms of those numbers.

However there are problems that can be modeled in terms of one or several systems and with a setting in some finite field, for example, as seen before, the decoding of convolutional codes.

Our aim will be to state a well known class of optimal control problems in the context of finite fields. This will allow to use known results on this kind of problems over infinite fields and as a consequence to give a solution to the analogous problem posed over a finite field.

4.3.1 The Classical Tracking Problem

Let us consider a discrete linear system defined by the equations

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t \\ y_t &= Cx_t \\ x_{t_0} &= x_0 \end{aligned} \tag{4.3}$$

where A, B, C are constant matrices over $\mathbb{R}$ or $\mathbb{C}$ and x_t, u_t, y_t are vectors of lengths δ, k, $n-k$ respectively. Given a sequence $\{\tilde{y}_t\}_{t_0}^T$, a tracking problem consist of finding an input sequence $\{u_t\}_{t_0}^T$ that minimizes the cost functional

$$J(x_0, u(\cdot), T) = \sum_{t=t_0}^{T} [u_t^\top R u_t + (y_t - \tilde{y}_t)^\top Q (y_t - \tilde{y}_t)] \tag{4.4}$$

with R and Q positive and nonnegative definite matrices respectively of the appropriate dimensions.

The usual strategy used to solve this kind of problems is to apply a transformation to convert this problem in a standard regulator problem [AM71], solved recursively using a Ricatti equation, and to interpret the solution in terms of the original problem.

4.3.2 A Tracking Problem over a Finite Field

Let us consider the linear system with equations (4.3), where A, B, C are constant matrices over a finite field $\mathbb{F}$, and $x_t \in \mathbb{F}^\delta$, $u_t \in \mathbb{F}^k$, $y_t \in \mathbb{F}^{n-k}$. Let us consider the Hamming weight function w. In the problem over a finite field the Hamming metric substitutes the Euclidean one. Then, a tracking problem over a finite field $\mathbb{F}$ can be stated as

Problem 4.5. *Given a sequence $\{\tilde{y}_t\}_{t_0}^T$, find an input sequence $\{u_t\}_{t_0}^T$ that minimizes the cost functional*

$$J(x_0, u(\cdot), T) = \sum_{t=t_0}^{T} [w(Ru_t) + w(Q(y_t - \tilde{y}_t))] \tag{4.5}$$

where Q and R are constant square matrices of dimensions $n-k$, k respectively and having R maximum rank.

Remark 4.6. In the usual statement of the problem where an Euclidean metric is considered, the conditions for Q and R are to be nonnegative and positive definite, respectively. This means that for a nonzero vector u_t, respectively $y_t - \tilde{y}_t$, the corresponding term in the sum (4.4) will always sum up, respectively will not subtract. As the Hamming weight takes values in the nonnegative integers (and in particular zero only for the zero vector), to have the same notion in our case, we should ask that Ru_t is nonzero for every nonzero vector u_t (i.e. R has maximum rank) while no condition is required on $Q(y_t - \tilde{y}_t)$.

A brute force solution given by checking all the possible sequences is infeasible unless T is very small. To solve the problem efficiently, as in the classical case, we will make use of a well-known optimality principle:

Bellman's optimality principle [Bel57]. *An optimal trajectory has the property that at an intermediate point, no matter how it was reached, the rest of the trajectory must coincide with an optimal trajectory computed from this intermediate point as the initial point.*

This allows to reduce the overall minimization problem to a sequence of single-stage minimizations in the following way:

Note that the term $w(Ru_T)$ is the only summand of $J(x_0, u(\cdot), T)$ that depends entirely just on the last vector of the solution sequence, u_T, and it is therefore minimized with $u_T = 0$. Then, for every $t < T$ we have

$$J(x_0, u(\cdot), t+1) = J(x_0, u(\cdot), t) + w(Ru_t) + w(QCx_{t+1} - Q\tilde{y}_{t+1}).$$

By Bellman's optimality principle, if we assume known the sequence $\{u_0, \ldots, u_t\}$ that minimizes $J(x_0, u(\cdot), t)$, to get the minimum value of $J(x_0, u(\cdot), t+1)$ we only need to minimize $w(Ru_t) + w(QCx_{t+1} - Q\tilde{y}_{t+1})$. Furthermore, for $t = t_0$ the value of $J(x_0, u(\cdot), t_0) = w(QCx_0 - Q\tilde{y}_{t_0})$ is fixed, as x_0 and $\tilde{y}_{t_0}$ are known.

Then, Problem 4.5 is equivalent to a sequence of minimization problems for the expressions

$$w(Ru_t) + w(QCx_{t+1} - Q\tilde{y}_{t+1}).$$

By the equations (4.3) we have

$$QCx_{t+1} = QCAx_t + QCBu_t$$

and we can group the known vectors at time t as

$$z_t = QCAx_t - Q\tilde{y}_{t+1}.$$

Then, each single minimization problem can be formulated as

Problem 4.7. *Given a vector z and two matrices $Q' = QCB$, R, with R of maximum rank, find a vector u that minimizes the expression*

$$w(Ru) + w(Q'u + z)\ .$$

Let us consider the vector and the matrix

$$z' = (z_1, \ldots, z_{n-k}, 0, \overset{k}{\ldots}, 0)^\top \quad B_1 = \begin{pmatrix} Q' \\ R \end{pmatrix}$$

so that $z' + B_1 u = (z + Q'u, Ru)$, and $w(z' + B_1 u) = w(Ru) + w(Q'u + z)$. Then, the problem consists of finding the vector of minimum weight in the coset $\{z' + B_1 u\}_u$.

At this point we make use of techniques from coding theory.

Let us consider the block code generated by B_1, $\mathcal{C}_{B_1}$, and let us take the vector z', which with respect to this code has an error e, $z' = e + B_1 v$ (being e the vector of minimum Hamming weight that allows this kind of decomposition of z'). Then these cosets are the same one

$$\{z' + B_1 u\}_u = \{e + B_1(u + v)\}_{w=u+v}$$

and the vector with minimum weight in the coset is precisely e. Thus, finding the vector $z' + B_1 u$ of minimum weight is equivalent to decode z' as a codeword from $\mathcal{C}_{B_1}$. In particular, the optimal vector u is equal to $-v$, the inverse of the information word corresponding to z'.

Therefore, the optimal u_t at time t can be obtained from a decoding process following this iterative procedure:

- consider the solution input u_{t-1}, the corresponding state vector x_{t-1} and the reference vector $\tilde{y}_{t+1}$ to calculate $x_t = Ax_{t-1} + Bu_{t-1}$ and $z_t = QCAx_t - Q\tilde{y}_{t+1}$.
- decode $z'_t = (z_t, 0)$ as a codeword $c \in \mathcal{C}_{B_1}$.
- find the vector v such that $B_1 v = c$, and that therefore minimizes the sum $w(z + QBu) + w(Ru)$.
- update the solution $\{u_{t_0}, \ldots, u_{t-1}\}$ with $u_t = -v$, the cost functional J with $J + w(z_t + QBu_t) + w(Ru_t)$ and the time instant t with $t + 1$.

Remarks 4.8. 1.- The decoding scheme used to decode z' must give a solution even if the error weight is bigger than half the minimum distance of the code. This is possible as the vectors of the ambient space representing the codewords may not be uniformly distributed. If we consider the disjoint spheres centered in the codewords with the same maximum radius, which is the error-correcting capacity of the code, each point on a sphere is decoded as the codeword in the center of it. But these spheres may not contain every vector of the ambient space, and the one needed to be decoded could be out of all of these spheres.[1] In technical applications of codes it may be possible to detect an error that cannot be corrected. This problem is usually overcome by asking for a retransmission of the message with the hope of a better reception. But this is not possible in our case. A solution for this could be then to use some list decoding algorithm. If such an algorithm is not known for the code $\mathcal{C}_{B_1}$, a strategy to avoid multiple solutions, explained in subsection 4.3.4, can be used as an alternative.

2.- It may happen that z' is equally distant to two or more vectors of $\mathcal{C}_{B_1}$.[2] In coding theory this is a kind of detectable non-correctable error. In our case this would mean that the solution at time t is not unique, which is a situation that we should avoid. We address this topic in subsection 4.3.4.

[1] An exception are the so called *perfect codes*, characterized by the fact that the disjoint spheres centered in the codewords cover the whole ambient space.

[2] This also won't happen if we have a perfect code.

3.- If $R = Id_k$, then B_1 is a *systematic generator matrix* of the code $\mathcal{C}_{B_1}$, i.e., it "contains" the rows of a maximum size Id matrix. This means that the symbols of the word encoded, u, appear in certain positions of the codeword (those corresponding to the rows of the Id submatrix). Then, once e is calculated, this allows us to get the coordinates of u from the positions of $z' - e$ corresponding to the rows of Id_k, i.e. the last ones, without any further calculation.

To solve an optimal control problem, it is not only important to obtain the optimal input sequence but also to estimate the final value of the cost functional for that input. In our study of the tracking problem over finite fields we can only give a bound on the optimal value of the cost functional, which will depend on the *covering radius* of the code $\mathcal{C}_{B_1}$.

Definition 4.9. The *covering radius* $\rho_{\mathcal{C}}$ of a code $\mathcal{C}$ is the maximum distance from any vector of $\mathbb{F}^n$ to its nearest codeword.

The covering radius of a code is the smallest radius needed for the (no necessarily disjoint) spheres centered in the codewords to cover the whole ambient space.

Theorem 4.10. *Let ρ_{B_1} be the covering radius of the code $\mathcal{C}_{B_1}$ and $\tilde{u}$ the input sequence that minimizes the cost functional of the tracking problem 4.5, then*

$$J(x_0, \tilde{u}, T) \leq (T-1)\rho_{B_1} + w(QCx_0 - Q\tilde{y}_{t_0}).$$

Proof. The optimal cost given by the solution input $\tilde{u}$ is

$$J(x_0, \tilde{u}, T) = w(QCx_0 - Q\tilde{y}_{t_0}) + \sum_{t=t_0}^{T-1} [w(R\tilde{u}_t) + w(QCx_{t+1} - Q\tilde{y}_{t+1}))] + w(R\tilde{u}_T).$$

As seen before, $\tilde{u}_T = 0$, and hence the last term of the sum is 0 while $w(QCx_0 - Q\tilde{y}_{t_0})$ is known. The term added to J at each of the $T-1$ intermediate time instants depends on the error of the vector z'_t with respect to the code $\mathcal{C}_{B_1}$.

For any vector, including the vector z'_t decoded in every single-step, there is a codeword at a distance less or equal to ρ_{B_1}. Then the cost added to the functional at every time step can be bounded by

$$\min_u \{w(z' + B_1 u)\} = w(e) \leq \rho_{B_1}.$$

Thus, the cost functional after T time instants is bounded by

$$J(x_0, \tilde{u}, T) \leq (T-1)\rho_{B_1} + w(QCx_0 - Q\tilde{y}_{t_0}).$$

□

4.3.3 An Infinite Time Tracking Problem over a Finite Field

If an infinite time tracking problem with the same optimality criterion as in Problem 4.5 is considered, the cost functional (4.5) with $T = \infty$ does not make sense. The Hamming weight is zero only for the zero vector and has a positive value in any other case. As a consequence, unless from a certain time t the vectors $\{u_i\}_i$ are in the right kernel of R and the state vectors $\{x_i\}_i$ are in the right kernel of Q, the final value of J will be infinite.

However, we may be interested in finding the infinite sequence $\{u_i\}_i$ which is optimal in some other sense. For that, we reformulate the infinite time problem with a slightly different notion of optimality.

Problem 4.11. *Given the discrete time system defined by equations (4.3) and a sequence $\{\tilde{y}_t\}_{t=t_0}^{\infty}$, find the sequence $\{u_t\}_{t=t_0}^{\infty}$ so that the cost functional (4.5) is minimal for every $T < \infty$.*

To solve this problem we will use the so called *receding horizon method*. This method consists on considering a finite time tracking problem up to a time instant N just to take the first element of the solution sequence. Then we slide the initial and final instants one time unit to get a new finite time tracking problem. Formally, the receding horizon method consists on following these steps at every time instant t:

- consider the initial (known) state x_t.
- solve an N-step finite tracking problem, i.e., find $\{u_{t+i}\}_{i=0}^{N-1}$ which minimizes

$$J(x_t, u(\cdot), N) = w(QCx_t - Q\tilde{y}_t) + \sum_{i=0}^{N-1} [w(Ru_{t+i}) + w(QCx_{t+i+1} - Q\tilde{y}_{t+i+1})]$$

- update just the vector u_t, which is given as input to the system to get x_{t+1}.
- update the cost functional J with $J + w(z_t + QBu_t) + w(Ru_t)$ and the time instant t with $t + 1$.

By Bellman's optimality principle, the vector u_t obtained in every time instant t is the first vector of the sequence $\{\tilde{u}_t, \ldots, \tilde{u}_T\}$ that minimizes the function $J(x_t, u(\cdot), T)$ for any $T < \infty$. Then, the receding horizon method gives as result the solution to the infinite time tracking problem.

In the second step of the method, instead of proceeding recursively with N decoding steps we look for a direct way.

In the case $N = 1$ the solution for the finite tracking problem can be achieved by a decoding process with respect to the code generated by B_1.

In the general case, given x_t we want to find the sequence $\{u_t, \ldots, u_{t+N-1}\}$ that minimizes

$$\sum_{i=0}^{N-1} [w(QCx_{t+i+1} - Q\tilde{y}_{t+i+1}) + w(Ru_{t+i})] = w(z_N)$$

with

$$z_N = (QCx_{t+N} - Q\tilde{y}_{t+N}, Ru_{t+N-1}, QCx_{t+N-1} - Q\tilde{y}_{t+N-1}, Ru_{t+N-2}, \ldots, QCx_{t+1} - Q\tilde{y}_{t+1}, Ru_t).$$

We know that

$$x_{t+i} = A^i x_t + \sum_{r=0}^{i-1} A^{i-r-1} B u_{t+r}.$$

Then, the vector z_N can be written as

$$z_N = \widehat{w_{t,N}} + \widehat{B_N} u_{t,N}$$

with

$$\widehat{w_{t,N}} = \begin{pmatrix} QCA^N x_t \\ 0_k \\ QCA^{N-1} x_t \\ 0_k \\ \vdots \\ QCAx_t \\ 0_k \end{pmatrix} \qquad u_{t,N} = \begin{pmatrix} u_{t+N-1} \\ u_{t+N-2} \\ \vdots \\ u_t \end{pmatrix}$$

$$\widehat{B_N} = \begin{pmatrix} QCB & QCAB & \ldots & \ldots & \ldots & QCA^{N-1}B \\ R & 0 & & \ldots & & 0 \\ 0 & QCB & QCAB & \ldots & \ldots & QCA^{N-2}B \\ & R & 0 & & \ldots & 0 \\ \vdots & & \ddots & \ddots & & \\ \vdots & & & \ddots & \ddots & \\ & & & & & QCB \\ 0 & & \cdots & & 0 & R \end{pmatrix}.$$

Sorting the components of of z_N properly we have a vector $w_{t,N} + B_N u_{t,N}$ with the same weight of z_N, being $w_{t,N} = \Theta_N x_t$ and

$$\Theta_N = \begin{pmatrix} QCA^N \\ QCA^{N-1} \\ \vdots \\ QCA \\ 0 \\ \vdots_{Nk} \\ 0 \end{pmatrix}, \quad B_N = \begin{pmatrix} QCB & QCAB & \ldots & QCA^{N-1}B \\ 0 & QCB & \ldots & QCA^{N-2}B \\ & & \ddots & \\ 0 & 0 & & QCB \\ R & & & \\ & R & & \\ & & \ddots & \\ & & & R \end{pmatrix}.$$

We want to calculate the vector with the minimum weight in the coset

$$\{w_{t,N} + B_N u_{t,N}\}_{u_{t,N} \in \mathbb{F}^{Nk}} \ .$$

The solution is obtained by decoding the vector $w_{t,N}$ with respect to the code generated by B_N.

The dimensions of the elements in this process are N times bigger than those in the single-step decoding. This seems to suggest that there is no gain in this method in comparison to decoding N smaller vectors. However, the vectors $w_{t,N}$ to be decoded are of a very particular form, which will make computations significantly simpler.

Theorem 4.12. *Decoding $w_{t,N}$ in the second step of the algorithm for the infinite time tracking problem with a length N receding horizon is as complex as decoding with respect to a code of length $\delta + Nk$.*

Proof. Let us consider the exact sequence

$$0 \longrightarrow \mathbb{F}^{Nk} \xrightarrow{B_N} \mathbb{F}^{Nn} \xrightarrow{H_N} \mathbb{F}^{N(n-k)} \longrightarrow 0$$

representing the block code generated by B_N. $\mathbb{F}^{N(n-k)}$ is the set of syndromes, i. e., the set of errors that the code allows to correct. However we are only interested in the errors contained in the vectors of the form $w_{t,N} = \Theta_N x_t$.

The matrix Θ_N defines an *injective* map

$$\begin{array}{c} \phi : \mathbb{F}^{\delta} \longrightarrow \mathbb{F}^{Nn} \\ x_t \longmapsto w_{t,N} \end{array}.$$

As ϕ and H_N are linear, its composition $\phi' = H_N \circ \phi$ is also linear, and we can define a morphism

$$\phi' : \mathbb{F}^{\delta} \longrightarrow \mathbb{F}^{N(n-k)}$$

with image of dimension at most δ. Then, the counterimage of $Im\phi'$ by H_N has dimension at most $\delta + Nk$. We have therefore a sequence

$$0 \longrightarrow \mathbb{F}^{Nk} \longrightarrow \mathbb{F}^{\delta+Nk} \xrightarrow{H_N} \mathbb{F}^{\delta} \longrightarrow 0$$

which represents the decoding process of the elements of $Im\phi$. □

4.3.4 Multiple Solutions

When studying an optimal control problem, it is also important to determine under which conditions is the solution unique. The solution discussed before for a tracking problem over a finite field can be non-unique in two cases.

The first possibility is that for a certain time instant t, there are two vectors $u_t \neq u'_t$ such that $z'_t + B_1 u_t = z'_t + B_1 u'_t$ with z'_t as defined before, i. e., $(QCx_{t+1} - Q\tilde{y}_{t+1}, Ru_t) = (QCx'_{t+1} - Q\tilde{y}_{t+1}, Ru'_t)$, with $x_{t+1} = Ax_t + Bu_t$ and $x'_{t+1} = Ax_t + Bu'_t$ which are obviously decoded to the same codeword with respect to $\mathcal{C}_{B_1}$. Then there are two different solutions u_t, u'_t which give the same minimal

value of the cost functional. However, this is not possible, as the square matrix R is assumed to have maximum rank, and in particular $u_t \neq u'_t \Rightarrow Ru_t \neq Ru'_t$.

The second case of multiple solutions occurs when in a single decoding step the vector $y_t = QCAx_t - Q\tilde{y}_{t+1}$ that has to be decoded is equidistant to two or more codewords of $\mathcal{C}_{B_1}$. There is no reason to prefer one solution to the others. To avoid this possibility we solve a $(N = 2)$-step finite horizon tracking problem as in the second step of the receding horizon solution for the infinite case, and then update the solution vectors u_t and u_{t+1}. Of course, in doing so it could again happen that we get two or more optimal solutions, in which case we would do the same for $N = 3$ and so on. We stop if we finally get a unique solution or if for two different solutions u, u' the N-th vectors are equal, $u_{t+N-1} = u'_{t+N-1}$, which means that from the time instant $t + N$ on, these two solutions will be equally valid.

However, the probability of this to happen is rather small and, as it is proven later, it decreases as we consider bigger values of N.

To formalize this intuition, let us first recall a geometrical interpretation of the decoding process.

If a code of length n has minimum distance d, it is a well-known fact that the spheres centered in the codewords with radius $\lfloor \frac{d-1}{2} \rfloor$ are disjoint. Any vector from the ambient space contained in one of these spheres can be then decoded as its unique nearest codeword, the center of the sphere. Each of these spheres contain a total of $\sum_{i=0}^{\lfloor \frac{d-1}{2} \rfloor} \binom{n}{i} (q-1)^i$ vectors, being q the size of the field where the code is defined.

Then, the *density* of an $[n, k, d]$ code $\mathcal{C}$ defined over $\mathbb{F}_q$ is

$$\delta_{\mathcal{C}} = \frac{\sum_{i=0}^{\lfloor \frac{d-1}{2} \rfloor} \binom{n}{i} (q-1)^i}{q^{n-k}}$$

which can be interpreted as the proportion of the ambient space covered by the biggest disjoint spheres centered in its codewords. The probability that a randomly chosen vector is out of all these spheres is $P_o^{\mathcal{C}} = 1 - \delta_{\mathcal{C}}$.

Remark 4.13. Note the similarity of the expression for the density of a code and that of the *Hamming bound* on the minimum distance of block codes, which is based in the same geometrical interpretation of codewords. That's the reason for the alternative name as *sphere packing bound*.

The subset of vectors that we are interested in decoding with respect to $\mathcal{C}_{B_1}$ is $\mathcal{H} = \mathbb{F}^{n-k} \times 0^k$. The vectors from $\mathcal{H}$ that are actually decoded are those inside one of the spheres of radius $t = \lfloor \frac{d-1}{2} \rfloor$, with d the minimum distance of $\mathcal{C}_{B_1}$, and center

one of the codewords from $\mathcal{C}_{B_1}$ which are at a distance $\leq t$ from $\mathcal{H}$. Taking into account the generator matrix of the code $\mathcal{C}_{B_1}$

$$B_1 = \begin{pmatrix} QCB \\ R \end{pmatrix}$$

where R has maximum rank, the codewords at a distance $w \leq t$ from $\mathcal{H}$ are those with w non-zeros in their last k components, and the number of them is $\binom{k}{w}(q-1)^w$. On the other hand, if a codeword c is at a distance w from $\mathcal{H}$, the vectors from $\mathcal{H}$ that belong to the sphere centered in c with radius t are those which differ from c in at most $t-w$ of their $n-k$ first components, and the number of them is $\sum_{j=0}^{t-w}\binom{n-k}{j}(q-1)^j$.

Then, the density of the intersection of $\mathcal{H}$ with the spheres of radius t centered in the codewords of $\mathcal{C}_{B_1}$ is

$$\delta_{\mathcal{H}} = \frac{\sum_{i=0}^{t}\binom{k}{i}(q-1)^i \sum_{j=0}^{t-i}\binom{n-k}{j}(q-1)^j}{q^{n-k}} \ .$$

The probability that a randomly chosen vector from $\mathcal{H}$ is decoded to a word from $\mathcal{C}_{B_1}$ is precisely $\delta_{\mathcal{H}}$. We will denote $P_o^{\mathcal{H}} = 1 - \delta_{\mathcal{H}}$.

In the case of N-step decoding, the vectors to be decoded are those from the plane $\mathcal{H}_N = \mathbb{F}^{N(n-k)} \times 0^{Nk}$. Let d_N be the minimum distance of the code $\mathcal{C}_{B_N}$ and denote $t_N = \lfloor \frac{d_N - 1}{2} \rfloor$ the number of errors that this code can correct. Then, the probability that a randomly chosen vector from $\mathcal{H}_N$ is decoded with respect to $\mathcal{C}_{B_N}$ is

$$\delta_{\mathcal{H}_N} = \frac{\sum_{i=0}^{t_N}\binom{Nk}{i}(q-1)^i \sum_{j=0}^{t_N-i}\binom{N(n-k)}{j}(q-1)^j}{q^{N(n-k)}}$$

and analogously we denote $P_o^{\mathcal{H}_N} = 1 - \delta_{\mathcal{H}_N}$.

Note that for all N, the zero vector belongs to $\mathcal{C}_{B_N} \cap \mathcal{H}_N$ for every code generated by a matrix of the form B_N, and hence the sphere centered in 0 intersects $\mathcal{H}_N$. Then, for every code $\mathcal{C}_{B_N}$ and for every N, $P_o^{\mathcal{H}_N} < 1$.

Before addressing the next result, and for the sake of simplicity, we fix the following notation

$$E_{k,t} = \sum_{i=0}^{t}\binom{k}{i}(q-1)^i \ .$$

On the other side, we will impose that the system is controllable and hence every state vector x_t can be reached. We will assume therefore that the occurrence of every vector $z' = (QCAx_t - Q\tilde{y}_{t+1}, 0) \in \mathcal{H}$ (respectively $\omega_{t,N} \in \mathcal{H}_N$) is equiprobable.

Theorem 4.14. *Let us consider a tracking problem over a finite field for a controllable system. The probability of the need to solve a N-step tracking problem to avoid multiple solutions is asymptotically 0.*

The probability of M different solutions in a N-step tracking problem which won't be discriminated with longer step problems can be upper-bounded by

$$\prod_{i=1}^{N-1} P_o^{\mathcal{H}_i} \frac{\delta_{\mathcal{H}}^M}{E_{k,t}}.$$

Proof. To have multiple solutions in a i-step tracking problem means that the vector to be decoded, which belongs to $\mathcal{H}_i$, cannot be uniquely associated with a codeword of $\mathcal{C}_{B_i}$, i. e., it is equidistant to two or more codewords. As we have seen, any vector with the same minimum distance to two or more codewords is not contained in any of the spheres of radius $\lfloor \frac{d_{\mathcal{C}_{B_i}}-1}{2} \rfloor$ and center a codeword. In particular if a vector from $\mathcal{H}_i$ is equidistant to two or more codewords, then it is not contained in the intersection of those spheres with $\mathcal{H}_i$, and as seen before the probability of this to happen to a random vector from $\mathcal{H}_i$ is $P_o^{\mathcal{H}_i}$.

In addition, the need to solve a N-step problem means that for each of the previous i-step tracking problems, $i < N$, there were more than one solution. Considering that the vector to be decoded in each problem could be any one from the corresponding space $\mathcal{H}_i$ with uniform probability, the probability that we need to solve an N-problem can be upper-bounded by

$$P_o^{\mathcal{H}_1} \cdot \ldots \cdot P_o^{\mathcal{H}_{N-1}} = \prod_{i=1}^{N-1} P_o^{\mathcal{H}_i} \longrightarrow 0. \tag{4.6}$$

Let us consider now the case of M different optimal solutions of a N-step tracking problem that cannot be discriminated with longer step problems, i. e., all with the same solution vector u_{t+N-1}.

By Bellman's optimality principle, for any optimal solution $(u_t, \ldots, u_{t+N-1})$ of a N-step finite horizon tracking problem each u_i is an optimal solution for a single step problem.

The condition for two optimal solutions u, u' to have $u_T = u'_T$, $T = t + N - 1$, (and therefore $u_{T+j} = u'_{T+j}\ \forall j$) is that in the T-th step two vectors from $\mathcal{H}$, $(QCAx_T - Q\tilde{y}_{T+1}, 0)$ and $(QCAx'_T - Q\tilde{y}_{T+1}, 0)$, are decoded as the same codeword and their errors have the same weight. Let us examine the probability of this to happen assuming that the occurrence of every vector in $\mathcal{H}$ is equiprobable.

Given two vectors $v, v' \in \mathcal{H}$, if v is decoded to a codeword $c \in \mathcal{C}_{B_1}$, the probability that v' is also decoded to c and both errors have the same weight depends on

- the probability that c is at a distance $i \leq t$ from $\mathcal{H}$

$$P(d(c, \mathcal{H}) = i) = \frac{\binom{k}{i}(q-1)^i}{\sum_{r=0}^{t} \binom{k}{r}(q-1)^r}$$

- the probability that the error vector $v - c$ has weight $e = i + j$ $(i \leq e \leq t)$ provided that $d(c, \mathcal{H}) = i$

$$P(w(v-c) = i + j \,|\, d(c,\mathcal{H}) = i) = \frac{\binom{n-k}{j}(q-1)^j}{\sum_{s=0}^{t-i}\binom{n-k}{s}(q-1)^s}$$

- the probability that the Hamming weight $w(v'-c) = e$ provided that $d(c, \mathcal{H}) = i$ and $w(v - c) = e$

$$P(w(v'-c) = i + j \,|\, d(c,\mathcal{H}) = i, w(v-c) = e) = \frac{\binom{n-k}{j}(q-1)^j}{q^{n-k}}$$

Considering all the possibilities for the possible values of i and j we have that the probability that a vector $v' \in \mathcal{H}$ is decoded to the same codeword as a decoded vector v and its error has the same weight as that of v is

$$\frac{\sum_{i=0}^{t}\binom{k}{i}(q-1)^i\sum_{j=0}^{t-i}\left(\binom{n-k}{j}(q-1)^j\right)^2\left(\sum_{s=0}^{t-i}\binom{n-k}{s}(q-1)^s\right)^{-1}}{q^{n-k}\sum_{r=0}^{t}\binom{k}{r}(q-1)^r}. \tag{4.7}$$

Let us note that

$$\sum_{j=0}^{t-i}\left(\binom{n-k}{j}(q-1)^j\right)^2 \leq \left(\sum_{j=0}^{t-i}\binom{n-k}{j}(q-1)^j\right)^2$$

(actually equality only holds for $i = t$), then, as $t > 0$, (4.7) is upper-bounded by

$$\frac{\sum_{i=0}^{t}\binom{k}{i}(q-1)^i\sum_{j=0}^{t-i}\binom{n-k}{j}(q-1)^j}{E_{k,t}\; q^{n-k}} = \frac{\delta_{\mathcal{H}}}{E_{k,t}}.$$

Two vectors $v, v' \in \mathcal{H}$ are decoded to the same codeword of $\mathcal{C}_{B_1}$ and have errors of the same weight when one is decoded, which occurs with probability $\delta_{\mathcal{H}}$, and the other is decoded to the same codeword as the first one and its error has the same weight, which occurs with probability (4.7). Then, the probability of this to happen is the product of $\delta_{\mathcal{H}}$ times (4.7) and this probability is upper bounded by

$$\frac{\delta_{\mathcal{H}}^2}{E_{k,t}}.$$

In general, given a vector $v \in \mathcal{H}$ decoded to $c \in \mathcal{C}_{B_1}$, the probability that for $l-1$ vectors $v'_1, \ldots, v'_{l-1} \in \mathcal{H}$, $w(v'_1 - c) = \ldots = w(v'_{l-1} - c) = e = i + j$ provided that $d(c, \mathcal{H}) = i$ and $w(v - c) = e$ is

$$\left(\frac{\binom{n-k}{j}(q-1)^j}{q^{n-k}} \right)^{l-1} .$$

Then, the probability that $v'_1, \ldots, v'_{l-1}$ are decoded to the same codeword as v and all the errors have the same weight is

$$\frac{\displaystyle\sum_{i=0}^{t} \binom{k}{i}(q-1)^i \sum_{j=0}^{t-i} \left(\binom{n-k}{j}(q-1)^j \right)^l \left(\sum_{s=0}^{t-i} \binom{n-k}{s}(q-1)^s \right)^{-1}}{q^{(l-1)(n-k)} \displaystyle\sum_{r=0}^{t} \binom{k}{r}(q-1)^r} , \tag{4.8}$$

and taking into account that

$$\sum_{j=0}^{t-i} \left(\binom{n-k}{j}(q-1)^j \right)^l \leq \left(\sum_{j=0}^{t-i} \binom{n-k}{j}(q-1)^j \right)^l$$

$$\tbinom{k}{i}(q-1)^i \leq (\tbinom{k}{i}(q-1)^i)^{l-1}$$

$$\sum_{i=0}^{t} \left(\binom{k}{i}(q-1)^i \right)^{l-1} \left(\sum_{j=0}^{t-i} \binom{n-k}{j}(q-1)^j \right)^{l-1} \leq \left(\sum_{i=0}^{t} \binom{k}{i}(q-1)^i \sum_{j=0}^{t-i} \binom{n-k}{j}(q-1)^j \right)^{l-1}$$

then (4.8) is upper-bounded by

$$\frac{\left(\displaystyle\sum_{i=0}^{t} \binom{k}{i}(q-1)^i \sum_{j=0}^{t-i} \binom{n-k}{j}(q-1)^j \right)^{l-1}}{E_{k,t}\, q^{(l-1)(n-k)}} = \frac{\delta_{\mathcal{H}}^{l-1}}{E_{k,t}} .$$

Thus, the probability that l vectors from $\mathcal{H}$ are decoded to the same codeword from $\mathcal{C}_{B_1}$ and all the errors have the same weight is the product of (4.8) times $\delta_{\mathcal{H}}$ and this probability is upper bounded by

$$\frac{\delta_{\mathcal{H}}^{l}}{E_{k,t}}.$$

To sum up, if we have M different optimal solutions of a N-step problem, all with the same u_{t+N-1} (which will result in M solutions of the global problem all of them equally valid) two things have happened:

– All the previous multiple step problems had more than one solution, which as seen in the first part of the proof can happen with probability bounded by

$$\prod_{i=1}^{N-1} P_o^{\mathcal{H}_i}.$$

– The M solutions "joined" to have the same solution vector u_{t+N-1}, the probability of which is bounded by

$$\frac{\delta_{\mathcal{H}}^M}{E_{k,t}}.$$

Then the probability of M different optimal solutions which differ in $N-1$ solution vectors u_i can be upper-bounded by

$$\prod_{i=1}^{N-1} P_o^{\mathcal{H}_i} \frac{\delta_{\mathcal{H}}^M}{E_{k,t}}.$$

□

Remark 4.15. As mentioned in Remark 4.8.1 the decoding method used to solve the finite tracking problem must always give a solution, even when we have a detectable non-correctable error. For that, a possibility is to use the same strategy proposed before to avoid multiple solutions, i. e., to consider N-step decoding till we get a solution. Note that in the previous theorem, to bound the probability of multiple solutions after N-step decoding, we have just assumed that the vector to be decoded has an error of weight $> t_N$. The same argument is valid to determine the probability of not getting a solution after N-step decoding by using a t_N-error correcting algorithm and hence this probability is also bounded by $\prod_{i=1}^{N-1} P_o^{\mathcal{H}_i}$.

4.4 Convolutional Decoding as a Tracking Problem

Decoding is together with the construction of efficient codes the main task in coding theory, and as that one, a non-trivial objective. As mentioned before, the decoding of convolutional codes can be considered as a tracking problem. Applying the solution for the tracking problem over finite fields studied in the previous section we can give a decoding algorithm for general convolutional codes.

Let us consider the partition (4.2) of a generator matrix $G(z)$ of a convolutional code, and the corresponding one on each codeword $c(z) = (y(z), u(z))$. We have a controllable state space representation of the code

$$\begin{array}{l} x_{t+1} = Ax_t + Bu_t \\ y_t = Cx_t + Du_t \\ x_0 = 0 \end{array} \quad . \tag{4.9}$$

Decoding a received sequence $\{\tilde{c_t} = (\tilde{y}_t, \tilde{u}_t)\}_t$ can be thought as a tracking problem with cost functional

$$\begin{aligned} J(x_0, c(\cdot), T) &= \sum_{t=0}^{T} [w(u_t - \tilde{u}_t) + w(y_t - \tilde{y}_t)] \\ &= J(x_0, c(\cdot), T-1) + w(u_T - \tilde{u}_T) + w(y_T - \tilde{y}_T). \end{aligned}$$

Remark 4.16. In this case the grouping of the terms differs slightly from the one in the previous section as in the system considered here, y_t depends also on u_t.

Then, by applying Bellman's optimality principle, we just need to minimize the term

$$w(u_t - \tilde{u}_t) + w(y_t - \tilde{y}_t)$$

for each $t \leq T$.

We have that

$$y_t = Cx_t + Du_t$$

and Cx_t, as well as $\tilde{u}_t$, $\tilde{y}_t$ are known at time t. Therefore, calling

$$\begin{aligned} z_1 &= Cx_t - \tilde{y}_t \\ z_2 &= -\tilde{u}_t \end{aligned}$$

the problem to be solved at each time step t can be generally stated as

Problem 4.17. *Given two vectors z_1, z_2 and a square matrix D, find the vector u that minimizes*

$$w(Du + z_1) + w(u + z_2) = w(z' + B_1 u)$$

for

$$z' = (z_1, z_2)^\top \quad B_1 = \begin{pmatrix} D \\ Id \end{pmatrix}.$$

As we have seen before, this problem can be solved by decoding the vector z' with respect to the block code generated by B_1.

In practice, in order to save time, the decoding of a convolutional codeword can start before the whole word is received, which is equivalent to consider decoding as an infinite process.

Then, we state the corresponding infinite time tracking problem which we solve by applying the receding horizon method in a similar way as in the previous section.

Problem 4.18. *Given the discrete time system defined by equations (4.9) and a sequence $\{(\tilde{y}_t, \tilde{u}_t)\}_{t=0}^{\infty}$, find the sequence $\{u_t\}_{t=0}^{\infty}$ so that the cost functional*

$$J(x_0, c(\cdot), T) = \sum_{t=0}^{T} [w(u_t - \tilde{u}_t) + w(y_t - \tilde{y}_t)]$$

is minimal for every $T < \infty$.

The steps to be followed at every time instant t are

- take the initial (known) state x_t.
- solve an N-step finite horizon tracking problem, i.e., find the sequence $\{u_{t+i}\}_{i=0}^{N-1}$ which minimizes

$$J(x_t, c, N) = \sum_{i=0}^{N-1} [w(y_{t+i} - \tilde{y}_{t+i}) + w(u_{t+i} - \tilde{u}_{t+i})]$$

- update the solution input with $\{u_t, \ldots, u_{t+L-1}\}$ and use it to update the solution output with $\{y_t, \ldots, y_{t+L-1}\}$ and to calculate x_{t+L}. L will depend on how many steps the code C_{B_N} ensures are decoded without errors.
- update the time instant t with $t + L$.

Step 2 represents the main problem to be solved. As before, we propose a direct solution instead of an iterative one.

If we join the vectors with weight the cost functional that we want to minimize we get a vector z_N such that $J(x_t, c, N) = w(z_N)$, with

$$z_N = \widehat{w_{t,N}} + \widehat{B_N} u_{t,N}$$

where

$$\widehat{w_{t,N}} = \begin{pmatrix} CA^{N-1}x_t - \tilde{y}_{t+N-1} \\ -\tilde{u}_{t+N-1} \\ CA^{N-2}x_t - \tilde{y}_{t+N-2} \\ -\tilde{u}_{t+N-2} \\ \vdots \\ CAx_t - \tilde{y}_{t+1} \\ -\tilde{u}_{t+1} \\ Cx_t - \tilde{y}_t \\ -\tilde{u}_t \end{pmatrix} \quad \widehat{B_N} = \begin{pmatrix} D & CB & CAB & \ldots & CA^{N-2}B \\ Id & 0 & 0 & 0 & 0 \\ 0 & D & CB & \ldots & CA^{N-3}B \\ 0 & Id & 0 & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ & & 0 & & D \\ & & & & Id \end{pmatrix}.$$

$$u_{t,N} = (u_{t+N-1}, u_{t+N-2}, \ldots, u_{t+1}, u_t)$$

Resorting its components we have a vector $w_{t,N} + B_N u_{t,N}$ with the same weight

as z_N, being

$$w_{t,N} = \begin{pmatrix} CA^{N-1} \\ \vdots \\ C \\ 0 \\ \vdots \\ 0 \end{pmatrix} x_t - \begin{pmatrix} \tilde{y}_{t+N-1} \\ \vdots \\ \tilde{y}_t \\ \tilde{u}_{t+N-1} \\ \vdots \\ \tilde{u}_t \end{pmatrix}$$

$$B_N = \begin{pmatrix} D & CB & CAB & \dots & CA^{N-2}B \\ 0 & D & CB & \dots & CA^{N-3}B \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & CB \\ 0 & \dots & \dots & 0 & D \\ & & Id_{Nk} & & \end{pmatrix} = \begin{pmatrix} D & H_0 & H_1 & \dots & H_{N-2} \\ 0 & D & H_0 & \dots & H_{N-3} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & H_0 \\ 0 & \dots & \dots & 0 & D \\ & & Id_{Nk} & & \end{pmatrix},$$

with $H_i = CA^iB$.

Then, the problem will be solved by decoding the vector $w_{t,N}$ with respect to the code generated by B_N.

The number of steps to update, L, depends on how many errors the method corrects in every N-step. The precise connection is given by the following theorem.

Theorem 4.19. *The decoding scheme can correct $\lfloor \frac{d'}{2} \rfloor$ errors, $d' \geq d_N - 1$, up to an admissible decoding error, if every codeword from $c \in \mathcal{C}_{B_N}$ of weight $w(c) \leq d'$ has zeros in the components $c_{(N-L)(n-k)+1}, \dots, c_{N(n-k)}$ and $c_{Nn-Lk+1}, \dots, c_{Nn}$.*

Proof. Note that the generator matrix B_N is systematic, and the check matrix of the code is well known to be

$$H_N = \begin{pmatrix} & & -D & -H_0 & -H_1 & \dots & -H_{N-2} \\ & & 0 & -D & -H_0 & \dots & -H_{N-3} \\ & Id_{N(n-k)} & \vdots & \ddots & \ddots & \ddots & \vdots \\ & & \vdots & & \ddots & \ddots & -H_0 \\ & & 0 & \dots & \dots & 0 & -D \end{pmatrix}.$$

The minimum distance of the code $\mathcal{C}_{B_N}$, d_N, is precisely the minimum number of linearly dependent columns of H [MS77], as the coefficients of one such linear dependency would be the components of a codeword from $\mathcal{C}_{B_N}$. This bounds the number of errors that can be corrected in an N-step.

Note however that after every N-step decoding, the method updates the partial solution just with $u_t, \dots, u_{t+L-1}$, i. e., decoding errors that occur in the components corresponding to $u_{t+L}, \dots, u_{t+N}$ (and hence also in those corresponding to

$y_{t+L}, \ldots, y_{t+N}$) are admissible. The set of components corresponding to these vectors is $\alpha = \{1, \ldots, (N-L)(n-k), N(n-k)+1, \ldots, Nn-Lk\}$. Let us denote its complementary by $\bar{\alpha}$.

An admissible error, with support in α, is also a codeword: an admissible error is the difference between the codeword "before" the error occurred and the codeword resulting from the decoding process, and by linearity the difference of two codewords is also a codeword.

Let us assume that the code doesn't allow to correct error vectors of weight $t' = \lfloor \frac{d'}{2} \rfloor$ up to an admissible decoding error. Then, there exists a vector v such that for two different codewords $c, c' \in \mathcal{C}_{B_N}$ it can be written as $v = c + e$ and $v = c' + e'$ with $w(e) = w(e') = t'$. As decoding up to an admissible error is not possible we have that $c_{\bar{\alpha}} \neq c'_{\bar{\alpha}}$, i. e., $e_{\bar{\alpha}} \neq e'_{\bar{\alpha}}$. Then, $c + e = c' + e'$ and by linearity $c - c' = e' - e = c''$ is a codeword from $\mathcal{C}_{B_N}$ with weight $w(c'') \leq w(e) + w(e') = 2t' \leq d'$ and such that $c''_{\bar{\alpha}} \neq 0$, which contradicts the assumption of the theorem. □

Example 4.20. Let us consider the convolutional code over $\mathbb{F}_5$ generated by the matrix

$$\begin{pmatrix} 1 & 4+z \\ 3 & z \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} P(z) \\ Q(z) \end{pmatrix}$$

which as we have seen before can be regarded as the linear system described by the equations

$$\begin{aligned} x_{t+1} &= (1,2)\, u_t \\ y_t &= 4x_t + (1,3)\, u_t \\ x_0 &= 0 \end{aligned} \quad ,$$

i. e., it has a minimal realization $(A, B, C, D) = ((0), (1,2), (4), (1,3))$.

Let us fix the values $N = 2$, $L = 1$. Then our algorithm will work with the received vectors v_t, v_{t-1} at each time instant t and it should correctly decode at least v_{t-1}. In each step a vector must be decoded with respect to the code that has as check matrix

$$H_N = \begin{pmatrix} 1 & 0 & 1 & 3 & 4 & 3 \\ 0 & 1 & 0 & 0 & 1 & 3 \end{pmatrix}$$

where the columns 1, 3, 4 correspond to the coordinates of v_t, i. e., to admissible errors. We observe that although the minimum distance of the the code is 2, there is no codeword of weight ≤ 2 with support in the positions 2, 5, 6 (the ones that have to be correctly decoded). This means that our scheme will be able to correct one error and produce the correct v_{t-1} in each decoding step.

The convolutional code has parameters $[n, k, \delta, d_{free}] = [3, 2, 1, 3]$. Hence our algorithm takes full advantage of the error correcting capacities of the code.

Conclusions

In this work convolutional codes have been studied by addressing some of the main problems concerning them, such as the study of their mathematical structure, the construction of new families of codes, and their decoding. For that, the well known interpretations of these codes in algebraic and linear systems theoretic terms have been applied.

In Chapter 2 a classification of convolutional codes has been proposed with the aim of improving their algebraic understanding. This classification makes it possible to associate to each convolutional code a point of a certain grassmannian variety. On the other side, the sets of points that represent convolutional codes have been determined. The insight obtained from this classification has been used to derive some new bounds on the free distance of convolutional codes and to propose a construction of convolutional codes from block codes with optimal minimum distance.

In Chapter 3 the family of convolutional Goppa codes defined over elliptic curves is presented. This construction results in a remarkable variety of convolutional codes with optimal free distance. Further, also a prominent number of strongly MDS convolutional codes have been obtained. Likewise, the analogous construction to the so called generalized AG codes has been developed for convolutional codes. This form of defining new codes has proved to be a fruitful way to obtain codes with optimal free distance.

In Chapter 4 the interpretation of decoding of convolutional codes as a tracking problem for linear discrete-time systems allowed to develop a decoding algorithm. For that, the tracking problem has been stated over a finite field and a solution for it, different from the classical ones, has been proposed. The resulting algorithm allows to make full use of the correcting capacity of the convolutional code.

Appendix A

Estudio de los Códigos Convolucionales. Clasificación, Nuevas Familias y Decodificación

El presente apéndice contiene un resumen en español con los principales resultados y conclusiones de la Tesis Doctoral, cumpliendo así con el Reglamento de Tercer Ciclo y Doctorado de la Universidad de Salamanca en lo relativo a la redacción de la tesis en otro idioma.

A.1 Introducción a la Teoría de Códigos

A.1.1 Motivación y Desarrollo de la Teoría de Códigos

La teoría de códigos es una parte fundamental de la teoría de la información. Aborda el problema de evitar errores en la trasmisión o almacenamiento de datos debidos a interferencias en el medio físico utilizado. Para ello distintas técnicas matemáticas han permitido desarrollar una gran variedad de códigos con diferentes características en función del contexto en que han de ser aplicados. Como ejemplos están la codificación del contenido de los CDs y DVDs, de la información transmitida a través del espacio, de la televisión y la radio digital o de la voz y datos en comunicaciones inalámbricas; pero también los códigos de barras, de las tarjetas de crédito o el código ISBN de los libros.

Matemáticamente, el proceso de codificación se representa mediante una aplicación sobre el conjunto de mensajes que valora en el conjunto de elementos o palabras del código. Esta aplicación añade al mensaje una redundancia, que permitirá detectar y corregir los posibles errores producidos. Si la señal recibida no se corresponde con ninguna de las palabras del código, se entiende que se ha producido un error en la transmisión, que se supondrá tiene la magnitud mínima posible. El

proceso de decodificación consiste en asignar a la señal recibida la palabra del código con más probabilidad de haber sido enviada.

El origen de la teoría de códigos está en el trabajo de Claude Shannon y su artículo "A Mathematical Theory of Communication" [Sha48] en el que prueba la existencia de códigos que permiten codificar un mensaje con la mínima redundancia para transmitirlo sin errores. Sin embargo, éste no contenía una construcción explícita de tales códigos. Desde entonces diversas técnicas matemáticas han sido utilizadas con este fin, dando como resultado diferentes familias de códigos, tales como los códigos de Hamming, de Reed-Muller y Reed-Solomon, los códigos BCH y los códigos de Goppa, éstos últimos construidos con métodos algebro-geométricos. Además, se generalizó la idea de la codificación, aplicando ésta a las distintas unidades de un mensaje con dependencia unas de otras. Se originaron así los códigos convolucionales. La búsqueda de códigos óptimos se acompañó de un estudio sus propiedades intrínsecas, lo que permitió poder utilizar dichas características para desarrollar diferentes algoritmos de decodificación (que junto con la construcción de códigos óptimos es uno de los grandes problemas de la teoría de códigos), tales como el de Berlekamp-Massey, el de Viterbi o el de Sudan.

Con el redescubrimiento en 1996 de los códigos LDPC (Low-Density Parity-Check), y la construcción de códigos de esta familia con prestaciones cada vez más próximas a la capacidad de transmisión del canal, el objetivo de encontrar los códigos predichos por Shannon se consideró alcanzado.

Sin embargo, el camino recorrido ha abierto multitud de problemas interesantes, y por otro lado, la naturaleza de los distintos códigos no está aún completamente explicada. Es por esto que la teoría de códigos se mantiene como un área de investigación muy activa.

A.1.2 Códigos de Bloques

Formalmente, un *código lineal de bloques de longitud n y dimensión k* es la imagen de un morfismo inyectivo $g : \mathbb{F}_q^k \longrightarrow \mathbb{F}_q^n$ que representa la codificación, donde $\mathbb{F}_q$ es un cuerpo finito, denominado *alfabeto*, con cuyos símbolos se expresarán los mensajes y las palabras del código. Una matriz generadora del código es cualquier matriz que representa al morfismo g.

La capacidad correctora de un código viene dada por su distancia mínima, que se define en función del llamado peso de Hamming. El peso de Hamming de un vector v es el número de sus componentes no nulas, $w(v) = \sharp\{i|v_i \neq 0\}$. La distancia de Hamming entre dos vectores es el número de componentes en que difieren, $d(v, v') = \sharp\{i|v_i \neq v'_i\} = \sharp\{i|v_i - v'_i \neq 0\}$. Las distancia mínima de un código es la mínima distancia de Hamming entre dos palabras del código cualesquiera, $d(\mathcal{C}) = \min_{x,y \in \mathcal{C}}\{d(x, y)\} = \min_{c \in \mathcal{C}}\{w(c)\}$. Es bien conocido que el número de errores que el código $\mathcal{C}$ permite corregir es $\left\lfloor \frac{d(\mathcal{C})-1}{2} \right\rfloor$.

Sin embargo no es fácil calcular la distancia mínima de un código, y a menudo

hay que conformarse con cotas. Una de las principales es la cota de Singleton, $d \leq n - k + 1$. Los códigos cuya distancia mínima alcanza dicha cota se denominan *MDS* (Maximum Distance Separable) y tienen propiedades que permiten tanto estudiarlos mejor como desarrollar algoritmos de decodificación particulares para estos códigos.

Una *matriz de control* de un código $\mathcal{C}$ es una matriz H^T de modo que $cH^T = 0$ $\forall\, c \in \mathcal{C}$. Esta matriz permite una caracterización alternativa de los vectores que pertenecen al código. Dado un vector $v \in \mathbb{F}_q^n$ se denomina síndrome de v a vH^T. En particular todos los elementos del conjunto $e+\mathcal{C}$ tienen el mismo síndrome, pues por linealidad

$$(e + c)H^T = eH^T + cH^T = eH^T.$$

Así, la decodificación mediante síndromes, que asigna a cada síndrome el vector de mínimo peso en el conjunto $e + \mathcal{C}$ correspondiente, permitiría decodificar cualquier código. Sin embargo este método sólo es práctico para códigos con baja distancia mínima.

Se define el *código dual* de $\mathcal{C}$ mediante

$$\mathcal{C}^\perp = \{x \in \mathbb{F}_q^n | x \cdot c = 0 \,\forall\, c \in \mathcal{C}\}.$$

Por definición, las matrices de control de $\mathcal{C}$ son las transpuestas de las matrices generadoras de $\mathcal{C}^\perp$.

Referencias básicas en teoría de códigos clásica y códigos de bloques son [MS77, McE77, vL82].

A.1.3 Introducción a los Códigos Convolucionales

Los códigos convolucionales fueron inventados por Elias en 1955. La idea básica es codificar las distintas unidades del mensaje de modo dependiente. Es decir, dada una sucesión de palabras de información $\{u_0, \ldots, u_l\}$, que se puede representar como el polinomio vectorial $u(z) = u_0 + u_1 z + \ldots + u_l z^l$, la codificación por bloques viene dada por $c(z) = u(z)G$, con G la matriz generadora. La idea de Elias fue considerar una matriz generadora polinómica, $G(z)$, de modo que el vector codificado c_i depende no sólo del vector de información u_i sino también de $u_{i-1}, u_{i-2}, \ldots$. Como consecuencia, un *código convolucional de longitud n y dimensión k* se define como un submódulo de rango k de $\mathbb{F}_q^n[z]$. Del mismo modo que para códigos de bloques, una matriz cuyas filas generan el submódulo correspondiente al código convolucional se denomina *matriz generadora*.

El peso de Hamming de un vector polinómico se define como la suma del número de coeficientes no nulos en todas sus componentes. La distancia entre dos vectores polinómicos es igual al peso de Hamming de su vector diferencia. Así, el concepto análogo a la distancia mínima es la *distancia libre* de un código convolucional, que se define como la mínima distancia entre dos cualesquiera de sus vectores polinómicos, o equivalentemente, el mínimo peso de Hamming de cualquier palabra del código.

Además, dado un código convolucional se puede definir su *j-ésima distancia por columnas*, d_j^c, que es el peso mínimo de cualquier palabra del código truncada en el grado j. También se define la *j-ésima distancia por filas*, d_j^r, como el peso mínimo de cualquier palabra del código resultante de codificar un vector de información de grado j. Se tiene [JZ99]

$$d_0^c \leq d_1^c \leq \ldots \leq d_\infty^c = d_{free} = d_\infty^r \leq \ldots \leq d_1^r \leq d_0^r.$$

Del mismo modo que en el caso de códigos de bloques, se define la *matriz de control* de un código convolucional $\mathcal{C}$ como una matriz H^T de modo que $cH^T = 0 \; \forall \, c \in \mathcal{C}$ y el *código dual* de $\mathcal{C}$ mediante

$$\mathcal{C}^\perp = \{x \in \mathbb{F}_q^n[z] | x \cdot c = 0 \; \forall \, c \in \mathcal{C}\}.$$

En el caso de códigos convolucionales, no todas las matrices generadoras son igualmente válidas, e incluso las hay que son inadecuadas. Para discriminar unas de otras consideraremos una serie de nociones que no aparecen en los códigos de bloques.

Una matriz generadora se dice que es *básica* si el máximo grado de sus menores $k \times k$ es mínimo entre todas las matrices generadoras del código. Una matriz es *reducida* (o minimal) si los grados de sus filas no se pueden reducir mediante operaciones elementales de filas. Una matriz generadora es *canónica* si es básica y reducida.

Dos extensas caracterizaciones de matrices básicas y reducidas pueden encontrarse en [McE98, Theorem A.1] y [McE98, Theorem A.2]. En particular, es importante asegurar que una matriz generadora no pueda codificar un vector con componentes no polinómicas en un vector polinómico. En ese caso, una palabra del código se podría corresponder con un vector de información de peso infinito, y si esa palabra se ha decodificado erróneamente, se tendría un error de magnitud infinita en la información recibida. Por ello, este tipo de matrices se denominan *catastróficas*. Toda matriz básica es no catastrófica.

Se denomina *grado i-ésimo por filas* de una matriz al máximo grado de los polinomios en la fila i-ésima de la matriz. Los grados por filas de una matriz canónica se denominan *índices de Forney*, y son, salvo el orden, invariantes del código. Su suma, denotada por δ se llama *grado* o *complejidad* del código, y el máximo índice de Forney es la *memoria* del código. Los índices de Forney pueden interpretarse en términos de geometría algebraica o teoría de sistemas. El grado del código da una medida de la influencia de los anteriores vectores de información en la codificación del presente y la memoria indica el número de vectores anteriores de que depende la codificación en cada paso.

Con estos parámetros, es posible dar cotas a la distancia libre de los códigos convolucionales, que en general es incluso más compleja de calcular que la distancia mínima de los códigos de bloques. Estas cotas son generalizaciones de las cotas conocidas para los códigos de bloques. En particular se tiene la cota de Singleton

generalizada,

$$d_{free} \leq S(n,k,\delta) = (n-k)\left(\left\lfloor \frac{\delta}{k} \right\rfloor + 1\right) + \delta + 1.$$

Los códigos cuya distancia libre alcanza esta cota se denominan *códigos convolucionales MDS*.

Otra particularidad de los códigos convolucionales es que son equivalentes a sistemas lineales discretos. Esta relación, expuesta en el capítulo 4, permite utilizar elementos de la teoría de sistemas para definir nuevas familias de códigos y diseñar algoritmos de decodificación.

Algunas referencias clásicas donde se desarrolla la teoría de códigos convolucionales son [Pir88, McE98, JZ99].

A.1.4 Aplicaciones de la Teoría de Códigos

Aunque la motivación inicial de la teoría de códigos era fundamentalmente práctica, a medida que ésta se fue desarrollando se vio que podía ser de gran utilidad para resolver problemas en otras áreas de las matemáticas.

Una de las primeras aplicaciones fue en el desarrollo de sistemas tolerantes a fallos: cuando una función booleana es implementada mediante un circuito físico que puede producir errores, la introducción de un código permite detectar y corregir esos errores de modo que la computación se haga correctamente. Otra aplicación natural se encuentra en la extracción de claves criptográficas biométricas. Éstas dependen de propiedades únicas de un individuo (huella digital, exploración de retina,...), pero por sus características están sujetas a errores de lectura o debidos a causas externas. Sin embargo una característica indispensable de las claves criptográficas es su exactitud. La corrección de este tipo de errores mediante códigos posibilita el uso de dichas claves.

Además, el interés hacia la geometría algebraica surgido a raíz de los códigos de Goppa abrió la puerta a la aplicación de la teoría de códigos también en esta rama. La relación entre ambas áreas estimuló la investigación respecto al número de puntos racionales de una curva en función de su género, y el uso de herramientas de teoría de códigos se mostró muy útil para investigar la proporción entre el número de puntos racionales y el género de curvas definidas sobre cuerpos finitos pequeños.

Muchas de las aplicaciones de los códigos a la criptografía y la teoría de la complejidad se basan en dos clases de códigos: los códigos localmente controlables y los códigos localmente decodificables. Los primeros permiten corregir muchos errores y tienen asociado un algoritmo probabilístico de detección de errores en tiempo sublineal. Este algoritmo muestrea cada vector recibido en un pequeño número de componentes con las que es capaz de decidir con alta probabilidad si existe o no error. Los algoritmos localmente decodificables tienen asociado un algoritmo probabilístico de corrección de errores en tiempo sublineal. Este algoritmo toma un conjunto aleatorio de componentes del vector recibido y da como resultado una componente del vector decodificado que con alta probabilidad es correcta.

Entre las aplicaciones de estos códigos están el criptoanálisis de algunos sistemas de cifrado, la generación de series pseudoaleatorias de números, algoritmos de firma digital, la prueba de la corrección de demostraciones, la verificación de software, la acotación de la complejidad de algoritmos, la medida de la complejidad de comunicación,...

A continuación presentamos algunos ejemplos que ilustran la utilización de códigos para resolver problemas en otras disciplinas.

Criptosistema de Clave Pública de McEliece

Un criptosistema de clave pública permite a cualquier usuario encriptar un mensaje, mediante una clave pública, mientras que sólo un usuario autorizado, mediante una clave privada, puede desencriptarlo. Ambas claves están fuertemente relacionadas, pero si bien la clave pública se obtiene fácilmente de la clave privada, es computacionalmente imposible obtener la clave privada a partir de la clave pública.

Como se ha visto antes, decodificar un un código de bloques genérico mediante síndromes es imposible en la práctica cuando la distancia mínima (y por tanto el número de errores que se pueden corregir) es muy alto. Sin embargo, hay códigos particulares que permiten corregir un gran número de errores mediante un algoritmo que aprovecha singularidades debidas a la estructura matemática con la que se han construido. La idea de McEliece fue usar uno de estos códigos y disimularlo. De este modo, un usuario autorizado sabría qué algoritmo utilizar, mientras que un usuario cualquiera sólo podría utilizar un algoritmo genérico que en la práctica no daría resultado.

Formalmente, la clave privada consiste en tres matrices (S, G, P), donde G es la matriz generadora del código, S es una matriz invertible aleatoria y P es una matriz de permutación. La clave pública consiste en el par $(\tilde{G}, t)$, donde $\tilde{G} = SGP$ y t es el número de errores que pueden añadirse en la encriptación (y que lógicamente debe estar por debajo de la capacidad correctora de G). Así, no puede establecerse relación entre $\tilde{G}$ y G. Tomando $t \geq 50$ y $n \geq 2^{10}$, este criptosistema es seguro.

Predicados Fuertes para Permutaciones Unidireccionales

Una *función unidireccional* $f : \mathbb{F}_2^k \to \mathbb{F}_2^k$ es una función fácil de calcular pero cuya inversa no puede calcularse en tiempo polinómico. Una función $P : \mathbb{F}_2^k \to \mathbb{F}_2^m$ es un *predicado fuerte* para una función f si computacionalmente no puede calcularse $P(x)$ a partir de $f(x)$. En general no es posible obtener predicados fuertes para permutaciones unidireccionales, por lo que se utiliza una cadena aleatoria auxiliar r, de modo que se dice que P es un *predicado fuerte* para f si computacionalmente no se puede calcular $P(x, r)$ a partir de $f(x)$ y r.

Dado un código con un algoritmo de decodificación por listas asociado, se puede obtener un predicado fuerte para cualquier función unidireccional del siguiente modo. Sea el código definido por $G : \mathbb{F}_2^k \to \mathbb{F}_2^n$ y $f : \mathbb{F}_2^k \to \mathbb{F}_2^k$ una permutación unidireccional. Sea $P : \mathbb{F}_2^k \times [n] \to \mathbb{F}_2$ un predicado con $P(x, j) = (Gx)_j$, y la función

$f_0 : \mathbb{F}_2^k \times [n] \to \mathbb{F}_2^k \times [n]$ dada por $f_0(x, j) = (f(x), j)$. P es un predicado fuerte para f_0. Si no fuera así, podría obtenerse el número suficiente de coordenadas $(Gx)_i$ para aplicar la decodificación por listas a Gx, y de forma sencilla obtener x, lo que contradiría la unidireccionalidad de $f(x)$.

Dos casos particulares del uso de los predicados fuertes son los generadores de números pseudoaleatorios, que a partir de una pequeña secuencia de números realmente aleatorios permiten obtener una larga cadena de números pseudoaleatorios, y la encriptación probabilística de clave pública, donde el uso de predicados fuertes permite añadir seguridad al sistema.

Secretos Compartidos

El problema a resolver consiste en dividir una información en n partes de modo que ésta sólo pueda ser recuperada cuando un número suficientemente alto de dichas partes se combinan.

Para ello se representa la información a dividir como un elemento $s \in \mathbb{F}_q$ y se considera un código de tipo $[n+1, k, d]$. La información se divide tomando cualquier palabra del código $(c_0, c_1, \ldots, c_n)$ de modo que $c_0 = s$ y se asigna a cada parte P_i el elemento c_i. Para recuperar la información, el código ha de ser capaz de decodificar un vector donde las partes ausentes se consideran componentes borradas. Para ello, el número mínimo de partes necesarias será $n - d + 2$.

Este método permite además obtener la información cuando entre las partes hay traidores, es decir, que comparten una información falsa. La correcta decodificación permite obtener la información verdadera y detectar a los traidores.

A.1.5 Objetivos de la Tesis

Este trabajo consiste en el estudio de los códigos convolucionales desde distintos puntos de vista. Se consideran algunos de los principales problemas de la teoría de códigos, tales como el análisis de los códigos convolucionales mediante la clasificación en función de sus parámetros, la construcción de nuevas familias de códigos como los códigos de Goppa convolucionales sobre curvas elípticas y otros códigos algebro-geométricos, y el desarrollo de un algoritmo de decodificación de códigos convolucionales. Para ello contempla la interpretación de los códigos convolucionales en términos de geometría algebraica y de teoría de sistemas lineales.

A.2 Clasificación de Códigos Convolucionales

A.2.1 Introducción

Para facilitar el estudio de los códigos convolucionales es deseable poder representarlos mediante puntos de una variedad. En el capítulo 2 se presenta una clasificación de códigos convolucionales que permite asociar cada código con un punto de una grassmanniana y determinar la subvariedad formada por los puntos asociados a códigos. Esta clasificación permite conocer mejor la estructura de los códigos convolucionales

y en particular puede utilizarse para dar cotas a la distancia libre y definir algunos códigos con buenos parámetros.

A.2.2 Preliminares Algebro-Geométricos

La clasificación de códigos convolucionales propuesta se hace en términos algebro-geométricos, haciendo uso de diversos elementos que permiten representar primero matrices generadoras, y después códigos, como puntos de una variedad grassmanniana.

Para ello se van a considerar una serie de haces coherentes, que son aquellos que localmente coinciden con el haz de localizaciones de un módulo, definidos sobre la recta proyectiva, $\mathbb{P}^1_K$. Nótese que se tiene una equivalencia de categorías entre los haces coherentes sobre $\mathbb{P}^1_K$ y las clases de $K[x_0, x_1]$-módulos graduados finito generados modulo la relación de equivalencia $M \sim M' \Leftrightarrow \exists d \in \mathbb{N}$ tal que $\underset{n\geq d}{\oplus} M_n \simeq \underset{n\geq d}{\oplus} M'_n$. Además se sabe que los espacios de secciones sobre $\mathbb{P}^1$ y sobre $\mathbb{A}^1$ del haz $\mathcal{O}(r)$ están generados por las bases $\{x_0^r, x_0^{r-1}x_1, \dots x_1^r\}$ y $\{1, z, \dots, z^r\}$ respectivamente, que se fijan como bases estándar.

Para determinar la grassmanniana en que se representa cada código se utilizará el polinomio de Hilbert de un haz coherente $\mathcal{F}$, definido por $P_{\mathcal{F}}(r) = \sum_{i=0}^{dimX} (-1)^i dim H^i(X, \mathcal{F}(r))$ para $r \in \mathbb{Z}$.

En general, se considerará una familia de haces $\mathcal{F}_S$ definida sobre un S-esquema $X \times S \longrightarrow S$, de modo que dicha familia es coherente y plana sobre S. Recordemos que, dado un morfismo de esquemas $f : T \longrightarrow S$, un haz cuasicoherente de $\mathcal{O}_T$-módulos $\mathcal{F}$ es *plano sobre* S si para todo $t \in T$ $\mathcal{F}_t$ es un $\mathcal{O}_{S,f(t)}$-módulo plano. En particular esto significa que el polinomio de Hilbert de las fibras $\mathcal{F}_{f(t)}$ es independiente del punto base.

El esquema que va a permitir interpretar los códigos como puntos de una variedad es el *esquema cociente*, que se define en función de su functor de puntos del siguiente modo.

Un haz cociente de $\mathcal{F}_S$, siendo $\mathcal{F}$ un haz coherente sobre un esquema X y S otro esquema, es un haz coherente sobre $X \times S$ tal que $q : \mathcal{F} \otimes \mathcal{O}_S \longrightarrow Q$ es epiyectiva. Se tiene la relación de equivalencia $Q \sim Q'$ si hay un isomorfismo $f : Q \longrightarrow Q'$ tal que $q' = f \circ q$.

Así, el *functor cociente* $\mathcal{Q}uot^P_{\mathcal{F}}$ definido por el haz $\mathcal{F}$ y el polinomio racional $P(z) \in \mathbb{Q}[z]$ es el functor que a cada K-esquema S le asocia el conjunto de clases de equivalencia de haces S-planos cocientes de $\mathcal{F}_S$ con polinomio de Hilbert $P(z)$.

Es un resultado clásico [Gro60] que el functor $\mathcal{Q}uot^P_{\mathcal{F}}$ es representable por un esquema para el que existe una inclusión como subesquema de las grassmannianas $Grass(H^0(\mathcal{F}(r)), P(r))$ para todo r tal que $Q(r), \mathcal{F}_S(r), Kerq(r)$ son haces generados por sus secciones globales. Este esquema se define como el *esquema cociente*.

A.2.3 Forma Canónica de Kronecker-Hermite

Como se explica en el capítulo 1, no todas las matrices generadoras de códigos convolucionales son igualmente útiles, y algunas, como las matrices catastróficas es deseable evitarlas. Por ello es necesario poder obtener una matriz de las llamadas *canónicas* a partir de cualquier matriz generadora del código. Éstas matrices tienen la característica de tener los mínimos grados por filas y el menor grado máximo de sus menores.

La *forma canónica de Kronecker-Hermite* y la *forma canónica modificada de Kronecker-Hermite* presentadas en [FH01] permiten reducir aún más los grados de las componentes de una matriz polinómica, y al mismo tiempo ofrecen un método para obtener matrices canónicas (ya que éstas dos formas lo son en el sentido de la teoría de códigos convolucionales) a partir de una matriz reducida.

La matriz en forma canónica de Kronecker-Hermite se caracteriza por tener un conjunto único de índices distinguidos $1 \leq j_1 < j_2 < \ldots j_k \leq n$ de modo que para todo $i \leq k$

1.- g_{ij_i} es un polinomio mónico de grado máximo en su fila

2.- el resto de polinomios en la misma columna que g_{ij_i} tienen grado menor que éste

3.- el resto de polinomios en la misma fila y a la izquierda de g_{ij_i} tienen grado menor o igual que éste mientras que los que están a la derecha tienen grado estrictamente menor.

La matriz canónica modificada de Kronecker-Hermite se obtiene de la anterior mediante permutación de filas, de modo que los grados por filas están ordenados mientras que para los índices distinguidos sólo se requiere que estén ordenados los correspondientes a las filas con el mismo grado.

En [FH01] se demuestra además que *toda matriz polinómica reducida puede transformarse de modo único mediante el producto por una matriz unimodular a forma canónica modificada de Kronecker-Hermite*, y se muestra cómo. Además, se prueba fácilmente que *en cada clase de matrices polinómicas equivalentes módulo el producto por matrices unimodulares, sólo hay una en forma canónica modifica de Kronecker-Hermite*. En particular, cada código convolucional sólo tiene una matriz generadora en esta forma.

A.2.4 Clasificación de Códigos Convolucionales

El objetivo es obtener información sobre la estructura algebraica de los códigos convolucionales. Para ello se pretende representarlos como puntos de una variedad, lo que en nuestra clasificación haremos identificando primero matrices generadoras y después códigos convolucionales con ciertos haces cocientes y consiguientemente con puntos de una variedad grassmanniana.

Códigos Convolucionales como Haces Cocientes

El primer paso en nuestra clasificación es verificar que *a todo código convolucional se le puede asociar un punto en un esquema cociente dado por una clase de haces sin torsión en la recta afín.*

Para ello, dado el morfismo de submódulos $\overline{\phi_G} : K[z]^k \hookrightarrow K[z]^n$ definido por una matriz generadora básica y reducida por columnas (es decir cuyos términos de grado máximo en cada columna forman una matriz constante de rango máximo), consideramos el correspondiente morfismo de módulos graduados que a su vez determina un morfismo de haces coherentes sobre $\mathbb{P}_1$, de donde se tiene la sucesión exacta

$$0 \longrightarrow \mathcal{O}_{\mathbb{P}^1}^k \xrightarrow{\widetilde{\phi_G}} \bigoplus_{i=1}^{n} \mathcal{O}_{\mathbb{P}^1}(m_i) \longrightarrow Q \to 0$$

donde $\{m_i\}_1^n$ son los grados por columnas de la matriz generadora. Además, todas las matrices generadoras del código, básicas y reducidas por columnas, con los mismos grados por columnas definen del modo anterior haces equivalentes a Q. Por tanto, se le puede asociar al código la clase de haces cocientes de $\bigoplus_{i=1}^{n} \mathcal{O}_{\mathbb{P}^1}(m_i)$ representada por Q. Además, por ser básicas las matrices elegidas, los haces de la clase representada por Q no tienen torsión en la recta afín.

Por otro lado, el polinomio de Hilbert de Q es $P(r) = (n-k)(r+1) + \theta$ con $\theta = gr(Q) = \sum m_i$.

Así pues, el código convolucional está representado por un punto del esquema cociente $Quot^{P(r)}_{\oplus \mathcal{O}_{\mathbb{P}}(m_i)}$.

El siguiente paso es dar la representación del código como punto de una grassmanniana, para ello se prueba que *un código convolucional de tipo* $[n,k]$ *que tiene una matriz generadora básica y reducida por columnas con grados* $\{m_i\}_1^n$ *está representado por un punto de la grassmanniana* $Grass(H^0(\bigoplus_{i=1}^{n} \mathcal{O}_{\mathbb{P}^1}(m_i)), n+\theta-k)$*, que además puede pensarse como código de bloques de tipo* $[n+\theta, k, d_0^r]$.

De entre todas las grassmannianas que contienen como subesquema a $Quot^{P(r)}_{\oplus \mathcal{O}_{\mathbb{P}}(m_i)}$ tomaremos la más pequeña, que vendrá determinada por el mínimo r_0 tal que $\mathcal{O}_{\mathbb{P}^1}^k(r_0)$, $\bigoplus_{i=1}^{n} \mathcal{O}_{\mathbb{P}^1}(m_i + r_0)$ y $Q(r_0)$ están generados por sus secciones globales, para lo que bastará tomar $r_0 = 0$. Así, se tiene la inclusión

$$Quot(\bigoplus_{i=1}^{n} \mathcal{O}_{\mathbb{P}^1}(m_i), P(r)) \xhookrightarrow{\phi_{\{m_i\}_1^n}} Grass(H^0(\bigoplus_{i=1}^{n} \mathcal{O}_{\mathbb{P}^1}(m_i)), P(0))$$

de modo que la imagen del haz cociente definido por

$$0 \to \mathcal{O}_{\mathbb{P}^1}^k \xrightarrow{\widetilde{\phi_G}} \oplus \mathcal{O}_{\mathbb{P}^1}(m_i) \to Q \to 0$$

es el subespacio obtenido al tomar secciones globales en la sucesión anterior,

$$0 \to H^0(\mathbb{P}^1, \mathcal{O}_{\mathbb{P}^1}^k) \xrightarrow{\psi_G} H^0(\mathbb{P}^1, \oplus \mathcal{O}_{\mathbb{P}^1}(m_i)) \to H^0(\mathbb{P}^1, Q) \to 0.$$

Y puesto que las equivalencias de matrices generadoras básicas y reducidas por columnas y de haces coinciden, el código convolucional está representado por un único punto de esta grassmanniana.

Por otro lado, considerando las bases estándar que hemos fijado, si la matriz del morfismo $\overline{\phi_G}$, $(g^{(ij)}(z))$, donde $g^{(ij)}(z) = \sum g_k^{(ij)} z^k$, es una matriz polinómica generadora del código entonces el subespacio $Im\psi_G$ está generado por las filas de la matriz

$$BG = \begin{pmatrix} g_0^{(11)} & g_1^{(11)} & \dots & g_{m_1}^{(11)} & \dots & \dots & g_0^{(1n)} & g_1^{(1n)} & \dots & g_{m_n}^{(1n)} \\ \vdots & & & & & & & & & \vdots \\ g_0^{(k1)} & g_1^{(k1)} & \dots & g_{m_1}^{(k1)} & \dots & \dots & g_0^{(kn)} & g_1^{(kn)} & \dots & g_{m_n}^{(kn)} \end{pmatrix}$$

que como fácilmente se comprueba genera un código de bloques de tipo $[n+\theta, k, d_0^r]$, siendo d_0^r la 0-ésima distancia por filas del código convolucional.

Una vez definida la representación de un código convolucional como punto de una grassmanniana se quiere determinar el conjunto de puntos de la grassmanniana que representan a algún código convolucional. A ese respecto se demuestra que *$Im\phi_{\{m_i\}_1^n}$ son precisamente los puntos de la grassmanniana que como subespacios están generados por una matriz BG de modo que si se particiona en bloques*

$$BG = (BG^{(1)}|BG^{(2)}|\dots|BG^{(n)}),$$

cada uno con $m_i + 1$ columnas, $BG^{(i)} = (g_0^{(i)}, \dots, g_{m_j}^{(i)})$, la matriz

$$\widehat{BG} = \left(\begin{array}{cccccc|cc|cccccc} g_0^{(1)} & g_1^{(1)} & g_2^{(1)} & \dots & g_{m_1}^{(1)} & 0 & \dots & \dots & g_0^{(n)} & g_1^{(n)} & g_2^{(n)} & \dots & g_{m_n}^{(n)} & 0 \\ 0 & g_0^{(1)} & g_1^{(1)} & \dots & g_{m_1-1}^{(1)} & g_{m_1}^{(1)} & \dots & \dots & 0 & g_0^{(n)} & g_1^{(n)} & \dots & g_{m_n-1}^{(n)} & g_{m_n}^{(n)} \end{array} \right)$$

tiene rango máximo. En particular $Im\phi_{\{m_i\}_1^n}$ es un abierto.

En efecto, para cada morfismo ψ_G se tiene un diagrama conmutativo

$$\begin{array}{ccc} H^0(\mathcal{O}_{\mathbb{P}}^k) \otimes \mathcal{O}_{\mathbb{P}} & \xrightarrow{\psi_G \otimes Id_{\mathcal{O}_{\mathbb{P}}}} & H^0(\bigoplus_{i=1}^n \mathcal{O}_{\mathbb{P}}(m_i)) \otimes \mathcal{O}_{\mathbb{P}} \\ \downarrow \wr & & \downarrow \psi \\ \mathcal{O}_{\mathbb{P}}^k & \xrightarrow{\widetilde{\phi_G}} & \bigoplus_{i=1}^n \mathcal{O}_{\mathbb{P}}(m_i) \end{array}$$

de modo que ψ_G representa un código convolucional si y sólo si $\widetilde{\phi_G}$ es inyectivo.

Por otro lado se tiene

$$0 \to \bigoplus_{i=1}^n \mathcal{O}_{\mathbb{P}}(-1)^{m_i} \xrightarrow{\phi} H^0(\bigoplus_{i=1}^n \mathcal{O}_{\mathbb{P}}(m_i)) \otimes \mathcal{O}_{\mathbb{P}} \xrightarrow{\psi} \bigoplus_{i=1}^n \mathcal{O}_{\mathbb{P}}(m_i) \to 0\ ,$$

y por tanto basta comprobar si $Im\phi \cap Im(\psi_G \otimes Id_{\mathcal{O}_{\mathbb{P}}}) = (0)$. Se pueden identificar las respectivas imágenes en términos matriciales, y como consecuencia se tiene que

$Im\phi \cap Im(\psi_G \otimes Id_{\mathcal{O}_{\mathbb{P}}}) = \{u_1x_1 + u_0x_0 \,|\, (u_0^\top, u_1^\top)\widehat{BG} = 0\}$.

Finalmente, de entre los puntos en $Im\phi_{\{m_i\}_1^n}$ sólo representan a un código convolucional aquellos relacionados del modo visto anteriormente con matrices polinómicas básicas y reducidas por columnas. En el Lemma 2.24 y el Teorema 2.25, respectivamente, se prueba que estas dos condiciones definen sendos abiertos de la grassmanniana. Como consecuencia, *el conjunto de puntos de la grassmanniana* $Grass(k, H^0(\bigoplus_{i=1}^n \mathcal{O}_{\mathbb{P}^1}(m_i)))$ *que representan códigos convolucionales es un abierto, determinado por la intersección de dichos abiertos e* $Im\phi_{\{m_i\}_1^n}$.

Clasificación de Haces versus Clasificación de Códigos

La construcción anterior clasifica códigos convolucionales a través de las matrices generadoras que tienen un conjunto particular de índices por columnas. Sin embargo, un mismo código puede estar generado por matrices con diferentes índices por columnas, y por tanto estar representado como un punto en diferentes esquemas, y consecuentemente en diferentes grassmannianas. Esto es debido a que la equivalencia de haces y la equivalencia de matrices polinómicas son diferentes. La primera se da mediante isomorfismos representados por una matriz constante mientras que la segunda se da en términos del producto por matrices unimodulares.

Así pues, podría haber haces no equivalentes que estén asociados al mismo código, es decir, haces sobre $\mathbb{P}^1$ que coincidan en la parte afín (y por tanto determinen el mismo submódulo) pero no globalmente.

Nuestra intención es clasificar cada código mediante un único punto de una variedad, y por tanto mediante una única clase de equivalencia de haces, por lo que será muy importante tener este hecho en cuenta.

Clasificación de Códigos Convolucionales

Para clasificar los códigos convolucionales en función de haces cocientes se considerará la inclusión de los haces correspondientes a las diferentes matrices generadoras de un mismo código en un único haz, que será el que represente al código.

Formalmente, *dado un código convolucional de tipo* $[n, k, \delta]$ *y memoria* ν_k*, se tiene una única sucesión de enteros* $\{n_i\}_{i=1}^n$*, llamados* mínimos índices por columnas*, menores o iguales a* ν_k*, de modo que el código está representado por un único punto de la grassmanniana* $Grass(k(\nu_k + 1) - \delta, H^0(\bigoplus_{i=1}^n \mathcal{O}_{\mathbb{P}^1}(n_i)))$.

En efecto. Para ver cuál es el haz que contiene a los haces asociados a las diferentes matrices generadoras de un código del modo descrito anteriormente, se consideran dos de éstas matrices G_1, G_2, que por ser equivalentes se pueden escribir como $G_1 = UG_2$, siendo U unimodular, y que definen sendos haces cocientes de $\bigoplus_{i=1}^n \mathcal{O}(m_i')$ y $\bigoplus_{i=1}^n \mathcal{O}(m_i'')$ respectivamente. Por otro lado, la matriz U define un morfismo $\mathcal{O}^k \hookrightarrow \bigoplus_{i=1}^k \mathcal{O}(\nu_k - \nu_i) \hookrightarrow \mathcal{O}(N)^k$ con $N = \nu_k - \nu_1$, lo que permite definir la in-

clusión $\overset{n}{\underset{i=1}{\oplus}}\mathcal{O}(m_i'') \hookrightarrow \overset{n}{\underset{i=1}{\oplus}}\mathcal{O}(m_i'+N)$ (y análogamente $\overset{n}{\underset{i=1}{\oplus}}\mathcal{O}(m_i') \hookrightarrow \overset{n}{\underset{i=1}{\oplus}}\mathcal{O}(m_i''+N)$). En particular, se pueden definir unos índices $\{n_i\}_{i=1}^n$, que se definen como los *mínimos índices por columnas* del código, de modo que $Im\phi' \subset \overset{n}{\underset{i=1}{\oplus}}\mathcal{O}(n_i)$ para todo ϕ' asociado a una matriz generadora del código y tal que $n_i \leq n_i'$ para cualquier otra colección de índices $\{n_i'\}$ que verifique esta condición.

Así pues, para todo morfismo de haces ϕ_1 que represente a una matriz canónica del código se tiene

$$\begin{array}{ccc} \mathcal{O}^k & \xrightarrow{\quad\phi_1\quad} & \overset{n}{\underset{i=1}{\oplus}}\mathcal{O}(m_i') \\ \downarrow & & \downarrow \\ \overset{k}{\underset{i=1}{\oplus}}\mathcal{O}(\nu_k-\nu_i) & \xrightarrow{\quad\widehat{\phi_1}\quad} & \overset{n}{\underset{i=1}{\oplus}}\mathcal{O}(n_i) \end{array}$$

y se demuestra que cualquier otra matriz canónica genera el mismo código convolucional si y sólo si para su morfismo de haces asociado, ϕ'', $Im\phi'' \subseteq Im\widehat{\phi_1}$.

Además, se demuestra que el haz $Im\widehat{\phi_1}$ no depende de ϕ_1. Por tanto, cada código convolucional tiene asociado un haz cociente, determinado por el morfismo $\overset{k}{\underset{i=1}{\oplus}}\mathcal{O}(\nu_k-\nu_i) \xrightarrow{\widehat{\phi_1}} \overset{n}{\underset{i=1}{\oplus}}\mathcal{O}(n_i)$. Además, este haz tiene polinomio de Hilbert $P_Q'(r) = (n-k)(r+1)+\sum_{i=1}^{n} n_i+\delta-k\nu_k$, luego el código convolucional está representado por un punto del esquema cociente $Quot(\overset{n}{\underset{i=1}{\oplus}}\mathcal{O}_{\mathbb{P}^1}(n_i), P_{Q'}(r))$. Y del mismo modo que en la construcción previa, esto permite representar al código convolucional como un único punto de la grasmanniana $Grass(k(\nu_k+1)-\delta, H^0(\overset{n}{\underset{i=1}{\oplus}}\mathcal{O}_{\mathbb{P}^1}(n_i)))$. Este punto, como subespacio contiene a los subespacios correspondientes a las distintas matrices generadoras del código convolucional según la construcción anterior. Además, por la unicidad del haz que contiene a todos los haces que pueden asociarse a matrices generadoras del código, este subespacio es el único con esa propiedad.

Este subespacio además puede representarse mediante una matriz constante asociada a una matriz generadora del código del siguiente modo: dada la matriz generadora del código convolucional

$$G = \begin{pmatrix} g^{(11)} & \dots & g^{(1n)} \\ \vdots & & \vdots \\ g^{(k1)} & \dots & g^{(kn)} \end{pmatrix}$$

con $g^{(ij)} = \sum g_k^{(ij)} x_0^{m_j-k} x_1^k$ un polinomio homogéneo de grado m_j, el subespacio que

representa el código está generado por las filas de la matriz

$$\begin{array}{l}
\nu_k-\nu_i+1\Big\{ \\
\quad\cdots \\
\nu_k-\nu_i+1\Big\{ \\
\quad\cdots \\
\nu_k-\nu_k+1\Big\{
\end{array}
\left(\begin{array}{ccccccc|ccc|ccccccc}
g_0^{(11)} & \dots & \dots & g_{m_1}^{(11)} & 0 & \dots & \dots 0 & \dots & \dots & g_0^{(n1)} & \dots & \dots & g_{m_n}^{(n1)} & 0 & \dots & \dots 0 \\
0\dots & g_0^{(11)} & \dots & \dots & g_{m_1}^{(11)} & 0\dots & \dots 0 & \dots & \dots & 0\dots & g_0^{(n1)} & \dots & \dots & g_{m_n}^{(n1)} & 0\dots & \dots 0 \\
0\dots & \dots 0 & g_0^{(11)} & \dots & \dots & g_{m_1}^{(11)} & 0\dots 0 & \dots & \dots & 0\dots & \dots 0 & g_0^{(n1)} & \dots & \dots & g_{m_n}^{(n1)} & 0\dots 0 \\
\hline
\multicolumn{16}{c}{\dots} \\
\hline
g_0^{(k1)} & g_1^{(k1)} & \dots & g_{m_1}^{(k1)} & 0 & \dots & 0 & \dots & \dots & g_0^{(kn)} & g_1^{(kn)} & \dots & g_{m_n}^{(kn)} & 0 & \dots & 0
\end{array}\right).$$

De la construcción anterior se pueden obtener varias deducciones

- Hay una relación entre códigos y haces, pero ésta no se da en términos de los haces asociados a las matrices generadoras del código, sino de ciertos haces que contienen a aquellos.
- Los vectores del subespacio único asociado al código convolucional se corresponden con el conjunto de palabras del código de grado máximo ν_k, que no depende de la matriz generadora. Esto prueba de otro modo que la construcción anterior es independiente de la matriz generadora que se considere para definir el morfismo $\overset{k}{\underset{i=1}{\oplus}}\mathcal{O}(\nu_k-\nu_i)\longrightarrow\overset{n}{\underset{i=1}{\oplus}}\mathcal{O}(n_i)$.
- Cuando todos los índices de Forney son iguales, las matrices unimodulares que definen la equivalencia de matrices generadoras son constantes. Así pues, ésta construcción y la desarrollada previamente coinciden.
- $n_j<\nu_k$ si y sólo si existe alguna matriz generadora cuyo grado de la columna j-ésima es menor que ν_1.

Además, teniendo en cuenta las inclusiones

$$Gr(\lambda,H^0(\mathbb{P}^1,\overset{n}{\underset{i=1}{\oplus}}\mathcal{O}(n_i)))\hookrightarrow Gr(\lambda,H^0(\mathbb{P}^1,\overset{n}{\underset{i=1}{\oplus}}\mathcal{O}(\nu_k))).$$

se deduce que *todo código de tipo* $[n,k,\delta]$ *con memoria* ν_k *se puede representar mediante un punto de la grassmanniana* $Gr(k(\nu_k+1)-\delta,K^{n(\nu_k+1)})$.

Una vez definida la relación de los códigos convolucionales de tipo $[n,k,\delta]$ y memoria m con puntos de la variedad $Gr(\lambda,H^0(\mathbb{P}^1,\overset{n}{\underset{i=1}{\oplus}}\mathcal{O}(m)))$, se quiere determinar cuáles de estos puntos representan de hecho a algún código convolucional. Para ello, teniendo en cuenta las inclusiones anteriores, bastará dar condiciones sobre las variedades $Gr(\lambda,H^0(\mathbb{P}^1,\overset{n}{\underset{i=1}{\oplus}}\mathcal{O}(n_i)))$.

En el Lema 2.30 se demuestra que *un punto de* $Gr(\lambda,H^0(\mathbb{P}^1,\overset{n}{\underset{i=1}{\oplus}}\mathcal{O}(n_i)))$ *representado por una matriz constante* M *se corresponde con un haz cociente de rango* $n-k$ *si y sólo si la matriz* $\widehat{M}$ *tiene rango* $\lambda+k$. En consecuencia, los puntos de dicha grassmanniana que representan haces cociente de rango $n-k$ con $k<\lambda$ forman un cerrado que se denota Z_k, mientras que aquellos para los que $k=\lambda$ forman un abierto denotado U_k. Por otro lado, como ya se vio, el conjunto de puntos de la grassmanniana que representan haces sin torsión en la recta afín (asociados a matrices básicas) forman un abierto, al igual que los puntos asociados del mismo modo a

matrices reducidas (por la caracterización de éstas en [McE98, Theorem A.2]). Así pues, el conjunto de puntos de la grassmanniana asociados a matrices canónicas es el abierto, denotado U_C, intersección de los dos abiertos anteriores.

Así, se tiene el siguiente resultado.

Los códigos convolucionales de longitud n, dimensión k y memoria m que tienen grado $\delta < km$ están representados por un abierto de un cerrado de la grassmanniana $Grass(k(m+1)-\delta, H^0(\bigoplus_{i=1}^{n} \mathcal{O}_{\mathbb{P}^1}(m)))$ definido por $U_C \cap Z_k$. Por su parte, los códigos que tienen grado $\delta = km$ están representados por el abierto de la grassmanniana $Gr(k, H^0(\mathbb{P}^1, \bigoplus_{i=1}^{n} \mathcal{O}(m)))$ definido por $U_C \cap U_k$.

La clasificación de códigos convolucionales presentada en este trabajo se diferencia de la que aparece en [RR94] en varios aspectos, entre los que destaca el considerar los grados por columnas del código e incluir la memoria como parámetro de clasificación lo que permite representar los códigos como puntos de variedades más pequeñas.

A.2.5 Consideraciones respecto a la Distancia Libre

Las cotas más conocidas para la distancia libre de códigos convolucionales son generalizaciones de algunas cotas para la distancia mínima de códigos de bloques. De modo similar a las generalizaciones presentadas en [GLS03] pueden considerarse ciertos códigos de bloques asociados a cada código convolucional cuya distancia mínima acota la distancia libre de éste. En los Teoremas 2.33 y 2.34 se presentan dos modos de asociar un código convolucional a una serie de códigos de bloques y como resultado se obtienen sendas generalizaciones de las cotas de Plotkin, Griesmer y Singleton. Una consecuencia particular es que *un código convolucional MDS de tipo $[n, k, \delta; m]$ con mínimos índices por columnas $\{n_i\}_1^n$ ha de verificar $\sum n_i \geq (m-1)n + k$ si $\delta < km$, o $n_i = m$ para todo $i \leq n$ si $\delta = km$.*

Además, la clasificación de códigos convolucionales realizada permite considerar un enfoque diferente y desarrollar otra cota a la distancia libre. La clave es la doble relación con las matrices generadoras de códigos cíclicos de la matriz asociada al espacio vectorial que representa al código convolucional como punto de la grassmanniana, y en concreto la particular forma de ésta (presentada anteriormente). Así, la cota presentada en el Teorema 2.36 depende de las distancias mínimas de ciertos códigos cíclicos que están determinados por las componentes de la matriz generadora del código convolucional.

A.2.6 Algunos Códigos Convolucionales Óptimos Obtenidos de los Códigos de Bloques Asociados

Puesto que la distancia mínima del punto que representa al código convolucional en la grassmanniana, pensado como código de bloques, permite dar una cota a su distancia libre, es natural preguntarse si códigos de bloques con óptima distancia mínima representan a códigos convolucionales con óptima distancia libre.

Téngase en cuenta que un mismo subespacio puede representar a distintos códigos convolucionales en función del espacio ambiente $H^0(\mathbb{P}^1, \bigoplus_{i=1}^{n} \mathcal{O}(n_i))$ que se considere, o lo que es lo mismo, de la sucesión de mínimos índices por columnas $\{n_i\}_1^n$ que se fije (y siempre que verifique las condiciones para representar a un código convolucional).

Así, se consideran diferentes tipos de códigos de bloques (Hamming, Hamming extendido, Reed-Solomon y otro código óptimo), definidos sobre distintos cuerpos base. Para cada uno de ellos, una o varias sucesiones $\{n_i\}_i^n$ se corresponden con códigos convolucionales con óptima distancia libre. Aunque no todos los códigos convolucionales así obtenidos son óptimos, la variedad de ejemplos expuestos dota de interés a este modo de definir códigos convolucionales.

A.3 Códigos de Goppa Convolucionales Asociados a Curvas Elípticas

A.3.1 Códigos de Goppa

La aparición a finales de los años setenta de los códigos de Goppa, que empleaban elementos algebro-geométricos para su construcción, fue el origen de una fuerte conexión entre la teoría de códigos y la geometría algebraica que ha dado numerosos frutos en ambas ramas. Por otro lado una intensa investigación permitió utilizar los mismos elementos algebro-geométricos para desarrollar o adaptar algoritmos de decodificación para estos códigos.

Construcción Geométrica de los Códigos de Goppa

Los códigos de Goppa se construyen sobre una curva irreducible no singular X de género g definida sobre un cuerpo finito $\mathbb{F}_q$. Para ello se emplean diversos elementos algebro-geométricos como puntos racionales, divisores, los haces asociados a éstos y sus correspondientes espacios de secciones globales y diferenciales. Las definiciones de dichos elementos pueden encontrarse en [Har77].

Un código de Goppa está determinado por dos divisores, $D = P_1 + \ldots + P_n$ y G, con soportes disjuntos y tal que los puntos P_i son puntos racionales de X distintos. Así, si $gr(G) < n = gr(D)$ puede darse un morfismo de evaluación inyectivo

$$\begin{aligned} \alpha : L(G) &\longrightarrow \mathbb{F}_q^n \\ f &\longmapsto (f(P_1), \ldots, f(P_n)) \end{aligned}$$

y *se define el código de Goppa determinado por* D *y* G, $\mathcal{C}(D, G)$, *como la imagen del morfismo* α.

Aplicando el teorema de Riemann-Roch se demuestra que $\mathcal{C}(D, G)$ *es un código de distancia mínima* $d \geq n - gr(G)$ *y dimensión* $k \geq gr(G) + 1 - g$. *Además, si* $gr(G) > 2g - 2$ *entonces* $k = gr(G) + 1 - g$. En particular, los códigos de Goppa definidos sobre la recta proyectiva son MDS por construcción.

Construcción Dual

Con los mismos divisores que en la construcción anterior y tomando los morfismos duales en la sucesión

$$0 \longrightarrow L(G) \xrightarrow{\alpha} \mathbb{F}_q^n \longrightarrow Coker\alpha \longrightarrow \dots$$

si se impone $gr(G) > 2g - 2$ puede darse un morfismo inyectivo

$$\begin{aligned} \beta : \Omega(G - D) &\longrightarrow \mathbb{F}_q^n \\ \omega &\longmapsto (\mathrm{res}_{P_1}(\omega), \dots, \mathrm{res}_{P_n}(\omega)) \end{aligned}$$

de modo que *se define el código dual de Goppa determinado por D y G, $\mathcal{C}^*(D, G)$, como la imagen del morfismo β.* Y aplicando el teorema de Riemann-Roch y la dualidad de Serre se tiene que *$\mathcal{C}^*(D, G)$ es un código de distancia mínima $d \geq gr(G) - 2g + 2$ y dimensión $k \geq n - gr(G) - 1 + g$. Además, si $gr(G) < n$ entonces $k = n - gr(G) - 1 + g$.*

Por otro lado, aplicando el teorema de los residuos, se comprueba que efectivamente $\mathcal{C}(D, G)$ y $\mathcal{C}^*(D, G)$ son códigos duales.

Los códigos de Reed-Solomon y de Reed-Solomon generalizados son casos particulares de códigos de Goppa.

A.3.2 Códigos de Goppa Convolucionales

Puesto que los códigos convolucionales son una generalización de los códigos de bloques, cabría esperar que una construcción similar a la de Goppa pudiera dar como resultado códigos convolucionales con buenas propiedades.

Dicha estrategia ha dado como resultado los denominados *códigos de Goppa convolucionales* presentados primero en [MDS04] y después con herramientas más simples en [MDIS06], cuya construcción exponemos brevemente.

Construcción General

Se considera una curva proyectiva no singular X de género g definida sobre el cuerpo $\mathbb{F}_q(z)$, y se supone $\mathbb{F}_q(z)$ algebraicamente cerrado en el cuerpo de funciones racionales de X, en cuyo caso se pueden aplicar los teoremas de los residuos y de Riemman-Roch.

Se consideran dos divisores $D = P_1 + \dots + P_n$ y G con soporte disjunto y tal que los puntos P_i son puntos $\mathbb{F}_q(z)$-racionales de X distintos. Así, de modo similar al caso clásico, si $gr(G) < n$ se tiene un morfismo inyectivo

$$\begin{aligned} \alpha : L(G) &\longrightarrow \mathbb{F}_q(z)^n \\ s &\longmapsto (s(P_1), \dots, s(P_n)) \end{aligned}$$

y *se define el código de Goppa convolucional determinado por D y G, $\mathcal{C}(D, G)$, como la imagen de α, mientras que dado un subespacio $S \subset L(G)$ se define el código de Goppa convolucional determinado por D y S, $\mathcal{C}(D, S)$, como $Im\alpha_{|S}$.*

De nuevo aplicando el teorema de Riemman-Roch se tiene que $\mathcal{C}(D,G)$ *es un código convolucional de dimensión* $k \geq deg(G) + 1 - g$, *y si* $deg(G) > 2g - 2$ *entonces* $k = deg(G) + 1 - g$.

El cálculo de la distancia libre es mucho más complejo que en la construcción clásica y requiere el desarrollo de herramientas más sofisticadas sobre cuerpos finitos.

Códigos de Goppa Convolucionales Duales

Con los mismos divisores, y tomando los morfismos duales en la sucesión que permite definir los códigos de Goppa convolucionales, imponiendo $deg(G) > 2g - 2$ se tiene el morfismo inyectivo

$$\begin{aligned} \beta : \Omega(G - D) &\longrightarrow \mathbb{F}_q^n \\ \omega &\longmapsto (\mathrm{res}_{P_1}(\omega), \ldots, \mathrm{res}_{P_n}(\omega)) \end{aligned}$$

Y *se define el código dual de Goppa convolucional determinado por* D *y* G, $\mathcal{C}^*(D,G)$, *como la imagen del morfismo* β. Además $\mathcal{C}^*(D,G)$ *es un código de dimensión* $k \geq n - gr(G) - 1 + g$ *y si* $gr(G) < n$ *entonces* $k = n - gr(G) - 1 + g$.

Como en el caso clásico, aplicando el teorema de los residuos, se comprueba que efectivamente $\mathcal{C}(D,G)$ y $\mathcal{C}^*(D,G)$ son códigos convolucionales duales.

Códigos de Goppa Convolucionales sobre la Recta Proyectiva

La siguiente construcción, presentada en [MDIS06], ilustra la obtención de una familia particular de códigos de Goppa convolucionales.

Sea $X = \mathbb{P}^1_{\mathbb{F}_q(z)}$, la recta proyectiva sobre $\mathbb{F}_q(z)$, P_0, P_∞ los puntos en el origen y el infinito respectivamente y $P_1, \ldots, P_n$, con $P_i = (1; x_i)$, otros n puntos racionales diferentes. Consideremos $D = P_1 + \ldots + P_n$ y $G = rP_\infty - sP_0$, con $0 \leq s < r < n$, de modo que $L(G)$ está generado por $\{t^s, t^{s+1} \ldots, t^r\}$. En esas condiciones el morfismo

$$\begin{aligned} \alpha : L(G) &\longrightarrow \mathbb{F}_q(z)^n \\ t^i &\longmapsto (x_1^i, \ldots, x_n^i) \end{aligned}$$

es inyectivo y su imagen constituye el código de Goppa convolucional determinado por D y G, $\mathcal{C}(D,G)$, de longitud n y dimensión $k = r - s + 1$.

A.3.3 Códigos de Goppa Convolucionales sobre Curvas Elípticas

Sea X una curva elíptica plana sobre $\mathbb{F}_q(z)$, de la que supondremos tiene un punto racional de orden al menos 4 (de modo que haya puntos racionales suficientes para definir un código) y por tanto se puede escribir en un abierto afín en forma normal de Tate

$$y^2 + axy + by = x^3 + bx^2$$

siendo x, y las coordenadas afines de este abierto. Sean P_0, P_∞ los puntos en el origen y el infinito respectivamente y $P_1, \ldots, P_n$, con $P_i = (x_i, y_i)$, otros n puntos racionales

de X diferentes. Consideremos los divisores $D = P_1 + \ldots + P_n$ y $G = rP_\infty - sP_0$, con $0 < r - s < n$, de modo que $L(G)$ está generado por $\{x^a y^b, \ldots, x^c y^d\}$ con $a + 2b = s$, $2c + 3d = r$ (siendo $b, d = 0, 1$ para evitar dependencias lineales). Así, se tiene el morfismo inyectivo

$$\begin{aligned} \alpha : L(G) &\longrightarrow \mathbb{F}_q(z)^n \\ t^i &\longmapsto (x_1^i, \ldots, x_n^i) \end{aligned}$$

cuya imagen constituye el código de Goppa convolucional determinado por D y G sobre X, $\mathcal{C}(D, G)$, de longitud n y dimensión $k = r - s$ puesto que $g = 1$.

A.3.4 Algunos Códigos de Goppa Convolucionales Óptimos Definidos sobre Curvas Elípticas

La construcción de códigos de Goppa convolucionales sobre curvas elípticas da como resultado múltiples códigos de diversos parámetros construidos sobre diferentes cuerpos finitos. En la sección 3.4 se presenta una selección de códigos convolucionales con óptima distancia libre, definidos en curvas elípticas sobre cuerpos finitos de cardinal primo mayor o igual a 2. La longitud de estos códigos varía entre 2 y 5, su dimensión entre 1 y 3 y su grado entre 1 y 8.

En particular es de resaltar el conjunto de códigos fuertemente MDS. Estos códigos permiten alcanzar su máxima capacidad correctora con el mínimo número posible de coeficientes vectoriales de la palabra recibida. Destaca el hecho de que puedan construirse códigos cuya longitud y la característica del cuerpo base no son primos entre sí. Por otro lado, los ejemplos expuestos sugieren que esta construcción es un modo prometedor de obtener códigos fuertemente MDS con una estructura algebraica para poder así adaptar algoritmos de decodificación conocidos para estos códigos que por ser genéricos no resultan útiles en la práctica.

A.3.5 Códigos AG Convolucionales

Códigos AG de Bloques

La aparición de los códigos de Goppa originó el desarrollo de múltiples familias de códigos definidos en términos algebro-geométricos. Muchas de esas familias, incluyendo los códigos de Goppa clásicos, son casos particulares de los llamados *códigos AG generalizados* [OS99]. Nuestra intención es trasladar dicha construcción al contexto de los códigos convolucionales de modo similar a como se ha hecho para los códigos de Goppa.

Códigos AG Generalizados

Sea X una curva proyectiva no singular de género g sobre $\mathbb{F}_q$, con s puntos $P_1, \ldots, P_s$ de grados $gr(P_i) = k_i$. Sea un divisor G cuyo soporte no contiene a los mencionados puntos, y sea para cada punto P_i un isomorfismo $\pi_i : \mathcal{O}_{P_i}/_{\mathfrak{m}_{P_i}} \to \mathcal{C}_i$ entre su cuerpo

residual y un código de bloques de tipo $[n_i, k_i, d_i]$. La imagen del morfismo

$$\begin{aligned} \pi : L(G) &\longrightarrow \mathbb{F}_q^{\sum n_i} \\ f &\longmapsto (\pi_1(f(P_1)), \ldots, \pi_s(f(P_s))) \end{aligned}$$

se define como el *código AG generalizado* determinado por los puntos $P_1, \ldots, P_s$, el divisor G y los códigos $\mathcal{C}_1, \ldots, \mathcal{C}_s$. Se define su *distancia mínima de diseño* como $\delta = min\left\{\sum_{i \notin S} d_i \mid S \in I\right\}$, siendo $I = \left\{S \subseteq \{1, \ldots, s\} \mid \sum_{i \in S} k_i \leq degG\right\}$.

Se demuestra que *si* $gr(G) < \sum_{i=1}^{s} k_i$ *el código AG generalizado es un código de longitud* $\sum n_i$*, dimensión* $k \geq gr(G) + 1 - g$ *y distancia mínima* $d \geq \delta$*.*

Códigos AG Convolucionales

De modo similar, se considera una curva proyectiva no singular sobre $\mathbb{F}_q(z)$, s puntos $P_1, \ldots, P_s$ de grados $gr(P_i) = k_i$, un divisor G cuyo soporte no contiene a los mencionados puntos, y para cada punto P_i un isomorfismo $\pi_i : \mathcal{O}_{P_i}/_{\mathfrak{m}_{P_i}} \to \mathcal{C}_i$ entre su cuerpo residual y un código convolucional (que como objeto algebraico es un $\mathbb{F}_q(z)$-espacio vectorial) de tipo $[n_i, k_i]$ y distancia libre d_i. La imagen del morfismo análogo $\pi : L(G) \longrightarrow \mathbb{F}_q(z)^{\sum n_i}$ se define como el *código AG convolucional* determinado por los puntos $P_1, \ldots, P_s$, el divisor G y los códigos convolucionales $\mathcal{C}_1, \ldots, \mathcal{C}_s$. De modo similar al caso de códigos de bloques se define su *distancia libre de diseño* δ_{free}.

Se demuestra que *si* $gr(G) < \sum_{i=1}^{s} k_i$ *el código AG convolucional es un código convolucional de longitud* $\sum n_i$*, dimensión* $k \geq gr(G) + 1 - g$ *y distancia libre* $d_{free} \geq \delta_{free}$*.*

La particularidad de la construcción convolucional consiste en que modificando ligeramente la construcción, se puede aumentar la distancia libre de diseño.

Dado el código AG convolucional $\mathcal{C}(P_1, \ldots, P_s; G; \mathcal{C}_1, \ldots, \mathcal{C}_s)$ *sea* $\{f_1, \ldots, f_k\}$ *una base de* $L(G)$ *tal que* $\{\pi(f_1), \ldots, \pi(f_k)\}$ *genera un* $\mathbb{F}_q[z]$*-submódulo libre, y sea* $\forall i\ d_i' \geq d_i$ *el peso mínimo de los vectores polinómicos del submódulo generado por* $\{\pi_i(f_1(P_i)), \ldots, \pi_i(f_k(P_i))\}$*. Entonces el código convolucional* $\langle \pi(f_1), \ldots, \pi(f_k) \rangle_{\mathbb{F}_q[z]}$ *tiene distancia libre de diseño* $\delta_{free}' := min\left\{\sum_{i \notin S} d_i' \mid S \in I\right\}$*.*

A.4 Sistemas Lineales y Códigos Convolucionales

A.4.1 Breve Introducción a los Sistemas Lineales

Un sistema es un modelo de un fragmento de la naturaleza utilizado para estudiar el comportamiento dinámico de éste. Para ello, el modelo tiene variables que representan los estímulos externos que recibe el sistema, las variables de entrada u, su estado o circunstancias actuales (que pueden modificar su comportamiento), las variables de estado x, y la respuesta dada por el sistema, las variables de salida y. Todas estas variables son función de la variable tiempo t.

Casi todos los sistemas pueden representarse mediante una serie de ecuaciones lineales en función de la variable t que en forma matricial se escriben

$$\begin{aligned} \dot{x}(t) &= A(t)x(t) + B(t)u(t) \\ y(t) &= C(t)x(t) + D(t)u(t) \end{aligned}$$

siendo A, B, C, D matrices de funciones racionales en t. El cuádruple (A, B, C, D) se denomina *realización* del sistema. Cuando estas matrices son constantes se dice que el sistema es *invariante en el tiempo.*

Por otro lado, no es raro que para una mejor modelización de ciertos sistemas la variable tiempo se considere discreta. Así, un sistema discreto invariante en el tiempo estaría representado por las ecuaciones

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t \\ y_t &= Cx_t + Du_t \end{aligned} \quad . \tag{A.1}$$

Esta representación no es única y en particular cuando la longitud del vector estado es la mínima posible se tiene una *realización mínima.*

Una herramienta fundamental para describir un sistema es la *matriz de transferencia*, que relaciona directamente las variables de entrada y de salida del sistema y que viene dada por $T(z) = C(zId - A)^{-1}B + D$.

Dos características importantes de algunos sistemas son la *observabilidad* y la *controlabilidad*. La primera se refiere a la posibilidad de determinar el estado del sistema en un instante a partir de las salidas en los instantes posteriores. La segunda consiste en la posibilidad de dirigir el sistema mediante unas *señales de control* adecuadas en la entrada para que pase de un instante inicial a otro estado final deseado. Ambas características son aspectos duales del mismo problema.

Una técnica particular de control es el control óptimo, basado en determinar las señales de control mediante una ley de modo que el sistema sea óptimo respecto a cierto criterio. Dicho criterio consiste a menudo en minimizar una función de coste que toma la forma de la suma o integral de alguna función.

Algunas referencias clásicas en teoría de sistemas lineales y teoría de control son [AM71, AM89, Son98].

A.4.2 Códigos Convolucionales como Sistemas Lineales

La relación entre códigos convolucionales y sistemas lineales ha sido largamente conocida y estudiada. Así, distintas nociones y resultados de una rama han sido interpretados en la otra, como por ejemplo la caracterización de los conceptos de controlabilidad y observabilidad en códigos convolucionales o la descripción del proceso de decodificación como un problema de control óptimo.

Un modo natural de asociar a cada código convolucional un sistema lineal es considerando los mensajes a codificar como entradas del sistema y las palabras codificadas como salidas. Es decir, en el sistema con ecuaciones (A.1), las variables u_t serían los mensajes y las variables y_t representarían las palabras del código. La

matriz de transferencia de este sistema sería $C(zId - A)^{-1}B + D = G(z^{-1})$, siendo $G(z)$ una matriz generadora del código. Sin embargo esta representación tiene la desventaja de que la matriz A debe ser nilpotente, lo que limita mucho el conjunto de sistemas que representan códigos convolucionales.

Otra representación que permite considerar un conjunto más extenso de sistemas lineales sería la siguiente. Sea $G(z)$ una matriz $n \times k$ generadora del código (representada "verticalmente" para adecuarse a la notación habitual en teoría de sistemas) y tomemos la partición

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

siendo $Q(z)$ una submatriz cuadrada $k \times k$ que supondremos (salvo permutación de filas) tiene por determinante un polinomio de grado δ, la complejidad del código. Del mismo modo, para cada palabra del código $c(z) = \sum c_i z^i$ consideremos la partición análoga en cada coeficiente vectorial $c_t = (y_t, u_t)$. Así se tiene la representación del código mediante un sistema con ecuaciones (A.1) cuyas entradas u_t y salidas y_t forman parte de cada coeficiente de $c(z)$. La matriz de transferencia de este sistema lineal sería $C(zId - A)^{-1}B + D = P(z)Q(z)^{-1}$.

Decodificación de Códigos Convolucionales desde el Punto de Vista de los Sistemas Lineales

Esta representación permite además interpretar el proceso de decodificación del siguiente modo. Dada una palabra recibida $v(z) = (y'(z), u'(z))$ (con la partición análoga a la anterior), se desea encontrar la palabra del código que minimiza el error

$$\min_{c(z)\in\mathcal{C}} d(c(z), v(z)) = min \left(\sum_{k=0}^{T} d(y_k, y'_k) + d(u_k, u'_k) \right) .$$

Este objetivo puede interpretarse en teoría de sistemas de dos modos: como un problema de rastreo o como un problema de filtrado. En el primer caso se pretende rastrear el mensaje recibido para obtener la palabra del código que con mayor probabilidad fue la enviada. En el segundo se trata de filtrar el error introducido en la palabra del código durante la comunicación.

Aunque estos problemas son clásicos en teoría de sistemas, se han estudiado sobre los cuerpos de los números reales o complejos, haciendo uso de una métrica euclídea que en los cuerpos finitos, el contexto donde habitualmente se definen los códigos, no se tiene. Por ello, las soluciones clásicas no pueden aplicarse directamente.

A.4.3 Problemas de Rastreo sobre Cuerpos Finitos

Nuestro objetivo es plantear un problema de rastreo sobre cuerpos finitos sustituyendo la métrica euclídea por la métrica que define la distancia de Hamming y utilizar ideas similares al caso clásico para su resolución. Así, este problema podrá ser utilizado en particular para desarrollar un algoritmo de decodificación de códigos convolucionales.

El Problema de Rastreo Clásico

Consideremos el sistema lineal discreto definido por las ecuaciones

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t \\ y_t &= Cx_t \\ x_{t_0} &= x_0 \end{aligned}$$

con A, B, C matrices constantes sobre $\mathbb{R}$ o $\mathbb{C}$. El problema consiste en *dada una sucesión* $\{\tilde{y}_t\}_{t_0}^T$ *determinar la sucesión* $\{\tilde{u}_t\}_{t_0}^T$ *que minimiza la función de coste*

$$J(x_0, u(\cdot), T) = \sum_{t=t_0}^{T} [u_t^\top R u_t + (y_t - \tilde{y}_t)^\top Q (y_t - \tilde{y}_t)]$$

siendo R y Q matrices definidas positiva y no negativa respectivamente.

Este problema se resuelve habitualmente transformándolo en un problema estándar de regulador que a su vez se resuelve aplicando una ecuación de Ricatti.

Un Problema de Rastreo sobre un Cuerpo Finito

Consideremos el sistema descrito por las ecuaciones anteriores definido sobre un cuerpo finito $\mathbb{F}$. Se toma como métrica la definida por la distancia de Hamming. Así, la función de coste a minimizar será

$$J(x_0, u(\cdot), T) = \sum_{t=t_0}^{T} [w(Ru_t) + w(Q(y_t - \tilde{y}_t))]$$

siendo Q y R matrices cuadradas y R con rango máximo.

Para resolver el problema se considera el conocido como *principio de optimalidad de Bellman: Toda trayectoria óptima tiene la propiedad de que para cualquier punto intermedio, el resto de la trayectoria es la trayectoria óptima obtenida con este punto como punto inicial.*

Aplicando este principio la minimización de la función de coste puede transformarse en una sucesión de minimizaciones de las expresiones $w(Ru_t)+w(QCx_{t+1}-Q\tilde{y}_{t+1})$. Agrupando los términos conocidos y los que dependen de u_t este problema equivale a encontrar el vector de mínimo peso en el conjunto $\{z' + Bu\}_u$, siendo $z' = (QCAx_t - Q\tilde{y}_{t+1}, 0, \overset{k}{\ldots}, 0)$ y $B = \binom{QCB}{R}$. Y a su vez, este problema es equivalente a considerar B como la matriz generadora de un código de bloques y decodificar z' con respecto a dicho código.

Nótese que para ello el algoritmo utilizado ha de dar una respuesta incluso en el caso de que haya un error detectable pero no corregible. Por otro lado, podría suceder que z' estuviera a igual distancia de dos palabras del código, lo que significaría que el problema tendría dos soluciones igualmente válidas. El modo de evitar esto se trata más adelante.

La solución de un problema de control óptimo no consiste solamente en dar la sucesión que optimiza la función de coste, sino además en estimar el valor que ésta

toma. En nuestro caso, sólo podemos dar una cota para ese valor, que dependerá del *radio de recubrimiento* del código generado por B, ρ_B, es decir, el mínimo radio de las circunferencias que centradas en las palabras del código, recubren todo el espacio ambiente. Así, se demuestra que *el valor óptimo de la función de coste verifica* $J(x_0, \tilde{u}, T) \leq (T-1)\rho_{B_1} + w(QCx_0 - Q\tilde{y}_{t_0})$.

Un Problema de Rastreo en Tiempo Infinito sobre un Cuerpo Finito

El planteamiento análogo del problema de rastreo en tiempo infinito no tendría sentido puesto que el peso de Hamming sólo toma valor cero sobre el vector nulo, por lo que en general el valor de la función de coste será infinito. Por ello se considera una pequeña modificación del enunciado, y *dada una sucesión* $\{\tilde{y}_t\}_{t=t_0}^{\infty}$, *se busca la sucesión* $\{u_t\}_{t=t_0}^{\infty}$ *de modo que la función de coste* $J(x_0, \tilde{u}, T)$ *es mínima para todo* $T < \infty$.

Para resolverlo se utiliza el llamado *método por retroceso de horizonte*, un método iterativo que consiste en tomar el estado en el instante actual como estado inicial para resolver un problema de rastreo en tiempo finito N y tomar de la solución solamente el primer vector, que a su vez se utiliza para calcular el estado en el instante siguiente.

La resolución del problema de rastreo en tiempo finito podría realizarse iterativamente como se vio antes, sin embargo también puede utilizarse un método directo. El objetivo es encontrar la sucesión $\{u_t, \ldots, u_{t+N-1}\}$ que minimiza $\sum_{i=0}^{N-1}[w(QCx_{t+i+1} - Q\tilde{y}_{t+i+1}) + w(Ru_{t+i})] = w(z_N)$, siendo el vector $z_N = w_{t,N} + B_N u_{t,N}$, donde

$$w_{t,N} = \begin{pmatrix} QCA^N \\ QCA^{N-1} \\ \vdots \\ QCA \\ 0 \\ \vdots_{Nk} \\ 0 \end{pmatrix} x_t, \quad B_N = \begin{pmatrix} QCB & QCAB & \ldots & QCA^{N-1}B \\ 0 & QCB & \ldots & QCA^{N-2}B \\ & & \ddots & \\ 0 & 0 & & QCB \\ R & & & \\ & R & & \\ & & \ddots & \\ & & & R \end{pmatrix}$$

$$u_{t,N} = \begin{pmatrix} u_{t+N-1}, & u_{t+N-2}, & \ldots & u_t \end{pmatrix}.$$

Es decir, se busca el vector $u_{t,N}$ con peso mínimo en el conjunto

$$\{w_{t,N} + B_N u_{t,N}\}_{u_{t,N} \in \mathbb{F}^{Nk}}$$

lo que, como antes, se realiza mediante la decodificación del vector $w_{t,N}$ con respecto al código de bloques generado por la matriz B_N. Se demuestra además que puesto que $w_{t,N}$ depende del vector x_t, de longitud δ, *este proceso de decodificación es como mucho tan complejo como una decodificación respecto a un código de longitud* $\delta + Nk$.

Soluciones Múltiples

Es importante en todo problema de control óptimo determinar en qué condiciones la solución al problema es única. En nuestro caso, la multiplicidad puede aparecer por dos razones.

La primera es que haya dos vectores u_t, u'_t diferentes que se correspondan con el mismo vector del conjunto $\{z' + Bu\}_u$ (en particular con el que tiene peso mínimo). Sin embargo, puesto que R se supone de rango máximo, también lo es B y por tanto este caso es imposible.

La segunda razón es que z' sea equidistante a dos o más palabras del código generado por B, en cuyo caso hay dos o más soluciones igualmente válidas. Para solucionarlo puede considerarse la resolución directa de un problema de rastreo en tiempo finito $N = 2$ como en el método por retroceso de horizonte y tomar la solución completa $\{u_t, u_{t+1}\}$. En caso de que de nuevo el vector a decodificar fuera equidistante a dos o más palabras del código generado por B_2 se repetiría el proceso para $N = 3$, y así sucesivamente hasta obtener una solución única o hasta que en algún instante $t + N$, las soluciones $u = \{u_t, \ldots, u_{t+N-1}\}$, $u' = \{u'_t, \ldots, u'_{t+N-1}\}$ tuvieran vectores $u_{t+N-1} = u'_{t+N-1}$ de modo que no es posible discriminar una de la otra tomando un horizonte de mayor longitud. Sin embargo, se prueba que la probabilidad de que eso ocurra es muy pequeña y decrece a medida que N es mayor.

Para verlo se considera la interpretación geométrica del proceso de decodificación, en que a cada palabra del código se le asocia la esfera centrada en ella y de radio $t = \lfloor \frac{d-1}{2} \rfloor$, siendo d la distancia mínima del código. Estas esferas son las que, centradas en las palabras del código y siendo disjuntas, tienen radio máximo. Así, un vector cualquiera que se encuentre en una de estas esferas se decodifica como la palabra del código en su centro, mientras que uno que no se encuentra en ninguna tiene un error detectable pero no corregible. Obsérvese que la cantidad de vectores contenidos en cada una de estas esferas es $\sum_{i=0}^{\lfloor \frac{d-1}{2} \rfloor} \binom{n}{i} (q-1)^i$ siendo q el tamaño del cuerpo base. Así, se tiene que la densidad del código, es decir, la proporción de espacio ambiente recubierto por estas esferas, es

$$\delta_{\mathcal{C}} = \frac{\sum_{i=0}^{\lfloor \frac{d-1}{2} \rfloor} \binom{n}{i} (q-1)^i}{q^{n-k}}.$$

En nuestro caso, el conjunto de vectores que se quieren decodificar respecto al código generado por B no es todo el espacio ambiente, sino $\mathcal{H} = \mathbb{F}^{n-k} \times 0^k$. A ese respecto, la proporción de $\mathcal{H}$ recubierta por estas esferas es

$$\delta_{\mathcal{H}} = \frac{\sum_{i=0}^{t} \binom{k}{i} (q-1)^i \sum_{j=0}^{t-i} \binom{n-k}{j} (q-1)^j}{q^{n-k}} \quad .$$

y de modo análogo para el código generado por B_N y los vectores de $\mathcal{H}_N = \mathbb{F}^{N(n-k)} \times 0^{Nk}$ se tiene la densidad $\delta_{\mathcal{H}_N}$. Así pues, la probabilidad de que un vector de $\mathcal{H}$, respectivamente $\mathcal{H}_N$, elegido aleatoriamente no pertenezca a alguna de dichas esferas (y por tanto no se pueda decodificar) es $P_o^{\mathcal{H}} = 1 - \delta_{\mathcal{H}}$, respectivamente $P_o^{\mathcal{H}_\mathcal{N}} = 1 - \delta_{\mathcal{H}_\mathcal{N}}$, y se tiene que $P_o^{\mathcal{H}}, P_o^{\mathcal{H}_\mathcal{N}} < 1$.

Así, si se considera una realización controlable del sistema, es decir, que cualquier estado puede ser alcanzado, y una probabilidad uniforme de tener el vector $z' \in \mathcal{H}$, se demuestra que *la probabilidad en un problema de rastreo sobre un cuerpo finito de necesitar resolver un problema con horizonte N para evitar soluciones múltiples es asintóticamente 0. Además, la probabilidad de M soluciones óptimas, que difieren en $N-1$ vectores y no pueden discriminarse considerando un problema con horizonte de mayor longitud, está acotada por*

$$\prod_{i=1}^{N-1} P_o^{\mathcal{H}_i} \frac{\delta_{\mathcal{H}}^M}{E_{k,t}}.$$

A.4.3 Decodificación Convolucional como Problema de Rastreo

Como ya se ha mencionado, la decodificación de un código convolucional puede interpretarse como un problema de rastreo.

Consideremos en la matriz generadora de un código convolucional la partición explicada anteriormente y la correspondiente para las palabras del código, que definen la realización correspondiente del sistema asociado, representado por ecuaciones de la forma (A.1). La decodificación de la sucesión $\{(\tilde{y}_t, \tilde{u}_t\}$ equivale a resolver el problema de rastreo con función de coste

$$J(x_0, u(\cdot), T) = \sum_{t=0}^{T} [w(u_t - \tilde{u}_t) + w(y_t - \tilde{y}_t)].$$

Aplicando el principio de Bellman, la resolución del problema equivale a una minimización multiple de términos de la forma $w(u_t - \tilde{u}_t) + w(y_t - \tilde{y}_t)$, lo que equivale a encontrar el vector de peso mínimo del conjunto $\{z' + Bu\}_u$ siendo $z' = (Cx_t - \tilde{y}_t, -\tilde{u}_t)$ y $B = \binom{D}{Id}$. Así, se resuelve este problema mediante la decodificación de z' respecto al código generado por B.

En aplicaciones prácticas es común que la decodificación comience antes de haber recibido toda la sucesión, lo que lleva a interpretar este proceso como un problema en tiempo infinito. Así pues, se considera el problema de rastreo en tiempo infinito asociado al proceso de decodificación, que se resuelve por el método por retroceso de horizonte. En este caso emplearemos la variante en que en cada paso se toman de la solución los L primeros vectores, $\{u_t, \ldots, u_{t+L-1}\}$ (y los correspondientes $\{y_t, \ldots, y_{t+L-1}\}$). L dependerá de los errores que puedan resolverse en cada paso del método con horizonte N.

De modo similar al explicado anteriormente, este problema se resuelve mediante la decodificación del vector $w_{t,N}$ con respecto al código generado por la matriz B_N, siendo en este caso

$$w_{t,N} = \begin{pmatrix} CA^N \\ \vdots \\ C \\ 0 \\ \vdots \\ 0 \end{pmatrix} x_t - \begin{pmatrix} \tilde{y}_{t+N} \\ \vdots \\ \tilde{y}_t \\ \tilde{u}_{t+N} \\ \vdots \\ \tilde{u}_t \end{pmatrix}, \quad B_N = \begin{pmatrix} D & H_0 & H_1 & \dots & H_{N-1} \\ 0 & D & H_0 & \dots & H_{N-2} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & H_0 \\ 0 & \dots & \dots & 0 & D \\ & & Id_{Nk} & & \end{pmatrix},$$

con $H_i = CA^iB$.

El método permite corregir $\lfloor \frac{d'}{2} \rfloor$ errores, con $d' \geq d_N - 1$, salvo un error admisible, si toda palabra de peso menor o igual que d' del código generado por B_N tiene soporte disjunto con las componentes $\{(N-L)(n-k)+1, \dots, N(n-k), Nn-Lk+1, \dots, Nn\}$.

Para probar esta característica se considera la matriz de control del código generado por B_N y se realiza un razonamiento similar al resultado clásico que caracteriza la distancia mínima de un código con el mínimo número de columnas dependientes de su matriz de control. Se tendrá en cuenta que en nuestro caso sólo se toman para la solución los vectores $u_t, \dots, u_{t+L}$ (y los correspondientes $y_t, \dots, y_{t+L}$), de modo que cualquier error en las posiciones pertenecientes al resto es admisible.

Conclusiones

Este trabajo se centra en el estudio de los códigos convolucionales, considerando algunos de los aspectos más importantes vinculados a ellos, como el estudio de su estructura matemática, la construcción de nuevas familias de códigos y su decodificación. Para ello se han empleado elementos de geometría algebraica y teoría de sistemas lineales.

La clasificación desarrollada en el capítulo 2 permite asociar a cada código convolucional con un punto de una grassmanniana. Asimismo, se determinan las condiciones bajo las cuales un punto representa a un código convolucional. La información obtenida se ha aplicado en la obtención de nuevas cotas a la distancia libre y en un método para obtener códigos convolucionales a partir de ciertos códigos de bloque con óptima distancia libre.

La construcción de códigos de Goppa convolucionales sobre curvas elípticas permite obtener una gran variedad de códigos con la máxima distancia libre, y en particular, códigos fuertemente MDS. Por otro lado, se ha desarrollado para códigos convolucionales la construcción análoga a la de los códigos AG generalizados.

Finalmente, interpretando la decodificación de códigos convolucionales como un problema de trazado y proponiendo una solución a dicho problema para sistemas lineales definidos sobre cuerpos finitos, se ha desarrollado un algoritmo de decodificación que permite explotar al máximo la capacidad correctora de cada código.

Index

Bibliography

[AK80] A. Altman and S. Kleiman, *Compactification of the picard scheme*, Advances in Mathematics (1980), no. 35, 50 112.

[AM71] B.D.O. Anderson and J.B. Moore, *Linear optimal control*, Prentice-Hall, January 1971.

[AM89] ———, *Optimal control: Linear quadratic methods*, Prentice-Hall, 1989.

[BC60a] R.C. Bose and D.K.Ray Chaudhuri, *Further results on error correcting binary group codes*, Information and Control **3** (1960), 279–290.

[BC60b] ———, *On a class of error correcting binary group codes*, Information and Control **3** (1960), 68–79.

[Bel57] R. Bellman, *Dynamic programming*, Princeton University Press, Princeton, NJ, 1957.

[Ber67] Elwyn R. Berlekamp, *Factoring polynomials over finite fields*, Bell Systems Technical Journal (1967), no. 46, 1853–1859, Later republished in: Elwyn R. Berlekamp. "Algebraic Coding Theory", Ch. 7. Mc-Graw Hill, 1968.

[Dou03] J. M. Doumen, *Some applications of coding theory in cryptography*, Ph.D. thesis, Technische Universiteit Eindhoven, 2003.

[Eli55] P. Elias, *Coding for noisy channels*, IRE Conv. Rec. **4** (1955), 37–46.

[FH01] P.A. Fuhrmann and U. Helmke, *On the parametrization of conditional invariant subspaces and observer theory*, Linear Algebra and its Applications (2001), no. 332-334, 265–353.

[For70] G. D. Forney, *Convolutional codes i: Algebraic structure*, IEEE Trans. Information Theory (1970).

[GBT93] A. Glavieux, C. Berrou, and P. Thitimajshima, *Near shannon limit error-correcting coding and decoding: turbo codes*, Proc. IEEE Int. Conf. on Communications, 1993, pp. 1064–1070.

[GL05] H. Gluesing-Luerssen, *On the weight distribution of convolutional codes*, Linear Algebra and its Applications **408** (2005), 298.

[GLRS03] Heide Gluesing-Luerssen, Joachim Rosenthal, and Roxana Smarandache, *Strongly mds-convolutional codes*, E-print math.RA/0303254, March 2003.

[GLS03] H. Gluesing-Luerssen and W. Schmale, *Distance bounds for convolutional codes and some optimal codes*, arXiv:math/0305135v1, 2003.

[Gol49] M. J. E. Golay, *Notes on digital coding*, Proc. I.R.E., 1949, p. 637.

[Gop77] V.D. Goppa, *Codes associated with divisors*, Probl. Peredachi Inform. **13** (1977), no. 1, 33–39, Translation: *Probl. Inform. Transmission*, vol. 13, pp. 22-26, 1977.

[Gop81] ———, *Codes on algebraic curves*, Dokl. Adad. Nauk SSSR **259** (1981), 1289–1290, Translation: *Soviet Math. Dokl.*, vol 24, pp. 170-172, 1981.

[Gro60] A. Grothendieck, *Techniques de construction et théorèmes d'existence en géométrie algébrique iv: les schémas de hilbert. séminaire bourbaki* **221**, Fondements de la Géométrie Algébrique, 1960.

[Ham50] R. W. Hamming, *Error detecting and error correcting codes*, Bell System Technical Journal **26** (1950), no. 2, 147–160.

[Har77] R. Hartshorne, *Algebraic geometry*, Grad. Texts in Math., vol. 52, Springer-Verlag, New York, 1977.

[Hoc59] A. Hocquenghem, *Codes correcteurs d'errors*, Chiffres **2** (1959), 147–156.

[HP95] T. Høholdt and R. Pellikaan, *On the decoding of algebraic-geometric codes*, IEEE Trans. Information Theory **IT-41** (1995), 1589–1614.

[HRS05] Ryan Hutchinson, Joachim Rosenthal, and Roxana Smarandache, *Convolutional codes with maximum distance profile*, Systems & Control Letters (2005), no. 54 (1), 53–63.

[Hus87] D. Husemoller, *Elliptic curves*, Springer Verlag, New York, 1987.

[HvLP98] T. Høholdt, J.H. van Lint, and R. Pellikaan, *Algebraic geometry codes*, Handbook of Coding Theory (V. Pless and W. Huffman, eds.), vol. 1, 1998, p. 871.

[JZ99] R. Johannesson and K. Sh. Zigangirov, *Fundamentals of convolutional coding*, IEEE Press, New York, 1999.

[Mas69] J. L. Massey, *Shift-register synthesis and bch decoding*, IEEE Trans. Information Theory (1969), no. 15, 122–127.

[McE77] Robert J. McEliece, *The theory of information and coding*, Addison-Wesley, 1977.

[McE78] ———, *A public-key cryptosystem based on algebraic coding theory*, Dsn progress report 42-44, Jet Propulsion Laboratory, Pasadena, CA, 1978.

[McE98] ———, *The algebraic theory of convolutional codes*, Handbook of Coding Theory (V. Pless and W. Huffman, eds.), vol. 1, 1998, p. 1065.

[MDIS06] J.M. Muñoz Porras, J.A. Dominguez Perez, J.I. Iglesias Curto, and G. Serrano Sotelo, *Convolutional goppa codes*, IEEE Trans. Inform. Theory **52** (2006), no. 1, 340.

[MDS04] J.M. Muñoz Porras, J.A. Domínguez Pérez, and G. Serrano Sotelo, *Convolutional codes of goppa type*, AAECC **15** (2004), 51.

[MN96] D. J. C. MacKay and R. M. Neal, *Near shannon limit performance of low density parity check codes*, Electronics Letters **32** (1996), no. 18, 1645–1646, Reprinted *Electronics Letters*, vol 33, no 6, 13th March 1997, p.457–458.

[MS67] J. L. Massey and M. K. Sain, *Codes, automata, and continuous systems: Explicit interconnections*, IEEE Trans. Automat. Contr. **AC-12** (1967), no. 6, 644.

[MS68] ———, *Inverses of linear sequential circuits*, IEEE Trans. Comput. (1968), 330.

[MS77] F. J. MacWilliams and N. J. A. Sloane, *The theory of error-correcting codes*, North-Holland, Amsterdam, 1977.

[Mul54] D. E. Muller, *Application of boolean algebra to switching circuit design and error detection*, IRE Trans. Electron. Comp. **EC-3** (1954), 6–12.

[Nit05] Nitin Nitsure, *Construction of hilbert and quot schemes*.

[NLX99] H. Niederreiter, K.Y. Lam, and C.P. Xing, *Constructions of algebraic geometry codes*, IEEE Trans. Inform. Theory **45** (1999), 1186–1193.

[NXL99] H. Niederreiter, C. P. Xing, and K. Y. Lam, *A new construction of algebraic geometry codes*, AAECC (1999), no. 9, 373–381.

[OS99] F. Özbudak and H. Stichtenoth, *Constructing codes from algebraic curves*, IEEE Trans. Inform. Theory **45** (1999), no. 7, 2502–2505.

[Pir88] Ph. Piret, *Convolutional codes, an algebraic approach*, MIT Press, Cambridge, MA, 1988.

[Pre01] O. Pretzel, *Extended classical goppa codes*, AAECC **11** (2001), 447–454.

[Ree54] I.S. Reed, *A class of multiple-error-correcting codes and their decoding scheme*, IRE Trans. on Information Theory **IT-4** (1954), 38–49.

[Ros] Joachim Rosenthal, *Some interesting problems in systems theory which are of fundamental importance in coding theory*, Proceedings of the 36th IEEE Conference on Decision and Control, 1997.

[Ros99] ———, *Dynamical systems, control, coding, computer vision: New trends, interfaces, and interplay*, ch. An algebraic decoding algorithm for convolutional codes, pp. 343–360, Birkäuser, 1999.

[Ros01] ———, *Codes, systems and graphical models*, IMA, vol. 123, ch. Connections between linear systems and convolutional codes, pp. 39–66, Springer-Verlag, 2001.

[RR94] M. S. Ravi and J. Rosenthal, *A smooth compactification of the space of transfer functions with fixed mcmillan degree*, Acta Appl. Math (1994), no. 34, 329–352.

[RR97] M. J. Riley and I. E. G. Richardson, *Digital video communications*, Artech House, 1997.

[RS60] I.S. Reed and G. Solomon, *Polynomial codes over certain finite fields*, Journal Society Indust. Appl. Math. **8** (1960), 300–304.

[RS97] Joachim Rosenthal and Roxana Smarandache, *Construction of convolutional codes using methods from linear systems theory*, 1997.

[RS99] ———, *Maximum distance separable convolutional codes*, AAECC **10** (1999), no. 1, 15.

[RSY96] Joachim Rosenthal, J. M. Schumacher, and E. V. York, *On behaviors and convolutional codes*, IEEE Trans. Inform. Theory **42** (1996), no. 6, 1881.

[RY99] Joachim Rosenthal and E.V. York, *Bch convolutional codes*, IEEE Trans. Inform. Theory **45** (1999), no. 6, 1833–1844.

[SGLR] Roxana Smarandache, Heide Gluesing-Luerssen, and Joachim Rosenthal, *Generalized first order descriptions and canonical forms for convolutional codes*.

[Sha48] C. E. Shannon, *A mathematical theory of communication*, Bell System Technical Journal **27** (1948), 379–423 and 623–656.

[SM69] M. K. Sain and J. L. Massey, *Invertibility of linear time-invariant dynamical systems*, IEEE Trans Automat. Contr. **AC-14** (1969), 141.

[Son98] Eduardo D. Sontag, *Mathematical control theory: Deterministic finite dimensional systems*, second edition ed., Springer, New York, 1998.

[Sud97] Madhu Sudan, *Decoding of reed solomon codes beyond the error-correction bound*, Journal of Complexity **1** (1997), no. 13, 180–193.

[Sud00] ———, *List decoding: Algorithms and applications*, Proceedings of the International Conference IFIP TCS 2000, Sendai, Japan, 17-19 August 2000 (M. Hagiya P.D. Mosses T. Ito J. van Leeuwen, O. Watanabe, ed.), Lecture Notes in Computer Science, vol. 1872, Springer, August 2000, pp. 25–41.

[Tre04] Luca Trevisan, *Some applications of coding theory in computational complexity*, Quaderni di Matematica **13** (2004), 347–424.

[Vit67] A. J. Viterbi, *Error bounds for convolutional codes and an asymptotically optimum decoding algorithm*, IEEE Trans. Inform. Theory **13** (1967), no. 2, 260.

[vL82] J. H. van Lint, *Introduction to coding theory*, Springer-Verlag, 1982.

[XNL99] C. P. Xing, H. Niederreiter, and K. Y. Lam, *A generalization of algebraic-geometry codes*, IEEE Trans. Inform. Theory **45** (1999), no. 7, 2498–2501.