MAXIMUM DISTANCE SEPARABLE CONVOLUTIONAL CODES, CONSTRUCTION AND DECODING

A Dissertation

Submitted to the Graduate School
of the University of Notre Dame
in Partial Fulfillment of the Requirements
for the Degree of

Doctor of Philosophy

by

Roxana Smarandache, B.S., M.S.

Joachim Rosenthal, Director

Department of Mathematics
Notre Dame, Indiana

August 2001

MAXIMUM DISTANCE SEPARABLE CONVOLUTIONAL CODES,

CONSTRUCTION AND DECODING

Abstract

by
Roxana Smarandache

In this dissertation maximum distance separable convolutional codes are introduced and studied.

In particular, the Singleton Bound on the minimum distance of block codes is generalized to an upper bound on the free distance of convolutional codes. The convolutional codes attaining this bound will be called maximum distance separable convolutional codes, or shortly, MDS codes. A general construction of a rate k/n, degree δ, MDS convolutional code will be provided, starting with a large Reed Solomon block code.

Following the same direction, strongly MDS convolutional codes are defined in the case of rate $1/2$. These are codes having optimal column distances. Properties of these codes are given and a concrete construction is provided. This construction has the advantage that the field required has the fewest elements among all other existing constructions. Finally, a decoding algorithm for these codes is given, along with several properties that improve on the time of decoding.

The input-state-output representation of a convolutional code is also introduced. A construction of rate $1/n$ MDS convolutional codes is developed and a theoretical algebraic geometric proof of the existence of a general k/n MDS convolutional code is given using this representation. This proof sets the MDS convolutional codes into the framework of algebraic geometry, which generalizes the algebraic geometric setting for the Reed Solomon block codes.

Finally the class of binary unit memory convolutional codes that are MDS is studied. Some algebraic constructions of convolutional codes over small fields are given.

To my parents,
and to Mihnea,
who are my light and joy,
with all my love.

CONTENTS

ACKNOWLEDGEMENTS

I would like to thank the people who encouraged and supported my work towards this dissertation. First, my advisor, Joachim Rosenthal, who has encouraged and supported me constantly along the way. I have benefited greatly from his advice and friendship, and learned a lot from him. I am grateful to him and his family, and to my nice friends, Heide, Elisa, Feride, Charlie, Hans, Scott, for giving me love and friendship.

I would like to thank the members of my defense committee, Karen Chandler, Michael Gekhtman and Daniel Costello, for their time, effort and observations. I have special thanks to Karen Chandler for her continuous feedback and valuable remarks during the reading of this dissertation. I greatly appreciated and enjoyed her involvement. I would like to thank all of the faculty, staff and graduate students of the Mathematics Department of University of Notre Dame for providing an extraordinary environment.

I would also like to acknowledge all of the institutions which have supported my work. The Department of Mathematics and the Center for Applied Mathematics at Notre Dame for their support and fellowships, also the National Science Foundation, for its grant support.

CHAPTER 1

INTRODUCTION

1.1 Overview

Coding theory has emerged out of the need for better communication and computer data storage and has rapidly developed as a mathematical theory in strong relationship with algebra, combinatorics and algebraic geometry.

Communication is often imperfect. Even a message accurately stated may be altered during transmission; the consequences of the mistake in the interpretation of the message may be unfortunate in financial, diplomatic, military or other domains. As a result, when a message travels from an information source to a destination, both the sender and the receiver would like to assure that the message received is free of error, and if it contains errors, to be able to detect, and ideally, to correct them. This can be achieved by encoding: adding redundancy to the information message in an optimal way, such that any two encoded messages differ substantially. In this way, a relatively small amount of errors will not change an encoded message into another, and thus the errors can at least be detected.

The general problem of communication may be described by the following concise map:

$$\begin{tikzcd} \boxed{\textbf{Source}} \arrow[r] & \boxed{\textbf{Source Encoder}} \arrow[r] & \boxed{\textbf{Channel Encoder}} \arrow[d] \\ & \textbf{Noise} \arrow[r, squiggly] & \boxed{\textbf{Channel}} \arrow[d] \\ \boxed{\textbf{Sink}} & \boxed{\textbf{Source Decoder}} \arrow[l] & \boxed{\textbf{Channel Decoder}} \arrow[l] \end{tikzcd}$$

The source information may be a picture, a sound pattern, a piece of text, etc., that needs to be stored or transmitted reliably over a channel that is more or less noisy. Examples of channels are phone lines, radio links or fiber optic cables.

In 1948, Claude Shannon showed that the communication problem may be broken into two parts, source coding and channel coding. He showed that the goal of finding error correcting codes that allowed for a high probability of successful transmission was attainable. Shannon defined for each channel a constant associated with it, called the *channel capacity*, and he showed that reliable transmission at a rate below capacity is possible. More precisely, his channel coding theorem asserts that there exist error correcting codes that achieve successful transmission with probability arbitrarily close to 1, with the rate of the code arbitrarily close and below the channel capacity.

Shannon's proof is existential and gives little indication of how to obtain such codes. The construction of these codes, along with efficient decoding algorithms, is the goal of modern coding theory.

Today, error control theory is used in many digital systems and its role is becoming more and more important. Some systems could never have been realized without the application of error control coding: the compact disk player is one example. Other examples of applications of error control coding are computer storage devices, satellite communications and implicitly the more recent mobile communication or space communications that transmit observed data from spacecraft to the ground.

Coding theory is naturally portioned into the study of block and convolutional codes. Block codes have been long studied and used before the development of convolutional codes. The rich algebraic structure of block codes has led to the development of algebraic decoding algorithms that are easy to implement, and the codes themselves have very good error correction properties. Convolutional codes generalize the class of block codes in a natural way. Due to their efficient non-algebraic decoding algorithms, they have enjoyed widespread use since their discovery. Nevertheless, the algebraic theory of convolutional codes is not as advanced as the algebraic theory for block codes. In fact, most implemented convolutional codes were found by exhaustive computer searches. From a mathematical point of view, a fundamental issue is to develop a suitable algebraic theory. My thesis develops coding theory in this direction.

The goal of this thesis is to enlarge the algebraic content exhibited on convolutional codes. The approach taken is to extend some results from the theory of block codes to the setting of convolutional codes. In this respect we consider MDS block codes which are a type of codes with very good error correcting property in block coding theory. They are defined as $[n, k]$ linear block codes, over a finite field $\mathbb{F}_q$, that have minimum distance d_{min} equal to the Singleton upper bound $n-k+1$. We generalize this bound to an upper bound on the free distance d_{free} of convolutional codes:

Theorem 1.1 *A rate k/n convolutional code with degree δ has the free distance d_{free} bounded above by*

$$d_free \leq (n-k)\left(\lfloor \delta/k \rfloor + 1\right) + \delta + 1. \tag{1.1}$$

We define MDS convolutional codes as codes with free distance attaining the bound (1.1). The thesis is centered around studying in detail these new codes. We will give a general construction starting from a Reed Solomon code, and then study them in some particular cases.

1.2 Outline of this Dissertation

The outline of this thesis is as following. A brief introduction of block codes and convolutional codes is presented in the first chapter, along with notation and certain representations used in the development of the results. An algebraic description of the link between quasi-cyclic codes and convolutional codes is given ([2, 7, 11, 30]). The following chapters contain the main new results obtained during the dissertation. In Chapter 2 we generalize the Singleton Bound for block codes to convolutional codes, and define the MDS convolutional codes as codes attaining this bound. These codes generalize the MDS block codes in a natural way, and the construction given in Section 2.3 emphasizes this. It starts from a Reed Solomon code of certain parameters and obtains an MDS convolutional code of parameters controlled by the parameters of the starting block code.

Chapter 3 defines a new type of MDS convolutional codes. We call them *strongly MDS*, since they have an extra desired property in error correction: their column distances are optimal. These codes prove to be better than the Reed Solomon codes with the same parameters. In each **sliding** window of length equal to the length of the block code, it can correct the number of errors the block code corrects in a **slotted** window of the same length.

Chapter 4 presents another type of MDS convolutional codes of rate $1/n$ and general degree δ. This chapter has a systems theory approach to coding theory. It uses some first order representations to describe the codes and the results in the systems language.

Motivated by the fact that binary codes are most commonly implemented in practice, Chapter 5 studies binary MDS codes. These codes need to have memory one. We therefore study unit memory MDS codes and present some algebraic binary constructions of k/n codes with $n = 2^{k-1}$. Then we generalize this result to the case of n odd, over larger fields.

Chapter 6 will give a non-constructive proof of the existence of the MDS convolutional codes as it was first given in [24]. This proof has its main value in setting the MDS convolutional codes into the frame of algebraic geometry. This setting generalizes the algebraic geometric setting for the Reed Solomon block codes.

1.3 Block Codes

Let $q = p^n$, p prime.

Definition 1.2 Let $\mathbb{F} = \mathbb{F}_q$ be the finite field with q elements. An $[n, k]$ *linear block code* is algebraically defined as a linear subspace $\mathcal{C} \subset \mathbb{F}^n$, of dimension k. An $[n, k]$ linear block code has associated with it a $k \times n$ matrix G, an $n \times (n-k)$ matrix H, and a short exact sequence:

$$0 \longrightarrow \mathbb{F}^k \xrightarrow{G} \mathbb{F}^n \xrightarrow{H} \mathbb{F}^{n-k} \longrightarrow 0,$$

such that the code $\mathcal{C}$ is given through $\mathcal{C} = \mathrm{im}_{\mathbb{F}}(G) = \ker_{\mathbb{F}}(H)$. The matrix G is called a *generator matrix*, or *encoder*, and the $n \times (n-k)$ matrix H is called a *parity check* matrix. They satisfy the relation: $GH = 0$.

An example of a block code is the code of the ISBN numbers, defined by the 1×10 parity check matrix $H = \begin{bmatrix} 1 & 2 & \ldots & 9 & 10 \end{bmatrix}$, through

$$\mathcal{C} = \{x \in \mathbb{F}_{11}^{10} \mid \sum_{i=1}^{10} i x_i \equiv 0 \bmod 11\} \subset \mathbb{F}_{11}^{10}.$$

The code is a $[10, 9]$ block code over $\mathbb{F}_{11}$.

The most important parameter of a block code is the *minimum distance* d_{min}. If we define the *Hamming weight* of a vector $v \in \mathbb{F}^n$ to be wt v, the number of its nonzero components, the minimum distance of the code is then defined as following:

$$d_{min}(\mathcal{C}) = \min\{\mathrm{wt}\ \ v \mid v \in \mathcal{C} - \{0\}\}.$$

A block code $\mathcal{C}$ having minimum distance d can detect up to $d-1$ errors and can correct up to $\lfloor \frac{d-1}{2} \rfloor$ errors.

A good code should be capable of correcting as many errors as possible, therefore a code with large minimum distance is to be desired. Consequently, the main linear block coding problem asks for the construction of codes with large distance. A natural upper bound on the minimum distance is given by the following:

Lemma 1.3 *(Singleton) An $[n, k]$ linear code $\mathcal{C}$ over any field $\mathbb{F}$ has minimum distance at most $n - k + 1$:*

$$d(\mathcal{C}) \leq n - k + 1. \tag{1.2}$$

Naturally, a code attaining this bound (1.2) will theoretically have the best error correcting capability. Such code is called *maximum distance separable* block or MDS block code.

An example of an MDS block code is the Reed Solomon code defined through the following parity check matrix. Let $\alpha \in \mathbb{F}_q$ be a primitive nth root of unity, with $n = q - 1$. For any $k < N$, consider the $(n - k) \times n$ matrix H of the following form:

$$H := \begin{bmatrix} 1 & \alpha & \alpha^2 & \cdots & \cdots & \alpha^{(n-1)} \\ 1 & \alpha^2 & \alpha^4 & \cdots & \cdots & \alpha^{2(n-1)} \\ \vdots & \vdots & \vdots & & & \vdots \\ 1 & \alpha^{n-k} & \alpha^{2(n-k)} & \cdots & \cdots & \alpha^{(n-1)(n-k)} \end{bmatrix}.$$

The code having matrix H as a parity check matrix has minimum distance $n-k+1$, therefore is an MDS code. It is called Reed Solomon and has the algebraic structure of a cyclic code. A polynomial generator is $g(x) = (x - \alpha)(x - \alpha^2)\dots(x - \alpha^{n-k})$. The Reed Solomon codes are very good codes due to their cyclic structure and large error correcting capability, but also due to an efficient algebraic decoding algorithm based on their nice algebraic structure. These codes are efficiently implemented in practice, on compact discs.

1.4 Convolutional Codes

If we introduce the delay operator to represent the message,

$$u(D) := \sum_{i=0}^{t} u_i D^i \in \mathbb{F}^k[D],$$

the codeword,

$$v(D) := \sum_{i=0}^{t} v_i D^i \in \mathbb{F}^n[D],$$

then the block code may be described by:

$$\mathcal{C} = \{v(D) = u(D)G\} = \mathrm{im}_{F[D]} G.$$

If we replace the scalar matrix G by a polynomial matrix of degree m

$$G(D) = (g_{ij}(D))_{i,j} = G_0 + G_1 D + \ldots + G_m D^m$$

then we obtain a *convolutional code.*

Definition 1.4 Let $\mathbb{F}[D]$ be the polynomial ring and $\mathbb{F}(D)$ the field of rational functions. Let $G(D)$ be a $k \times n$ matrix over the polynomial ring $\mathbb{F}[D]$, with $\operatorname{rank} G(D) = k$. We define the *rate k/n convolutional code* generated by $G(D)$ as the set

$$\mathcal{C} = \{u(D)G(D) \in \mathbb{F}^n(D) \mid u(D) \in \mathbb{F}^k(D)\}$$

and say that $G(D)$ is a *generator matrix* or an *encoder* for the convolutional code $\mathcal{C}$.

If matrices $G(D)$ and $G'(D)$ both generate the same convolutional code $\mathcal{C}$, then there exists a $k \times k$ invertible matrix $U(D)$ with $G'(D) = U(D)G(D)$ and we say $G(D)$ and $G'(D)$ are equivalent encoders.

Hence we may assume without loss of generality that the code $\mathcal{C}$ is represented by a *minimal basic encoder* $G(D)$. For this, let ν_i be the ith row degree of $G(D)$, i.e. $\nu_i = \max_j \deg g_{ij}(D)$. In the literature [6] the index ν_i is also called the *constraint length for the ith input* of the matrix $G(D)$. Then one defines:

Definition 1.5 A polynomial generator matrix $G(D)$ is called *basic* if it has a polynomial right inverse. It is called *minimal* if the sum $\sum_{i=1}^k \nu_i$ of its indices attains the minimal value among all generator matrices of $\mathcal{C}$.

A basic generator matrix is automatically *non-catastrophic.* This means that finite weight codewords can only be produced from finite weight messages. If $G(D)$ is a minimal basic encoder one defines the *degree* [17] of $\mathcal{C}$ as the number $\delta := \sum_{i=1}^k \nu_i$. In the literature the degree δ is sometimes also called the *total memory* [12], or the *overall constraint length* [6], or the *complexity* [18] of the minimal basic generator matrix $G(D)$, a number only dependent on $\mathcal{C}$. Among all these equivalent expressions we prefer the term degree best since it relates naturally to equal objects appearing in systems theory and algebraic geometry. The following remarks explain this:

Remark 1.6 It has been shown by Forney [4] that the set $\{\nu_1, \ldots, \nu_k\}$ of row degrees is the same for all minimal basic encoders of $\mathcal{C}$. Because of this reason McEliece [17] calls these indices the Forney indices of the code $\mathcal{C}$. These indices are also the same as the Kronecker indices of the row-module $\mathcal{M} = \{u(D)G(D) \in \mathbb{F}^n[D] \mid u(D) \in \mathbb{F}^k[D]\}$, when $G(D)$ is a basic encoder. The Pontryagin dual of $\mathcal{M}$ defines a linear time invariant behavior in the sense of Willems [23, 31], i.e. a linear system. Under this identification, the Kronecker indices of $\mathcal{M}$ correspond to the observability indices of the linear system [22]. The sum of the observability indices is equal to the McMillan degree of the system. Finally $\mathcal{M}$ defines in a natural way a quotient sheaf [19] over the projective line. In this context, one refers to the indices $\{\nu_1, \ldots, \nu_k\}$ as the Grothendieck indices of the quotient sheaf, and to $\delta = \sum_{i=1}^k \nu_i$ as the degree of the quotient sheaf.

We think that the degree is the single most important code parameter besides the transmission rate k/n. In the sequel we will adopt the notation used by McEliece [17, p. 1082] and denote by (n, k, δ) a rate k/n convolutional code of degree δ.

We define the weight wt $(v(D))$ of a vector $v(D) \in \mathbb{F}^n(D)$ as the sum of the weights of all its $\mathbb{F}^n$-coefficients. We define the *free distance* of the convolutional code $\mathcal{C} \subset \mathbb{F}^n(D)$ through

$$d_{\text{free}} = \min\{\text{wt } (v(D)) \mid v(D) \in \mathcal{C}, v(D) \neq 0\}. \tag{1.3}$$

It is an easy but crucial observation that in the case of a basic encoder $G(D)$, the free distance may also be obtained as

$$d_{\text{free}} = \min\{\text{wt } (v(D)) \mid v(D) \in \mathcal{M}, v(D) \neq 0\}.$$

This follows simply from the fact that, if $G(D)$ has a polynomial right inverse, a non-polynomial message $u(D)$ would result in a non-polynomial codeword $v(D)$, which, of course, has infinite weight.

In the rest of this section we link the free distance to two types of distances known in the literature. We will use these distances to obtain an upper bound on the free distance of a convolutional code and to construct codes attaining this bound.

Following the approach in [5, 6] we define the column distances d_j^c and the row distances d_j^r. In order to do so let us suppose $G(D) = G_0 + G_1 D + G_2 D^2 + \ldots + G_{\nu_k} D^{\nu_k}$ is an encoder with row degrees $\nu_1 \leq \ldots \leq \nu_k$. We denote by

$$G = \begin{bmatrix} G_0 & G_1 & \ldots & G_{\nu_1} & G_{\nu_1+1} & \ldots & G_{\nu_k} & & & \\ & G_0 & G_1 & \ldots & G_{\nu_1} & G_{\nu_1+1} & \ldots & G_{\nu_k} & & \\ & & \ddots & \ddots & & \ddots & \ddots & & \ddots & \end{bmatrix} \tag{1.4}$$

the *semi-infinite sliding generator matrix.* Then the convolutional code may be defined as

$$\mathcal{C} = \{(u_0, u_1, \ldots, u_\gamma, \ldots) \cdot G \mid u_j \in \mathbb{F}^k, \text{ for } j = 0, 1, \ldots\}.$$

The jth *order column distance* d_j^c is defined as the minimum of the weights of the truncated codewords $v_{[0,j]} := (v_0, v_1, \ldots, v_j)$ resulting from an information sequence $u_{[0,j]} := (u_0, u_1, \ldots, u_j)$ with $u_0 \neq 0$. Precisely, if G_j^c denotes the $k(j+1) \times n(j+1)$ upper-left sub-matrix of the semi-infinite matrix G, then

$$d_j^c = \min_{u_0 \neq 0} \text{wt } (u_{[0,j]} \cdot G_j^c).$$

The quantity $d_{\nu_k}^c$ is called the *minimum distance* of the code and the tuple

$$d^{\text{p}} = [d_0^c, d_1^c, \ldots, d_{\nu_k}^c]$$

is called the *distance profile.* The limit $d_\infty^c = \lim_{j \to \infty} d_j^c$ exists and we have the relation $d_0^c \leq d_1^c \leq \ldots \leq d_\infty^c$. Then d_∞^c is the minimal distance computed over all finite or infinite codewords of $\mathcal{C}$. It is shown in [6] that $d_\infty^c = d_{\text{free}}$.

The jth *row distance* d_j^r is defined as the minimum of the weights of all the finite codewords $v_{[0,j+\nu_k]} := (v_0, v_1, \ldots, v_{j+\nu_k})$ resulting from an information sequence $u_{[0,j]} := (u_0, u_1, \ldots, u_j) \neq 0$. Thus, if we denote by G_j^r the $k(j+1) \times n(j+\nu_k+1)$ upper-left sub-matrix of the semi-infinite matrix G, the jth row distance is

$$d_j^r = \min_{u_{[0,j]} \neq 0} \text{wt } (u_{[0,j]} \cdot G_j^r). \tag{1.5}$$

The limit $d_\infty^r = \lim_{j\to\infty} d_j^r$ exists and one has (see e.g. [6]) for every encoder $G(D)$ the relation:

$$d_0^c \leq d_1^c \leq \ldots \leq d_\infty^c = d_{\text{free}} \leq d_\infty^r \leq \ldots \leq d_1^r \leq d_0^r. \tag{1.6}$$

In terms of state space descriptions [25, 22] d_∞^r is equal to the minimal weight of a nonzero trajectory that starts from and returns to the all zero state, and d_∞^c is equal to the minimal weight of a nonzero trajectory that starts from and not necessarily returns to the all zero state. Furthermore, if the generator matrix $G(D)$ is minimal basic, then $d_\infty^c = d_\infty^r = d_{\text{free}}$ (see [25, 6] for details). It follows that for a basic encoder the minimal weight codewords are generated by finite information sequences.

1.5 Quasi-cyclic Codes: a Link between Block Codes and Convolutional Codes

After previously defining block codes and convolutional codes, in this section we describe a link between the two types: the quasi-cyclic codes. More exactly we describe a correspondence:

Cyclic Codes—Quasi cyclic Codes—Convolutional Codes.

The approach we take follows the lines of [7, 11, 30, 2]. New is the discussion of the possible representations of a block code and of the relationship that exists between them in the special case of quasi-cyclic codes and of cyclic codes. Also the fact that a cyclic code may be represented as a quasi-cyclic code is proved a little differently. We will look only at a polynomial generator matrix and rewrite it in a special way that brings us directly to the polynomial description of a generator matrix of a quasi-cyclic code. This differs from the approach taken in the literature, which analyzes instead the scalar generator matrices.

As an $\mathbb{F}$-subspace of $\mathbb{F}^N$ of rank K, a general $[N, K]$ linear block code $\mathcal{C}$ has different representations. Thus if $n \mid N$, $N = nm$, using the vector space isomorphism:

$$\mathbb{F}^N \xrightarrow{\sim} \frac{\mathbb{F}[X]}{X^N - 1} \xrightarrow{\sim} \left(\frac{\mathbb{F}[X]}{X^m - 1}\right)^n \xrightarrow{\sim} (\mathbb{F}^m)^n,$$

we may describe a code word $v \in \mathcal{C} \subset F^N$ in the following ways:

$$v = (v_0, \ldots, v_{N-1}) \tilde{\mapsto} v(X) = v_0 + \ldots + v_{N-1}X^{N-1} \tilde{\mapsto} \bar{v}(X) = (\bar{v_0}(X), \ldots, \bar{v}_{n-1}(X)). \tag{1.7}$$

Here $\bar{v}_0(X^n) + \bar{v}_1(X^n)X + \ldots + \bar{v}_{n-1}(X^n)X^{n-1} = v_0 + v_1X + \ldots + v_{N-1}X^{N-1} = v(X)$. In other words the $\bar{v}(X)$ representation is obtained from $v(X)$ by grouping together the coefficients of the powers of X that differ by a multiple of n and then forming with them the polynomials $\bar{v}_j(X)$ of degrees at most m. Corresponding to these three representations we have three different types of generator matrices: a $K \times N$ scalar matrix, a $K \times 1$ polynomial matrix, with polynomial entries of degree less than $N-1$ and multiplication taken $\mod (X^N-1)$, and a $K \times n$ polynomial matrix with entries of degree less than m and multiplication made $\mod (X^m-1)$:

$$G_{K\times N} = \begin{bmatrix} g_{0,0} & \cdots & g_{0,N-1} \\ \vdots & \ldots & \vdots \\ g_{K-1,0} & \cdots & g_{K-1,N-1} \end{bmatrix}, \tag{1.8}$$

$$G_{K\times 1}(X) = \begin{bmatrix} g_{0,0} & + & \ldots & + & g_{0,N-1}X^{N-1} \\ \vdots & & \ldots & & \vdots \\ g_{K-1,0} & + & \ldots & + & g_{K-1,N-1}X^{N-1} \end{bmatrix} \text{ and} \tag{1.9}$$

$$\bar{G}_{K\times n}(X) = \begin{bmatrix} \bar{g}_{0,0}(X) & \ldots & \bar{g}_{0,n-1}(X) \\ \vdots & \ldots & \vdots \\ \bar{g}_{K-1,0}(X) & \ldots & \bar{g}_{K-1,n-1}(X) \end{bmatrix}. \tag{1.10}$$

The relationship between the last two matrices is given by:

$$\bar{g}_{i,0}(X^n)+\bar{g}_{i,1}(X^n)X+\ldots+\bar{g}_{i,n-1}(X^n)X^{n-1} = g_{i,0}+g_{i,1}X+\ldots+g_{i,N-1}X^{N-1}, \tag{1.11}$$

for $i = \overline{0, K-1}$.

1.5.1 Quasi-cyclic Codes

The code $\mathcal{C} \subset \mathbb{F}^n$ has a natural $\mathbb{F}$-module structure. Let us now define an extra $\mathbb{F}[X^n]$-module structure on $\mathcal{C}$, where $n \mid N$, through:

$$X^n(\underbrace{v_0,\ldots,v_{n-1}}_{n},\underbrace{v_n,\ldots,v_{2n-1}}_{n},\ldots,\underbrace{v_{(m-1)n},\ldots,v_{mn-1}}_{n}) =$$

$$= (\underbrace{v_{(m-1)n},\ldots,v_{mn-1}}_{n},\underbrace{v_0,\ldots,v_{n-1}}_{n},\ldots,\underbrace{v_{(m-2)n},\ldots,v_{(m-1)n-1}}_{n}).$$

In this way we obtain a quasi-cyclic code invariant under the n-cyclic shifts. Because of this invariance, by permuting the components of the codewords as we showed above we can find a $k \times n$ generator matrix

$$G = \begin{bmatrix} G_{0,0} & \ldots & G_{0,n-1} \\ \vdots & \ldots & \vdots \\ G_{k-1,0} & \ldots & G_{k-1,n-1} \end{bmatrix}$$

where each $G_{i,j}$ is a circulant matrix of size $m \times m$. To each $G_{i,j}$ circulant matrix may be associated a polynomial $\bar{g}_{i,j}(X)$ of degree at most m so that the matrix $\bar{G}_{K\times n}(X)$ from the above representation has this special block form:

$$\bar{G}_{K\times n} = \begin{bmatrix} \begin{pmatrix} \bar{g}_{0,0}(X) \\ X\bar{g}_{0,0}(X) \\ \vdots \\ X^{m-1}\bar{g}_{0,0}(X) \end{pmatrix} & \cdots & \begin{pmatrix} \bar{g}_{0,n-1}(X) \\ X\bar{g}_{0,n-1}(X) \\ \vdots \\ X^{m-1}\bar{g}_{0,n-1}(X) \end{pmatrix} \\ \vdots & \cdots & \vdots \\ \begin{pmatrix} \bar{g}_{k-1,0}(X) \\ X\bar{g}_{k-1,0}(X) \\ \vdots \\ X^{m-1}\bar{g}_{k-1,0}(X) \end{pmatrix} & \cdots & \begin{pmatrix} \bar{g}_{k-1,n-1}(X) \\ X\bar{g}_{k-1,n-1}(X) \\ \vdots \\ X^{m-1}\bar{g}_{k-1,n-1}(X) \end{pmatrix} \end{bmatrix} \bmod (X^m - 1). \tag{1.12}$$

In each polynomial block $\mod (X^m - 1)$ of the matrix $\bar{G}_{K\times n}(X)$ the coefficients of the entries form the circulant block matrices $G_{i,j}$. Therefore the multiplication $(u_0, \ldots, u_{k-1}) \cdot G_{i,j}$ is equivalent to the polynomial multiplication

$$(u_0 + u_1 X + \ldots + u_{k-1}X^{k-1}) \cdot \bar{g}_{i,j}(X) \quad \mod (X^m - 1).$$

Hence the code can be described by the $k \times n$ polynomial matrix

$$\bar{G}_{k\times n}(X) = (\bar{g}_{i,j}(X))_{\substack{i=\overline{0,k-1} \\ j=\overline{0,n-1}}} = \begin{bmatrix} \bar{g}_{0,0}(X) & \ldots & \bar{g}_{0,n-1}(X) \\ \vdots & \ldots & \vdots \\ \bar{g}_{k-1,0}(X) & \ldots & \bar{g}_{k-1,n-1}(X) \end{bmatrix} \tag{1.13}$$

as follows:

$$\mathcal{C} = \{(\bar{v}_0(X), \ldots, \bar{v}_{n-1}(X)) = (u_0(X), \ldots, u_{k-1}(X)) \cdot \bar{G}_{k\times n}(X) \bmod (X^m - 1)\}. \tag{1.14}$$

As above we have that the coefficients of

$$\bar{v}_0(X^n) + \bar{v}_1(X^n)X + \ldots + \bar{v}_{n-1}(X^n)X^{n-1} = v(X)$$

form the components of the codeword v corresponding to $v(X)$ through 1.7.

Also we see that:

$$v(X) = u_0(X^n)\left(\bar{g}_{0,0}(X^n) + \bar{g}_{0,1}(X^n)X + \ldots + \bar{g}_{0,n-1}(X^n)X^{n-1}\right) + \ldots +$$

$$+u_{k-1}(X^n)\left(\bar{g}_{k-1,0}(X^n)+\bar{g}_{k-1,1}(X^n)X+\ldots+\bar{g}_{k-1,n-1}(X^n)X^{n-1}\right)=$$

$$=(u_0(X^n),\ldots,u_{k-1}(X^n))\begin{bmatrix} \bar{g}_{0,0}(X^n) & + & \ldots & + & \bar{g}_{0,n-1}(X^n)X^{n-1} \\ & & \vdots & & \\ \bar{g}_{k-1,0}(X^n) & + & \ldots & + & \bar{g}_{k-1,n-1}(X^n)X^{n-1} \end{bmatrix}, \tag{1.15}$$

where the multiplications are done $\bmod(X^N-1)$.

1.5.2 Cyclic Codes

If $n=1$ then the code $\mathcal{C}$ has an $\mathbb{F}[X]$-module structure, in other words is invariant under the 1-cyclic shifts. Therefore $\mathcal{C}$ is a cyclic code. Then there exist generator matrices with the following representations of (1.10):

$$G_{K\times N}=\begin{bmatrix} g_0 & g_1 & \cdots & g_{N-1} \\ g_{N-1} & g_0 & \cdots & g_{N-2} \\ \vdots & \vdots & \cdots & \vdots \\ g_{N-K+1} & g_{N-K+2} & \cdots & g_{N-K} \end{bmatrix}, G_{K\times 1}(X)=\begin{bmatrix} g(X) \\ Xg(X) \\ \vdots \\ X^{K-1}g(X) \end{bmatrix}.$$

The polynomial $g(X)=g_0+g_1X+\ldots+g_{N-K}X^{N-K}$ is called a *generator polynomial* of the cyclic code $\mathcal{C}$. Let n be a divisor of N other than 1 (we denote it also by n for easier writing). After expanding $G_{K\times 1}(X)$ to an $N\times 1$ polynomial matrix, by taking all the powers of X, the last representation $\bar{G}_{K\times n}(X)$ of (1.10) becomes:

$$\bar{G}_{K\times n}(X)=\begin{bmatrix} \bar{g}_0(X) & \bar{g}_1(X) & \ldots & \bar{g}_{n-1}(X) \\ X\bar{g}_{n-1}(X) & \bar{g}_0(X) & \ldots & \bar{g}_{n-2}(X) \\ X\bar{g}_{n-2}(X) & X\bar{g}_{n-1}(X) & \ldots & \bar{g}_{n-3}(X) \\ \vdots & \vdots & \ldots & \vdots \\ X\bar{g}_0(X) & X\bar{g}_1(X) & \ldots & X\bar{g}_{n-1}(X) \\ X^2\bar{g}_{n-1}(X) & X\bar{g}_0(X) & \ldots & X\bar{g}_{n-2}(X) \\ \vdots & \vdots & \ldots & \vdots \\ X^{m-1}\bar{g}_0(X) & X^{m-1}\bar{g}_1(X) & \ldots & X^{m-1}\bar{g}_{n-1}(X) \end{bmatrix} \bmod (X^m-1).$$

Here $g(X)=\bar{g}_0(X^n)+\bar{g}_1(X^n)X+\ldots+\bar{g}_{n-1}(X^n)X^{n-1}$. Performing some permutations on the rows of $\bar{G}_{K\times n}(X)$ we get the polynomial block representation of (1.12).

After cutting the linearly dependent rows among the last rows, the $k \times n$ representation $G_{k\times n}(X)$ of (1.13) has the special form:

$$G_{k\times n}(X) = \begin{bmatrix} \bar{g}_0(X) & \bar{g}_1(X) & \dots & \bar{g}_{n-1}(X) \\ X\bar{g}_{n-1}(X) & \bar{g}_0(X) & \dots & \bar{g}_{n-2}(X) \\ \vdots & \vdots & \dots & \vdots \\ X\bar{g}_{n-k+1}(X) & X\bar{g}_{n-k+2}(X) & \dots & \bar{g}_{n-k}(X) \end{bmatrix} \mod (X^m - 1). \tag{1.16}$$

As we have seen in (1.14) the code $\mathcal{C}$ is given by:

$$\mathcal{C} = \{(u_0(X), \dots, u_{k-1}(X)) \cdot \bar{G}_{k\times n}(X) \mod (X^m - 1)\} = \tag{1.17}$$

$$= \{(u_0(X^n), \dots, u_{k-1}(X^n)) \begin{bmatrix} g(X) \\ Xg(X) \\ \vdots \\ X^{k-1}g(X) \end{bmatrix} \mod (X^N - 1)\} = \tag{1.18}$$

$$= \{(u_0(X^n) + u_1(X^n)X + \dots + u_{k-1}(X^n)X^{k-1})g(X) \mod (X^N - 1)\}. \tag{1.19}$$

In other words we have that a cyclic code (N, K) with $N = nm, n \neq 1$ is also a quasi-cyclic code invariant to the n-cyclic shift and with the special representations given by (1.17), (1.18), (1.19).

1.5.3 Convolutional codes

Looking at the polynomial representation (1.14) and (1.16) of the quasi-cyclic, respectively cyclic, block codes of sizes (N, K), $N = nm, K \leq km$, it is easy to make the connection with the (n, k) convolutional codes. By dropping " $\mod (X^m - 1)$" in the multiplication of the message $(u_0(X), \dots, u_{k-1}(X))$ with the $k \times n$ polynomial generator matrix $\bar{G}_{k\times n}(X)$ in (1.14) and (1.16), and changing the shift operator X into the delay operator D, we obtain an (n, k) convolutional code generated by $G(D) := \left(\bar{G}_{k\times n}(X)\right)_{X=D}$ through:

$$\mathcal{C} = \{(u_0(D), \dots, u_{k-1}(D)) \cdot G(D) \mid (u_0(D), \dots, u_{k-1}(D)) \in \mathbb{F}^k[D]\}$$

$$= \{(u_0(D), \dots, u_{k-1}(D)) \begin{bmatrix} \bar{g}_0(D) & \bar{g}_1(D) & \dots & \bar{g}_{n-1}(D) \\ D\bar{g}_{n-1}(D) & \bar{g}_0(D) & \dots & \bar{g}_{n-2}(D) \\ \vdots & \vdots & \dots & \vdots \\ D\bar{g}_{n-k+1}(D) & D\bar{g}_{n-k+2}(D) & \dots & \bar{g}_{n-k}(D) \end{bmatrix} \} \tag{1.20}$$

$$= \left\{ \left(u_0(D^n) + u_1(D^n)D + \ldots + u_{k-1}(D^n)D^{k-1} \right) g(D) \right\} \tag{1.21}$$

where

$$g(D) = \bar{g}_0(D^n) + \bar{g}_1(D^n)D + \ldots + \bar{g}_{n-1}(D^n)D^{n-1}. \tag{1.22}$$

In fact, the relations (1.15) and (1.18) without $\mod (X^N - 1)$ represent the description we find in [12] of a convolutional code through the polynomial generators. The above correspondence is therefore natural.

The further relation between the minimum distance of the block code and the free distance of the convolutional code, or between the rate of the two codes is discussed in detail in [30, 11] and [7]. The authors of the first two papers choose to start the construction with a parity check matrix instead of a generator matrix, avoiding in this way the discussion on the catastrophicity of the convolutional code thus obtained. Justesen starts with a convolutional code generated by a matrix of the form in (1.16) (without the modulo multiplication) and discusses some cases in which the free distance of this code is bounded below by the minimum distance of the cyclic block code generated by $g(X)$. The next chapter will use one of his results.

1.6 Certain First Order Representations for Convolutional Codes

This section reviews some first order representations for convolutional codes that will be used in the fourth chapter. As shown in [23] we have the following existence and uniqueness theorems:

Theorem 1.7 *[24] Assume $\mathcal{C} \subset \mathbb{F}^n[D]$ is a rate k/n convolutional code of degree δ. Then there exist matrices $K, L \in \mathbb{F}^{(\delta+n-k)\times\delta}$ and $M \in \mathbb{F}^{(\delta+n-k)\times n}$ such that:*

$$\mathcal{C} = \{v(D) \in \mathbb{F}^n[D] \mid \ (\exists)\ x(D) \in \mathbb{F}^\delta[D] : (DK + L)x(D) + Mv(D) = 0\}. \tag{1.23}$$

Moreover, the following conditions are satisfied:

1. *K has full column rank;*
2. *$(K \mid M)$ has full row rank;*
3. *rank $(D_0K + L \mid M) = \delta + n - k, \forall D_0 \in \mathbb{K}$, where $\mathbb{K}$ is the algebraic closure of $\mathbb{F}$.*

The theorem allows one to work with matrix triples (K, L, M) rather than of a polynomial description. A convolutional code that is described by the matrices (K, L, M) will be simply denoted by $\mathcal{C}(K, L, M)$. If $\delta = 0$, (1.23) reduces to the parity check equation $Mv(D) = 0$. The representation (1.23) is unique in the following sense:

Theorem 1.8 *[24] Let (K, L, M) and (K', L', M') be two matrix triples with the sizes as in the previous theorem and satisfying the minimality conditions* $1, 2, 3$.

Then $\mathcal{C}(K, L, M) = \mathcal{C}(K', L', M')$ if and only if

$$(K', L', M') = (SKT^{-1}, SLT^{-1}, SM) \tag{1.24}$$

for some $T \in Gl_\delta(\mathbb{F})$ and $S \in Gl_{\delta+n-k}(\mathbb{F})$.

Starting with a (K, L, M) representation for a convolutional code $\mathcal{C}$ we may derive an input/state/output representation. Performing a suitable similarity transformation and permutation of the components of $v(D)$ we may rewrite the (K, L, M) matrix triple in the following way (compare with [23, Section IV]):

$$K = \begin{bmatrix} I_\delta \\ 0 \end{bmatrix}, L = \begin{bmatrix} -\mathbf{A} \\ -\mathbf{C} \end{bmatrix}, M = \begin{bmatrix} 0 & -\mathbf{B} \\ I_{n-k} & -\mathbf{D} \end{bmatrix}.$$

In the partitioning, $\mathbf{A} \in \mathbb{F}^{\delta\times\delta}, \mathbf{B} \in \mathbb{F}^{\delta\times k}, \mathbf{C} \in \mathbb{F}^{(n-k)\times\delta}$ and $\mathbf{D} \in \mathbb{F}^{(n-k)\times k}$. Let:

$$x(D) = x_0 D^\gamma + x_1 D^{\gamma-1} + \ldots + x_\gamma;\ x_t \in \mathbb{F}^\delta, t = 0, \ldots, \gamma,$$

$$v(D) = v_0 D^\gamma + v_1 D^{\gamma-1} + \ldots + v_\gamma;\ v_t \in \mathbb{F}^n, t = 0, \ldots, \gamma.$$

If one partitions the vector v_t into $v_t = \binom{y_t}{u_t}$, where y_t has $n-k$ components and u_t has k components, then the convolutional code is equivalently described by the familiar-looking $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$ representation:

$$\begin{aligned} x_{t+1} &= \mathbf{A}x_t + \mathbf{B}u_t \\ y_t &= \mathbf{C}x_t + \mathbf{D}u_t, \quad x_0 = 0,\ x_{\gamma+1} = 0. \end{aligned} \tag{1.25}$$

This system is known as the input/state/output representation for a convolutional code. It describes the dynamics for a *systematic and rational encoder*. We refer to [23, 25, 32] for more details.

We say that the matrices $\mathbf{A}, \mathbf{B}$ form a *controllable* pair if

$$\text{rank}\ \begin{bmatrix}\mathbf{B}\ \mathbf{AB} \ldots\ \mathbf{A}^{\delta-1}\mathbf{B}\end{bmatrix} = \delta,$$

and we say that $\mathbf{A}, \mathbf{C}$ form an *observable pair* if $\mathbf{A}^t, \mathbf{C}^t$ is a controllable pair. Once $\mathbf{A}, \mathbf{B}$ form a controllable pair and $\mathbf{A}, \mathbf{C}$ form an observable pair then, as shown in [23, 25, 32], the system (1.25) represents a non-catastrophic convolutional code of degree δ and rate k/n.

CHAPTER 2

MDS CONVOLUTIONAL CODES

Convolutional codes naturally generalize block codes. Therefore it makes sense to study them in parallel. Among block codes the maximum distance separable ones (MDS block codes) are codes with very good error correcting capability. Endowed with a fast and low complexity algebraic decoding algorithm, these codes become very efficient in practice. A parallel study to convolutional coding theory raises then the question of the existence of analagous convolutional codes. In order to answer this question we first need to find an upper bound on the free distance of rate k/n and degree δ codes, bound that would generalize the Singleton bound (1.2) for block codes. This constitutes the aim of this chapter. Then we define MDS convolutional codes as codes attaining this bound. McEliece [17] calls codes having the largest free distance among all (n, k, δ) codes *distance optimal*. We call them MDS since they generalize naturally the MDS block codes. In Section 2.3 we give a general construction of a k/n rate, degree δ MDS code, starting from a Reed Solomon block code.

2.1 The Generalized Singleton Bound

Once the row degrees $\nu_1, \ldots, \nu_k$ of the minimal basic encoder $G(D)$ are specified one has a natural upper bound on the free distance of a convolutional code. The following result was derived in [24].

Theorem 2.1 *Let $\mathcal{C}$ be a rate k/n convolutional code generated by a minimal-basic encoding matrix $G(D)$. Let $\nu_1, \ldots, \nu_k$ be the row degrees of $G(D)$ and $\nu = \min\{\nu_1, \ldots, \nu_k\}$ denote the value of the smallest row degree. Finally, let ℓ be the number of indices ν_i among the indices $\nu_1, \ldots, \nu_k$ having the value ν. Then the free distance must satisfy*

$$d_{\text{free}} \leq n(\nu + 1) - \ell + 1. \tag{2.1}$$

We give two proofs for a better intuition of the bound. One uses the sliding matrix G introduced in (1.4) and the other is based on the polynomial generator matrix $G(D)$.

Proof: Without loss of generality we may assume $\nu = \nu_1 \le \ldots \le \nu_k$. Let G be the infinite sliding generator matrix associated to $G(D)$ as in (1.4). We show that the bound (2.1) is actually a bound on the 0-th row distance d_0^r defined in (1.5), in other words we show that $d_0^r \le n(\nu+1) - \ell + 1$. From this the claim follows using (1.6). To prove the bound on d_0^r, we only need to look at the first block-row of the sliding matrix G denoted by $G_0^r = [G_0\ G_1\ \ldots\ G_{\nu_1}\ G_{\nu_1+1}\ \ldots\ G_{\nu_k}]$. For each $j = 0, \ldots, \nu_k$, let G'_j denote the $\ell \times n$ matrix formed by the first ℓ rows of the matrix G_j. All matrices $G'_{\nu_1+1}, \ldots, G'_{\nu_k}$ are zero. Hence the minimum distance d_0^r of the $[n(\nu_k+1), k]$ block code generated by $[G_0\ G_1\ \ldots\ G_{\nu_1}\ G_{\nu_1+1}\ \ldots\ G_{\nu_k}]$ is smaller than the minimum distance of the $[n(\nu+1), \ell]$ block code generated by $G'^r_0 := [G'_0\ G'_1\ \ldots\ G'_{\nu_1}]$, which is upper bounded by the Singleton bound $n(\nu+1) - \ell + 1$. Therefore we obtain the desired bound on d_0^r and hence on the free distance:

$$d_{\text{free}} = d_\infty^r \le \ldots \le d_2^r \le d_1^r \le d_0^r \le n(\nu+1) - \ell + 1.0$$

$\square$

The second proof uses the polynomial description of the code.

Proof: Let G_∞ be the leading coefficient matrix of $G(D)$. After some possible permutation of the rows of $G(D)$ we may use elementary column operations to transform the last ℓ columns of the matrix G_∞ into a matrix $\left[\begin{smallmatrix} I_\ell \\ M \end{smallmatrix}\right]$ where M is a matrix of size $(n-\ell) \times \ell$ over $\mathbb{F}$. The operations may be done through an invertible matrix $T \in Gl_\ell$ which acts on the last ℓ columns of the matrix $G(D)$. This transformation has no effect on the column space of $G(D)$ and it also does not affect the column degrees ν_i. The last column of the new generator matrix $G(D)$ has now $(\ell-1)$ polynomials of weight strictly less than $\nu_k + 1$, one with weight exactly $\nu_k + 1$, and the remaining $(n-\ell)$ polynomials with weight less than or equal to $\nu_k + 1$. Therefore the input $(0, 0, \ldots, 0, 1)^t$ gives a codeword with weight less than or equal to

$$(\ell-1)\nu_k + (\nu_k+1) + (n-\ell)(\nu_k+1) = n(\nu_k+1) - \ell + 1.$$

This gives the upper bound (2.1). $\square$

Remark 2.2 Theorem 2.1 may also be derived from [17, Theorem 4.4] and [17, Corollary 4.3].

In the case of a block code, i. e. when $\nu = 0$ and $\ell = k$, the upper bound in (2.1) is identical to the Singleton-bound (1.2).

It is easy to see that for given n, k, and δ the upper bound (2.1) is maximized if and only if ν is as large as possible while ℓ is as small as possible, which results into

$$\nu = \lfloor \delta/k \rfloor = \nu_1 = \ldots = \nu_\ell < \nu_{\ell+1} = \ldots = \nu_k = \lfloor \delta/k \rfloor + 1 = \nu + 1. \qquad (2.2)$$

We will call the above set of indices the *generic set of row degrees* as they are sometimes referred to in the systems literature.

Remark 2.3 McEliece [17, p. 1083] calls a code $\mathcal{C}$ having the generic set of row degrees *compact*. In systems theory the set of row degrees $\nu_1, \ldots, \nu_k$ correspond to the *observability indices* of the associated (Pontryagin dual) linear system. (Compare with Remark 2.2 and [22]). It is known that the set of all linear systems having a fixed input number k, a fixed output number $n-k$ and a fixed McMillan degree δ has in a natural way the structure of a smooth projective variety [19]. The subset of systems having the generic set of row degrees forms a Zariski open subset of this variety.

Applying the above result to a generator matrix with row degrees equal to the generic set of degrees, we obtain the following upper bound in terms of the degree δ.

Theorem 2.4 *For every base field $\mathbb{F}$ and every rate k/n convolutional code $\mathcal{C}$ of degree δ, the free distance is bounded by*

$$d_{\text{free}} \leq (n-k)\left(\lfloor \delta/k \rfloor + 1\right) + \delta + 1. \tag{2.3}$$

The following result will be proved and discussed in detail in the next chapter:

Theorem 2.5 *For any positive integers $k < n$, δ and for any prime p there exists a field F_q of characteristic p and a rate k/n convolutional code $\mathcal{C}$ of degree δ over $\mathbb{F}_q$, whose free distance is equal to the upper bound* (2.3).

Based on Theorem 2.4 and Theorem 2.5 we introduce the following notions.

Definition 2.6 The upper bound (2.3) is called the *generalized Singleton bound.* A rate k/n code of degree δ whose free distance achieves the generalized Singleton bound is called an *MDS convolutional code.*

A theoretical proof of Theorem 2.5 will be given in Chapter 6. It is non-constructive and it makes use of algebraic geometry. In the sections that follow we provide a constructive proof of the theorem 2.5.

Remark 2.7 It follows from Theorem 2.1 that MDS convolutional codes necessarily have the generic set of row degrees as in (2.2). It is worth mentioning that within the class of all rate k/n codes with fixed degree δ, the distribution (2.2) of the row degrees leads to the smallest possible memory.

The set of convolutional codes of rate k/n and degree δ is subdivided into codes whose encoding matrices $G(D)$ have a fixed set of row degrees $\nu_1, \ldots, \nu_k$ with $\delta = \sum_{i=1}^{k} \nu_i$. In Theorem 2.1 we gave an upper bound for the free distance for a code whose row degrees are not necessarily the generic set of indices.

We conclude the section with a simple theorem that tells how to obtain MDS-convolutional codes of rate k'/n from MDS-codes of rate k/n where $k' < k$.

Theorem 2.8 *Let $\mathcal{C}$ be a convolutional code of rate k/n generated by the minimal-basic encoding matrix $G(D) \in \mathbb{F}[D]^{k\times n}$ with row indices $\nu = \nu_1 = \ldots = \nu_\ell < \nu_{\ell+1} \leq \ldots \leq \nu_k$, where $\ell < k$. Let $\bar{G}(D) \in \mathbb{F}[D]^{(k-1)\times n}$ be the matrix obtained from $G(D)$ by omitting any of the last $k-\ell$ last rows of $G(D)$. If the free distance of $\mathcal{C}$ achieves the upper bound (2.1), then the same is true for the code $\bar{\mathcal{C}}$ generated by the encoder $\bar{G}$. In particular, if $\mathcal{C}$ is an MDS-code, then so is $\bar{\mathcal{C}}$.*

Proof: First note that non-catastrophicity as well as the full rank conditions carry over to the matrix $\bar{G}$. Moreover, the codes $\mathcal{C}$ and $\bar{\mathcal{C}}$ both have the same minimal row degree ν and the same number ℓ of rows having this degree ν. Therefore, the upper bound (2.1) has the same value for both codes and the theorem follows from the inclusion $\bar{\mathcal{C}} \subseteq \mathcal{C}$. □

2.2 A Few Examples of MDS Codes

We give here a few examples of MDS convolutional codes.

Example 2.9 Let

$$G(D) = \begin{bmatrix} 1 & 1 & 1 \\ D-1 & D-2 & 2D-3 \end{bmatrix}$$

be an encoder for a rate $2/3$ convolutional code $\mathcal{C}$ of degree $\delta = 1$, over $\mathbb{F}_5$. The encoder is non-catastrophic and the bound (2.3) is $d_{free} \leq 3$. We claim that the code $\mathcal{C}$ defined by $G(D)$ has free distance equal to 3, i.e. $G(D)$ represents an MDS code. Indeed, writing

$$G = G_0 + G_1 D,\ G_0 = \begin{bmatrix} 1 & 1 & 1 \\ -1 & -2 & -3 \end{bmatrix}, G_1 = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 2 \end{bmatrix},$$

a codeword $v(D)$ can be written as:

$$\begin{aligned} v(D) &= v_0 + v_1 D + \ldots + v_{\gamma+1} D^{\gamma+1} = \\ &= ((i_0, j_0) + (i_1, j_1)D + \ldots + (i_\gamma, j_\gamma)D^\gamma)\,(G_0 + G_1 D), i_t, j_t \in \mathbb{F}_5. \end{aligned}$$

Equating coefficients we obtain:

$$v_0 = (i_0, j_0)G_0 \quad \text{and} \quad v_1 = (i_1, j_1)G_0 + (i_0, j_0)G_1 = \begin{bmatrix} i_1 & j_1 & j_0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ -1 & -2 & -3 \\ 1 & 1 & 2 \end{bmatrix}. \tag{2.4}$$

Without loss of generality we may assume that $v_0 \neq 0$. If $i_0 = 0$ then $j_0 \neq 0$ and the weight wt $(v_0) = 3$. On the other hand if $i_0 \neq 0$ then wt $(v_0) \geq 2$ since G_0 is a generator matrix for a $[3, 2]$ MDS block code and wt $(v_1) \geq 1$ since the 3×3 matrix appearing in (2.4) is invertible.

It follows that wt $(v(D)) \geq 3 \Rightarrow d_{free} = 3 \Rightarrow \mathcal{C}$ is MDS.

Example 2.10 Let

$$G(D) = \begin{bmatrix} D^2+1 & 3D^2+1 & 5D^2+1 \\ D-1 & D-2 & 2D-3 \end{bmatrix}$$

be defined over the field $\mathbb{F}_7$. Then $G(D)$ defines a non-catastrophic encoder of rate $2/3$ and degree $\delta = 3$. A similar argument to the one in the previous example shows that $d_{free} = 6$, i.e., $G(D)$ defines an MDS convolutional code.

2.3 A General Construction of MDS Convolutional Codes

In this section we construct an MDS convolutional code for each degree δ and each rate k/n. The underlying idea here follows the lines of [7, 16]. We start with a generator matrix of an $[N, K]$ Reed Solomon code, with $N = na, a > 1$ and regard it as an instance of the sliding generator matrix of a convolutional code (n, k, δ) as follows.

Let $g(D) = c_0 + c_1 D + \cdots + c_{N-K} D^{N-K}$ be the generator polynomial of the Reed Solomon code and $\mathcal{G}$ be the associated generator matrix:

$$\mathcal{G} = \begin{bmatrix} c_0 & c_1 & \dots & c_{N-K} & & & & \\ & c_0 & c_1 & \dots & c_{N-K} & & & \\ & & \ddots & \ddots & & \ddots & & \\ & & & c_0 & & c_1 & \dots & c_{N-K} \end{bmatrix}. \tag{2.5}$$

We cut out all rows $(jn+k+1)$th,$(jn+k+2)$th,... , $(jn+n)$th, $j \geq 0$, and group the first k rows of $\mathcal{G}$ in $k \times n$ matrices:

$$G_0 = \begin{bmatrix} c_0 & c_1 & \dots & & \dots & c_{n-1} \\ & c_0 & \dots & & \dots & c_{n-2} \\ & & \ddots & & & \\ & & & c_0 & \dots & c_{n-k} \end{bmatrix}, G_1 = \begin{bmatrix} c_n & c_{n+1} & \dots & c_{2n-1} \\ c_{n-1} & c_n & \dots & c_{2n-2} \\ & & \dots & \\ c_{n-k+1} & c_{n-k+2} & \dots & c_{2n-k} \end{bmatrix}, \dots. \tag{2.6}$$

We denote by G_ν the last matrix that is not all zeroes. We obtain:

$$\mathcal{G}' = \begin{bmatrix} G_0 & G_1 & \dots & G_\nu & & & \\ & G_0 & G_1 & \dots & G_\nu & & \\ & & \ddots & \ddots & & \ddots & \\ & & & G_0 & G_1 & \dots & G_\nu \end{bmatrix}.$$

This matrix may be regarded as an instance of the semi-infinite sliding generator matrix of the convolutional code generated by

$$G(D) = G_0 + G_1 D + \ldots + G_\nu D^\nu.$$

Writing the polynomial entries of $G(D)$ we obtain the following form for the polynomial encoder:

$$G(D) = \begin{bmatrix} g_0(D) & g_1(D) & \cdots & g_{n-1}(D) \\ Dg_{n-1}(D) & g_0(D) & \cdots & g_{n-2}(D) \\ \vdots & \vdots & \cdots & \vdots \\ Dg_{n-k+1}(D) & Dg_{n-k+2}(D) & \cdots & g_{n-k}(D) \end{bmatrix}. \tag{2.7}$$

The relationship between the polynomial entries $g_0(D), g_1(D), \ldots, g_{n-1}(D)$ and the coefficients $c_0, c_1, \ldots, c_{N-K}$ of the starting generator polynomial $g(D)$ is given by the following:

$$g(D) = c_0 + c_1 D + \cdots + c_{N-K} D^{N-K} = g_0(D^n) + g_1(D^n)D + \ldots + g_{n-1}(D^n)D^{n-1}, \tag{2.8}$$

As defined in [7, 16], a convolutional code described through a polynomial encoder of the form (2.7) and (2.8) is said to be *generated by the polynomial* $g(D)$. We work out the details of this constructions by starting with the polynomial encoder description. With this technique we obtain a convolutional code of rate k/n and memory ν starting from a Reed Solomon block code, i. e. an MDS block code. We need to check now that this new obtained convolutional code borrows the MDS property from the starting MDS block code. In certain conditions this is true.

In the following we present rigorously the construction. Let $\mathcal{C}$ be a convolutional code generated by a polynomial encoder $G(D)$ of the form (2.7). We have that $\operatorname{rank} G(0) = k$ if $g(0) = g_0(0) \neq 0$. We also have that the code

$$\mathcal{C} = \{(u_0(D), \ldots, u_{k-1}(D)) \cdot G(D) \mid (u_0(D), \ldots, u_{k-1}(D)) \in \mathbb{F}^k[D]\}$$

is isomorphic to

$$\left\{ \left(u_0(D^n) + u_1(D^n)D + \ldots + u_{k-1}(D^n)D^{k-1} \right) \cdot g(D) \right\}. \tag{2.9}$$

The isomorphism is simply given by the multiplexing process and therefore is weight-preserving.

The following theorem will lead us to our desired construction of MDS convolutional codes. Two elements a, $b \in \mathbb{F}$ are called n-equivalent if $a^n = b^n$. The following theorem gives us the missing link.

Theorem 2.11 [7, Theorem 3]. *Let p be a prime and $r \in \mathbb{N}$. Let $g(D) \in \mathbb{F}_{p^r}[D]$ generate a cyclic code over $\mathbb{F}_{p^r}$ of length N relatively prime to p and of distance d_g. Let n be any positive divisor of N and $k < n$. If $g(D)$ has at most $n - k$ roots in each n-equivalence class, then the generator matrix $G(D)$ defined in* (2.7) *is basic minimal and describes a k/n convolutional code of free distance $d_{\text{free}} \geq d_g$.*

The idea of how to construct MDS codes of any rate k/n and any degree δ becomes clear now. We will choose a polynomial $g(D) \in \mathbb{F}_{p^r}[D]$ of some degree $N-K$ that generates a rate $[N,K]$ Reed-Solomon block code, hence its minimum distance is equal to the Singleton bound $N-K+1$. The parameters N and K will be chosen such that $n \mid N$ and $d_g = N-K+1 = (n-k)\left(\lfloor \delta/k \rfloor + 1\right) + \delta + 1$. This is the Generalized Singleton Bound for the given parameters n, k, and δ (see Inequality (2.3)). If the polynomial $g(D)$ satisfies the conditions of Theorem 2.11, we obtain the desired MDS-convolutional code.

To accomplish this the following technical lemma will be needed.

Lemma 2.12 *Let p be a prime and k, n, δ fixed positive integers such that p and n are relatively prime and $k < n$. Then there exist positive integers r and a,*

$$a \geq \lfloor \delta/k \rfloor + 1 + \delta/(n-k), \tag{2.10}$$

solving the Diophantine equation

$$an = p^r - 1. \tag{2.11}$$

Proof: Consider the multiplicative group $(\mathbb{Z}/n\mathbb{Z})^*$ which has order $\phi(n)$. Since $(p,n) = 1$ we know that $p^{i\phi(n)} \equiv 1 \mod n$ for all $i \geq 1$. In particular $p^{i\cdot\phi(n)} - 1$ is divisible by n. Choose i such that (2.10) is satisfied for $a := \frac{p^{i\cdot\phi(n)}-1}{n}$. □

In the following assume that a, r is a solution of (2.11) satisfying the inequality (2.10). Let $N = an$ and let $K = N - (n-k)\left(\lfloor \delta/k \rfloor + 1\right) - \delta$. It is easily seen that $0 < K < N$. Let $\alpha \in \mathbb{F}_{p^r}$ be a primitive element of $\mathbb{F}_{p^r}$ and define

$$g(D) = (D-\alpha^0)(D-\alpha^1)\cdots(D-\alpha^{N-K-1}) \in \mathbb{F}_{p^r}[D]. \tag{2.12}$$

The polynomial $g(D)$ defines a rate $[N,K]$ Reed-Solomon block code with distance $d_g = N-K+1 = (n-k)\left(\lfloor \delta/k \rfloor + 1\right) + \delta + 1$ as desired.

The following theorem formulates the construction result obtained. This gives also a constructive proof of the Theorem 2.5 as we state in the remark that follows this theorem.

Theorem 2.13 *Let p, n, k and δ be integers with $k < n$ and n not divisible by p. Then there exists a field $\mathbb{F}_q$ of characteristic p and an MDS convolutional code of rate k/n and degree δ over $\mathbb{F}_{p^r}$. The generator matrix $G(D)$ in* (2.7) *induced by the polynomial $g(D)$ given in* (2.12) *defines an MDS convolutional code of rate k/n and degree δ over $\mathbb{F}_{p^r}$.*

Proof: First we show that the generator matrix $G(D)$ is of degree δ. In order to do so, we calculate the degrees of the polynomials $g_i(D)$ in the expansion (2.8) of $g(D)$. First note that $\deg g(D) = N - K = n\nu + n - \ell$, where $\nu = \lfloor \delta/k \rfloor$ and

$\ell = k\left(\lfloor \delta/k \rfloor + 1\right) - \delta > 0$. Since $g(D)$ defines a Reed Solomon block code it follows that all its coefficients are nonzero and one obtains

$$\begin{aligned} \deg g_i(D) &= \nu \qquad \text{for } i = 0, \ldots, n-\ell, \\ \deg g_i(D) &= \nu - 1 \quad \text{for } i = n-\ell+1, \ldots, n-1. \end{aligned}$$

This implies that the row degrees of $G(D)$ are indeed as in (2.2) and that $G(D)$ is minimal. Thus, the degree of the code generated by $G(D)$ is simply given by the sum of the row degrees, which is in fact

$$\ell\nu + (k-\ell)(\nu+1) = k\left(\lfloor \delta/k \rfloor + 1\right) - \ell = \delta.$$

Observe also that $\operatorname{rank} G(0) = k$.

Next we prove that g satisfies the root condition given in Theorem 2.11. To do so, observe that the n-equivalence class of α^s, where $0 \leq s \leq a-1$, consists of

$$\alpha^s, \alpha^{s+a}, \alpha^{s+2a}, \ldots, \alpha^{s+a(n-k-1)}, \alpha^{s+a(n-k)}, \ldots \ .$$

The form of $g(D)$ in (2.12) shows that each such n-equivalence class contains at most $n-k$ roots of $g(D)$ if $N-K \leq (n-k)a$. This is indeed guaranteed by construction of a in (2.10):

$$a \geq \lfloor \delta/k \rfloor + 1 + \frac{\delta}{n-k} = \frac{N-K}{n-k}. \tag{2.13}$$

Now Theorem 2.11 implies that the encoder $G(D)$ given in (2.7) is minimal-basic and generates an MDS code with the given parameters n, k, and δ. $\square$

Remark 2.14 We formulated Thm. 2.13 with a prescribed characteristic p of the field over which we construct the MDS convolutional code. If one is interested in the smallest possible field where this construction works, regardless of characteristic, one should of course choose a to be the smallest integer such that $a \geq \lfloor \delta/k \rfloor + 1 + \delta/(n-k)$ and $an+1$ is a prime power. In any case it follows immediately from (2.10) and (2.11) that the field size is the smallest possible prime power q for which

$$n \,|\, (q-1) \text{ and } q \geq \delta \frac{n^2}{k(n-k)} + 2. \tag{2.14}$$

Remark 2.15 In conclusion, if taking a subcode of a Reed Solomon block code formed by cutting the rows $(jn+k+1)$th,$(jn+k+2)$th,... ,$(jn+n)$th, $j \geq 0$, of its generator matrix, and sliding it to infinity, we obtain a convolutional code. This code is MDS if k is small enough, more exactly if $N-K \leq (n-k)a$. The proof that the weight of a long message, of degree greater than N, is still larger than $N-K+1$ uses Theorem 2.11 of Justesen and in particular the 'weight retaining property' as studied by Massey, Costello and Justesen [16].

We close this section with two constructions to exemplify the technique.

Example 2.16 Suppose we want to construct a $(3,2,5)$ MDS convolutional code. The MDS-bound is in this case 9 and from (2.14) we need the smallest prime power p^r bigger than 24, such that $p^r - 1$ is divisible by 3. The smallest possible field is $\mathbb{F}_{5^2}$ and we will need a rate $[24,16]$ Reed-Solomon code for the construction.

If we want however an MDS-code in characteristic 2, the smallest field is $\mathbb{F}_{2^6}$, and we need a rate $[63,55]$ Reed-Solomon code. Using, e. g., MAPLE, one calculates

$$\begin{aligned} g(D) &= \Pi_{i=0}^{7}(D-\alpha^i) \\ &= D^8 + \alpha^{42}D^7 + \alpha^{57}D^6 + \alpha^{26}D^5 + \alpha^6 D^4 + \alpha^{35}D^3 + \alpha^8 D^2 + D + \alpha^{28} \\ &= (\alpha^{28} + \alpha^{35}D^3 + \alpha^{57}D^6) + D(1 + \alpha^6 D^3 + \alpha^{42}D^6) + D^2(\alpha^8 + \alpha^{26}D^3 + D^6), \end{aligned}$$

where α is a primitive of $\mathbb{F}_{2^6}$. Hence an encoder for a $(3,2,5)$ MDS convolutional code is given by

$$G(D) = \begin{bmatrix} \alpha^{28} + \alpha^{35}D + \alpha^{57}D^2 & 1 + \alpha^6 D + \alpha^{42}D^2 & \alpha^8 + \alpha^{26}D + D^2 \\ \alpha^8 D + \alpha^{26}D^2 + D^3 & \alpha^{28} + \alpha^{35}D + \alpha^{57}D^2 & 1 + \alpha^6 D + \alpha^{42}D^2 \end{bmatrix}.$$

Example 2.17 Another example that we give is a $(5,2,12)$ MDS convolutional code. The MDS bound is $5(6+1) - 2 + 1 = 34$ and as before we will need the smallest prime power p^r greater than 55 such that $p^r - 1$ is divisible by 5. The smallest possible field is $\mathbb{F}_{61}$ and we need a $[60,27]$ Reed-Solomon code for the construction.

If we wish to have the construction over a field of characteristic 2 we must take $a = 51$ in Equation (2.11) which yields $N = q-1 = 2^8 - 1 = 255$. The Reed-Solomon code that we use has parameters $N = 255$ and $K = 222$.

2.4 The Dual of MDS Convolutional Codes

After generalizing the notion of MDS block code the question on the nature of the dual code naturally arises. We know that a dual of an MDS-block code is MDS, so we wish to see if this generalizes to the case of convolutional codes with degree $\delta > 0$. In this section we cover two special cases where this does hold and then we give two examples of MDS-convolutional codes whose dual is not MDS. In the end we study the dual of the MDS convolutional code constructed in 2.3. There the dual has a similar description as the constructed code and it proves to be MDS as well in certain situations.

In order to introduce the notion of a dual convolutional code in our module theoretical setting, consider the following bilinear form:

$$\begin{array}{rrcl} <,>\colon & \mathbb{F}^n[D] \times \mathbb{F}^n[D] & \longrightarrow & \mathbb{F}[D] \\ & (v(D), w(D)) & \longmapsto & v(D)w(D)^t. \end{array} \tag{2.15}$$

Using this bilinear form we define the dual of a code $\mathcal{C}$ as

$$\mathcal{C}^\perp := \{w(D) \mid < v(D), w(D) >= 0, \forall v(D) \in \mathcal{C}\}.$$

One always has that

$$\mathcal{C}^{\perp\perp} \supseteq \mathcal{C}.$$

If the code $\mathcal{C}$ has a minimal basis encoder (i.e. it is non-catastrophic) then $\mathcal{C}^{\perp\perp} = \mathcal{C}$.

The following two lemmas cover some cases where the dual of an MDS convolutional code is MDS.

Lemma 2.18 *If $\mathcal{C}$ is a convolutional code of degree $\delta = 0$, i.e. a block code, then $\mathcal{C}$ is MDS if and only if $\mathcal{C}^{\perp}$ is MDS.*

Lemma 2.19 *Assume $k = 1, n = 2$. A non-catastrophic code $\mathcal{C}$ of rate $1/2$ is MDS if and only if $\mathcal{C}^{\perp}$ is MDS.*

We will present now a very simple example of a rate $1/3$ MDS convolutional code which has a non-MDS convolutional code of rate $2/3$ as its dual. In this example the degree $\delta = 1$ and the finite field is $\mathbb{F}_3$:

Example 2.20 Let $k = 1$, $n = 3$, $\delta = 1$ and consider the generator matrix

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

Then the code generated by $G(D)$ is non-catastrophic and MDS but the dual code is not an MDS convolutional code.

Indeed it is easy to see that any codeword $v(D) = i(D)G(D), i(D) \in \mathbb{F}^k[D]$ has weight at least 6, so the code generated by $G(D)$ is MDS. The dual code has a generator matrix given by:

$$G^{\perp} = \begin{pmatrix} D+1 & 0 & 2D+1 \\ 0 & 1 & 2 \end{pmatrix},$$

which is not MDS.

The above example shows that in general the dual code of an MDS convolutional code is not an MDS convolutional code anymore in contrast to the situation of block codes.

We will give here another example of a 1/3 MDS-convolutional code obtained from Theorem 4.1. The dual code is again non-MDS.

Example 2.21 Let α be a primitive element in the field $\mathbb{F}_8$, satisfying $\alpha^3+\alpha+1 = 0$. Then the code of degree 2 given by the generator matrix

$$G = \begin{pmatrix} g_1 & g_2 & g_3 \end{pmatrix} = \begin{pmatrix} (D-1)(D-\alpha) & (D-\alpha)(D-\alpha^2) & (D-\alpha^2)(D-\alpha^3) \end{pmatrix}$$

is an MDS-convolutional code. The dual of this code in not MDS.

Proof: The code generated by G is an $1/3$-MDS convolutional code accordingly with Theorem 4.1.

Looking now to find a generator matrix for the dual code, we need to find a 2×3 polynomial matrix having the full size minors equal to the full size minors of the matrix G. By inspection we obtain that

$$G' = \begin{pmatrix} (D-\alpha^2) & (D-1) & 0 \\ 0 & (D-\alpha^3) & (D-\alpha) \end{pmatrix}.$$

is a generator matrix for the dual code. The code obtained in this way has a codeword of weight 4 strictly less than the upper bound $3(1+1)-1=5$ of (2.3). Therefore the dual code is not MDS. □

The above two example show that in general the dual code of a MDS convolutional code is not an MDS convolutional code anymore in contrast to the situation of block codes.

2.5 The Dual of the General Construction 2.3

In this section we discuss the situations in which the dual of the construction presented in 2.3 may or may not be MDS. We give a minimal generator matrix of the dual of this code and describe the connection with the parity check polynomial of the cyclic code generated by the generator polynomial $g(D)$.

Let $\mathcal{C}$ be generated by $g(D)$ through (2.7), (2.8) where $g(X)$ is a generator polynomial of an MDS-cyclic block code (N, K). Let $h(X)$ be the parity check polynomial associated to $g(X)$, i.e. $g(X)h(X) = X^N - 1$.

A generator matrix for the dual convolutional code $\mathcal{C}^\perp$ is then given by the matrix:

$$H(D) = \begin{bmatrix} \bar{h}_{n-1}(D) & \bar{h}_{n-2}(D) & \dots & \bar{h}_1(D) & \bar{h}_0(D) \\ \bar{h}_{n-2}(D) & \bar{h}_{n-3}(D) & \dots & \bar{h}_0(D) & D\bar{h}_{n-1}(D) \\ \vdots & \vdots & \dots & \vdots & \vdots \\ \bar{h}_k(D) & \bar{h}_{k-1}(D) & \dots & D\bar{h}_{k+2}(D) & D\bar{h}_{k+1}(D) \end{bmatrix} \tag{2.16}$$

where $h(D) = \bar{h}_0(D^n) + \bar{h}_1(D^n)D + \ldots + \bar{h}_{n-1}(D^n)D^{n-1}$.

Therefore if we have $X^N - 1 = (X-\alpha)(X-\alpha^2)\ldots(X-\alpha^N)$ where α is a N-th root of unity in the field $\mathbb{F}$ and we take $g(X) = (X-\alpha)(X-\alpha^2)\ldots(X-\alpha^{N-K})$ and $h(X) = (X-\alpha^{N-K+1})(X-\alpha^{N-K+2})\ldots(X-\alpha^N)$ then the convolutional codes defined by $g(D)$ through (2.7), (2.8), respectively $h(D)$ through (2.16) are dual.

The matrix $H(D)$ in (2.16) gives us also a parity check matrix for the convolutional code constructed in Section 2.3. In their papers, Tanner [30], and Levy and Costello [11] choose to work with this parity check matrix instead of the generator matrix, avoiding in this way the case of catastrophic encoder.

Remark 2.22 Let $\mathcal{C}^\perp$ be the dual of an MDS convolutional code $\mathcal{C}$ generated by a polynomial $g(D)$ through (2.7) and (2.8). Then $\mathcal{C}^\perp$ is MDS iff $k \mid \delta$ and $(n-k) \mid \delta$. Indeed we need to have

$$N - K + 1 = (n-k)((\lfloor \delta/k \rfloor + 1) + \delta + 1$$

and

$$K + 1 = k((\lfloor \delta/(n-k) \rfloor + 1) + \delta + 1$$

which is satisfied only if both k and $n-k$ divide δ.

CHAPTER 3

STRONGLY MDS CONVOLUTIONAL CODES

3.1 Introduction of Strongly MDS Codes

In this chapter we study in more detail MDS convolutional code of rate $1/2$ and general degree δ. In this case there exists a construction different from the one presented in section 2.3, that proves to be better in many regards. We call the new constructed codes strongly MDS codes, because there are MDS codes having the column distances optimal. In the following we study these codes.

Let $\mathcal{C}$ be a $1/2$ rate convolutional code over a field $\mathbb{F}$, generated by $G(D) = \begin{bmatrix} a(D) & b(D) \end{bmatrix}$, where

$$a(D) = a_0 + a_1 D + \ldots + a_\delta D^\delta, b(D) = b_0 + b_1 D + \ldots + b_\delta D^\delta$$

are degree δ polynomials over $\mathbb{F}$. We suppose $G(D)$ is basic minimal, i.e. $a_0 \neq 0$ or $b_0 \neq 0$, and $a(D), b(D)$ are coprime.

A parity check matrix for $\mathcal{C}$ is given by $H(D) = \begin{bmatrix} -b(D) & a(D) \end{bmatrix}$.

We expand the matrices $G(D)$ and $H(D)$ into

$$G(D) = G_0 + G_1 D + \ldots G_\delta D^\delta, G_j \in \mathbb{F}^{1\times 2}, j = 0, \ldots, \delta$$

and

$$H(D) = H_0 + H_1 D + \ldots + H_\delta D^\delta, H_j \in \mathbb{F}^{1\times 2}, j = 0, \ldots, \delta.$$

Let:

$$G_j^c = \begin{bmatrix} G_0 & G_1 & \ldots & G_j \\ & G_0 & \ldots & G_{j-1} \\ & & \ddots & \vdots \\ & & & G_0 \end{bmatrix}, H_j^c = \begin{bmatrix} H_0 & & & \\ H_1 & H_0 & & \\ \vdots & \vdots & \ddots & \\ H_j & H_{j-1} & \ldots & H_0 \end{bmatrix}, \qquad (3.1)$$

both $(j+1) \times 2(j+1)$ matrices. Since

$$G(D)H^T(D) = 0 \Rightarrow (G_0 + G_1 D + \ldots G_\delta D^\delta)(H_0^T + H_1^T D + \ldots + H_\delta^T D^\delta) = 0,$$

we have

$$G_j^c \cdot (H_j^c)^T = 0.$$

Let $\mathcal{C}_j$ be the set given by:

$$\mathcal{C}_j := \{(u_0, \ldots, u_j)G_j^c, \mid u_0 \neq 0\} = \{(v_0, \ldots, v_j) \mid (v_0, \ldots, v_j)(H_j^c)^T = 0,\ v_0 \neq 0\}. \tag{3.2}$$

This is the subset of the code generated by G_j^c given by the difference of two subspaces:

$$\mathcal{C}_j = \{(u_0, \ldots, u_j)G_j^c\} - \{(u_0, \ldots, u_j)G_j^c, \mid u_0 = 0\}.$$

$\mathcal{C}_j$ is not a linear block code since it does not include the 0 vector. However, if we define the minimum distance as

$$d(\mathcal{C}_j) = \min_{u_0 \neq 0} \text{wt } (u_0, \ldots, u_j) \cdot G_j^c,$$

we obtain the definition of d_j^c, the jth column distance of the convolutional code $\mathcal{C}$, as was given in Chapter 1. This is an important notion for the convolutional code, therefore the set $\mathcal{C}_j$ is valuable, although it is not a linear block code.

We will see that this set, endowed with some properties, becomes very important in the construction of this section.

Based on 3.2, we say that G_j^c is a generator matrix, and H_j^c a parity check matrix. We write explicitly the matrix G_j^c for future reference:

$$G_j^c = \begin{bmatrix} a_0 & b_0 & a_1 & b_1 & \ldots & a_j & b_j \\ & & a_0 & b_0 & \ldots & a_{j-1} & b_{j-1} \\ & & & \ddots & & \vdots & \vdots \\ & & & & & a_0 & b_0 \end{bmatrix}.$$

We have the following natural bound on the d_j^c.

Theorem 3.1 *A convolutional code of rate 1/2 has the jth column distance bounded above by:*

$$d_j^c \leq j + 2.$$

Remark 3.2 This theorem gives an upper bound on the minimum distance of $\mathcal{C}_j$. The upper bound $j+2$ is exactly the Singleton bound for a block code of parameters $[2(j+1), (j+1)]$, the parameters of the matrix G_j^c. Therefore the set $\mathcal{C}_j$ behaves like a block code with respect to the Singleton bound.

Proof: We need to show that there exists a message $(u_0, u_1, \ldots, u_j)$, with $u_0 \neq 0$, that is encoded into a codeword $(u_0, u_1, \ldots, u_j)\ G_j^c$ of weight less or equal than $j+2$.

We can suppose $a_0 \neq 0$. If $a_0 = 0, b_0 \neq 0$ the proof follows the same pattern. Taking $u_0 = 1$, there exist $u_1, u_2, \ldots, u_j$ such that

$$(1, u_1, \ldots, u_j)G_j^c = (a_0, b_0, 0, \star, 0, \star, \ldots, 0, \star),$$

a message having at least j zeros. Hence

$$\text{wt } \left\{(1, u_1, \ldots, u_j) \cdot G_j^c\right\} \leq 2(j+1) - j = j + 2.$$

□

The following corollaries will bring us closer to our MDS construction.

Corollary 3.3 *A rate* $1/2$ *convolutional code has:*

1. $d_{2\delta-1}^c \leq 2\delta + 1$
2. $d_{2\delta}^c \leq 2\delta + 2.$

Corollary 3.4 *The index* $j = 2\delta$ *is the earliest step at which a rate* $1/2$ *MDS convolutional code (*$d_{free} = 2\delta + 2$*) can attain equality* $d_j^c = d_{free}$ *in the distance inequality (1.6):*

$$d_0^c \leq d_1^c \leq \ldots \leq d_\infty^c = d_{\text{free}} = 2\delta + 2.$$

Proof: Indeed

$$d_0^c \leq d_1^c \leq \ldots \leq d_{2\delta-1}^c \leq 2\delta + 1 < d_{\text{free}} = 2\delta + 2.$$

Hence the first index j at which $d_j^c = d_{\text{free}}$ will be after the 2δ index. □

We conclude:

Theorem 3.5 *A convolutional code of rate* $1/2$ *having*

$$d_{2\delta}^c = 2\delta + 2$$

is MDS. Such a code is also optimal in the sense of the previous corollary.

Definition 3.6 A rate $1/2$, degree δ, convolutional code is called *strongly MDS* if $d_{2\delta}^c = 2\delta + 2 = d_{free}$.

In the following we study these codes in detail.

3.2 Equivalent Definitions of Strongly MDS Codes

The following theorem gives a characterization of the strongly MDS codes in terms of the parity check matrix of $\mathcal{C}_{2\delta}$.

Theorem 3.7 *Let* $\mathcal{C}$ *be a* $1/2$ *rate convolutional code of degree* δ*. The following statements are equivalent:*

1. *The code* $\mathcal{C}$ *is strongly MDS;*

2. $d^c_{2\delta} = 2\delta + 2 = d_{\text{free}}$;

3. *The first column* $[a_0, a_1, \ldots, a_\delta, 0, \ldots, 0]^T$ *of the column parity check matrix*

$$H^c_{2\delta} = \begin{bmatrix} a_0 & & & & & & b_0 & & & & & \\ a_1 & a_0 & & & & & b_1 & b_0 & & & & \\ \vdots & \vdots & \ddots & & & & \vdots & \vdots & \ddots & & & \\ a_\delta & a_{\delta-1} & \ldots & a_0 & & & b_\delta & b_{\delta-1} & \ldots & b_0 & & \\ & a_\delta & \ldots & a_1 & a_0 & & & b_\delta & \ldots & b_1 & b_0 & \\ & & \ddots & & \ddots & \ddots & & & \ddots & & \ddots & \ddots \\ & & & a_\delta & \ldots & a_1 & a_0 & & & b_\delta & \ldots & b_1 & b_0 \end{bmatrix} \tag{3.3}$$

can not be written as a linear combination of any other 2δ *columns.*

Proof: The equality $d^c_{2\delta} = 2\delta + 2$ implies that the minimum distance of the set $\mathcal{C}_{2\delta}$ is $2\delta + 2$ which is equivalent to the third statement. □

This theorem is the version for the set $\mathcal{C}_{2\delta}$ of the well known equivalence theorem for block codes:

Theorem 3.8 *The following are equivalent:*

1. *An* $[n, k]$ *block code is MDS,*

2. *The parity check matrix has all full size minors invertible,*

3. *Any* $n - k$ *columns of the parity check matrix are linearly independent.*

This is another point where $\mathcal{C}_j$ behaves almost like a linear block code of same parameters. The following theorem will add more on this resemblance. It is a $\mathcal{C}_j$ version of [15, Theorem 8, ch.11]:

Theorem 3.9 *[15] An* $[n, k, d]$ *code* $\mathcal{C}$ *with generator matrix* $G = \begin{bmatrix} I & A \end{bmatrix}$, *where* A *is a* $k \times (n-k)$ *matrix, is MDS iff every square* $i \times i$ *submatrix of* A, *for any* $i = 1, 2, \ldots, \min\{k, n-k\}$, *is nonsingular.*

Our result is:

Theorem 3.10 *Let* $h_0, \ldots, h_n$ *nonzero in* $\mathbb{F}$. *Let* T *and* H *be the matrices*

$$T = \begin{bmatrix} h_0 & & & \\ h_1 & h_0 & & \\ \vdots & \vdots & \ddots & \\ h_n & h_{n-1} & \ldots & h_0 \end{bmatrix}, \tag{3.4}$$

$$H = \begin{bmatrix} T & I \end{bmatrix}, \tag{3.5}$$

where I *is the* $(n+1) \times (n+1)$ *identity matrix. Then the following are equivalent:*

1. *The column $(h_0, h_1, \ldots, h_n)^T$ is not a linear combination of n other columns of H.*

2. *The matrix T has the property that all its square submatrices having no zero rows or columns, are invertible.*

Proof: Assume that (1) does not hold. Then we may find an $(n+1) \times (n+1)$ submatrix B of H, with determinant 0, having $(h_0, h_1, \ldots, h_n)^T$ as first column. Since T is nonsingular, at least one column of I appears. Expanding the determinant along that column we find an $n \times n$ submatrix of B (given by erasing the last column and one of the rows) with determinant 0. Continuing along this way we obtain a square submatrix T' of T whose determinant vanishes. Since its first column is, by construction, a subcolumn of $(h_0, h_1, \ldots, h_n)^T$, each row of T' is nonzero. Since every zero entry of T lies strictly above the diagonal, the same holds for T', hence T' has no zero column. Conversely, suppose that (2) does not hold. For convenience, let us assume that (2) $\Rightarrow$ (1) holds in dimension n by induction (the case $n = 0$ is vacuous and $n = 1$ is clear).

Writing T as

$$T = \begin{bmatrix} h_0 & 0 \ \ldots 0 \\ h_1 & \\ \vdots & T_1 \\ h_n & \end{bmatrix}$$

it is easy to see from the induction process that if T_1 has the property (2) then T obtains property (1). So let us assume that every square submatrix of T_1 with nonzero rows and columns is nonsingular. Then, of course, the $k \times k$ submatrix of T given by the property (2) must involve the first column. Take the columns $\bar{c}_1, \ldots, \bar{c}_k$ of T corresponding to this submatrix (noting that these are now uniquely determined). Then we may complete the matrix $\begin{bmatrix} \bar{c}_1 & \bar{c}_2 & \ldots & \bar{c}_k \end{bmatrix}$ to an $(n+1) \times (n+1)$ matrix with vanishing determinant by inserting columns from I, namely, each column j for which row j is not "involved" in the $k \times k$ submatrix. Expanding as before, we obtain an $(n+1) \times (n+1)$ submatrix of H whose determinant is 0, and which by construction gives its first column as linear combination of the remaining columns. $\square$

We shall apply this theorem to the case $n = 2\delta$, where we shall refer to H as $\hat{H}^c{}_{2\delta}$.

Remark 3.11 We exemplify here the process used in the proof of the previous theorem: adding to a $k \times k$ square submatrix of the matrix (3.4) some columns of the matrix $I_{2\delta+1}$ to complete it to a square matrix of size $2\delta + 1$. We use the following $\delta+1$ square submatrix that is needed for future reference:

$$M := \begin{bmatrix} h_\delta & h_{\delta-1} & \dots & h_1 & h_0 \\ h_{\delta+1} & h_\delta & \dots & h_2 & h_1 \\ \vdots & \vdots & & \vdots & \vdots \\ h_{2\delta-1} & h_{2\delta-2} & \dots & h_\delta & h_{\delta-1} \\ h_{2\delta} & h_{2\delta-1} & \dots & h_{\delta+1} & h_\delta \end{bmatrix}. \tag{3.6}$$

We complete it with columns from the identity matrix:

$$\begin{bmatrix} h_0 & & & & & 1 & & & \\ h_1 & h_0 & & & & 0 & 1 & & \\ \vdots & \vdots & \ddots & & & \vdots & \vdots & \ddots & \\ h_{\delta-1} & h_{\delta-2} & \dots & h_0 & & 0 & 0 & \dots & 1 \\ h_\delta & h_{\delta-1} & \dots & h_1 & h_0 & 0 & 0 & \dots & 0 \\ h_{\delta+1} & h_\delta & \dots & h_2 & h_1 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots \\ h_{2\delta-1} & h_{2\delta-2} & \dots & h_\delta & h_{\delta-1} & 0 & 0 & \dots & 0 \\ h_{2\delta} & h_{2\delta-1} & \dots & h_{\delta+1} & h_\delta & 0 & 0 & \dots & 0 \end{bmatrix}$$

and obtain a $2\delta+1$ square submatrix of (3.5). Its columns are linearly independent, therefore the starting matrix (3.6) is invertible.

We link now the two theorems. We show that the two representation theorems are equivalent. Starting with a matrix of the form (3.3) and with the property 3 of Theorem 3.7, we obtain a matrix of the form (3.5) and with the property 1 of Theorem 3.10. The other way around: suppose we have a matrix T as in (3.8) with the property 2 of Theorem 3.10. We want to find A, B as in (3.7), with the property of 3 of Theorem 3.7.

Theorem 3.12 *The statements of Theorem 3.7 and Theorem 3.10 (in the case $n = 2\delta$) are equivalent.*

Proof: Let

$$A = \begin{bmatrix} a_0 & & & & & & \\ a_1 & a_0 & & & & & \\ \vdots & \vdots & \ddots & & & & \\ a_\delta & a_{\delta-1} & \dots & a_0 & & & \\ & a_\delta & \dots & a_1 & a_0 & & \\ & & \ddots & & \ddots & \ddots & \\ & & & a_\delta & \dots & a_1 & a_0 \end{bmatrix}, B = \begin{bmatrix} b_0 & & & & & & \\ b_1 & b_0 & & & & & \\ \vdots & \vdots & \ddots & & & & \\ b_\delta & b_{\delta-1} & \dots & b_0 & & & \\ & b_\delta & \dots & b_1 & b_0 & & \\ & & \ddots & & \ddots & \ddots & \\ & & & b_\delta & \dots & b_1 & b_0 \end{bmatrix} \tag{3.7}$$

such that the matrix $\begin{bmatrix} A & B \end{bmatrix}$ has property 3 of Theorem 3.7. Hence $a_0 \neq 0, b_0 \neq 0$, which implies that the matrices A, B are both invertible.

Therefore multiplying the matrix $\begin{bmatrix} A & B \end{bmatrix}$ of (3.3) with B^{-1} we obtain a matrix $\begin{bmatrix} T & I_{2\delta+1} \end{bmatrix}$ that has the same property as the matrix $\begin{bmatrix} A & B \end{bmatrix}$; i. e. property 1 of Theorem 3.10. The fact that T has a Toeplitz form as in (3.4) is easy now: B^{-1} is also a Toeplitz matrix and thus $T = B^{-1}A$ is a product of two lower triangular Toeplitz matrices, it is also lower triangular Toeplitz:

$$T = \begin{bmatrix} h_0 & & & & & & \\ h_1 & h_0 & & & & & \\ \vdots & & \ddots & & & & \\ h_\delta & h_{\delta-1} & \dots & h_0 & & & \\ h_{\delta+1} & h_\delta & \dots & h_1 & h_0 & & \\ \vdots & \vdots & \ddots & \vdots & & \ddots & \\ h_{2\delta} & h_{2\delta-1} & & h_\delta & h_{\delta-1} & \dots & h_0 \end{bmatrix} \tag{3.8}$$

To find $h_0, h_1, \dots, h_{2\delta}$ we solve the system:

$$BT = A \Leftrightarrow (h_0, h_1, \dots, h_{2\delta})^T = B^{-1} \cdot (a_0, a_1, \dots, a_\delta, 0, \dots, 0)^T \Leftrightarrow \tag{3.9}$$

$$\Leftrightarrow h(D)b(D) = a(D) \mod D^{2\delta+1}, \tag{3.10}$$

where

$$h(D) = h_0 + h_1 D + \dots + h_{2\delta} D^{2\delta}. \tag{3.11}$$

Therefore $h_0, h_1, \dots, h_{2\delta}$ are uniquely determined by $a(D)$ and $b(D)$. They must each be nonzero, otherwise the first column $[a_0, a_1, \dots, a_\delta, 0, \dots, 0]^T$ of the matrix $H_{2\delta}^c$ could be written as a linear combination of 2δ other columns.

Conversely, suppose we have a matrix T as in (3.8) with property 2 of Theorem 3.10.

From the above development we see that the problem reduces to finding, if they exist, two polynomials $a(D), b(D)$, both of degree δ, with all coefficients nonzero, uniquely given by the equation (3.10) $h(D)b(D) = a(D) \mod D^{2\delta+1}$.

Equating the coefficients of the powers of D larger than δ we obtain that $b_0, b_1, \dots, b_\delta$ must satisfy:

$$\begin{bmatrix} h_\delta & h_{\delta-1} & \dots & h_1 & h_0 \\ h_{\delta+1} & h_\delta & \dots & h_2 & h_1 \\ \vdots & \vdots & & \vdots & \vdots \\ h_{2\delta-1} & h_{2\delta-2} & \dots & h_\delta & h_{\delta-1} \\ h_{2\delta} & h_{2\delta-1} & \dots & h_{\delta+1} & h_\delta \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{\delta-1} \\ b_\delta \end{bmatrix} = \begin{bmatrix} a_\delta \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}. \tag{3.12}$$

The matrix of this system is the matrix M of (3.6), hence invertible by property 2 of Theorem 3.10. Therefore the system has a unique solution modulo the factor a_δ which may be chosen arbitrarily.

If we denote by C_{1j} the $\delta \times \delta$ cofactor obtained by cutting out the first row and the jth column of the matrix M, we obtain the solution:

$$b_j = a_\delta \cdot \frac{C_{1j}}{\det(M)}, j = 0, \dots, \delta. \tag{3.13}$$

From the relations:

$$\begin{bmatrix} h_0 & & & \\ h_1 & h_0 & & \\ \vdots & & \ddots & \\ h_{\delta-1} & h_{\delta-2} & \dots & h_0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_\delta \end{bmatrix} = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{\delta-1} \end{bmatrix}$$

we obtain:

$$a_j = a_\delta \cdot \frac{\begin{vmatrix} h_j & h_{j-1} & \dots & h_0 & 0 & \dots & 0 \\ h_{\delta+1} & h_\delta & \dots & h_{\delta-j+1} & h_{\delta-j} & \dots & h_1 \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ h_{2\delta-1} & h_{2\delta-2} & \dots & h_{2\delta-j-1} & h_{2\delta-j-2} & \dots & h_{\delta-1} \\ h_{2\delta} & h_{2\delta-1} & \dots & h_{2\delta-j} & h_{2\delta-j-1} & \dots & h_\delta \end{vmatrix}}{\det(M)}, j = 0, \dots, \delta-1. \tag{3.14}$$

The matrix that appears in the formula for a_j is a $(\delta+1)\times(\delta+1)$ submatrix of T, hence invertible.

Therefore for each nonzero value of a_δ chosen arbitrarily, we obtain two polynomials $a(D), b(D)$ uniquely determined by $h_0, h_1, \dots, h_{2\delta}$ through (3.13), (3.14). These equations give that the coefficients b_j, a_j are equal to the cofactors of the matrix M defined by (3.6), and respectively, to the determinant of the $(\delta+1)\times(\delta+1)$ formed by the last δ rows of the matrix M and one of the other first δ rows of the matrix T. Since all the matrices are invertible from property 2 of Theorem 3.10, we obtain that all a_j, b_j obtained from T are nonzero and also that $A = BT$, hence the matrix $\begin{bmatrix} A & B \end{bmatrix}$ has the property 3 of Theorem 3.7. $\square$

Remark 3.13 As parity check matrices for a block codes, the matrices $\begin{bmatrix} A & B \end{bmatrix}$, $\begin{bmatrix} T & I_{2\delta+1} \end{bmatrix}$, give a block code very far from being MDS. In fact the minimum distance of this code is 2. However, as part of the sliding parity check matrix of a rate 1/2 convolutional code, this matrix gives an excellent code!

Therefore this is one example where a convolutional code performs better than the block code that stays at the base of the convolution construction.

3.3 The Question of the Existence of Good Matrices

A question is whether there are Toeplitz matrices T,

$$T = \begin{bmatrix} h_0 & & & \\ h_1 & h_0 & & \\ \vdots & & \ddots & \\ h_n & h_{n-1} & \dots & h_0 \end{bmatrix} = \begin{bmatrix} T_0 & \vdots & T_1 & \vdots & \dots & \vdots & T_n \end{bmatrix}, \tag{3.15}$$

having the property that all square sub-matrices of T that do not have any row or column all zeros are invertible. We denote with T_i the column vectors of T.

Looking in the literature for matrices having this dependence property we came across the two examples presented in [15]:

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 1 & 3 & 4 & \\ 1 & 4 & & \end{bmatrix} /\mathbb{F}_5 \text{ and } \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 3 & 6 & 4 & 2 & 5 \\ 1 & 6 & 4 & 2 & 5 & \\ 1 & 4 & 2 & 5 & & \\ 1 & 2 & 5 & & & \\ 1 & 5 & & & & \end{bmatrix} /\mathbb{F}_7.$$

These matrices have the property that every rectangular submatrix with no zero entry, has all square submatrices invertible. The generalization for larger q is stated there as an open problem. A solution is given by R. Roth, [26], where such a matrix is formed over a field with at least $n+1$ elements. Unfortunately these matrices can have submatrices with nonzero rows and columns, but some of the entries zero, singular. For example the minor:

$$\begin{bmatrix} 3 & 6 & 4 & 2 \\ 6 & 4 & 2 & 5 \\ 4 & 2 & 5 & 0 \\ 2 & 5 & 0 & 0 \end{bmatrix}$$

of the second matrix, is singular over $\mathbb{F}_7$.

The second avenue to take in proving the existence of matrices T with the desired minor property is the one of*totally positive* matrices over the real numbers. These are square matrices over $\mathbb{R}$ with all minors positive real numbers. Lower triangular totally positive Toeplitz matrices are products $T' := \prod_1^n X_i$ of matrices X_i,

$$X_i = \begin{bmatrix} 1 & & & & & \\ x_i & 1 & & & & \\ & x_i & 1 & & & \\ & & \ddots & \ddots & & \\ & & & x_i & 1 & \\ & & & & x_i & 1 \end{bmatrix}, \tag{3.16}$$

with x_i positive real numbers ([14]). The first column of T' has the entry h'_i equal to the ith elementary function in $x_1, \ldots, x_n$, denoted e_i in the literature, $1 \le i \le n$.

For our purpose let $x_i, 1 \le i \le n$ be n positive integers. We compute all the minors of the matrix T'. These are positive integers. Let p be the smallest prime that does not divide any of the minors of T'. The matrix $T := T' \mod p$ has then the property that all its square sub-matrices that do not have any zero rows or columns are invertible over $\mathbb{F}_p$.

Example 3.14 Let the matrices $X_i, 1 \le i \le n$, be equal to the $(n+1) \times (n+1)$ matrix

$$X = \begin{bmatrix} 1 & & & & & \\ 1 & 1 & & & & \\ & 1 & 1 & & & \\ & & \ddots & \ddots & & \\ & & & 1 & 1 & \\ & & & & 1 & 1 \end{bmatrix}, \tag{3.17}$$

and let $T' = \prod_1^n X_i = X^n$. The matrix T' has the form:

$$T' = \begin{bmatrix} 1 & & & & & \\ n & 1 & & & & \\ \binom{n}{2} & n & 1 & & & \\ \vdots & \vdots & \ddots & \ddots & & \\ \binom{n}{n-1} & \binom{n}{n-2} & \ldots & n & 1 & \\ 1 & \binom{n}{n-1} & \ldots & & n & 1 \end{bmatrix}. \tag{3.18}$$

Note that the entries on first column of T' are the coefficients of the expanded polynomial $(X+1)^n$. The minors are all positive integers ([14]), and taking the smallest prime p that does not divide any of them we obtain the matrix T, $T := T'$ mod p, with the desired property.

Remark 3.15 The field over which we construct the matrix T may be large. An alternative that we currently consider is to take all the entries $x_i, 1 \le i \le n$ equal to consecutive powers of a primitive element of an arbitrary finite field $\mathbb{F}_q$, with at least $n+2$ elements. The minor in the lower part of the matrix T thus obtained, are the *skew-Schur-functions* in terms of $\{x_1, \ldots, x_n\}$, [29, page 344]. Imposing the nonzero conditions on this functions, we might be able to obtain estimates for the field size, and a matrix T with the desired property, over a suitable field. This method is subject of further research.

3.4 Construction of the Code

Having the matrix T given by the method above, how do we now obtain a rate 1/2 convolutional code?

Out of the matrix T obtained in the previous section, we have to obtain a generator matrix $G(D) = \begin{bmatrix} a(D) & b(D) \end{bmatrix}$ with $a(D), b(D)$ of degree δ. But this is exactly the process we used in the previous sections to prove the equivalence between an $[A\ B]$ representation and a $[T\ I]$ one. We take the coefficients a_i, b_j of $a(D)$, respectively $b(D)$, given by the expressions in (3.13), (3.14) and Theorem 3.12 gives us that the convolutional code generated by $G(D)$ is MDS.

3.5 Remarks on this Construction

We conclude by a few remarks on the importance of the construction:

Remark 3.16 This construction is based on a matrix of no immediate value in block coding. As a parity check matrix (or generator matrix) for a linear block code, this matrix gives a very poor $[4\delta+2, 2\delta+1, 2]$ block code.

However, as part of the sliding parity check matrix of a rate 1/2 convolutional code, this matrix gives an MDS code!

Remark 3.17 We obtain here a rate 1/2 MDS code of degree δ, capable to correct δ errors in any **sliding** window of length $4\delta+2$.

The best known MDS block code with parameters $[n, n/2], n = 4\delta+2$, can correct δ errors in any **slotted** window of length $4\delta+2$.

3.6 Decoding

Let $\mathcal{C}$ be a rate 1/2 MDS convolutional matrix generated by
$G(D) = \begin{bmatrix} a(D) & b(D) \end{bmatrix}$ with $a(D), b(D)$ of degree δ, satisfying the properties of Theorem 3.7.

Then the code $\mathcal{C}$ is theoretically capable of correcting δ errors in any sliding window of length $4\delta+2$.

Let $(y(D), z(D)) \in (\mathbb{F}[D])^2$ be a received message.

Then there exists a codeword $(v(D), w(D)) \in \mathcal{C}$, and an error vector
$(f(D), e(D)) \in (\mathbb{F}[D])^2$ such that $y(D) = v(D) + f(D), z(D) = w(D) + e(D)$.

Let $y_0, \ldots, y_{2\delta}, z_0, \ldots, z_{2\delta}$ be some $4\delta+2$ consective components of the received message $y(D), z(D)$. Multiplying the received message with the sliding parity check matrix of the code we obtain the syndrome equations.

We consider a window of $2\delta+1$ syndrome equations:

$$\begin{bmatrix} T & I \end{bmatrix} \begin{bmatrix} y_0 \\ \vdots \\ y_{2\delta} \\ z_0 \\ \vdots \\ z_{2\delta} \end{bmatrix} = \begin{bmatrix} T & I \end{bmatrix} \begin{bmatrix} f_0 \\ \vdots \\ f_{2\delta} \\ e_0 \\ \vdots \\ e_{2\delta} \end{bmatrix} = \begin{bmatrix} s_0 \\ s_1 \\ \vdots \\ s_{2\delta} \end{bmatrix}. \tag{3.19}$$

3.6.1 The Algorithm

Suppose we have corrected all the components received before y_0, z_0. Assuming that the weight of the error $\begin{bmatrix} f_0 & \dots & f_{2\delta} & e_0 & \dots & e_{2\delta} \end{bmatrix}^T$ in this $4\delta+2$ window is δ, we find an algorithm that computes f_0 and e_0. Knowing f_0 and e_0 we update our received message, and move one step further. We consider the next sliding window and the sequence $f_1, \dots, f_{2\delta+1}, e_1, \dots, e_{2\delta+1}$ and correct now f_1, e_1.

The following theorem tells that such an algorithm theoretically exists.

Theorem 3.18 *Let $f = (f_0, \dots, f_{2\delta})^T, e = (e_0, \dots, e_{2\delta})^T$ be two vectors in $\mathbb{F}^{2\delta+1}$ such that* $\mathrm{wt} \begin{bmatrix} f \\ e \end{bmatrix} \leq \delta$. *Let*

$$\begin{bmatrix} T & I \end{bmatrix} \begin{bmatrix} f_0 \\ \vdots \\ f_{2\delta} \\ e_0 \\ \vdots \\ e_{2\delta} \end{bmatrix} = \begin{bmatrix} s_0 \\ s_1 \\ \vdots \\ s_{2\delta} \end{bmatrix}. \tag{3.20}$$

If $\begin{bmatrix} \tilde{f} \\ \tilde{e} \end{bmatrix}$ is another solution of the equation (3.20) with $\mathrm{wt} \begin{bmatrix} \tilde{f} \\ \tilde{e} \end{bmatrix} \leq \delta$ *then* $f_0 = \tilde{f}_0, e_0 = \tilde{e}_0$.

Proof: The difference $\begin{bmatrix} f-\tilde{f} \\ e-\tilde{e} \end{bmatrix}$ is in the kernel of the matrix $\begin{bmatrix} T & I \end{bmatrix}$. The weight of $\mathrm{wt} \begin{bmatrix} f-\tilde{f} \\ e-\tilde{e} \end{bmatrix} \leq 2\delta$. It implies that $f_0 - \tilde{f}_0 = 0, e_0 - \tilde{e}_0 = 0$, otherwise strictly less than $2\delta+2$ columns including the first one would be linearly dependent. Therefore $f_0 = \tilde{f}_0, e_0 = \tilde{e}_0$. $\square$

We sketch here an algorithm for finding f_0 and e_0. It is a searching algorithm and it uses heavily the Gaussian elimination method for finding if a vector is in the column space of a certain matrix. For any s, $s = 1, 2, \ldots, (\delta - 1)$, form all the $(2\delta+1-s)\times(2\delta+1)$ submatrices of the matrix T, (column indices are consecutive):

$$T_{i_0,\ldots,i_{2\delta-s}} = \begin{bmatrix} h_{i_0} & h_{i_0-1} & \ldots & h_0 & & & \\ h_{i_1} & h_{i_1-1} & \ldots & \ldots & h_0 & & \\ \vdots & \vdots & & & & & \\ h_{i_{2\delta-s}} & h_{i_{2\delta-s}-1} & \ldots & \ldots & \ldots & h_0 \end{bmatrix}. \tag{3.21}$$

For any l, $l = 1, 2, \ldots, (\delta - s)$ check if

$$\begin{bmatrix} s_{i_0} \\ s_{i_1} \\ \vdots \\ s_{i_{2\delta-s}} \end{bmatrix}$$

can be written as a linear combination of l columns of the matrix $T_{i_0,\ldots,i_{2\delta-s}}$. We start with $s = 1$ and let $l = 1, l = 2, \ldots, l = \delta - 1$, then $s = 2$ and try all possible values for l.

After finding one such matrix, we check if

$$\begin{bmatrix} s_{i_0} \\ s_{i_1} \\ \vdots \\ s_{i_{2\delta}} \end{bmatrix}$$

is a linear combination of the corresponding l columns of the matrix T. If it is, then the coefficients will be the corresponding components of an error f. Store f_0, compute $e_0 = s_0 - h_0 f_0$, and move to the next window. If not, keep searching until a good matrix is found.

This algorithm becomes impractical for large δ and q. The following theorems will study certain situations where, at one step, we have some extra information about the error.

The next theorem shows that in certain situations we may determine f_0 and e_0 if the weight of the syndrome is small enough:

Theorem 3.19 *Let* $s = \begin{bmatrix} s_0 & s_1 & \ldots & s_{2\delta} \end{bmatrix}^T$ *be the* $2\delta+1$ *syndrome of the equation (3.19). If* $\mathrm{wt}\ s \leq \delta + 1$ *then* $f_0 = 0 \Rightarrow e_0 = s_0$

Proof: Suppose $f_0 \neq 0$. Since $\text{wt} \begin{bmatrix} f \\ e \end{bmatrix} \leq \delta$ then s is a linear combination of at most δ columns of $\begin{bmatrix} T & I \end{bmatrix}$, therefore, from Corollary 3.23, the weight of s is at least

$$2\delta + 1 - \delta + 1 = \delta + 2 > \delta + 1.$$

This is in contradiction with our supposition on the weight of s, therefore $f_0 = 0$. □

Because of the symmetry of the matrix $[A\ B]$ we have also the following:

Theorem 3.20 *If* wt $T^{-1}s \leq \delta + 1$ *then* $e_0 = 0 \Rightarrow f_0 = s_0/h_0$

Corollary 3.21 *If* wt $s \leq \delta + 1$ *and* $s_0 \neq 0$*, then* $f_0 = 0, f_1 = 0$.

Moreover if wt $s \leq \delta + 1$ *and* $s_0 \neq 0, s_1 \neq 0, \ldots, s_i \neq 0$*, then* $f_0 = f_1 = \ldots = f_{i+1} = 0$ *and* $e_0 = s_0, e_1 = s_1, \ldots, e_{i+1} = s_{i+1}$ *are uniquely determined.*

.

Proof: From Theorem 3.19 we have that $f_0 = 0$. Since $s_0 \neq 0$ it implies that wt $[s_1\ s_2\ \ldots\ s_{2\delta}]^T \leq \delta$. Applying again Theorem 3.19 we get that $f_1 = 0$. We use this argument $i + 1$ steps to obtain: $f_0 = f_1 = \ldots = f_{i+1} = 0$.

□

3.6.2 Theorems for Improving the Algorithm

We have the following lemma:

Lemma 3.22 *A column*

$$T_i = \begin{bmatrix} 0 & 0 & \ldots & 0 & h_0 & h_1 & \ldots & h_{2\delta-i} \end{bmatrix}^T, i = 0, \ldots, 2\delta,$$

of the matrix $\begin{bmatrix} T & I \end{bmatrix}$*, is in the span of at least* $2\delta - i + 1$ *other columns. To write* T_i *as a linear combination of a minimum number of columns of* $\begin{bmatrix} T & I \end{bmatrix}$*, it is necessary to choose columns among* T_j *and* J_k *with* $j < i, k < i$.

J_k *denotes the* k*th column of the identity matrix* $I_{2\delta+1}$*,* $k = 0, \ldots, 2\delta$.

Proof: It is obvious that a minimum representation of T_i as a linear combination of other columns will have only columns T_j, J_k with $j < i, k < i$. If there is such a representation with strictly less than $2\delta - i + 1$ such columns, then there would exist a singular submatrix of dimension less than $2\delta - i$ of the matrix T. But this is in contradiction with how we constructed the matrix T. □

Corollary 3.23 *For any $m \geq (2\delta+1)$, a linear combination of any columns*

$$T_{i_1}, \ldots, T_{i_k}, J_{j_1}, J_{j_2}, \ldots, J_{j_s}, J_{j_{s+1}}, \ldots, J_{j_l},$$

with $k+l=m, i_1 < \ldots < i_k, j_1 < \ldots < j_l, j_s < i_1 < j_{s+1}$, and the coefficient of T_{i_1} nonzero, has weight at least

$$2\delta + 2 - i_1 - m + s.$$

Proof: The columns $T_{i_1}, \ldots, T_{i_k}$ form a $(2\delta+1) \times k$ matrix with the first i_1 rows zero. Its $(2\delta - i_1 + 1) \times k$ submatrix formed by omitting the all zero rows has the property borrowed from T, that all square submatrices are invertible. The transpose of this matrix will then generate an MDS block code with minimum distance at least $(2\delta - i_1 + 1) - k + 1$. Then the weight of a linear combination of the columns $T_{i_1}, \ldots, T_{i_k}$ will be greater than $(2\delta - i_1 + 1) - k + 1$. Among the columns $J_{j_1}, J_{j_2}, \ldots, J_{j_s}, J_{j_{s+1}}, \ldots, J_{j_l}$ only $J_{j_{s+1}}, \ldots, J_{j_l}$ could decrease the weight of the linear combination of the columns $T_{i_1}, \ldots, T_{i_k}$ by at most the number of such columns, $l-s$. Therefore the weight of a linear combination of the columns $T_{i_1}, \ldots, T_{i_k}, J_{j_1}, J_{j_2}, \ldots, J_{j_s}, J_{j_{s+1}}, \ldots, J_{j_l}$, is at least $[(2\delta - i_1 + 1) - k + 1] - (l - s)$. $\square$

Theorem 3.24 *If $\begin{bmatrix} f \\ e \end{bmatrix}$ is a solution of the equation (3.20), and* wt $\begin{bmatrix} f \\ e \end{bmatrix} = i \leq \delta$, *then the first $\delta - i + 2$ components of f, respectively e, are uniquely determined.*

Proof: Let $\begin{bmatrix} \tilde{f} \\ \tilde{e} \end{bmatrix}$ be another solution of the equation (3.20), with weight less than δ. The difference $\begin{bmatrix} f - \tilde{f} \\ e - \tilde{e} \end{bmatrix}$ is in the kernel of the matrix $\begin{bmatrix} T & I \end{bmatrix}$. The weight of wt $\begin{bmatrix} f - \tilde{f} \\ e - \tilde{e} \end{bmatrix} \leq \delta + i$. With the same argument as the one in the proof of Theorem 3.18 it implies $f_0 = \tilde{f}_0, e_0 = \tilde{e}_0$.

Depending on whether f_0 or e_0 is nonzero or not, the weight

$$\text{wt} \begin{bmatrix} f_1 - \tilde{f}_1 \\ \vdots \\ f_{2\delta} - \tilde{f}_{2\delta} \\ e_1 - \tilde{e}_1 \\ \vdots \\ e_{2\delta} - \tilde{e}_{2\delta} \end{bmatrix} \leq \delta + i \text{ (or } i + \delta - 2).$$

If $i + \delta \leq 2\delta + 1$ we get, with the argument of Theorem 3.18 and Theorem 3.22 , that $f_1 - \tilde{f}_1 = 0, e_1 - \tilde{e}_1 = 0$. We can use the same argument $\delta + 2 - i$ steps. $\square$

The following are corollaries of this theorem:

Corollary 3.25 *If f_0 or e_0 are nonzero then f_1 and e_1 are uniquely determined. If at least one of them is nonzero then f_2 and e_2 are unique and so on.*

Corollary 3.26 *If $\begin{bmatrix} f \\ e \end{bmatrix}$ is a solution of the equation (3.20), with* $\mathrm{wt}\begin{bmatrix} f \\ e \end{bmatrix} \leq \delta$, *and $f_{\lfloor \delta/2 \rfloor+1} = f_{\lfloor \delta/2 \rfloor+2} = \ldots = f_{2\delta} = e_{\lfloor \delta/2 \rfloor+1} = e_{\lfloor \delta/2 \rfloor+2} = \ldots = e_{2\delta} = 0$, then $\begin{bmatrix} f \\ e \end{bmatrix}$ is uniquely determined.*

Remark 3.27 The previous results show that the code constructed has a very good error correcting capability on the burst error channels.

CHAPTER 4

ANOTHER CONSTRUCTION OF A RATE 1/n MDS CODE

In the case of rate $1/n$ there exists a different construction of an MDS convolutional code starting from MDS block codes. The field required has less elements than the general construction we presented in Chapter 2.3. This construction was provided by Justesen in [8]. A rate $1/n$, degree δ convolutional code is given there, having free distance equal to $n(\delta+1)$, the maximal possible distance of all codes with these parameters. Hence the code is maximal distance separable (MDS).

In this section we state the result of Justesen [8] and we describe the construction through the first order representations presented in 1.6. This rewriting permits a new proof that gives an iterative intuition to the polynomial proof of Justesen. Also, given their representation, the presented codes are very suitable for the decoding algorithm of Rosenthal [21].

The following result was obtained in [8] for rate 1/2:

Theorem 4.1 *Let $\mathbb{F}$ denote an arbitrary finite field with q elements and α a primitive element. Let*

$$g_1(D) = (D-\alpha)(D-\alpha^2)\dots(D-\alpha^\delta), \quad g_1(D) = (D-\alpha^{-1})(D-\alpha^{-2})\dots(D-\alpha^{-\delta}),$$

and $G(D) = \begin{bmatrix} g_1(D) & g_2(D) \end{bmatrix}$ be a polynomial encoder. If $q-1 \geq 3\delta$, then the $1/2$ rate convolutional code generated by $G(D)$ is a non-catastrophic MDS code.

In this construction each polynomial component $v_i(D)$ of an arbitrary codeword $(v_1(D), v_2(D))$ is a multiple of the generator polynomial $g_i(D)$. The $q-2$ degree polynomial $v_i(D) \mod (x^{q-1}-1)$ is then a codeword in the $[q-1, q-1-\delta]$ Reed Solomon code generated by $g_i(D)$. Therefore its weight is greater than $\delta+1$. One may consider whether the weight of $v_i(D)$ is also greater than $\delta+1$. The answer is negative. Its weight may be as small as 2. However, the above theorem states that even if this happens the sum of the weights of the components will still be greater then twice the minimum distances of the Reed Solomon codes: $2(\delta+1)$. The only condition we need to impose is $q-1 \geq 3\delta$.

We describe this construction in terms of the **ABCD** representation given in the introduction.

Let $\mathcal{C}$ be a convolutional code over $\mathbb{F}$ of rate $1/n$. As shown in [23, 32] we may represent $\mathcal{C}$ through the input/state/output description:

$$\begin{aligned} x_{t+1} &= \mathbf{A}x_t + \mathbf{B}u_t \\ y_t &= \mathbf{C}x_t + \mathbf{D}u_t, \quad x_0 = 0, \; x_{\gamma+1} = 0. \end{aligned} \tag{4.1}$$

Let $k = 1$ and

$$\mathbf{A} := \begin{bmatrix} \alpha & 0 & \cdots & 0 \\ 0 & \alpha^2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \alpha^\delta \end{bmatrix}, \quad \mathbf{B} := \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}, \tag{4.2}$$

where α is a primitive element of the field $\mathbb{F}$. By choosing $\mathbf{C}, \mathbf{D}$ such that the pair $(\mathbf{A}, \mathbf{C})$ is observable it was shown in [23] that the code obtained in this way has distance more than $(\delta + 1)$, provided that we allow large enough fields. In this section we shall explain how to choose matrices $\mathbf{C}, \mathbf{D}$ to get an MDS convolutional code, that is a code with distance $n(\delta + 1)$. Let $\mathbf{A}, \mathbf{B}$ be defined as above and let

$$\mathbf{D} := \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}.$$

For a better understanding of the construction we present the case $n = 2$ first. In the end we outline the cases $n > 2$. Let $\mathbb{F} = \mathbb{F}_q$ with $q - 1 \geq 3\delta$, $\mathbf{A}$ and $\mathbf{B}$ as above, $\mathbf{D} = (1)$, and let

$$\mathbf{A}' := \begin{bmatrix} \alpha^{\delta+1} & 0 & \cdots & 0 \\ 0 & \alpha^{\delta+2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \alpha^{2\delta} \end{bmatrix}.$$

The system (1.25) may be rewritten as

$$\begin{aligned} x_{t+1} &= (\mathbf{A} - \mathbf{B}\mathbf{C})x_t &+& \mathbf{B}y_t \\ u_t &= -\mathbf{C}x_t &+& \mathbf{D}y_t. \end{aligned} \tag{4.3}$$

In (4.3) the input u_t interchanged its place with the output y_t. We would like to transform (4.3) into an equivalent system, having the form $x_{t+1} = \mathbf{A}'x_t + \mathbf{B}'y_t$,

$u_t = \mathbf{C}'x_t + y_t$, with $\mathbf{A}'$ defined as above. This is possible if there exists an invertible matrix S such that

$$S(\mathbf{A} - \mathbf{BC})S^{-1} = \mathbf{A}'$$

or else if $\det(DI - (\mathbf{A} - \mathbf{BC})) = \det(DI - \mathbf{A}') = \prod_{k=1}^{\delta}(D - \alpha^{\delta+k})$. In order to achieve this we must solve a linear equation resulting in a matrix $\mathbf{C} := \begin{pmatrix} c_1 & c_2 & \dots & c_\delta & 0 \end{pmatrix}$ for which

$$\det(DI - (\mathbf{A} - \mathbf{BC})) = \det(DI - \mathbf{A}').$$

In particular there exists an invertible matrix S such that $S(\mathbf{A} - \mathbf{BC})S^{-1} = \mathbf{A}'$. Now (4.3) is equivalent to:

$$\begin{array}{rcrcl} x_{t+1} & = & \mathbf{A}'x_t & + & S\mathbf{B}y_t \\ u_t & = & -\mathbf{C}S^{-1}x_t & + & y_t. \end{array} \tag{4.4}$$

Let $\mathbf{B}' := S\mathbf{B}$ and $\mathbf{C}' := \mathbf{C}S^{-1}$. It can be proved that $(\mathbf{A}', \mathbf{B}')$ forms a controllable pair and that $(\mathbf{A}', \mathbf{C}')$ forms an observable pair. It remains to be shown that the obtained code has distance $2(\delta + 1)$. First we recall that if

$$\begin{array}{rcl} u(D) & = & u_0 D^\gamma + u_1 D^{\gamma-1} + \cdots + u_\gamma, \\ y(D) & = & y_0 D^\gamma + y_1 D^{\gamma-1} + \cdots + y_\gamma, \end{array}$$

where γ is the degree of v, the first equations of the systems (1.25) and (4.4) yield (see [23, 32]):

$$(u_\gamma, ..., u_0)^t \in \ker \left(\mathbf{B}\ \mathbf{AB} \dots\ \mathbf{A}^\gamma \mathbf{B} \right)$$

and

$$(y_\gamma, \dots, y_0)^t \in \ker \left(\mathbf{B}'\ \mathbf{A}'\mathbf{B}' \dots\ \mathbf{A}'^\gamma \mathbf{B}' \right).$$

We suppose $u_0 \neq 0$ hence $y_0 \neq 0$. We consider the degree γ of a codeword v. In case $\gamma < q - 1$ then

$$\left(\mathbf{B}\ \mathbf{AB} \dots\ \mathbf{A}^\gamma \mathbf{B} \right)$$

and

$$\left(\mathbf{B}'\ \mathbf{A}'\mathbf{B}' \dots\ \mathbf{A}'^\gamma \mathbf{B}' \right)$$

are full rank Vandermonde matrices (multiplied eventually by some nonsingular diagonal matrices), therefore $(u_\gamma, \dots, u_0)^t$ and $(y_\gamma, \dots, y_0)^t$ both have weight greater than $\delta + 1$. Hence $(v_\gamma, ..., v_0)^t$ has weight more than $2(\delta + 1)$. If $\gamma \geq q - 1$, $\mathbf{A}^{q-1} = I$ so

$$(u_\gamma, \dots, u_0)^t \in \ker \left(\mathbf{B}\ \mathbf{AB} \dots\ \mathbf{A}^\gamma \mathbf{B} \right)$$

implies that

$$u' := \begin{bmatrix} u_0 + u_{q-1} + \dots \\ u_1 + u_q + \dots \\ \vdots \\ u_{q-2} + u_{2q-3} + \dots \end{bmatrix} \in \ker \left(\mathbf{A}^{q-2}\mathbf{B} \dots \ \mathbf{AB} \ \mathbf{B} \right)$$

that has rank δ. The case $u' \neq 0$ gives that the weight of $u' \geq \delta + 1$, hence the weight of u will be $\geq \delta + 1$ as well. Also defining y' in the same way, we have that the weight of $y' \geq \delta + 1$ unless $y' = 0$, therefore again wt $(v) \geq 2(\delta + 1)$. If $u' = 0$ and $y' \neq 0$, from the first equations of the systems (1.25) and (4.4) we have

$$\begin{aligned} x_1 + x_q + \dots &= \mathbf{A}(x_0 + x_{q-1} + \dots) \\ &\vdots \\ x_{q-2} + x_{2q-3} + \dots &= \mathbf{A}^{q-2}(x_t + x_{q-1} + \dots). \end{aligned} \tag{4.5}$$

That yields

$$\begin{bmatrix} y_o + y_{q-1} + \dots \\ \vdots \\ y_{q-2} + y_{2q-3} + \dots \end{bmatrix} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA}^{q-2} \end{bmatrix} (x_0 + x_{q-1} + \dots)\,. \tag{4.6}$$

Since $\begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA}^{q-2} \end{bmatrix}$ is a Vandermonde matrix multiplied by a nonsingular diagonal matrix we obtain the estimate

$$\text{wt} \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA}^{q-2} \end{bmatrix} (x_0 + x_{q-1} + \dots) \geq q - 1 - \delta \geq 3\delta - \delta \geq 2\delta.$$

So wt (v) = wt $\binom{y}{u}$ = wt (y) + wt $(u) \geq 2\delta + 2 = 2(\delta+1)$. The case $u' \neq 0$ and $y' = 0$ is analogous. The case $u' = 0$, $y' = 0$ implies that $x_0 + x_{q-1} + x_{2(q-1)} + \ldots = 0$ and it may be reduced to the anterior cases.

Let us now consider the situation where $n \geq 3$. Let

$$y_t = \begin{bmatrix} y_t^{(1)} \\ \vdots \\ y_t^{(n-1)} \end{bmatrix}, \mathbf{C} = \begin{bmatrix} (c_1) \\ \vdots \\ (c_{n-1}) \end{bmatrix},$$

where $(c_i)^t \in \mathbb{F}^\delta$ represents the ith row vector of $\mathbf{C}$, and let

$$\mathbf{A}_i = \begin{bmatrix} \alpha^{r_i+1} & 0 & \cdots & 0 \\ 0 & \alpha^{r_i+2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \alpha^{r_i+\delta} \end{bmatrix}, \ i = 0, \ldots, n-1$$

where $r_0, \ldots, r_{n-1}$ are chosen for simplicity such that no two matrices among matrices $\mathbf{A}_0, \ldots, \mathbf{A}_{n-1}$ have the same entries. This requires that the field is sufficiently large, i.e. $q - 1 \geq n\delta$.

Split the system

$$\begin{aligned} x_{t+1} &= \mathbf{A}_0 x_t + \mathbf{B} u_t \\ \begin{bmatrix} y_t^{(1)} \\ \vdots \\ y_t^{(n-1)} \end{bmatrix} &= \begin{bmatrix} (c_1) \\ \vdots \\ (c_{n-1}) \end{bmatrix} x_t + \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} u_t \end{aligned} \tag{4.7}$$

into $n-1$ systems:

$$\begin{aligned} x_{t+1} &= (\mathbf{A} - \mathbf{B}(c_i))x_t + \mathbf{B} y_t^{(i)} \\ u_t &= -(c_i)x_t + \mathbf{D} y_t^{(i)} \end{aligned} \tag{4.8}$$

and choose (c_i), such that $\det(DI - (\mathbf{A} - \mathbf{B}(c_i)) = \prod_{k=1}^{\delta}(D - \alpha^{r_i+k})$. In analogy to the discussion of the case $n = 2$ (of course more cases have to be considered) one can show that the resulting code is a MDS convolutional code of rate $1/n$ and degree δ.

Remark 4.2 A polynomial generator matrix for the $1/n$ code defined above is given by $G(D) = \left[\begin{array}{cccc} g_1(D) & g_2(D) & \ldots & g_n(D) \end{array}\right]$ where

$$g_i(D) = (D - \alpha^{r_i+1})(D - \alpha^{r_i+2}) \ldots (D - \alpha^{r_i+\delta}).$$

Indeed, in polynomial terms the iterative representation (4.8) becomes:

$$\begin{array}{rcl} (DI - (\mathbf{A} - \mathbf{B}(c_i)))x(D) & = & \mathbf{B}y^{(i)}(D) \\ u(D) & = & -(c_i)x(D) + \mathbf{D}y^{(i)}(D) \end{array}.$$

Since $\det(DI - (\mathbf{A} - \mathbf{B}(c_i)) = \prod_{k=1}^{\delta}(D - \alpha^{r_i+k})$ it implies that the components $y^{(i)}(D)$ are multiples of $\prod_{k=1}^{\delta}(D - \alpha^{r_i+k})$, $i = 1, 2, \ldots, n$. Therefore these codes have the same structure as the codes found by Justesen in [8].

CHAPTER 5

BINARY UNIT MEMORY MDS CODES

After constructing MDS convolutional codes, a natural question is to find out what is the smallest field where MDS convolutional codes exist. Since unit memory codes have the simplest representation among the codes of nonzero memory we started our study by analyzing the conditions these codes must satisfy in order to be maximum distance separable over the binary field. In this way we came to study the binary partial unit memory codes with degree $\delta = k-1$, this being the only nontrivial case where binary MDS codes exist. We follow here [27]

In this chapter we give conditions for the partial unit memory codes with degree $k-1$ to be MDS. Then we study a binary construction of unit memory codes with $\delta = k-1$ for the cases that satisfy the optimality conditions. This construction is generalized for codes over fields of characteristic $p > 2$.

Binary partial unit memory codes were studied in the literature by Lauer [10] and Justesen [9] who showed that in some situations a unit memory code performs better than the codes having the same rate and degree but memory larger than 1. Some constructions given in [10] are the inspiration of this chapter. In [9] quasi-cyclic unit memory convolutional codes are studied and some constructions and computer search results are presented. Furthermore some of the basic structural properties are discussed, such as noncatastrophicity, minimality conditions, distance measures, properties that we will use in this dissertation.

This chapter consists of 6 sections, the first two being introductory and the last one an appendix section containing material that we will heavily use. In Section 5.2 we state some equivalent conditions for binary PUM codes to be optimal and in Section 5.3 we give a method of construction for this type of code. Section 5.4 generalizes the binary construction to the case where the field has characteristic larger than 2. We add examples of both methods in Section 5.5.

5.1 Unit Memory Codes

Let $\mathbb{F}$ denote a finite field. A *unit memory code* UMC is defined through the following encoding scheme:

$$v_t = u_t G_0 + u_{t-1} G_1 \tag{5.1}$$

where $u_t \in \mathbb{F}^k$ is the k information tuple at time $t, t = 0, 1, \ldots$ and $v_t \in \mathbb{F}^n$ is the n-tuple denoting the encoded vector at time t. By convention $u_t = 0$ for $t < 0$. The matrices G_0 and G_1 are defined over the field $\mathbb{F}$ and have size $k \times n$. We assume that G_0 has rank k.

Also in polynomial representation the code is defined through the $k \times n$ polynomial encoder matrix

$$G(D) = G_0 + DG_1.$$

Following [10] we will say that two unit memory encoders: (G_0, G_1) and (G_0', G_1') are *equivalent* if there exists a nonsingular matrix T such that $G_0' = TG_0$ and $G_1' = TG_1$. Two equivalent encoders generate the same code. Two equivalent encoders are either both catastrophic or both noncatastrophic. We have the following criteria from [9]:

Theorem 5.1 *[9] A UM encoder (G_0, G_1) is catastrophic if and only if there exists an $s \times k$ matrix P of rank $s, s > 0$ and a nonsingular $s \times s$ matrix Q such that*

$$QPG_0 = PG_1$$

As a remark the degree δ of the encoder is equal to the rank of G_1 in the unit memory case. We therefore have that for any PUM code generated by (G_0, G_1) there exists an encoder (TG_0, TG_1) with T nonsingular such that the first $k - \delta$ rows of TG_1 are zero. We say that this encoder is in *standard form*. We say that a standard form encoder (G_0, G_1) is *minimal* if among all equivalent encoders, G_1 has the smallest number of nonzero rows. We have from [9]:

Theorem 5.2 *[9] A noncatastrophic UM encoder (G_0, G_1) of the form:*

$$\begin{bmatrix} G_0 & G_1 \end{bmatrix} = \begin{bmatrix} G_0' & 0 \\ G_0'' & G_1'' \end{bmatrix},$$

where G_0'', G_1'' have δ rows, is minimal if and only if:

$$\operatorname{rank} \begin{bmatrix} G_1'' \\ G_0' \end{bmatrix} = k.$$

Unit memory codes having $\delta < k$ are called *partial unit memory codes* (PUM) since the encoder requires only δ memory cells for storage.

We will discuss now the PUM codes with degree $\delta = k - 1$ and we will search for conditions they must satisfy in order to be MDS. First we work over the binary field and then over larger finite fields.

The Generalized Singleton Bound for a PUM code with degree $\delta = k-1$ becomes:

$$d_{free} \leq n,$$

and a generator matrix (G_0, G_1) has the form:

$$\begin{bmatrix} G_0 & G_1 \end{bmatrix} = \begin{bmatrix} g_1 & \dots & g_n & 0 & \dots & 0 \\ & G_0' & & & G_1' & \end{bmatrix} \tag{5.2}$$

We know that we may construct PUM codes of degree $k-1$ attaining the maximum bound n, over some finite field with enough elements (2.3). We would like to see if we may obtain such codes over a small field. We start with the field $\mathbb{F}_2$ and discuss the cases when maximum distance codes exist. A construction will be given in these specific cases. Then we generalize the construction for fields $\mathbb{F}_p, p > 2$ obtaining constructions in some other cases.

5.2 Partial Unit Memory Codes over $\mathbb{F}_2$

If $G_0,\ G_1$ generate a k/n PUM code of degree $\delta = k-1$ with maximum distance n over $\mathbb{F}_2$, then the matrices must have the following form:

$$\left[\begin{array}{cc} G_0 & G_1 \end{array}\right] = \left[\begin{array}{cccccc} 1 & \dots & 1 & 0 & \dots & 0 \\ & G_0' & & & G_1' & \end{array}\right], \text{ with rank}\,(G_1') = k-1, \tag{5.3}$$

where G_0', G_1' satisfy conditions that make the encoder G_0, G_1 noncatastrophic and minimal and the code generated using (5.1) optimal, i.e., MDS.

Remark 5.3 It can be easily shown that if $2k-1 \leq n$, the code is noncatastrophic provided that the matrix:

$$\left[\begin{array}{ccc} & G_1' & \\ 1 & \dots & 1 \\ & G_0' & \end{array}\right] \text{ has full rank } 2k-1. \tag{5.4}$$

That assures the minimality as well.

For the next theorem we need the following definition:

Definition 5.4 A block code (k,n) is called *equidistant* if all nonzero codewords have the same weight d_{min}.

If a code is equidistant and G an arbitrary $k \times n$ encoder, then the entries of G have the property that all $\mathbb{F}$-linear combinations of its rows have the same weight d_{min}. Such a matrix will be called an equidistant matrix.

Then we obtain the following theorem:

Theorem 5.5 *Let (G_0, G_1) of the form (5.3) generate an PUM-MDS code over $\mathbb{F}_2$. Then:*

1. *n is even*
2. *G_0', G_1' generate equidistant $(k-1, n)$ block codes.*

Proof: Let $u \in \mathbb{F}_2^{k-1},\ u \neq 0$ arbitrarily chosen. Let $x = \text{wt}\,[uG_0'],\ y = \text{wt}\,[uG_1']$. We prove that $x = y = n/2$. Let $u_1,\ u_{k+1} \in \mathbb{F}_2$.

Since $d_1^r = n$ we have that the weight of

$$(u_1, u, u_{k+1}) \begin{bmatrix} 1 & \dots & 1 & 0 & \dots & 0 \\ & G_0' & & & G_1' & \\ & & & 1 & \dots & 1 \end{bmatrix}$$

is greater or equal to n. By giving different values to u_1, u_{k+1} we have:
$x + y \geq n, \quad n - x + y \geq n, \quad x + n - y \geq n, \quad n - x + n - y \geq n$
$\Rightarrow \quad x = y, \quad x + y = n.$

Hence, we obtain that n is even and that

$$x = y = n/2,$$

which means that G_0', G_1' generate equidistant $(k-1, n)$ block codes. We actually can prove that $2^{k-1} \mid n$. Hence $n = 2^{k-1}j$. □

We have the following straightforward lemma:

Lemma 5.6 *The matrices G_0' and G_1' generate equidistant $(k, 2^{k-1})$ block codes if and only if the matrix*

$$\begin{bmatrix} G_0 & G_1 \end{bmatrix}$$

given in (5.3) is a generator matrix for a $(k+1, 2^k)$ equidistant block code.

Proof: Suppose G_0', G' are equidistant. If $u = (u_1, \dots, u_k) \in \mathbb{F}_2^k$ then uG_0 and uG_1 have the weight either n and respectively 0, if $(u_2, \dots, u_k) = 0$, or $n/2$, if not. Hence $\begin{bmatrix} G_0 & G_1 \end{bmatrix}$ is equidistant as well.

The other implication was just proved by the previous theorem 5.5. □

We therefore have a stronger statement:

Theorem 5.7 *Suppose (G_0, G_1) of the form (5.3) generate a PUM code over $\mathbb{F}_2$. Suppose $2k - 1 \leq n$ and condition (5.4) is satisfied (therefore the code is noncatastrophic). Then $\mathcal{C}$ is a noncatastrophic PUM-MDS convolutional code over $\mathbb{F}_2$ if and only if*

1. $n = 2^{k-1}j$.

2. G_0' and G_1' generate equidistant $(k-1, n)$ block codes.

In other words this statement gives us all the k/n MDS-PUM codes for $k \geq 4$ (so that $2k - 1 \leq 2^{k-1}$).

Proof: Theorem 5.5 gives us the necessity implication. We still need to prove the sufficiency of the two conditions. From *2.* we have that $d_0^r = n$. Let $u_1, u_{k+1} \in \mathbb{F}_2$

and $u, v \in \mathbb{F}_2^{k-1}$, so that $(u_1, u) \neq 0$. The weight

$$\text{wt } (u_1, u, u_{k+1}, v) \begin{bmatrix} 1 & \dots & 1 & 0 & \dots & 0 & & & \\ & G'_0 & & & G'_1 & & & & \\ & & & 1 & \dots & 1 & 0 & \dots & 0 \\ & & & & G'_0 & & & G'_1 & \end{bmatrix} \geq$$

$$\geq \begin{cases} \text{wt } (u_1, u)G_0 + \text{wt } (v \cdot G'_1) \geq n, \text{ if } v \neq 0 \\ \text{wt } (u_1, u)G_0 + \text{wt } (u, u_{k+1}) \begin{bmatrix} & G'_1 & \\ 1 & \dots & 1 \end{bmatrix} \geq n, \text{ if } v = 0 \end{cases},$$

because of condition (5.4). Hence $d_r^1 = n$. In the same way we have $d_i^r = n$. Also by (5.4) we have:

$$d_0^c \geq n/2,\ d_1^c \geq n/2 + 1,\ d_2^c \geq n/2 + 2, \dots, d_{n/2}^c \geq n/2 + n/2 = n \Rightarrow$$

$$\Rightarrow d_{free} = n.$$

Hence the code is MDS.

The noncatastrophicity is implied by the full rank condition on the $(2k-1, n)$ matrix. Due to this condition an infinite weight input can not produce a finite output. □

Therefore in order to construct rate $\frac{k}{2^{k-1}j}$ PUM codes with degree $\delta = k-1$ and maximum distance over $\mathbb{F}_2$, it is enough to construct rate $\frac{k}{2^{k-1}}$, $\delta = k-1$, $d_{free} = n$, MDS codes and concatenate them j times. From this, using Theorem 2.8, we get PUM-MDS codes of rate $\frac{i}{2^{k-1}j}$, $1 \leq i \leq k$.

5.3 A Binary Construction of Partial Unit Memory Codes with Maximum Free Distance

For the construction of PUM codes having maximal distance n over $\mathbb{F}_2$ we use an idea found in [10] but we will take a slightly different approach.

For that we introduce the following natural association:

Remark 5.8 Through the following isomorphism of vector spaces:

$$\begin{array}{ccc} \mathbb{F}_2[X]/(X^{2^k} - 1) & \longrightarrow & \mathbb{F}_2^{2^k} \\ a_0 + a_1 X + \dots + a_{2^k-1} X^{2^k-1} & \longmapsto & (a_0, a_1, \dots, a_{2^k-1}), \end{array} \tag{5.5}$$

any scalar encoded sequence in a PUM code $(v_0, v_1, v_2, \dots)$, given by (5.1), where $v_i \in \mathbb{F}_2^{2^k}$, may be viewed as a polynomial encoded sequence:
$(v_0(X), v_1(X), v_2(X), \dots)$, where all $v_i(X)$ are polynomials of degree at most $2^k - 1$.

Using the above isomorphism (5.5) we may also define an association between polynomial matrices $k \times 1$ and their coefficient matrices $k \times 2^k$:

$$A = \begin{bmatrix} a_{1,0} & \dots & a_{1,2^k-1} \\ a_{2,0} & \dots & a_{2,2^k-1} \\ \dots & \dots & \dots \\ a_{k,0} & \dots & a_{k,2^k-1} \end{bmatrix} \mapsto A(X) := \begin{bmatrix} a_{1,0} + a_{1,1}X + \ldots + a_{1,2^k-1}X^{2^k-1} \\ a_{2,0} + a_{2,1}X + \ldots + a_{2,2^k-1}X^{2^k-1} \\ \dots \\ a_{k,0} + a_{k,1}X + \ldots + a_{k,2^k-1}X^{2^k-1} \end{bmatrix}. \tag{5.6}$$

With this association we have that

$$\text{wt } [(u_1, \ldots, u_k)A] = \text{wt } [(u_1, \ldots, u_k)A(X)], \forall (u_1, \ldots, u_k) \in \mathbb{F}^k.$$

It follows from Definition 5.6 and the above associations that an equidistant scalar matrix has the property that the associated polynomial matrix of (5.6) has all the polynomial entries of weight 2^{k-1} and any $\mathbb{F}_2$-linear combination of those polynomials gives another polynomial of the same weight. (The weight of a polynomial is defined as the sum of the Hamming weights of all the coefficients.)

Therefore, instead of looking for $(k-1) \times (2^{k-1}-1), k \geq 2$, equidistant scalar matrices G_0', G_1', we may search for $(k-1) \times 1$ polynomial matrices with the equivalent property. For this we will heavily use Lemmas 5.19 and 5.20 in the appendix. These lemmas provide such polynomial matrices. We have the following theorem:

Theorem 5.9 *Let G_0', G_1' be $(k-1) \times (2^{k-1}-1), k \geq 4$, scalar matrices associated with*

$$G_0'(X) := \begin{bmatrix} P_1(X) \\ P_2(X) \\ \dots \\ P_{k-1}(X) \end{bmatrix}, \; G_1'(X) := \begin{bmatrix} Q_1(X) \\ Q_2(X) \\ \dots \\ Q_{k-1}(X) \end{bmatrix}, \tag{5.7}$$

where all polynomials $P_i(X)$, $Q_j(X), i, j = \overline{1, k-1}$, have degree less than or equal to $2^{k-1}-2$. Then the rate $\frac{k}{2^{k-1}}$ PUM convolutional code generated by G_0, G_1 of the form in (5.3) is a noncatastrophic MDS code over $\mathbb{F}_2$ (i.e., it has maximal distance n) if and only if:

1. Any $\mathbb{F}_2$-linear combination of polynomials $P_1(X), \ldots, P_{k-1}(X)$ and any $\mathbb{F}_2$-linear combination of polynomials $Q_1(X), \ldots, Q_{k-1}(X)$ have weight 2^{k-2}.

2. The polynomials $P_1(X), \ldots, P_{k-1}(X), Q_1(X), \ldots, Q_{k-1}(X)$ are linearly independent.

Proof: The linear independence of the polynomials is equivalent to the noncatastrophicity of the code condition given by 5.4 and the fact that all polynomials have degree strictly less than $2^{k-1}-1$. □

The following theorem will give an inductive construction of PUM codes with maximal distance n over $\mathbb{F}_2$:

Theorem 5.10 *Let $P_1(X)$, $\ldots$, $P_{k-1}(X)$ be polynomials of degree less than or equal to $2^{k-1}-2$ and weight 2^{k-2}. Moreover, suppose that any linear combination of the $k-1$ polynomials has also weight 2^{k-2}. Then the following polynomials:*

$$P_1(X)(X^{2^{k-1}}+1), \quad \ldots, \quad P_{k-1}(X)(X^{2^{k-1}}+1), \quad (X+1)^{2^{k-1}-1} \tag{5.8}$$

form a set of k polynomials with the property that any linear combination of the polynomials has degree less than 2^k and weight 2^{k-1}.

The same weight property holds for the set of k polynomials :

$$P_1(X)(X^{2^{k-1}}+1), \quad \ldots, \quad P_{k-1}(X)(X^{2^{k-1}}+1), \quad [X(X+1)]^{2^{k-1}-1}. \tag{5.9}$$

Moreover if $Q_1(X)$, $\ldots$, $Q_{k-1}(X)$ forms also a set of $k-1$ polynomials of degree less than or equal to $2^{k-1}-2$ with the same property that any linear combination of the polynomials has weight 2^{k-2} and if the polynomials

$$P_1(X), \quad \ldots, \quad P_{k-1}(X), \quad Q_1(X), \quad Q_2(X), \quad \ldots, \quad Q_{k-1}(X) \tag{5.10}$$

$$\text{and } 1+X+X^2+\ldots+X^{2^{k-1}-2}$$

are $\mathbb{F}_2$-linearly independent, then the polynomials:

$$P_1(X)(X^{2^{k-1}}+1), \ldots, P_{k-1}(X)(X^{2^{k-1}}+1), (X+1)^{2^{k-1}-1}, \tag{5.11}$$

$Q_1(X)(X^{2^{k-1}}+1), \ldots, Q_{k-1}(X)(X^{2^{k-1}}+1), [X(X+1)]^{2^{k-1}-1}$
are $\mathbb{F}_2$-linearly independent.

Proof: Let $P(X) = u_1P_1(X)+u_2P_2(X)+\ldots+u_{k-1}P_{k-1}(X), u_i \in \mathbb{F}_2$, for $i = \overline{1, k-1}$, be a linear combination of $P_1(X), P_2(X), \ldots, P_{k-1}(X)$. A linear combination of the new k polynomials has the form:

$$u(X+1)^{2^{k-1}-1} + P(X)(X^{2^{k-1}}+1) = u(X+1)^{2^{k-1}-1} + P(X)(X+1)^{2^{k-1}} =$$

$= (X+1)^{2^{k-1}-1}(u+P(X)(X+1))$, with $u \in \mathbb{F}_2$, or:
$u[X(X+1)]^{2^{k-1}-1} + P(X)(X^{2^{k-1}}+1) =$

$$= (X+1)^{2^{k-1}-1}(uX^{2^{k-1}-1} + P(X)(X+1)).$$

If $u=0$ we obtain $P(X)(X+1)^{2^{k-1}}$ that has weight twice the weight of $P(X)$ as stated in Lemma 5.22, in the appendix. If $u=1$ we use the weight retaining property (5.21):

$$\text{wt } \left[(X+1)^{2^{k-1}-1}(u+P(X)(X+1))\right] \geq$$

$$\geq \text{wt } \left[(X+1)^{2^{k-1}-1}\right] \cdot \text{wt } \left[(u+P(X)(X+1)) \mod (X+1)\right] = 2^{k-1}.$$

The second case goes the same way.

For the second part let $Q(X) = v_1Q_1(X) + \ldots + v_{k-1}Q_{k-1}(X), v_i \in \mathbb{F}_2$, $\forall i = \overline{1, k-1}$ be a linear combination of $Q_1(X), Q_2(X), \ldots, Q_{k-1}(X)$. Let

$$(X+1)^{2^{k-1}-1}(u + P(X)(X+1)) + (X+1)^{2^{k-1}-1}(vX^{2^{k-1}-1} + Q(X)(X+1)) =$$
$$= (X+1)^{2^{k-1}-1}(u + vX^{2^{k-1}-1} + (Q(X)+P(X))(X+1)) = 0,\ u, v \in \mathbb{F}_2,$$

be a linear combination of the new polynomials that is equal to zero. It implies $u = v$ and we obtain:

$$u(1 + X^{2^{k-1}-1}) + (Q(X) + P(X))(X+1) = 0 \Leftrightarrow$$
$$u(1 + X + X^2 + \ldots + X^{2^{k-1}-2}) + Q(X) + P(X) = 0,$$

which leads to $u = u_1 = \ldots = u_{k-1} = v_1 = \ldots = v_{k-1} = 0$ because of (5.11). This proves the linear independence of the new polynomials. □

Basically, Theorem 5.10 says that if we have two equidistant matrices G_0' and G_1' of sizes $(k-1) \times (2^{k-1}-1)$, $k \geq 4$, associated with the polynomial matrices $G_0'(X), G_1'(X)$ through (5.7), where the sets of polynomials $P_1(X), \ldots, P_{k-1}(X)$ and $Q_1(X), \ldots, Q_{k-1}(X)$ satisfy the conditions in Theorem 5.10, we may inductively construct equidistant matrices of size $j \times (2^j - 1)$, $j \geq k$.

For example, if we take 1 ($k = 2$), multiply it by (X^2+1) and add the extra polynomial $1 + X$, respectively $X(1+X)$, we obtain the 2×4 coefficient matrices of:

$$G_0'(X) = \begin{bmatrix} 1+X \\ 1+X^2 \end{bmatrix},\ G_1'(X) = \begin{bmatrix} (1+X)X \\ (1+X^2) \end{bmatrix},$$

and, after the next step, the 3×8 coefficient matrices of:

$$G_0'(X) = \begin{bmatrix} (1+X)^3 \\ (1+X)^5 \\ (1+X)^6 \end{bmatrix},\ G_1'(X) = \begin{bmatrix} (1+X)^3X^3 \\ (1+X)^5X \\ (1+X)^6 \end{bmatrix}.$$

Of course this is not a good choice, since the polynomials obtained are not linearly independent, the code generated in this way is catastrophic. Therefore we must change somehow these matrices in order to have the properties of Theorem 5.10. We change only the matrix $G_1'(X)$ by multiplying the entries with different powers of X modulo $X^7 - 1$. The following choice for $G_1'(X)$:

$$G_1'(X) = \begin{bmatrix} (1+X)^3 \cdot X^4 \mod (X^7-1) \\ (1+X)^5 \cdot X^3 \mod (X^7-1) \\ (1+X)^6 \cdot X^3 \mod (X^7-1) \end{bmatrix}.$$

together with the $G_0'(X)$ constructed above will satisfy the condition of the theorem. We use the polynomial entries of $G_0'(X), G_1'(X)$ for the inductive construction of Theorem 5.10. We have the following theorem:

Theorem 5.11 *Let $P_1 = (1+X)^3,\ P_2 = (1+X)^5,\ P_3 = (1+X)^6$ and $Q_1 = (1+X)^3 \cdot X^4 \mod (X^7-1),\ Q_2 = (1+X)^5 \cdot X^3 \mod (X^7-1),\ Q_3 = (1+X)^6 \cdot X^3 \mod (X^7-1)$.*

Applying Theorem 5.10 inductively we obtain rate $\frac{k}{2^{k-1}}$ noncatastrophic convolutional codes that have maximal free distance 2^{k-1} over $\mathbb{F}_2$, for all $k \geq 4$.

Remark 5.12 The rate $\frac{k}{2^{k-1}}$ code constructed above has the matrix G_0' associated with the following polynomial matrix:

$$G_0'(X) = \begin{bmatrix} (X+1)^{i_1} \\ (X+1)^{i_2} \\ \dots \\ (X+1)^{i_{k-1}} \end{bmatrix}$$

with $i_1, i_2, \dots, i_{k-1}$ nonnegative integers strictly less than 2^{k-1} of weight $k-2$, where we defined the weight of an integer in (5.19), Appendix. We apply (5.20), Appendix, to show directly that the matrix G_0' generates an equidistant $(k-1, 2^{k-1})$ block code. We use this direct approach rather than the inductive one, in the following section, for constructing MDS convolutional codes of rate k/n where n is odd. The field size must be larger.

5.4 Constructions of Partial Unit Memory Codes with Maximum Free Distance over $\mathbb{F}_p$

Let $\mathbb{F}_p$ be the field with p elements. Let $k \geq 1,\ n = p^{k-1}$.

Theorem 5.13 *Let G_0, G_1 be the $k \times n$ scalar matrices associated to the following polynomial matrices:*

$$G_0(X) = \begin{bmatrix} (X+1)^{i_0} \\ (X+1)^{i_1} \\ \dots \\ (X+1)^{i_{k-1}} \end{bmatrix},\ G_1(X) = \begin{bmatrix} 0 \\ (X+1)^{j_1} \\ \dots \\ (X+1)^{j_{k-1}} \end{bmatrix}$$

with $i_0 = (p-1) + (p-1)p + \ldots + (p-1)p^{k-2} = p^{k-1} - 1 = n-1$,
$I := \{i_1, \ldots i_{k-1}\}$ the set of all nonnegative integers with radix-p form (Lemma 5.19) having one component equal to $p-2$ and the other $k-2$ components equal to $p-1$, and $J := \{j_1, \ldots, j_{k-1}\}$, the set of all nonnegative integers having one component equal to 0 and the other $k-2$ components equal to $p-1$. Both sets have $\binom{k-1}{k-2} = k-1$ elements. Then the convolutional code generated by G_0, G_1 over $\mathbb{F}_p$ is noncatastrophic and MDS.

Proof: We compute d_0^c and d_1^c.

By (5.19) we have: wt $[(X+1)^{i_l}] = \begin{cases} (p-1)p^{k-2}, & l \neq 0 \\ p^{k-1}, & l = 0 \end{cases}$ and wt $[(X+1)^{j_l}] = p^{k-2}$.

Let $u = (u_0, \ldots, u_{k-1}) \in \mathbb{F}_p^k, u \neq 0$. Then:

wt $[uG_0] =$ wt $[uG_0(X)] =$ wt $\left[\sum_{l=0}^{k-1} u_l(X+1)^{i_l}\right] \geq$

$$\geq \text{wt } \left[(X+1)^{i_{min}}\right] \geq (p-1)p^{k-2},$$

by (5.20). We denoted by i_{min} the smallest of all integers $i_l, l \in \{0, \ldots, k-1\}$ with, $u_l \neq 0$. Therefore $d_0^c \geq (p-1)p^{k-2}$ and since there is a row of this weight we have:

$$d_0^c = (p-1)p^{k-2}.$$

For d_1^c we do the same. Let $u = (u_0, \ldots, u_{2k-1}), \in \mathbb{F}_p^{2k-1}, u \neq 0$. If $(u_1, \ldots, u_{2k-1}) = 0$, we obtain the codeword associated to $u_0(X+1)^{i_0}$ which has weight p^{k-1} by the choice of i_0. If $(u_1, \ldots, u_{2k-1}) \neq 0$ then the weight

wt $u \begin{bmatrix} G_1 \\ G_0 \end{bmatrix} =$ wt $u \begin{bmatrix} G_1(X) \\ G_0(X) \end{bmatrix} =$

$$\text{wt } \left[\sum_{s=1}^{k-1} u_l(X+1)^{j_s} + \sum_{l=1}^{k-1} u_{l+k-1}(X+1)^{i_l}\right] \geq p^{k-2},$$

by (5.20), since all the powers $i_l,\ l = \overline{0, k-1}$ differ from $j_s,\ s == \overline{1, k-1}$. Then

$$d_1^c \geq (p-1)p^{k-2} + p^{k-2} = p^{k-1} = n.$$

Therefore $d_1^c = d_{free} = n$ and the code is noncatastrophic and MDS. □

Remark 5.14 Since the main fact we used was that the sum

$$\text{wt } [(X+1)^{i_l}] + \text{wt } [(X+1)^{j_s}] \geq p^{k-1},$$

for any $l = \overline{0, k-1},\ s = \overline{1, k-1}$, we could use instead in the construction the sets I and J, with I and J formed by all nonnegative integers having the radix-p form with one component equal to $p-i$, respectively $i-2$, and the rest $k-2$ components equal to $p-1$, for all i such that $p-i > i-2$, i.e. for all $i = \overline{2, \left\lfloor \frac{p+2}{2} \right\rfloor}$. The weights wt $[(X+1)^{i_l}] = \begin{cases} (p-i+1)p^{k-2}, & l \neq 0 \\ p^{k-1}, & l = 0 \end{cases}$ and wt $[(X+1)^{j_l}] = (i-1)p^{k-2}$ have also the sum greater than p^{k-1}. Also the sets I and J formed in this way both have $k-1$ elements as it is needed.

5.5 Examples

We will give here two concrete examples to show how Theorems 5.11 and 5.13 are applied, for rates 4/8 and 3/9. After that we will discuss also the cases $k = 2, k = 3$ that are not covered by Theorem 5.11. We will use here the polynomial matrix representation $G(D) = G_0 + DG_1$.

Example 5.15 We already showed in the previous section how to choose the matrices G_0, G_1 of sizes 4×8 over $\mathbb{F}_2$. In conformity with Theorem 5.11 we have that the polynomial matrix $G(D) = G_0 + DG_1$, given by:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1+D & 1 & 1 & 1 & D & D & D & 0 \\ 1+D & 1+D & 0 & D & 1+D & 1 & 0 & 0 \\ 1+D & 0 & 1+D & D & 1 & D & 1 & 0 \end{bmatrix}$$

generates a rate 4/8 PUM convolutional code of degree $\delta = 3$ and maximum distance $d_{free} = 8$.

Example 5.16 Let G_0, G_1 be the 3×9 matrices over $\mathbb{F}_3$ associated with the following polynomial matrices:

$$\begin{bmatrix} (X+1)^8 \\ (X+1)^5 \\ (X+1)^7 \end{bmatrix}, \quad \begin{bmatrix} 0 \\ (X+1)^2 \\ (X+1)^6 \end{bmatrix}.$$

The convolutional code generated by

$$G(D) = \begin{bmatrix} 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \\ 1+D & -1 & 1 & 1-D & -1 & 1 & D & 0 & 0 \\ 1+D & 1-D & D & -1 & -1 & 0 & 1 & 1 & 0 \end{bmatrix}$$

is noncatastrophic, has degree $\delta = 2$, and maximum distance 9.

Example 5.17 We will construct rate $2/n$ PUM convolutional codes that are MDS and noncatastrophic, for all $n \geq 3$.

1. In the case n even we have a construction over the binary field. Let:

$$G(D) = \left[\underbrace{\begin{matrix} 1 & 1 & 1 & 1 \\ 1+D & 1 & 0 & D \end{matrix}}_{j \text{ times}} \right], G(D) = \left[\underbrace{\begin{matrix} 1 & 1 & 1 & 1 \\ 1+D & 1 & 0 & D \end{matrix}}_{j \text{ times}} \begin{matrix} 1 & 1 \\ 1+D & 0 \end{matrix} \right],$$

for $n = 4j$, respectively $n = 4j + 2$. Then the $2/n$ code generated by $G(D)$ is noncatastrophic and has distance n over $\mathbb{F}_2$. The code has the column distances: $d_0^c = n/2, d_1^c = n = d_{free}$ in both cases.

2. The cases where n is odd require larger fields. It turns out that the field with 3 elements is enough for a construction. Therefore, over $\mathbb{F}_3$ we obtain:

$$G(D) = \left[\underbrace{\begin{matrix} 1 & 1 & 1 & 1 \\ 1+D & 1 & 0 & D \end{matrix}}_{j \text{ times}} \begin{matrix} 1 \\ D+2 \end{matrix} \right], \tag{5.12}$$

for $n = 4j + 1$, and

$$G(D) = \left[\underbrace{\begin{array}{cccc} 1 & 1 & 1 & 1 \\ 1+D & 1 & 0 & D \end{array}}_{j \text{ times}} \begin{array}{ccc} 1 & 1 & 1 \\ 2+D & 1+D & 0 \end{array} \right], \tag{5.13}$$

for $n = 4j + 3$. The column distances are in both cases:
$d_0^c = \lfloor n/2 \rfloor + 1,\ d_1^c = n = d_{free}$.

Example 5.18 The construction of Theorem 5.7 cannot be applied in the case of $k = 3$. It turns out that any choice of binary matrices $G_0,\ G_1$ we take gives a catastrophic encoder. Therefore there is no noncatastrophic PUM binary convolutional code of rate 3/4, degree 2, having distance 4. The smallest field for which there exists a rate 3/4 code with degree 2, distance 4, is $\mathbb{F}_3$. Taking

$$G_0 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix},\ G_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

we obtain an MDS code but over a field of characteristic $p \neq 2$. The column distances are $d_0^c = 2,\ d_1^c = 2,\ d_2^c = 3,\ d_3^c = 3,\ d_4^c = 4 = d_{free}$.

5.6 Appendix

We state here some results that we used in this Chapter.
For more details see [16].

Lemma 5.19 *[16] Let $c \in \mathbb{F}, c \neq 0$ and let $i \geq 1$ with radix-p form $[i_0, i_1, \ldots, i_{m-1}]$, i.e. $i = i_0 + i_1 p + \ldots + i_{m-1} p^{m-1}$. Then:*

$$\text{wt } [(X+c)^i] = \prod_{j=0}^{m-1} (i_j + 1). \tag{5.14}$$

In particular, for $p = 2$,

$$\text{wt } [(X+1)^i] = 2^{\text{wt } (i)}, \tag{5.15}$$

where wt (i) *is the number of* 1*'s in* $\{i_0, i_1, \ldots, i_{m-1}\}$.

Lemma 5.20 *[16] Let I be any nonempty finite set of nonnegative integers with least integer i_{min} and let*

$$P(X) = \sum_{i \in I} b_i (X - c)^i,$$

where $c, b_i \in \mathbb{F},$ all nonzero. Then:

$$\text{wt } [P(X)] \geq \text{wt } [(X+c)^{i_{min}}]. \tag{5.16}$$

Lemma 5.21 *[16] For any polynomial $P(X)$ over $\mathbb{F}$, any $c \in \mathbb{F}, c \neq 0$, and any nonnegative integers n and N,*

$$\mathrm{wt}\ \left[P(X)(X^n + c)^N\right] \geq \mathrm{wt}\ \left[(X + c)^N\right] \mathrm{wt}\ \left[P(X) \mod (X^n - c)\right]. \tag{5.17}$$

The following lemma gives an obvious result that we used in the constructions of this chapter. Note that it may be seen as a corollary to Lemma 5.21:

Lemma 5.22 *If $P(X)$ is a polynomial over $\mathbb{F}_2$ of degree less or equal to $2^k - 1$, then the weight* $\mathrm{wt}\ \left[P(X)(X^{2^k} + 1)\right] = 2wt[P(X)]$.

CHAPTER 6

A GEOMETRIC PROOF OF THE EXISTENCE OF MDS CONVOLUTIONAL CODES

In this section we give a proof of the existence of MDS convolutional codes. Although it is non-constructive this proof has its value in integrating the theory of MDS convolutional codes in the general frame of algebraic geometry.

Let $\mathcal{C}$ be a convolutional code of rate k/n and degree δ defined over an arbitrary base field $\mathbb{F}$. Let G be a polynomial encoder in column proper form with ordered column degrees $\nu_1 \geq \cdots \geq \nu_k$.

The following lemma gives sufficient conditions for a code to be an MDS convolutional code:

Lemma 6.1 *If a codeword $v(D)$ in $\mathcal{C}$ has the property that any of its k components have weight at least $(\delta + 1)$ then the weight of the codeword $v(D)$ is necessarily greater than or equal to*

$$(n-k)\left(\lfloor \delta/k \rfloor + 1\right) + \delta + 1. \tag{6.1}$$

We shall refer to the property that any k components of an n component vector have weight more than $\delta + 1$ as the *weight property.*

Proof: Let

$$v(D) = \begin{pmatrix} v_1(D) & ,\ldots, & v_n(D) \end{pmatrix}^t \in \mathcal{C}.$$

The weight property implies that at least $n-k+1$ of the components of $v(D)$ must have weight at least $\lfloor \delta/k \rfloor + 1$. Indeed, taking the first k components of v, by the weight property, the sum of their weight is $\geq \delta+1$, therefore there is one component, say v_1 , with weight $\geq \lfloor \delta/k \rfloor + 1$. Cut v_1 from the sequence and add v_{k+1}.The new sequence of components has again the weight property, so there is once again a component, say v_2 with weight $\geq \lfloor \delta/k \rfloor + 1$. With this reasoning we obtain that at least $n-k+1$ of the components must have the weight at least $\lfloor \delta/k \rfloor + 1$. We have now that $n-k$ of the components have weight at least $\geq \lfloor \delta/k \rfloor + 1$, and from the weight property that the remaining k components have weight greater than $\delta + 1$. Therefore

$$\text{wt}\ (v(D)) \geq (n-k)\left(\lfloor \delta/k \rfloor + 1\right) + (\delta + 1)$$

which is equal to the upper bound (6.1).

□

The main result of this chapter now states:

Theorem 6.2 *For any rate k/n and any degree δ there exist MDS convolutional codes for sufficiently large field sizes.*

6.1 The Proof for the Situation where the Degree $\delta = 0$

We first give the proof in the particular case of the block codes in order to illustrate some major ideas used in the general case.

Let $\mathbb{K}$ denote the algebraic closure of $\mathbb{F}$. As an algebraically closed field $\mathbb{K}$ is infinite.

We know that all block codes of rate $[n, k]$ may be described by a parity check matrix of sizes $(n-k) \times n$. We prove the existence of an $(n-k) \times n$-matrix with all the full size minors invertible, without making use of the Vandermonde-parity check matrix construction. This will result into an MDS block code.

Let $\mathbf{j} = (1 \leq j_1 < j_2 < \ldots < j_{n-k} \leq n)$ be a fixed index set. We look at the sets $S_{\mathbf{j}}$ of all $(n-k) \times n$ matrices having the property that the minor formed by columns $j_1, j_2, \ldots, j_{n-k}$ is invertible. We identify a parity check matrix M with a point in the vector space $\mathbb{K}^{n(n-k)}$. Consider the polynomial ring $\mathbb{K}[X_{ij}]$, $1 \leq i \leq n-k$, $1 \leq j \leq n$. Let $P_{\mathbf{j}} \in \mathbb{K}[X_{ij}]$ be the polynomial $P_{\mathbf{j}} = \det(X_{i,j_s})_{1 \leq i,\ s \leq n-k}$. In other words $P_{\mathbf{j}} = 0$ if and only if the determinant of the minor formed by the columns $j_1, j_2, \ldots, j_{n-k}$ is zero. Let $S_{\mathbf{j}}$ be the Zariski open set defined by

$$S_{\mathbf{j}} = \{M \in \mathbb{K}^{(n-k)n} \mid P_j(M) \neq 0\}.$$

All of these sets $S_{\mathbf{j}}$ form a finite number of nonempty open sets in $\mathbb{K}^{(n-k)n}$. Indeed, for each $\mathbf{j} = (1 \leq j_1 < j_2 < \ldots < j_{n-k} \leq n)$ we choose a matrix having the identity matrix I_{n-k} on the $j_1, j_2, \ldots, j_{n-k}$ columns and this shows that $S_{\mathbf{j}} \neq \emptyset$. Therefore, the intersection of all the sets $S_{\mathbf{j}}$ is nonempty inside $\mathbb{K}^{(n-k)n}$. It implies the existence of a matrix M having the properties of all the sets $S_{\mathbf{j}}$, hence all its full size minors are invertible.

Now we have a matrix $M \in \mathbb{K}^{(n-k)n}$ with entries in $\mathbb{K}$, the algebraic closure of $\mathbb{F}$, and sitting in the intersection of all the sets $S_{\mathbf{j}}$. Since the extension $\mathbb{F} \subset \mathbb{K}$ is algebraic, it implies that every component of M is algebraic over $\mathbb{F}$, therefore also finite. If we denote with m_{ij} the components of M we have that all $m_{ij} \in \mathbb{F}[m_{ij}, 1 \leq i \leq n-k, 1 \leq j \leq n]$, which is a finite extension over $\mathbb{F}$, therefore is finite of degree, say m. Therefore the code having M as a parity check matrix will be an MDS code over a finite field $\mathbb{F}_{q^m}$, (with m possibly rather large).

Of course in the block code case we know already that taking M to be Vandermonde

$$M = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & x & \cdots & x^{n-1} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x^{(n-k-1)} & \cdots & x^{(n-1)(n-k-1)} \end{bmatrix},$$

with x a primitive element of the field $\mathbb{F} = \mathbb{F}_q$, we obtain an MDS block code. But in the situation of convolutional codes with the degree $\delta > 0$ we do not have such an explicit construction. Therefore the above proof has its value.

6.2 The General Case

In this section we prove of Theorem 6.2 in the general case. The proof mimics the one we gave in the previous section, in the block code situation. We exhibit a parameterization on the set of all rate k/n convolutional codes of degree δ using a large $\mathbb{F}$-vector space, where $\mathbb{F}$ is a finite field. Then we show that the set of codes that are not MDS forms an algebraic subset. Over the algebraic closure the algebraic subset describing the convolutional codes which are not MDS forms a strictly proper closed subset. This fact allows us to predict an MDS convolutional code with entries in a finite extension of the (finite) base field $\mathbb{F}$, (which is a finite field as well).

For the parameterization we use the first order representation as presented in Theorem 1.7 of Section 1.6. We do this by viewing a triple of matrices (K, L, M) as a point in the vector space $\mathbb{F}^{(\delta+n-k)(2\delta+n)}$. By Theorem 1.8 this parameterization is not unique. This is however of minor importance in the proof. We start the proof with a short Lemma:

Lemma 6.3 *The set of matrices (K, L, M) satisfying the property 1., 2. and 3. of Theorem 1.7 is open and nonempty inside $\mathbb{F}^{(\delta+n-k)(2\delta+n)}$.*

Proof: We recall from the paper of Ravi and Rosenthal [20] that the conditions 2. and 3. can be equivalently written as the following rank condition:

$$\underbrace{\left[\begin{array}{cccc|ccccc} K & 0 & \dots & 0 & M & 0 & \dots & \dots & 0 \\ L & K & \ddots & \vdots & 0 & M & \ddots & & \vdots \\ 0 & L & \ddots & 0 & \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & K & \vdots & & \ddots & M & 0 \\ 0 & \dots & 0 & L & 0 & \dots & \dots & 0 & M \end{array}\right]}_{2\delta-1 \text{ blocks}} \delta \text{ blocks} \tag{6.2}$$

has full row rank. Thus all (K, L, M) matrices satisfying conditions 1., 2., 3. are in the complementary set of all zeros of the polynomial equations describing the determinant of some full size minors of the matrices K and of (6.2) being 0. Therefore the set of all matrix 3-tuples (K, L, M) satisfying the conditions 1., 2., 3. is open in $\mathbb{F}^{(\delta+n-k)(2\delta+n)}$ and it is obviously nonempty since there is an one to one correspondence between this set and the set of all convolutional codes as we defined them. $\square$

The rest of the section will be devoted to the proof of the existence of MDS convolutional codes.

If one is interested in the construction of convolutional codes with some designed distance there is no limitation if one attempts to construct matrices A, B, C, D, with (A, B) controllable and (A, C) an observable pair. The following result was obtained by such a construction:

Theorem 6.4 *[23] Let $r := \max\{n-k, k\}$, and assume that the cardinality of the field $\mathbb{F}$ satisfies*

$$|\mathbb{F}| > \delta r \left\lceil \frac{\delta}{n-k} \right\rceil .$$

Then there exists a rate k/n convolutional code of degree δ having free distance

$$d_{free} \geq \delta + 1.$$

Remark 6.5 The proof of Theorem 6.4 as given in [23] comes with a concrete construction of a set of matrices A, B, C, D. Observe that for very high rates the free distance of $\delta + 1$ is only a fraction away from the optimal upper bound (2.3). For low rates the distance of $\delta + 1$ is less than optimal.

In order to prove Theorem 2.5 we will require a strengthening of Theorem 6.4:

Theorem 6.6 *Let δ, k, n, ρ be fixed and assume that*

$$\rho = \delta \left(2 \left\lceil \frac{\delta}{n-k} \right\rceil + \left\lfloor \frac{\delta}{k} \right\rfloor + 1 \right) .$$

If the matrices A, B, C have the property that $(B\ AB \ldots\ A^{\rho-1}B)$ is the parity check of an MDS block code and that $\left(C^t\ A^tC^t \ldots\ {A^{\rho-1}}^tC^t\right)$ is the generator matrix of an MDS block code then for any codeword

$$v(D) = \begin{pmatrix} y(D) \\ u(D) \end{pmatrix} \in \mathbb{F}^n[D]$$

either

$$\text{wt }(u(D)) \geq \delta + 1 \quad \textit{or} \quad \text{wt }(v(D)) \geq (n-k)\left(\lfloor \delta/k \rfloor + 1\right) + \delta + 1.$$

Proof: Assume

$$\begin{aligned} u(D) &= u_0 D^\gamma + u_1 D^{\gamma-1} + \cdots + u_\gamma, \\ y(D) &= y_0 D^\gamma + y_1 D^{\gamma-1} + \cdots + y_\gamma, \end{aligned}$$

where γ is the degree of v, and that $u_0 \neq 0$. The first equations of the systems (1.25) give that (see [23]):

$$(u_\gamma, \ldots, u_0)^t \in \ker \left(B\ AB \ldots\ A^\gamma B\right).$$

If $\gamma < \rho$ then wt $(u(D)) \geq \delta + 1$ and the proof is complete.

We therefore assume that $\gamma \geq \rho$ and that wt $(u(D)) \leq \delta$. By the 'pigeonhole principle' there exist an index $i < \rho - \frac{\rho}{\delta}$ and an input sequence

$$u_{i+1} = u_{i+2} = \cdots = u_{i+\frac{\rho}{\delta}} = 0.$$

In analogy to the proof of [25, Theorem 3.1] it follows that the state $x_{i+1} \neq 0$ and that

$$\begin{pmatrix} y_{i+1} \\ y_{i+2} \\ \vdots \\ y_{i+\frac{\rho}{\delta}} \end{pmatrix} = \begin{pmatrix} C \\ CA \\ \vdots \\ CA^{\frac{\rho}{\delta}-1} \end{pmatrix} x_{i+1}.$$

The assumption on the matrix $\left(C^t \ A^tC^t \ldots \ A^{\rho-1^t}C^t\right)$ gives that

$$\begin{aligned} \text{wt } (y) &\geq (n-k)\cdot\frac{\rho}{\delta} - \delta + 1 = (n-k)\left(2\left\lceil\frac{\delta}{n-k}\right\rceil + \left\lfloor\frac{\delta}{k}\right\rfloor + 1\right) - \delta + 1 \\ &\geq 2\delta + (n-k)\left(\left\lfloor\frac{\delta}{k}\right\rfloor + 1\right) - \delta + 1 = (n-k)\left(\left\lfloor\frac{\delta}{k}\right\rfloor + 1\right) + \delta + 1. \end{aligned}$$

□

In the proof of Theorem 6.2 we need the following lemma:

Lemma 6.7 *Let δ, k, n, ρ be fixed, $r = \max\{n-k, k\}$ and assume that the cardinality of the field $\mathbb{F}$ satisfies $|\mathbb{F}| > r\rho$. Then there exist matrices A, B, C satisfying the conditions of Theorem 6.6.*

Proof: Let $\alpha \in \mathbb{F}$ be an element of multiplicative order at least $r\rho$. Then

$$A := \begin{pmatrix} \alpha^r & 0 & \cdots & 0 \\ 0 & \alpha^{2r} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \alpha^{\delta r} \end{pmatrix}, \quad B := \begin{pmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^{k-1} \\ 1 & \alpha^2 & \alpha^4 & \cdots & \alpha^{2(k-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \alpha^\delta & \alpha^{2\delta} & \cdots & \alpha^{\delta(k-1)} \end{pmatrix},$$

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

satisfy the conditions of Theorem 6.6. □

Proof of Theorem 6.2: Let $\mathbb{F}$ be a fixed finite field, with q elements, having characteristic p. Let $\mathbb{K}$ denote the algebraic closure of $\mathbb{F}$, $\mathbb{K}$ is infinite. We shall call a matrix with all full size minors invertible, an MDS matrix.

Consider now some fixed numbers δ, k, n, ρ with $k < n$ and ρ chosen as in Theorem 6.6.

We look at the set of all 3-tuple matrices (K, L, M) with the properties $1, 2, 3$ and of sizes as in Theorem 1.7, such that the matrix $[K \mid M]$ is an MDS matrix. Let this set be denoted by V. Then V is the intersection of two open nonempty sets, one given by all (K, L, M) such that the conditions $1, 2, 3$ are satisfied, and the other given by the complement of the set of the zeroes of all the full size minors of $[K \mid M]$. Over the algebraic closure $\mathbb{K}$, the intersection of nonempty open sets is nonempty and V is therefore a nonempty Zariski open set in $\mathbb{K}^{(\delta+n-k)(2\delta+n)}$.

Let now (K, L, M) be an element of V, and let

$$\mathbf{j} = \{1 \le j_1 < j_2 < \ldots < j_k \le n\}$$

be a subset of the set $\{1, \ldots, n\}$ having cardinality k. We would like to show that the code $\mathcal{C}$ defined by (K, L, M) has the property that the k components $\{v_{j_i}(D) \mid i = 1, \ldots, k\}$, of a code word $v(D) \in \mathcal{C}$ satisfy either

$$\sum_{i=1}^{k} \text{wt } (v_{j_i}(D)) \ge \delta + 1 \quad \text{or wt } (v(D)) \ge (n-k)\left(\lfloor \delta/k \rfloor + 1\right) + \delta + 1.$$

In order to apply Theorem 6.6, let $P_{\mathbf{j}}$ be an $n \times n$ permutation matrix such that

$$P_{\mathbf{j}} v(D) = \begin{pmatrix} y(D) \\ u(D) \end{pmatrix}$$

where the k components $v_{j_1}(D), \ldots, v_{j_k}(D)$ of $v(D)$ are mapped onto the k components of $u(D)$.

Partition the matrix $MP_{\mathbf{j}}^{-1} = [M_1 \mid N]$ where M_1 is the matrix formed by the first $n-k$ columns of $MP_{\mathbf{j}}^{-1}$ and N denotes the rest of the columns in $MP_{\mathbf{j}}^{-1}$. The property of V tells us that the matrix $[K \mid M_1]$ is invertible.

For every K, L, M and every $\mathbf{j}$ we define matrices $\mathbf{A_j}, \mathbf{B_j}, \mathbf{C_j}, \mathbf{D_j}$ in the following way:

$$[K \mid M_1]^{-1} \left[K \mid L \mid MP_{\mathbf{j}}^{-1}\right] =: \left[\begin{array}{c|c|cc} I_\delta & -\mathbf{A_j} & 0 & -\mathbf{B_j} \\ 0 & -\mathbf{C_j} & I_{n-k} & -\mathbf{D_j} \end{array}\right]. \tag{6.3}$$

Rewriting the equation (1.23) in the new terms we obtain the $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$ polynomial description from the previous chapter:

$$\begin{bmatrix} DI_\delta - \mathbf{A_j} & 0 & -\mathbf{B_j} \\ -\mathbf{C_j} & I_{n-k} & -\mathbf{D_j} \end{bmatrix} \begin{bmatrix} x(D) \\ y(D) \\ u(D) \end{bmatrix} = 0. \tag{6.4}$$

If the matrices $\mathbf{A_j}, \mathbf{B_j}, \mathbf{C_j}$ satisfy the conditions of Theorem 6.6 then the weight $\sum_{i=1}^{k} \text{wt } (v_{j_i}(D)) \ge \delta + 1$ or the weight of $v(D)$ is larger than the bound (2.3).

The algebraic conditions on $\mathbf{A}, \mathbf{B}, \mathbf{C}$ expressed in Theorem 6.6 translate into algebraic conditions inside the parameter space $\mathbb{K}^{(\delta+n-k)(2\delta+n)}$. Let

$$S_{\mathbf{j}} = \{(K, L, M) \in \mathbb{F}^{(\delta+n-k)(2\delta+n)} \text{ s.t.}$$

$$\left(\mathbf{B_j}\ \mathbf{A_j}\mathbf{B_j} \ldots\ \mathbf{A_j}^{\rho-1}\mathbf{B_j}\right) \text{ and } \left(\mathbf{C}_{\mathbf{j}}^t\ \mathbf{A}_{\mathbf{j}}^t\mathbf{C}_{\mathbf{j}}^t \ldots\ \mathbf{A_j}^{\rho-1^t}\mathbf{C}_{\mathbf{j}}^t\right) \text{ are MDS}\}.$$

Applying Lemma 6.7 one sees that $S_{\mathbf{j}} \cap V$ is a nonempty Zariski open subset of $\mathbb{K}^{(\delta+n-k)(2\delta+n)}$.

Let $J = \{\mathbf{j} = \{1 \le j_1 < j_2 < \ldots < j_k \le n\}\}$ be the set of all k-subsets of $\{1, \ldots, n\}$, and consider all $\{S_{\mathbf{j}} \cap V \mid \mathbf{j} \in J\}$. All of these sets form a finite number of open nonempty sets in V, therefore their intersection is nonempty. It implies the existence of a vector $x = (K, L, M)$ having the property of all the sets $S_{\mathbf{j}} \cap V$.

We have obtained a vector $x \in V$ having the components in $\mathbb{K}$, the algebraic closure of $\mathbb{F}$, and lying in the intersection

$$\bigcap_{\mathbf{j} \in J} (S_{\mathbf{j}} \cap V) \subset V \subset \mathbb{K}^{(\delta+n-k)(2\delta+n)}.$$

Since the extension $\mathbb{F} \subset \mathbb{K}$ is algebraic, it implies that every component of x is algebraic over $\mathbb{F}$, therefore in a finite extension. If we denote by x_j the components of x we have that all $x_j \in \mathbb{F}[x_j, 1 \le j \le (\delta+n-k)(2\delta+n)]$, which is a finite extension over $\mathbb{F}$, therefore is finite of degree say m. Therefore the code $\mathcal{C} = \mathcal{C}(K, L, M)$ associated to the matrices (K, L, M) will be a code over a finite field $\mathbb{F}_{q^m}$, (with m possibly rather large).

We show that this code is actually an MDS convolutional code, in other words it has the free distance equal to the upper bound (2.3). Let

$$v(D) = \begin{pmatrix} v_1(D) & , \ldots, & v_n(D) \end{pmatrix}^t \in \mathcal{C}$$

be a nonzero code word. We will show that the weight of $v(D)$ is larger than the upper bound by applying Lemma 6.1 and Theorem 6.6.

Since the code $\mathcal{C}$ belongs to the intersection of all the Zariski open sets $S_{\mathbf{j}} \cap V$, we can apply Theorem 6.6 for all the k-combinations of the components $v_1, v_2, \ldots, v_n$ to form the part u of the codeword. By construction of the sets $S_{\mathbf{j}} \cap V$, we obtain that either the weight of the k-combination of components $v_1, v_2, \ldots, v_n$ is greater than $\delta + 1$, or the weight of the whole codeword is larger than

$$(n-k)\left(\lfloor \delta/k \rfloor + 1\right) + \delta + 1$$

which is the bound we want. From the first situation for all k-combinations of the components we obtain the conditions of Lemma 6.1. The weight of the codeword is therefore greater then the upper bound (2.3). In either case we predict the existence of an MDS code $\mathcal{C}$ over the finite field $\mathbb{F}_{q^m}$. $\square$

6.3 Remarks on the Geometry of the Proof

We conclude this section with some remarks about the algebraic geometric aspect of the constructions considered above.

As explained in [13, 19, 20] a submodule of rank k and degree δ in $\mathbb{F}^n[D]$ describes a quotient sheaf of rank k and degree δ over the projective line $\mathbb{P}^1$. The column degrees $\nu_1 \geq \cdots \geq \nu_k$ of the submodule $\mathcal{C} \subset \mathbb{F}^n[D]$ correspond then to the Grothendieck indices of the quotient sheaf. By a general theorem of Grothendieck it is possible to equip the set of all rank k submodules (quotient sheaves) of degree δ with the structure of a scheme. Such a scheme is referred to as a *quot scheme* in the algebraic geometry literature. The quot scheme that parameterizes the rank k submodules of degree δ is a smooth projective variety [19].

If the degree $\delta = 0$, the Grothendieck quot scheme is exactly the Grassmannian variety $\mathrm{Grass}(k, \mathbb{F}^n)$ consisting of all k dimensional subspaces of the vector space $\mathbb{F}^n$. This variety parameterizes the set of all linear block codes of rate $\frac{k}{n}$ defined over the field $\mathbb{F}$. For an arbitrary degree δ the Grothendieck quot scheme parameterizes in a natural way all rate $\frac{k}{n}$ convolutional codes of degree δ.

CHAPTER 7

CONCLUSIONS

In this dissertation we have defined the notion of MDS convolutional codes as codes attaining the upper bound:

$$d_{free} \leq (n-k)\left(\lfloor \delta/k \rfloor + 1\right) + \delta + 1.$$

These codes naturally generalize the MDS block codes. One example of a rate k/n, degree δ, MDS convolutional code, is given by a code $\mathcal{C} \subset \mathbb{F}^n[D]$ having the property that any of its codewords $v(D) \in \mathcal{C}, \mod (D^N - 1), n \mid N$, is a codeword in an $[N, K, N-K+1], K/N \leq k/n$, Reed Solomon code.

Another method of obtaining MDS convolutional codes is by taking codes $\mathcal{C} \subset \mathbb{F}^n[D]$ that have the property that the truncations of its codewords, $v(D) \mod D^j$, have large weight. The codewords will then have their weight uniformly distributed, a property desired for a high error correction capability. In the dissertation, we present this construction in the particular case of rate $1/2$, degree δ, the most interesting case in convolutional coding theory. The results of this case can be generalized to the case of rate k/n codes. This is planned for future research.

Another study of this dissertation is the existence of MDS code over the binary field. Binary MDS block codes exist only for parameters $[n, 1, n]$, and the dual, $[n, n-1, 2]$, and they are called trivial MDS block codes. Binary MDS convolutional codes exist in the case of rate $k/2^{k-1}$ and degree $\delta = k-1$, the case of partial unit memory codes. We discuss this case and then generalize it to codes over fields $\mathbb{F}_p$ with p prime.

Finally, we have taken a systems theory approach and showed a way of constructing MDS convolutional codes of rate $1/n$ and degree δ in this language. Also, using these representation tools and some results from algebraic geometry, we have given a non-constructive proof of the existence of MDS codes.

BIBLIOGRAPHY

[1] B.M. Allen. *Linear Systems Analysis and Decoding of Convolutional Codes*. PhD thesis, University of Notre Dame, August 1999. Available at http://www.nd.edu/~rosen/preprints.html.

[2] A. R. Calderbank, G. D. Forney, Jr., and A. Vardy. Minimal tail-biting trellises: the Golay code and more. *IEEE Trans. Inform. Theory*, 45(5):1435–1455, 1999.

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

[4] G. D. Forney. Minimal bases of rational vector spaces, with applications to multivariable linear systems. *SIAM J. Control*, 13(3):493–520, 1975.

[5] R. Johannesson and K. Zigangirov. Distances and distance bounds for convolutional codes – an overview. In *Topics in Coding Theory. In honour of L. H. Zetterberg.*, Lecture Notes in Control and Information Sciences # 128, pages 109–136. Springer Verlag, 1989.

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

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

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

[9] J. Justesen, E. Paaske, and M. Ballan. Quasi-cyclic unit memory convolutional codes. *IEEE Trans. Inform. Theory*, IT-36(3):540–547, 1990.

[10] G.S. Lauer. Some optimal partial-unit-memory codes. *IEEE Trans. Inform. Theory*, 25:240–243, 1979.

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

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

[13] V. Lomadze. Finite-dimensional time-invariant linear dynamical systems: Algebraic theory. *Acta Appl. Math*, 19:149–201, 1990.

[14] I. G. Macdonald. *Symmetric functions and Hall polynomials.* The Clarendon Press Oxford University Press, New York, second edition, 1995. With contributions by A. Zelevinsky, Oxford Science Publications.

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

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

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

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

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

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

[21] J. Rosenthal. An algebraic decoding algorithm for convolutional codes. In G. Picci and D.S. Gilliam, editors, *Dynamical Systems, Control, Coding, Computer Vision: New Trends, Interfaces, and Interplay*, pages 343–360. Birkäuser, Boston-Basel-Berlin, 1999.

[22] J. Rosenthal. Connections between linear systems and convolutional codes. In B. Marcus and J. Rosenthal, editors, *Codes, Systems and Graphical Models*, IMA Vol. 123, pages 39–66. Springer-Verlag, 2001.

[23] J. Rosenthal, J. M. Schumacher, and E. V. York. On behaviors and convolutional codes. *IEEE Trans. Inform. Theory*, 42(6, part 1):1881–1891, 1996.

[24] J. Rosenthal and R. Smarandache. Maximum distance separable convolutional codes. *Appl. Algebra Engrg. Comm. Comput.*, 10(1):15–32, 1999.

[25] J. Rosenthal and E. V. York. BCH convolutional codes. *IEEE Trans. Inform. Theory*, 45(6):1833–1844, 1999.

[26] R. M. Roth and G. Seroussi. On generator matrices of MDS codes. *IEEE Trans. Inform. Theory*, 31(6):826–831, November 1985.

[27] R. Smarandache. Unit memory convolutional codes with maximum distance. In B. Marcus and J. Rosenthal, editors, *Codes, Systems and Graphical Models*, IMA Vol. 123, pages 381–396. Springer-Verlag, 2001.

[28] R. Smarandache, H. Gluesing-Luerssen, and J. Rosenthal. Constructions for MDS-convolutional codes. *IEEE Trans. Inform. Theory*, 47(5):2045–2049, 2001.

[29] R. P. Stanley. *Enumerative combinatorics. Vol. 2.* Cambridge University Press, Cambridge, 1999. With a foreword by Gian-Carlo Rota and appendix 1 by Sergey Fomin.

[30] R. M. Tanner. Convolutional codes from quasi-cyclic codes: A link between the theories of block and convolutional codes. University of California, Santa Cruz, Tech Report UCSC-CRL-87-21, November 1987.

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

[32] E. V. York. *Algebraic Description and Construction of Error Correcting Codes, a Systems Theory Point of View.* PhD thesis, University of Notre Dame, 1997. Available at http://www.nd.edu/˜rosen/preprints.html.