University of Zurich

Master Thesis

Constructions of Rank Metric Convolutional Codes for Random Linear Network Coding

Author:
Kilian Werder

Supervisor:
Prof. Dr. Joachim Rosenthal

Co-supervisor:
Dr. Julia Lieb

University of Zurich[UZH]

Institute of Mathematics
University of Zurich

February 1, 2022

"*Quod Erat Demonstrandum*"

Euclid

Abstract

Physical channels like the internet are highly used in the modern world. This high usage ultimately leads to transmission bottlenecks and longer waiting times. Random linear network coding, a network protocol, enables an increase in data throughput compared to the widely used broadcasting protocol. It uses linear combinations of encoded data packets. Within the proposed network protocol, rank metric codes are applied to improve the error handling and decoding capability. Thereby, the selection of the code is crucial and depends on the chosen use case and the desired properties of the protocol. Several code constructions were published for this purpose.

This thesis introduces linear network coding, defines upper bounds on code distances, gives an overview of different constructions and presents two concrete network models: the rank deficient sliding window network with delay and without delay. We introduce four suitable constructions of rank metric codes: memory maximum sum rank code, maximum rank profile code, maximum rank distance code and (partial) unit memory code. The codes differ in structure, distance properties and the ground field. These differences are inspected, advantages and disadvantages are highlighted and the range of applications according to network coding for these codes are analysed. Furthermore, we compare the maximum rank distance codes with the memory maximum sum rank codes in detail and conclude that both codes can be applied in random linear network coding. Additionally, the generalised Singleton bound and the j-th column sum rank upper bound are juxtaposed and we prove that these bounds are not connected.

Acknowledgements

I would like to thank Prof. Dr. Joachim Rosenthal for the three exciting courses and the possibility to write my master thesis under his supervision. He sparked my interest in writing my thesis in the field of algebra.

I am grateful to my co-supervisor Dr. Julia Lieb for all the e-mails and hours of Zoom meetings. She was always willing to answer my questions and to help me with problems of understandings. Moreover, she supported me in finding the current topic for my Master's thesis and shared constructive thoughts. Without her, this thesis would not have been possible. It was a pleasure to have her as my co-supervisor.

I would also like to express my deepest gratitude to my friends at the university, especially Luca Festini, Simon Egli and Yu Lucy Huan. Together, we overcame a lot of frustration, mathematical problems and exams but also celebrated passed exams and semesters. I do not know how I survived the last five years without these amazing people.

I also wish to thank Vera Bernhard for her motivation, troubleshooting of my English language and encouragement throughout this work and my studies before.

Last but not least, I would like to thank my family for their unwavering support throughout my entire life. Without them, I would not be where I am now.

Contents

Abstract **iii**

Acknowledgements **v**

1 Introduction **1**
1.1 Background . . . 1
1.2 Outline . . . 2

2 Introduction to Coding Theory and Rank Metric Codes **5**
2.1 Coding Theory . . . 5
2.1.1 Block Codes . . . 5
2.1.2 Convolutional Codes . . . 7
2.2 Rank Metric Codes . . . 14

3 Introduction to Random Linear Network Coding **21**
3.1 Overview . . . 21
3.2 Basic Concepts of Random Linear Network Coding . . . 23
3.3 Error Handling . . . 25
3.3.1 Constant Dimension Codes . . . 26
3.3.2 Rank Metric Codes . . . 26

4 Network Models **27**
4.1 Introduction . . . 27
4.2 Framework . . . 27
4.3 Models . . . 28
4.3.1 Rank Deficient Sliding Window Network without Time Delay . . . 28
Encoder . . . 29
Transmission . . . 29
Decoder . . . 30
4.3.2 Rank Deficient Sliding Window Network with Delay . . . 32
Encoder . . . 32
Transmission . . . 32
Decoder . . . 32

5 Constructions **39**
5.1 Construction of Memory Maximum Sum Rank Codes . . . 40
5.1.1 Code Construction . . . 40
5.1.2 Numerical Results . . . 54
5.1.3 Reducing the Field Size . . . 55
5.2 Construction of Maximum Rank Profile Codes . . . 57
5.3 Construction of Maximum Rank Distance Codes . . . 62
5.3.1 Preliminaries . . . 62
5.3.2 Code Construction . . . 65

5.4 (Partial) Unit Memory Codes . . . 72
5.4.1 Preliminaries . . . 72
5.4.2 Code Construction . . . 73
5.4.3 Properties of (P)UM$_G$ Codes . . . 77
UM$_G(n,k)$ codes . . . 77
PUM$_G(n,k,k^{(1)})$ codes . . . 79
Degree and Field Size . . . 80
5.4.4 Summary of (P)UM$_G$ Codes . . . 80
5.5 Summary of the Constructions . . . 81

6 Comparison of Constructions and Bounds **83**
6.1 Juxtaposition of MRD and m-MSR Convolutional Codes . . . 84
6.1.1 Code Structures . . . 84
Equalities on Rank Metric Block Codes . . . 84
Equalities on Rank Metric Convolutional Codes . . . 87
6.1.2 Decoding Properties . . . 90
6.1.3 Summary . . . 93
6.2 Comparison of the Sum Rank Bounds . . . 94

7 Conclusion and Further Research **101**

A Overview of the Presented Code Constructions **103**

List of Figures

1.1 Channel coding. 1

2.1 Shift register of a 1/2-rate convolutional code. 7
2.2 Shift register in state 10 with input 1 over $\mathbb{F}_2$. 8
2.3 State graph. 8
2.4 Minimal trellis graph for codewords in $\mathcal{C}_j^{(r)}$. 19
2.5 Minimal trellis graph for codewords in $\mathcal{C}_j^{(c)}$. 19
2.6 Minimal trellis graph for codewords in $\mathcal{C}_j^{(rc)}$. 20

3.1 The butterfly network. 21
3.2 The butterfly network with linear network coding. 22
3.3 Linear network coding example. 24

4.1 Example of a rank deficient sliding window network. 30

List of Tables

5.1 Achievable code constructions by Mahmood et al. 54
5.2 Achievable code constructions by Almeida et al. 55
5.3 Submatrices of the (P)UM$_G$ code. 75

6.1 Comparison of m-MSR and MRD codes. 84

Chapter 1

Introduction

1.1 Background

Data transfer is of great importance in human society, especially in the current age of digitalisation and with the impact of the corona pandemic. A lot of us surf through the internet, send e-mails or stream films and series. All these activities send a vast amount of data through the internet. Therefore, it is not surprising that researchers try to optimise and improve the network. During this kind of investigation, it has been shown that certain connections in the network constitute a bottleneck. As a consequence, some data has to wait for a long time to be transmitted through these links. This delay creates several issues: For example, it can cause a longer loading time when streaming a video. To overcome this problem, a new protocol combines some data in one package, which reduces the bottleneck. More precisely, the idea is to send linear combinations of packets instead of individual packets through the network. This strategy is called linear network coding. [FS07, HL08, MS12]

Today, information is sent through physical channels. The internet constitutes one of the most famous physical channels. Further examples of such channels are the wireless LAN, network or USB cable. In addition to the problem of a low throughput, it is usually not possible to transmit the data without errors due to physical limitations. The current solution to ensure that the receiver is able to decipher the erroneous message is provided by coding theory: The original message is encoded such that the receiver can correct or at least detect errors. Again, the errors occur due to the transmission channel. In the best case, the receiver can correct the mistake. In the other case, the receiver is able to identify the occurrence of an error which in turn can be reported back to the transmitter. Then, the sender should resend the data. [Lin99, Rot06, Bie17]

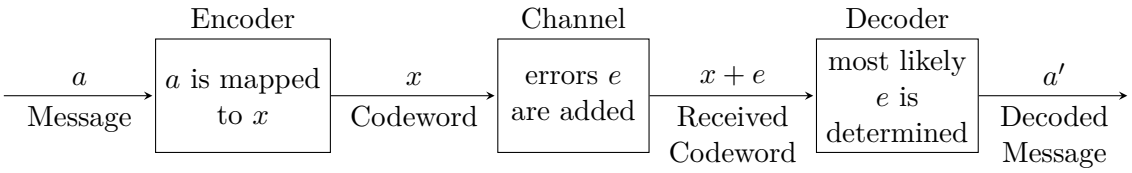


FIGURE 1.1: Channel coding. [Lin99, Figure 1]

Figure 1.1 summarises the described protocol according to [Lin99] and [Rot06]. First, a message a is sent by the transmitter. This message is encoded into a codeword x, which then contains some redundancy. This redundancy can be used to detect or correct errors. Afterwards, the message is sent over a noisy channel. For this reason, some unintended errors e are added. The receiver obtains the received codeword $x + e$. With error correction techniques, one can estimate e and can calculate the original codeword. As the last step, the primary message is reconstructed, called a'.
The following example should clarify this procedure:

Example 1.1.1. We have the following message-codeword mapping over $\mathbb{F}_2^4$:

Message a	Codeword x
Nord	0000
South	1111

Alice sends the message "South" to Bob to signal him that she is south of him. In other words, $a =$"South". Alice encodes the message and sends the codeword 1111 to Bob over the internet. During the transmission, an error occurs because the internet is a noisy channel. The error appears at the rightmost bit. This means $e = 0001$ and $x + e = 1110$. Bob receives $x + e$. He realises that 1110 is not an image of the encoding map and hence it is an invalid codeword. In turn, this implies that an error has occurred during the transmission. If Bob assumes that only one error occurs, he can reconstruct the original codeword 1111. This works since all possible vectors of the received codewords with one changed position are:

0110 1010 1100 1111

The only valid codeword of this list is 1111. As a consequence, the transmitted codeword is 1111, which translates to "South". Therefore, Alice is south of him.

Example 1.1.2. The second example shows an extended situation to the first example. Again, Alice sends her position. To have more flexibility, she is also able to send "West" or "East". The new encoding is:

Message a	Codeword x
Nord	0000
East	0101
South	1111
West	1010

As before, Alice sends the message "South". The sent codeword x is 1111 and the error $e = 0001$. This results in the received codeword 1110. Bob again recognises that this codeword is not valid. He assumes that one error occurred and lists the possible codewords:

0110 1010 1100 1111

This time, he cannot definitely identify the original codeword since 1010 and 1111 are both valid codewords. This means he cannot reconstruct the message, but he knows that something went wrong and can ask Alice to send the message a second time.

1.2 Outline

Chapter 2 will give an introduction and the formal definitions of coding theory. It will introduce block codes, convolutional codes and rank metric codes.
The idea of network coding will be explained in Chapter 3. Examples will show the differences to the broadcasting protocol and will explain the procedure of the random linear network coding protocol. The effects of transmission errors will be addressed at the end of that chapter.
The basic concept of network coding can be adapted to different network models of which two of them will be presented in Chapter 4. One permits time delay during the transmission, whereas the other model allows it. The chapter will also provide an overview of fundamental

properties corresponding to the particular model. Furthermore, we will investigate requirements that guarantee that the original message can be reconstructed by the recipient with certainty.
This thesis will focus on specific constructions of codes for network coding. In the last couple of years, suitable codes for network coding were developed by mathematicians and engineers. They are equipped with convenient properties which meet the requirements for the discussed networks. We will present four constructions in detail in Chapter 5. Furthermore, we will show the corresponding properties and will compare them with the given network models. This will allow us to determine which construction fits which network model.
The last Chapter 6 will compare the constructions defined before. First, we will analyse the two different matrix constructions and will decide if they coincide. Second, we will contrast two different distance bounds and will evaluate if they correlate.
The thesis will be ended with an overall conclusion and we will highlight some possible further works.

Chapter 2

Introduction to Coding Theory and Rank Metric Codes

2.1 Coding Theory

Chapter 1 has given a short overview of coding theory. This chapter will now provide a detailed introduction to coding theory. It is based on [Lin99, Rot06, Bie17].

2.1.1 Block Codes

This subsection will introduce the simplest category of codes: The block codes. We have already seen some examples of block codes (see Example 1.1.1 and Example 1.1.2). Now, we will give the definition of block codes and some corresponding theory.

Definition 2.1.1. (Block Code) [Rot06, Section 1.3] An (n, M) *block code* over a finite alphabet, usually over a finite field $\mathbb{F}_q$ with q elements, is a nonempty subset $\mathcal{C}$ of cardinality M of $\mathbb{F}_q^n$. n is called the *code length* and M is called *code size*. The elements of a code are called *codewords*.

Definition 2.1.2. (Hamming Weight) [Rot06, Section 1.3] Let x be an element in $\mathbb{F}_q^n$. The *Hamming weight* $wt(x)$ of x is the number of non-zero entries in x.

In general, we use codes to gain good error detection and correction performances. Hence, we are interested in measuring how much the codewords differ from each other. For that reason we define the following Hamming distance:

Definition 2.1.3. (Hamming Distance) [Rot06, Section 1.3] Let x_1 and x_2 be two codewords in $\mathbb{F}_q^n$. The *Hamming distance* $d(x_1, x_2)$ of the two codewords is defined as the number of coordinates on which x and y differ. It holds:

$$d(x_1, x_2) = wt(x_1 - x_2)$$

Definition 2.1.4. (Minimum Distance) [Rot06, Section 1.3] Let $\mathcal{C}$ be an (n, M) block code with $M > 1$. The *minimum distance* of $\mathcal{C}$ is the minimum Hamming distance between any two distinct codewords of $\mathcal{C}$. This means the minimum distance, denoted as d, is defined as:

$$d = \min_{c_1, c_2 \in \mathcal{C} \,:\, c_1 \neq c_2} d(c_1, c_2) = \min_{c_1, c_2 \in \mathcal{C} \,:\, c_1 \neq c_2} wt(c_1 - c_2)$$

Usually, we denote the minimum distance d of $\mathcal{C}$ as $d(\mathcal{C})$.

Definition 2.1.5. (Linear Block Code) [Rot06, Section 2.1] Let $\mathbb{F}_q$ be a finite field of order q. An (n, k, d) block code is called *linear* if $\mathcal{C}$ is a linear subspace of $\mathbb{F}_q^n$ over $\mathbb{F}_q$. Namely, for every two codewords $c_1, c_2 \in \mathcal{C}$ and two scalars $a_1, a_2 \in \mathbb{F}_q$, we have $a_1 c_1 + a_2 c_2 \in \mathcal{C}$. n is the length of the block code, k is the dimension of the subspace and d is the minimum distance.

Proposition 2.1.6. (Minimum Distance for Linear Block Codes) [Rot06, Proposition 2.1]
Let $\mathcal{C}$ be a linear block code. Then, it holds:

$$d(\mathcal{C}) = \min_{c_1,c_2 \in \mathcal{C} \,:\, c_1 \neq c_2} d(c_1, c_2) = \min_{c_1,c_2 \in \mathcal{C} \,:\, c_1 \neq c_2} wt(c_1 - c_2) = \min_{c \in \mathcal{C} \,:\, c \neq 0} wt(c)$$

Definition 2.1.7. (Code Rate) [Rot06, Section 2.1] Let k be the dimension and n be the code length of a linear block code. Then, the *code rate* R is defined as:

$$R = \frac{k}{n}$$

Each linear block code is a subspace that is spanned by a set of vectors. This set of vectors form a generator matrix that represents the encoding function.

Definition 2.1.8. (Generator Matrix) [Rot06, Section 2.1 and Section 2.2] Let G be a matrix in $\mathbb{F}_q^{k \times n}$. We define the linear block code $\mathcal{C}$ as:

$$\mathcal{C} = \{a \cdot G \,|\, a \in \mathbb{F}_q^k\}$$

G is then the *generator matrix* of $\mathcal{C}$.

Definition 2.1.9. (Systematic Encoder) [Rot06, Section 2.2] G is a *systematic encoder* if it has the form

$$\left[\; I \mid A \; \right]$$

where I is a $k \times k$ identity matrix and A is a $k \times (n-k)$ matrix.

Lemma 2.1.10. [Rot06, p. 29]
Each linear block code up to permutations of coordinates has a systematic encoder G.

Proof. We can assume that the k row vectors are linearly independent since k is the dimension of the subspace. We are allowed to permutate the coordinates by assumption. This implies that we can switch the columns of G such that the first k columns are linearly independent as well. Then, we use fundamental row operations to get an equivalent matrix G'. It is a systematic encoder and creates an equivalent code. □

Definition 2.1.11. (Parity Check Matrix) [Rot06, Section 2.3] Let $\mathcal{C}$ be a linear block code and H be an $(n-k) \times n$ matrix such that

$$\mathcal{C} = \{c \in \mathbb{F}_q^n \,|\, Hc^T = 0\}.$$

Then, H is the *parity check matrix* of $\mathcal{C}$.

Lemma 2.1.12. [Rot06, p. 29] Let G be the generator matrix and H be the parity check matrix of a linear block code $\mathcal{C}$. Then

$$HG^T = 0 \qquad \text{and} \qquad GH^T = 0$$

holds.

Theorem 2.1.13. [Rot06, Theorem 2.2] Let H be a parity check matrix of a linear block code $\mathcal{C} \neq \{0\}$. The minimum distance d is the largest integer such that every $d-1$ columns in H are linearly independent.

Lemma 2.1.14. (Error Correction) [Rot06, Proposition 1.3 and Proposition 1.4] Let $\mathcal{C}$ be a block code with minimum distance d. Then, the following holds:

- One can detect up to $d-1$ errors.
- One can correct up to $\left\lfloor \frac{d-1}{2} \right\rfloor$ errors.

Theorem 2.1.15. (Singleton Bound) [Bie17, Theorem 4.1 and Corollary 4.2] If $\mathcal{C}$ is a block code over $\mathbb{F}_q$ with minimum distance d and length n, then we have

$$|\mathcal{C}| \leq q^{n-d+1}.$$

Additionally, if $\mathcal{C}$ is a linear block code with dimension k, then it holds $|\mathcal{C}| = q^k$ and we get

$$d \leq n - k + 1.$$

These bounds are called the *Singleton bounds*.

Definition 2.1.16. (Maximum Distance Separable (MDS) Block Code) [Bie17, Theorem 4.1] A block code is called *MDS* if it achieves the Singleton bound from Theorem 2.1.15.

2.1.2 Convolutional Codes

Convolutional codes can be seen as a generalisation of block codes. The biggest difference is that convolutional codes connect different message encodings. More general, the encoding of message number t depends on the inputs before. This means if we encode three messages, the encoding of the third message can depend on the first, second and third input. In other words, convolutional codes concatenate the different inputs. We first start with the most popular introductory example of convolutional codes, inspired by [Lin99, Chapter 13]. Subsequently, we will provide the proper definition and some consequences.

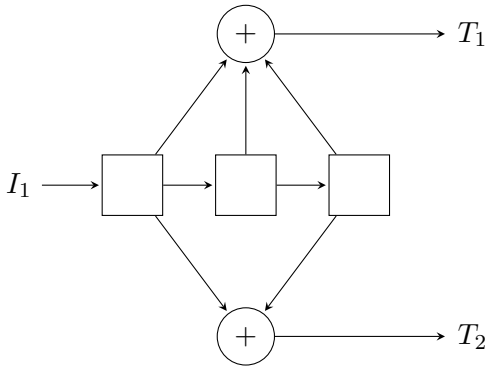


FIGURE 2.1: Shift register of a 1/2-rate convolutional code. [Lin99, Figure 4]

Figure 2.1 shows a shift register which illustrates a 1/2-rate convolutional code over $\mathbb{F}_2$. I_1 is the input value, also called message, and (T_1, T_2) is the two-dimensional codeword. The first square in the middle stands for the input and the following two represent the state. Each of these two squares stands for a previous input. The length of the state is called the memory of the code. The memory in the example is equal to 2 because the encoder must remember two previous inputs to generate the new output when the current input is presented. Usually, the shift register starts with previous inputs equal to zero. This means the starting value is 000. If we load a 1 as input, every entry in the state is shifted one place to the right. The number in the rightmost state position is disregarded. The new number receives the free space on the left. The new entries are 100. The output is $T_1 = 1, T_2 = 1$ over $\mathbb{F}_2$. Then

the state changes to 10. If the next message is again a 1, we reach the entries 110. This results in the output $T_1 = 0$, $T_2 = 1$ (see Figure 2.2). The new state for the next codeword is 11.

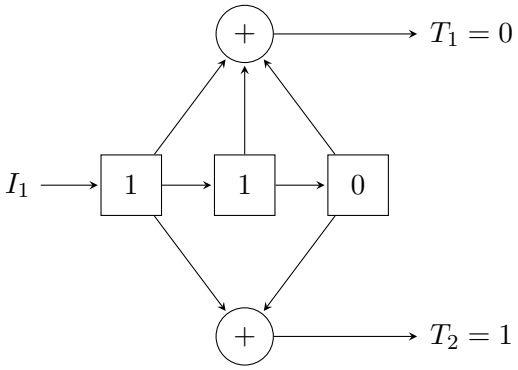


FIGURE 2.2: Shift register in state 10 with input 1 over $\mathbb{F}_2$. [Lin99, Figure 4]

A state graph shows the encoding of a convolutional code graphically. Every possible state of the code is depicted with a corresponding node. The outgoing edges represent the chosen input values and the generated output. They are labelled in the form "input / output". The edges link to the new state we get after the new input is processed. Figure 2.3 shows such a state graph for the example above. The dashed lines represent the input 0 and the solid lines stand for the input 1. For example, the left upper solid edge shows that if the encoder is in state 10 and we add the input 1, we receive 0 and 1 as output. The encoder ends in the state 11.

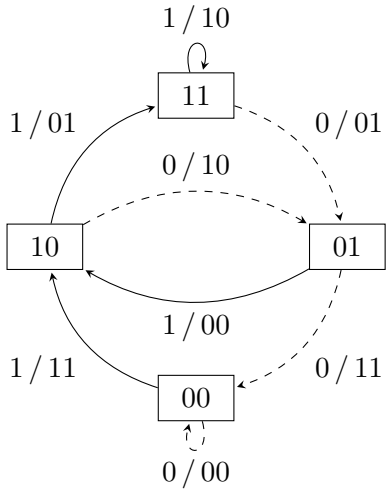


FIGURE 2.3: State graph. [Lin99, Figure 4]

We can set convolutional codes in a mathematical context by writing the input as a polynomial $I_1(D) = i_0 + i_1 D + i_2 D^2 + \dots$ where i_0 is the value of the first message, i_1 the second and so on. The same holds for the output sequences $T_1(D) = t_0^{(1)} + t_1^{(1)} D + t_2^{(1)} D^2 + \dots$ and $T_2(D) = t_0^{(2)} + t_1^{(2)} D + t_2^{(2)} D^2 + \dots$. We can translate the example above into the following formulas:

$$i_0 = 1,\ i_1 = 1 \implies I_1(D) = 1 + D$$

$$t_0^{(1)} = 1,\ t_1^{(1)} = 0 \implies T_1(D) = 1$$

$$t_0^{(2)} = 1,\, t_1^{(2)} = 1 \implies T_2(D) = 1 + D$$

The number of input lines (here I_1) is called the dimension $k = 1$. The code length of a convolutional code is the number of output lines (here T_1, T_2) $n = 2$.

The code rate is defined in the same way as for block codes by $k/n = 1/2$.

There also exists a generator matrix that defines the entire convolutional code. The size of this generator matrix is $k \times n$ similar to block codes but over $\mathbb{F}_q[D]$. The generator matrix for the beginning example is

$$G = \begin{bmatrix} 1 + D + D^2 & 1 + D^2 \end{bmatrix}.$$

The output sequence of the input $1 + D$ can be calculated in the following way, remember that the base field in this example is $\mathbb{F}_2$:

$$(1 + D) \cdot G = [1 + D^3, 1 + D + D^2 + D^3]$$

This shows that the input sequence

$$1 + D = 1\ 1\ 0\ 0\ \ldots$$

produces the output

$$[1 + D^3, 1 + D + D^2 + D^3] = [1, 1] + [0, 1]D + [0, 1]D^2 + [1, 1]D^3 = 11\ 01\ 01\ 11\ 00\ 00\ \ldots\ .$$

At this point, we can present the formal definition of a convolutional code. This is based on [LPR21].

Definition 2.1.17. (Convolutional Code) [LPR21, Subsection 1.2.1] A *convolution code* $\mathcal{C}$ is an $\mathbb{F}_q[D]$-submodule of $\mathbb{F}_q[D]^n$ of rank k. A $k \times n$ matrix $G(D)$ with entries in $\mathbb{F}_q[D]$, whose rows constitute a basis of $\mathcal{C}$, is called a *generator matrix* for $\mathcal{C}$. Hence,

$$\mathcal{C} = \{u(D)G(D) \,|\, u(D) \in \mathbb{F}_q[D]^k\} \subseteq \mathbb{F}_q[D]^n.$$

Definition 2.1.18. (Rate of a Convolutional Code) [LPR21, Subsection 1.2.1] The *rate* of a convolutional code is defined by k/n, like for block codes.

Definition 2.1.19. (Basic Generator Matrix) [WZSS15, Section IV.A] A generator matrix $G(D)$ is *basic* if it is polynomial and if there exists a polynomial right inverse $G^{-1}(D)$ such that $G(D) \cdot G^{-1}(D) = I_{k\times k}$.

Definition 2.1.20. (Degree) [LPR21, Subsection 1.2.1] Let $G(D)$ be a generator matrix. The maximal degree of all full-size $k \times k$ minors of the generator matrix of a convolutional code $\mathcal{C}$ is called the *degree* of $\mathcal{C}$. It is usually denoted by δ.

Definition 2.1.21. (Row Degree) [WZSS15, Section II.D] Let $i = 1, 2, \ldots, k$. The largest degree of any entry in row i of the generator matrix $G(D)$ is called the *row degree* ν_i. More precise, if $G(D) = (g_{ij}(D))_{i,j=1}^{i=k,\,j=n}$ then

$$\nu_i = \max_{j\in\{1,2,\ldots,n\}} \deg(g_{ij}).$$

Definition 2.1.22. (Memory) [WZSS15, Section II.D] Let ν_i for $i = 1, \ldots, k$ be the row degrees. The *memory* of $G(D)$ is:

$$\mu = \max_{i=1,\ldots,k} \nu_i$$

Remark 2.1.23. In other words, the memory μ is the largest degree of any entry in the generator matrix $G(D)$. More precise, if $G(D) = (g_{ij}(D))_{i,j=1}^{i=k,\,j=n}$ then

$$\mu = \max_{\substack{i\in\{1,2,\dots,k\}\\ j\in\{1,2,\dots,n\}}} \deg(g_{ij}).$$

Definition 2.1.24. (Code Notation) We will use two different notations for a convolutional code $\mathcal{C}$ with length n, dimension k, memory μ and degree δ:

- $\mathcal{C}[n, k, \mu]$ with the declaration of the memory μ
- $\mathcal{C}(n, k, \delta)$ with the declaration of the degree δ

Definition 2.1.25. (Overall Constraint Length) [WZSS15, Section II.D] Let $\mathcal{C}$ be a convolutional code with length n, dimension k and $\nu_1, \dots, \nu_k$ the row degrees. The *overall constraint length* ν is defined as:

$$\nu = \sum_{i=1}^{k} \nu_i$$

Definition 2.1.26. (Minimal Generator Matrix) [LPR21, Subsection 1.2.1] Let $G(D)$ be a generator matrix, $\nu = \sum_{i=1}^{k} \nu_i$ be the overall constraint length and δ be the degree of the code. If $\delta = \nu$, then we call the generator matrix *minimal*.

Theorem 2.1.27. (Existence of Minimal Generator Matrix) [LPR21, p. 7]
Given a generator matrix $G(D)$, there always exists a minimal generator matrix equivalent to $G(D)$. As a consequence, all convolutional codes admit a minimal generator matrix.

Definition 2.1.28. (Leading Coefficient Matrix) [WZSS15, Section IV.A] Let $G(D) = [g_{ij}(D)]$ be a $k \times n$ matrix with $i \in [1, k]$ and $j \in [1, n]$, ν_i be the row degrees. Then, $[G(D)]_h$ is the *leading coefficient matrix* of $G(D)$ over a finite field $\mathbb{F}_q$ if it has the following entries:

$$(i, j) = \begin{cases} \text{leading coefficient of } g_{ij}(D) & \text{if } \deg(g_{ij}(D)) = \nu_i \\ 0 & \text{otherwise} \end{cases}$$

Lemma 2.1.29. [LPR21, Lemma 1.2.4] An encoding matrix $G(D)$ over $\mathbb{F}_q[D]$ is minimal if and only if the leading coefficient matrix $[G(D)]_h$ has full row rank.

Example 2.1.30. Let $G(D)$ be the generator matrix over $\mathbb{F}_5$:

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

This code has length $n = 3$ and dimension $k = 2$. The degree can be determined by calculating all 2×2 minors. Therefore, the degree is $\delta = 4$. The row degrees are $\nu_1 = 2$ and $\nu_2 = 2$ which implies that the overall constraint length is 4. It is equal to the degree δ. As a result, the generator matrix is minimal.

The corresponding leading coefficient matrix $[G(D)]_h$ is

$$\begin{bmatrix} 0 & 3 & 1 \\ 0 & 0 & 1 \end{bmatrix}.$$

As expected, this matrix has full row rank.

Now, we give the definition of the Hamming distance for convolutional codes. There exist different distances for convolutional codes. Later in this chapter, there will be presented some other distances.

Definition 2.1.31. (Hamming Weight for Convolutional Codes) [LPR21, Definition 1.2.12] Recall that the Hamming weight $wt(c)$ of $c \in \mathbb{F}_q^n$ is defined as the number of non-zero components of c. The Hamming weight of a polynomial vector $v(D) = \sum_{i=0}^{\deg(v(D))} v_i D^i$ with $v(D) \in \mathbb{F}_q[D]^n$ and $v_i \in \mathbb{F}_q^n$ is defined as

$$wt(v(D)) = \sum_{i=0}^{\deg(v(D))} wt(v_i).$$

Definition 2.1.32. (Hamming Distance) [LPR21, Definition 1.2.12] Let $v_1(D), v_2(D) \in \mathbb{F}_q[D]^n$ be two codewords. Then, the *Hamming Distance* is defined as:

$$d_H(v_1(D), v_2(D)) = wt(v_1(D) - v_2(D)).$$

Definition 2.1.33. (The Free Hamming Distance) [LPR21, Definition 1.2.13 and Lemma 1.2.15] Let $\mathcal{C}$ be a convolutional code. The *free Hamming distance* is defined as

$$d_H(\mathcal{C}) = \min_{\substack{v_1(D), v_2(D) \in \mathcal{C}, \\ v_1(D) \neq v_2(D)}} d_H(v_1(D) - v_2(D)) = \min_{\substack{v(D) \in \mathcal{C}, \\ v(D) \neq 0}} wt(v(D)).$$

Definition 2.1.34. (Catastrophic Generator Matrix) [JZ15, p. 56] A convolutional code is *catastrophic* if there exists an infinite weight input $u(D)$ that results in a finite weight codeword $v(D)$.

Theorem 2.1.35. [Dho94, p. 60] The convolutional code $\mathcal{C} = \mathrm{Im}(G(D))$ is *non-catastrophic* $\iff$ $G(D)$ is basic.

Remark 2.1.36. [LPR21, p. 8] Being catastrophic is a code property. If we have a code with a non-catastrophic generator matrix, then all other generator matrices of this code are non-catastrophic as well. This is the case since generator matrices of a convolutional code differ by left multiplication with a unimodular matrix.

Theorem 2.1.37. (Parity Check Matrix for Convolutional Codes) [LPR21, Theorem 1.2.8] A convolutional code $\mathcal{C}$ can be equivalently described by using an $(n-k) \times n$ full rank polynomial *parity check matrix* $H(D)$ instead of the generator matrix $G(D)$ if and only if $\mathcal{C}$ is non-catastrophic. $H(D)$ is defined as

$$\mathcal{C} = \ker(H(D)) = \left\{ v(D) \in \mathbb{F}_q[D]^n \mid H(D)v(D)^T = 0 \in \mathbb{F}_q[D]^{n-k} \right\}.$$

Lemma 2.1.38. [Dho94, p. 36] Let $G(D)$ be the generator matrix and $H(D)$ be the parity check matrix of a convolutional code $\mathcal{C}$. Then, as for block codes,

$$H(D)G(D)^T = 0 \qquad \text{and} \qquad G(D)H(D)^T = 0$$

hold.

Definition 2.1.39. (Systematic Encoder for Convolutional Codes) [AMPN20, p. 3] $G(D)$ is a *systematic encoder* if it has the form

$$G(D) = \left[\; I_k \mid P(D) \; \right]$$

where I_k is a $k \times k$ identity matrix and $P(D)$ is a $k \times (n-k)$ matrix.
The corresponding parity check matrix

$$H(D) = \left[\begin{array}{c|c} -(P(D))^T & I_{n-k} \end{array} \right]$$

where I_{n-k} is an $(n-k) \times (n-k)$ identity matrix and $-(P(D))^T$ is an $(n-k) \times k$ matrix. We can check that $G(D)H(D)^T = -I_k P(D) + P(D)I_{n-k} = 0$.

Remark 2.1.40. [AMPN20, p. 3] In contrary to block codes, not every convolutional code admits a systematic encoder. This means there exist convolutional codes that do not have a systematic encoder, not even after column permutations.

Theorem 2.1.41. (Error Correction) [LPR21, Theorem 1.2.14] Let $\mathcal{C}$ be a convolutional code with free Hamming distance d. Then, the following holds:

- One can detect up to $d-1$ errors.
- One can correct up to $\left\lfloor \frac{d-1}{2} \right\rfloor$ errors.

With convolutional codes, we can encode (and connect) an infinite number of messages. This works as the input stream is a polynomial. In some cases, we are interested in a fixed number j of messages and codewords. This happens for example if we encode messages with a fixed predefined length. A truncated sliding generator matrix extracts from the initial $G(D)$ the parts we need for our window. Then, we get a new generator matrix over $\mathbb{F}_q$, which can be seen as a block code.

Definition 2.1.42. (Truncated Sliding Generator Matrix) [LPR21, p. 12] Let $G(D)$ be the generator matrix of an (n, k, δ) convolutional code. Let μ be the memory of $G(D)$. $G(D) = G_0 + G_1 D + G_2 D^2 + \cdots + G_\mu D^\mu$, where $G_\mu \neq 0$ and $G_i = 0 \ \forall i > \mu$. Then, the truncated sliding generator matrix is

$$G_{trunc}(j) = \begin{bmatrix} G_0 & G_1 & \dots & G_j \\ & G_0 & \dots & G_{j-1} \\ & & \ddots & \vdots \\ & & & G_0 \end{bmatrix} \in \mathbb{F}_q^{k(j+1) \times n(j+1)}$$

for any $j \geq 0$, and a truncated codeword can be represented as

$$v_{[0,j]} = (v_0, v_1, \cdots, v_j) = (u_0, u_1, \cdots, u_j) G_{trunc}(j),$$

where

$$v_{[0,j]} \in \mathbb{F}_q^{n(j+1)} \qquad \text{and} \qquad u_{[0,j]} \in \mathbb{F}_q^{k(j+1)}.$$

Definition 2.1.43. (Truncated Sliding Parity Check Matrix) [AN21, p. 6] There also exists a *truncated sliding party check matrix* $H_{trunc}(j)$ of $H(D) = \sum_{i=0}^{\mu} H_i D^i$:

$$H_{trunc}(j) = \begin{bmatrix} H_0 & & & \\ H_1 & H_0 & & \\ \vdots & \vdots & \ddots & \\ H_j & H_{j-1} & \dots & H_0 \end{bmatrix} \in \mathbb{F}_q^{(n-k)(j+1) \times n(j+1)}$$

for any $j \geq 0$, $H_i = 0$ when $i > \mu$. A truncated codeword can be checked by

$$H_{trunc}(j) v_{[0,j]}^T = 0.$$

There also exists a distance measure for truncated codewords:

Definition 2.1.44. (j-th Column Distance) [LPR21, Definition 1.2.18] Let $\mathcal{C}$ be a convolutional code and $j \geq 0$. The *j-th column distance* is defined as

$$d_H^j(\mathcal{C}) = \min_{v(D)\in\mathcal{C}, v_0\neq 0} \{wt(v_{[0,j]}(D))\} = \min_{v_0\neq 0} \{wt((v_0, v_1, \ldots, v_j) \mid H_{trunc}(j)(v_0, \ldots, v_j)^T = 0)\}.$$

If G_0 has full row rank and therefore, $v_0 \neq 0$ if and only if $u_0 \neq 0$, we get

$$d_H^j(\mathcal{C}) = \min_{u_0\neq 0} wt((u_0, u_1, \ldots, u_j)G_{trunc(j)}).$$

Theorem 2.1.45. (Generalised Singleton Bound) [LPR21, Theorem 1.2.21] Let $\mathcal{C}$ be an (n, k, δ) convolutional code. Then, the generalised Singleton bound states

$$d_H(\mathcal{C}) \leq (n-k)\left(\left\lfloor \frac{\delta}{k} \right\rfloor + 1\right) + \delta + 1.$$

Remark 2.1.46. The generalised Singleton bound can be reduced to the Singleton bound for block codes by setting $\delta = 0$.

Definition 2.1.47. (Maximum Distance Separable (MDS) Convolutional Code) [LPR21, p. 13] A convolutional code is called *MDS* if it achieves the generalised Singleton bound from Theorem 2.1.45.

Theorem 2.1.48. (j-th Column Distance Bound) [LPR21, Theorem 1.2.21] Let $\mathcal{C}$ be an (n, k, δ) convolutional code. Then, it holds $\forall j \geq 0$

$$d_H^j(\mathcal{C}) \leq (n-k)(j+1) + 1.$$

Definition 2.1.49. (Memory Maximum Distance Separable (m-MDS) Code) [MBK16, Section I] A code is called *m-MDS* if it reaches the j-th column distance upper bound for all j until the code memory μ, i.e., $d_H^j(\mathcal{C}) = (n-k)(j+1)+1$ for all $0 \leq j \leq \mu$.

2.2 Rank Metric Codes

So far, codewords of block codes are vectors and with convolutional codes specifically, polynomial vectors. Now, let us consider a new class of codes, the rank metric codes, where the codewords are matrices. This means $\mathcal{C} \subseteq \mathbb{F}_q^{m\times n}$. Rank metric codes are used for postquantum cryptography, local repair in distributed storage, space-time coding and secure linear network coding [AMPN20].

We will use the $\mathbb{F}_q-$linear isomorphism ϕ between $\mathbb{F}_{q^m}^n$ and $\mathbb{F}_q^{m\times n}$ to receive a rank metric code. ϕ denotes the $\mathbb{F}_q$-linear vector space isomorphism that expands every scalar in $\mathbb{F}_{q^m}$ to a column vector in $\mathbb{F}_q^m$ depending on a basis. With this approach, one can extend the block codes and convolutional codes to rank metric codes.

Example 2.2.1. Consider ϕ as the isomorphism between $\mathbb{F}_{2^2}^3$ and $\mathbb{F}_2^{2\times 3}$:

$$\mathbb{F}_{2^2} \cong \frac{\mathbb{F}_2[x]}{x^2+x+1} \cong \{a+bx \mid a,b \in \mathbb{F}_2\}$$

Let α be a zero of x^2+x+1. So, $\{1,\alpha\}$ is a basis of $\mathbb{F}_{2^2}$ over $\mathbb{F}_2$. Then,

$$\phi\left([\alpha, 1, 1+\alpha]\right) = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}.$$

We will now formally define the rank metric codes. That part is based on [AMPN20, AN21, WZSS15, MBK16, Mah15].

Definition 2.2.2. (Rank Metric Code) [Mah15, Section 2.5] A *rank metric code* is a $\mathbb{F}_{q^m}$-linear block code $\mathcal{C} \subseteq \mathbb{F}_{q^m}^n$ endowed with the *rank distance*. Let c be a codeword in $\mathcal{C}$. $c \in \mathbb{F}_{q^m}^n$ is a vector, but can also be seen as a matrix: $\phi(c) \in \mathbb{F}_q^{m\times n}$. The rank of the codeword c is the rank of the corresponding matrix: $\operatorname{rank}(c) = \operatorname{rank}(\phi(c))$.
The *rank distance* between any two codewords $c_1, c_2 \in \mathbb{F}_{q^m}^n$ is defined as:

$$d_R(c_1, c_2) = \operatorname{rank}(\phi(c_1) - \phi(c_2))$$

Definition 2.2.3. (Minimum Rank Distance) [AMPN20, p. 4] Let $\mathcal{C}$ be a rank metric code. As for the Hamming distance, the *minimum rank distance* of $\mathcal{C}$ is defined as:

$$d_R(\mathcal{C}) = \min\{d_R(v,w) \mid v,w \in \mathcal{C}, v \neq w\} = \min\{\operatorname{rank}(v) \mid v \in \mathcal{C}, v \neq 0\}$$

Theorem 2.2.4. (Singleton Bound for Rank Metric Codes) [AMPN20, p. 4] Let $\mathcal{C} \subseteq \mathbb{F}_{q^m}^n$ be an (n,k)-rank metric code. If $m \geq n$, the following bound holds:

$$d_R(\mathcal{C}) \leq n-k+1 \tag{2.1}$$

Definition 2.2.5. (Maximum Rank Distance - MRD) [AMPN20, p. 4] A rank metric code is called *MRD* if its minimum rank distance achieves the Singleton bound (2.1).

Remark 2.2.6. [AMPN20, p. 4] If we disregard the restriction of $m \geq n$, then the Singleton bound looks like:

$$d_R(\mathcal{C}) \leq \min\left\{1, \frac{m}{n}\right\}(n-k)+1$$

Therefore, a code with $m < n$ cannot be an MRD code and has a lower minimum distance. In addition, a codeword has at most rank m. So, the code cannot have codewords with rank up to n. These are the reasons why we are more interested in codes with $m \geq n$. From now on, we assume that $m \geq n$ if not stated otherwise.

Theorem 2.2.7. (Gabidulin Code) [Gab85] Let $g_0, \dots, g_{n-1} \in \mathbb{F}_{q^m}$ be a set of elements that are linearly independent over $\mathbb{F}_q$. The generator matrix for a Gabidulin code $\mathcal{GA}[n,k] \subseteq \mathbb{F}_{q^m}^n$ is given by

$$\begin{bmatrix} g_0 & g_1 & \cdots & g_{n-1} \\ g_0^{[1]} & g_1^{[1]} & \cdots & g_{n-1}^{[1]} \\ \vdots & \vdots & \ddots & \vdots \\ g_0^{[k-1]} & g_1^{[k-1]} & \cdots & g_{n-1}^{[k-1]} \end{bmatrix}$$

where $g^{[j]} = g^{q^j}$. Then, this code is MRD.

We can also define convolutional rank metric codes. As with the block codes, the convolutional codes in $\mathbb{F}_{q^m}[D]^n$ and the isomorphism ϕ can be employed for this purpose. As a result, the codewords are sequences of matrices:

Definition 2.2.8. (Rank Metric Convolutional Code) [AMPN20, Section 2.2] A *rank metric convolutional code* is a $\mathbb{F}_{q^m}[D]$-linear convolutional code $\mathcal{C} \subseteq \mathbb{F}_{q^m}[D]^n$ endowed with the *sum rank distance*. Let $v(D) = v_0 + v_1 D + v_2 D^2 + \dots$ be a codeword in $\mathcal{C}$. $v_i \in \mathbb{F}_{q^m}^n$ is a vector, but can also be seen as a matrix: $\phi(v_i) \in \mathbb{F}_q^{m \times n}$ for all $i \geq 0$. Again, the rank of v_i is the rank of the corresponding matrix: $\operatorname{rank}(v_i) = \operatorname{rank}(\phi(v_i))$.
The *sum rank distance* between any two codewords $v(D), w(D) \in \mathbb{F}_{q^m}[D]^n$ is defined as:

$$d_{SR}(v(D), w(D)) = \sum_{i \geq 0} \operatorname{rank}(\phi(v_i) - \phi(w_i))$$

Definition 2.2.9. (Free Sum Rank Distance) [AN21, p. 7] The *free sum rank distance* of a convolutional code $\mathcal{C} \subseteq \mathbb{F}_{q^m}[D]^n$ is defined as:

$$\begin{aligned} d_{SR}(\mathcal{C}) &= \min \{ d_{SR}(v(D), w(D)) \,|\, v(D), w(D) \in \mathcal{C}, v(D) \neq w(D) \} \\ &= \min \left\{ \sum_{i \geq 0} \operatorname{rank}(v_i) \,|\, v(D) \in \mathcal{C}, v(D) \neq 0 \right\} \end{aligned}$$

Since the sum rank distance is upper bounded by the Hamming distance, the generalised Singleton bound also holds for the sum rank free distance:

Theorem 2.2.10. [AN21, p. 7] Let $\mathcal{C}$ be an (n,k,δ)-rank metric convolutional code. Then, it holds:

$$d_{SR}(\mathcal{C}) \leq (n-k) \left(\left\lfloor \frac{\delta}{k} \right\rfloor + 1 \right) + \delta + 1$$

Proof. Let $v(D) = v_0 + v_1 D + v_2 D^2 + \dots \in \mathcal{C}$. For all $i \geq 0$, each zero entry in v_i reduces the highest possible column rank of $\phi(v_i)$ by one. This implies that $\operatorname{rank}(\phi(v_i)) \leq wt_H(v_i)$ and $d_{SR}(v(D)) \leq d_H(v(D))$. As a consequence, $d_{SR}(\mathcal{C}) \leq d_H(\mathcal{C})$. As the Hamming distance is upper bounded by the generalised Singleton bound by Theorem 2.1.45, the sum rank free distance is bounded by it as well. □

Definition 2.2.11. (j-th Column Sum Rank Distance) [AN21, p. 7] The *j-th column sum rank distance* for $j \geq 0$ of a convolutional code $\mathcal{C} \subseteq \mathbb{F}_{q^m}[D]^n$ is defined as:

$$d_{SR}^j(\mathcal{C}) = \min \left\{ \sum_{i=0}^{j} \operatorname{rank}(v_i) \,|\, v(D) \in \mathcal{C}, v_0 \neq 0 \right\}$$

Remark 2.2.12. [GLRS06, Section II.] There is the following property for the j-th column sum rank distance:

$$0 \leq d_{SR}^0(\mathcal{C}) \leq d_{SR}^1(\mathcal{C}) \leq \dots \leq d_{SR}(\mathcal{C})$$

Hence,

$$d_{SR}(\mathcal{C}) = \lim_{j\to\infty} d^j_{SR}(\mathcal{C}).$$

Proposition 2.2.13. (j-th Column Sum Rank Distance Bound) [AMPN20, p. 7] Let $\mathcal{C} \subseteq \mathbb{F}_{q^m}[D]^n$, then

$$d^j_{SR}(\mathcal{C}) \leq (j+1)(n-k)+1.$$

Proof. It is known that $d^j_H(\mathcal{C}) \leq (j+1)(n-k)+1$ and the sum rank distance is upper bounded by the Hamming distance. So, we get:

$$d^j_{SR}(\mathcal{C}) \leq d^j_H(\mathcal{C}) \leq (j+1)(n-k)+1$$

□

Remark 2.2.14. [LPR21, p. 13]

$$d^j_{SR}(\mathcal{C}) \leq d_{SR}(\mathcal{C}) \leq (n-k)\left(\left\lfloor\frac{\delta}{k}\right\rfloor+1\right)+\delta+1$$

So, the j-th column sum rank distance is also bounded by the generalised Singleton bound. Hence, $j = L := \left\lfloor\frac{\delta}{k}\right\rfloor + \left\lfloor\frac{\delta}{n-k}\right\rfloor$ is the largest possible value of j for which $d^j_{SR}(\mathcal{C})$ can attain the upper bound $(n-k)(j+1)+1$.

Proposition 2.2.15. [GLRS06, Proposition 2.1] If one of the first n columns of $H_{trunc}(j)$ is in the span of some other $d-1$ columns of that matrix, it follows $d^j_H(\mathcal{C}) \leq d$.

Proposition 2.2.16. [AMPN20, p. 7] For a systematic (n,k) convolutional code $\mathcal{C}$ with an encoder of memory μ, it can be seen that the number $(n-k)(\mu+1)+1$ is the maximum possible value for the free sum rank (and Hamming) distance of $\mathcal{C}$.

Proof. Since we have a systematic convolutional code, there exists a systematic parity check matrix of the form $H(D) = \left[\begin{array}{c|c} -(P(D))^T & I_{n-k}\end{array}\right]$ where I_{n-k} is an $(n-k)\times(n-k)$ identity matrix and $-(P(D))^T$ is an $(n-k)\times k$ matrix (see Definition 2.1.39). This implies that the block matrices of $H_{trunc}(j)$ are $H_0 = \left[\begin{array}{c|c} -(P_0)^T & I_{n-k}\end{array}\right]$, $H_1 = \left[\begin{array}{c|c} -(P_1)^T & 0\end{array}\right]$, …, $H_j = \left[\begin{array}{c|c} -(P_j)^T & 0\end{array}\right]$ with $H_i = 0$ if $i > \mu$, where μ is the memory. The truncated matrix will look like the following:

$$H_{trunc}(j) = \begin{bmatrix} \left[\begin{array}{c|c} -(P_0)^T & I_{n-k}\end{array}\right] & & & \\ \left[\begin{array}{c|c} -(P_1)^T & 0_{n-k}\end{array}\right] & \left[\begin{array}{c|c} -(P_0)^T & I_{n-k}\end{array}\right] & & \\ \vdots & \vdots & \ddots & \\ \left[\begin{array}{c|c} -(P_j)^T & 0_{n-k}\end{array}\right] & \left[\begin{array}{c|c} -(P_{j-1})^T & 0_{n-k}\end{array}\right] & \cdots & \left[\begin{array}{c|c} -(P_0)^T & I_{n-k}\end{array}\right] \end{bmatrix}$$

$P_i = 0$ when $i > \mu$. We assume that $j \geq \mu$. Next, we consider the columns at position $(k+1)$ to n. The first rows of the selected submatrix contain the identity matrix. Next, we collect the columns at position $(n+k+1)$ to $2n$. They have the identity matrix in the second block from above. This step is repeated $\mu+1$ times. This results in a set of $(\mu+1)(n-k)$ columns which contain all the canonical vectors from position 1 to $(\mu+1)(n-k)$. Since each of the first k columns only contains elements up to row $(\mu+1)(n-k)$, it follows that each of these columns lies in the span of our chosen set of $(\mu+1)(n-k)$ columns. According to Proposition 2.2.15 by Gluesing-Luerssen et al., for all $j \geq \mu$, $d^j_H(\mathcal{C}) \leq (\mu+1)(n-k)+1$. This implies that for all $j \geq 0$, $d^j_{SR}(\mathcal{C}) \leq (\mu+1)(n-k)+1$ because the sum rank is upper bounded by

the Hamming distance. It also holds that $d_{SR}(\mathcal{C}) = \lim_{j\to\infty} d^j_{SR}(\mathcal{C})$. With that it follows $d_{SR}(\mathcal{C}) \leq (\mu+1)(n-k)+1$ and the statement is proven. □

Lemma 2.2.17. [MBK16, Lemma 1]
If $d^j_{SR}(\mathcal{C}) = (n-k)(j+1)+1$, then $d^i_{SR}(\mathcal{C}) = (n-k)(i+1)+1$ for all $i \leq j$.

Proposition 2.2.18. [AMPN20, p. 7] Let $\mathcal{C}$ be a systematic rank metric convolutional code. Then, the j-th column sum rank distance bound (see Proposition 2.2.13) can be reached for at most $j = \mu$, where μ is the memory.

Proof. According to Proposition 2.2.16, in case of a systematic encoder, the free sum rank distance is smaller or equal than $(n-k)(\mu+1)+1$, where μ is the memory of the code. Since the j-th column sum rank distance is bounded by the free distance, we have:

$$d^j_{SR}(\mathcal{C}) \leq d_{SR}(\mathcal{C}) \leq (n-k)(\mu+1)+1, \ \forall j \geq 0 \tag{2.2}$$

Now, Proposition 2.2.13 states the mentioned upper bound of the j-th column sum rank distance:

$$d^j_{SR}(\mathcal{C}) \leq (n-k)(j+1)+1, \ \forall j \geq 0.$$

Since (2.2) holds, the j-th column sum rank distance bound can be achieved for at most $j = \mu$. □

Definition 2.2.19. (Memory Maximum Sum Rank (m-MSR) Convolutional Code) [AMPN20, p. 8] A rank metric convolutional code $\mathcal{C}$ is called *m-MSR* if the j-th column sum rank distance bound (see Proposition 2.2.13) is achieved for the memory μ ($j = \mu$):

$$d^\mu_{SR}(\mathcal{C}) = (\mu+1)(n-k)+1$$

Remark 2.2.20. [MBK16, p. 3043] Since $d^j_{SR}(\mathcal{C}) \leq d^j_H(\mathcal{C})$, m-MSR codes automatically maximise the column Hamming distance and can be seen as a special case of m-MDS codes.

Remark 2.2.21. If a rank metric convolutional code is non-systematic and reaches the j-th column sum rank distance bound for the best possible $j = L = \left\lfloor \frac{\delta}{k} \right\rfloor + \left\lfloor \frac{\delta}{n-k} \right\rfloor$, then this code is called L-MSR.

Definition 2.2.22. (j-th Column Rank Distance (j-th CRD)) [AN21, p. 8] Let $\mathcal{C}$ be a convolutional code. Then the *j-th column rank distance* of $\mathcal{C}$, $d^j_{CR}(\mathcal{C})$, is defined as:

$$\begin{aligned} d^j_{CR}(\mathcal{C}) &= \min\left\{ \operatorname{rank}(\phi(v_0)|\cdots|\phi(v_j)) \mid v(D) = \sum_{i=0}^{j} v_i D^i \in \mathcal{C} \text{ and } v_0 \neq 0 \right\} \\ &= \min\left\{ \operatorname{rank}(\phi(v_{[0,j]})) \mid H_{trunc}(j) v^T_{[0,j]} = 0 \text{ and } v_0 \neq 0 \right\} \\ &= \min\left\{ \operatorname{rank}(\phi(v_{[0,j]})) \mid v_{[0,j]} = u_{[0,j]} G_{trunc}(j) \text{ and } u_0 \neq 0 \right\} \end{aligned}$$

The third equality holds if G_0 has full row rank. We assume here that this is the case.

We now define $\mathcal{C}^c_j$. It is the j-th column block code of $\mathcal{C}$:

$$\begin{aligned} \mathcal{C}^c_j &= \left\{ (u_0, u_1, \cdots, u_j) G_{trunc}(j) \mid u_0, u_1, \cdots, u_j \in \mathbb{F}^k_{q^m}, u_0 \neq 0 \right\} \\ &= \left\{ (v_0, v_1, \cdots, v_j) \mid \sum_{h\in\mathbb{N}} v_h D^h \in \mathcal{C}, v_0 \neq 0 \right\} \subseteq \mathbb{F}^{(j+1)n}_{q^m} \end{aligned}$$

With that it holds:

$$d_{CR}^{j}(\mathcal{C}) = d_R(\mathcal{C}_j^c)$$

Note that the block code $\mathcal{C}_j^c$ is *non-linear* (over $\mathbb{F}_{q^m}$) as the zero vector is not in $\mathcal{C}_j^c$.

Theorem 2.2.23. [AN21, p. 8] Clearly, $d_{CR}^{j}(\mathcal{C}) \leq d_{SR}^{j}(\mathcal{C})$, for any $j \geq 0$. Therefore, we have the following Singleton bound for the j-th column rank distance:

$$d_{CR}^{j}(\mathcal{C}) \leq (n-k)\left(\left\lfloor \frac{\delta}{k} \right\rfloor + 1\right) + \delta + 1$$

Theorem 2.2.24. [AN21, Corollary 2] Let $\mathcal{C}$ be an (n, k, δ) convolutional code. Then, for any $j \geq 0$,

$$d_{CR}^{j}(\mathcal{C}) \leq (j+1)(n-k) + 1. \tag{2.3}$$

Lemma 2.2.25. [AN21, Lemma 5]
If $d_{CR}^{j}(\mathcal{C}) = (j+1)(n-k)+1$, then $d_{CR}^{i}(\mathcal{C}) = (i+1)(n-k)+1$ for $i \leq j$.

Lemma 2.2.26. [AN21, p. 12] There exists an integer L for which the bound (2.3) could be attained for all $j \leq L$ and it is a strict upper bound for $j > L$:

$$L = \left\lfloor \frac{\delta}{k} \right\rfloor + \left\lfloor \frac{\delta}{n-k} \right\rfloor \tag{2.4}$$

Definition 2.2.27. (Maximum Rank Profile (MRP) codes) [AN21, p. 12] An (n, k, δ) convolutional code $\mathcal{C}$ where every $d_{CR}^{j}(\mathcal{C})$ is maximal, i.e., $d_{CR}^{j}(\mathcal{C}) = (n-k)(j+1)+1$ for each $j \leq L$, is called a *maximal rank profile* (MRP) code with L as in (2.4). In particular, $\mathcal{C}$ is an MRP convolutional code if $d_{CR}^{L}(\mathcal{C}) = d_R(\mathcal{C}_L^c) = (L+1)(n-k)+1$.

There exist some active distances which are defined for the j-th order, i.e. sequences of length j. Additionally, they require some conditions on the past states of the encoder. Wachter-Zeh et al. [WZSS15, p. 3202] define the following three active distances:

- active row rank distance $d_{SR}^{(r)}(j, \mathcal{C})$
- active column rank distance $d_{SR}^{(c)}(j, \mathcal{C})$
- active reverse column rank distance $d_{SR}^{(rc)}(j, \mathcal{C})$

Any convolutional code can be described by a minimal code trellis, which has a certain number of states. The input/output blocks are associated with the edges of the trellis [WZSS15]. For each active distance, we draw a minimal code trellis to visualise the corresponding codewords.

Definition 2.2.28. (Active Row Rank Distance) [WZSS15, Definition 6]
Let $\mathcal{C}_j^{(r)}$ denote the set of all codewords in a convolutional code $\mathcal{C}$ which diverge from the zero state at depth zero and return to the zero state at depth j for the *first* time. Figure 2.4 gives some examples of such codewords. The *active row rank distance* of order j of the convolutional code $\mathcal{C}$ is defined as:

$$d_{SR}^{(r)}(j, \mathcal{C}) = \min_{v(D) \in \mathcal{C}_j^{(r)}} \left\{ \sum_{i=0}^{j} \operatorname{rank}(\phi(v_i)) \right\}, \qquad \forall j \geq 1$$

Remark 2.2.29. W.l.o.g., we assume that we start at depth zero as we only consider time-invariant convolutional codes.

Remark 2.2.30. [WZSS15, p. 3202] If the encoder is non-catastrophic, we have:

$$d_{SR}(\mathcal{C}) = \min_j \left\{ d_{SR}^{(r)}(j, \mathcal{C}) \right\}$$

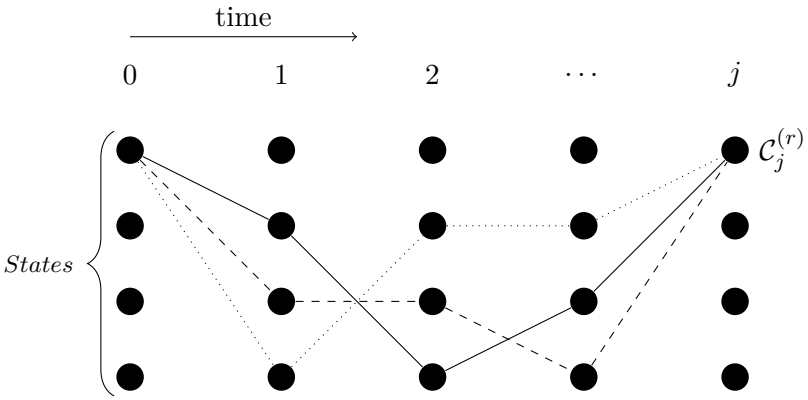


FIGURE 2.4: The minimal trellis graph shows some examples for codewords in $\mathcal{C}_j^{(r)}$. The codewords in this set start from the zero state and re-enter the zero state at depth j for the first time. [WZSS15, Figure 1]

Definition 2.2.31. (Active Column Rank Distance) [WZSS15, p. 3203]
Let $\mathcal{C}_j^{(c)}$ denote the set of all codewords in a convolutional code $\mathcal{C}$ which start from the zero state at depth zero and do not return to the zero state between 1 and $j-1$. At depth j, the zero state is allowed as well as all other states. Figure 2.5 provides some examples of such codewords. The *active column rank distance* of order j of the convolutional code $\mathcal{C}$ is defined as:

$$d_{SR}^{(c)}(j, \mathcal{C}) = \min_{v(D) \in \mathcal{C}_j^{(c)}} \left\{ \sum_{i=0}^{j} \operatorname{rank}(\phi(v_i)) \right\}, \qquad \forall j \geq 1$$

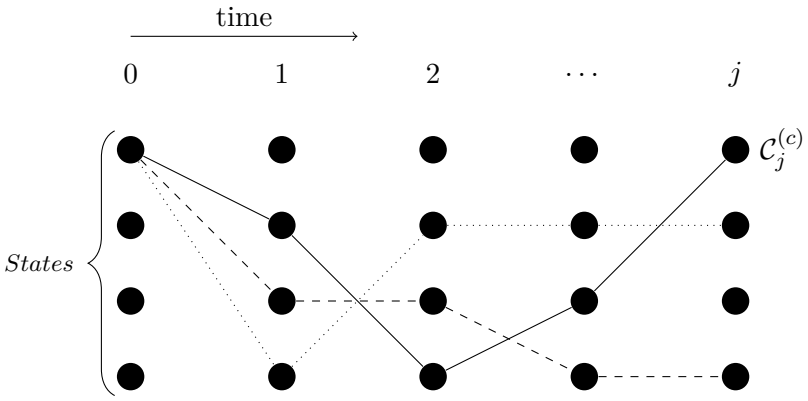


FIGURE 2.5: The minimal trellis graph shows some examples for codewords in $\mathcal{C}_j^{(c)}$. The codewords in this set start from the zero state and do not re-enter the zero state before j. [WZSS15, Figure 2]

Definition 2.2.32. (Active Reverse Column Rank Distance) [WZSS15, p. 3203]
Let $\mathcal{C}_j^{(rc)}$ denote the set of all codewords in a convolutional code $\mathcal{C}$ which start from any state at depth zero and end in the zero state at depth j. In between is no zero state allowed.

Figure 2.6 gives some examples of such codewords. The *active reverse column rank distance* of order j of the convolutional code $\mathcal{C}$ is defined as:

$$d_{SR}^{(rc)}(j,\mathcal{C}) = \min_{v(D)\in\mathcal{C}_j^{(rc)}} \left\{ \sum_{i=0}^{j} \operatorname{rank}(\phi(v_i)) \right\}, \qquad \forall j \geq 1$$

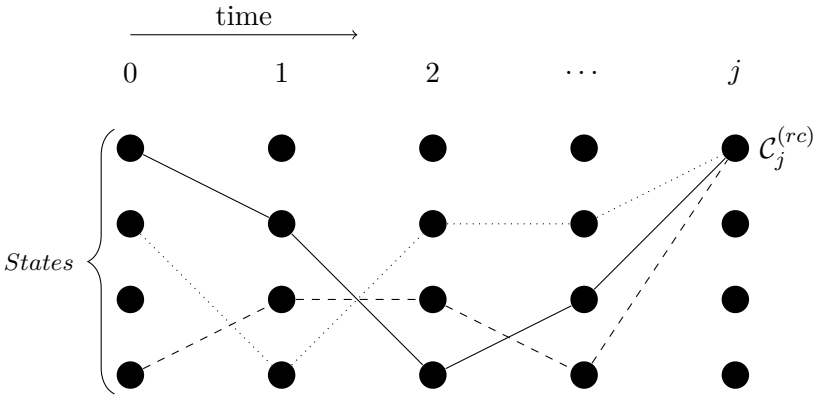


FIGURE 2.6: The minimal trellis graph shows some examples for codewords in $\mathcal{C}_j^{(rc)}$. The codewords in this set start from any state and return to the zero state at time j. Between depth 0 and j is no zero state allowed. [WZSS15, Figure 3]

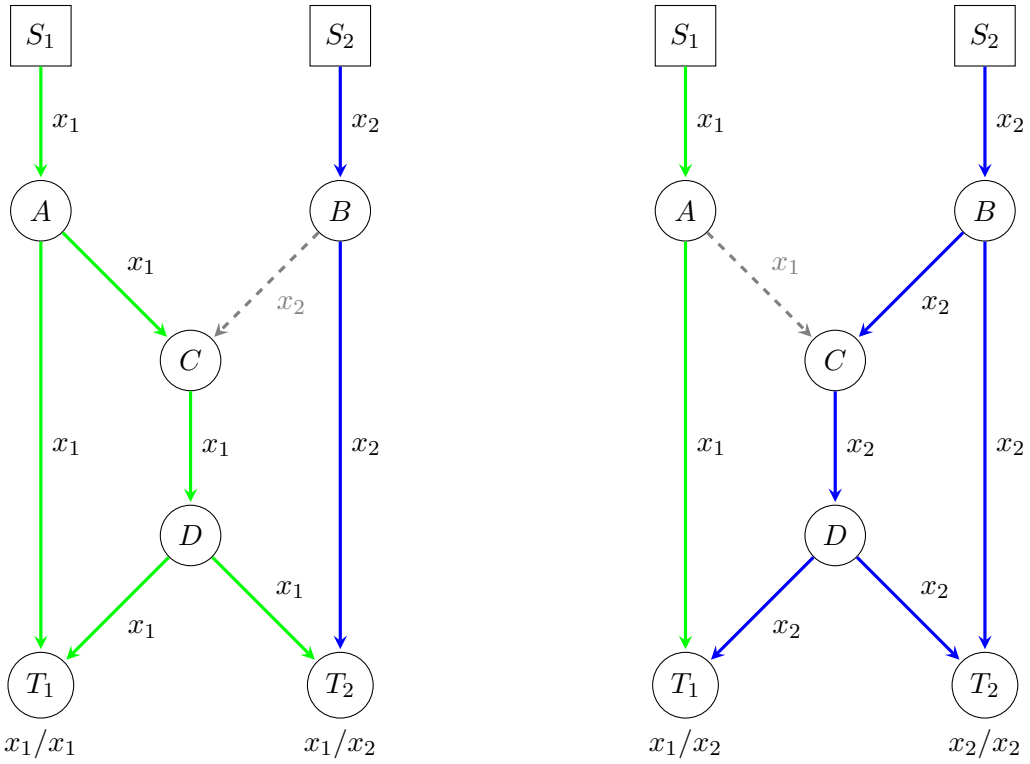
Chapter 3

Introduction to Random Linear Network Coding

In this chapter, we will outline the basic approach of network coding and how it can be used to improve the transmissions in the network. Some examples will exactly show how the procedure works. This chapter is based on [FS07, HL08, MS12].

3.1 Overview

We start this section with an example to illustrate how linear network coding can improve network throughput.



(A) Routing of x_1 at C.

(B) Routing of x_2 at C.

FIGURE 3.1: The butterfly network. [FS07, Figure 1.2(a),(b)]

First, the currently used broadcasting network protocol will be shown. Let us assume that S_1 and S_2 are two transmitters of information. They will send information x_1 and information x_2 to the receivers T_1 and T_2. In the field of communication, the transmitters are called *source nodes* and the receivers *sink nodes*. We consider the network in Figure 3.1. Each node

receives packets and has to forward them to guarantee the information flow. Unfortunately, each node can only forward one piece of information at once. In this case, node C has to determine whether it will forward x_1 or x_2 at timestamp 1. One possibility is to send x_1 at timestamp 1 and x_2 at the timestamp 2. However, we aim for the best possible solution in one timestamp. Figure 3.1 represents one of the simplest network models, the butterfly network model. The arrows represent the links and they are labelled with the transmitted message. The path C to D creates a bottleneck as the network has to decide at node C which packet to route first [FS07].
Figure 3.1a shows what happens if node C decides to transmit packet x_1. T_2 is content since it receives both packets in one timestamp. However, the same packet reaches T_1 twice and hence T_1 lacks the second information at timestamp 1. In the inverted situation (see Figure 3.1b), node C passes x_2 which results in T_1 obtaining both packets but T_2 only the second one. We observe that no matter which routing variant the network chooses, in the end, only one receiver accepts both packets in one timestamp.

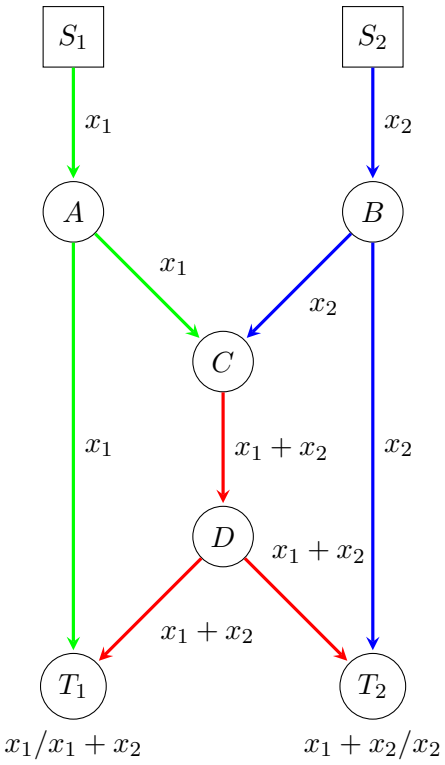


FIGURE 3.2: The butterfly network with linear network coding. [FS07, Figure 1.2(c)]

It is physically impossible to send two packets simultaneously through one connection. The idea of linear network coding comprises sending a linear combination of both packets through the link. For example, the node C adds both incoming packets and sends the sum through the network (see Figure 3.2). T_1 receives x_1 through the direct link from A and $x_1 + x_2$ through D. The calculations are tracked in the header of the corresponding packet. This means the sink node T_1 knows that the received packet p from D, $p = x_1 + x_2$, consists of x_1 and x_2. With this information, the receiver can solve the system of linear equations by the following elementary calculation:

$$(x_1 + x_2) - x_1 = x_2$$

T_1 can then reconstruct x_2 and therefore receives both packets in one timestamp. Not only

T_1 receives the complete information by reconstructing the second packet but also T_2 which, unlike T_1, receives x_2 and $(x_1 + x_2)$ and hence reconstructs x_1 instead of x_2.
In other words, both sink nodes can receive x_1 and x_2 in one timestamp which was not possible with the standard routing mechanism in Figure 3.1. We assume that solving the system of linear equations does not require additional time [FS07].

Besides the improvements of throughput, linear network coding also increase network security. If sensitive data is sent, no third party should be able to intercept the message. Usually, the data is already sent encrypted by the transmitter. Additionally, linear network coding adds a second layer of security. Consider the link C to D in Figure 3.2. If an eavesdropper could only listen to this channel, it would catch the message $x_1 + x_2$ but nothing else. That is too little information to reconstruct x_1 or x_2. We can compare this to a previously discussed broadcasting network like in Figure 3.1a where x_1 is sent via the link between C and D. If an eavesdropper listens to this channel, they gain access to x_1 which may contain sensitive information [FS07, HL08].

3.2 Basic Concepts of Random Linear Network Coding

We will now generalise the butterfly network and set these things in an algebraic context. In this thesis, we analyse linear network coding over finite fields $\mathbb{F}_q$. This gives a useful structure to multiply, divide, add and subtract. This concept depends on [MS12].
First, for simplicity, we assume that the source node sends scalar values through the network as illustrated, for example, in Figure 3.2 with $x_1, x_2 \in \mathbb{F}_q$. We also assume that the number of input packets n is equal to the number of output values n at the sink node. The packets go from the source node to an arbitrary number of internal nodes. In the end, they reach the sink node. Each internal node takes the input packets and creates a linear combination of them. This linear combination is created randomly at each internal node when using a *random* linear network coding process. There also exists the possibility to use *non-random* linear network coding. In this model, predefined encoding vectors to produce the linear combinations are used. However, a real network changes its structure fluently such that the random approach is more stable than using predefined linear encoding vectors. Each randomly created linear combination will then be forwarded to the next node. In the end, each of the n output values at the sink is a linear combination of the input packets. We can write this linear transformation in a matrix representation. Let $x_1, \ldots, x_n \in \mathbb{F}_q$ be the input packets that will be sent through the network. $A \in \mathbb{F}_q^{n \times n}$ is the n times n channel matrix which represents the linear combinations of the packets through the network. $y_1, \ldots, y_n$ are the n packets at the sink node. That results in the following dependency:

$$(y_1, \ldots, y_n) = (x_1, \ldots, x_n) \cdot A$$

Next, we will show an explicit example how such a network can look like (see Figure 3.3). The links are drawn with arrows labelled with the transmitted message. The source node S sends 3 distinct packets $x_1, x_2, x_3 \in \mathbb{F}_{31}$ through 3 distinct links. Node A, B and C simply forward the message to each connected link. Node D receives two different packets. It decides to combine these two into the linear combination $x_1 + 3x_2$ which is then forwarded. Similarly, node E sends $x_2 + 4x_3$. The last internal nodes F, G and H add both inputs together and

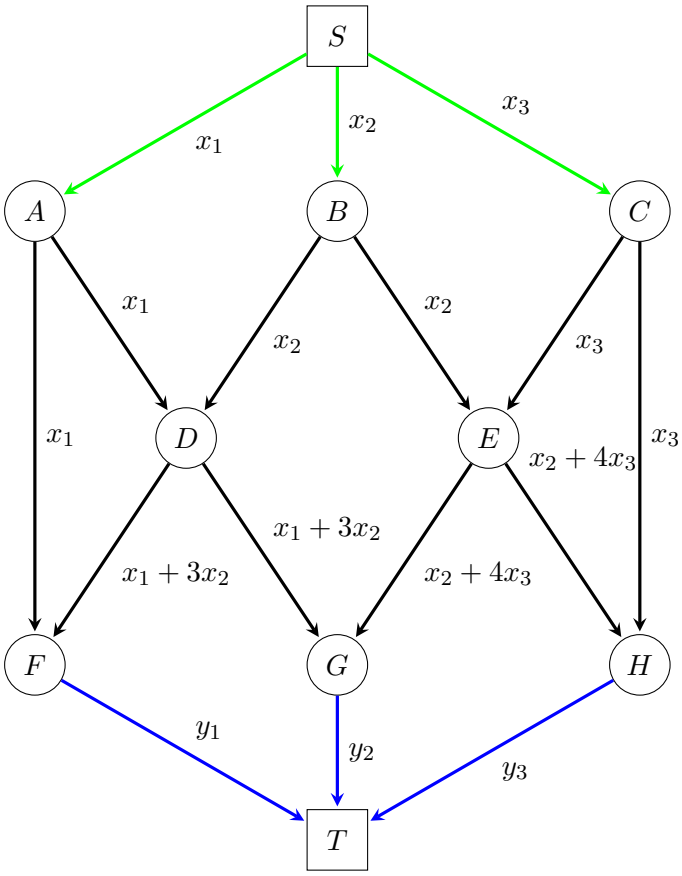


FIGURE 3.3: Linear network coding example.

transmit the sum. Consequently, this results in the following sequence for the sink node T:

$$\begin{aligned} y_1 &= x_1 + (x_1 + 3x_2) = 2x_1 + 3x_2 \\ y_2 &= (x_1 + 3x_2) + (x_2 + 4x_3) = x_1 + 4x_2 + 4x_3 \\ y_3 &= (x_2 + 4x_3) + x_3 = x_2 + 5x_3 \end{aligned} \tag{3.1}$$

The receiver is interested in the original packets x_1, x_2, x_3. For that reason, one has to solve the system of linear equations (3.1). We can translate this problem to the following matrix representation:

$$(y_1, y_2, y_3) = (x_1, x_2, x_3) \cdot \begin{bmatrix} 2 & 1 & 0 \\ 3 & 4 & 1 \\ 0 & 4 & 5 \end{bmatrix}$$

$A = \begin{bmatrix} 2 & 1 & 0 \\ 3 & 4 & 1 \\ 0 & 4 & 5 \end{bmatrix}$ is the channel matrix. In *random* linear network coding, A is usually not known by the source and the sink node at the beginning of the transmission. However, the information about the linear combinations is transmitted by the packet header. This allows the sink node to reconstruct the channel matrix. Therefore, the sink node knows the channel matrix A and the output packets y_1, y_2, y_3. With this knowledge it would like to reconstruct the input packets x_1, x_2, x_3 [MS12].

We can then calculate the determinant of A:

$$\begin{vmatrix} 2 & 1 & 0 \\ 3 & 4 & 1 \\ 0 & 4 & 5 \end{vmatrix} = 40 - 8 - 15 = 17$$

Since the determinant of the channel matrix is non-zero in $\mathbb{F}_{31}$, we know that the linear equation system has a solution. Finally, the receiver can reconstruct x_1, x_2 and x_3.

3.3 Error Handling

According to Silva et al. [SKK08], linear network coding is a useful technic to increase the throughput in the network. Section 3.1 has shown an example of how such an improvement can be reached. Nevertheless, linear network coding is very error-prone. Just one corrupted packet or erasure can affect all received packets. Recap Figure 3.3 for a concrete example. We now assume that the link between H and T is deleted. This means that the new vector y_3 is equal to 0. The new channel matrix A is

$$\begin{bmatrix} 2 & 1 & 0 \\ 3 & 4 & 0 \\ 0 & 4 & 0 \end{bmatrix}.$$

Clearly, the matrix A has not full rank anymore. So, the linear system is unsolvable. Therefore, error control in linear network coding is essential.

The general idea of error handling is adding more redundant information to the packets. One can do that by using coding theory. We take the original packets of the source node and encode those packets. Precisely, the source packets are not directly sent through the network. Instead, the new encoded packets are transmitted. The receiver gets the encoded packets that may be altered by errors or erasures through the transmission. Due to coding theory, the receiver can then reconstruct the original packets and the source packets with a higher probability. This leads to better error handling.

The linear network coding process generates linear combinations of the incoming packets in each internal node. Now, we define the source packets as scalars over $\mathbb{F}_{q^m}$ instead of $\mathbb{F}_q$. These scalars are isomorphic to vectors in $\mathbb{F}_q^m$. With this new definition, it follows that the network creates linear combinations of vectors over $\mathbb{F}_q^m$. Each linear combination lies in the span of the original vectors. It follows that the span of the source vectors will not change during the transmission over the network, except errors or erasures occur. This invariant will be used to find appropriate codes and equips the network with error handling capabilities.

There are different coding techniques applied in this context to create a code for the packets over $\mathbb{F}_{q^m}$. The main code concepts for random linear network coding are:

- Constant Dimension Codes
- Rank Metric Codes

In this work, we focus on the rank metric approach described in Section 2.2. For the sake of completeness, we give a short recap in this chapter. We also give an overview of constant dimension codes, which are, however, not explored further in this thesis.

3.3.1 Constant Dimension Codes

The codewords of a constant dimension code are vector *spaces* built by a basis of vectors. These vectors are the source packets that are sent by the source node. Since the span of the vectors remains unchanged, the subspace at the source and sink node should be the same. A constant dimension code additionally requires that each codeword must have the same predefined dimension. The metric of a constant dimension code is defined by

$$d(A, B) = \dim(A + B) - \dim(A \cap B), \quad \text{where } A, B \text{ are subspaces.}$$

$A + B$ means the sum of two subspaces. This sum is defined as the smallest possible subspace which contains both A and B [KK08].

3.3.2 Rank Metric Codes

Rank metric codes (described in Section 2.2) also profit from the fact that the subspace of the input packets does not change during the transmission to the destination node. However, we do not use subspaces as codewords but matrices instead. We write all n encoded source packets (over $\mathbb{F}_{q^m}$) of one timeslot in a vector $\mathbb{F}_{q^m}^n$. Then, the isomorphism $\mathbb{F}_{q^m}^n \to \mathbb{F}_q^{m \times n}$ is used to obtain the matrices which are the codewords [SKK08].
Since we work with matrices, the following rank metric distance is defined:

$$d_R(A, B) = \operatorname{rank}(A - B), \quad \text{where } A, B \in \mathbb{F}_q^{m \times n}$$

Also, rank metric convolutional codes can be used.

Chapter 4

Network Models

4.1 Introduction

Chapter 3 has shown the general idea of random linear network coding and has presented some illustrating examples. The current chapter will build on that, will set the fundamental framework and will define the network models we will use during this thesis. With this set-up, we will be able to analyse and compare different constructions. This chapter will combine network models with different frameworks. [Mah15, MBK16, AMPN20, AN21]

The basic concept remains unchanged: We would like to send a message through the network from a source node to a sink node. It is assumed that we have one source node and one sink node although the source node can send multiple packets into the network at the same time. Our models require that at time t, n packets can be sent. Also, the sink node can receive n packets at the same time. In the network, we use rank metric codes to improve the error capability (see Section 2.2 for the theory of rank metric codes).

4.2 Framework

This section will show the general pattern of network coding. It will provide some definitions and will describe the procedure of how a message is sent from the transmitter to the receiver via the network.

If a message is sent or a video is streamed through the network, then the source node takes the first k packets and combines them into to a source packet vector.

Definition 4.2.1. (Source Packet) [Mah15] The source node splits the message into separate packets. Such a packet is called *source packet* p and is an element of $\mathbb{F}_{q^m}$.

Definition 4.2.2. (Source Packet Vector) [Mah15] A *source packet vector* at time t is a vector $s_t \in \mathbb{F}_{q^m}^k$. It consists of k source packets. $s_{[i,j]}$ defines the concatenated vector $(s_i, s_{i+1}, \ldots, s_j) \in \mathbb{F}_{q^m}^{k(j-i+1)}$ for $i \leq j$.

In network coding, we do not send these source packets directly through the network. Instead, we encode a source packet vector into a codeword packet vector x of length n.

Definition 4.2.3. (Codeword Packet Vector) [Mah15] A *codeword packet vector* (or channel packet sequence) is the encoded source packet vector $x_t \in \mathbb{F}_{q^m}^n$ at time t. It consists of n packets in $\mathbb{F}_{q^m}$. Such a single packet is called the *codeword packet* (or channel packet). x_t can be calculated by the source packet vector s_t or by s_t and all source packet vectors before: $s_{[0,t]}$. This depends on the chosen code type (block code versus convolutional code).

We define g as the encoding map of the code. As a result, x_t is generated by:

$$x_t = g(s_{[0,t]}).$$

Again, we can write $x_{[i,j]} = (x_i, x_{i+1}, \ldots, x_j) \in \mathbb{F}_{q^m}^{n(j-i+1)}$, $\forall i \leq j$.

These n codeword packets of the codeword packet vector are sent through the network at time t. The receiver gets the packet vector y_t at time t which contains n packets. These packets are random linear combinations of the sent codeword packets since the random linear network coding creates them during the transmission. Section 3.2 will give an example of a network and how it creates these linear combinations out of the codeword packets.

The intermediate nodes of the network have different possibilities on how they forward the message and which errors can occur. Each network model has its specific settings. Therefore, we can distinguish different network model properties:

- **Time Delay:** If the network model is time delay free, then y_t only consists of linear combinations of x_t. So, y_t only depends on packets of the same time. In that case, we can write $y_t = x_t A_t$, where $A_t \in \mathbb{F}_q^{n \times n}$ is the channel matrix at time t. If the model allows time delay, y_t depends on $x_{[0,t]}$. This means the received message at time t depends on codeword packets at time t and before. We write $y_{[0,t]} = x_{[0,t]} A_{[0,t]}$, where $A_{[0,t]} \in \mathbb{F}_q^{n(t+1) \times n(t+1)}$ is a block upper triangular matrix.
- **Erasures:** It could occur that some links in the network do not work during the transmission. In that case, some linear combinations will change. These erasures are also considered by the channel matrix A.
- **Errors:** In some network models, we also consider additive errors. This means that in addition to the linear combinations, the network can produce unintended additive errors at some links. Unfortunately, these errors will be propagated through the whole network.

The channel matrix A represents the creation of the linear combination in a matrix. $x \cdot A$ gives the received packets at the sink node (see Section 3.2).

Definition 4.2.4. (Received Packet Vector) [Mah15] $y_t \in \mathbb{F}_{q^m}^n$ described above is the received vector of packets at time t. Also, we can write $y_{[i,j]} = (y_i, y_{i+1}, \ldots, y_j) \in \mathbb{F}_{q^m}^{n(j-i+1)}, \forall i \leq j$. This vector is also called the output channel packet sequence.

At the end of the transmission, the destination node tries to decode the original source vector. As we use coding techniques, the sink node can, in some cases, reconstruct the source vector, despite some errors or erasures happened since the codeword packet vector contains redundant information.

4.3 Models

The network model is a central part of the analysis of different encoding, decoding and metric strategies. We will first present two network models and their properties.

4.3.1 Rank Deficient Sliding Window Network without Time Delay

The rank deficient sliding window network model is also known as the *streaming problem*. We will explain this network model in three steps: Encoder, Transmission and Decoder.

Encoder

The encoder creates a codeword x_t from the source packets. As explained above, we use the encoding map g that depends on the code type. For example, in case of a block code, x_t depends on the source packet s_t. In the same way, we can choose a convolutional code. Since convolutional codes link all codewords, x_t depends in that case on $s_{[t-\mu,t]}$, where μ is the memory of the convolutional code. Usually, we use a convolutional code to encode the source packets. The reasons why convolutional codes are preferred over block codes are as follows:

- Multishot Network: The transmitted message is split into several packets and shots. In each shot, different packets are encoded. This implies that we have to encode multiple codewords. This can profit from an inter-shot connection via a convolutional code.
- Free distance: The upper bound of the free distance of convolutional codes can be constructed such that it is bigger than a comparable block code. A higher free distance implies that we can correct more erasures and it provides better decoding properties.
- Moreover, one can see that convolutional codes have a better performance in the decoding than block codes. This is due to the capability of considering particular parts (windows) of the sequence of different sizes through the decoding process and slide along this sequence to choose the window to start decoding. Such capability is called the sliding window property and it allows the decoder to adapt the process. [LPR21]

Transmission

The rank deficient sliding window network transmits the codeword with **zero delays** through the links. The destination node sequentially receives $y_t = x_t A_t$, where $A_t \in \mathbb{F}_q^{n\times n}$ is the channel matrix at time t and all the received packets at time t consist of codeword packets sent at the same time t. We can also look at a transmission over a window $[t, t+W-1]$ of W timestamps, called shots. This communication is defined by a block diagonal channel matrix $A_{[t,t+W-1]} = \mathrm{diag}(A_t, \ldots, A_{t+W-1})$:

$$y_{[t,t+W-1]} = x_{[t,t+W-1]} A_{[t,t+W-1]},$$

where $A_{[t,t+W-1]} \in \mathbb{F}_q^{Wn\times Wn}$.

If the encoder uses a convolutional code and $G_{trunc}(W-1)$ is the corresponding truncated sliding generator matrix, we can reformulate the formula above:

$$y_{[t,t+W-1]} = s_{[t,t+W-1]} \cdot G_{trunc}(W-1) \cdot A_{[t,t+W-1]},$$

where $A_{[t,t+W-1]} \in \mathbb{F}_q^{Wn\times Wn}$ and $G_{trunc}(W-1) \in \mathbb{F}_{q^m}^{kW\times nW}$.

Definition 4.3.1. (Diagonal Truncated Channel Matrix) [MBK16] A *diagonal truncated channel matrix* is a diagonal matrix $A_{[0,t]}$ which represents the linear combinations of the packets through the network from timestamp 0 to t:

$$A_{[0,t]} = \mathrm{diag}(A_0, A_1, \ldots, A_t) = \begin{bmatrix} A_0 & & & \\ & A_1 & & \\ & & \ddots & \\ & & & A_t \end{bmatrix} \in \mathbb{F}_q^{n(t+1)\times n(t+1)}$$

where A_i are square (not necessarily non-singular) matrices of order n. Hence, the received packets are $y_{[0,t]} = x_{[0,t]} A_{[0,t]}$ where $y_{[0,t]} = (y_0, y_1, \cdots, y_t)$, $x_{[0,t]} = (x_0, x_1, \cdots, x_t) \in \mathbb{F}_{q^m}^{n(t+1)}$.

Let $p_t = \text{rank}(A_t)$. The total rank of the channel matrix in the window is equal to the sum of the individual ranks of the channel matrices:

$$\sum_{i=t}^{t+W-1} p_i = \text{rank}(A_{[t,t+W-1]}).$$

In a situation of perfect communication, it holds $p_t = n$ but erasures or bad network connections might result in a rank-deficient channel matrix.

Definition 4.3.2. (Rank Deficient Sliding Window Network $\mathcal{CH}(S,W)$) [Mah15, Defintion 4] Consider a network where for all $t \geq 0$, the sink node receives $y_t = x_t A_t$ with $\text{rank}(A_t) = p_t \leq n$. The *rank deficient sliding window network* $\mathcal{CH}(S,W)$ has the property that in any sliding window of length W the rank of the block diagonal channel matrix $A_{[t,t+W-1]}$ decreases by at most S, i.e.,

$$\sum_{i=t}^{t+W-1} p_i \geq nW - S.$$

Figure 4.1 shows an example of such a rank deficient sliding window network:

4	5	1	3	5	2	2	5	5	5	3	...

FIGURE 4.1: Example of a rank deficient sliding window network with window size 3 and maximum rank decrease of 6: $\mathcal{CH}(6,3)$. n, the size of one channel matrix, is equal to 5. The rank of each channel matrix is labelled inside the box. The rank of the channel matrix in any window of $W = 3$ is greater than or equal to $9 = 3 \cdot 5 - 6$. [Mah15, Concept of Figure 3.1]

Remark 4.3.3. This network model does not support additive errors. We only consider erasures that are included in the channel matrix A.

Decoder

Let T be the maximum delay before the destination node should be able to decode the source packet. This means a packet received at time t must be recovered by time $t+T$. If the source packet is not recovered correctly, it is declared lost. In linear network coding, we can reduce this problem to inverting the channel matrix. We assume that $T = W - 1$.
The sum rank distance is especially designed for such a network type. A famous result of the sum rank distance is the following theorem:

Theorem 4.3.4. [MBK16, Theorem 2] Let $\mathcal{C}[n,k,\mu]$ be a rank metric convolutional code used over the window $[0, W-1]$. For $0 \leq t \leq W-1$, let $A_t \in \mathbb{F}_q^{n\times n}$ be matrices with $\text{rank}(A_t) = p_t$ and $A_{[0,W-1]} = \text{diag}(A_0, \ldots, A_{W-1})$ be the channel matrix. The following statements are true:

1) If $d_{SR}^{W-1}(\mathcal{C}) > nW - \sum_{t=0}^{W-1} p_t$, then the source packet vector s_0 is always recoverable by time $W-1$.

2) If $d_{SR}^{W-1}(\mathcal{C}) \leq nW - \sum_{t=0}^{W-1} p_t$, then there exists at least one channel packet sequence $x'_{[0,W-1]}$ and channel matrix $A'_{[0,W-1]}$ for which s_0 is not recoverable by time $W-1$.

Proof. Due to code linearity (convolutional codes are linear), we only need to show that all output channel packet sequences are distinguishable from the all-zero sequence. We prove this by contradiction.

First, we prove statement 1). We assume that the inequality holds and the output packet is not distinguishable from the zero vector. Take a source packet sequence $s_{[0,W-1]} = (s_0, \dots, s_{W-1}) \in \mathbb{F}_{q^m}^{kW}$, where $s_0 \neq 0$. Suppose that this sequence generates the channel packet sequence $x_{[0,W-1]}$, for which $x_{[0,W-1]}A_{[0,W-1]} = 0$. This implies that $\text{rank}(\phi(x_t)) \leq n - \text{rank}(A_t)$ for $0 \leq t \leq W-1$. This follows since x_t has to lie in the kernel of A_t (otherwise $x_{[0,W-1]}A_{[0,W-1]} \neq 0$) and the kernel has dimension $n - \text{rank}(A_t)$. By summing up the $\text{rank}(\phi(x_t))$ and use the given inequality, we get the following calculation:

$$\sum_{t=0}^{W-1} \text{rank}(\phi(x_t)) \leq nW - \sum_{t=0}^{W-1} p_t < d_{SR}^{W-1}(\mathcal{C})$$

This is a contradiction to the minimum sum rank distance since the weight of a codeword x cannot be smaller than the minimum distance.
For the statement 2), we use a direct proof. First, we assume that we have $d_{SR}^{W-1}(\mathcal{C}) = nW - \sum_{t=0}^{W-1} p_t$. Let $s_{[0,W-1]} = (s_0, \dots, s_{W-1})$, with $s_0 \neq 0$ be a source packet sequence that maps to $x_{[0,W-1]}$, which achieves the minimum distance, i.e., $\sum_{t=0}^{W-1} \text{rank}(\phi(x_t)) = d_{SR}^{W-1}(\mathcal{C})$. For $0 \leq t \leq W-1$, let $r_t = n - \text{rank}(\phi(x_t))$. There exist matrices $A_t \in \mathbb{F}_q^{n \times n}$ with $\text{rank}(A_t) = r_t$ such that each $x_t A_t = 0$. Let $A_{[0,W-1]} = \text{diag}(A_0, \dots, A_{W-1})$ be the channel matrix. Summing all of the p_t reveals:

$$\text{rank}(A_{[0,W-1]}) = \sum_{t=0}^{W-1} r_t = nW - \sum_{t=0}^{W-1} \text{rank}(\phi(x_t)) = nW - d_{SR}^{W-1}(\mathcal{C}).$$

Furthermore, $s_{[0,W-1]}$ is indistinguishable from the all-zero source packet sequence over this channel. So, we have constructed a channel matrix and codeword vector $x_{[0,W-1]}$ for which s_0 is not recoverable. If $d_{SR}^{W-1}(\mathcal{C}) < nW - \sum_{t=0}^{W-1} p_t$, it implies that $\text{rank}(A_{[0,W-1]}) < nW - d_{SR}^{W-1}(\mathcal{C})$. We can create an appropriate A by using the procedure above but decrease the degree of some A_t. □

A strength of this theorem is the "if and only if" statement. It not only guarantees the sufficient condition to recover s_0 but also states the necessary condition. This implies that the sum rank metric is the most accurate measure to determine if a source packet is recoverable in any case or not.

Mahmood et al. [MBK16] describe in Theorem 4.3.4 a precise decoding property for a rank deficient sliding window network without time delay. The greater the j-th column sum rank distance the better the decoding property. For that reason, we are interested in codes with a maximum j-th column sum rank distance. The problem is reduced to systematic generator matrices. Proposition 2.2.18 tells us that in case of a systematic encoder, the j-th column sum rank distance bound can be reached for at most $j = \mu$, where μ is the memory of the code. This kind of codes are called *m-MSR*, as already defined in Definition 2.2.19. This is the reason why we are interested in the construction of such m-MSR codes if we work over the rank deficient sliding window network without delay. Chapter 5 will present an m-MSR construction.

4.3.2 Rank Deficient Sliding Window Network with Delay

The second network model which we will examine in the context of this thesis is the rank deficient sliding window network with delay. This network model is a generalisation of the rank deficient sliding window network without delay. It allows that the received packet vector y_t consists of linear combinations of x_t and the entire window before.

Encoder

The codewords will be built the same way as for the rank deficient sliding window network without delay. Again, we use block codes or more frequently convolutional codes.

Transmission

Since this network model allows time delay, it is possible that packets with different timestamps can be combined during the transmission. This means that y_t does not only depend on codeword packets of the same timestamp t.
More precisely,

$$y_{[0,t]} = x_{[0,t]} A_{[0,t]},$$

where y_t are the last n coordinates of $y_{[0,t]}$.
$A_{[0,t]}$ is a block upper triangular matrix for that reason:

Definition 4.3.5. (Truncated Channel Matrix) [AN21, Subsection 2.1.2] A truncated channel matrix is an upper triangular matrix $A_{[0,t]}$ which represents the linear combinations of the packets through the network from timestamp 0 to t:

$$A_{[0,t]} = \begin{bmatrix} A_{00} & A_{01} & \dots & A_{0t} \\ & A_{11} & \dots & A_{1t} \\ & & \ddots & \vdots \\ & & & A_{tt} \end{bmatrix} \in \mathbb{F}_q^{n(t+1)\times n(t+1)},$$

where A_{il}, $i, l \in \{0, \dots, t\}$, are square (not necessarily non-singular) matrices of order n. Hence, the received packets are $y_{[0,t]} = x_{[0,t]} A_{[0,t]}$ where $y_{[0,t]} = (y_0, y_1, \cdots, y_t)$, $x_{[0,t]} = (x_0, x_1, \cdots, x_t) \in \mathbb{F}_{q^m}^{n(t+1)}$.

Since $A_{[0,t]}$ is in this network model not block diagonal anymore, we cannot use the equality from above:

$$\operatorname{rank}(A_{[0,t]}) \neq \sum_{i=0}^{t} p_i$$

Remark 4.3.6. This network model also does not support additive errors. We only consider erasures that are included in the channel matrix A.

Decoder

In the section about the rank deficient sliding window network without time delay (diagonal channel matrix A), we have seen that the following holds: The first source packet vector s_0 is recoverable if

$$d_{SR}^{W-1}(\mathcal{C}) > Wn - \operatorname{rank}(A_{[t,t+W-1]}). \tag{4.1}$$

The following example will show that this bound is not good enough for the rank deficient sliding window network with delay with the upper triangular channel matrix A.

Example 4.3.7. [AN21, Example 1] Let $G(D) = G_0 + G_1 D \in \mathbb{F}_{2^6}[D]^{2\times 3}$ be the generator matrix of the convolutional code $\mathcal{C} \subset \mathbb{F}_{2^6}[D]^3$, α a primitive element of $\mathbb{F}_{2^6}$ such that $\alpha^6 = \alpha+1$ and

$$G_0 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & \alpha & \alpha^2 \end{bmatrix}, \qquad G_1 = \begin{bmatrix} 0 & 1 & 0 \\ \alpha^3 & \alpha^4 & \alpha^5 \end{bmatrix}.$$

We can check that $d^1_{SR}(\mathcal{C}) = 2$ and that there exists a $u_{[0,1]} = (1, 0 \,|\, 0, 0)$ such that

$$u_{[0,1]} \cdot G_{trunc}(1) = v_{[0,1]} = (1, 0, 0 \,|\, 0, 1, 0) \in \mathbb{F}_{2^6}^6.$$

By the above Statement (4.1) one can always recover u_0 in delay-free networks as far as the j-th column sum rank distance of $\mathcal{C}$ is larger than the rank deficiency in the window $[0, j]$, i.e., if $n(j+1) - \operatorname{rank}(A_{[0,j]}) < d^j_{SR}(\mathcal{C})$. Thus, in this example with $j = 1$ one can recover u_0 if $6 - \operatorname{rank}(A_{[0,1]}) < 2$ or equivalently, if $\operatorname{rank}(A_{[0,1]}) \geq 5$. However, in the presence of delays in the network, this does not necessarily hold. Take the channel matrix with delays

$$A_{[0,1]} = \left[\begin{array}{ccc|ccc} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right] \in \mathbb{F}_2^{6\times 6}$$

that has rank equal to 5 and yields

$$v_{[0,1]} A_{[0,1]} = 0$$

i.e., $v_{[0,1]}$ is indistinguishable from the zero sequence and therefore u_0 cannot be recovered.

The example shows that we need a better property to decide if we can recover u_0 in the case of a delay transmission, i.e., A is upper triangular. The column rank distance defined in Definition 2.2.22 has a more useful property for time-delay networks. The following theorem shows this statement:

Theorem 4.3.8. [AN21, Theorem 4] Let $\mathcal{C} = \operatorname{im}(G(D)) = \ker(H(D)) \subset \mathbb{F}_{q^m}[D]^n$ be an (n, k, δ) convolutional code with $G(D)$ basic, $v(D) = u(D)G(D) \in \mathcal{C}, u_0 \neq 0$ and $A_{[0,j]}$ be the upper triangular truncated channel matrix, where $j \geq 0$. Then, u_0 is recoverable at time instant j if

$$d^j_{CR}(\mathcal{C}) > n(j+1) - \operatorname{rank}(A_{[0,j]}).$$

We will give here the proof of the Theorem 4.3.8. The proof follows the explanations in [AN21, p. 9-10] with some adjustments to our definitions. It should be remembered that A is the upper triangular channel matrix. Let v be the channel packet sequence. The received vector y is defined as:

$$y = vA \iff y^T = A^T v^T$$

To prove the final theorem, we need another theorem as a baseline:

Theorem 4.3.9. [AN21, Theorem 3] Let $\mathcal{C} = \ker(H(D)) \subset \mathbb{F}_{q^m}[D]^n$. Then, the following statements are equivalent

1) $d^j_{CR}(\mathcal{C}) = d_R(\mathcal{C}^c_j) = d$ ($\mathcal{C}^c_j$ is defined in Definition 2.2.22)

2) None of the first n columns of $H_{trunc}(j)A_{[0,j]}^T$ are contained in the $\mathbb{F}_{q^m}$-span of any other $d-2$ column of $H_{trunc}(j)A_{[0,j]}^T$ for all non-singular block upper triangular $A_{[0,j]} \in \mathbb{F}_q^{n(j+1)\times n(j+1)}$. And, moreover, one of the first n columns of $H_{trunc}(j)A_{[0,j]}^T$ is in the $\mathbb{F}_{q^m}$-span of other $d-1$ columns of $H_{trunc}(j)A_{[0,j]}^T$.

Proof. (2 $\Rightarrow$ 1): Suppose that one of the first n columns of $H_{trunc}(j)A_{[0,j]}^T$ is in the $\mathbb{F}_{q^m}$-span of other $d-1$ columns of $H_{trunc}(j)A_{[0,j]}^T$ for a non-singular upper triangular $A_{[0,j]} \in \mathbb{F}_q^{n(j+1)\times n(j+1)}$. Thus, $H_{trunc}(j)A_{[0,j]}^T$ has d columns linearly dependent, say $c_{i_1}, c_{i_2}, \ldots, c_{i_d}$, therefore there exists $\alpha_l \in \mathbb{F}_{q^m}$, $l \in S = \{i_1, i_2, \ldots, i_d\}$ such that

$$\sum_{l\in S} \alpha_l c_l = 0 \tag{4.2}$$

and at least one element in S, say i_1, belongs to $\{1, \ldots n\}$. Now take $w = (w_1, w_2, \ldots, w_{n(j+1)}) \in \mathbb{F}_{q^m}^{n(j+1)}$ with:

$$w_l = \begin{cases} 0 & \text{if } l \notin S \\ \alpha_l & \text{if } l \in S \end{cases}$$

$H_{trunc}(j)A_{[0,j]}^T w^T = 0$ since (4.2), then $(A_{[0,j]}^T w^T)^T = wA_{[0,j]}$ is a codeword. Since $\phi(w)$ has at most d non-zero columns and $A_{[0,j]}$ is non-singular then $\phi(wA_{[0,j]})$ has rank at most d. Thus, $d_{CR}^j(\mathcal{C}) \leq d$.
It will be shown by contradiction that $d_{CR}^j(\mathcal{C}) \geq d$. Assume $d_{CR}^j(\mathcal{C}) < d$. Then, there exists a codeword $v = (v_0, v_1, \ldots, v_j) \in \mathbb{F}_{q^m}^{n(j+1)}$ with $v_0 \neq 0$, $\text{rank}(\phi(v)) \leq d-1$ and $vH_{trunc}(j)^T = H_{trunc}(j)v^T = 0$. Since $v_0 \neq 0$, at least one of the first n coordinates of v is non-zero. Take a non-singular block upper triangular matrix $B_{[0,j]} \in \mathbb{F}_q^{n(j+1)\times n(j+1)}$ such that $\phi(v)B_{[0,j]}$ has at most $d-1$ non-zero columns. Then, $vB_{[0,j]}$ has at most $d-1$ non-zero coordinates. Let $A_{[0,j]}^T = (B_{[0,j]}^T)^{-1}$, then

$$H_{trunc}(j)A_{[0,j]}^T(B_{[0,j]}^T v^T) = 0.$$

This implies that one of the first n columns of $H_{trunc}(j)A_{[0,j]}^T$ is contained in the span of other $d-2$ columns of $H_{trunc}(j)A_{[0,j]}^T$. By the first part of the statement 2) we obtain a contradiction. Hence, $d_{CR}^j(\mathcal{C}) = d$.

(1 $\Rightarrow$ 2): This direction follows the same idea as the other direction. We like to prove in (a) that if the first statement of 2) is not fulfilled, then $d_{CR}^j(\mathcal{C}) < d$. This is equivalent to $d_{CR}^j(\mathcal{C}) \geq d$ which implies the first statement of 2). In (b) we show that $d_{CR}^j(\mathcal{C}) = d$ implies the second statement of 2). (a) and (b) together prove that 1 $\Rightarrow$ 2.
(a): We assume it is true that one of the first n columns in $H_{trunc}(j)A_{[0,j]}^T$ lies in the span of $d-2$ other columns in $H_{trunc}(j)A_{[0,j]}^T$. Thus, $H_{trunc}(j)A_{[0,j]}^T$ has $d-1$ linearly dependent columns, say $c_{i_1}, c_{i_2}, \ldots, c_{i_{d-1}}$, therefore there exists $\alpha_l \in \mathbb{F}_{q^m}$, $l \in S = \{i_1, i_2, \ldots, i_{d-1}\}$ such that

$$\sum_{l\in S} \alpha_l c_l = 0 \tag{4.3}$$

and at least one element in S, say i_1, belongs to $\{1, \ldots n\}$. Now take $w = (w_1, w_2, \ldots, w_{n(j+1)}) \in \mathbb{F}_{q^m}^{n(j+1)}$ with:

$$w_l = \begin{cases} 0 & \text{if } l \notin S \\ \alpha_l & \text{if } l \in S \end{cases}$$

$H_{trunc}(j)A_{[0,j]}^T w^T = 0$ by Equation (4.3) and then $wA_{[0,j]}$ is a codeword. Since $\phi(w)$ has at most $d-1$ non-zero columns and $A_{[0,j]}$ is non-singular, then $\phi(wA_{[0,j]})$ has rank at most $d-1$. Thus, $d_{CR}^j(\mathcal{C}) \leq d-1$ and $d_{CR}^j(\mathcal{C}) < d$.
(b): The second part is proven by a direct proof. We have $d_{CR}^j(\mathcal{C}) = d$. Then, there exists a codeword $v = (v_0, v_1, \ldots, v_j) \in \mathbb{F}_{q^m}^{n(j+1)}$ with $v_0 \neq 0$, $\text{rank}(\phi(v)) = d$ and $H_{trunc}(j)v^T = 0$. Since $v_0 \neq 0$, at least one of the first n coordinates of v is non-zero. Take a non-singular block upper triangular matrix $B_{[0,j]} \in \mathbb{F}_q^{n(j+1)\times n(j+1)}$ such that $\phi(v)B_{[0,j]}$ has at most d non-zero columns. Then, $vB_{[0,j]}$ has at most d non-zero coordinates. Let $A_{[0,j]}^T = (B^T[0,j])^{-1}$, then

$$H_{trunc}(j)A_{[0,j]}^T(B_{[0,j]}^T v^T) = 0.$$

This implies that one of the first n columns of $H_{trunc}(j)A_{[0,j]}^T$ is contained in the span of other $d-1$ columns of $H_{trunc}(j)A_{[0,j]}^T$.

With that, the equivalence is proven. □

Proof of the Theorem 4.3.8

Proof. As $G(D)$ is basic, G_0 is full row rank and therefore to recover u_0, it is enough to decode v_0 ($u_0G_0 = v_0$ is solvable). Obviously, if $\text{rank}(A_{00}) = n$, there is no rank deficiency at time instant zero and v_0 can be recovered immediately. Assume now that the $\text{rank}(A_{00}) < n$. The received vector $y = (y_0, y_1, \ldots, y_j) \in \mathbb{F}_{q^m}^{n(j+1)}$ and $A_{[0,j]}$ are known and we aim to find $v = (v_0, v_1, \ldots, v_j) \in \mathbb{F}_{q^m}^{n(j+1)}$ such that $y = vA_{[0,j]} = (A_{[0,j]}^T v^T)^T$ and $vH_{trunc}^T(j) = H_{trunc}v^T = 0$. We will show that v_0 is uniquely determined if $d_{CR}^j(\mathcal{C}) > n(j+1) - \text{rank}(A_{[0,j]})$. By the Gauss elimination, there exists a non-singular block lower triangular matrix $B_{[0,j]} \in \mathbb{F}_q^{n(j+1)\times n(j+1)}$ such that the non-zero columns of $A_{[0,j]}^T B_{[0,j]}$ are linearly independent. Clearly, the number of non-zero columns of $A_{[0,j]}^T B_{[0,j]}$ is equal to the $\text{rank}(A_{[0,j]})$. Denote $\bar{v}^T = (\bar{v}_0, \ldots, \bar{v}_j)^T = B_{[0,j]}^{-1}(v_0, \ldots, v_j)^T$. First, we find $\bar{v}$ in order to obtain v_0. From the equation

$$y^T = A_{[0,j]}^T B_{[0,j]} \bar{v}^T,$$

$\text{rank}(A_{[0,j]})$ coordinates of $\bar{v}$ can be uniquely determined since $A_{[0,j]}^T B_{[0,j]}$ has $\text{rank}(A_{[0,j]})$ non-zero columns which are all linearly independent. Notice that this system of equations always admits a non-zero solution. The remaining coordinates must satisfy the system of linear equations

$$(H_{trunc}(j)B_{[0,j]})\bar{v}^T = 0.$$

The unknowns in $\bar{v}$ satisfy a system of linear equations $Dx = c$ where x represents the vector of t unknowns, where $t = n(j+1) - \text{rank}(A_{[0,j]})$ and $c \in \mathbb{F}_{q^m}^{k(j+1)}$ is a constant vector. Notice that some of the $k(j+1)$ equations may be trivial. This system is always solvable since $H_{trunc}(j)v^T = 0$. By the assumption that $d_{CR}^j(\mathcal{C}) > n(j+1) - \text{rank}(A_{[0,j]})$, it follows that $t < d_{CR}^j(\mathcal{C})$. Let x_i be one of the unknowns of $\bar{v}_0$. Since $t - 1 \leq d_{CR}^j(\mathcal{C}) - 2$, the column corresponding to x_i of D is not a linear combination of the other $t-1$ columns by Theorem 4.3.9. Therefore, x_i can be uniquely determined. Since this is true for any unknown of $\bar{v}$, we can recover the whole vector $\bar{v}_0$. Finally, as $B_{[0,j]}^{-1}$ is also non-singular and a block lower triangular matrix, v_0 is recovered as well. □

Example 4.3.10. [AN21, Example 2] The proof of Theorem 4.3.8 is constructive and provides a decoding procedure to recover u_0 when the conditions of the theorem are satisfied which is illustrated in this example.

Let $H(D) = H_0 + H_1 D \in \mathbb{F}_{2^6}[D]^{1\times 3}$ be a parity-check matrix of the convolutional code $\mathcal{C} \subset \mathbb{F}_{2^6}[D]^3$, α a primitive element of $\mathbb{F}_{2^6}$ such that $\alpha^6 = \alpha + 1$ and

$$H_0 = \begin{bmatrix} \alpha^5 & \alpha^3 & \alpha^2 \end{bmatrix}, \qquad H_1 = \begin{bmatrix} \alpha^4 & 1 & 1 \end{bmatrix}.$$

Thus,

$$H_{trunc}(1) = \begin{bmatrix} \alpha^5 & \alpha^3 & \alpha^2 & 0 & 0 & 0 \\ \alpha^4 & 1 & 1 & \alpha^5 & \alpha^3 & \alpha^2 \end{bmatrix}$$

and it is straightforward to verify that $d^1_{CR}(\mathcal{C}) = 2$ and therefore by Theorem 4.3.8 each truncated codeword $v_{[0,1]}$ can be recovered if the rank deficiency in the time interval $[0, 1]$ is not larger than 1. In other words, the rank of $A_{[0,1]}$ must be greater than or equal to 5 such that

$$d^1_{CR}(\mathcal{C}) = 2 > 6 - \operatorname{rank}(A_{[0,1]}) = n(j+1) - \operatorname{rank}(A_{[0,1]}).$$

Consider for instance that the channel matrix is

$$A_{[0,1]} = \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right]$$

and the received vector is $y = (0, 1, 0 \,|\, 0, 1 + \alpha^3, \alpha)$. We aim at finding $v_0 \in \mathbb{F}^3_{2^6}$ with $v = (v_0, v_1) \in \mathbb{F}^6_{2^6}$ such that $y = vA_{[0,1]}$ and $H_{trunc}(1)v^T = 0$. Following the notation and the steps described in the proof of Theorem 4.3.8, one obtains

$$A^T_{[0,1]}B_{[0,1]} = \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{array}\right] \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{array}\right] = \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right].$$

Since the base field has characteristic 2, $B^{-1}_{[0,1]} = B_{[0,1]}$. Define $\bar{v}^T = B^{-1}_{[0,1]}v^T$. Let $\bar{v} = (\bar{v}_0, \bar{v}_1) = (\bar{v}_{00}, \bar{v}_{01}, \bar{v}_{02} \,|\, \bar{v}_{10}, \bar{v}_{11}, \bar{v}_{12})$, then from equation

$$y^T = A^T_{[0,1]}B_{[0,1]}\bar{v}^T$$

we get

$$y = (0, 1, 0 \,|\, 0, 1 + \alpha^3, \alpha) = (\bar{v}_{00}, \bar{v}_{01}, 0 \,|\, \bar{v}_{10}, \bar{v}_{11} + \bar{v}_{01}, \bar{v}_{12})$$

and obtain

$$\bar{v}_0 = (0, 1, x_2)$$
$$\bar{v}_1 = (0, \alpha^3, \alpha),$$

where x_2 can be computed solving $H_{trunc}(1)B_{[0,1]}\bar{v}^T = 0$, which yields

$$\begin{cases} \alpha^3 + x_2\alpha^2 & = 0 \\ 1 + (1 + \alpha^2)x_2 + \alpha^6 + \alpha^3 & = 0 \end{cases}$$

and $x_2 = \alpha$. With that we get:

$$\bar{v} = (0, 1, \alpha \,|\, 0, \alpha^3, \alpha)$$

The recovered codeword is then $v^T = B_{[0,1]}\bar{v}^T = (0, 1, \alpha \,|\, 0, \alpha^3, 0)^T$.

Example 4.3.11. In the Example 4.3.7 above, the codeword $v_{[0,1]} = (1, 0, 0 \,|\, 0, 1, 0)$ has column rank distance 1. This implies that $d^1_{CR}(\mathcal{C}) = 1$. Theorem 4.3.8 states that u_0 is recoverable if:

$$d^1_{CR}(\mathcal{C}) > n(1+1) - \operatorname{rank}(A_{[0,1]}) \quad \Leftrightarrow \quad 1 > 6 - \operatorname{rank}(A_{[0,1]}) \quad \Leftrightarrow \quad \operatorname{rank}(A_{[0,1]}) \geq 6$$

This condition is not fulfilled for the matrix A in Example 4.3.7. So, u_0 is not recoverable for sure. This coincides with the calculated result.

Since $\mathcal{C}$ is a time-invariant convolutional code, we can reformulate Theorem 4.3.8 to the following corollary:

Corollary 4.3.12. [AN21, Corollary 1] Assume that we have been able to correctly recover all packets up to an instant $T_0 - 1$. Let $T_f \geq T_0$ and $T = T_f - T_0$. Then, we can recover u_{T_0} at time instant T_f if

$$d^T_{CR}(\mathcal{C}) > n(T+1) - \operatorname{rank}(A_{[T_0,T_f]}).$$

In contrast to the network model in the last section, the network with delay uses the column rank as a good decoding measure. Theorem 2.2.24 shows that the column rank of a window of length W (e.g. from 0 to $W-1$) is bounded by

$$d^{W-1}_{CR} \leq W(n-k) + 1.$$

Lemma 2.2.26 states that this bound can be achieved in the best case up to a window size of $L+1$ where L is the following constant:

$$L = \left\lfloor \frac{\delta}{k} \right\rfloor + \left\lfloor \frac{\delta}{n-k} \right\rfloor$$

If this bound is achieved, the code is called *MRP* (see Definition 2.2.27). Since the column rank distance is a meaningful measure, we are interested in the construction of such MRP codes. Chapter 5 will show a construction of this type.
We can now state a stronger version of the above Theorem 4.3.8 and Corollary 4.3.12 if an MRP code is given.

Theorem 4.3.13. [AN21, Theorem 7] Let $\mathcal{C} = \operatorname{Im}(G(D))$ be an MRP code. Assume that we have been able to recover all packets up to an instant $T_0 - 1$ and that T_f is the first instant such that

$$d^T_{CR}(\mathcal{C}) > n(T+1) - \operatorname{rank}(A_{[T_0,T_f]}), \quad T = T_f - T_0$$

is satisfied. Then, the whole source packet vector $(u_{T_0}, \ldots, u_{T_f})$ can be recovered.

Proof. Let $A_{[T_0,T_0+j]}$ be the truncated channel matrix, $0 \leq j \leq T$. Trying to recover u_{T_0} unsuccessfully up to $T_f - 1$ using an MRP code is equivalent to

$$\operatorname{rank}(A_{[T_0,T_0+j]}) \leq n(j+1) - d^j_{CR}(\mathcal{C}) = k(j+1) - 1 \tag{4.4}$$

for $0 \leq j < T - 1$. The first inequality follows with Corollary 4.3.12. For the equality, we used that $\mathcal{C}$ is an MRP code with the fact in Lemma 2.2.25.

Since

$$d^T_{CR}(\mathcal{C}) > n(T+1) - \operatorname{rank}(A_{[T_0,T_f]})$$

holds by assumption for T_f, it follows:

$$\operatorname{rank}(A_{[T_0,T_f]}) > n(T+1) - d_{CR}^{T}(\mathcal{C}) = n(T+1) - ((n-k)(T+1)+1) = k(T+1) - 1$$
$$\Longleftrightarrow$$
$$\operatorname{rank}(A_{[T_0,T_f]}) \geq k(T+1)$$

By Corollary 4.3.12, u_{T_0} can be recovered. Now, shift one time instant forward and consider the time window $[T_0+1, T_f]$ for decoding. As $\operatorname{rank}(A_{[T_0,T_0]}) \leq k-1$ by Equation (4.4), it follows from the upper block triangular configuration of $A_{[T_0,T_f]}$ that

$$\operatorname{rank}(A_{[T_0+1,T_f]}) = \operatorname{rank}(A_{[T_0,T_f]}) - \operatorname{rank}(A_{[T_0,T_0]}) \geq k(T+1) - (k-1) > kT,$$

and thus again Corollary 4.3.12 guarantees that u_{T_0+1} can be computed. The same procedure is applied sequentially to retrieve the remaining u_{T_0+j}, $j = 2, 3, \ldots, T$. □

Chapter 5

Constructions

In this chapter, we will present different constructions of codes which are well suited for our discussed network coding models. These code constructions differ from code to code since different coding models have distinct requirements. In the course of this chapter, we will provide most of the proofs and show the connections to theorems and lemmas already mentioned. This will help to give a better intuition and improves the level of understandability.

The following rank metric convolutional codes will be presented in this chapter:

- Memory Maximum Sum Rank Code (m-MSR) in Section 5.1
- Maximum Rank Profile Code (MRP) in Section 5.2
- Maximum Rank Distance Code (MRD) in Section 5.3
- (Partial) Unit Memory Code ((P)UM) in Section 5.4

5.1 Construction of Memory Maximum Sum Rank Codes

In this section, we will present a construction of an m-MSR convolutional code as defined in Definition 2.2.19. It is based on [MBK16]. m-MSR codes play an important role for the rank deficient sliding window network without time delay (see Section 4.3.1) because they have useful decoding properties for this model.

5.1.1 Code Construction

First of all, we need the definition of a superregular matrix:

Definition 5.1.1. (Trivial Term of the Determinant) [AN21, Section 5.1] Let $F = [a_{i,j}]$ be a square matrix of order n over $\mathbb{F}_{q^m}$ and S_n the symmetric group of order n. Recall that the determinant of F is given by

$$|F| = \sum_{\sigma \in S_n} sgn(\sigma) a_{1,\sigma(1)} \cdots a_{n,\sigma(n)}. \tag{5.1}$$

A *trivial term* of the determinant is a term of (5.1), $a_{1,\sigma(1)} \cdots a_{n,\sigma(n)}$, equal to zero.

Definition 5.1.2. (Trivial Minor) [AN21, Section 5.1] If F is a square submatrix of a matrix B with entries in $\mathbb{F}_{q^m}$ and all the terms of the determinant of F are trivial, we say that $|F|$ is a *trivial minor* of B. We can also say that F has a trivial determinant.

Definition 5.1.3. (Superregular Matrix) [AN21, Section 5.1] We say that a matrix B is *superregular* if all its non-trivial minors are different from zero.

Example 5.1.4. Consider the matrix

$$A = \begin{bmatrix} 2 & 4 & 1 \\ 3 & 0 & 0 \end{bmatrix}$$

over $\mathbb{F}_5$. We can calculate all minors of length 2:

$$\begin{vmatrix} 2 & 4 \\ 3 & 0 \end{vmatrix} = 2 \cdot 0 - 4 \cdot 3 = 3 \qquad \begin{vmatrix} 2 & 1 \\ 3 & 0 \end{vmatrix} = 2 \cdot 0 - 1 \cdot 3 = -3 = 2 \qquad \begin{vmatrix} 4 & 1 \\ 0 & 0 \end{vmatrix} = 4 \cdot 0 - 1 \cdot 0 = 0$$

Then, the first two minors are not zero and the last minor is zero but also trivial. Additionally, all 1×1 minors are non-zero or trivial. Summing up, every minor fulfils the condition for superregularity and A is superregular.

Theorem 5.1.5. [MBK16, Theorem 3] For $0 \leq t \leq j$, let $0 \leq p_t \leq n$ satisfy

$$\sum_{i=0}^{t} p_i \leq k(t+1) \tag{5.2}$$

for all $t \leq j$ and with *equality* for $t = j$, $\sum_{i=0}^{j} p_i = k(j+1)$. The following are equivalent for any rank metric convolutional code $\mathcal{C}$:

1) $d_{SR}^{j}(\mathcal{C}) = (n-k)(j+1)+1$

2) Let $G_{trunc}(j)$ be the truncated generator matrix of $\mathcal{C}$. For all full rank $A^*_{[0,j]} = \text{diag}(A_0^*, \ldots, A_j^*)$ matrices constructed from full rank blocks $A_t^* \in \mathbb{F}_q^{n \times p_t}$ and p_t that satisfy (5.2), the product $G_{trunc}(j)A^*_{[0,j]}$ is non-singular.

Proof. (1 ⇒ 2) Proof by contraposition: Suppose there exists an $A^*_{[0,j]}$ whose blocks satisfy (5.2) and for which $G_{trunc}(j)A^*_{[0,j]}$ is singular. Then, there exists a channel packet sequence $x_{[0,j]}$ where $x_{[0,j]}A^*_{[0,j]} = 0$. We show that this leads to a contradiction of 1).
The contradiction is immediate if $x_0 \neq 0$. In this case, the sum rank of $x_{[0,j]}$ is at least $d^j_{SR}(\mathcal{C})$, i.e.,

$$\sum_{t=0}^{j} \operatorname{rank}(\phi(x_t)) \geq d^j_{SR}(\mathcal{C})$$

must hold. Note however that $x_{[0,j]}$ lies in the kernel of $A^*_{[0,j]}$. The kernel has dimension $n(j+1) - \sum_{t=0}^{j} p_t$ and it follows:

$$\sum_{t=0}^{j} \operatorname{rank}(\phi(x_t)) \leq n(j+1) - \sum_{t=0}^{j} p_t = (n-k)(j+1)$$

This contradicts $d^j_{SR}(\mathcal{C}) = (n-k)(j+1)+1$. Note that we use (5.2) with equality at $t = j$ in the last step.
If $x_0 = 0$, then the sum rank of $x_{[0,j]}$ is not constrained by $d^j_{SR}(\mathcal{C})$. Let $l = \arg\min_t x_t \neq 0$ be the smallest index for which x_t is non-zero and consider the channel packet sequence $x_{[l,j]}$, whose sum rank is at least $d^{j-l}_{SR}(\mathcal{C})$. As $x_t A^*_t = 0$ for $t = l, \dots j$, we bound $\operatorname{rank}(\phi(x_t)) \leq n - p_t$ in this window. The sum rank of $x_{[l,j]}$ is bounded:

$$\sum_{t=l}^{j} \operatorname{rank}(\phi(x_t)) \leq n(j-l+1) - \sum_{t=l}^{j} p_t \leq (n-k)(j-l+1)$$

The second inequality follows from $\sum_{t=l}^{j} p_t \geq k(j-l+1)$ which can be derived when (5.2) is met with equality for $t = j$:

$$\sum_{t=l}^{j} p_t = \sum_{t=0}^{j} p_t - \sum_{t=0}^{l-1} p_t = k(j+1) - \sum_{t=0}^{l-1} p_t \geq k(j+1) - k(l-1+1) = k(j-l+1)$$

Due to Lemma 2.2.17, the column sum rank achieves $d^{j-l}_{SR}(\mathcal{C}) = (n-k)(j-l+1)+1$ if $d^j_{SR}(\mathcal{C}) = (n-k)(j+1)+1$. The sum rank of $x_{[l,j]}$ is less than $d^{j-l}_{SR}(\mathcal{C})$ which implies $d^{j-l}_{SR}(\mathcal{C}) \neq (n-k)(j-l+1)+1$. Thus gives $d^j_{SR}(\mathcal{C}) \neq (n-k)(j+1)+1$ which is a contradiction to 1).

(2 ⇒ 1) Proof by contraposition: We use a code with $d^j_{SR}(\mathcal{C}) \leq (n-k)(j+1)$ and construct a full rank $A^*_{[0,j]}$ for which $G_{trunc}(j)A^*_{[0,j]}$ is singular. Let $m = \arg\min_i \{d^i_{SR}(\mathcal{C}) \leq (n-k)(i+1)\}$ be the first instance where the column sum rank fails to attain its upper bound and consider the sequence $x_{[0,m]}$ with the weight equal to $d^m_{SR}(\mathcal{C})$. We show that there exist full rank matrices $A^*_t \in \mathbb{F}_q^{n \times p_t}$ satisfying both (5.2) and $x_t A^*_t = 0$ for $0 \leq t \leq m$. In addition, we aim to have equality in (5.2) at $t = m$ and thus $A^*_{[0,m]}$ will be of dimension $(m+1)n \times (m+1)k$. This is relevant later in the proof.

In the case $m = 0$, the column rank of x_0 cannot exceed $n-k$. For every $p_0 \leq n - \operatorname{rank}(\phi(x_0))$, there exits an A^*_0 for which $x_0 A^*_0 = 0$. Clearly, we can always choose an A^*_0 with rank $p_0 = k$.

When $m > 0$, the sum rank of $x_{[0,t]}$ satisfies

$$\sum_{i=0}^{t} \text{rank}(\phi(x_t)) \geq (n-k)(t+1)+1 \tag{5.3}$$

for $0 \leq t \leq m-1$, and

$$\sum_{i=0}^{m} \text{rank}(\phi(x_t)) \leq (n-k)(m+1). \tag{5.4}$$

Let $p_t = n - \text{rank}(\phi(x_t))$ for $0 \leq t \leq m-1$ and choose an appropriate A_t^* for which $x_t A_t^* = 0$. For all $0 \leq t \leq m-1$, we have

$$\sum_{i=0}^{t} p_i = n(t+1) - \sum_{i=0}^{t} \text{rank}(\phi(x_i)) \leq k(t+1) - 1 \tag{5.5}$$

confirming that (5.2) is satisfied for $t \leq m-1$. Note that in the last step the Inequality (5.3) was applied. We next specify an appropriate choice for p_m. We will select

$$p_m = (m+1)k - \sum_{t=0}^{m-1} p_t \tag{5.6}$$

and show that there exists an associated $A_m^* \in \mathbb{F}_q^{n \times p_m}$ that will satisfy $x_m A_m^* = 0$. It thus suffices to show that p_m also satisfies $p_m \leq n - \text{rank}(\phi(x_m))$. Note that

$$\begin{aligned} n - p_m &= n - (m+1)k + \sum_{i=0}^{m-1} p_i && (5.7)\\ &= (n-k)(m+1) - \sum_{i=0}^{m-1} \text{rank}(\phi(x_i)) && (5.8)\\ &= (n-k)(m+1) - \sum_{i=0}^{m} \text{rank}(\phi(x_i)) + \text{rank}(\phi(x_m)) \\ &\geq \text{rank}(\phi(x_m)) && (5.9) \end{aligned}$$

where Equation (5.7) follows from Equation (5.6), Equation (5.8) follows by using (5.5) with $t = m-1$ and (5.9) follows via the inequality in (5.4). Finally, our choice (5.6) also guarantees that $A_{[0,m]}^*$ has dimension $(m+1)n \times (m+1)k$ as claimed.

The remaining $A_{m+1}^*, \ldots, A_j^*$ can be any full rank $n \times k$ matrices, thus satisfying (5.2) for all $t \leq j$ with equality on $t = j$:

$$\sum_{i=0}^{j} p_i = \sum_{i=0}^{m} p_i + (j-m)k = k(m+1) + k(j-m) = k(j+1)$$

The product $G_{trunc}(j) A_{[0,j]}^*$ can be written as

$$G_{trunc}(j) A_{[0,j]}^* = \begin{bmatrix} G_{trunc}(m) & X \\ & Y \end{bmatrix} \begin{bmatrix} A_{[0,m]}^* & \\ & A_{[m+1,j]}^* \end{bmatrix} = \begin{bmatrix} G_{trunc}(m) A_{[0,m]}^* & X A_{[m+1,j]}^* \\ & Y A_{[m+1,j]}^* \end{bmatrix}$$

where X and Y denote the remaining blocks that comprise $G_{trunc}(j)$. The block $G_{trunc}(m) A_{[0,m]}^*$ is a square matrix with a zero determinant. Therefore, $\det(G_{trunc}(j) A_{[0,j]}^*)$ is also zero. $\square$

Theorem 5.1.6. [MBK16, Theorem 4] For $n, m \in \mathbb{N}$, let $M = q^{n(m+2)-1}$. Let $\alpha \in \mathbb{F}_{q^M}$ be a primitive element and a root of the minimal polynomial $p_\alpha(X)$. For $0 \leq j \leq m$, let the blocks $T_j \in \mathbb{F}_{q^M}^{n\times n}$ be defined by

$$T_j = \begin{bmatrix} \alpha^{[nj]} & \alpha^{[nj+1]} & \dots & \alpha^{[n(j+1)-1]} \\ \alpha^{[nj+1]} & \alpha^{[nj+2]} & \dots & \alpha^{[n(j+1)]} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha^{[n(j+1)-1]} & \alpha^{[n(j+1)]} & \dots & \alpha^{[n(j+2)-2]} \end{bmatrix}, \quad \text{where } g^{[j]} = g^{q^j}.$$

Then, the following block Hankel matrix

$$T = \begin{bmatrix} & & & T_0 \\ & & T_0 & T_1 \\ & \iddots & \vdots & \vdots \\ T_0 & \dots & T_{m-1} & T_m \end{bmatrix} \in \mathbb{F}_{q^M}^{n(m+1)\times n(m+1)} \tag{5.10}$$

is superregular. A block Hankel matrix is a square matrix in which each ascending skew-block-diagonal from left to right is constant [Par89].

Proof. First, we will highlight some properties of the matrix T.
Let $T_j(r, s)$ be the element in the r-th row and s-th column of T_j, where $0 \leq r, s \leq n-1$ and $0 \leq j \leq m$. Each entry of the matrix is a linearised monomial, which is a linearised polynomial with one term, evaluated at $X = \alpha$, in particular

$$T_j(r, s) = \alpha^{[nj+r+s]}. \tag{5.11}$$

The q-degrees of these monomials increase as one moves further down and to the right inside any T_j. This can be described with the following inequalities

$$\deg_q(T_j(r', s)) > \deg_q(T_j(r, s)),\ 0 \leq r < r' \leq n-1 \tag{5.12}$$
$$\deg_q(T_j(r, s')) > \deg_q(T_j(r, s)),\ 0 \leq s < s' \leq n-1 \tag{5.13}$$

for all $0 \leq j \leq m$. Combining all of the T_j to construct T in (5.10) preserves the property of q-degrees strictly increasing downwards (along a single column) or rightwards (along a single row), i.e.,

$$\deg_q(T_{j'}(r', s)) > \deg_q(T_j(r, s)),\ 0 \leq j < j' \leq n-1 \tag{5.14}$$
$$\deg_q(T_{j'}(r, s')) > \deg_q(T_j(r, s)),\ 0 \leq j < j' \leq n-1 \tag{5.15}$$

for all $0 \leq r, r' \leq n-1$ and $0 \leq s, s' \leq n-1$.

To show that T is superregular, let D be any $l \times l$ submatrix of T with $1 \leq l \leq n(m+1)$. All sub-matrices of T preserve the degree inequalities (5.12), (5.13), (5.14) and (5.15). T being superregular is equivalent to $\det(D) \neq 0$ for every D with a non-trivial determinant. The determinant is evaluated using the Leibniz formula as a polynomial $D(\alpha) = \sum_{\sigma\in S_r} sgn(\sigma) D_\sigma$. We define the r-th row and c-th column of D as $D(r, c)$. Each non-zero term in the formula is a product of linearised monomials

$$D_\sigma = \prod_{i=0}^{l-1} D(i, \sigma(i)).$$

Note that the determinant polynomial itself is not linearised. Using (5.12), (5.13), (5.14) and (5.15), the degree of this polynomial can be bounded. We will now first prove the following Property A:

Property A: [MBK16, Lemma 2] For T defined as in (5.10), let D be any square submatrix which is a non-trivial minor. Let $D(X)$ be a polynomial such that $D(\alpha) = det(D)$. The degree of $D(X)$ is bounded by:

$$1 \leq deg(D(X)) < q^{n(m+2)-1}$$

First, we will show the upper bound of Property A. Almeida et al. [ANP13, p. 2150] showed this statement for $q = 2$. Mahmood [Mah15, Appendix A.1] extended the proof to the general case for q being a prime power. We will present this proof here:
The degree of $D(X) = \sum_{\sigma \in S_r} sgn(\sigma) D_\sigma(X)$ is bounded by considering the degree of the polynomial terms $D_\sigma(X) = \prod_{i=0}^{l-1} D(i, \sigma(i))(X)$. $D(r, c)(X)$ defines the r-th row and c-th column of D with X instead of α. When constructing each $D_\sigma(X) = \prod_{i=0}^{l-1} D(i, \sigma(i))(X)$, up to two entries from the last row and column of (5.10) can be used. Choosing $D(l-1, l-1)(X)$ prevents selecting a second entry but due to (5.12), (5.13), (5.14) and (5.15), $D(l-1, l-1)(X)$ has a greater degree than the product of any other two entries of the submatrix D. Note that we can bound $\deg(D(l-1, l-1)(X)) \leq q^{n(m+2)-2}$. We next consider the second last row and column. However, the degrees of the entries here are bounded by $\deg(D(l-1, l-1)(X)) \leq q^{n(m+2)-4}$. This argument is used recursively to bound the degree of $D_\sigma(X)$:

$$\begin{aligned}
\sum_{i=0}^{l-1} \deg(D(i, \sigma(i))(X)) &\leq \sum_{k=0}^{l-1} q^{n(m+2)-2-2k} \\
&= q^{n(m+2)-2} \sum_{k=0}^{l-1} q^{-2k} \\
&< q^{n(m+2)-2} \frac{1}{1 - q^{-2}} \\
&< q^{n(m+2)-1}
\end{aligned}$$

It results in $\deg(D(X)) < q^{n(m+2)-1}$. This bound is a reason why we assumed that $M = q^{n(m+2)-1}$.

Almeida et al. [ANP13] proved the following Property B which is important to prove the lower bound of Property A:

Property B: [ANP13, Statement 1.] If there are non-trivial terms in $D(\alpha) = det(D)$, then there exists a unique *term α^β with highest exponent β.*

Property B also holds for the general case with q being a prime power not equal to 2. It can be derived by algorithmically finding a unique permutation $\bar{\sigma} = \arg\max_\sigma \deg(D_\sigma(X))$ which generates the highest degree monomial term in the Leibniz formula. Because $\bar{\sigma}$ is unique, there is no other term which can negate this monomial. Then, $\deg(D(X)) = \deg(D_{\bar{\sigma}}(X))$, meaning $D(X)$ is not the zero degree polynomial and hence $\deg(D(X)) \geq 1$. Using this, Property A is proven.

In conclusion: Recall that $p_\alpha(X)$ is the minimal polynomial of α over $\mathbb{F}_{q^M}$. The upper bound gives $\deg(D(X)) < \deg(p_\alpha(X))$ which implies that $p_\alpha(X) \nmid D(X)$. So it can be deduced that α cannot be a root of $D(X)$. Together with the lower bound ($\deg(D(X)) \geq 1$) it follows that

$D(\alpha) \neq 0$ if it is a non-trivial minor. Since this is true for an arbitrary minor D, it follows that T is superregular and the theorem is proven. □

Remark 5.1.7. **(Construction of an m-MDS Code)** [MBK16, Section VI.] We permute the rows of T in (5.10) to form the block Toeplitz structure of a truncated generator matrix

$$\bar{T} = \begin{bmatrix} T_0 & T_1 & \dots & T_m \\ & T_0 & \dots & T_{m-1} \\ & & \ddots & \vdots \\ & & & T_0 \end{bmatrix}. \tag{5.16}$$

A block Toeplitz structure defines a matrix in which each block diagonal from left to right remains constant [BG00]. Hence, every submatrix of T has a counterpart in $\bar{T}$ identical up to row permutations and this block Toeplitz matrix is also superregular. $G_{trunc}(m)$ is then constructed as a submatrix of $k(m+1)$ rows from T. This process parallels the construction of m-MDS generator matrices with memory $\mu = m$. More information about the construction of an m-MDS code out of a superregular matrix can be found in [GLRS06] and [HST08].

We will use a similar technique to construct m-MSR codes. Superregularity alone, however, is insufficient. We show that the superregular matrix construction technique above can be modified to produce m-MSR codes.

Theorem 5.1.8. [MBK16, Theorem 5] For $0 \leq t \leq m$, let $A_t \in \mathbb{F}_q^{n\times n}$ be any non-singular matrices. We then construct $A_{[0,m]} = \text{diag}(A_0, \dots, A_m)$. Let $T \in \mathbb{F}_{q^M}^{n(m+1)\times n(m+1)}$ be the superregular matrix in (5.10). If $M \geq q^{n(m+2)-1}$ and α is a primitive *normal* element of $\mathbb{F}_{q^M}$, then $F = TA_{[0,m]}$ is superregular.

$$\begin{aligned} F = TA_{[0,m]} &= \begin{bmatrix} & & & T_0 \\ & & T_0 & T_1 \\ & \iddots & \vdots & \vdots \\ T_0 & \dots & T_{m-1} & T_m \end{bmatrix} \begin{bmatrix} A_0 & & & \\ & A_1 & & \\ & & \ddots & \\ & & & A_m \end{bmatrix} \\ &= \begin{bmatrix} & & & T_0A_m \\ & & T_0A_{m-1} & T_1A_m \\ & \iddots & \vdots & \vdots \\ T_0A_0 & \dots & T_{m-1}A_{m-1} & T_mA_m \end{bmatrix} \in \mathbb{F}_{q^M}^{n(m+1)\times n(m+1)} \end{aligned} \tag{5.17}$$

Proof. First, we will provide a brief overview of the proof to clarify the concept. The detailed proof will follow afterwards.

Schema [AN21, p. 17]: We will prove the superregularity of $F = TA_{[0,m]}$ where the matrices T_j are given by

$$T_j = \begin{bmatrix} \alpha^{[nj]} & \alpha^{[nj+1]} & \dots & \alpha^{[n(j+1)-1]} \\ \alpha^{[nj+1]} & \alpha^{[nj+2]} & \dots & \alpha^{[n(j+1)]} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha^{[n(j+1)-1]} & \alpha^{[n(j+1)]} & \dots & \alpha^{[n(j+2)-2]} \end{bmatrix},$$

A_i are invertible matrices, for $0 \leq i \leq m$ and α be a primitive and normal element of $\mathbb{F}_{q^M}$. To obtain the proof, we first show the following properties:

I) Let $F_{ij} = T_i A_j$, for $0 \le i \le j \le m$. Then

$$F = \begin{bmatrix} & & & F_{0m} \\ & & F_{0(m-1)} & F_{1m} \\ & \iddots & \vdots & \vdots \\ F_{00} & \dots & F_{(m-1)(m-1)} & F_{mm} \end{bmatrix}$$

and

$$F_{ij} = \begin{bmatrix} f_{0j}^{[ni]} & f_{1j}^{[ni]} & \dots & f_{(n-1)j}^{[ni]} \\ f_{0j}^{[ni+1]} & f_{1j}^{[ni+1]} & \dots & f_{(n-1)j}^{[ni+1]} \\ \vdots & \vdots & \ddots & \vdots \\ f_{0j}^{[n(i+1)-1]} & f_{1j}^{[n(i+1)-1]} & \dots & f_{(n-1)j}^{[n(i+1)-1]} \end{bmatrix},$$

where for each $0 \le r \le j \le n-1$, f_{rj} is the value of a linearised polynomial evaluated at α and

(a) $(f_{0j}, f_{1j}, \dots, f_{(n-1)j})$ are linearly independent over $\mathbb{F}_q$

(b) If $s' < s$, the entries of row s of F_{ij} are q-powers of the corresponding entries in row s' by Freshman's Rule.

(c) If $i' < i$, the entries of row s of F_{ij} are q-powers of the corresponding entries in row s of $F_{i'j}$ by Freshman's Rule.

II) The q-degrees of the polynomials associated to the elements of F_{ij} strictly increase downwards on any fixed column for any $0 \le i \le j \le m$.

III) If $i < i'$ and $s \ge i$ then the q-degrees of the polynomials associated to any entry of the row r of F_{is} is smaller than the q-degrees of the polynomials associated to any entry of the row r of $F_{i'(s+i'-i)}$ for any $0 \le r \le n-1$. Notice that the block matrices F_{ij} and $F_{i'(s+i'-i)}$ are in the same block row of F.

IV) If $i < i' \le j$ then the q-degree of the polynomial associated to any entry of the column s of F_{ij} is smaller than the q-degree of the polynomial associated to any entry of the column s of $F_{i'j}$ for any $0 \le s \le n-1$.

V) If D is a square submatrix of F, there exists an invertible matrix M such that

(a) the q-degrees of the polynomials associated with the entries of any row of DM are strictly increasing.

(b) the q-degrees of the polynomials associated with the entries of any column of DM are strictly increasing.

Next, if a matrix $\hat{D} = DM$ satisfies properties V)(a) and V)(b) above then $\hat{D}$ is invertible by Property A in the proof of Theorem 5.1.6. Therefore, DM is invertible and since M is invertible, D is also invertible. This implies that F is superregular.

To draw the connection to the schema, we will write the roman letters at the specific positions:

Detailed Proof: We first introduce some properties of F which is defined in (5.17). To simplify the notation, we assume that

$$A_0 = A_1 = \cdots = A_m = A.$$

Then, the product can be written as:

$$F = TA_{[0,m]} = \begin{bmatrix} & & & T_0A \\ & & T_0A & T_1A \\ & \iddots & \vdots & \vdots \\ T_0A & \dots & T_{m-1}A & T_mA \end{bmatrix}$$

This allows free use of the polynomial structures and bounds from the previous theorem. Later, we will explain why the proof extends immediately to the general case. Now to the properties of F:

Property 1: The product $F = TA_{[0,m]}$ is a block Hankel matrix, whose blocks each possess a Moore structure (Schema I). A Moore structure describes a matrix which has successive powers of the Frobenius automorphism applied to its rows [Moo96]. Let $F_j = T_jA$ for $j \in \{0, \dots, m\}$. To show that F_j has a Moore structure, let $(f_0, \dots, f_{n-1})$ be the first row of F_0, i.e.,

$$(f_0, \dots, f_{n-1}) = (\alpha^{[0]}, \dots, \alpha^{[n-1]})A.$$

Each element f_i is a linearised polynomial $f_i(X)$ evaluated at $X = \alpha$ with coefficients from $A = [A_{l,i}]$, i.e.,

$$f_i = \sum_{l=0}^{n-1} A_{l,i}\alpha^{[l]}. \tag{5.18}$$

Since $\alpha \in \mathbb{F}_{q^M}$ and $A_{l,i} \in \mathbb{F}_q$, invoking Freshman's rule

$$f_i(X^{[s]}) = f_i^{[s]}(X) \quad \text{for } s \geq 0, \tag{5.19}$$

results in the following Moore structure,

$$F_j = \begin{bmatrix} f_0^{[nj]} & f_1^{[nj]} & \dots & f_{n-1}^{[nj]} \\ f_0^{[nj+1]} & f_1^{[nj+1]} & \dots & f_{n-1}^{[nj+1]} \\ \vdots & \vdots & \ddots & \vdots \\ f_0^{[n(j+1)-1]} & f_1^{[n(j+1)-1]} & \dots & f_{n-1}^{[n(j+1)-1]} \end{bmatrix} \quad \text{for } 0 \leq j \leq m. \tag{5.20}$$

Since α is normal over $\mathbb{F}_{q^M}$ and A has full rank, $(f_0, \dots, f_{n-1})$ are linearly independent over $\mathbb{F}_q$. So, F_j resembles a Gabidulin code [Gab85].

Property 2: The q-degrees of the polynomial elements of F_j strictly increase downwards on any fixed column. They are not necessarily monotonic across a fixed row. From (5.18) and (5.19), the q-degree of each linearised polynomial is bounded by:

$$s \leq \deg_q(f_i^{[s]}(X)) \leq n - 1 + s \quad \text{for } 0 \leq s \leq n(m+1) - 1 \tag{5.21}$$

Note that both the upper and lower bound in (5.21) do not depend on the column index i. Furthermore, all elements on any fixed row in F_j share the same $[s]$ (5.20). Consequently, the polynomial entries on any fixed row of the block F_j share the same bound and are not necessarily in increasing order. Thus, we do not have a counterpart of (5.13) which guarantees that the q-degrees of elements are monotonically increasing across each row in T_j. The counterpart of (5.12), however, is satisfied. Let $F_j(r, s)$ be the element in the r-th row

and s-th column of F_j, i.e.,

$$F_j(r,s) = f_s^{[nj+r]}, \text{ where } 0 \le r,s \le n-1 \text{ and } 0 \le j \le m. \tag{5.22}$$

The Moore structure of F_j implies that the q-degrees of the polynomial entries strictly increase from top to bottom along any fixed column of F_j (Schema II), i.e.,

$$\deg_q(F_j(r',s)) > \deg_q(F_j(r,s)),\ 0 \le r < r' \le n-1, \text{ for all } 0 \le j \le m. \tag{5.23}$$

The following property establishes the relations between q-degrees of the polynomial entries inside F (Schema III and IV).

Property 3: The following inequalities

$$\deg_q(F_{j'}(r',s)) > \deg_q(F_j(r,s)),\ 0 \le j < j' \le m \tag{5.24}$$
$$\deg_q(F_{j'}(r,s')) > \deg_q(F_j(r,s)),\ 0 \le j < j' \le m \tag{5.25}$$

hold for all $0 \le r,r' \le n-1$ and $0 \le s,s' \le n-1$.
The Inequality (5.24) is established by

$$\begin{aligned}\deg_q(F_{j'}(r',s)) - \deg_q(F_j(r,s)) &= \deg_q(f_s^{[nj'+r']}) - \deg_q(f_s^{[nj+r]}) \\ &= (nj'+r') + \deg_q(f_s) - (nj+r) - \deg_q(f_s) \\ &\ge n + r' - r \\ &> 0\end{aligned}$$

where the first inequality follows from $j' \ge j+1$ and the second due to $r' - r > -n$.
We prove (5.25):

$$\begin{aligned}\deg_q(F_{j'}(r,s')) - \deg_q(F_j(r,s)) &= \deg_q(f_{s'}^{[nj'+r]}) - \deg_q(f_s^{[nj+r]}) \\ &\ge (nj'+r) - (n-1+nj+r) = n(j'-j-1)+1 \\ &> 0\end{aligned}$$

Here, the first inequality follows from (5.21), and the second due to $j' \ge j+1$.

We note that (5.23), (5.24) and (5.25) are similar to the Inequalities (5.12), (5.14) and (5.15) for entries of T. It remains to find an equivalent relationship of (5.13) for the entries of F. To state the next property, we let $C_i \in \mathbb{F}_{q^M}^{n(m+1)\times n}$ and $R_i \in \mathbb{F}_{q^M}^{n\times n(m+1)}$ be the i-th column and row blocks of F, i.e.,

$$F = \Big[C_0, C_1, \ldots, C_m\Big] = \begin{bmatrix} R_0 \\ R_1 \\ \vdots \\ R_m \end{bmatrix} \tag{5.26}$$

where

$$C_i = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ F_0 \\ F_1 \\ \vdots \\ F_i \end{pmatrix} \quad \text{and} \quad R_i = \Big(0, \ldots, 0, F_0, F_1, \ldots, F_i\Big). \tag{5.27}$$

Property 4: [ANP13, p. 2151] Let D be an $l \times l$ submatrix of F where $|D|$ is a non-trivial minor. The matrix D can be written as

$$D = \left[\begin{array}{c|c|c} O_h & & \\ \hline O_{h-1} & & \\ \hline \vdots & D_{h-1} & D_h \\ \hline O_0 & & \\ \hline D_0 & \cdots & \end{array}\right], \quad \text{where } 0 \leq h \leq m. \tag{5.28}$$

Each O_i is a zero matrix drawn from a single row block of F, whereas each D_i is a matrix containing non-zero polynomials drawn from a single column block of F. Let $l_i \in \{1, \ldots, n\}$ be the number of columns in each D_i for $i \in \{1, \ldots, h\}$. Then, $\sum_{i=0}^{h} l_i = l$.
In the special case of $h = m$, D_i consists of columns drawn from the non-zero elements of C_i, whereas O_i consists of rows drawn from the zero elements of R_{m-i}. When $h < m$, it can be shown that D preserves the structure in (5.28). If the j-th column block C_j of F was skipped when generating the submatrix D, one has to skip the $(m-j)$-th row block R_{m-j} to avoid generating a trivial submatrix D. The complete proof can be found in [ANP13]. Besides, the degrees of the polynomial entries of D satisfy (5.23), (5.24) and (5.25).

Now, we present Property 5 that is needed later for the proof of Theorem 5.1.8.
Property 5: [MBK16, Lemma 3] For $1 \leq l \leq n$, let $i_1, i_2, \ldots, i_l$ be l distinct elements of the set $\{0, \ldots, n-1\}$. There exists a full rank matrix $M \in \mathbb{F}_q^{l \times l}$ such that the q-degrees of

$$(\hat{f}_{i_1}, \hat{f}_{i_2}, \ldots, \hat{f}_{i_l}) = (f_{i_1}, f_{i_2}, \ldots, f_{i_l})M$$

are monotonically increasing. Moreover, the q-degrees of the polynomial entries of the product $(f_{i_1}^{[s]}, f_{i_2}^{[s]}, \ldots, f_{i_l}^{[s]})M$ are also sorted in the same order for all $0 \leq s \leq n(m+1)-1$. Remember that the f_j are defined in Property 1.

We will now prove Property 5. Because α is a normal element of $\mathbb{F}_{q^M}$ and A is full rank, the polynomials $(f_{i_1}, f_{i_2}, \ldots, f_{i_l})$ are linearly independent over $\mathbb{F}_q$. As a result, the isomorphic matrix $\phi((f_{i_1}, f_{i_2}, \ldots, f_{i_l}))$ (ϕ maps the vector of length l to the corresponding matrix in $\mathbb{F}_q^{M \times l}$, see Example 2.2.1) has full column rank and can be transformed to reduced column echelon form $\phi((\hat{f}_{i_1}, \hat{f}_{i_2}, \ldots, \hat{f}_{i_l}))$ through elementary column operations, i.e.,

$$\phi((\hat{f}_{i_1}, \hat{f}_{i_2}, \ldots, \hat{f}_{i_l})) = \phi((f_{i_1}, f_{i_2}, \ldots, f_{i_l}))M,$$

where $M \in \mathbb{F}_q^{l \times l}$ is a full rank matrix in the ground field. This is equivalent to say that the degrees of the polynomials $(\hat{f}_{i_1}, \hat{f}_{i_2}, \ldots, \hat{f}_{i_l})$ are strictly increasing since $\phi((\hat{f}_{i_1}, \hat{f}_{i_2}, \ldots, \hat{f}_{i_l}))$ is

in reduced column echelon form. The second part immediately follows, as

$$(f_{i_1}^{[s]}, f_{i_2}^{[s]}, \ldots, f_{i_l}^{[s]})M = (\hat{f}_{i_1}^{[s]}, \hat{f}_{i_2}^{[s]}, \ldots, \hat{f}_{i_l}^{[s]})$$

by using Freshman's rule. This proves Property 5.

We are now able to prove the whole theorem:
Proof of Theorem 5.1.8: Consider any non-trivial minor D of F. We will show that $\det(D) \neq 0$. First, note from Property 2 and 3 that the elements of D satisfy (5.23), (5.24) and (5.25), which are counterparts of (5.12), (5.14) and (5.15) respectively. We will prove that D can be transformed into a matrix $\hat{D}$ that satisfies the counterpart of (5.13).
Using Property 4, we have that the structure of D should satisfy (5.28). Consider the submatrices $D_0, \ldots, D_h$ associated with D. Each D_j is contained entirely within a column block of F. Thus, it follows from (5.20) that any row of D_j can be expressed in the form $(f_{i_1}^{[s]}, f_{i_2}^{[s]}, \ldots, f_{i_{l_j}}^{[s]})$ for some $0 \leq s \leq n(m+1)-1$ and $0 \leq i_1 < i_2 < \cdots < i_{l_j} \leq n-1$. In particular, the indices $i_1, \ldots, i_{l_j}$ are common across all rows in D_j and only the q-degree, denoted by s, varies across the rows. Using Property 5, we construct a full rank matrix $M_j \in \mathbb{F}_q^{l_j \times l_j}$ such that the degrees of the polynomials of $\hat{D}_j = D_j \cdot M_j$ will increase monotonically across the columns within each fixed row. Performing this for each D_j, we obtain

$$\hat{D} = D \cdot M,$$

where $M = \mathrm{diag}(M_0, \ldots, M_h)$ has full rank with elements in $\mathbb{F}_q$ (Schema V). It follows that the transformed matrix $\hat{D}$ satisfies the counterpart of (5.13). We argue that the remaining conditions are also preserved under the transformation of M. Note that the elements on any given row of $\hat{D}_j$ are obtained through linear combinations over $\mathbb{F}_q$ of the elements in the corresponding row of D_j. Since both (5.21) and (5.22) are satisfied by all elements belonging to a fixed row of D_j, they must also be satisfied by $\hat{D}_j$. Then, the proof of (5.23), (5.24) and (5.25) can be carried out for elements in $\hat{D}_j$.
Consequently, $\hat{D}_j$ completely satisfies all conditions of (5.12), (5.13), (5.14) and (5.15), in contrast to D which only satisfies (5.23), (5.24) and (5.25). The proof showing that $\hat{D}$ is non-singular then follows verbatim to the proof of Theorem 5.1.6 with Property A. So, we have that $\hat{D}$ is non-singular. Then, $\det(\hat{D}) = \det(D)\det(M)$ and $\det(M) \neq 0$ since M has full rank. This implies that D is also non-singular and therefore, F is superregular.
Recall for this proof that we fixed $A_0 = A_1 = \cdots = A_m$. Without this assumption, the product F would be comprised of a different set of $f_i(\alpha)$ for each column block. Because each D_i is a submatrix from a single column block and column operations are performed for each D_i independently, the polynomial degrees can always be transformed in order to satisfy (5.12), (5.13), (5.14) and (5.15). Consequently, the proof is identical when considering an $A_{[0,m]}$ constructed from different blocks. □

Interested readers can find a summary of the proof in [MBK16, p. 3048-3049].

We will now present an illustrating example of the proof of Theorem 5.1.8.

Example 5.1.9. [MBK16, Section V.D] We provide an example of the preservation of superregularity, detailing the properties of F and the key concepts in the proof of Theorem 5.1.8. Let $q = 2$, $n = 4$, $m = 1$ and α be a primitive normal element of $\mathbb{F}_{2^M}$ where

$M = q^{n(m+2)-1} = 2^{11} = 2048$ satisfies Theorem 5.1.8. Then,

$$T = \begin{bmatrix} 0 & T_0 \\ T_0, & T_1 \end{bmatrix}$$

$$= \begin{bmatrix} & & & & \alpha^{[0]} & \alpha^{[1]} & \alpha^{[2]} & \alpha^{[3]} \\ & & & & \alpha^{[1]} & \alpha^{[2]} & \alpha^{[3]} & \alpha^{[4]} \\ & & & & \alpha^{[2]} & \alpha^{[3]} & \alpha^{[4]} & \alpha^{[5]} \\ & & & & \alpha^{[3]} & \alpha^{[4]} & \alpha^{[5]} & \alpha^{[6]} \\ \alpha^{[0]} & \alpha^{[1]} & \alpha^{[2]} & \alpha^{[3]} & \alpha^{[4]} & \alpha^{[5]} & \alpha^{[6]} & \alpha^{[7]} \\ \alpha^{[1]} & \alpha^{[2]} & \alpha^{[3]} & \alpha^{[4]} & \alpha^{[5]} & \alpha^{[6]} & \alpha^{[7]} & \alpha^{[8]} \\ \alpha^{[2]} & \alpha^{[3]} & \alpha^{[4]} & \alpha^{[5]} & \alpha^{[6]} & \alpha^{[7]} & \alpha^{[8]} & \alpha^{[9]} \\ \alpha^{[3]} & \alpha^{[4]} & \alpha^{[5]} & \alpha^{[6]} & \alpha^{[7]} & \alpha^{[8]} & \alpha^{[9]} & \alpha^{[10]} \end{bmatrix}$$

is a superregular matrix. Let A_0 and A_1 be the following two non-singular square matrices:

$$A_0 = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad A_1 = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

We use $A_{[0,1]} = \text{diag}(A_0, A_1)$ as the block diagonal matrix and generate the product F:

$$F = TA_{[0,1]}$$

$$= \begin{bmatrix} 0 & 0 & 0 & 0 & \alpha^{[0]} + \alpha^{[1]} & \alpha^{[1]} + \alpha^{[2]} + \alpha^{[3]} & \alpha^{[1]} & \alpha^{[0]} + \alpha^{[2]} \\ 0 & 0 & 0 & 0 & \alpha^{[1]} + \alpha^{[2]} & \alpha^{[2]} + \alpha^{[3]} + \alpha^{[4]} & \alpha^{[2]} & \alpha^{[1]} + \alpha^{[3]} \\ 0 & 0 & 0 & 0 & \alpha^{[2]} + \alpha^{[3]} & \alpha^{[3]} + \alpha^{[4]} + \alpha^{[5]} & \alpha^{[3]} & \alpha^{[2]} + \alpha^{[4]} \\ 0 & 0 & 0 & 0 & \alpha^{[3]} + \alpha^{[4]} & \alpha^{[4]} + \alpha^{[5]} + \alpha^{[6]} & \alpha^{[4]} & \alpha^{[3]} + \alpha^{[5]} \\ \alpha^{[1]} + \alpha^{[2]} & \alpha^{[0]} & \alpha^{[0]} + \alpha^{[2]} & \alpha^{[3]} & \alpha^{[4]} + \alpha^{[5]} & \alpha^{[5]} + \alpha^{[6]} + \alpha^{[7]} & \alpha^{[5]} & \alpha^{[4]} + \alpha^{[6]} \\ \alpha^{[2]} + \alpha^{[3]} & \alpha^{[1]} & \alpha^{[1]} + \alpha^{[3]} & \alpha^{[4]} & \alpha^{[5]} + \alpha^{[6]} & \alpha^{[6]} + \alpha^{[7]} + \alpha^{[8]} & \alpha^{[6]} & \alpha^{[5]} + \alpha^{[7]} \\ \alpha^{[3]} + \alpha^{[4]} & \alpha^{[2]} & \alpha^{[2]} + \alpha^{[4]} & \alpha^{[5]} & \alpha^{[6]} + \alpha^{[7]} & \alpha^{[7]} + \alpha^{[8]} + \alpha^{[9]} & \alpha^{[7]} & \alpha^{[6]} + \alpha^{[8]} \\ \alpha^{[4]} + \alpha^{[5]} & \alpha^{[3]} & \alpha^{[3]} + \alpha^{[5]} & \alpha^{[6]} & \alpha^{[7]} + \alpha^{[8]} & \alpha^{[8]} + \alpha^{[9]} + \alpha^{[10]} & \alpha^{[8]} & \alpha^{[7]} + \alpha^{[9]} \end{bmatrix}$$

Because $A_0 \neq A_1$, the matrix F is not a Hankel matrix. However, note that every block matrix posses a Moore structure. The elements in every row of a given block are Frobenius powers of the corresponding elements in the first row. Since α is a normal element in $F_{2^{2048}}$ and A_0 and A_1 have full rank, the isomorphism ϕ implies that the entries generating the left column block, $\alpha^{[1]} + \alpha^{[2]}, \alpha^{[0]}, \alpha^{[0]} + \alpha^{[2]}, \alpha^{[3]}$, are linearly independent polynomials. Similarly, the entries generating the right column block, $\alpha^{[0]} + \alpha^{[1]}, \alpha^{[1]} + \alpha^{[2]} + \alpha^{[3]}, \alpha^{[1]}, \alpha^{[0]} + \alpha^{[2]}$, are also respectively linearly independent amongst each other. Note that the q-degree of these polynomials are upper and lower bounded between 3 and 0 which satisfies (5.21).
Furthermore, the q-degrees are sorted according to Property 3. On a given column of F, the q-degrees are increasing by 1 from top to bottom. Moreover, for any fixed row, any polynomial in the left block has a lower degree than any polynomial in the right block.
Now, consider the submatrix formed from rows r_i, $i \in \{1, 2, 4, 5\}$ and columns c_j, $j \in \{3, 4, 5, 6\}$. This matrix

$$D = \begin{bmatrix} 0 & \alpha^{[1]} + \alpha^{[2]} & \alpha^{[2]} + \alpha^{[3]} + \alpha^{[4]} & \alpha^{[2]} \\ 0 & \alpha^{[2]} + \alpha^{[3]} & \alpha^{[3]} + \alpha^{[4]} + \alpha^{[5]} & \alpha^{[3]} \\ \alpha^{[3]} & \alpha^{[4]} + \alpha^{[5]} & \alpha^{[5]} + \alpha^{[6]} + \alpha^{[7]} & \alpha^{[5]} \\ \alpha^{[4]} & \alpha^{[5]} + \alpha^{[6]} & \alpha^{[6]} + \alpha^{[7]} + \alpha^{[8]} & \alpha^{[6]} \end{bmatrix}$$

does not have increasing q-degrees along the rows, i.e., (5.13) is not satisfied. The degree of the polynomials in the second and fourth columns are equal and lower than the degree of the polynomials in the third column. The degrees can be sorted by subtracting the fourth column from the second and then swapping the third and the fourth columns. The transformed matrix

$$\hat{D} = \begin{bmatrix} 0 & \alpha^{[1]} & \alpha^{[2]} & \alpha^{[2]} + \alpha^{[3]} + \alpha^{[4]} \\ 0 & \alpha^{[2]} & \alpha^{[3]} & \alpha^{[3]} + \alpha^{[4]} + \alpha^{[5]} \\ \alpha^{[0]} & \alpha^{[4]} & \alpha^{[5]} & \alpha^{[5]} + \alpha^{[6]} + \alpha^{[7]} \\ \alpha^{[10]} & \alpha^{[5]} & \alpha^{[6]} & \alpha^{[6]} + \alpha^{[7]} + \alpha^{[8]} \end{bmatrix}$$

completely satisfies (5.12), (5.13), (5.14) and (5.15). The above column operations are equivalent to right multiplication with the matrix

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

which has full rank. It follows that $\hat{D}$ is non-singular and therefore, D is non-singular as well.

Theorem 5.1.10. (Construction of an m-MSR Code) [MBK16, Theorem 6] We construct a memory maximum sum rank code with memory μ. Let $\bar{T}$ be the superregular matrix as in (5.16) but now generated using a primitive *normal* $\alpha \in \mathbb{F}_{q^M}$, where $M = q^{n(\mu+2)-1}$. Let $0 \leq i_1 < \cdots < i_k < n$ and construct a $k(\mu+1) \times n(\mu+1)$ submatrix $G_{trunc}(\mu)$ of $\bar{T}$ from rows indexed $jn + i_1, \ldots, jn + i_k$ for $0 \leq j \leq \mu$. This matrix is the truncated sliding generator matrix of an m-MSR convolutional code $\mathcal{C}[n, k, \mu]$.

Proof. We show that $G_{trunc}(\mu)$ satisfies Theorem 5.1.5. Assume w.l.o.g. that $i_1 = 0, \ldots, i_k = k - 1$. Each T_i is divided into

$$T_i = \begin{pmatrix} G_i \\ T_i' \end{pmatrix},$$

where $G_i \in \mathbb{F}_{q^M}^{k \times n}$ are the blocks of the truncated generator matrix. For $0 \leq t \leq \mu$, let $A_t \in \mathbb{F}_q^{n \times n}$ be a non-singular matrix. We similarly divide $A_t = \begin{pmatrix} A_t^* & A_t' \end{pmatrix}$ where the two blocks $A_t^* \in \mathbb{F}_q^{n \times p_t}$ and $A_t' \in \mathbb{F}_q^{n \times (n-p_t)}$ represent the reduced channel matrix and some remaining matrix respectively. Let $A_{[0,\mu]} = \text{diag}(A_0, \ldots, A_\mu)$. The product can be written as

$$\bar{T}A_{[0,\mu]} = \begin{bmatrix} T_0A_0 & T_1A_1 & \ldots & T_\mu A_\mu \\ & T_0A_1 & \ldots & T_{\mu-1}A_\mu \\ & & \ddots & \vdots \\ & & & T_0A_\mu \end{bmatrix},$$

where

$$T_iA_t = \begin{bmatrix} G_iA_t^* & G_iA_t' \\ T_i'A_t^* & T_i'A_t' \end{bmatrix}.$$

The submatrix of $\bar{T}A_{[0,\mu]}$ containing only the rows and columns involving $G_iA_t^*$ is equal to the product $G_{trunc}(\mu)A_{[0,\mu]}^*$. $\bar{T}A_{[0,\mu]}$ is superregular as shown in Theorem 5.1.8. Then, the determinant of $G_{trunc}(\mu)A_{[0,\mu]}^*$ is either trivially zero or it is non-trivial and therefore non-zero.

It can be shown that for all $A^*_{[0,\mu]}$ whose blocks satisfy (5.2) of Theorem 5.1.5, the product

$$G_{trunc}(\mu)A^*_{[0,\mu]} = \begin{bmatrix} G_0A_0^* & G_1A_1^* & \dots & G_\mu A_\mu^* \\ & G_0A_1^* & \dots & G_{\mu-1}A_\mu^* \\ & & \ddots & \vdots \\ & & & G_0A_\mu^* \end{bmatrix}$$

has a non-trivial determinant. Note that the structure is reversed from that of D in (5.28). Almeida et al. [ANP13, p. 2151] showed that for D to be a non-trivial minor, the number of rows of each D_j block cannot be less than the number of columns of each O_j block. In this case, each D_j block of $G_{trunc}(\mu)A^*_{[0,\mu]}$ contains $k(j+1)$ rows and each O_j block contains $\sum_{t=0}^{j} p_t$ columns. Consequently, if the condition in (5.2) of Theorem 5.1.5 is satisfied, then $k(j+1) \geq \sum_{t=0}^{j} p_t$ for $j \leq \mu$ implies that $G_{trunc}(\mu)A^*_{[0,\mu]}$ has a non-trivial determinant and is therefore non-singular. Thus, $G_{trunc}(\mu)$ satisfies Theorem 5.1.5 and $\mathcal{C}[n,k,\mu]$ achieves $d^{\mu}_{SR}(\mathcal{C}) = (n-k)(\mu+1)+1$. □

Remark 5.1.11. [MBK16, p. 3043] We have an m-MSR code $\mathcal{C}$ with memory $T = W-1$. This implies:

$$d^{W-1}_{SR}(\mathcal{C}) = (n-k)(W-1+1)+1 = (n-k)W+1 \tag{5.29}$$

By Theorem 4.3.4:

$$\begin{aligned} s_0 \text{ recoverable} &\Leftrightarrow d^{W-1}_{SR}(\mathcal{C}) > nW - \sum_{i=0}^{W-1} p_t \\ &\Leftrightarrow \sum_{i=0}^{W-1} p_t > nW - d^{W-1}_{SR}(\mathcal{C}) \end{aligned}$$

We use the Equation 5.29 above and obtain:

$$\sum_{i=0}^{W-1} p_t > nW - d^{W-1}_{SR}(\mathcal{C}) = nW - (n-k)W - 1 = k(W) - 1 = k(T+1) - 1$$

Therefore, an m-MSR code with memory $T = W-1$ recovers the source packet s_0 with delay T, when the rank of the channel matrix is at least $k(T+1)$ in each sliding window of length W.

Remark 5.1.12. [MBK16, p. 3044] Several theorems and proofs have been presented in this section. However, it also communicates a strategy to decode in a sliding window network model. We can reduce the decoding strategy to matrix inversion. Assume we have an m-MSR code with memory $T = W-1$ and all source packets before time t have been recovered. We would like to recover the packet s_t within the deadline of time $t+T$. To do that, we consider increasingly larger windows $[t, t+j]$ for $0 \leq j \leq T$. Theorem 4.3.4 states that if $\sum_{i=t}^{t+j} p_i < k(j+1)$, then the decoder cannot guarantee a recovery by time $t+j$. The window length j is incremented at each time instance up to the first point where $\sum_{i=t}^{t+j} p_i \geq k(j+1)$ is achieved. By Theorem 5.1.5, the rank condition in (5.2) is satisfied and $G_{trunc}(j)A^*_{[t,t+j]}$ is non-singular. Therefore, we invert this matrix and solve for $s_{[t,t+j]}$. Consequently, all packets can be recovered in the window simultaneously.

5.1.2 Numerical Results

Mahmood et al. [MBK16, p. 3050] emphasise that the bound given for $M \geq q^{n(\mu+2)-1}$ in Theorem 5.1.8 is only a sufficiency constraint. For small code parameters, it is possible that Theorem 5.1.5 also holds for a smaller M. Mahmood et al. present examples that underline this statement.

Example 5.1.13. [MBK16, Example 3] Let $\alpha = X + 1$ be a primitive normal element in $\mathbb{F}_{2^{11}} = \mathbb{F}_2/(X^{11} + X^2 + 1)$. The truncated generator matrix of length 3 is given by

$$G_{trunc}(2) = \begin{bmatrix} \alpha & \alpha^{[1]} & \alpha^{[2]} & \alpha^{[3]} & \alpha^{[4]} & \alpha^{[5]} & \alpha^{[6]} & \alpha^{[7]} & \alpha^{[8]} \\ \alpha^{[2]} & \alpha^{[3]} & \alpha^{[4]} & \alpha^{[5]} & \alpha^{[6]} & \alpha^{[7]} & \alpha^{[8]} & \alpha^{[9]} & \alpha^{[10]} \\ & & & \alpha & \alpha^{[1]} & \alpha^{[2]} & \alpha^{[3]} & \alpha^{[4]} & \alpha^{[5]} \\ & & & \alpha^{[2]} & \alpha^{[3]} & \alpha^{[4]} & \alpha^{[5]} & \alpha^{[6]} & \alpha^{[7]} \\ & & & & & & \alpha & \alpha^{[1]} & \alpha^{[2]} \\ & & & & & & \alpha^{[2]} & \alpha^{[3]} & \alpha^{[4]} \end{bmatrix}.$$

One can check that this matrix satisfies Theorem 5.1.5, making this matrix the generator for an m-MSR code $\mathcal{C}[3, 2, 2]$ where $n = 3$, $k = 2$, $\mu = m = 2$. Theorem 5.1.10 guarantees a code with the same parameters for $M = q^{n(m+2)-1} = 2^{3(4)-1} = 2^{11} = 2048$, i.e., over $\mathbb{F}_{2^{2048}}$. This field is much bigger than the one we use in this example.

Table 5.1 shows more examples of codes which are achievable over a lower field than required in Theorem 5.1.10.

Parameters $[n, k, \mu]$	Field Bound in Theorem 5.1.10	Achievable Field
$[3, 2, 2]$	$\mathbb{F}_{2^{2048}}$	$\mathbb{F}_{2^{11}} = \mathbb{F}_2/(X^{11} + X^2 + 1)$
$[4, 2, 1]$	$\mathbb{F}_{2^{2048}}$	$\mathbb{F}_{2^{11}} = \mathbb{F}_2/(X^{11} + X^2 + 1)$
$[3, 1, 2]$	$\mathbb{F}_{2^{2048}}$	$\mathbb{F}_{2^{11}} = \mathbb{F}_2/(X^{11} + X^2 + 1)$
$[2, 1, 2]$	$\mathbb{F}_{2^{128}}$	$\mathbb{F}_{2^7} = \mathbb{F}_2/(X^7 + X^3 + 1)$
$[2, 1, 1]$	$\mathbb{F}_{2^{32}}$	$\mathbb{F}_{2^5} = \mathbb{F}_2/(X^5 + X^2 + 1)$

TABLE 5.1: Achievable code constructions by Mahmood et al. over $\mathbb{F}_2$. [MBK16, Table 1]

Mahmood [Mah15, p. 27] presents an alternative to the structured construction in his thesis. Here, we randomly generate blocks G_j for $j \leq \mu$ and test whether the truncated generator matrix satisfies the Theorem 5.1.5. If the generator matrix satisfies the theorem, then the code naturally achieves the maximum μ-th column sum rank distance. Random linear m-MSR codes exist for significantly smaller fields than the sufficiency statement in the construction in Theorem 5.1.10 requires. Mahmood shows that the probability of finding random truncated generator matrices, which satisfy Theorem 5.1.5, is relatively high (for more details see [Mah15, p. 27-28]). The probability increases if we generate a matrix over a bigger ground field $\mathbb{F}_{q^M}$. Sometimes, it can be a good strategy to search *nearly m-MSR* codes. These codes have generator matrices that fulfil the statement 2) in Theorem 5.1.5 for almost all channel matrices. Although they are not capable of perfect recovery, codes that produce full rank products for almost every channel matrix can be used with a small error in practical scenarios.

5.1.3 Reducing the Field Size

We have already seen that the field sizes of Theorem 5.1.10 are high. For that reason, it is not practical for every use case. Almeida et al. [AMPN20, Section 4] constructed some systematic m-MSR codes that need a smaller field size compared to Theorem 5.1.10.

Parameters $[n, k, \mu]$	Achievable field	Bound in Theorem 5.1.10
$[3, 2, 2]$	$\mathbb{F}_{2^7}$	$\mathbb{F}_{2^{2048}}$
$[4, 2, 1]$	$\mathbb{F}_{2^6}$	$\mathbb{F}_{2^{2048}}$
$[3, 1, 2]$	$\mathbb{F}_{2^9}$	$\mathbb{F}_{2^{2048}}$
$[2, 1, 2]$	$\mathbb{F}_{2^3}$	$\mathbb{F}_{2^{128}}$
$[2, 1, 1]$	$\mathbb{F}_{2^2}$	$\mathbb{F}_{2^{32}}$

TABLE 5.2: Achievable code constructions by Almeida et al. over $\mathbb{F}_2$. [AMPN20, Table 1]

Table 5.2 shows the achieved field sizes for the systematic m-MSR codes. The constructions were verified by the following theorem:

Theorem 5.1.14. [AMPN20, Theorem 6] Let $\mathcal{C}$ be an $[n, k]$ convolutional code with memory μ and a systematic encoder $G(D) = [I_k \ P(D)]$, where $P(D) = \sum_{i=0}^{\mu} P_i D^i \in \mathbb{F}_{q^M}^{k\times(n-k)}[D]$ and let $0 \leq j \leq \mu$. Given $\underline{A}_l \in \mathbb{F}_q^{(n-k)\times(n-k)}$, $B_l \in \mathbb{F}_q^{k\times k}$ and $C_l \in \mathbb{F}_q^{k\times(n-k)}$ with $l = 0, 1 \ldots, j$, consider the matrix $T_j = T_j((\underline{A}_l, B_l, C_l)_{l=0}^{j})$ defined by

$$T_j = \begin{bmatrix} B_0 & & & \\ & B_1 & & \\ & & \ddots & \\ & & & B_j \end{bmatrix} \begin{bmatrix} P_0 & P_1 & \ldots & P_j \\ & P_0 & \ldots & P_{j-1} \\ & & \ddots & \vdots \\ & & & P_0 \end{bmatrix} \begin{bmatrix} \underline{A}_0 & \underline{A}_1 & \ldots & \underline{A}_j \\ & \underline{A}_0 & \ldots & \underline{A}_{j-1} \\ & & \ddots & \vdots \\ & & & \underline{A}_0 \end{bmatrix} + \begin{bmatrix} C_0 & & & \\ & C_1 & & \\ & & \ddots & \\ & & & C_j \end{bmatrix}$$

$$= \begin{bmatrix} T_{0,0} & T_{0,1} & \ldots & T_{0,j} \\ & T_{1,1} & & \vdots \\ & & \ddots & T_{j-1,j} \\ & & & T_{j,j} \end{bmatrix}.$$

The following statements are equivalent:

1) $d_{SR}^{j}(\mathcal{C}) = (j+1)(n-k)+1$

2) Every square submatrix of T_j with all its diagonal entries in the matrices $T_{s,t}$, where $s, t \in \{0, 1, \ldots, j\}$, is non-singular, for all $C_l \in \mathbb{F}_q^{k\times(n-k)}$ and all non-singular upper triangular matrices $B_l \in \mathbb{F}_q^{k\times k}$, $\underline{A}_l \in \mathbb{F}_q^{(n-k)\times(n-k)}$, $l = 0, 1, \ldots, j$.

3) T_j is superregular for all $C_l \in \mathbb{F}_q^{k\times(n-k)}$ and all non-singular upper triangular matrices $B_l \in \mathbb{F}_q^{k\times k}$, $\underline{A}_l \in \mathbb{F}_q^{(n-k)\times(n-k)}$, $l = 0, 1, \ldots, j$.

Conjecture 5.1.15. [AMPN20, p. 18-19] Let α be a primitive element of a finite field $\mathbb{F}_{q^M}$ with q^M elements and consider $G(D) = [I_k \ P(D)]$ with $P(D) = \sum_{i=0}^{\mu} P_i D^i$ where P_i, for

$0 \leq i \leq \mu$, is equal to

$$P_i = \begin{bmatrix} \alpha^{[Ri]} & \alpha^{[Ri+1]} & \dots & \alpha^{[Ri+n-k-1]} \\ \alpha^{[Ri+1]} & \alpha^{[Ri+2]} & \dots & \alpha^{[Ri+n-k]} \\ \vdots & \vdots & & \vdots \\ \alpha^{[Ri+k-1]} & \alpha^{[Ri+k]} & \dots & \alpha^{[Ri+n-2]} \end{bmatrix} \in \mathbb{F}_{q^M}^{k\times(n-k)}$$

with $R = \max\{k, n-k\}$ and $M \geq q^{R\mu+n-1}$. We use the usual notation $\alpha^{[j]} = \alpha^{q^j}$ to denote the j-th Frobenius power. The matrix

$$P_j^c = \begin{bmatrix} P_0 & P_1 & \dots & P_j \\ & P_0 & \dots & P_{j-1} \\ & & \ddots & \vdots \\ & & & P_0 \end{bmatrix}$$

is superregular for all $j \leq \mu$ if the field size is sufficiently large.

This statement is a conjecture and not proven yet. Nevertheless, Almeida et al. [AMPN20] conjecture further that for specific parameters $[n, k, \mu]$ and field sizes this generator matrix satisfies the conditions of Theorem 5.1.14. Hence it can be used to construct systematic m-MSR convolutional codes with memory μ.

5.2 Construction of Maximum Rank Profile Codes

MRP codes, defined in Definition 2.2.27, are useful for the network model with time delay as described in Section 4.3.2. In this framework, MRP codes are equipped with good decoding properties. For this reason, we would like to construct maximum rank profile codes.
This section follows the paper of Almeida et al. [AN21, Section 5].

Theorem 5.2.1. [AN21, Theorem 6] Let $\mathcal{C}$ be an (n, k, δ) rank metric convolutional code with generator matrix $G(D)$. The following statements are equivalent:

1) $d_{CR}^{j}(\mathcal{C}) = d_R(\mathcal{C}_j^c) = (j+1)(n-k)+1$ ($\mathcal{C}_j^c$ is defined in Definition 2.2.22)

2) Every $(j+1)k \times (j+1)k$ full-size minor of $G_{trunc}(j)A_{[0,j]}$ formed by the columns with indices $1 \leq c_1 < \cdots < c_{(j+1)k}$ where $c_{sk+1} > sn$ for $s = 1, \ldots, j$, is non-zero for all $A_{[0,j]} \in \mathbb{F}_q^{n(j+1)\times n(j+1)}$ non-singular block upper triangular channel matrices with entries in the base field $\mathbb{F}_q$.

Proof. ($1 \Rightarrow 2$) Proof by contradiction: Suppose that 2) does not hold. For that, let $A_{[0,j]}$ be the block upper triangular matrix such that $G_{trunc}(j)A_{[0,j]}$ contains a singular square submatrix $\tilde{G}$ of size $(j+1)k$ formed by columns of $G_{trunc}(j)A_{[0,j]}$ with indices $c_1 < c_2 < \cdots < c_{(j+1)k}$, $c_{sk+1} > sn$ for $s = 1, \ldots j$. Take $(u_0, \ldots, u_j) \neq 0$ such that $(u_0, \ldots, u_j)\tilde{G} = 0$. Such a $(u_0, \ldots, u_j) \neq 0$ exists since $\tilde{G}$ is singular. Then,

$$\operatorname{rank}(\phi((u_0, \ldots, u_j)G_{trunc}(j)A_{[0,j]})) \leq n(j+1) - k(j+1) < (n-k)(j+1)+1$$

as at least $k(j+1)$ columns go to 0 by the definition of $(u_0, \ldots, u_j)$. If $u_0 \neq 0$, then $d_{CR}^{j}(\mathcal{C}) < (j+1)(n-k)+1$ contradicting 1). Assume then that $u_0 = 0$ and consider $i > 0$ such that $u_i \neq 0$ and $u_l = 0$ for $0 \leq l < i \leq j$. The last $n(j-i)$ rows and columns of $A_{[0,j]}$ are again a non-singular upper block triangular matrix. Then, due to the Toeplitz structure of $G_{trunc}(j)$ and that $c_{sk+1} > sn$ for $s = 1, 2, \ldots, j$, it holds that $(u_i, \ldots, u_j)G_{trunc}(j-i)A_{[i,j]}$ has weight (and therefore rank) strictly less than $(j-i+1)(n-k)+1$. This implies that $d_{CR}^{j-i}(\mathcal{C}) \leq (j-i+1)(n-k)$ and by Lemma 2.2.25 it follows $d_{CR}^{j}(\mathcal{C}) \leq (j+1)(n-k)$ which again contradicts 1).

($2 \Rightarrow 1$) Proof by contradiction: Suppose that $d_{CR}^{j}(\mathcal{C}) \leq (j+1)(n-k)$ and let l be the first instant such that the upper bound $d_{CR}^{i}(\mathcal{C}) = (i+1)(n-k)+1$ is not achieved. Hence, there exists a $(u_0, \ldots, u_l)$, $u_0 \neq 0$ and an $\bar{A}_{[0,l]}$ non-singular upper triangular matrix such that $(v_0, \ldots, v_l) = (u_0, \ldots, u_l)G_{trunc}(l)\bar{A}_{[0,l]}$ with

$$\operatorname{rank}(\phi((v_0, \ldots, v_l))) \leq (n-k)(l+1).$$

This means that there exists an $\hat{A}_{[0,l]}$ non-singular upper triangular matrix such that $(v_0, \ldots, v_l)\hat{A}_{[0,l]}$ has at least $k(l+1)$ zeros. Let $A_{[0,l]} = \bar{A}_{[0,l]}\hat{A}_{[0,l]}$ and build a square matrix L by selecting the columns of $G_{trunc}(l)A_{[0,l]}$ where $(u_0, \ldots, u_l)G_{trunc}(l)A_{[0,l]}$ is zero and denote these indices by $c_1, \ldots, c_{(l+1)k}$. At this point, we observe that, as $d_{CR}^{i}(\mathcal{C}) = (i+1)(n-k)+1$ for $i = 0, 1, 2, \ldots, l-1$, it follows that $(u_0, \ldots, u_i)G_{trunc}(i)A_{[0,i]}$ has at most $k(i+1)-1$ zeros for $i = 0, 1, \ldots, l-1$. This, in particular, implies that $c_{sk+1} > sn$ for $s = 1, 2, \ldots, l$. Extend the square matrix L by taking the matrix formed by the last $k(j-l)$ columns of $G_{trunc}(j)A_{[0,j]}$, denoted by J, and build a $k(j+1) \times k(j+1)$ submatrix of $G_{trunc}(j)A_{[0,j]}$ of the form

$$K = \left[\begin{array}{c|c} L & \\ \hline 0 & \mathrm{J} \end{array}\right].$$

We call the indices of the selected columns $c_1, \ldots, c_{(j+1)k}$. Clearly, the indices satisfy $c_{sk+1} > sn$, for $s = 1, 2, \ldots, j$ and K has a determinant equal to zero as L is singular by construction. This means that statement 2) does not hold which concludes the proof. □

Theorem 5.2.2. [AN21, Theorem 8] Let $\mathcal{C} = \mathrm{im}(G(D)) \subset F_{q^M}^n$ be an (n, k, δ) convolutional code with $L = \left\lfloor \frac{\delta}{k} \right\rfloor + \left\lfloor \frac{\delta}{n-k} \right\rfloor$ as in (2.4) and the entries of the coefficients G_i of $G(D)$ are all non-zero. If $G_{trunc}(L)A_{[0,L]}$ is superregular for all $A_{[0,L]} \in \mathbb{F}_q^{n(L+1)\times n(L+1)}$ non-singular block upper triangular channel matrices as in the Theorem 5.2.1 above with $j = L$, then $\mathcal{C}$ is MRP.

Proof. We look at the full-size minors of $G_{trunc}(L)A_{[0,L]}$ with the column condition in Theorem 5.2.1. Since the entries of the coefficients G_i of $G(D)$ are all non-zero, these full-size minors are non-trivial. By Lemma 2.2.25, it is enough to consider $j = L$. Finally, the non-trivial full-size minors of $G_{trunc}(L)A_{[0,L]}$ are non-zero as $G_{trunc}(L)A_{[0,L]}$ is assumed to be superregular. With Theorem 5.2.1 follows that $\mathcal{C}$ is MRP. □

Theorem 5.2.3. [AN21, Theorem 10] Let $L = \left\lfloor \frac{\delta}{k} \right\rfloor + \left\lfloor \frac{\delta}{n-k} \right\rfloor$, $n \in \mathbb{N}$, α be a primitive and normal element of a finite field $\mathbb{F}_{q^M}$ of characteristic p and consider $G_{trunc}(L) \in \mathbb{F}_{q^M}^{(L+1)k\times(L+1)n}$. The G_i submatrices are of the form

$$\begin{bmatrix} \alpha^{[ni]} & \alpha^{[ni+1]} & \ldots & \alpha^{[n(i+1)-1]} \\ \alpha^{[ni+1]} & \alpha^{[ni+2]} & \ldots & \alpha^{[n(i+1)]} \\ \alpha^{[ni+2]} & \alpha^{[ni+3]} & \ldots & \alpha^{[n(i+1)+1]} \\ \vdots & \vdots & & \vdots \\ \alpha^{[ni+k-1]} & \alpha^{[ni+k]} & \ldots & \alpha^{[n(i+1)+k-2} \end{bmatrix}, \tag{5.30}$$

where we use the notation $\alpha^{[j]} = \alpha^{q^j}$ to denote the j-th Frobenius power. If $M \geq q^{n(L+2)-1}$, then the matrix $G_{trunc}(L)$ is superregular over $\mathbb{F}_{q^M}$.

Proof.

$$G_{trunc}(L) = \begin{bmatrix} G_0 & G_1 & \ldots & G_L \\ & G_0 & \ldots & G_{L-1} \\ & & \ddots & \vdots \\ & & & G_0 \end{bmatrix} \in \mathbb{F}_q^{k(L+1)\times n(L+1)}$$

We observe that the permutation of rows does not affect superregularity. So, we can adapt Theorem 5.1.6 for $k \times n$ matrices. The proof works the same way and the result follows immediately. □

Theorem 5.2.4. [AN21, Theorem 12] Let $\mathcal{C}$ be an (n, k, δ) convolutional code with generator matrix $G(D) = \sum_{i=0}^{v} G_i D^i$, where each G_i is a $k \times n$ matrix given by (5.30). Let $M \geq q^{n(L+2)-1}$ and let α be a primitive and normal element of $\mathbb{F}_{q^M}$. Furthermore, let $A_{[0,L]}$ be the non-singular block upper triangular matrix over $\mathbb{F}_q^{n(L+1)\times n(L+1)}$. Then

$$S = G_{trunc}(L)A_{[0,L]} \text{ is a superregular matrix,}$$

and therefore by Theorem 5.2.2

$$\mathcal{C} \text{ is an MRP convolutional code.}$$

Proof. We will use here the same procedure as in the proof of Theorem 5.1.8. For this reason, the schema of that proof should be recalled.

Now, we can prove the current theorem:

First, we have that $G_{trunc}(L)$ is superregular by Theorem 5.2.3.
There is

$$S = G_{trunc}(L)A_{[0,L]} = \begin{bmatrix} G_0 & G_1 & \dots & G_L \\ & G_0 & \dots & G_{L-1} \\ & & \ddots & \vdots \\ & & & G_0 \end{bmatrix} \begin{bmatrix} A_{00} & A_{01} & \dots & A_{0L} \\ & A_{11} & \dots & A_{1L} \\ & & \ddots & \vdots \\ & & & A_{LL} \end{bmatrix}$$
$$= \begin{bmatrix} S_{00} & S_{01} & \dots & S_{0L} \\ & S_{11} & \dots & S_{1L} \\ & & \ddots & \vdots \\ & & & S_{LL} \end{bmatrix}$$

where

$$S_{ij} = \sum_{l=0}^{j-i} G_l A_{(i+l)j} \,.$$

Let P be the permutation matrix corresponding to the permutation σ defined by:

$$\sigma = \begin{pmatrix} 1 & \dots & k & k+1 & \dots & 2k & \dots & kL+1 & \dots & k(L+1) \\ kL+1 & \dots & k(L+1) & k(L-1)+1 & \dots & kL & \dots & 1 & \dots & k \end{pmatrix}$$

σ maps the first k elements to the last k elements. Then, the second set of k elements to the second last k elements and so on. It follows that S is superregular if and only if PS is superregular. Define

$$F = PS = \begin{bmatrix} & & & F_{0L} \\ & & F_{0(L-1)} & F_{1L} \\ & \iddots & \vdots & \vdots \\ F_{00} & \dots & F_{(L-1)(L-1)} & F_{LL} \end{bmatrix}$$

where $F_{ij} = S_{(j-i),j}$.
Each element of F_{ij} is the value of a linearised polynomial evaluated at α. Moreover, if the first row of F_{ij} is $(f_{ij0}, \dots, f_{ij(n-1)})$, then for $s \leq k$, the s-th row is $(f_{ij0}^{[s]}, \dots, f_{ij(n-1)}^{[s]})$. So, we obtain similar properties as in Schema I(b). The corresponding property in Schema I(c) is not valid, though. In our case, if $i' < i$, the entries of a row s of F_{ij} are not q-powers of the corresponding entries in row s of $F_{i'j}$ since our F_{ij}, instead of being just the product of two matrices, is the sum of products of matrices. So for $i \neq i'$, the linearised polynomials from which we construct the entries of F_{ij} may be completely different from the ones we use to construct the entries of $F_{i'j}$. This is also the reason why the exponents of the linearised polynomials used to obtain the entries of F_{ij} do not depend on i. This is in contrast to what happened above in the proof of Theorem 5.1.8. Hence, we have

$$F_{ij} = \begin{bmatrix} f_{ij0}^{[0]} & f_{ij1}^{[0]} & \dots & f_{ij(n-1)}^{[0]]} \\ f_{ij0}^{[1]} & f_{ij1}^{[1]} & \dots & f_{ij(n-1)}^{[1]} \\ f_{ij0}^{[2]} & f_{ij1}^{[2]} & \dots & f_{ij(n-1)}^{[2]} \\ \vdots & \vdots & \ddots & \vdots \\ f_{ij0}^{[k-1]} & f_{ij1}^{[k-1]} & \dots & f_{ij(n-1)}^{[k-1]} \end{bmatrix}.$$

Also, α is normal over $\mathbb{F}_q$ and each A_{ii} is invertible, for $1 \leq i \leq n$, then it holds that $(f_{ij0}, f_{ij1}, \ldots, f_{ij(n-1)})$ are linearly independent over $\mathbb{F}_q$, which corresponds to property I(a) in the schema. The q-degrees of the polynomials associated with the elements of F_{ij} strictly increase downwards on any fixed column, for any $0 \leq i \leq j \leq L$. So we get Property II) in the schema. It is also easy to see that the properties corresponding to III), IV) and V) in the schema are also satisfied. Hence, F is superregular. The last statement follows from Theorem 5.2.2, as already stated. That finishes the proof. □

We will now present an illustrating example of this proof.

Example 5.2.5. [AN21, Example 3] Let $k = 3$, $n = 4$, $q = 2$ and $\delta = 1$. Let $M = 2048$ and α be a primitive and normal element of $\mathbb{F}_{2^M}$. Then, $L = 1$ and

$$G_{trunc}(1) = \begin{bmatrix} \alpha^{[0]} & \alpha^{[1]} & \alpha^{[2]} & \alpha^{[3]} & \alpha^{[4]} & \alpha^{[5]} & \alpha^{[6]} & \alpha^{[7]} \\ \alpha^{[1]} & \alpha^{[2]} & \alpha^{[3]} & \alpha^{[4]} & \alpha^{[5]} & \alpha^{[6]} & \alpha^{[7]} & \alpha^{[8]} \\ \alpha^{[2]} & \alpha^{[3]} & \alpha^{[4]} & \alpha^{[5]} & \alpha^{[6]} & \alpha^{[7]} & \alpha^{[8]} & \alpha^{[9]} \\ 0 & 0 & 0 & 0 & \alpha^{[0]} & \alpha^{[1]} & \alpha^{[2]} & \alpha^{[3]} \\ 0 & 0 & 0 & 0 & \alpha^{[1]} & \alpha^{[2]} & \alpha^{[3]} & \alpha^{[4]} \\ 0 & 0 & 0 & 0 & \alpha^{[2]} & \alpha^{[3]} & \alpha^{[4]} & \alpha^{[5]} \end{bmatrix}.$$

Take

$$A_{[0,1]} = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}.$$

We have $G_{trunc}(1)A_{[0,1]} =$

$$\begin{bmatrix} \alpha^{[1]}+\alpha^{[2]} & \alpha^{[0]} & \alpha^{[0]}+\alpha^{[2]} & \alpha^{[3]} & \alpha^{[0]}+\alpha^{[2]}+\alpha^{[4]}+\alpha^{[5]} & \alpha^{[1]}+\alpha^{[2]}+\alpha^{[5]}+\alpha^{[6]}+\alpha^{[7]} & \alpha^{[0]}+\alpha^{[1]}+\alpha^{[5]} & \alpha^{[3]}+\alpha^{[4]}+\alpha^{[6]} \\ \alpha^{[2]}+\alpha^{[3]} & \alpha^{[1]} & \alpha^{[1]}+\alpha^{[3]} & \alpha^{[4]} & \alpha^{[1]}+\alpha^{[3]}+\alpha^{[5]}+\alpha^{[6]} & \alpha^{[2]}+\alpha^{[3]}+\alpha^{[6]}+\alpha^{[7]}+\alpha^{[8]} & \alpha^{[1]}+\alpha^{[2]}+\alpha^{[6]} & \alpha^{[4]}+\alpha^{[5]}+\alpha^{[7]} \\ \alpha^{[3]}+\alpha^{[4]} & \alpha^{[2]} & \alpha^{[2]}+\alpha^{[4]} & \alpha^{[5]} & \alpha^{[2]}+\alpha^{[4]}+\alpha^{[6]}+\alpha^{[7]} & \alpha^{[3]}+\alpha^{[4]}+\alpha^{[7]}+\alpha^{[8]}+\alpha^{[9]} & \alpha^{[2]}+\alpha^{[3]}+\alpha^{[7]} & \alpha^{[5]}+\alpha^{[6]}+\alpha^{[8]} \\ 0 & 0 & 0 & 0 & \alpha^{[0]}+\alpha^{[1]} & \alpha^{[1]}+\alpha^{[2]}+\alpha^{[3]} & \alpha^{[1]} & \alpha^{[0]}+\alpha^{[2]} \\ 0 & 0 & 0 & 0 & \alpha^{[1]}+\alpha^{[2]} & \alpha^{[2]}+\alpha^{[3]}+\alpha^{[4]} & \alpha^{[2]} & \alpha^{[1]}+\alpha^{[3]} \\ 0 & 0 & 0 & 0 & \alpha^{[2]}+\alpha^{[3]} & \alpha^{[3]}+\alpha^{[4]}+\alpha^{[5]} & \alpha^{[3]} & \alpha^{[3]}+\alpha^{[4]} \end{bmatrix}.$$

Let

$$P = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

and we obtain $F = PS = PG_{trunc}(1)A_{[0,1]} =$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & \alpha^{[0]}+\alpha^{[1]} & \alpha^{[1]}+\alpha^{[2]}+\alpha^{[3]} & \alpha^{[1]} & \alpha^{[0]}+\alpha^{[2]} \\ 0 & 0 & 0 & 0 & \alpha^{[1]}+\alpha^{[2]} & \alpha^{[2]}+\alpha^{[3]}+\alpha^{[4]} & \alpha^{[2]} & \alpha^{[1]}+\alpha^{[3]} \\ 0 & 0 & 0 & 0 & \alpha^{[2]}+\alpha^{[3]} & \alpha^{[3]}+\alpha^{[4]}+\alpha^{[5]} & \alpha^{[3]} & \alpha^{[3]}+\alpha^{[4]} \\ \alpha^{[1]}+\alpha^{[2]} & \alpha^{[0]} & \alpha^{[0]}+\alpha^{[2]} & \alpha^{[3]} & \alpha^{[0]}+\alpha^{[2]}+\alpha^{[4]}+\alpha^{[5]} & \alpha^{[1]}+\alpha^{[2]}+\alpha^{[5]}+\alpha^{[6]}+\alpha^{[7]} & \alpha^{[0]}+\alpha^{[1]}+\alpha^{[5]} & \alpha^{[3]}+\alpha^{[4]}+\alpha^{[6]} \\ \alpha^{[2]}+\alpha^{[3]} & \alpha^{[1]} & \alpha^{[1]}+\alpha^{[3]} & \alpha^{[4]} & \alpha^{[1]}+\alpha^{[3]}+\alpha^{[5]}+\alpha^{[6]} & \alpha^{[2]}+\alpha^{[3]}+\alpha^{[6]}+\alpha^{[7]}+\alpha^{[8]} & \alpha^{[1]}+\alpha^{[2]}+\alpha^{[6]} & \alpha^{[4]}+\alpha^{[5]}+\alpha^{[7]} \\ \alpha^{[3]}+\alpha^{[4]} & \alpha^{[2]} & \alpha^{[2]}+\alpha^{[4]} & \alpha^{[5]} & \alpha^{[2]}+\alpha^{[4]}+\alpha^{[6]}+\alpha^{[7]} & \alpha^{[3]}+\alpha^{[4]}+\alpha^{[7]}+\alpha^{[8]}+\alpha^{[9]} & \alpha^{[2]}+\alpha^{[3]}+\alpha^{[7]} & \alpha^{[5]}+\alpha^{[6]}+\alpha^{[8]} \end{bmatrix}.$$

Now, the above properties can be easily checked.

We again observe that the large size of the finite field in Theorem 5.2.4 is a sufficient condition. However, it is possible to find MRP codes over a smaller field. Nevertheless, it is not easy to find a general statement with a lower field size. As seen in Subsection 5.1.2 for m-MSR codes, random truncated generator matrices can also be constructed with lower field sizes. Then, we use Theorem 5.2.1 to check weather the chosen matrix really constructs an MRP code. With this procedure, it should be possible to find MRP codes over a smaller finite field. Although, this needs further research.

5.3 Construction of Maximum Rank Distance Codes

This section will present another class of codes, the so-called *MRD convolutional codes*. These codes maximise the sum rank free distance. In contrast to an m-MSR code, an MRD convolutional code has no requirements on the truncated distance. We will show the underlying theory and concrete constructions of such MRD codes. This section is based on the paper of Napp et al. [NPRV17].

5.3.1 Preliminaries

Definition 5.3.1. (Map Rowvec) [NPRV17, Section I.] Let R be a ring. The map rowvec: $R^{n\times m} \to R^{nm}$ with $M \mapsto v$ is an R-isomorphism defined by $v_{mi+j} = M_{i,j}$ with $0 \leq i < n$ and $0 \leq j < m$.
The inverse of the rowvec map will be denoted by rowmat or $\text{rowmat}_{n\times m}$ folding an nm (row) vector into an $n \times m$ matrix.

We usually apply the isomorphism $\phi : \mathbb{F}_{q^m}^n \to \mathbb{F}_q^{m\times n}$ to construct rank metric codes. Napp et al. [NPRV17] present another structure to create rank metric codes. This approach will now be shown.
It is assumed in this section that $n \leq m$.

Definition 5.3.2. (Rank Metric Code) [NPRV17, Section III.] An $(n\times m, k)$ $\mathbb{F}_q$-linear *rank metric code* $\mathcal{C} \subset \mathbb{F}_q^{n\times m}$ of rate $k/nm < 1$ is the image of a monomorphism $\varphi : \mathbb{F}_q^k \to \mathbb{F}_q^{n\times m}$. We write $\varphi = \psi \circ \gamma$ as a composition of an isomorphism ψ and a monomorphism γ:

$$\begin{aligned} \varphi : \mathbb{F}_q^k \xrightarrow{\gamma} \mathbb{F}_q^{nm} \xrightarrow{\psi} \mathbb{F}_q^{n\times m} \\ u \mapsto v = uG \mapsto V = \psi(v) \end{aligned}$$

where $G \in \mathbb{F}_q^{k\times nm}$. If $\psi = \text{rowmat}_{n\times m}$, the rows of V are simply the n consecutive blocks with m elements of v.

ψ is usually set equal to the rowmat function. In that case, we define the rank weight, rank distance and free rank distance as follows:
The rank weight of a codeword $v_1 \in \mathbb{F}_q^{nm}$ is:

$$wt_{\bar{R}}(v_1) = \text{rank}(\text{rowmat}(v_1))$$

The rank distance between $v_1, v_2 \in \mathbb{F}_q^{nm}$ is:

$$d_{\bar{R}}(v_1, v_2) = \text{rank}(\text{rowmat}(v_1) - \text{rowmat}(v_2))$$

The free rank distance of $\mathcal{C} \subset \mathbb{F}_q^{nm}$ is defined as:

$$d_{\bar{R}}(\mathcal{C}) = \min\{d_{\bar{R}}(v, w) \,|\, v, w \in \mathcal{C}, v \neq w\} = \min\{\text{rank}(\text{rowmat}(v)) \,|\, v \in \mathcal{C}, v \neq 0\}$$

Example 5.3.3. This example shows a rank metric code with the new structure over $\mathbb{F}_2^{2\times 3}$, described in Definition 5.3.2. ψ is set equal to $\text{rowmat}_{n\times m}$. Let

$$G = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} \in \mathbb{F}_2^{2\times 6}$$

be the generator matrix. Then, we can calculate a codeword in the following way:

$$(1,1)\cdot G = (1,1,0,1,0,0) \implies \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

This codeword has rank 2.

Theorem 5.3.4. (Rank Distance Bound) [NPRV17, Theorem 1] The *rank distance* of an $(n \times m, k)$ linear rank metric code is upper bounded by:

$$d_{\bar{R}}(\mathcal{C}) \leq n - \left\lfloor \frac{k-1}{m} \right\rfloor = n - \left\lceil \frac{k}{m} \right\rceil + 1$$

Definition 5.3.5. (Rank Metric Convolutional Code) [NPRV17, Section IV.] A *rank metric convolutional code* $\mathcal{C} \subset \mathbb{F}_q[D]^{n\times m}$ is the image of a homomorphism $\varphi : \mathbb{F}_q[D]^k \to \mathbb{F}_q[D]^{n\times m}$. It is $\mathbb{F}_q[D]$-linear and we write $\varphi = \psi \circ \gamma$ as a composition of a monomorphism γ and an isomorphism ψ:

$$\begin{aligned} \varphi : \mathbb{F}_q[D]^k &\xrightarrow{\gamma} \mathbb{F}_q[D]^{nm} \xrightarrow{\psi} \mathbb{F}_q[D]^{n\times m} \\ u(D) &\mapsto v(D) = u(D)G(D) \mapsto V(D) = \psi(v(D)) \end{aligned}$$

where $G(D) \in \mathbb{F}_q[D]^{k\times nm}$ is a full row rank polynomial matrix, called *encoder* of $\mathcal{C}$, and we may choose, as before, $V(D) = \text{rowmat}_{n\times m}(v(D))$ such that $V_{i,j}(D) = v_{mi+j}(D)$.

In this case, the definitions of the sum rank weight, sum rank distance and free sum rank distance are given by:
The sum rank weight of a codeword $v(D) = \sum_{i\geq 0} v_i D^i \in \mathbb{F}_q[D]^{nm}$, $V(D) = \sum_{i\geq 0} V_i D^i \in \mathbb{F}_q[D]^{n\times m}$ with $V_i = \text{rowmat}(v_i)$, is:

$$wt_{\overline{SR}}(v(D)) = \sum_{i\geq 0} \text{rank}(\text{rowmat}(v_i)) = \sum_{i\geq 0} \text{rank}(V_i) = \text{rkwt}(V(D))$$

The sum rank distance between $v(D) = \sum_{i\geq 0} v_i D^i, w(D) = \sum_{i\geq 0} w_i D^i \in \mathbb{F}_q[D]^{nm}$ is:

$$d_{\overline{SR}}(v(D), w(D)) = \sum_{i\geq 0} \text{rank}(\text{rowmat}(v_i) - \text{rowmat}(w_i))$$

The free sum rank distance of $\mathcal{C} \subset \mathbb{F}_q[D]^{nm}$ is defined as:

$$\begin{aligned} d_{\overline{SR}}(\mathcal{C}) &= \min\left\{ d_{\overline{SR}}(v(D), w(D)) \,|\, v(D), w(D) \in \mathcal{C}, v(D) \neq w(D) \right\} \\ &= \min\left\{ \sum_{i\geq 0} \text{rank}(\text{rowmat}(v_i)) \,|\, v(D) \in \mathcal{C}, v(D) \neq 0 \right\} \end{aligned}$$

Example 5.3.6. We show an example of a rank metric convolutional code according to Definition 5.3.5. Set $\psi = \text{rowmat}_{2\times 3}$ and let

$$G(D) = \begin{bmatrix} D & 1 & 1 & 1+D^2 & 0 & 0 \\ 1 & D & 1 & 0 & 0 & D^3 \end{bmatrix} \in \mathbb{F}_2[D]^{2\times(2\cdot 3)}$$

be the generator matrix. A codeword can be established as:

$$(1,1+D)\cdot G(D) = (1, 1+D+D^2, D, 1+D^2, 0, D^3+D^4) \implies \begin{bmatrix} 1 & 1+D+D^2 & D \\ 1+D^2 & 0 & D^3+D^4 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} D + \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} D^2 + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} D^3 + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} D^4$$

The sum rank distance of this codeword is $2 + 1 + 2 + 1 + 1 = 7$.

Theorem 5.3.7. (Sum Rank Upper Bound for Rank Metric Convolutional Codes) [NPRV17, Theorem 3] Let $\mathcal{C}$ be an $(n \times m, k, \delta)$-rank metric convolutional code. Then, the sum rank distance of $\mathcal{C}$ is upper bounded by:

$$d_{\overline{SR}}(\mathcal{C}) \leq n \left(\left\lfloor \frac{\delta}{k} \right\rfloor + 1 \right) - \left\lceil \frac{k \left(\left\lfloor \frac{\delta}{k} \right\rfloor + 1 \right) - \delta}{m} \right\rceil + 1 \tag{5.31}$$

Proof. Let $G(D)$ be a minimal generator matrix of $\mathcal{C}$ with row degrees $\nu_0, \nu_1, \ldots, \nu_{k-1}$. This can be assumed since there always exists an equivalent minimal generator matrix (see Theorem 2.1.27). Let $p = \min\{\nu_i : 0 \leq i < k\}$ denote the value of the smallest row degree and t the number of indexes ν_i among the indexes $\nu_0, \nu_1, \ldots, \nu_{k-1}$ having the value p. W.l.o.g., we shall assume that the row degrees of $G(D)$ are in nonincreasing order, i.e.,

$$\nu_0 \geq \cdots \geq \nu_{k-1-t} > \nu_{k-t} = \cdots = \nu_{k-1} = p$$

and that $\psi = \text{rowmat}_{n \times m}$ as in Definition 5.3.5.
Now, take a non-zero and constant $u(D) = u = (0, \ldots, 0, u_{k-t}, \ldots, u_{k-1}) \in \mathbb{F}_q^k$. Note that, due to this choice of $u(D)$, the degree of $v(D) = u(D)G(D)$ goes up to p and not to ν_0, which results in $v(D) = v_0 + v_1 D + v_2 D^2 + \cdots + v_p D^p$. Denote $V(D) = \psi(v(D)) = V_0 + V_1 D + \cdots + V_p D^p$. Observe that $v_0 = uG(0)$ is a linear combination of the last t rows of $G(0)$. Thus, we can select $u_{k-t}, \ldots, u_{k-1}$ such that not all are zero and the first $t - 1$ components of v_0 are zero. Therefore, also the first $\left\lfloor \frac{t-1}{m} \right\rfloor$ rows of $V_0 = \psi(v_0)$ are zero, implying that

$$\text{rank}(V_0) \leq n - \left\lfloor \frac{t-1}{m} \right\rfloor = n - \left\lceil \frac{t}{m} \right\rceil + 1.$$

Thus,

$$\text{rkwt}(V(D)) = \sum_{i=0}^{p} \text{rank}(V_i) \leq n - \left\lceil \frac{t}{m} \right\rceil + 1 + np = n(p+1) - \left\lceil \frac{t}{m} \right\rceil + 1.$$

This upper bound is maximised when p is as large as possible and t as small as possible. The maximum is achieved if as many row degrees as possible are equal. This happens if as many row degrees as possible are equal to the minimum row degree and the other row degrees are $p + 1$. It can be checked that, for given k and $\delta = \sum_{i=0}^{k-1} \nu_i$, these values are

$$p = \left\lfloor \frac{\delta}{k} \right\rfloor$$

and

$$t = k \left(\left\lfloor \frac{\delta}{k} \right\rfloor + 1 \right) - \delta.$$

Everything together results in:

$$\text{rkwt}(V(D)) \leq n(p+1) - \left\lceil \frac{t}{m} \right\rceil + 1 \leq n(\left\lfloor \frac{\delta}{k} \right\rfloor + 1) - \left\lceil \frac{k \left(\left\lfloor \frac{\delta}{k} \right\rfloor + 1 \right) - \delta}{m} \right\rceil + 1$$

So, if a convolutional code reaches the upper bound (5.31), then $G(D)$ must have [NPRV17, Corollary 4]:

- $k\left(\left\lfloor\frac{\delta}{k}\right\rfloor+1\right)-\delta$ rows of degree $\left\lfloor\frac{\delta}{k}\right\rfloor$
- and $\delta-k\left\lfloor\frac{\delta}{k}\right\rfloor$ rows of degree $\left\lfloor\frac{\delta}{k}\right\rfloor+1$

We can check:

$$\left(k\left(\left\lfloor\frac{\delta}{k}\right\rfloor+1\right)-\delta\right)+\left(\delta-k\left\lfloor\frac{\delta}{k}\right\rfloor\right)=k$$

and

$$\left(k\left(\left\lfloor\frac{\delta}{k}\right\rfloor+1\right)-\delta\right)\left\lfloor\frac{\delta}{k}\right\rfloor+\left(\delta-k\left\lfloor\frac{\delta}{k}\right\rfloor\right)\left(\left\lfloor\frac{\delta}{k}\right\rfloor+1\right)=\delta$$

This concludes the proof. □

Definition 5.3.8. [NPRV17, Section IV.] **(Maximum Rank Distance (MRD) convolutional code)** A rank metric convolutional code whose distance attains the upper bound in (5.31) is called *maximum rank distance* (MRD) convolutional code.

5.3.2 Code Construction

Proposition 5.3.9. [NPRV17, Section V.] Let $\mathbb{F}_q$ be a finite field and $A\in\mathbb{F}_q^{m\times m}$ a matrix with an irreducible characteristic polynomial $\chi(\lambda)$. Note that A^i, $0\leq i<m$, are linearly independent over $\mathbb{F}_q$ and

$$\mathbb{F}_q[A]=\left\{\sum_{i=0}^{m-1}u_iA^i : u_i\in\mathbb{F}_q, i=0,\dots,m-1\right\}\cong\mathbb{F}_{q^m}$$

is a field.

Proof. Since the characteristic polynomial is irreducible, the minimal polynomial $t(x)$ of A is equal to the characteristic polynomial $\chi(x)$. So, $\deg(t(x))=m$.
Next, we prove by contradiction that the A^i are linearly independent. Assume some A^i, $i<m$ are linearly dependent. This implies that there exists a polynomial $p(x)$ such that $\deg(p(x))<m$ and $p(A)=0$. It follows $t(x)|p(x)$ which is a contradiction to the degrees ↯

The characteristic polynomial $\chi(\lambda)$ is monic, irreducible and has degree m. This implies:

$$\frac{\mathbb{F}_q[x]}{\chi(x)}\cong\mathbb{F}_{q^m}$$

$\chi(x)$ is the minimal polynomial of A and $\chi(A)=0$. This provides the following isomorphism:

$$\frac{\mathbb{F}_q[x]}{\chi(x)}\cong\mathbb{F}_q[A]$$

Combining both isomorphism results in the statement of the proposition and shows that $\mathbb{F}_q[A]$ is also a field. □

First, we will show a construction of a specific MRD convolutional code. Later, there will also be give a general construction.

Lemma 5.3.10. [NPRV17, Section V.] Let $m>\delta$ and A as in Proposition 5.3.9. Then, each $(m\times m,1,\delta)$-rank metric convolutional code $\mathcal{C}$ generated by

$$G(D)=\sum_{i=0}^{\delta}\psi^{-1}(A^i)D^i\in\mathbb{F}_q[D]^{1\times m^2}$$

is MRD.

Proof. The code $\mathcal{C}$ is MRD if it achieves the upper bound (5.31), which in this case is equal to

$$\begin{aligned} d_{\overline{SR}}(\mathcal{C}) &= n\left(\left\lfloor\frac{\delta}{k}\right\rfloor+1\right)-\left\lceil\frac{k\left(\left\lfloor\frac{\delta}{k}\right\rfloor+1\right)-\delta}{m}\right\rceil+1 \\ &= m\left(\left\lfloor\frac{\delta}{1}\right\rfloor+1\right)-\left\lceil\frac{1\left(\left\lfloor\frac{\delta}{1}\right\rfloor+1\right)-\delta}{m}\right\rceil+1 \\ &= m\left(\delta+1\right)-\left\lceil\frac{\delta+1-\delta}{m}\right\rceil+1 \\ &= m(\delta+1)-1+1=m(\delta+1) \end{aligned}$$

since $n = m$ and $k = 1$ by assumption. Actually, we will show that for any non-zero $u(D) \in \mathbb{F}_q[D]$ the codeword $V(D) = \psi(u(D)G(D))$ has rank weight $\mathrm{rkwt}(V(D)) \geq m(\delta+1)$. Let $u(D) = \sum_{i\in\mathbb{N}} u_i D^i$ be a non-zero polynomial. W.l.o.g., we may suppose that $u_0 \neq 0$. Then, the first $\delta+1$ coefficients of $v(D) = u(D)G(D)$ are given by

$$v_i = \sum_{l=0}^{i} u_{i-l}G_l = \sum_{l=0}^{i} u_{i-l}\psi^{-1}(A^l) = \psi^{-1}\left(\sum_{l=0}^{i} u_{i-l}A^l\right), \quad i \leq \delta,$$

due to linearity of ψ^{-1}. It follows that the polynomial matrix $V(D) = \psi(v(D))$ has coefficients

$$V_i = \psi(v_i) = \psi\left(\psi^{-1}\left(\sum_{l=0}^{i} u_{l-i}A^l\right)\right) = \sum_{l=0}^{i} u_{i-l}A^l \in \mathbb{F}_q[A], \quad i \leq \delta.$$

Since the powers of A with degree less than m form a basis of $\mathbb{F}_q[A]$ over $\mathbb{F}_q$ by Proposition 5.3.9 and the scalar $u_0 \neq 0$ occurs in every linear combination, $V_i \neq 0$ for every $i \leq \delta < m$. Then, such a V_i is a non-zero element of the field in Proposition 5.3.9. Being an element of a field, a non-zero matrix is invertible and so V_i has rank m. Therefore:

$$\mathrm{rkwt}(V(D)) = \sum_{i\in\mathbb{N}} \mathrm{rank}(V_i) \geq \sum_{i=0}^{\delta} \mathrm{rank}(V_i) = m(\delta+1)$$

□

The next example will illustrate the above reasoning.

Example 5.3.11. [NPRV17, Example 5] Consider the companion matrix A of the irreducible polynomial $\chi(\lambda) = \lambda^3 + \lambda + 1 \in \mathbb{F}_2[\lambda]$,

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \in \mathbb{F}_2^{3\times 3},$$

to get a matrix A as in Proposition 5.3.9. Let $\mathcal{C}$ be the rank metric convolutional code with encoder

$$G(D) = \psi^{-1}(I) + \psi^{-1}(A)D = \psi^{-1}(I + AD) \in \mathbb{F}_2[D]^{1\times 9}.$$

For the sake of simplicity, let $\psi = \text{rowmat}$ (folding a row vector into a matrix), thus ψ^{-1} is the rowvec map (unfolding a matrix into a row vector). This means that

$$G(D) = \left[\begin{array}{ccc|ccc|ccc} 1 & D & 0 & 0 & 1 & D & D & D & 1 \end{array}\right].$$

Since $m = 3 > \delta = 1$, it follows by the Lemma 5.3.10 above that $\mathcal{C}$ is an MRD $(3 \times 3, 1, 1)$ rank metric convolutional code with distance equal to the upper bound (5.31), i.e., $d_{\overline{SR}}(\mathcal{C}) = m(\delta + 1) = 6$, as we will show.
Observe that due to the rather simple structure of the code, its codewords are

$$V(D) = \varphi(u(D)) = \psi(u(D)G(D)) = u(D)\psi(G(D)) = u(D)(I + AD).$$

Furthermore, notice that any message starting with a non-zero coefficient is equal either to $u(D) = 1 + \tilde{u}(D)D^2$ (5.32) or to $u(D) = 1 + D + \tilde{u}(D)D^2$ (5.33), for some $\tilde{u}(D) \in \mathbb{F}_2[D]$.
As a consequence, every codeword in $\mathcal{C}$ must be of the form

$$V(D) = (1 + \tilde{u}(D)D^2)(I + AD) = I + AD + \tilde{V}(D)D^2 \tag{5.32}$$

or of the form

$$V(D) = (1 + D + \tilde{u}(D)D^2)(I + AD) = I + (I + A)D + \tilde{V}(D)D^2, \tag{5.33}$$

for some $\tilde{V}(D) \in \mathbb{F}_2[D]^{3\times 3}$. Since in both cases the first two coefficients of $V(D)$ ($V_0 = I$ and $V_1 = A$ or $V_1 = I + A = A^3$) are nonzero, V_0 and V_1 are full rank matrices (invertible in the field $\mathbb{F}_2[A]$). Thus, the rank weight of any codeword of $\mathcal{C}$ is

$$\text{rkwt}(V(D)) = \text{rank}(V_0) + \text{rank}(V_1) + \text{rkwt}(\tilde{V}(D)) \geq 6.$$

The next page now describes the general construction of an $(n \times m, k, \delta)$-rank metric convolutional code under the condition $m \geq \delta + k$:

Theorem 5.3.12. [NPRV17, Theorem 6] **(Construction of on MRD Rank Metric Convolutional Code)** For the general construction of an MRD $(n \times m, k, \delta)$-rank metric convolutional code over $\mathbb{F}_q$, with $m \geq n$, we still need a matrix $A \in \mathbb{F}_q^{m \times m}$ with irreducible characteristic polynomial. Moreover, let $X \in \mathbb{F}_q^{n \times m}$ be any full row rank matrix and define the $k \times nm$ matrices:

$$G_i = \begin{bmatrix} \psi^{-1}(XA^{ki}) \\ \psi^{-1}(XA^{ki+1}) \\ \vdots \\ \psi^{-1}(XA^{ki+k-1}) \end{bmatrix}, \ 0 \leq i \leq \left\lfloor \frac{\delta}{k} \right\rfloor$$

and

$$G_{\lfloor \frac{\delta}{k} \rfloor + 1} = \begin{cases} 0 & \text{if } k \text{ divides } \delta \\ \begin{bmatrix} \psi^{-1}\left(XA^{k\lfloor \frac{\delta}{k} \rfloor + k}\right) \\ \vdots \\ \psi^{-1}\left(XA^{k+\delta-1}\right) \\ 0 \\ \vdots \\ 0 \end{bmatrix} & \text{otherwise} \end{cases} \tag{5.34}$$

Then, each $(n \times m, k, \delta)$-rank metric convolutional code $\mathcal{C}$ with encoder

$$G(D) = \sum_{i=0}^{\lfloor \frac{\delta}{k} \rfloor + 1} G_i D^i \in \mathbb{F}_q^{k \times nm},$$

whose coefficients are defined in (5.34), is MRD when $m \geq \delta + k$.

Proof. First of all, note that

$$r = \delta - k \left\lfloor \frac{\delta}{k} \right\rfloor$$

is the remainder of the integer division of δ by k, being $0 \leq r < k$. Accordingly,

$$0 < k - r = k\left(\left\lfloor \frac{\delta}{k} \right\rfloor + 1\right) - \delta \leq k.$$

By hypothesis, $m \geq \delta + k$, and therefore it follows that

$$0 < \frac{k\left(\left\lfloor \frac{\delta}{k} \right\rfloor + 1\right) - \delta}{m} \leq 1$$

for every $\delta \geq 0$. This shows that

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

In this case, Theorem 5.3.7 gives the following upper bound:

$$d_{\overline{SR}}(\mathcal{C}) \leq n\left(\left\lfloor\frac{\delta}{k}\right\rfloor+1\right)-\left\lceil\frac{k\left(\left\lfloor\frac{\delta}{k}\right\rfloor+1\right)-\delta}{m}\right\rceil+1 = n\left(\left\lfloor\frac{\delta}{k}\right\rfloor+1\right)-1+1$$
$$= n\left(\left\lfloor\frac{\delta}{k}\right\rfloor+1\right) \tag{5.35}$$

Consider now any message $u(D)=\sum_{i\in\mathbb{N}} u_i D^i \in \mathbb{F}_q[D]^k$ having, w.l.o.g., a non-zero constant term u_0. We will show that (5.35) is actually a lower bound for the rank weight of the codeword $V(D)=\sum_{i\in\mathbb{N}} V_i D^i = \varphi(u(D))$.
Let $v(D)=u(D)G(D)\in\mathbb{F}_q[D]^{1\times nm}$. Then, the first $\left\lfloor\frac{\delta}{k}\right\rfloor+1$ row vector coefficients of $v(D)$ are

$$v_i=\sum_{h=0}^{i} u_{i-h}G_h, \quad 0\leq i\leq\left\lfloor\frac{\delta}{k}\right\rfloor.$$

To carry out the proof, a more detailed result is needed: If $u_i=(u_{i,0},\ldots,u_{i,k-1})$ then

$$\begin{aligned}
v_i &= \sum_{h=0}^{i} u_{i-h}G_h = \sum_{h=0}^{i}\sum_{l=0}^{k-1} u_{i-h,l}\psi^{-1}(XA^{kh+l}) \\
&= \sum_{h=0}^{i}\sum_{l=kh}^{kh+k-1} u_{i-h,l-kh}\psi^{-1}(XA^{l}) \\
&= \sum_{h=0}^{i}\sum_{l=kh}^{kh+k-1} u_{i-\lfloor\frac{l}{k}\rfloor,l-k\lfloor\frac{l}{k}\rfloor}\psi^{-1}(XA^{l}) \\
&= \sum_{l=0}^{ki+k-1} u_{i-\lfloor\frac{l}{k}\rfloor,l-k\lfloor\frac{l}{k}\rfloor}\psi^{-1}(XA^{l}) \\
&= \psi^{-1}(X\sum_{l=0}^{ki+k-1} u_{i-\lfloor\frac{l}{k}\rfloor,l-k\lfloor\frac{l}{k}\rfloor}A^{l}) \\
&= \psi^{-1}(XB_i)
\end{aligned}$$

where $B_i=\sum_{l=0}^{ki+k-1} u_{i-\lfloor\frac{l}{k}\rfloor,l-k\lfloor\frac{l}{k}\rfloor}A^l$.
Once $k\left\lfloor\frac{\delta}{k}\right\rfloor-1<\delta$ and, by hypothesis $\delta+k\leq m$, it follows that $k\left\lfloor\frac{\delta}{k}\right\rfloor+k-1<\delta+k\leq m$. Thus, for all $0\leq i\leq\left\lfloor\frac{\delta}{k}\right\rfloor$, the matrices B_i are linear combinations of some powers A^l with exponents $0\leq l<m$, which are therefore linearly independent over $\mathbb{F}_q$ by Proposition 5.3.9. In particular, since there exist non-zero components of u_0 (which appear in the defining expression of B_i for $l\geq ki$), it follows that $B_i\neq 0$. Each $B_i\in\mathbb{F}_q[A]$ being a non-zero element of a field, it is invertible, thus it has full rank m. By definition, $V(D)=\psi(v(D))$, hence $V_i=\psi(v_i)=XB_i$ has full row rank n for every $0\leq i\leq\left\lfloor\frac{\delta}{k}\right\rfloor$ since X has full row rank as well. Finally, the rank weight of $V(D)$ is

$$\operatorname{rkwt}(V(D))=\sum_{i\in\mathbb{N}}\operatorname{rank}(V_i)\geq\sum_{i=0}^{\lfloor\frac{\delta}{k}\rfloor}\operatorname{rank}(V_i)=n\left(\left\lfloor\frac{\delta}{k}\right\rfloor+1\right).$$

To conclude the proof, we check that code $\mathcal{C}$ has degree δ.
If k does not divide δ, then looking at the definition of $G(D)$ and its coefficients (5.34), it is clear that the leading row coefficient matrix $[G(D)]_h$ of $G(D)$ has the first $\delta-k\left\lfloor\frac{\delta}{k}\right\rfloor$ rows of

$G_{\lfloor \frac{\delta}{k} \rfloor + 1}$ and the last $k\left(\left\lfloor \frac{\delta}{k} \right\rfloor + 1\right) - \delta$ rows of $G_{\lfloor \frac{\delta}{k} \rfloor}$. Since all XA^i, for $i < k + \delta \leq m$, are linearly independent, the rows of $[G(D)]_h$ are linearly independent as well. This implies that $[G(D)]_h$ has full row rank k and with Lemma 2.1.29 $G(D)$ is minimal. Therefore, the degree of $\mathcal{C}$ is equal to the sum of the row degrees and we can check:

$$\left(\left\lfloor \frac{\delta}{k} \right\rfloor + 1\right)\left(\delta - k\left\lfloor \frac{\delta}{k} \right\rfloor\right) + \left\lfloor \frac{\delta}{k} \right\rfloor\left(k\left(\left\lfloor \frac{\delta}{k} \right\rfloor + 1\right) - \delta\right) = \delta$$

If k does divide δ, then the row coefficient matrix $[G(D)]_h$ contains all k rows from $G_{\lfloor \frac{\delta}{k} \rfloor}$. $[G(D)]_h$ has again full row rank k, which implies minimality of $G(D)$ by Lemma 2.1.29. It follows

$$k\left\lfloor \frac{\delta}{k} \right\rfloor = k\left(\frac{\delta}{k}\right) = \delta.$$

□

The next example will show how to construct an MRD rank metric convolutional code with parameters $n < m$ and $k > 1$.

Example 5.3.13. [NPRV17, Example 8] To construct a $(3\times 4, 2, 2)$ code, using the proposed algorithm, matrices A and X have to be defined. Being arbitrarily, we choose once more ψ =rowmat and ψ^{-1} =rowvec.
Consider the companion matrix A of the irreducible polynomial $\chi(\lambda) = \lambda^4 + \lambda + 1 \in \mathbb{F}_2[\lambda]$ and the full row rank matrix $X = \left[\begin{array}{cc} I_3 & 0_{3\times 1} \end{array}\right]$, i.e.,

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix} \in \mathbb{F}_2^{4\times 4} \qquad \text{and} \qquad X = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \in \mathbb{F}_2^{3\times 4}.$$

Thus,

$$XA = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad XA^2 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}, \qquad XA^3 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix},$$

whence, by (5.34), we obtain

$$G_0 = \begin{bmatrix} \psi^{-1}(X) \\ \psi^{-1}(XA) \end{bmatrix} = \left[\begin{array}{cccc|cccc|cccc} 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{array}\right]$$
$$G_1 = \begin{bmatrix} \psi^{-1}(XA^2) \\ \psi^{-1}(XA^3) \end{bmatrix} = \left[\begin{array}{cccc|cccc|cccc} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \end{array}\right].$$

Consequently, a generator matrix for the requested code is

$$G(D) = G_0 + G_1 D = \left[\begin{array}{cccc|cccc|cccc} 1 & 0 & D & 0 & 0 & 1 & 0 & D & D & D & 1 & 0 \\ 0 & 1 & 0 & D & D & D & 1 & 0 & 0 & D & D & 1 \end{array}\right].$$

Note that, if $u(D) = (u_0, u_1) + (u_2, u_3)D + \cdots \in \mathbb{F}_2[D]^2$, then the first two coefficients of the codeword

$$V(D) = \varphi(u(D)) = u(D)\psi(G_0 + G_1 D) = ((u_0, u_1) + (u_2, u_3)D + \dots)\left(\begin{bmatrix} X \\ XA \end{bmatrix} + \begin{bmatrix} XA^2 \\ XA^3 \end{bmatrix} D\right)$$

are

$$V_0 = X(u_0 I + u_1 A) \text{ and } V_1 = X(u_2 I + u_3 A + u_0 A^2 + u_1 A^3).$$

Both matrices have rank equal to 3 for any choice of u_0, u_1, u_2, and u_3, with at least u_0 or u_1 non-zero. That follows from the same reasoning as in the proof of Theorem 5.3.12 and with Proposition 5.3.9. This confirms that the distance of the code is at least $3\left(\left\lfloor \frac{2}{2} \right\rfloor + 1\right) = 6$.

Remark 5.3.14. **(Decoding)** Napp et al. describe in their paper [NPRV18] an algorithm to decode these MRD codes. The algorithm is based on the Viterbi algorithm. More information can be found in the paper. This thesis will not focus on the decoding of MRD codes.

5.4 (Partial) Unit Memory Codes

This section is based on the paper of Wachter-Zeh et al. [WZSS15]. It is written by engineers, who also focus on the practical point of view. Nevertheless, we will investigate it in a mathematical context.

In this section, we will consider so-called *(partial) unit memory ((P)UM) codes*. (P)UM codes are a special class of convolutional codes that have *memory one*. Wachter-Zeh et al. [WZSS15] additionally describe the lifting of (P)UM codes to get constant dimension codes. As stated at the end of Chapter 3, these types can also be used for random linear network coding but they will not be discussed in this thesis.

5.4.1 Preliminaries

Definition 5.4.1. (Unit Memory (UM) Codes) [WZSS15, Section II.D.] *UM*(n,k) *codes* are a special class of convolutional codes with memory $\mu = 1$. The generator matrix consists therefore of two $k \times n$ submatrices $G^{(0)}$ and $G^{(1)}$. In case of a *UM*(n,k) *code*, these matrices have both full rank k.

Definition 5.4.2. (Partial Unit Memory (PUM) Codes) [WZSS15, Section II.D.] *PUM*$(n,k,k^{(1)})$ *codes* are a special class of convolutional codes with memory $\mu = 1$. The generator matrix consists therefore of two $k \times n$ submatrices $G^{(0)}$ and $G^{(1)}$. In case of a *PUM*$(n,k,k^{(1)})$ *code*, $\operatorname{rank}(G^{(0)}) = k$ and $\operatorname{rank}(G^{(1)}) = k^{(1)} < k$ has to hold.
W.l.o.g., for PUM codes, we assume that the lowermost $k - k^{(1)}$ rows of $G^{(1)}$ are zero and we denote:

$$G^{(0)} = \begin{bmatrix} G^{(00)} \\ G^{(01)} \end{bmatrix} \quad \text{and} \quad G^{(1)} = \begin{bmatrix} G^{(10)} \\ 0 \end{bmatrix}$$

where $G^{(00)}$, $G^{(10)}$ are $k^{(1)} \times n$ matrices of full rank and $G^{(01)}$ is a full rank $(k - k^{(1)}) \times n$ matrix over $\mathbb{F}_{q^m}$.

The encoding rule for each code block of a (P)UM code is hence given by

$$c^{(i)} = u^{(i)} \cdot G^{(0)} + u^{(i-1)} \cdot G^{(1)}, \qquad \forall i = 1, 2, \ldots,$$

where $u^{(i)}$ and $u^{(i-1)} \in \mathbb{F}_{q^m}^k$ for all i. The memory of (P)UM codes is $\mu = 1$. The overall constraint length is $\nu = k$ for UM codes and $\nu = k^{(1)}$ for PUM codes (see Definition 2.1.25). The current state in the trellis of a (P)UM code over $\mathbb{F}_{q^m}$ can be associated with $s^{(i)} = u^{(i-1)} \cdot G^{(1)}$. Therefore, there are $q^{mk^{(1)}}$ possible states (see Subsection 2.1.2).

This section requires knowledge about active rank distances. The definitions can be found at the end of Section 2.2.

Definition 5.4.3. (Slope of Active Row Rank Distance) [WZSS15, Definition 7] The *slope of the active row rank distance* is:

$$\sigma = \lim_{j \to \infty} \left\{ \frac{d_{SR}^{(r)}(j, \mathcal{C})}{j} \right\}$$

Theorem 5.4.4. [Det94, Theorem 2.2] Let $\mathcal{C} = \operatorname{Im}(G(D))$ be a catastrophic code. Then, the slope $\sigma = 0$.

Proof. If $\mathcal{C}$ is catastrophic, there exists an input $u(D)$ with infinite weight such that the output $v(D) = u(D)G(D)$ has finite sum rank weight. This implies that $\lim_{j\to\infty} d_{SR}^{(r)}(j, \mathcal{C})$ is bounded and $\sigma = 0$. □

Corollary 5.4.5. (Upper Bounds) [WZSS15, Corollary 1] For a UM(n, k) code, where the overall constraint length is given by $\nu = k$, the free sum rank distance is bounded by:

$$d_{SR}(UM) \leq 2n - k + 1 \tag{5.36}$$

For a PUM$(n, k, k^{(1)})$ code, where the overall constraint length is given by $\nu = k^{(1)} < k$, the free sum rank distance is bounded by:

$$d_{SR}(PUM) \leq n - k + \nu + 1 \tag{5.37}$$

For both, UM and PUM codes, the slope is bounded by:

$$\sigma \leq n - k. \tag{5.38}$$

Proof. Wachter-Zeh et al. [WSBZ11, Theorem 2] show that the column sum rank distance is smaller than the Hamming distance. Precisely, let $c \in \mathbb{F}_{q^m}$ and we have

$$\text{rank}(c) \leq wt_H(c).$$

Hence,

$$d_{SR}(\mathcal{C}) \leq d_H(\mathcal{C}).$$

Since [TJ83] and [PMAG06] state the desired bounds (5.36), (5.37) and (5.38) for the free Hamming distance, we obtain the upper bounds of the corollary also for the sum rank distance. □

5.4.2 Code Construction

Definition 5.4.6. [WZSS15, Definition 8] **((P)UM Code based on Gabidulin Code)** Let $k + k^{(1)} \leq n \leq m$, where $k^{(1)} \leq k$. Further, let $g_0, g_1, \ldots, g_{n-1} \in \mathbb{F}_{q^m}$ be linearly independent over $\mathbb{F}_q$. We define a PUM$_G(n, k, k^{(1)})$ code, respectively a UM$_G(n, k)$ code, over $\mathbb{F}_{q^m}$ by a generator matrix $G(D) = G^{(0)} + G^{(1)}D$ with memory $\mu = 1$. We use the $k \times n$ submatrices $G^{(0)}$ and $G^{(1)}$:

$$G^{(0)} = \left[\frac{G^{(00)}}{G^{(01)}}\right] = \left[\begin{array}{cccc} g_0 & g_1 & \cdots & g_{n-1} \\ g_0^{[1]} & g_1^{[1]} & \cdots & g_{n-1}^{[1]} \\ \vdots & \vdots & \ddots & \vdots \\ g_0^{[k^{(1)}-1]} & g_1^{[k^{(1)}-1]} & \cdots & g_{n-1}^{[k^{(1)}-1]} \\ \hline g_0^{[k^{(1)}]} & g_1^{[k^{(1)}]} & \cdots & g_{n-1}^{[k^{(1)}]} \\ g_0^{[k^{(1)}+1]} & g_1^{[k^{(1)}+1]} & \cdots & g_{n-1}^{[k^{(1)}+1]} \\ \vdots & \vdots & \ddots & \vdots \\ g_0^{[k-1]} & g_1^{[k-1]} & \cdots & g_{n-1}^{[k-1]} \end{array}\right]$$

and

$$G^{(1)} = \left[\frac{G^{(10)}}{0}\right] = \left[\begin{array}{cccc} g_0^{[k]} & g_1^{[k]} & \cdots & g_{n-1}^{[k]} \\ g_0^{[k+1]} & g_1^{[k+1]} & \cdots & g_{n-1}^{[k+1]} \\ \vdots & \vdots & \ddots & \vdots \\ g_0^{[k+k^{(1)}-1]} & g_1^{[k+k^{(1)}-1]} & \cdots & g_{n-1}^{[k+k^{(1)}-1]} \\ \hline & & 0 & \end{array}\right].$$

As usual, we denote $x^{[i]} = x^{q^i}$.

Remark 5.4.7. The added index G to PUM_G or UM_G as in Definition 5.4.6 emphasises that these are specific codes constructed as in this definition.

Theorem 5.4.8. [WZSS15, Theorem 1] The construction in Definition 5.4.6 is basic and minimal.

Proof. First, $G(D)$ is a q-Vandermonde matrix and therefore has full rank. So, $G(D)$ really defines an encoding matrix.
Second, we show that $G(D)$ is in basic form. According to the definition, $G(D)$ is basic if it is polynomial and if there exists a polynomial right inverse $G^{-1}(D)$ such that $G(D) \cdot G^{-1}(D) = I_{k\times k}$. By definition, $G(D)$ is polynomial. By Dettmar [Det94, Theorem A.4] and Theorem 5.4.4, a polynomial right inverse exists if and only if $G(D)$ is non-catastrophic and hence, if the slope $\sigma > 0$. The slope is calculated later in Corollary 5.4.10, proving that $\sigma > 0$.
Third, we show that $G(D)$ is minimal. Indeed, the leading coefficient matrix

$$[G(D)]_h = \begin{bmatrix} G^{(10)} \\ G^{(01)} \end{bmatrix}$$

has full rank since g_i are linearly independent and $k + k^{(1)} \leq m$. Lemma 2.1.29 shows that $G(D)$ is minimal due to the fact that $[G(D)]_h$ has full row rank. □

Theorem 5.4.9. (Lower Bounds on Active Distance) [WZSS15, Theorem 2] Let $k + k^{(1)} \leq n \leq m$, where $k^{(1)} \leq k$. Let $\mathcal{C}$ be a $\mathrm{UM}_G(n,k)$ code, respectively $\mathrm{PUM}_G(n,k,k^{(1)})$ code, over $\mathbb{F}_{q^m}$ as in Definition 5.4.6. Then,

$$\begin{aligned} d_{SR}^{(r)}(1,\mathcal{C}) &\geq d_{01} = \delta_1^{(r)} \\ d_{SR}^{(r)}(j,\mathcal{C}) &\geq d_0 + (j-2)\cdot d_\sigma + d_1 = \delta_j^{(r)}, \quad \forall j \geq 2 \\ d_{SR}^{(c)}(j,\mathcal{C}) &\geq d_0 + (j-1)\cdot d_\sigma = \delta_j^{(c)}, \quad \forall j \geq 1 \\ d_{SR}^{(rc)}(j,\mathcal{C}) &\geq (j-1)\cdot d_\sigma + d_1 = \delta_j^{(rc)}, \quad \forall j \geq 1 \end{aligned}$$

where

$$d_{01} = \begin{cases} n-k+k^{(1)}+1 & \text{if } k^{(1)} < k \\ \infty & \text{if } k^{(1)} = k \end{cases}$$

$$d_0 = d_1 = n-k+1$$

$$d_\sigma = n-k-k^{(1)}+1.$$

Proof. We first define some Gabidulin codes (see Theorem 2.2.7) in Table 5.3 which are defined by submatrices of G.

Generator Matrix	Code Notation	Code Parameters	Minimum Rank Distance
$G^{(0)}$	$\mathcal{C}_0$	$\mathcal{GA}[n,k]$	$d_0 = n-k+1$
$\begin{bmatrix} G^{(01)} \\ G^{(10)} \end{bmatrix}$	$\mathcal{C}_1$	$\mathcal{GA}[n,k]$	$d_1 = n-k+1$
$G^{(01)}$	$\mathcal{C}_{01}$	$\mathcal{GA}[n,k-k^{(1)}]$	$d_{01} = n-k+k^{(1)}+1$
$G_\sigma = \begin{bmatrix} G^{(00)} \\ G^{(01)} \\ G^{(10)} \end{bmatrix}$	$\mathcal{C}_\sigma$	$\mathcal{GA}[n,k+k^{(1)}]$	$d_\sigma = n-k-k^{(1)}+1$

TABLE 5.3: Submatrices of the (P)UM$_G$ code from Definition 5.4.6 and their block codes. [WZSS15, Table I]

Since they are Gabidulin codes, we can calculate the minimum rank distance with Theorem 2.2.7. d_{01} is infinite if $k = k^{(1)}$ since in this case $G^{(01)}$ does not exist.
Recall the encoding rule:

$$c^{(i)} = u^{(i)} \cdot G^{(0)} + u^{(i-1)} \cdot G^{(1)} \qquad \forall i = 1, 2, \dots$$

where $u^{(i)}$ and $u^{(i-1)} \in \mathbb{F}_{q^m}^k$ for all i. The definition of the state is:

$$s^{(i)} = u^{(i-1)} \cdot G^{(1)}$$

Take care that each code block of length n can be seen as a codeword of $\mathcal{C}_\sigma$.
For the estimation of the active row rank distance, the encoder starts in the zero state and hence, $u^{(-1)} = 0$. For the first order active row distance $d_{SR}^{(r)}(1, \mathcal{C})$, we look at all code sequences of the form $(\dots, 0, c^{(0)}, 0, \dots)$, where the state at position 1 is again 0. This is only possible if $u^{(0)} = (0, \dots, 0, u_{k^{(1)}}^{(0)}, \dots, u_{k-1}^{(0)})$ and $u^{(i)} = 0$, $\forall i \geq 1$. In this case, $c^{(0)} \in \mathcal{C}_{01}$ with distance d_{01} and the encoder returns immediately to the zero state. For the UM$_G$ case, $u^{(0)} = 0$ and the only codeword in $\mathcal{C}_0^{(r)}$ is the all-zero codeword and thus, $d_{SR}^{(r)}(1, \mathcal{C}) = \infty$.

For higher orders of $d_{SR}^{(r)}(j, \mathcal{C})$, we have to consider all code sequences starting with $c^{(0)} \in \mathcal{C}_0$ (since $u^{(-1)} = 0$ and $c^{(0)} = u^{(0)} \cdot G^{(0)} + u^{(-1)} \cdot G^{(1)}$), followed by $j-2$ non-zero codewords of C_σ and one final code block, resulting from $u^{(j-1)} = (0, \dots, 0, u_{k^{(1)}}^{(j-1)}, \dots, u_{k-1}^{(j-1)})$ and for the UM$_G$ case $u^{(j-1)} = 0$. For the UM$_G$ and the PUM$_G$ case, the block $u^{(j-2)}$ is arbitrary, therefore $c^{(j-1)} = u^{(j-1)} \cdot G^{(0)} + u^{(j-2)} \cdot G^{(1)} \in \mathcal{C}_1$. Summing up gives: $d_{SR}^{(r)}(j, \mathcal{C}) \geq d_0 + (j-2) \cdot d_\sigma + d_1 = \delta_j^{(r)}$.

For the estimation of $d_{SR}^{(c)}(j, \mathcal{C})$, the encoder starts in the zero state but ends in any state. Thus, $c^{(0)} \in \mathcal{C}_0$ is followed by $j-1$ arbitrary information blocks resulting in codewords from $\mathcal{C}_\sigma$. So, we obtain $d_{SR}^{(c)}(j, \mathcal{C}) \geq d_0 + (j-1) \cdot d_\sigma = \delta_j^{(c)}$.

For the active reverse column rank distances, we start in any block. Hence, all first $j-1$ blocks are from $\mathcal{C}_\sigma$. The last block is from $\mathcal{C}_1$ in order to end in the zero state. So, we have $d_{SR}^{(rc)}(j, \mathcal{C}) \geq (j-1) \cdot d_\sigma + d_1 = \delta_j^{(rc)}$. □

Corollary 5.4.10. [WZSS15, Corollary 2] Let $k + k^{(1)} \leq n \leq m$, where $k^{(1)} \leq k$ and ν be the overall constraint length. Let $\mathcal{C}$ be a $\mathrm{UM}_G(n,k)$, respectively a $\mathrm{PUM}_G(n,k,k^{(1)})$, code over $\mathbb{F}_{q^m}$ as in Definition 5.4.6. Then, its free sum rank distance d_{SR} for $k^{(1)} = k$ is

$$d_{SR}(\mathrm{UM}_G) \geq 2(n-k+1),$$

and for $k^{(1)} < k$

$$d_{SR}(\mathrm{PUM}_G) = n-k+\nu+1.$$

The slope σ of $\mathcal{C}$ for both cases is:

$$\sigma \geq n-k-k^{(1)}+1.$$

Thus, for any $k^{(1)} < k$, the construction attains the upper bound on the free sum rank distance of PUM codes (5.37). When $k^{(1)} = k = 1$, we reach the upper bound on the free sum rank distance of UM codes (5.36). For $k^{(1)} = 1 \leq k$, the upper bound on the slope is attained (5.38).

Proof. We use the fact that

$$d_{SR}(\mathcal{C}) = \min_j \left\{ d_{SR}^{(r)}(j,\mathcal{C}) \right\}$$

and the lower bounds of Theorem 5.4.9.

For the UM_G case with $k^{(1)} = k$, it is known that $d_{01} = \infty$. So, the minimal $\delta_j^{(r)}$ is at $j = 2$. We obtain:

$$d_{SR}(\mathrm{UM}_G) = \min_j \left\{ d_{SR}^{(r)}(j,\mathrm{UM}_G) \right\} \geq \min_j \left\{ \delta_j^{(r)} \right\} = d_0 + d_1 = 2(n-k+1)$$

For a PUM_G code with $k^{(1)} < k$, the minimal $\delta_j^{(r)}$ is at $j = 1$. This holds as d_{01} is here not infinity and $d_{01} = \delta_1^{(r)} < \delta_2^{(r)}$ since

$$\delta_1^{(r)} = n-k+k^{(1)}+1 \overset{k^{(1)} \leq n-k}{\leq} (n-k+1)+(n-k) < 2(n-k+1) = d_0+d_1 = \delta_2^{(r)}.$$

Then, we get:

$$d_{SR}(\mathrm{PUM}_G) = \min_j \left\{ d_{SR}^{(r)}(j,\mathrm{PUM}_G) \right\} \geq \min_j \left\{ \delta_j^{(r)} \right\} = d_{01} = n-k+k^{(1)}+1 \overset{k^{(1)}=\nu}{=} n-k+\nu+1$$

Theorem 5.4.5 with (5.37) shows that in the case of a PUM_G code, we have

$$d_{SR}(\mathrm{PUM}_G) \leq n-k+\nu+1.$$

Bringing everything together results in:

$$d_{SR}(\mathrm{PUM}_G) = n-k+\nu+1$$

For the slope σ, we calculate

$$\sigma = \lim_{j\to\infty} \left\{ \frac{d_{SR}^{(r)}(j,\mathcal{C})}{j} \right\} \geq \lim_{j\to\infty} \left\{ \frac{\delta_j^{(r)}}{j} \right\} = d_\sigma = n-k-k^{(1)}+1.$$

This finishes the proof. □

Remark 5.4.11. [WZSS15, p. 3205] The construction in Definition 5.4.6 creates a low-rate code since $k + k^{(1)} \leq n$ has to hold.

Remark 5.4.12. There exists an extended construction of Definition 5.4.6 for an arbitrary code rate. Wachter-Zeh et al. [WZSS15] present the adapted construction in Section IV.B. This newly constructed generator matrix is in minimal basic encoding form and allows a flexible code rate. However, this new construction cannot guarantee achieving the free distance upper bound as it is possible for PUM_G codes.

Remark 5.4.13. **(Decoding)** The (P)UM_G construction in Definition 5.4.6 can be decoded by multiple applications of the bounded minimum distance (BMD) error-erasure decoding algorithms. More details about the BMD decoding algorithms can be found in [SKK08] and [GP08]. Wachter-Zeh et al. [WZSS15] describe the adapted procedure in Section V.

5.4.3 Properties of (P)UM_G Codes

This subsection will list the properties of the specific (P)UM_G codes constructed in Definition 5.4.6.

The presented (P)UM_G code construction uses minimal generator matrices. Let ν be the sum of the row degrees and δ the degree of the code.

$\text{UM}_G(n, k)$ codes

- $G^{(0)}$ and $G^{(1)}$ have both full rank and so $k = k^{(1)}$.
- $\nu = \delta$ (by minimality) and $\nu = k$ (by definition of a UM code) $\implies \delta = k$.
- The generalised Singleton bound in Theorem 2.2.10 gives:

$$d_{SR}(\text{UM}_G) \leq (n-k)\left(\left\lfloor \frac{\delta}{k} \right\rfloor + 1\right) + \delta + 1 = 2n - k + 1$$

  This is also the upper bound for the free distance of a UM code (5.36) and a UM_G code can reach the Singleton bound.

- Unfortunately, the construction in Definition 5.4.6 of a UM_G code only guarantees reaching the upper bound if $k = 1$ (see Corollary 5.4.10).

We will now present a lower bound for the truncated sum rank distance of a UM_G code. The same procedure as in the proof of Theorem 5.4.9 by [WZSS15] will be used. Let $c = (c^{(0)}, c^{(1)}, \dots)$ be a codeword. $c^{(0)} = u^{(0)} \cdot G^{(0)} + u^{(-1)} \cdot G^{(1)}$ by definition of a UM code. Since $u^{(-1)} = 0$, it follows that $c^{(0)} \in \mathcal{C}_0$ and

$$\text{rank}(\phi(c^{(0)})) \geq n - k + 1.$$

$c^{(1)} = u^{(1)} \cdot G^{(0)} + u^{(0)} \cdot G^{(1)}$. $u^{(0)}$ and $u^{(1)}$ can be freely chosen which implies that $c^{(1)} \in \mathcal{C}_\sigma$ and

$$\text{rank}(\phi(c^{(1)})) \geq n - k - k^{(1)} + 1.$$

The same holds for $c^{(2)}$ and $\text{rank}(\phi(c^{(2)})) \geq n - k - k^{(1)} + 1$. We can sum up and get that:

$$d^1_{SR}(\text{UM}_G) \geq n - k + 1 + n - k - k^{(1)} + 1 = 2n - 2k + 2 - k^{(1)} = 2n - 3k + 2 \tag{5.39}$$

$$d^2_{SR}(\text{UM}_G) \geq 3n - 3k + 3 - 2k = 3n - 5k + 3 \tag{5.40}$$

We can compare these bounds to the upper bounds of the truncated sum rank distance in Proposition 2.2.13 to check the m-MSR property of a UM_G code:

$$d_{SR}^{j}(\mathrm{UM}_G) \leq (j+1)(n-k)+1$$

For example, $d_{SR}^{1}(\mathrm{UM}_G) \leq (2)(n-k)+1 = 2n-2k+1$ does not coincide with the lower bound of the UM_G code (5.39). This shows that we cannot guarantee that an arbitrary UM_G code is m-MSR.

To decide if a UM_G code is MRP, we recall the Definition 2.2.22:

$$\begin{aligned} d_{CR}^{j}(\mathcal{C}) &= \min\left\{\mathrm{rank}(\phi(v_0)|\cdots|\phi(v_j)) \mid v(D)=\sum_{i=0}^{j} v_i D^i \in \mathcal{C} \text{ and } v_0 \neq 0\right\} \\ &= \min\left\{\mathrm{rank}(\phi(v_{[0,j]})) \mid H_{trunc}(j) v_{[0,j]}^{T} = 0 \text{ and } v_0 \neq 0\right\} \\ &= \min\left\{\mathrm{rank}(\phi(v_{[0,j]})) \mid v_{[0,j]} = u_{[0,j]} G_{trunc}(j) \text{ and } u_0 \neq 0\right\} \end{aligned}$$

The third equality holds if $G^{(0)}$ has full row rank. We know that this is true for (P)UM codes.

It holds that $d_{CR}^{j}(\mathcal{C}) \leq d_{SR}^{j}(\mathcal{C})$ and $d_{CR}^{j}(\mathcal{C}) \leq m$ which follows directly from the definition.
A code is MRP if

$$d_{CR}^{L}(\mathcal{C}) = (n-k)(L+1)+1 \text{ with } L = \left\lfloor \frac{\delta}{k} \right\rfloor + \left\lfloor \frac{\delta}{n-k} \right\rfloor.$$

If m is smaller than this bound, then the code cannot be MRP.
We will give an example of a UM_G code which is not MRP although m would be large enough.

Example 5.4.14. Let $k = k^{(1)} = 1, n = 3$ and $m = 5$. It follows that $L = 1$ and the MRP bound is $(n-k)(L+1)+1 = 5 \leq m$. We define $G(D)$ as in Theorem 5.4.6 over $\mathbb{F}_{2^5} \cong \frac{\mathbb{F}_2[x]}{1+x^2+x^5}$ and let α be a zero of the used irreducible polynomial to construct the field. Then, α, α^{19} and α^3 are linearly independent over $\mathbb{F}_2$ and we get:

$$G(D) = \begin{bmatrix} \alpha & \alpha^{19} & \alpha^3 \end{bmatrix} + \begin{bmatrix} \alpha^2 & \alpha^7 & \alpha^6 \end{bmatrix} D$$

$u(D) = 1$ generates the codeword

$$(\alpha, \alpha^{19}, \alpha^3) + (\alpha^2, \alpha^7, \alpha^6)D = (\alpha, \alpha+\alpha^2, \alpha^3) + (\alpha^2, \alpha^2+\alpha^4, \alpha+\alpha^3)D.$$

This results in the following matrix if we choose $\{1, \alpha, \alpha^2, \alpha^3, \alpha^4\}$ as the basis:

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

This codeword has rank 4 and so the code cannot reach the free column rank distance of 5. Hence, it is not MRP.

This example shows that an arbitrary UM_G code is not MRP in general.

$\mathbf{PUM}_G(n, k, k^{(1)})$ codes

We will now analyse the same properties for a PUM_G code.

- By definition of a PUM code, we have that $k^{(1)} < k$.
- $\nu = \delta$ (by minimality) and $\nu = k^{(1)} < k$ (by definition of a PUM code) $\implies \delta = k^{(1)}$.
- The generalised Singleton bound in Theorem 2.2.10 gives:

$$d_{SR}(\text{PUM}_G) \leq (n-k)\left(\left\lfloor \frac{\delta}{k} \right\rfloor + 1\right) + \delta + 1 = n - k + k^{(1)} + 1$$

  This is also the upper bound for the free distance of a PUM code (5.37) and a PUM_G code can reach the Singleton bound.
- The construction in Definition 5.4.6 creates a PUM_G code which reaches that upper bound.

We will again present a lower bound for the truncated sum rank distance of a PUM_G code. It works in the same way as for UM_G codes in the subsection above. The only difference is that we cannot set $k^{(1)} = k$. With that we obtain:

$$d^1_{SR}(\text{PUM}_G) \geq n - k + 1 + n - k - k^{(1)} + 1 = 2n - 2k + 2 - k^{(1)} \tag{5.41}$$

We can compare this to the upper bounds of the truncated sum rank distance in Proposition 2.2.13:

$$d^j_{SR}(\text{PUM}_G) \leq (j+1)(n-k) + 1$$

For example, $d^1_{SR}(\text{PUM}_G) \leq (2)(n-k)+1 = 2n-2k+1$ does not coincide with the lower bound of the PUM_G code (5.41). This shows that we cannot guarantee that an arbitrary PUM_G code is m-MSR.

We will now give an example of a PUM_G code which is not MRP to show that, as for UM_G codes, PUM_G codes are not MRP in general:

Example 5.4.15. Let $k^{(1)} = 1$, $k = 2$, $n = 3$ and $m = 3$. It follows that $L = 1$ and the MRP bound is $(n-k)(L+1)+1 = 3 \leq m$. We define $G(D)$ as in Theorem 5.4.6 over $\mathbb{F}_{2^3} \cong \frac{\mathbb{F}_2[x]}{1+x+x^3}$ and let α be a zero of the used irreducible polynomial to construct the field. Then, α^4, α^5 and α^6 are linearly independent over $\mathbb{F}_2$ and we get:

$$G(D) = \begin{bmatrix} \alpha^4 & \alpha^5 & \alpha^6 \\ \alpha^8 & \alpha^{10} & \alpha^{12} \end{bmatrix} + \begin{bmatrix} \alpha^{16} & \alpha^{20} & \alpha^{24} \\ 0 & 0 & 0 \end{bmatrix} D = \begin{bmatrix} \alpha^4 & \alpha^5 & \alpha^6 \\ \alpha & \alpha^3 & \alpha^5 \end{bmatrix} + \begin{bmatrix} \alpha^2 & \alpha^6 & \alpha^3 \\ 0 & 0 & 0 \end{bmatrix} D$$

$u(D) = (1 + D, 1)$ generates the codeword

$$\begin{aligned}(\alpha^4 + \alpha, \alpha^5 + \alpha^3, \alpha^6 + \alpha^5) + (\alpha^4 + \alpha^2, \alpha^5 + \alpha^6, \alpha^6 + \alpha^3)D + (\dots)D^2 \\ = (\alpha^2, \alpha^2, \alpha) + (\alpha, \alpha, \alpha + \alpha^2)D + (\dots)D^2.\end{aligned}$$

This results in the following matrix if we choose $\{1, \alpha, \alpha^2\}$ as the basis and consider the truncated codeword of length two:

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

This codeword has rank 2. So, the code cannot reach the free column rank distance of 3 and, as a consequence, is not MRP.

Degree and Field Size

The (P)UM$_G$ code has a fixed degree ($\delta = k$ or $k^{(1)}$). This locks the variable δ in the Singleton bound and makes it less flexible to construct a (P)UM$_G$ code with an arbitrary Singleton bound. However, the limited memory provides a simpler code construction which makes the analysis of the code easier.

There are some disadvantages of PUM$_G$ codes, i.e., having fixed degree, not being m-MSR for sure, memory being equal to 1 and having low code rate. However, this type of code can be constructed with a small field size of $\mathbb{F}_{q^m}$ where $m \geq n$.

5.4.4 Summary of (P)UM$_G$ Codes

The construction of the PUM$_G$ code reaches the generalised Singleton bound. However, UM$_G$ codes, i.e., codes where also the $G^{(1)}$ has full rank, and PUM$_G$ codes do not guarantee to reach the maximum truncated sum rank distance. Furthermore, an arbitrary (P)UM$_G$ code is generally not MRP. One can remark that the (P)UM$_G$ code is usually constructed over $\mathbb{F}_{q^m}$ (see Definition 5.4.6), where m is just bigger or equal to $n \geq k+k^{(1)}$. On the contrary, MRP or m-MSR codes are constructed over $\mathbb{F}_{q^M}$, where $M \geq q^{n(L+2)-1}$ or $M \geq q^{n(\mu+2)-1}$ respectively. These fields are bigger than the ones for the (P)UM$_G$ code constructions. However, these other constructions like MRP or m-MSR codes have higher and guaranteed truncated distance properties. Instead, the (P)UM codes have memory $\mu = 1$ and therefore they are less complex.

5.5 Summary of the Constructions

We have seen different code constructions. First, m-MSR codes have been presented which maximise the j-th column sum rank distance for a good decoding property in a sliding window network without time delay. Almeida et al. [AMPN20] conjecture an improvement of the field size for the m-MSR code construction. Next, the MRP code construction maximises the free column rank distance for a good error correction in a network with time delay. Furthermore, we have constructed maximum sum rank distance codes over another field structure resulting in MRD codes designed by Napp et al. [NPRV17]. This construction has promising properties. We will analyse and compare MRD codes to m-MSR codes in Section 6.1. Lastly, (P)UM codes have been designed, which are convolutional codes with memory one. PUM_G codes achieve the maximum free sum rank distance and are constructed in combination with Gabidulin codes. As a downside, they are not m-MSR or MRP codes in general. But (P)UM codes have advantages in terms of simplicity and applicability. Therefore, they are a good compromise in contrast to the more complex m-MSR or MRP code constructions. An overview of all these codes is given in the table in Appendix A.
Chapter 6 will show comparisons of these constructions. Especially, Section 6.1 will compare the MRD matrix construction to the one used for m-MSR codes. Section 6.2 will present the connection between the truncated column sum rank upper bound and the generalised Singleton bound.

Chapter 6

Comparison of Constructions and Bounds

In this chapter, we will take a closer look at the constructions of the previous chapter. In the first Section 6.1, we will compare the two different matrix constructions and the prominent representatives of those: the m-MSR (see Section 5.1) and the MRD (see Section 5.3) codes. The later Section 6.2 will describe the upper bounds of the free sum rank and truncated sum rank distance. Additionally, it will contrast these two and will show whether one implies the other.

6.1 Juxtaposition of MRD and m-MSR Convolutional Codes

MRD [NPRV17] and m-MSR [MBK16] convolutional codes aim to maximise the sum rank distance to guarantee good decoding properties. On the one hand, the MRD code maximises the overall free sum rank distance and uses a rowmat function to create matrices. On the other hand, m-MSR codes use the isomorphism between $\mathbb{F}_{q^m}^n$ and $\mathbb{F}_q^{m\times n}$ to construct a rank metric code and reach the upper bound for the truncated distance up to the memory μ instead of the free sum rank distance. Table 6.1 summarises these features.

Criterions	m-MSR	MRD convolutional code
Code subfield	$\mathcal{C} \subseteq \mathbb{F}_{q^m}[D]^n$	$\mathcal{C} \subseteq \mathbb{F}_q[D]^{mn}$
Code length	n	mn
Parameter constraint	$m \geq n$	$m \geq n$
Construction of matrix	$\phi:\ \mathbb{F}_{q^m}[D]^n \to \mathbb{F}_q[D]^{m\times n}$	rowmat: $\mathbb{F}_q[D]^{mn} \to \mathbb{F}_q[D]^{n\times m}$
Bound	$d_{SR}^{\mu}(\mathcal{C}) =$ $(\mu+1)(n-k)+1$	$d_{\overline{SR}}(\mathcal{C}) =$ $n\left(\left\lfloor\frac{\delta}{k}\right\rfloor+1\right)-\left\lceil\frac{k\left(\left\lfloor\frac{\delta}{k}\right\rfloor+1\right)-\delta}{m}\right\rceil+1$

TABLE 6.1: Comparison of m-MSR and MRD codes.

In this section, we will examine and compare the different structures and measures. First, we will investigate whether the different code structures can define the same rank metric code. This means we will try to find a corresponding $\mathcal{C}_2 \subseteq \mathbb{F}_q[D]^{mn}$ for each $\mathcal{C}_1 \subseteq \mathbb{F}_{q^m}[D]^n$ and the other way round where $\mathcal{C}_1$ and $\mathcal{C}_2$ produce the same set of matrices. That would answer the question whether the two code structures are freely interchangeable or if there are fundamental differences. Second, the differences in the bounds will be analysed. Since we work in the context of sliding window network models, we are interested in the j-th column sum rank distance of the MRD codes.

Remark 6.1.1. One can observe that Napp et al. [NPRV17] construct $n\times m$ matrices, whereas m-MSR codes are $m \times n$ matrices even though, both have the same constraint of $m \geq n$. Additionally, the rank of a transposed matrix is equal to the original matrix. However, transposing a matrix after applying $\text{rowmat}_{n\times m}$ does not result in the same matrix as when we use $\text{rowmat}_{m\times n}$ directly. But we are allowed to redefine ψ in the definition of Napp et al. (see Section 5.3) so that these differences will cancel out. This implies that we can compare both sets of matrices by redefining ψ in the definition of Napp et al. and transposing one of them. This will be used interchangeably in this section. For that reason, the $\text{rowmat}_{m\times n}$ function instead of the $\text{rowmat}_{n\times m}$ will be used.

6.1.1 Code Structures

This subsection will compare the approaches to construct a code over $\mathbb{F}_{q^m}[D]^n$ or $\mathbb{F}_q[D]^{mn}$.

Equalities on Rank Metric Block Codes

We will first prove that for each code $\mathcal{C}_1$ over $\mathbb{F}_{q^m}^n$, there exists a suitable code $\mathcal{C}_2$ in $\mathbb{F}_q^{mn}$.

Definition 6.1.2. Let $f(x)$ be a function over a field $\mathbb{F}$ and $\mathcal{C} \subseteq \mathbb{F}$ a code, then

$$f(\mathcal{C}) = \{f(c) | \forall c \in \mathcal{C}\}.$$

Definition 6.1.3. (Matrix Equal Codes) Let $\mathcal{C}_1 \subseteq \mathbb{F}_{q^m}^n$ and $\mathcal{C}_2 \subseteq \mathbb{F}_q^{mn}$. $\mathcal{C}_1$ and $\mathcal{C}_2$ are matrix equal if

$$\phi(\mathcal{C}_1) = \text{rowmat}_{m\times n}(\mathcal{C}_2).$$

Theorem 6.1.4. For each $\mathcal{C}_1 \subseteq \mathbb{F}_{q^m}^n$ there exists a $\mathcal{C}_2 \subseteq \mathbb{F}_q^{mn}$ such that $\mathcal{C}_1$ and $\mathcal{C}_2$ are matrix equal. Moreover, if the code $\mathcal{C}_1$ has dimension k then $\mathcal{C}_2$ has dimension $m \cdot k$.

Proof. Let $b_1, b_2, \dots b_m \in \mathbb{F}_{q^m}$ be a basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$, $G_1 \in \mathbb{F}_{q^m}^{k\times n}$ be the generator matrix of $\mathcal{C}_1$ and $r_1, \dots, r_k$ be the row vectors of G_1. We will now give a construction of the generator matrix G_2 of $\mathcal{C}_2$. We transform all r_i with $i \leq k$ to matrices in $\mathbb{F}_q^{m\times n}$ and afterwards to vectors in $\mathbb{F}_q^{m\cdot n}$. So, we get the following set:

$$\{\text{rowvec}(\phi(r_i)) \mid i \leq k\} = S$$

Since we work now over $\mathbb{F}_q$ and not over $\mathbb{F}_{q^m}$ anymore, $\text{span}_q(\text{rowmat}(S)) \subset \phi(\mathcal{C}_1)$. So, some matrices are missing and S is not enough to generate all matrices in $\phi(\mathcal{C}_1)$. To overcome this fact, we construct the following new set:

$$T = \{\text{rowvec}(\phi(b_j \cdot r_i)) \mid \forall i \leq k, j \leq m\}$$

G_2 is now constructed by taking each element of T as a row of G_2 and $|T| = m \cdot k$. Since the r_i are rows of G_1, they are linearly independent. Furthermore, $b_1, b_2, \dots b_m \in \mathbb{F}_{q^m}$ is a basis of $\mathbb{F}_{q^m}$ and all elements of T are linearly independent over $\mathbb{F}_q$. This shows that $\mathcal{C}_2$ has the dimension $m \cdot k$ and hence $G_2 \in \mathbb{F}_q^{k\cdot m\times m\cdot n}$. By construction it is clear that $\text{rowmat}(\mathcal{C}_2) \subseteq \phi(\mathcal{C}_1)$. For the other inclusion, we can compare the cardinalities. $\mathcal{C}_1$ contains $(q^m)^k$ elements and $\mathcal{C}_2$ consists of $q^{k\cdot m}$ elements. Since $\text{rowmat}(\mathcal{C}_2)$ and $\phi(\mathcal{C}_1)$ are $\mathbb{F}_q$ linear, $\text{rowmat}(\mathcal{C}_2) \subseteq \phi(\mathcal{C}_1)$ and as the cardinalities are equal, it follows that $\text{rowmat}(\mathcal{C}_2) = \phi(\mathcal{C}_1)$. □

We will now give an example of the constructive proof above.

Example 6.1.5. First, we work on a code $\mathcal{C}_1$ over $\mathbb{F}_{2^3}^2$ with dimension $k = 1$, $n = 2$. Remark that $x^3 + x + 1$ is irreducible over $\mathbb{F}_2$:

$$\mathbb{F}_{2^3} \cong \frac{\mathbb{F}_2[x]}{x^3 + x + 1} \cong \{a + bx + cx^2 | a, b, c \in \mathbb{F}_2\}$$

Let α be a zero of $x^3 + x + 1$ and the elements in $\mathbb{F}_{2^3}$ are the following:

0	α^2	$\alpha^5 = 1 + \alpha + \alpha^2$
1	$\alpha^3 = \alpha + 1$	$\alpha^6 = 1 + \alpha^2$
α	$\alpha^4 = \alpha^2 + \alpha$	$\alpha^7 = 1$

We choose $\{1, \alpha, \alpha^2\}$ as a basis over $\mathbb{F}_2$. Consider the following matrix as the generator of $\mathcal{C}_1$:

$$\mathbb{F}_{2^3} \to \mathbb{F}_{2^3}^2$$
$$x \mapsto x \cdot \begin{bmatrix} \alpha & \alpha^2 \end{bmatrix}$$

It results in the following codewords over $\mathbb{F}_{2^3}^2$:

$(0, 0)$	$(\alpha^2, 1 + \alpha)$	$(\alpha + \alpha^2, 1 + \alpha + \alpha^2)$	$(1 + \alpha^2, 1)$
(α, α^2)	$(1 + \alpha, \alpha + \alpha^2)$	$(1 + \alpha + \alpha^2, 1 + \alpha^2)$	$(1, \alpha)$

And we get the following matrices:

$$\begin{bmatrix}0&0\\0&0\\0&0\end{bmatrix} \quad \begin{bmatrix}0&1\\0&1\\1&0\end{bmatrix} \quad \begin{bmatrix}0&1\\1&1\\1&1\end{bmatrix} \quad \begin{bmatrix}1&1\\0&0\\1&0\end{bmatrix}$$

$$\begin{bmatrix}0&0\\1&0\\0&1\end{bmatrix} \quad \begin{bmatrix}1&0\\1&1\\0&1\end{bmatrix} \quad \begin{bmatrix}1&1\\1&0\\1&1\end{bmatrix} \quad \begin{bmatrix}1&0\\0&1\\0&0\end{bmatrix}$$

Then, we construct G_2. For that, we consider $r_1 = (\alpha, \alpha^2)$ since $k = 1$. Next, r_1 is multiplied by the elements of the basis and converted to a row vector in $\mathbb{F}_2$, i.e., $\text{rowvec}(\phi(1 \cdot r_1))$, $\text{rowvec}(\phi(\alpha \cdot r_1))$ and $\text{rowvec}(\phi(\alpha^2 \cdot r_1))$. This implies that:

$$T = \{(0,0,1,0,0,1), (0,1,0,1,1,0), (1,0,1,1,0,1)\}$$

Using this we create the generator matrix for $\mathcal{C}_2$:

$$\begin{aligned} \mathbb{F}_2^3 &\to \mathbb{F}_2^{3\cdot 2} \\ (x,y,z) &\mapsto (x,y,z) \cdot \begin{bmatrix}0&0&1&0&0&1\\0&1&0&1&1&0\\1&0&1&1&0&1\end{bmatrix} \end{aligned}$$

This generates the following codewords:

$$\begin{array}{ll} (0,0,0,0,0,0) & (0,1,1,1,1,1) \\ (0,0,1,0,0,1) & (1,1,1,0,1,1) \\ (0,1,0,1,1,0) & (1,1,0,0,1,0) \\ (1,0,1,1,0,1) & (1,0,0,1,0,0) \end{array}$$

We realise that both codes have the same matrices. Also, the cardinalities are equal since $(2^3)^1 = 2^{(3\cdot 1)}$. This implies that:

$$\phi(\mathcal{C}_1) = \text{rowmat}(\mathcal{C}_2)$$

The other direction is not true in general. We can observe this by the following example:

Example 6.1.6. Consider the $\mathcal{C}_2$ code from the example above. However, in this example, we delete one of the generator vectors. For example, we can delete the first one, $(0,0,1,0,0,1)$, and get a new generator matrix:

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

With that, we obtain a $\mathcal{C}_2'$ of dimension 2. This implies $\mathcal{C}_2'$ has cardinality 4. We now try to find a code over $\mathbb{F}_{2^3}^2$ such that the matrices are equal. Each linear block code over $\mathbb{F}_{2^3}^2$ has cardinality $(2^3)^i$ for some dimension $i \in \mathbb{N}$. Hence, we get a contradiction since there does not exist any $i \in \mathbb{N}$ such that $(2^3)^i = 4$ ↯
To conclude, there exists no code $\mathcal{C}_1$ over $\mathbb{F}_{2^3}^2$ such that $\phi(\mathcal{C}_1) = \text{rowmat}(\mathcal{C}_2')$.

One could now come up with the conjecture that for each $\mathcal{C}_2$ over $\mathbb{F}_q^{m\cdot n}$ with the dimension

k exists a code $\mathcal{C}_1$ over $\mathbb{F}_{q^m}^n$ if m divides k. However, this does not hold in general as we can see in the next example:

Example 6.1.7. We recall the Example 6.1.5 and start with a $\mathcal{C}_2 \subset \mathbb{F}_2^{3\cdot 2}$ and dimension $k = 3$. We take almost the same generator matrix as before but exchanging the first generator vector with one which is not contained in the code of the Example 6.1.5. The new added generator vector is $(1, 0, 0, 0, 0, 0)$ which results in the following new generator matrix:

$$G_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

According to the cardinality, we try to find a $\mathcal{C}_1 \in \mathbb{F}_{2^3}^2$ with dimension 1 which is matrix equal to $\mathcal{C}_2 = \text{Im}(G_2)$. $a = (1, 0, 0, 0, 0, 0)$ and $b = (0, 1, 0, 1, 1, 0)$ are in $\mathcal{C}_2$. $\phi^{-1}(\text{rowmat}(a)) = (1, 0)$ and $\phi^{-1}(\text{rowmat}(b)) = (\alpha^2, \alpha+1)$ are representations of a, b in $\mathcal{C}_1$. Since $\alpha^2 \cdot (1, 0) = (\alpha^2, 0) \neq (\alpha^2, \alpha+1)$, it follows that $(1, 0)$ and $(\alpha^2, \alpha+1)$ are linearly independent over $\mathbb{F}_{2^3}$. This implies that $\mathcal{C}_1$ has a dimension bigger than 1. Thus, $|\mathcal{C}_1| \neq |\mathcal{C}_2|$ and we have a contradiction ↯
This proves that there does not exist a matrix equal code for our $\mathcal{C}_2$ even though $k = 3$ divides $m = 3$.

Theorem 6.1.8. (Singleton Bound for Matrix Equal Codes) Assume we have two matrix equal codes $\mathcal{C}_1(n, k) \subseteq \mathbb{F}_{q^m}^n$ and $\mathcal{C}_2(mn, k') \subseteq \mathbb{F}_q^{mn}$. If one of the two codes achieves the upper bound on the sum rank free distance (see Theorem 2.2.4 and Theorem 5.3.4), the other reaches it as well. In other words:

$$d_{SR}(\mathcal{C}_1) = n - k + 1 \iff d_{\overline{SR}}(\mathcal{C}_2) = n - \left\lceil \frac{k'}{m} \right\rceil + 1$$

Proof. Since the cardinalities of two matrix equal codes are equal, it follows that $k' = k \cdot m$. Using this, we can calculate

$$n - \left\lceil \frac{k'}{m} \right\rceil + 1 = n - \left\lceil \frac{km}{m} \right\rceil + 1 = n - k + 1$$

and both bounds are equal. □

Equalities on Rank Metric Convolutional Codes

In this subsection, we will observe matrix equal codes in the context of convolutional codes. The first definitions and theorems are almost the same as for block codes. We will shortly recap them and then inspect the specific convolutional code properties in combination with matrix equal codes.

Definition 6.1.9. (Matrix Equal Codes for Convolutional Codes) Let $\mathcal{C}_1 \subseteq \mathbb{F}_{q^m}[D]^n$ and $\mathcal{C}_2 \subseteq \mathbb{F}_q[D]^{mn}$. $\mathcal{C}_1$ and $\mathcal{C}_2$ are matrix equal if

$$\phi(\mathcal{C}_1) = \text{rowmat}_{m\times n}(\mathcal{C}_2).$$

Theorem 6.1.10. For each $\mathcal{C}_1 \subseteq \mathbb{F}_{q^m}[D]^n$, there exists a $\mathcal{C}_2 \subseteq \mathbb{F}_q[D]^{mn}$ such that $\mathcal{C}_1$ and $\mathcal{C}_2$ are matrix equal. Moreover, if the Code $\mathcal{C}_1$ has dimension k then $\mathcal{C}_2$ has dimension $m \cdot k$.

Proof. We will prove an even stronger result in Theorem 6.1.14. □

Although the proof works similar to the one in Theorem 6.1.4, we will again give a small example to illustrate the idea:

Example 6.1.11. We extend the examples above. Let $\mathcal{C}_1 \subseteq \mathbb{F}_{2^3}[D]^2$ with generator matrix $\begin{bmatrix}\alpha D & \alpha^2\end{bmatrix}$.
We multiply the row of the generator matrix by 1, α and α^2 and we obtain:

$$(\alpha D, \alpha^2) \qquad (\alpha^2 D, 1+\alpha) \qquad (D+\alpha D, \alpha+\alpha^2)$$

This results in the following matrices

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

which can be transformed into the corresponding vectors

$$(0,0,D,0,0,1) \qquad (0,1,0,1,D,0) \qquad (D,0,D,1,0,1).$$

These vectors can be stacked to the new generator matrix of the code $\mathcal{C}_2 \subseteq \mathbb{F}_2[D]^{3\cdot 2}$:

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

Remark 6.1.12. As in the case of block codes, we cannot give a general theorem about the existence of a matrix equal code $\mathcal{C}_1 \subseteq \mathbb{F}_{q^m}[D]^n$ which matches a given $\mathcal{C}_2 \subseteq \mathbb{F}_q[D]^{m\cdot n}$.

Remark 6.1.13. The example above shows that not only the dimension changes during the transformation from a code over $\mathbb{F}_{q^m}[D]^n$ to one over $\mathbb{F}_q[D]^{m\cdot n}$ but also the degree of the code changes. While $\mathcal{C}_1$ has degree 1, the new code $\mathcal{C}_2$ has degree 3. The degree of $\mathcal{C}_2$ can be seen with the corresponding leading coefficient matrix:

$$[G_2(D)]_h = \begin{bmatrix}0 & 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 1 & 0\\ 1 & 0 & 1 & 0 & 0 & 0\end{bmatrix}$$

This matrix has full row rank. By Lemma 2.1.29, it follows that the generator matrix is minimal, implying that the degree is equal to the sum of the row degrees, which is 3.
Therefore, we formulate a stronger version of Theorem 6.1.10:

Theorem 6.1.14. For each $\mathcal{C}_1 \subseteq \mathbb{F}_{q^m}[D]^n$ there exists a $\mathcal{C}_2 \subseteq \mathbb{F}_q[D]^{mn}$ such that $\mathcal{C}_1$ and $\mathcal{C}_2$ are matrix equal. Moreover, if code $\mathcal{C}_1$ has dimension k and *degree* δ, then $\mathcal{C}_2$ has dimension $m \cdot k$ and *degree* $m \cdot \delta$.

Proof. The proof starts in the same way as in Theorem 6.1.4. Remember that in the current theorem, we work with submodules with the variable D. Let $b_1, b_2, \dots b_m \in \mathbb{F}_{q^m}$ be a basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$, $G_1(D) \in \mathbb{F}_{q^m}[D]^{k\times n}$ be the generator matrix of $\mathcal{C}_1$ and $r_1(D), \dots, r_k(D)$ be the row vectors of $G_1(D)$. We can assume that $G_1(D)$ is minimal (see Theorem 2.1.27). Let ν_i be the row degree of row $r_i(D)$. Minimality implies the fact that the sum of all row degrees is equal to the degree of the convolutional code:

$$\sum_{i=1}^{k} \nu_i = \nu = \delta$$

We will now give a construction of the generator matrix $G_2(D) \in \mathbb{F}_q[D]^{k\cdot m\times m\cdot n}$ of $\mathcal{C}_2$. We transform all $r_i(D)$, $i \le k$ to matrices in $\mathbb{F}_q[D]^{m\times n}$ and further to vectors in $\mathbb{F}_q[D]^{m\cdot n}$. So,

we get the following set:

$$S = \{\text{rowvec}(\phi(r_i(D))) \mid i \leq k\}$$

Since we work now over $\mathbb{F}_q[D]$ and not over $\mathbb{F}_{q^m}[D]$ anymore, $\text{span}_q(\text{rowmat}(S)) \subset \phi(\mathcal{C}_1)$. So, some matrices are missing and S is not enough to generate all matrices in $\phi(\mathcal{C}_1)$. To overcome this fact, the following new set is constructed:

$$T = \{\text{rowvec}(\phi(b_j \cdot r_i(D))) \mid \forall i \leq k, j \leq m\}$$

$G_2(D)$ is now constructed by taking each element of T as a row of $G_2(D)$ and $|T| = m \cdot k$. Since the $r_i(D)$ are rows of $G_1(D)$, they are linearly independent. Further, $b_1, b_2, \dots b_m \in \mathbb{F}_{q^m}$ is a basis of $\mathbb{F}_{q^m}$ and all elements of T are linearly independent over $\mathbb{F}_q$. This shows that $\mathcal{C}_2$ has dimension $m \cdot k$. By construction, it is clear that $\text{rowmat}(\mathcal{C}_2) \subseteq \phi(\mathcal{C}_1)$. For the other inclusion, we can compare the dimensions. $\mathcal{C}_1$ has dimension k over $\mathbb{F}_{q^m}$ and $\mathcal{C}_2$ has dimension $k \cdot m$ over $\mathbb{F}_q$. Since $\text{rowmat}(\mathcal{C}_2)$ and $\phi(\mathcal{C}_1)$ are $\mathbb{F}_q$ linear, $\text{rowmat}(\mathcal{C}_2) \subseteq \phi(\mathcal{C}_1)$ and the dimensions are equal over $\mathbb{F}_q$. It follows that $\text{rowmat}(\mathcal{C}_2) = \phi(\mathcal{C}_1)$.
Since $G_1(D)$ is a minimal generator matrix, it follows that the corresponding leading coefficient matrix $[G_1(D)]_h$ has full row rank by Lemma 2.1.29. Since each row $t_i(D)$ of $[G_1(D)]_h$ is linearly independent, it follows that each $\text{rowvec}(\phi(t_i))$ is linearly independent as well. The other row vectors in $[G_2(D)]_h$, $\text{rowvec}(\phi(b_j \cdot t_i(D)))$, are also linearly independent over $\mathbb{F}_q$ since they are multiplied by a basis element of $\mathbb{F}_{q^m}$. This implies that $[G_2(D)]_h$ has full row rank and $G_2(D)$ is minimal. Now, we can count the row degrees of the new generator matrix $G_2(D)$. Consider an arbitrary row $r_j(D)$ of $G_1(D)$. $r_j(D)$ appears also in a transformed way in $G_2(D)$ (compare set T). This new row $r_j^i(D) = \text{rowvec}(\phi(b_i \cdot r_j(D)))$ in $G_2(D)$ has the same row degree as the original row. Each row of $G_1(D)$ occurs m different times (multiplied by every basis element: $r_j^1(D), \dots, r_j^m(D)$) in $G_2(D)$. Since we get the m new rows by multiplying with a basis scalar, every new row has the same row degree as the original one. This verifies the following calculation for the degree δ' of $G_2(D)$:

$$\delta' = \sum_{i=1}^{k \cdot m} \nu_i' = \sum_{i=1}^{k} m \cdot \nu_i = m \cdot \sum_{i=1}^{k} \nu_i = m \cdot \nu = m \cdot \delta$$

This shows that the degree of $G_2(D)$ is $m \cdot \delta$. □

Theorem 6.1.15. (Singleton Bound for Matrix Equal Codes) Assume there are two matrix equal codes $\mathcal{C}_1(n, k, \delta) \subseteq \mathbb{F}_{q^m}^n$ and $\mathcal{C}_2(mn, k', \delta' = \delta m) \subseteq \mathbb{F}_q^{mn}$. If one of the two codes achieves the upper bound on the sum rank free distance (see Theorem 2.2.10 and Theorem 5.3.7), the other achieves it as well. In other words:

$$d_{SR}(\mathcal{C}_1) = (n-k)\left(\left\lfloor \frac{\delta}{k} \right\rfloor + 1\right) + \delta + 1$$

$$\iff$$

$$d_{\overline{SR}}(\mathcal{C}_2) = n\left(\left\lfloor \frac{\delta'}{k'} \right\rfloor + 1\right) - \left\lceil \frac{k'\left(\left\lfloor \frac{\delta'}{k'} \right\rfloor + 1\right) - \delta'}{m} \right\rceil + 1$$

Proof. It is assumed that $\delta' = \delta m$. Due to the dimension constraint, we have $k' = km$. Now, we insert these equalities in the given formula:

$$\begin{aligned}
d_{\overline{SR}}(\mathcal{C}_2) &= n\left(\left\lfloor \frac{\delta'}{k'} \right\rfloor + 1\right) - \left\lceil \frac{k'\left(\left\lfloor \frac{\delta'}{k'} \right\rfloor + 1\right) - \delta'}{m} \right\rceil + 1 \\
&= n\left(\left\lfloor \frac{\delta m}{km} \right\rfloor + 1\right) - \left\lceil \frac{km\left(\left\lfloor \frac{\delta m}{km} \right\rfloor + 1\right) - \delta m}{m} \right\rceil + 1 \\
&= n\left(\left\lfloor \frac{\delta}{k} \right\rfloor + 1\right) - \left\lceil \frac{km\left(\left\lfloor \frac{\delta}{k} \right\rfloor + 1\right) - \delta m}{m} \right\rceil + 1 \\
&= n\left(\left\lfloor \frac{\delta}{k} \right\rfloor + 1\right) - \left\lceil \frac{k\left(\left\lfloor \frac{\delta}{k} \right\rfloor + 1\right) - \delta}{1} \right\rceil + 1 \\
&= (n-k)\left(\left\lfloor \frac{\delta}{k} \right\rfloor + 1\right) + \delta + 1 \\
&= d_{SR}(\mathcal{C}_1)
\end{aligned}$$

This shows that both bounds are the same and equality is proven. □

Napp et al. [NPRV17] present a specific MRD construction. We are now interested if we can find matrix equal codes to these MRD codes. Recall Theorem 5.3.12 of this construction. Pay attention that $\text{rowmat}_{m\times n}$ is used to obtain comparable matrices. It states the condition that $m \geq \delta + k$ and $m \geq n$. If we consider only convolutional codes, then $\delta \geq 1$. This implies that $m > k$. The dimension of the MRD code is k over $\mathbb{F}_q$ and a corresponding code over $\mathbb{F}_{q^m}[D]^n$ has dimension k'. It should hold that $(q^m)^{k'} = q^k$, implying that $k' = \frac{k}{m}$. Otherwise, the dimension constraint is not satisfied. Since $k' \in \mathbb{N}$ and $m > k$, this is not possible and no corresponding matrix equal code over $\mathbb{F}_{q^m}[D]^n$ exists.

The above theory shows that codes of the form $\mathcal{C}_1 \subset \mathbb{F}_{q^m}[D]^n$ and $\mathcal{C}_2 \subset \mathbb{F}_q[D]^{mn}$ do not generate the same set of matrices in general. While for each $\mathcal{C}_1 \subset \mathbb{F}_{q^m}[D]^n$, there exists a corresponding code $\mathcal{C}_2 \subset \mathbb{F}_q[D]^{mn}$, it does not hold for the other direction. If there are two matrix equal codes, the dimension δ and degree k differ by a factor of m. There are structural differences between the two constructions. m-MSR codes are $\mathbb{F}_{q^m}$ linear, whereas the MRD codes are $\mathbb{F}_q$ linear. This shows that there exist more codes over $\mathbb{F}_q[D]^{mn}$ since they have the weaker condition to be $\mathbb{F}_q$ linear instead of $\mathbb{F}_{q^m}$ linear. Additionally, by Byrne and Ravagnani [RB19], MRD $\mathbb{F}_{q^m}$-linear block codes are dense in the set of all $\mathbb{F}_{q^m}$-linear block codes. However, MRD $\mathbb{F}_q$-linear block codes are not dense in the set of all $\mathbb{F}_q$-linear block codes. Nevertheless, it is more intuitive to use a code over $\mathbb{F}_{q^m}[D]^n$ for network coding since we can define an element of $\mathbb{F}_{q^m}$ as a packet. In contrast, in a code over $\mathbb{F}_q[D]^{mn}$, a packet is defined as a vector in $\mathbb{F}_q^m$. In spite of all differences, both codes are useful and feasible for network coding.

6.1.2 Decoding Properties

We are interested in the decoding property of an MRD code in context of random linear network coding. For that reason, we would like to determine if an MRD code has a good truncated sum rank distance. The first approach was to transform the MRD codes into better-known codes in the field $\mathbb{F}_{q^m}^n$. Unfortunately, the study of matrix equal codes shows that there exist no such equal codes for the MRD code constructed by Napp et al. [NPRV17].

In this subsection, we will have a closer look at the MRD codes and the j-th column sum rank distance of them.
Let $\mathcal{C} \subset \mathbb{F}_q[D]^{mn}$. We obtain for $c = c_0 + c_1 D + c_2 D^2 + \cdots \in \mathbb{F}_q[D]^{mn}$ the free distance

$$d_{\overline{SR}}(\mathcal{C}) = \min_{c \in \mathcal{C} \setminus \{0\}} \left\{ \sum_{i \in \mathbb{N}} \operatorname{rank}(\operatorname{rowmat}(c_i)) \right\}$$

and the j-th column sum rank distance

$$d^j_{\overline{SR}}(\mathcal{C}) = \min_{\substack{c \in \mathcal{C} \\ c_0 \neq 0}} \left\{ \sum_{i=0}^{j} \operatorname{rank}(\operatorname{rowmat}(c_i)) \right\}.$$

Since the sum in the truncated distance is bounded, we directly oberserve that

$$d^j_{\overline{SR}}(\mathcal{C}) \leq d_{\overline{SR}}(\mathcal{C}).$$

A good measure for decodability in a sliding window network model is the j-th column sum rank distance. We will first deduce an upper bound for the j-th column sum rank distance. It is known from Theorem 2.1.48 that for the Hamming weight and a $\mathcal{C}(n, k, \delta)$ code, the following holds:

$$d^j_H(\mathcal{C}) \leq (n-k)(j+1) + 1$$

Adapting this to a $\mathcal{C}(m \times n, k, \delta)$ code results in:

$$d^j_H(\mathcal{C}) \leq (nm-k)(j+1) + 1$$

Since the sum rank distance is upper bounded by the Hamming distance, this leads to the following upper bound:

$$d^j_{\overline{SR}}(\mathcal{C}) \leq (nm-k)(j+1) + 1 \tag{6.1}$$

We will now check whether this bound is optimal. Remark that we cannot use the theory of codes over $\mathbb{F}_{q^m}[D]^n$ since the two constructions are different. For the same reason, it is not allowed to use the maximum $L = \left\lfloor \frac{\delta}{k} \right\rfloor + \left\lfloor \frac{\delta}{n-k} \right\rfloor$ bound for the j-th column sum rank distance. Also, it is not guaranteed that the bound (6.1) is optimal and that it can be achieved.

We will look at another possible bound. Recap how a codeword is formed: $c = c_0 + c_1 D + c_2 D^2 + \cdots \in \mathbb{F}_q[D]^{mn}$. For each c_i holds $\operatorname{rank}(\operatorname{rowmat}(c_i)) \leq n$. This allows the following calculations:

$$d^j_{\overline{SR}}(\mathcal{C}) = \min_{\substack{c \in \mathcal{C} \\ c_0 \neq 0}} \left\{ \sum_{i=0}^{j} \operatorname{rank}(\operatorname{rowmat}(c_i)) \right\} \leq \sum_{i=0}^{j} n = n(j+1) \tag{6.2}$$

We compare the two calculated bounds (6.1) (denoted as a) and (6.2) (denoted as b). As long as $nm - k \geq n$, it holds that $b < a$. We can observe this when inserting $nm - k \geq n$ in a, then it follows

$$(nm-k)(j+1) + 1 \geq n(j+1) + 1$$

and the bound in (6.2) is smaller and so more exact.
In the other case, $nm - k < n$

$$(nm-k)(j+1) + 1 < n(j+1) + 1 \Rightarrow (nm-k)(j+1) + 1 \leq n(j+1)$$

and the bound in (6.1) is smaller ($a \leq b$) and determines the upper bound. This results in the following Theorem:

Theorem 6.1.16. Let $\mathcal{C} \subset \mathbb{F}_q[D]^{mn}$, then the j-th column sum rank distance is bounded by

$$d^j_{\overline{SR}}(\mathcal{C}) \leq \begin{cases} n(j+1) & \text{if } nm-k \geq n \\ (nm-k)(j+1)+1 & \text{otherwise.} \end{cases}$$

Remark 6.1.17. The choice of the upper bound is independent of j. Only the code parameters determine which bound outperforms the other.

Back to the MRD construction of Napp et al. [NPRV17].

Recall Theorem 5.3.12. The requirements guarantee that $m \geq n$ and $m \geq \delta + k$. If $\delta > 0$ (in case of a convolutional code), then $m \geq \delta + k > k$. We use these inequalities to determine which of the two bounds in Theorem 6.1.16 holds for this MRD construction.

First for $n \geq 2$:

$$nm - k \geq mn - m = m(n-1) \geq n(n-1) \overset{n\geq 2}{\geq} n$$

For $n = 1$:

$$mn - k = m - k \geq 1 = n$$

This proves that in both cases, the bound (6.2) is meaningful. Next, we determine the j-th column sum rank distance of the MRD codes of Theorem 5.3.12:

Theorem 6.1.18. The MRD codes $\mathcal{C}$ of Theorem 5.3.12 by [NPRV17] have the following j-th column sum rank distance:

$$d^j_{\overline{SR}}(\mathcal{C}) = n(j+1), \quad j \leq \left\lfloor \frac{\delta}{k} \right\rfloor$$

Proof. The proof follows from the calculations of the proof of Theorem 5.3.12 in which $u_0 \neq 0$ is assumed. G_0 has full row rank as $\delta \geq 1$ which implies $m > k$ and the A^l for $0 \leq l \leq m-1$ are linearly independent. We also have $u_0 \neq 0 \Leftrightarrow v_0 \neq 0$, proving that $V_i = \psi(v_i) = XB_i$ has full row rank n for every $0 \leq i \leq \left\lfloor \frac{\delta}{k} \right\rfloor$. Finally, the rank weight of $V(D)$ is

$$d^j_{\overline{SR}}(V(D)) = \sum_{i=0}^{j} \operatorname{rank}(V_i) = n(j+1)$$

for all $j \leq \left\lfloor \frac{\delta}{k} \right\rfloor$. □

This theorem shows that the constructed MRD codes achieve the upper bound of the Theorem 6.1.16 and confirms that this bound is reachable.

We can now compare the j-th column sum rank distance of an m-MSR code $\mathcal{C}_1 \subset \mathbb{F}_{q^m}[D]^n$ with memory μ and an MRD code $\mathcal{C}_2 \subset \mathbb{F}_q[D]^{mn}$. The m-MSR code achieves the following bound:

$$d^j_{SR}(\mathcal{C}_1) = (n-k)(j+1)+1 = n(j+1) - (k(j+1)-1), \qquad j \leq \mu$$

Whereas the MRD code of Theorem 5.3.12 attains:

$$d^j_{\overline{SR}}(\mathcal{C}_2) = n(j+1), \qquad j \leq \left\lfloor \frac{\delta}{k} \right\rfloor$$

Since $k \geq 1$ it follows

$$d^j_{\overline{SR}}(\mathcal{C}_2) > d^j_{SR}(\mathcal{C}_1), \qquad \text{for } 1 \leq j \leq \min\left\{\mu, \left\lfloor \frac{\delta}{k} \right\rfloor\right\}.$$

This calculation shows that MRD codes can have a better j-th column sum rank distance than m-MSR codes. That equips MRD codes with a decoding probability which has a big potential in the context of network coding (see Theorem 4.3.4).

6.1.3 Summary

MRD codes should be seen as a separate class of codes that can be used in network coding. They are not a special version of the well known m-MSR codes but rather a second option. MRD codes use $\mathbb{F}_q[D]$ as a ground field ($\mathbb{F}_q[D]$-linear), whereas m-MSR codes use $\mathbb{F}_{q^m}[D]$ ($\mathbb{F}_{q^m}[D]$-linear). In contrast, MRD codes have length nm whereas m-MSR codes have length n. The upper bound of the j-th column sum rank distance is higher for MRD codes. If we compare the construction by Napp et al. [NPRV17] of an MRD code with an m-MSR code constructed by Mahmood et al. [MBK16], MRD codes achieve this higher upper bound for the j-th column sum rank distance. This shows that the higher bound is achievable and feasible. It seems that MRD codes are more useful than m-MSR codes. However, we have to take the aspects of the network model and the decoding capability into consideration. This requires further research and is not part of this thesis.

6.2 Comparison of the Sum Rank Bounds

This section will highlight the differences between the j-th column sum rank upper bound and the generalised Singleton bound. We will show that these bounds do not imply each other in general. As a consequence, they are two independent properties.

Let $\mathcal{C}(n,k,\delta) \subseteq \mathbb{F}_{q^m}[D]^n$ be a convolutional code. The following bounds on the sum rank distance hold:

a) The generalised Singleton bound in Theorem 2.2.10 for the free sum rank distance:

$$d_{SR}(\mathcal{C}) \leq (n-k)\left(\left\lfloor \frac{\delta}{k} \right\rfloor + 1\right) + \delta + 1$$

b) The j-th column sum rank upper bound in Proposition 2.2.13:

$$d_{SR}^{j}(\mathcal{C}) \leq (n-k)(j+1) + 1$$

This bound can be reached in the best case for $j = L = \left\lfloor \frac{\delta}{k} \right\rfloor + \left\lfloor \frac{\delta}{n-k} \right\rfloor$, i.e.,

$$d_{SR}^{L}(\mathcal{C}) = (n-k)\left(\left\lfloor \frac{\delta}{k} \right\rfloor + 1\right) + 1 + (n-k)\left\lfloor \frac{\delta}{n-k} \right\rfloor.$$

Remark 6.2.1. The upper bound of the truncated column sum rank distance can be reached for L defined above. If we have a systematic generator matrix $G(D) = [I_{k\times k}|P(D)]$, then the upper bound can only be achieved up to $j = \mu$, where μ is the memory of the code (see Proposition 2.2.18).

Recalling the sum rank definition: Let $c = c_0 + c_1 D + c_2 D^2 + \cdots \in \mathbb{F}_{q^m}[D]^n$ be a codeword then

$$d_{SR}(\mathcal{C}) = \min_{c\in\mathcal{C}\setminus\{0\}} \left\{ \sum_{i\in\mathbb{N}} \operatorname{rank}(\phi(c_i)) \right\}$$

and

$$d_{SR}^{j}(\mathcal{C}) = \min_{\substack{c\in\mathcal{C} \\ c_0\neq 0}} \left\{ \sum_{i=0}^{j} \operatorname{rank}(\phi(c_i)) \right\}.$$

We directly see that

$$d_{SR}^{j}(\mathcal{C}) \leq d_{SR}(\mathcal{C})$$

since the sum in the free distance runs over more indices than the truncated one.

Theorem 6.2.2. [GLRS06, Modification of Hamming metric codes, top of p. 587] Let $\mathcal{C}$ be an (n,k,δ) convolutional code which reaches the upper bound of the truncated sum rank distance at $j = L = \left\lfloor \frac{\delta}{k} \right\rfloor + \left\lfloor \frac{\delta}{n-k} \right\rfloor$, i.e.,

$$d_{SR}^{L}(\mathcal{C}) = (n-k)\left(\left\lfloor \frac{\delta}{k} \right\rfloor + 1\right) + 1 + (n-k)\left\lfloor \frac{\delta}{n-k} \right\rfloor. \tag{6.3}$$

Then, it follows that $\mathcal{C}$ reaches the generalised Singleton bound

$$d_{SR}(\mathcal{C}) = (n-k)\left(\left\lfloor \frac{\delta}{k} \right\rfloor + 1\right) + \delta + 1$$

if $(n-k)$ divides δ.

Proof. We can assume that $(n-k)|\delta$ and $\mathcal{C}$ fulfils the Equation (6.3). It follows that:

$$\begin{aligned} d_{SR}^{L}(\mathcal{C}) &= (n-k)\left(\left\lfloor\frac{\delta}{k}\right\rfloor+1\right)+1+(n-k)\left\lfloor\frac{\delta}{n-k}\right\rfloor \\ &= (n-k)\left(\left\lfloor\frac{\delta}{k}\right\rfloor+1\right)+1+\delta \end{aligned}$$

Since $d_{SR}^{L}(\mathcal{C}) \leq d_{SR}(\mathcal{C})$ and $d_{SR}(\mathcal{C}) \leq (n-k)\left(\left\lfloor\frac{\delta}{k}\right\rfloor+1\right)+\delta+1$ by the Singleton bound, we deduce

$$d_{SR}(\mathcal{C}) = (n-k)\left(\left\lfloor\frac{\delta}{k}\right\rfloor+1\right)+1+\delta.$$

□

Remark 6.2.3. If $(n-k) \nmid \delta$ then we have according to the proof above:

$$\begin{aligned} d_{SR}^{L}(\mathcal{C}) &= (n-k)\left(\left\lfloor\frac{\delta}{k}\right\rfloor+1\right)+1+(n-k)\left\lfloor\frac{\delta}{n-k}\right\rfloor \\ &< (n-k)\left(\left\lfloor\frac{\delta}{k}\right\rfloor+1\right)+1+\delta \end{aligned}$$

And the theorem does not hold in general.

To see what happens in the cases which are not covered by Theorem 6.2.2, we can look at some examples over $\mathbb{F}_{2^3}^2$. The field is constructed by the irreducible polynomial x^3+x+1 and α is a zero of that polynomial. For the map ϕ from a vector over $\mathbb{F}_{2^3}^n$ to a matrix in $\mathbb{F}_2^{3\times n}$, $\{1, \alpha, \alpha^2\}$ is chosen as the basis.

Example 6.2.4. (Systematic Encoder, μ-th Bound Fulfilled but not the Singleton Bound)
Let $G(D) = [1, \alpha + \alpha^2 D]$ be the systematic generator matrix of $\mathcal{C}$. This code has memory $\mu = 1$, $n = 2$ and $k = 1$. Since there are only two full-size minors, 1 and $\alpha + \alpha^2 D$ with the maximum degree of 1, it follows $\delta = 1$.

Let us check the distance of $\mathcal{C}$. We want to determine the truncated sum rank of length 1. So, we consider codewords generated by u_0 and $u_0 + u_1 D$ with $u_0, u_1 \in \mathbb{F}_{2^3}$ and $u_0, u_1 \neq 0$:

$$u_0 \cdot G(D) = [u_0, \alpha u_0 + \alpha^2 u_0 D] = [u_0, \alpha u_0] + [0, \alpha^2 u_0]D$$

Since $u_0 \neq 0$, u_0 and αu_0 are linearly independent and we have $wt_{SR}^{1}(u_0 \cdot G(D)) = 3$.

$$\begin{aligned} (u_0 + u_1 D) \cdot G(D) &= [u_0 + u_1 D, \alpha u_0 + \alpha u_1 D + \alpha^2 u_0 D + \alpha^2 u_1 D^2] \\ &= [u_0, \alpha u_0] + [u_1, \alpha u_1 + \alpha^2 u_0]D + [0, \alpha^2 u_1]D^2 \end{aligned}$$

As above, u_0 and αu_0 remain linearly independent. $\alpha u_1 + \alpha^2 u_0$ can reach zero for a specific choice of u_0 and u_1. Nevertheless, $u_0, u_1 \neq 0$ which implies $wt_{SR}^{1}([u_0, \alpha u_0] + [u_1, \alpha u_1 + \alpha^2 u_0]D) \geq 3$. The sum rank of $(u_0 + u_1 D) \cdot G(D)$ is greater than or equal to 4 (without truncated at length 1).

We do not consider input polynomials of degree 2 or higher, which would be greater than the code memory, since this would not affect the μ-th column sum rank distance. To see this

fact, we analyse the input polynomial $u_0 + u_1D + \tilde{u}$, where $\tilde{u} = \tilde{u}_2D^2 + \tilde{u}_3D^3 + \cdots \in \mathbb{F}_{2^3}[D]$:

$$\begin{aligned}(u_0 + u_1D + \tilde{u}) \cdot G(D) &= [u_0 + u_1D + \tilde{u}, \alpha u_0 + \alpha u_1 D + \alpha^2 u_0 D + \alpha^2 u_1 D^2 + \alpha\tilde{u} + \alpha^2 D\tilde{u}] \\ &= [u_0, \alpha u_0] + [u_1, \alpha u_1 + \alpha^2 u_0]D + \tilde{B}\end{aligned}$$

$\tilde{B} \in \mathbb{F}_{2^3}[D]^2$ with terms only of a degree higher than 1. We recognise that the first two codewords remain unchanged compared to $(u_0 + u_1D) \cdot G(D)$. That implies that an input of a higher degree does not change the first two codewords and as a consequence, $wt^1_{SR}((u_0 + u_1D) \cdot G(D)) \leq wt^1_{SR}((u_0 + u_1D + \tilde{u}) \cdot G(D))$.

The 1st column distance is 3 by taking the minimum of the calculations above. This results in $d^1_{SR}(\mathcal{C}) = 3$.

We only have to consider input polynomials of degree 1 or less to calculate the free sum rank distance because the memory of the code is 1. Consider $u_0 + u_1D + \tilde{u}$, where $\tilde{u} = \tilde{u}_2D^2 + \tilde{u}_3D^3 + \cdots \in \mathbb{F}_{2^3}[D]$ as above:

$$\begin{aligned}(u_0 + u_1D + \tilde{u}) \cdot G(D) &= [u_0 + u_1D + \tilde{u}, \alpha u_0 + \alpha u_1 D + \alpha^2 u_0 D + \alpha^2 u_1 D^2 + \alpha\tilde{u} + \alpha^2 D\tilde{u}] \\ &= [u_0, \alpha u_0] + [u_1, \alpha u_1 + \alpha^2 u_0]D + [\tilde{u}_2, \alpha\tilde{u}_2 + \alpha^2 u_1]D^2 + \tilde{C}\end{aligned}$$

$\tilde{C} \in \mathbb{F}_{2^3}[D]^2$ with terms only of a degree higher than 2. The first three parts of the codeword resemble $[u_0, \alpha u_0] + [u_1, \alpha u_1 + \alpha^2 u_0]D + [0, \alpha^2 u_1]D^2$ from the calculation $(u_0 + u_1D) \cdot G(D)$. Since $\tilde{u}_2$ is arbitrary, we have $wt_{SR}(((u_0 + u_1)D) \cdot G(D)) \leq wt_{SR}((u_0 + u_1D + \tilde{u}) \cdot G(D))$ and polynomials with a higher degree have no impact on the free sum rank distance.

It follows that $d_{SR}(\mathcal{C}) = 3$.

The truncated memory upper bound is $(n-k)(\mu+1)+1 = 3 = d^1_{SR}(\mathcal{C})$ and the bound is reached. The generalised Singleton bound is $(n-k)\left(\left\lfloor \frac{\delta}{k} \right\rfloor + 1\right) + \delta + 1 = 4$ and it is not attained.

Example 6.2.5. (Non-Systematic Encoder, Singleton Bound and L-MSR Fulfilled) Let $G(D) = [1 + \alpha + \alpha D, \alpha + \alpha^2 D]$ be the generator matrix of $\mathcal{C}$. This code has memory $\mu = 1$, $n = 2$, $k = 1$ and $\delta = 1$ when considering the full-size minors.

Let us check the distances of $\mathcal{C}$. Again, we only consider input polynomials of degree 0 and 1 by applying the same reasoning as in Example 6.2.4. Let $u_0, u_1 \in \mathbb{F}_{2^3}$ both be non-zero:

$$u_0 \cdot G(D) = [u_0 + \alpha u_0 + \alpha u_0 D, \alpha u_0 + \alpha^2 u_0 D] = [u_0 + \alpha u_0, \alpha u_0] + [\alpha u_0, \alpha^2 u_0]D$$

It results $wt_{SR}(u_0 \cdot G(D)) = 4$. This follows from the fact that $u_0 + \alpha u_0$ and αu_0 as well as αu_0 and $\alpha^2 u_0$ are linearly independent since $u_0 \neq 0$.

$$\begin{aligned}&(u_0 + u_1D) \cdot G(D) \\ &= [u_0 + \alpha u_0 + \alpha u_0 D + u_1 D + \alpha u_1 D + \alpha u_1 D^2, \alpha u_0 + \alpha u_1 D + \alpha^2 u_0 D + \alpha^2 u_1 D^2] \\ &= [u_0 + \alpha u_0, \alpha u_0] + [\alpha u_0 + u_1 + \alpha u_1, \alpha u_1 + \alpha^2 u_0]D + [\alpha u_1, \alpha^2 u_1]D^2\end{aligned}$$

We can choose u_0, u_1 such that $\alpha u_1 + \alpha^2 u_0$ can reach zero. $\alpha u_0 + u_1 + \alpha u_1$ will not be zero at the same time, which implies $wt^1_{SR}([u_0 + \alpha u_0, \alpha u_0] + [\alpha u_0 + u_1 + \alpha u_1, \alpha u_1 + \alpha^2 u_0]D) \geq 3$. Since the free distance is not truncated, we get $wt^2_{SR}((u_0 + u_1D) \cdot G(D)) = wt_{SR}((u_0 + u_1D) \cdot G(D)) \geq 5$.

The computations above imply that the 1st column distance is 3 and the free sum rank distance is 4 by taking the minimum of both calculations above.

The memory upper bound is $(n-k)(\mu+1)+1=3$ and attained. The generalised Singleton bound is $(n-k)\left(\left\lfloor\frac{\delta}{k}\right\rfloor+1\right)+\delta+1=4$ and it is reached.

We also have to consider the truncated sum rank distance bound for $j=L$ since the generator matrix is non-systematic (called L-MSR): $L=2$, $(n-k)(L+1)+1=4$.

We now calculate the 2nd truncated column sum rank distance. Additionally, the codewords generated by a 2nd degree polynomial have to be calculated. Let $u_0,u_1,u_2\in\mathbb{F}_{2^3}$ all non-zero:

$$
\begin{aligned}
&(u_0+u_1D+u_2D^2)\cdot G(D)\\
&=[u_0+\alpha u_0+\alpha u_0D+u_1D+\alpha u_1D+\alpha u_1D^2+u_2D^2+\alpha u_2D^2+\alpha u_2D^3,\dots\\
&\dots\alpha u_0+\alpha u_1D+\alpha^2u_0D+\alpha^2u_1D^2+\alpha u_2D^2+\alpha^2u_2D^3]\\
&=[u_0+\alpha u_0,\alpha u_0]+[\alpha u_0+u_1+\alpha u_1,\alpha u_1+\alpha^2u_0]D+[\alpha u_1+u_2+\alpha u_2,\alpha^2u_1+\alpha u_2]D^2+\dots\\
&\cdots+[\alpha u_2,\alpha^2u_2]D^3=c_0+c_1D+c_2D^2+c_3D^3
\end{aligned}
$$

c_0 and c_1 remain as above and reach the distance of at least 3. $[\alpha u_1+u_2+\alpha u_2,\alpha^2u_1+\alpha u_2]D^2$ cannot result in $[0,0]$. This implies that the truncated weight of length two of this codeword is at least 4. It follows that $d^2_{SR}(\mathcal{C})=4$ and as a consequence, this code is L-MSR.

Remark 6.2.6. It is not possible to find a $n=2$, $k=1$ systematic encoder which fulfils the generalised Singleton bound. We can check this by looking at the generator matrix. $G(D)$ is of the form $[1,p(D)]$, where $p(D)=p_0+p_1D+p_2D^2+\dots p_\delta D^\delta\in\mathbb{F}_{2^3}[D]$ with degree δ, which is also the degree of the code. Take $u_0\in\mathbb{F}_{2^3}$ and calculate the following codeword:

$$u_0\cdot[1,p(D)]=[u_0,p_0\cdot u_0]+[0,p_1\cdot u_0]D+\cdots+[0,p_\delta\cdot u_0]D^\delta$$

This codeword can have at most sum rank equal to $\delta+2$. This implies $d_{SR}(\mathcal{C})\leq\delta+2$ but the generalised Singleton bound is $2\delta+2$, which is strictly bigger.

Example 6.2.7. (Non-Systematic Encoder, Singleton Bound Fulfilled but not L-MSR)
Let $G(D)=[1+\alpha D,\alpha+\alpha^2D]$ be the generator matrix of the code $\mathcal{C}$. This code has memory $\mu=1$, $n=2$, $k=1$ and $\delta=1$. We also have to analyse the L bound since the generator matrix is non-systematic: $L=2$, $(n-k)(L+1)+1=4$.
Let us check the distances of $\mathcal{C}$. We consider input polynomials of degree 2 and smaller by applying the same reasoning as in Example 6.2.4. Let $u_0,u_1,u_2\in\mathbb{F}_{2^3}$ with $u_0,u_1,u_2\neq0$:

$$u_0\cdot G(D)=[u_0+\alpha u_0D,\alpha u_0+\alpha^2u_0D]=[u_0,\alpha u_0]+[\alpha u_0,\alpha^2u_0]D$$

Since $u_0\neq0$ and all parts are linearly independent, we deduce $wt_{SR}(u_0\cdot G(D))=4$.

$$
\begin{aligned}
(u_0+u_1D)\cdot G(D)&=[u_0+\alpha u_0D+u_1D+\alpha u_1D^2,\alpha u_0+\alpha u_1D+\alpha^2u_0D+\alpha^2u_1D^2]\\
&=[u_0,\alpha u_0]+[\alpha u_0+u_1,\alpha u_1+\alpha^2u_0]D+[\alpha u_1,\alpha^2u_1]D^2
\end{aligned}
$$

We can choose u_0,u_1 such that αu_0+u_1 reaches zero, i.e., $\alpha u_0=u_1$. Then, $\alpha u_1+\alpha^2u_0$ will also go to zero at the same time implying $wt^1_{SR}((u_0+u_1D)\cdot G(D))=2$ and $wt^2_{SR}((u_0+u_1D)\cdot G(D))=wt_{SR}((u_0+u_1D)\cdot G(D))=4$.

$$
\begin{aligned}
&(u_0 + u_1D + u_2D^2) \cdot G(D) \\
&= [u_0 + \alpha u_0 D + u_1 D + \alpha u_1 D^2 + u_2 D^2 + \alpha u_2 D^3, \alpha u_0 + \alpha^2 u_0 D + \alpha u_1 D + \dots \\
&\cdots + \alpha^2 u_1 D^2 + \alpha u_2 D^2 + \alpha^2 u_2 D^3] \\
&= [u_0, \alpha u_0] + [\alpha u_0 + u_1, \alpha u_1 + \alpha^2 u_0]D + [\alpha u_1 + u_2, \alpha^2 u_1 + \alpha u_2]D^2 + [\alpha u_2, \alpha^2 u_2]D^3 \\
&= c_0 + c_1 D + c_2 D^2 + c_3 D^3
\end{aligned}
$$

Again as above, choosing $\alpha u_0 = u_1$ and $u_2 = \alpha^2 u_0$, then both c_1 and c_2 go to zero. Since $[u_0, \alpha u_0]$ cannot be $[0, 0]$ as $u_0 \neq 0$, $wt^1_{SR}(c_0 + c_1 D + c_2 D^2 + c_3 D^3) \geq 2$ and $wt^2_{SR}(c_0 + c_1 D + c_2 D^2 + c_3 D^3) \geq 2$. For the free sum rank weight, we obtain $wt_{SR}(c_0 + c_1 D + c_2 D^2 + c_3 D^3) \geq 4$ since c_3 adds a rank of two.

This implies that $d^2_{SR}(\mathcal{C}) = 2$ and the code is not L-MSR since $(n-k)(L+1)+1 = 4$.

The 1st truncated distance is $d^1_{SR}(\mathcal{C}) = 2$. The memory upper bound is $(n-k)(\mu+1)+1 = 3$ and it is also not reached.

The free sum rank distance is $d_{SR}(\mathcal{C}) = 4$. The generalised Singleton bound is $(n-k)\left(\left\lfloor \frac{\delta}{k} \right\rfloor + 1\right) + \delta + 1 = 4$ and the bound is attained.

Example 6.2.8. (Non-Systematic Encoder, L-MSR Fulfilled but not the Singleton Bound)
Let $G(D) = [\alpha + \alpha D, 1 + \alpha^2 D, \alpha^2]$ be the generator matrix of the code $\mathcal{C}$. This code has memory $\mu = 1$, $n = 3$, $k = 1$ and $\delta = 1$. We also have to consider the L bound since the generator matrix is non-systematic: $L = 1$, $(n-k)(L+1)+1 = 5$. We will check that $d_{SR} = 5$ and $d^1_{SR} = 5$.
As usual, we analyse input polynomials of degree less or equal to 1 by applying the same reasoning as in Example 6.2.4. Let $u_0, u_1 \in \mathbb{F}_{2^3}$ with $u_0, u_1 \neq 0$:

$$u_0 \cdot G(D) = [\alpha u_0 + \alpha u_0 D, u_0 + \alpha^2 u_0 D, \alpha^2 u_0] = [\alpha u_0, u_0, \alpha^2 u_0] + [\alpha u_0, \alpha^2 u_0, 0]D$$

Since $u_0 \neq 0$ and $\alpha u_0, u_0, \alpha^2 u_0$ are linearly independent, we have $wt_{SR}(u_0 \cdot G(D)) = 5$.

$$
\begin{aligned}
&(u_0 + u_1 D) \cdot G(D) \\
&= [\alpha u_0 + \alpha u_0 D + \alpha u_1 D + \alpha u_1 D^2, u_0 + \alpha^2 u_0 D + u_1 D + \alpha^2 u_1 D^2, \alpha^2 u_0 + \alpha^2 u_1 D] \\
&= [\alpha u_0, u_0, \alpha^2 u_0] + [\alpha u_0 + \alpha u_1, \alpha^2 u_0 + u_1, \alpha^2 u_1]D + [\alpha u_1, \alpha^2 u_1, 0]D^2 \\
&= c_0 + c_1 D + c_2 D^2
\end{aligned}
$$

Neither can c_0 be zero nor are its components linearly dependent thus, it follows $\text{rank}(c_0) = 3$. The same holds for c_2 which implies $\text{rank}(c_2) = 2$ and hence, $wt_{SR}((u_0 + u_1 D) \cdot G(D)) \geq 5$. Consider $c_1 = [\alpha u_0 + \alpha u_1, \alpha^2 u_0 + u_1, \alpha^2 u_1]$ and we distinguish five cases:

- Choose u_1 arbitrarily and define u_0 such that $\alpha^2 u_0 + u_1 = \alpha^2 u_1$, then $u_0 = u_1 + \alpha^5 u_1 = \alpha^4 u_1$. This implies $\alpha u_0 + \alpha u_1 = \alpha^5 u_1 + \alpha u_1 = \alpha^6 u_1$. Then, $c_1 = [\alpha^6 u_1, \alpha^2 u_1, \alpha^2 u_1]$ and the $\text{rank}(c_1) = 2$.

- Choose u_1 arbitrarily and define u_0 such that $\alpha u_0 + \alpha u_1 = \alpha^2 u_1$, then $u_0 = \alpha u_1 + u_1 = \alpha^3 u_1$. This implies $\alpha^2 u_0 + u_1 = \alpha^5 u_1 + u_1 = \alpha^4 u_1$. Then, $c_1 = [\alpha^2 u_1, \alpha^4 u_1, \alpha^2 u_1]$ and the $\text{rank}(c_1) = 2$.

- Choose u_0, u_1 such that $\alpha u_0 + \alpha u_1 = \alpha^2 u_0 + u_1$, then $u_1 = \alpha u_0$. This implies $\alpha^2 u_0 + u_1 = \alpha^4 u_0$. Then, $c_1 = [\alpha^4 u_0, \alpha^4 u_0, \alpha^3 u_0]$ and the $\text{rank}(c_1) = 2$.
- Choose u_1 arbitrarily and define u_0 such that $\alpha u_0 + \alpha u_1 = 0$, then $u_0 = u_1$. This implies $\alpha^2 u_0 + u_1 = \alpha^6 u_1$. Then, $c_1 = [0, \alpha^6 u_1, \alpha^2 u_1]$ and the $\text{rank}(c_1) = 2$.
- Choose u_0 arbitrarily and define u_1 such that $\alpha^2 u_0 + u_1 = 0$, then $u_1 = \alpha^2 u_0$. This implies $\alpha u_0 + \alpha u_1 = u_0$. Then, $c_1 = [u_0, 0, \alpha^4 u_0]$ and the $\text{rank}(c_1) = 2$.

The case distinction ensures that $wt^1_{SR}(c_{[0,1]}) \geq 5$.

It follows that $d^1_{SR}(\mathcal{C}) = 5$ and $d_{SR}(\mathcal{C}) = 5$.
The generalised Singleton bound is $(n-k)\left(\left\lfloor \frac{\delta}{k} \right\rfloor + 1\right) + \delta + 1 = 6$, concluding that the Singleton bound is not reached but the L-MSR bound is.

Example 6.2.9. (Systematic Encoder, Singleton Bound Fulfilled but not m-MSR) Let

$$G(D) = \begin{bmatrix} 1 & 0 & \alpha & \alpha^2 + D \\ 0 & 1 & \alpha^3 & \alpha^{13} + \alpha^8 D \end{bmatrix}$$

be the generator matrix of $\mathcal{C}$ over the field $\mathbb{F}_{2^4}$ with α the zero of the polynomial $x^4 + x + 1$. It has the following parameters: $k = 2$, $n = 4$, $\delta = 1$ and $\mu = 1$. The Singleton bound is equal to 4 and the m-MSR bound is 5. Since the m-MSR bound is bigger than the Singleton bound, it follows that the m-MSR bound cannot be achieved. We can check that $d_{SR}(\mathcal{C}) = 4$. This shows that the code reaches the generalised Singleton bound but is not m-MSR.

Let $(p_1, p_2) \in \mathbb{F}^2_{2^4}$ with $(p_1, p_2) \neq (0, 0)$ then

$$\begin{aligned}(p_1, p_2) \cdot G(D) &= [p_1, p_2, \alpha p_1 + \alpha^3 p_2, \alpha^2 p_1 + \alpha^{13} p_2 + p_1 D + \alpha^8 p_2 D] \\ &= [p_1, p_2, \alpha p_1 + \alpha^3 p_2, \alpha^2 p_1 + \alpha^{13} p_2] + [0, 0, 0, p_1 + \alpha^8 p_2] D.\end{aligned}$$

Since $(p_1, p_2) \neq 0$, only one of p_1, p_2 can go to zero at the same time and all entries are linearly independent. Therefore, it follows that $wt_{SR}((p_1, p_2) \cdot G(D)) = 4$. To prove that, we look at the following cases:

- If $p_1 = 0$ or $p_2 = 0$ it follows directly that the rank is 4.
- Choose p_1 such that $\alpha p_1 = \alpha^3 p_2 \Rightarrow p_1 = \alpha^2 p_2$.
  We obtain $[\alpha^2 p_2, p_2, 0, \alpha^4 p_2 + \alpha^{13} p_2] + [0, 0, 0, \alpha^2 p_2 + \alpha^8 p_2] D = [\alpha^2 p_2, p_2, 0, \alpha^3 p_2 + \alpha^2 p_2 + \alpha p_2] + [0, 0, 0, \alpha^2 p_2 + \alpha^8 p_2] D$ which has rank 4.
- Choose p_1 such that $\alpha^2 p_1 = \alpha^{13} p_2 \Rightarrow p_1 = \alpha^{11} p_2$.
  We obtain $[\alpha^{11} p_2, p_2, \alpha^{12} p_2 + \alpha^3 p_2, 0] + [0, 0, 0, \alpha^{11} p_2 + \alpha^8 p_2] D = [\alpha^3 p_2 + \alpha^2 p_2 + \alpha p_2, p_2, \alpha^2 p_2 + \alpha p_2 + p_2, 0] + [0, 0, 0, \alpha^{11} p_2 + \alpha^8 p_2] D$ which has rank 4.
- Choose p_1 such that $p_1 = \alpha^8 p_2$.
  We obtain $[\alpha^8 p_2, p_2, \alpha^9 p_2 + \alpha^3 p_2, \alpha^{10} p_2 + \alpha^{13} p_2] + [0, 0, 0, 0] D = [p_2 + \alpha^2 p_2, p_2, \alpha p_2, \alpha p_2 + \alpha^3 p_2] + [0, 0, 0, 0] D$ which has rank 4.

As conjectured above, it follows that $wt_{SR}((p_1, p_2) \cdot G(D)) = 4$. We will now check this for codewords produced by a degree one polynomial. Polynomials of a higher degree can be omitted by applying the same reasoning as in Example 6.2.4.

Let $p_1, p_2, p_3, p_4 \in \mathbb{F}_{2^4}$ with $(p_1, p_2) \neq (0, 0)$ and $(p_3, p_4) \neq (0, 0)$ then

$$\begin{aligned}
&((p_1, p_2) + (p_3, p_4)D) \cdot G(D) \\
&= [p_1 + p_3 D, p_2 + p_4 D, \alpha p_1 + \alpha^3 p_2 + \alpha p_3 D + \alpha^3 p_4 D, \dots \\
&\dots \alpha^2 p_1 + \alpha^{13} p_2 + p_1 D + \alpha^8 p_2 D + \alpha^2 p_3 D + \alpha^{13} p_4 D + p_3 D^2 + \alpha^8 p_4 D^2] \\
&= [p_1, p_2, \alpha p_1 + \alpha^3 p_2, \alpha^2 p_1 + \alpha^{13} p_2] + [p_3, p_4, \alpha p_3 + \alpha^3 p_4, p_1 + \alpha^8 p_2 + \alpha^2 p_3 + \alpha^{13} p_4] D \dots \\
&+ [0, 0, 0, p_3 + \alpha^8 p_4] D^2 = c_0 + c_1 D + c_2 D^2.
\end{aligned}$$

As above, c_0 has at least rank 3 in each case. Since $(p_3, p_4) \neq (0, 0)$, the sum rank weight of c_1 is at least 1. So, $wt_{SR}(c_0 + c_1 D + c_2 D^2) \geq 4$. With the calculation above, it follows that $d_{SR}(\mathcal{C}) = 4$.

We have compared the generalised Singleton bound with the truncated distance upper bound. If the code is L-MSR, it achieves the j-th column sum rank distance for $j = L$. If $(n - k)$ divides the degree δ of the code, then this L-MSR code also achieves the generalised Singleton bound. We have also tried to find some regularities in the other direction. During this investigation, we have found different examples which disprove the existence of such a theorem. On the one hand, there exist codes achieving the generalised Singleton bound but not the m-MSR or L-MSR bound. On the other hand, there exist codes achieving both upper bounds. It also does not matter whether the encoder is systematic or non-systematic. This shows that for general parameters, the properties of being m-MSR or L-MSR and reaching the generalised Singleton bound are not related. Gluesing-Luerssen et al. [GLRS06, p. 587] discovered a similar property. Consequently, one has to examine each code separately and determine their properties.

Chapter 7

Conclusion and Further Research

In this thesis, we presented and analysed multiple code constructions which can be used with random linear network coding. First, coding theory, especially rank metric codes, were introduced. This theory constitutes the foundation for linear network coding. Afterwards, we announced the fundamental idea of network coding and its advantage over the other network transmission models.

Then, two different channel transmission network models were defined. These models define how data is sent through the network and which packets and information the receiver obtains. We distinguished between the rank deficient sliding window network model with time delay and without time delay. The first model permits combining packets from two different shots. In other words, this network model allows combining two packets from different timestamps whereas the latter forbids mixing different time intervals. We evaluated several properties depending on the network models by discussing their concrete definitions, examples and decoding capability. We proved that one single decoding measure for both models is not feasible. The truncated j-th column sum rank distance is an appropriate measure for the model without delay while the column rank distance is appropriate for the model with delay. In particular, for the rank deficient sliding window network model without time delay, there is a necessary and sufficient decoding condition.

The main part of the thesis showed four useful rank metric code constructions. We presented m-MSR, MRP, MRD and (P)UM codes. m-MSR codes maximise the j-th column sum rank distance up to the memory. Although the field size of this construction is large, this code perfectly suits the delay-free use case. The drawback of the field size can be counteracted by two new approaches: First, one can construct a code over a smaller field by random. Then, one has to check whether it fulfils the properties of an m-MSR code. With a high probability, the code satisfies these properties and we constructed an m-MSR code over a smaller field. Second, Almeida et al. [AMPN20] conjecture a construction of an m-MSR code that requires a smaller field size than the original construction.
Compared to m-MSR codes, MRP codes resemble the construction of it but reach the upper bound for the column rank distance. It satisfies the requirements for the network model with delays. However, the problem with the large field size persists.
As the third code construction, we presented the MRD code. It applies another isomorphism than the other codes to obtain the desired matrix structure. This enables the construction of different codes to the m-MSR code construction. We showed in the last chapter that the two matrix constructions are not equivalent and the MRD code construction allows building a larger range of codes. This new construction for MRD codes provides another truncated sum rank upper bound. The new bound is attained by this specific MRD code construction and is higher than the bound for m-MSR codes. That endows the MRD codes with a beneficial decoding property and makes them useful for the discussed network models.

As a final code construction, we scrutinised the (P)UM codes which are convolutional codes with memory one. Therefore, (P)UM codes are less complex than high memory codes. The introduced construction of a partial unit memory code based on Gabidulin codes attains the upper free sum rank distance. However, it does not reach the upper truncated sum rank distance, nor it is an MRP code in general. Nevertheless, (P)UM codes are attractive from a mathematical and engineering perspective since they are endowed with practical properties.

In the last section of the thesis, we produced a comparison between the related bounds of the sum rank distance. In particular, we looked at the generalised Singleton bound and the j-th column sum rank upper bound. The main result is that the two bounds do not correlate. This means we cannot conclude from one bound to the other and we have to calculate the distances separately in both cases.

During the work on this thesis, we revealed some points for further research. These are listed below:

- Constant dimension codes can be used for the application in network coding as well. This thesis concentrated on rank metric codes. One could also analyse the different constructions of constant dimension codes and compare them to rank metric codes.
- MRD codes have great potential to be used in network coding. Although, it is not yet evaluated how to integrate these codes in the network coding model with the transmission channel matrix.
- The m-MSR and MRP code constructions show a decoding strategy adapted to the network model. The constructions of (P)UM and MRD codes are also endowed with a decoding algorithm. One could now inspect the adaption of these general decoding algorithms to concrete network models.
- The presented m-MSR code construction has a large field size. We outlined methods to reduce this field size. A similar idea might be possible for the MRP code construction.

Appendix A

Overview of the Presented Code Constructions

Code Type	Fulfilled Bound	Field Size	Decoding Property	Construction
m-MSR For network models without time delay in the transmissions	$d_{SR}^{\mu}(\mathcal{C}) = (\mu+1)(n-k)+1$ Column Sum Rank Distance μ the memory of the code	$\mathbb{F}_{q^M}$ $M = q^{n(\mu+2)-1}$	$d_{SR}^{W-1}(\mathcal{C}) > nW - \sum_{t=0}^{W-1} p_t \implies s_0$ is recoverable by time $W-1$.	Theorem 5.1.10: α is a primitive and normal element.
m-MSR Conjecture For network models without time delay in the transmissions, improved field size	$d_{SR}^{\mu}(\mathcal{C}) = (\mu+1)(n-k)+1$ Column Sum Rank Distance μ the memory of the code	$\mathbb{F}_{q^M}$ $M \geq q^{R\mu+n-1}$ $R = \max\{k, n-k\}$	$d_{SR}^{W-1}(\mathcal{C}) > nW - \sum_{t=0}^{W-1} p_t \implies s_0$ is recoverable by time $W-1$.	Conjecture 5.1.15: α is a primitive element. It creates a systematic encoder of the form $G(D) = [I_k \; P(D)]$.

Code Type	Fulfilled Bound	Field Size	Decoding Property	Construction
MRP For transmissions with delay, i.e., A is an upper triangular matrix.	$d_{CR}^{L}(\mathcal{C}) = (L+1)(n-k)+1$ $L = \left\lfloor \frac{\delta}{k} \right\rfloor + \left\lfloor \frac{\delta}{n-k} \right\rfloor$ Column Rank Distance	$\mathbb{F}_{q^M}$ $M \geq q^{n(L+2)-1}$	$d_{CR}^{T}(\mathcal{C}) > n(T+1) - \text{rank}(A_{[T_0,T_f]})$ $\implies (u_{T_0}, \cdots, u_{T_f})$ can be recovered where $T = T_f - T_0$.	Theorem 5.2.4: α is a primitive and normal element.
MRD $(n \times m, k, \delta)$-rank metric convolutional code with $n \leq m$. The code is constructed over $\mathbb{F}_q^{nm}$ and the isomorphism between $\mathbb{F}_q^{nm}$ and $\mathbb{F}_q^{n \times m}$ is used.	$d_{\overline{SR}}(\mathcal{C}) =$ $n\left(\left\lfloor \frac{\delta}{k} \right\rfloor + 1\right) - \left\lceil \frac{k\left(\left\lfloor \frac{\delta}{k} \right\rfloor + 1\right) - \delta}{m} \right\rceil + 1$ Sum Rank Free Distance	$\mathbb{F}_q^{n \times m}$ $m \geq \delta + k$	It achieves the upper bound of the free sum rank distance. The Viterbi decoding algorithm is described in [NPRV18].	Theorem 5.3.12: Take a matrix A in $\mathbb{F}_q^{m \times m}$ with irreducible characteristic polynomial such that $A^i, 0 \leq i < m$, are linearly independent over $\mathbb{F}_q$. $X \in \mathbb{F}_q^{n \times m}$ full row rank. We use XA^0 to $XA^{k+\delta-1}$.
(P)UM$_G$ Convolutional code of memory one with $\text{rank}(G^{(0)}) = k$ and $\text{rank}(G^{(1)}) = k^{(1)} \leq k$. ν = sum of row degrees	$d_{SR}(\text{UM}) \leq 2n - k + 1$ $d_{SR}(\text{PUM}) \leq n - k + \nu + 1$ $\sigma \leq n - k$ (Slope) Sum Rank Distance Achieved by Construction: $d_{SR}(\text{UM}_G) \geq 2(n-k+1)$ $d_{SR}(\text{PUM}_G) = n - k + \nu + 1$ $\sigma \geq n - k - k^{(1)} + 1$ (Slope)	$\mathbb{F}_{q^m}$ $m \geq n$	PUM$_G$ achieves the maximum free sum rank distance. Further, one can use BDM decoding algorithms to decode (P)UM$_G$.	Definition 5.4.6: $k + k^{(1)} \leq n \leq m$, take $g_0, \ldots, g_{n-1} \in \mathbb{F}_{q^m}$ linearly independent over $\mathbb{F}_q$.

Bibliography

[AMPN20] Paulo Almeida, Umberto Martínez-Peñas, and Diego Napp. Systematic maximum sum rank codes. *Finite Fields and Their Applications*, 65:101677, 2020.

[AN21] Paulo Almeida and Diego Napp. A new rank metric for convolutional codes. *Designs, Codes and Cryptography*, 89:53–73, 2021.

[ANP13] Paulo Almeida, Diego Napp, and Raquel Pinto. A new class of superregular matrices and mdp convolutional codes. *Linear Algebra and its Applications*, 439(7):2145–2157, 2013.

[BG00] Albrecht Böttcher and Sergei M. Grudsky. *Toeplitz Matrices, Asymptotic Linear Algebra and Functional Analysis*. Cambridge University Press, 2000.

[Bie17] Jurgen Bierbrauer. *Introduction to Coding Theory*. Chapman & Hall/CRC, 2nd edition, 2017.

[Det94] Uwe Dettmar. *Partial Unit Memory Codes*. VDI-Verlag GmbH, Düsseldorf, Germany, 1994.

[Dho94] Ajay Dholakia. *Introduction to Convolutional Codes with Applications*. Springer US, 1994.

[FS07] Christina Fragouli and Emina Soljanin. Network coding fundamentals. *Foundations and Trends® in Networking*, 2, January 2007.

[Gab85] Èrnest Mukhamedovich Gabidulin. Theory of codes with maximum rank distance. *Problems of Information Transmission*, 21:1–12, 1985.

[GLRS06] Heide Gluesing-Luerssen, Joachim Rosenthal, and Roxana Smarandache. Strongly-mds convolutional codes. *IEEE Transactions on Information Theory*, 52(2):584–598, 2006.

[GP08] Ernst Gabidulin and Nina Pilipchuk. Error and erasure correcting algorithms for rank codes. *Des. Codes Cryptography*, 49:105–122, December 2008.

[HL08] Tracey Ho and Desmond S. Lun. *Network Coding: An Introduction*. Cambridge University Press, 2008.

[HST08] Ryan Hutchinson, Roxana Smarandache, and Jochen Trumpf. On superregular matrices and mdp convolutional codes. *Linear Algebra and its Applications*, 428(11):2585–2596, June 2008.

[JZ15] Rolf Johannesson and Kamil Sh. Zigangirov. *Fundamentals of Convolutional Coding*. Wiley-IEEE Press, 2015.

[KK08] Ralf Koetter and Frank R. Kschischang. Coding for errors and erasures in random network coding. *IEEE Transactions on Information Theory*, 54(8):3579–3591, 2008.

[Lin99] Jacobus H. van Lint. *Introduction to Coding Theory.* Springer-Verlag, Berlin, Heidelberg, 3rd edition, 1999.

[LPR21] Julia Lieb, Raquel Pinto, and Joachim Rosenthal. Convolutional codes, in Concise encyclopedia of coding theory (eds. Huffman, Cary; Kim, Jon-Lark; Solé, Patrick), 2021.

[Mah15] Rafid Mahmood. Rank metric convolutional codes with applications in network streaming. Master's thesis, Graduate Department of Electrical and Computer Engineering, University of Toronto, November 2015.

[MBK16] Rafid Mahmood, Ahmed Badr, and Ashish Khisti. Convolutional codes with maximum column sum rank for network streaming. *IEEE Transactions on Information Theory*, 62(6):3039–3052, 2016.

[Moo96] Eliakim Hastings Moore. A two-fold generalization of fermat's theorem. *Bulletin of the American Mathematical Society*, 2(7):189–199, April 1896.

[MS12] Muriel Médard and Alex Sprintson. *Network Coding Fundamentals and Applications.* Academic Press Elsevier, 2012.

[NPRV17] Diego Napp, Raquel Pinto, Joachim Rosenthal, and Paolo Vettori. MRD rank metric convolutional codes. In *2017 IEEE International Symposium on Information Theory (ISIT)*, pages 2766–2770, 2017.

[NPRV18] Diego Napp, Raquel Pinto, Joachim Rosenthal, and Paolo Vettori. Faster decoding of rank metric convolutional codes. In *23rd International Symposium on Mathematical Theory of Networks and Systems, Hong Kong*, pages 507–510, July 2018.

[Par89] Jonathan R. Partington. *An Introduction to Hankel Operators.* London Mathematical Society Student Texts. Cambridge University Press, 1989.

[PMAG06] Fabrizio Pollara, Robert J. McEliece, and Khaled Abdel-Ghaffar. Finite-state codes. *IEEE Transactions on Information Theory*, 34(5):1083–1089, September 2006.

[RB19] Alberto Ravagnani and Eimear Byrne. On the sparsity of mrd codes. In *11th International Workshop on Coding and Cryptography - WCC 2019*, 2019.

[Rot06] Ron Roth. *Introduction to Coding Theory.* Cambridge University Press, USA, 2006.

[SKK08] Danilo Silva, Frank R. Kschischang, and Ralf Koetter. A rank-metric approach to error control in random network coding. *IEEE Transactions on Information Theory*, 54(9):3951–3967, 2008.

[TJ83] Christian Thommesen and Jørn Justesen. Bounds on distances and error exponents of unit memory codes. *IEEE Transactions on Information Theory*, 29(5):637–649, 1983.

[WSBZ11] Antonia Wachter, Vladimir R. Sidorenko, Martin Bossert, and Victor V. Zyablov. On (partial) unit memory codes based on gabidulin codes. *Problems of Information Transmission*, 47(2):117–129, June 2011.

[WZSS15] Antonia Wachter-Zeh, Markus Stinner, and Vladimir Sidorenko. Convolutional codes in rank metric with application to random network coding. *IEEE Transactions on Information Theory*, 61(6):3199–3213, 2015.