AALBORG UNIVERSITY

Department Of Mathematical Sciences

Coding for the Operator Channel

MSc Thesis

Henning Thomsen

1st of June 2012
Supervisor: Olav Geil

AALBORG UNIVERSITY

Department of Mathematical Sciences I-17
Fredrik Bajers Vej 7G
9220 Aalborg Øst
http://www.math.aau.dk/

Title:
Coding for the Operator Channel

Project Period:
February 1st 2012 - June 1st, 2012

Author:

Henning Thomsen

Supervisor:
Olav Geil

Number of copies printed and pages: 4, 102

Finished the 1st of June 2012.

Synopsis:

The topic of coding over the operator channel by means of subspace codes is considered in this thesis. This channel model and codes are motivated by random linear network coding. We first study a class of codes known as Koetter-Kschischang (KK) codes, whose definition parallels the definition of Reed-Solomon codes in classical coding theory. Various properties of KK-codes are shown, including the result that they approach the Singleton bound asymptotically. Decoding algorithms for KK-codes are also investigated. Furthermore, list decoding of subspace codes is considered. The codes introduced by Mahdavifar and Vardy, called MV-codes, are presented, and their list decoding capabilities are shown, along with correctness of the decoding. Finally, we study the problem of decoding using the theory of modules from abstract algebra.

Summary

In this thesis, we consider a channel model known as the operator channel, along with coding over such a channel. The channel model is due to Koetter and Kschischang, and is motivated by random linear network coding. The input and output of this channel is a vector space over a finite field. Coding over such a channel is done by selecting a vector space corresponding to the message to be sent. The effects of errors and erasures in the transmission of a subspace is also captured by this model.

In chapter 1, we give a short introduction to network coding, including linear network coding. Random linear network coding is also considered, along with its success probability. Here a useful fact is that the input-output relationship of random linear network coding is captured by transmission and reception of subspaces.

Chapter 2 contains various results from linear and abstract algebra, such as vector spaces over finite fields, which is used in the channel model. In order to address the erasure and error correcting capabilities of subspace codes, a metric is introduced on the set of subspaces. Also various properties of a class of polynomials, known as linearized polynomials, are shown. These polynomials are used in constructing the codes. Finally, we look briefly at some concepts from the theory of modules, which can be used in the decoding procedure.

One important class of subspace codes, due to Koetter and Kschischang, and known as KK-codes, are presented in chapter 3. The channel model is defined, and a bound on the number of codewords of a subspace codes is established. A decoding algorithm, similar to the Sudan decoding algorithm of Reed-Solomon codes, is shown, and correctness of this algorithm is established. Examples of coding over the operator channel are also provided.

List decoding is a method of being able to decode when the number of errors is larger than half the minimum distance. In chapter 4, we study a class of subspace codes due to Mahdavifar and Vardy. These codes, which are called MV-codes, are an extension of KK-codes such that list-decoding is possible. This extension is similar to the Guruswami Sudan list decoding algorithm of Reed-Solomon codes. In this chapter, we also look at how a part of the decoding procedure of subspace codes can be cast in the language of modules over non-commutative rings.

Resumé

Nærværende speciale behander en kanalmodel, kalder operator kanalen, samt kodning over denne kanal. Denne model blev fremsat af Koetter og Kschischang, og er motiveret af netværkskodning. Koderne som behandles kaldes underrumskoder, og består af vektorrum over et endeligt legeme. Afsenderen vælger et vektorrum svarende til den besked som ønskes afsendt. Dette vektorrum bliver sendt over operatorkanalen. Kanalen kan forårsage fejl og sletninger, således at det modtagne vektorrum ikke er lig det afsendte, og modellen tager højde for dette.

Der lægges ud med at give en kort introduktion til netværkskodning i kapitel 1. Indenfor dette emne er specielt lineær netværkskodning vigtig, og tilfældig lineær netværkskodning behandles.

I kapitel 2 presentæres diverse resultater fra abstrakt og lineær algebra, som skal anvendes senere i specialet. Disse emner inkluderer vektorrum over endelige legemer, afstandsbestemmelse mellem underrum, og en speciel klasse af polynomier, kaldet lineariserede polynomier. Disse polynomier anvendes i konstruktionen af koderne. Nogle emner fra teorien om moduler over ringe gennemgås. Denne teori anvendes i nogle af dekodningsmetoderne.

En vigtig klasse af underrumskoder blev indført af Koetter og Kschischang i 2008. Disse koder, som kaldes KK-koder, bliver presentæret i kapitel 3. Vi ser på forskellige egenskaber for disse koder, så som minimumsafstand og fejlretningsevne. En dekodningsalgoritme lignende Sudan dekodningsalgoritmen for Reed-Solomon koder, belyses, og der ses på korrektheden af denne algoritme.

I kodningsteori spiller listedekodning en vigtig rolle. Denne dekodningsmetode gør det muligt at dekode korrekt, selv om antallet af fejl er større end halvdelen af minimumsafstanden. I kapitel 4 ser vi på en klasse af koder fremsat af Mahdavifar og Vardy. Disse koder, som kaldes MV-koder, er en modifikation af KK-koder, hvilket muliggør listedekodning. Egenskaber for disse koder vises også i dette kapitel. Ydermere ses der på en listedekodningsalgoritme til MV-koder. Denne algoritme drager paralleler til Guruswami Sudan listedekodning af Reed-Solomon koder. Desuden belyses en del af dekodningen gennem teorien om moduler over ikke-kommutative ringe.

Preface

The current thesis is written by Henning Thomsen in the spring semester of 2012. This thesis deals with various theoretical results in the branch of mathematics known as coding theory, more specifically subspace codes.

The prerequisites for reading this report is a knowledge of linear and abstract algebra, and the theory of algorithms

Citations are done in the format [Ln01], where Ln is the last name of the author or authors and the last two digits is the year of publication. Equations, figures and tables are referred to in the format $(a.b)$, where a denotes the chapter and b the order, starting with 1 in each chapter.

I would like to take the oppurtunity to thank my supervisor Olav Geil for providing feedback and answering my questions. Furthermore, thanks to Hessam Mahdavifar, Department of Electrical and Computer Engineering, University of California San Diego, for answering questions regarding the paper [MV12].

Henning Thomsen
Aalborg, June 2012

List of symbols

$G = (V, E)$	Graph with vertex set V and edge set E
$\Gamma_I(v)$, $\Gamma_O(v)$	Set of edges entering and leaving vertex v, respectively
$\mathcal{N}$	Network coding problem
$\mathbf{A}_r$	Transfer matrix for receiver r
$\mathbb{F}$	Field
$\mathbb{F}_q$	Finite field with q elements
$\mathbb{N}_0$	The set of nonnegative integers $\{0, 1, 2, 3, \ldots\}$
$\mathbb{N}$	The set of positive integers$\{1, 2, 3, \ldots\}$
$Y(e)$	Information on edge e
c_e^ℓ	Local coding vector for edge e
c_e	Global coding vector for edge e
$\mathcal{P}(W)$	Set of all subspaces of the vector space W over a finite field
$\mathcal{P}(W, \ell)$	Set of all ℓ-dimensional subspaces W
$\begin{bmatrix} N \\ \ell \end{bmatrix}_q$	Gaussian coefficient, see def. 2.7
$[i]$	Alternative notation for q^i
$\langle \alpha \rangle$	Linear span of α over finite field
$\mathscr{L}_q[X]$	Ring of linearized polynomials with coefficients in $\mathbb{F}_q$
$\mathscr{L}_q^k[X]$	Set of linearized polynomials with coefficients in $\mathbb{F}_q$, of degree less than q^k
$f^{\otimes L}(x)$	$\underbrace{f(x) \otimes \cdots \otimes f(x)}_{L \text{ times}}$
$\mathrm{lt}(f)$	Leading term of f
$\mathrm{lm}(f)$	Leading monomial of f
$\mathrm{lc}(f)$	Leading coefficient of f
$\prec$	Monomial ordering
$\deg_{1,k-1}(f)$	$(1, k-1)$-degree of f, see def. 3.25

Contents

Chapter 1

Network Coding

This chapter is a short introduction to network coding. We will start with looking briefly at networks, including some concepts from flows in networks. Then we will define network coding, which is a method where a sender transmits data to one or many receivers through a network. We will look at one branch of network coding called linear network coding, followed by random linear network coding.

1.1 Linear Network Coding

In this section, we will look at a method of communicating information through a network. This method has its origins in computer science and electrical engineering, and put briefly, is about communicating from one sender to one or many receivers. The traditional way of doing this is by using routing. However, if the network contains bottlenecks, then another method, called network coding, is superior to routing. The communcations network is modeled as a directed, acyclic graph $G = (V, E)$, where V is the set of vertices, sometimes called nodes, and E is the set of directed edges. We assume that every edge has capacity of one, that is, we can transmit one unit of information per time unit. Also, it is assumed that there is no delay in the transmission on the edges. The thing that distinguishes network coding from routing is that intermediate nodes in the network can form combinations of the incoming information, before forwarding. We will look at at particular form of network coding, where these encoding funcions are linear. The resulting method is then called linear network coding. It was shown in [ACLY00] that if intermediate nodes use coding instead of forwarding, the throughput is increased. Further, it was shown in [LYC03]

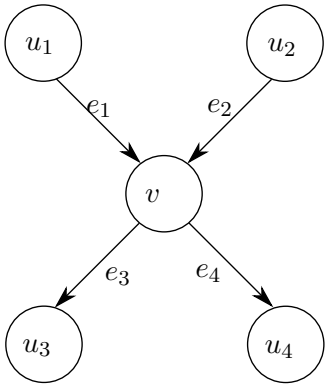

Figure 1.1: Example of edge sets of incoming and outgoing edges of a vertex.

that it suffices for the coding to be linear, that is, linear network coding is enough to get the maximum throughput. Network coding from an algebraic point of view was first devised in [KM03]. This last paper shows the existence of optimal network coding schemes for for both delay-free acyclic networks, as well as networks with delay or cycles.

Let $G = (V, E)$ be a directed, acyclic graph. Fix a vertex $s \in V$, which we call the source. Also, let $R = \{r_1, \ldots, r_k\} \subseteq V \backslash \{s\}$ be a set of receivers. A vertex which is neither source nor receiver is called an intermediate vertex. The source wants to send the symbols $X_1, \ldots, X_h \in \mathbb{F}_q$, to each of the receivers. A directed edge $e \in E$ with starting vertex u and ending vertex v is denoted (u, v). For a vertex $v \in V$, we denote the set of edges incident to v as $\Gamma_I(v)$, and the set of edges going out of v as $\Gamma_O(v)$. For example, in fig. 1.1, the edges $e_1 = (u_1, v)$ and $e_2 = (u_2, v)$ are in $\Gamma_I(v)$, while $e_3 = (v, u_3)$ and $e_4 = (v, u_4)$ are in $\Gamma_O(v)$.

Definition 1.1. Let G be a graph. A path P from s to r in G is a sequence of edges $\{(s, v_1), (v_1, v_2), \ldots, (v_{k-1}, v_k), (v_k, r)\}$, where $s, r, v_j \in V$.

We call two paths P_1 and P_2 from s to r edge-disjoint if they do not have any edge in common. From graph theory, we have the definition of a flow. A flow is an assignment of numbers to each edge, such that for each edge, the number assigned satisfies the flow conservation constraint. This means that the flow on incoming edges to a vertex v equals the flow of its outgoing edges. Also, the flow must satisfy the capacity constraint, which means that the flow on an edge e is always less than the capacity of that edge. Because we assume in this text that the capacity of each edge is one, and we work with integer flows, a flow $f(e)$ on an edge can be either zero or one. Then we can identify a flow on a graph G with another graph G_f, where the vertex sets of G_f is the same as the one in G, and there is an edge in $e_f = (u, v)$

in G_f if and only if the flow on the edge e in G is one. Then, we can define a flow as a set of h edge-disjoint paths from s to r in G, and h is called the size of this flow.

Definition 1.2. Let G be a graph, with source s and a receiver $r \in R$. A cut in G is a set of edges in E whose removal separates s from r. The value of the cut is the number of edges removed.

From graph theory, we have the maximum flow - minimum cut theorem [Wes01, th. 4.3.11]. It states that the minimum value of a cut between two vertices in a graph, where each edge has been assigned a capacity, is equal to the maximum flow between the same two vertices. In our case, if the minimum cut between s and r is h, then there is a flow of size h between s and r. Such a flow can be found by using the Ford-Fulkerson algorithm [CLRS01, sec. 26.2].

With these concepts, we now define a network coding problem.

Definition 1.3 (Multicast Network Coding Problem)**.** Let $G = (V, E)$ be a directed, acyclic graph, with one source $s \in V$ and a set of receivers $R \subseteq V \backslash \{s\}$. The source wants to send h symbols $X_1, \ldots, X_h \in \mathbb{F}_q$ each of the receivers. This situation is a called multicast network coding problem, and we denote it by $\mathcal{N}$.

In the remainder of this section, we will state and prove various results related to linear network coding, and look at an example. This part is based on [FS07]. Let $v \in V$ be a vertex other than the source. For an edge e, the information on this edge is written $Y(e)$. Because we assume that the vertices only do linear operations on the information on incoming edges, for an edge $e = (v, u) \in \Gamma_O(v)$, the information on this edge can be written in terms of the incoming edges to v as

$$Y(e) = \sum_{e_j \in \Gamma_I(v)} \alpha_j Y(e_j),$$

where $\alpha_j \in \mathbb{F}_q$. Since the source s wants to transmit h symbols $X_1, \ldots, X_h$ to each of the receivers, it needs to have at least h outgoing edges. Suppose that we have h edge-disjoint paths $P_1, \ldots, P_h$ from s to a receiver r, and let e_i be the first edge on path P_i. The source then transmits symbol X_i on outgoing edge e_i, so $Y(e_i) = X_i$.

For an edge $e = (v, u) \in E$, we define the local coding vector c_e^ℓ as the $1 \times |\Gamma_I(v)|$ vector containing the encoding coefficients of edges incident to v, so

$$c_e^\ell = \left[\alpha_1, \cdots, \alpha_{|\Gamma_I(v)|}\right].$$

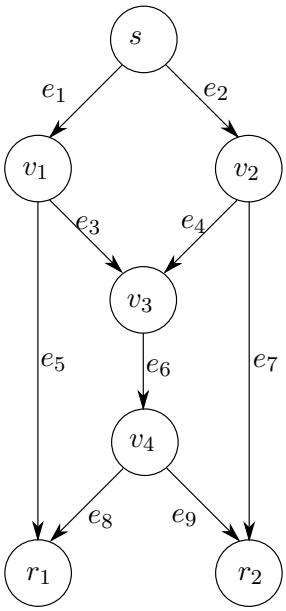

Figure 1.2: Butterfly network.

Because we assume that the graph is directed and acyclic, we can order the edges in an ancestral ordering. Using this ordering, we can start at an edge $e \in E$, and backtrack through the graph, until we arrive at the edges from the source. Then we can write the information $Y(e)$ on edge e as a linear combination of the transmitted symbols,

$$\begin{aligned} Y(e) &= c_e(1)Y(e_1) + \cdots + c_e(h)Y(e_h) \\ &= c_e(1)X_1 + \cdots + c_e(h)X_h, \end{aligned}$$

where $e_1, \ldots, e_h$ are the first edges on the h paths from s to r. For the edge e, the $1 \times h$ vector

$$c_e = [c_e(1), \cdots, c_e(h)]$$

is called the global coding vector for edge e.

Example 1.4. In this example we consider the butterfly network, depicted in fig. 1.2. The source s transmits the symbol X_1 on edge e_1, and X_2 on edge e_2. Both recievers r_1 and r_2 demand the two symbols. Using the definiton of local coding vectors, the information on edge e_5, $Y(e_5)$ can be written as

$$Y(e_5) = \alpha_{1,5}Y(e_1)$$

Similarly, the information on edge e_6 is

$$Y(e_6) = \alpha_{3,6}Y(e_3) + \alpha_{4,6}Y(e_4).$$

Here all $\alpha_{i,j}$ are in some finite field $\mathbb{F}_q$.

Now, we consider the incoming edges to receiver r_1, which are e_5 and e_8. To express $Y(e_5)$ and $Y(e_8)$ in terms of X_1 and X_2, we backtrack through the network. Then we have

$$Y(e_5) = \alpha_{1,5}Y(e_1) = \alpha_{1,5}X_1,$$

while the information on e_8 can be written as

$$\begin{aligned} Y(e_8) &= \alpha_{6,8}Y(e_6) \\ &= \alpha_{6,8}\left(\alpha_{3,6}Y(e_3) + \alpha_{4,6}Y(e_4)\right) \\ &= \alpha_{6,8}\left(\alpha_{3,6}\alpha_{1,3}Y(e_1) + \alpha_{4,6}\alpha_{2,4}Y(e_2)\right) \\ &= \alpha_{6,8}\left(\alpha_{3,6}\alpha_{1,3}X_1 + \alpha_{4,6}\alpha_{2,4}X_2\right) \\ &= \alpha_{6,8}\alpha_{3,6}\alpha_{1,3}X_1 + \alpha_{6,8}\alpha_{4,6}\alpha_{2,4}X_2 \end{aligned}$$

Similarly, for receiver r_2, the information on the incoming edges can be written as

$$\begin{aligned} Y(e_7) &= \alpha_{2,7}X_1 \\ Y(e_9) &= \alpha_{6,9}\alpha_{3,6}\alpha_{1,3}X_1 + \alpha_{6,9}\alpha_{4,6}\alpha_{2,4}X_2 \end{aligned}$$

We see that the local coding vector for edge e_8 is $[\alpha_{6,8}]$, while the global coding vector for this edge is $[\alpha_{6,8}\alpha_{3,6}\alpha_{1,3}, \alpha_{6,8}\alpha_{4,6}\alpha_{2,4}]$.

Returning to the general case, let $r \in R$ be a receiver, and consider the information on an incoming edge $e \in \Gamma_I(r)$. We write it in terms of the global coding vector

$$Y(e) = c_e(1)X_1 + \cdots + c_e(h)X_h. \tag{1.1}$$

For the h edges in a flow, incoming to r, we write the information on each of those edges as in eq. (1.1). We can then arrange the global coding vectors as the rows of a $h \times h$ matrix, and get the linear system

$$\begin{bmatrix} Y(e_1) \\ Y(e_2) \\ \vdots \\ Y(e_h) \end{bmatrix} = \begin{bmatrix} c_{e_1}(1) & c_{e_1}(2) & \cdots & c_{e_1}(h) \\ c_{e_2}(1) & c_{e_2}(2) & \cdots & c_{e_2}(h) \\ \vdots & \vdots & \vdots & \vdots \\ c_{e_h}(1) & c_{e_h}(2) & \cdots & c_{e_h}(h) \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_h \end{bmatrix},$$

which can be written more compactly as $\mathbf{y} = \mathbf{A}_r\mathbf{x}$. We see that for this receiver to decode, we require that the matrix $\mathbf{A}_r$ is invertible. We call the matrix $\mathbf{A}_r$ the transfer matrix for reciever r. But this is the same as asking wheter the determinant of the transfer matrix $\mathbf{A}_r$ is non-zero. Also, if all

receivers are to be able to decode the information transmitted, then we must have that

$$\prod_{r \in R} \det(\mathbf{A}_r)$$

is nonzero. With this, we can provide the following definition.

Definition 1.5. A network coding problem $\mathcal{N}$ is called solvable if there exists an assignment of coefficients α_i such that the transfer matrix $\mathbf{A}_r$ is invertible for each receiver $r \in R$.

If we regard the coefficients α_i as variables, then $\det(\mathbf{A}_r)$ is a multivariate polynomial in the α_i. Suppose there are k receivers. Then $\prod_{r \in R} \det(\mathbf{A}_r)$ is a multivariate polynomial in the α_i, and we denote it by $f(\alpha_1, \ldots, \alpha_\eta)$. Because the degree of any variable in $\det(\mathbf{A}_r)$ is at most 1, the degree of any variable in $f(\alpha_1, \ldots, \alpha_\eta)$ is at most k. In order to show when a network coding problem is solvable, we need the following lemma.

Lemma 1.6. *Let $f(x_1, \ldots, x_n) \in \mathbb{F}_q[x_1, \ldots, x_n]$, and assume the degree of any variable x_i is at most d. Then for any field of size $q > d$, where $f(x_1, \ldots, x_n)$ is not identically zero, there exists $p_1, \ldots, p_n \in \mathbb{F}_q$ such that*

$$f(p_1, \ldots, p_n) \neq 0.$$

Proof. We prove this lemma on induction on n, the number of variables. For the base case, $n = 1$, and therefore $f(x_1)$ is a univariate polynomial of degree d. Because f is assumed to be a non-zero polynomial, it can have at most d roots, so if $d > q$, then there is an element $p \in \mathbb{F}_q$, such that $f(p) \neq 0$.

For the induction step, suppose the claim holds for polynomials in $n - 1$ and fewer variables. Let $f(x_1, \ldots, x_n)$ be a non-zero polynomial, and write this polynomial as

$$\sum_{i=0}^{d} f_i(x_1, \ldots, x_{n-1}) x_n^i.$$

Using the induction hypothesis, for at least one $f_i(x_1, \ldots, x_{n-1})$, there are elements $p_1, \ldots, p_{n-1} \in \mathbb{F}_q$, such that $f_i(p_1, \ldots, p_{n-1}) \neq 0$. Then

$$f(p_1, p_2, \ldots, p_{n-1}, x_n)$$

is a polynomial in one variable x_n, where the degree of x_n is at most d. Using the base case, there is an element $p_n \in \mathbb{F}_q$ such that $f(p_1, \ldots, p_n) \neq 0$. By induction, the statement is true for all n. □

The following theorem characterizes solvability in terms of network flow.

Theorem 1.7. *Let $\mathcal{N}$ be a multicast network coding problem, and suppose that there are h edge disjoint paths from the source to each of the receivers $r \in R$. Then the problem $\mathcal{N}$ is solvable.*

Proof. Consider a receiver $r \in R$, with transfer matrix $\mathbf{A}_r$. Because there are h edge-disjoint paths from the source s to this receiver, the determinant $\det(\mathbf{A}_r)$, which is a multivariate polynomial, is not identically zero. Since this holds for all receivers, the product $\prod_{r\in R} \det(\mathbf{A}_r) =: f(x_1, \ldots, x_n)$ is a multivariate polynomial which is not identically zero. Then, using lemma 1.6, there exists a large enough finite field $\mathbb{F}_q$, such that $f(p_1, \ldots, p_n) \neq 0$, for $p_j \in \mathbb{F}_q$. This shows that the product $\prod_{r\in R} \det(\mathbf{A}_r)$ is non-zero, so all transfer matrices are invertible, implying that $\mathcal{N}$ is solvable. □

We now show that if the field size q is sufficiently large, then there is a solution to a given network coding problem $\mathcal{N}$.

Corollary 1.8. *Let $\mathcal{N}$ be a multicast network coding problem, and consider the product*

$$\prod_{r\in R} \det(\mathbf{A}_r), \tag{1.2}$$

Suppose there are h edge disjoint paths from the source s to each receiver. Then if $q > |R|$, there exists an assignment of coefficients $\alpha_{i,j} \in \mathbb{F}_q$, such that $\mathcal{N}$ is solvable.

Proof. Suppose $q > |R|$. Because $\prod_{r\in R} \det(\mathbf{A}_r)$ is a multivariate polynomial, where the degree of any variable is at most $|R|$, we can use lemma 1.6 to find an assignment of coefficients such that $\prod_{r\in R} \det(\mathbf{A}_r)$ evaluates to a non-zero value. Then all matrices $\mathbf{A}_r, r \in R$ are invertible, and so the network coding problem is solvable. □

Example 1.9. This is a continuation of ex. 1.4. Here we find the transfer matrices for the two receivers. Consider first receiver r_1. We can arrange the information on incoming edges to r_1 in a vector, and get

$$\begin{bmatrix} Y(e_5) \\ Y(e_8) \end{bmatrix} = \begin{bmatrix} \alpha_{1,5} & 0 \\ \alpha_{6,8}\alpha_{3,6}\alpha_{1,3} & \alpha_{6,8}\alpha_{4,6}\alpha_{2,4} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}, \tag{1.3}$$

and for receiver r_2, we have

$$\begin{bmatrix} Y(e_7) \\ Y(e_9) \end{bmatrix} = \begin{bmatrix} 0 & \alpha_{2,7} \\ \alpha_{6,9}\alpha_{3,6}\alpha_{1,3} & \alpha_{6,9}\alpha_{4,6}\alpha_{2,4} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}, \tag{1.4}$$

We write the linear systems in eqs. (1.3) and (1.4) as $\mathbf{y}_1 = \mathbf{A}_1\mathbf{x}$ and $\mathbf{y}_2 = \mathbf{A}_2\mathbf{x}$ respectively. For both receivers to be able to decode, the matrices $\mathbf{A}_1$ and $\mathbf{A}_2$ must be invertible. Their determinants are

$$\det(\mathbf{A}_1) = \alpha_{1,5}\alpha_{6,8}\alpha_{4,6}\alpha_{2,4},$$
$$\det(\mathbf{A}_2) = -\alpha_{2,7}\alpha_{6,8}\alpha_{3,6}\alpha_{1,3}.$$

The above determinants are polynomials in the $\alpha_{i,j}$. By setting all $\alpha_{i,j}$ equal to 1, we get that the matrices $\mathbf{A}_1$ and $\mathbf{A}_2$ are invertible. Therefore, a solution to the network coding problem is to set all encoding coefficients $\alpha_{i,j}$ equal to 1.

1.2 Success probability in Random Network coding

From the results in the previous section, we saw that if there were enough edge disjoint paths from the source to each reciever, the was a solution to the network coding problem. Further, if we considered some finite field with more elements than there were receivers, then a solution could be found in that field. Because these results are non-constructive, one might ask how to find a solution. One method is to select the coefficients $\alpha_{i,j}$ at random. The resulting method is then called random linear network coding. In this section, we prove a lower bound on the probability of random linear network coding being successful. This bound was first proven in [HMK+06]. We follow the presentation from [FS07].

Theorem 1.10. *Let $\mathcal{N}$ be a network coding problem, and η be the number of encoding coefficients chosen uniformly at random from $\mathbb{F}_q$. Let $f(x_1, \ldots, x_\eta)$ be a non-zero multivariate polynomial over $\mathbb{F}_q$, where the maximum degree of each variable x_i is upper bounded by N. If $q > N$, then the probability of selecting encoding coefficients $\alpha_1, \ldots, \alpha_\eta$ such that $f(\alpha_1, \ldots, \alpha_\eta)$ evaluates to zero is upper bounded by*

$$1 - \left(1 - \frac{N}{q}\right)^\eta.$$

Proof. We prove this theorem by induction on η. For the base case $\eta = 1$, $f(x_1)$ is a univariate polynomial. The degree of $f(x_1)$ is upper bounded by N by assumption. Because $f(x_1)$ has at most N roots, the probability of chooosing a root from $\mathbb{F}_q$ is upper bounded by

$$\frac{N}{q} = 1 - \left(1 - \frac{N}{q}\right).$$

To prove the induction step, suppose $\eta > 1$ and that the statement is true for polynomials in less than η variables. We write the polynomial $f(x_1, \ldots, x_\eta)$

as sum of two polynomials f_1 and f_2, where f_1 is a polynomial in $\eta - 1$ variables and f_2 is a polynomial in η variables where x_η is raised to at most the $(d-1)$'th power, with $d \leq N$:

$$f(x_1, \ldots, x_\eta) = x_\eta^d f_1(x_1, \ldots, x_{\eta-1}) + f_2(x_1, \ldots, x_\eta).$$

We choose elements $\alpha_1, \ldots, \alpha_\eta$ uniformly at random from $\mathbb{F}_q$, $q > N$. The probability that we choose a root of f is written as $P(f = 0)$, and can be written as

$$\begin{aligned} P(f=0) &= P(f=0|f_1=0)P(f_1=0) + P(f=0|f_1\neq 0)P(f_1\neq 0) \\ &= P(f=0|f_1=0)P(f_1=0) + P(f=0|f_1\neq 0)\left(1 - P(f_1=0)\right). \end{aligned}$$

We can bound the terms as follows. We have $P(f_1 = 0) \leq 1 - \left(1 - \frac{N}{q}\right)^{\eta-1}$ using the induction hypothesis, $P(f = 0|f_1 = 0) \leq 1$, and $P(f = 0|f_1 \neq 0) \leq \frac{N}{q}$ using the base case. Therefore,

$$\begin{aligned} P(f=0) &\leq P(f_1=0) + \frac{N}{q}\left(1 - P(f_1=0)\right) \\ &= P(f_1=0)\left(1 - \frac{N}{q}\right) + \frac{N}{q} \\ &\leq \left(1 - \left(1 - \frac{N}{q}\right)^{\eta-1}\right)\left(1 - \frac{N}{q}\right) + \frac{N}{q} \\ &= 1 - \frac{N}{q} - \left(1 - \frac{N}{q}\right)^{\eta} + \frac{N}{q} = 1 - \left(1 - \frac{N}{q}\right)^{\eta}. \end{aligned}$$

By induction, the statement is true for all $\eta \geq 1$. □

From this, we see that the probability of a random selection of coefficients being successful is lower bounded by $\left(1 - \frac{N}{q}\right)^{\eta}$. By choosing the finite field to be very large, this probability can be made to be very close to unity. The bound in thm. 1.10 was first shown by Ho et. al in [HMK+06], and it is a rather weak bound. There are various improvements on this bound on success probability. In [GT11], the authors use algebraic geometry and Groebner basis theory to identify a flow in a graph with a monomial. Then a flow system, which is defined to be a set of flows, one for each receiver, is identified with a multivariate polynomial, with coefficients in some finite field. The roots of this polynomial constitute an algebraic variety, and using the footprint bound ([CLO07, prop.8, sec. 5.3]), the authors get an upper bound on the number of roots, which translates into a lower bound on the success probability. This bound is shown to be an improvement over the bound of Ho et. al. in [HMK+06].

1.3 Subspace Approach to Network Coding

In the previous sections, we considered some aspects of random network coding, where the network topology was known. But there is another way of looking at the problem of network coding, which relies heavily on linear algebra over finite fields. This approach, which was devised by Koetter and Kschischang in [KK08], is non-coherent, that is, it does not require knowledge of the network topology. Instead, one considers the network as a "black box", and the information to be sent is a vector *space* over some finite field. The relationship between the input and the output of such a network is captured in a channel model called an operator channel, and a collection of such vector spaces used for coding is called a subspace code. In the next chapter, we will look at various results from linear algebra, including how to measure distance between such spaces, and counting how many spaces there are of a given dimension. That chapter is followed by a description of the operator channel model, and issues related to coding over such a channel are addressed. In this section, we will look at the random linear network coding in the light of vector spaces and subspaces thereof.

1.3.1 The Model

Suppose we have a network where there is one sender and one reciever. The sender wants to send a number of packets to the receiver. By a packet, we mean a vector of length N over $\mathbb{F}_q$. For example, suppose we use the field $\mathbb{F}_2$, so that the symbols are binary. The field $\mathbb{F}_{256}$ can be regarded as an 8-dimensional vector space over $\mathbb{F}_2$, and elements of $\mathbb{F}_{256}$ are then binary vectors of length 8, i.e. bytes.

The packets propagate through the network, passing through intermediate nodes. Since we consider random linear network coding, each intermediate node forms a random $\mathbb{F}_q$-linear combination of packets on its incoming edges. Suppose the sender transmits M packets $\mathbf{x}_1, \ldots, \mathbf{x}_M$, where $\mathbf{x}_j \in \mathbb{F}_q^N$. At the reciever, the j'th packet can be written as

$$\mathbf{y}_j = \sum_{i=1}^{M} \alpha_{j,i}\mathbf{x}_i,$$

where $\alpha_{j,i} \in \mathbb{F}_q$. Suppose the receiver gets L packets $\mathbf{y}_1, \ldots, \mathbf{y}_L$. Then we can arrange the recieved packets in a $L \times N$ matrix $\mathbf{Y}$, and the sent packets in a $M \times N$ matrix $\mathbf{X}$. We then have

$$\mathbf{Y} = \mathbf{A}\mathbf{X},$$

where $\mathbf{A}$ is an $L \times M$ matrix with entries in $\mathbb{F}_q$. The entries are random, since we do random linear network coding. We can also add the possibility of erroneous packets injected into the network. Suppose T such packets $\mathbf{z}_1, \ldots, \mathbf{z}_T$ are injected. Then we write the relationship between the input and output as

$$\mathbf{y}_j = \sum_{i=1}^{M} \alpha_{j,i} \mathbf{x}_i + \sum_{t=1}^{T} \beta_{j,t} \mathbf{z}_t,$$

or more compactly as $\mathbf{Y} = \mathbf{AX} + \mathbf{BZ}$, where $\mathbf{B}$ is a $L \times T$ matrix and $\mathbf{Z}$ is a $T \times N$ matrix. The entries of $\mathbf{B}$ are chosen randomly from $\mathbb{F}_q$.

Because the matrix $\mathbf{A}$ is a random matrix, the only property left fixed in the product $\mathbf{AX}$ is the row space of $\mathbf{X}$, so we have that the row space of $\mathbf{AX}$ is a subspace of the row space of $\mathbf{X}$. At the recieving end, all such subspaces can be considered equivalent, so the reciever just wants some generating set for the row space of $\mathbf{AX}$. It will then try to decode to $\mathbf{X}$.

Chapter 2

Algebraic Preliminaries

In this chapter, we introduce the algebraic toos needed to deal with subspace codes and communication over the operator channel. We will start by looking at quotient spaces over finite fields, which relates to the operator channel model. Then we look at a way to measure distance between subspaces, followed by counting how many subspaces are of a given dimension. A special kind of polynomials, called linearized polynomials, play an important role in the code constructions to be considered. These polynomials are introduced, and various properties of such polynomials are shown. Finally, we look at an algebraic structure called a module, which can be used in the decoding procedure of subspace codes. This material comes from [KK08], [Cam00], [HK71], [DF04] and [MS77].

2.1 Vector Spaces over Finite Fields

Let $\mathbb{F}_q$ be the finite field with $q = p^m$ elements, where p is a prime number, and let W be a vector space over $\mathbb{F}_q$. The codes to be considered are subsets of W. So a code consists of vector spaces, and we start by looking at this topic. Let U and V be subspaces of W. Then

$$\begin{aligned} U \cap V =& \{w \in W \mid w \in U \text{ and } w \in V\}, \\ U + V =& \{w \in W \mid w = u + v \text{ for some } u \in U \text{ and } v \in V\}, \text{ and} \\ U \oplus V =& \{w \in W \mid w = u + v \\ & \qquad \text{for unique } u \in U \text{ and } v \in V \text{ where } U \cap V = \{0\}\}. \end{aligned}$$

are also subspaces of W. The set of all subspaces of W is denoted $\mathcal{P}(W)$, and we denote the set of subspaces of W of dimension ℓ by $\mathcal{P}(W, \ell)$.

Let U be a subspace of W. Define a relation $\sim$ on W as follows. For $v, w \in W$, let

$$v \sim w \Leftrightarrow v - w \in U.$$

It can be shown that this relation is an equivalence relation on W, and the classes are cosets of the form $[w] = w + U$. We can then define addition of two classes $[v]$ and $[w]$ as $[v] + [w] = [v + w]$. This definition is independent of choice of class-representative, because if v', w' are two different representatives of $[v]$ and $[w]$, then

$$\begin{aligned}[v + w] - [v' + w'] &= v + w + U - v' - w' + U \\ &= v - v' + w - w' + U \\ &= 0 + 0 + U = [0].\end{aligned}$$

Hence $[v] + [w] = [v'] + [w']$. Also, we can define scalar-multiplication as follows. Let $\alpha \in \mathbb{F}_q$ and $[v] \in W/U$. Then

$$\alpha[v] = \alpha(v + U) = \alpha v + \alpha U = \alpha v + U = [\alpha v].$$

It can be shown that with these two operations, the set W/U is a vector space over $\mathbb{F}_q$. This space is called the quotient space of W modulo U. We define the mapping ϕ as

$$\begin{aligned}\phi : W &\to W/U, \\ w &\mapsto [w] = w + U.\end{aligned}$$

The mapping ϕ sending a vector w to its equivalence class $[w]$ under $\sim$ is called the quotient mapping. From the above discussion, ϕ is well-defined. Also, it is linear, because

$$\begin{aligned}\phi(w + w') = [w + w'] = [w] + [w'] = \phi(w) + \phi(w'), \text{ and} \\ \phi(\alpha w) = [\alpha w] = \alpha[w] = \alpha\phi(w),\end{aligned}$$

for $w, w' \in W$. The following theorem, based on [HK71, theorem, p.396], shows the relationship between the direct sum and the quotient space construction.

Theorem 2.1. *Let V and V' be subspaces of W. Then we can write W as*

$$V \oplus V',$$

if and only if the quotient map $\phi : W \to W/V$ restricted to V' is a vector space isomorphism of V' with W/V.

Proof. Suppose $W = V \oplus V'$. Then, for any $w \in W$, we can uniquely write $w = v + v'$, where $v \in V$ and $v' \in V'$. From this, we want to define

a vector space isomorphism from V' to W/V. Consider the quotient map $\phi : W \to W/V$ restricted to V', i.e. $\phi|_{V'}$. We show that this mapping is surjective. Let $[w] \in W/V$. From hypothesis, $w = v + v'$, so $w - v = v'$. Hence

$$\phi|_{V'}(v') = \phi|_{V'}(w - v) = (w - v) + V = [w],$$

because $v \in V$. We now show that $\phi|_{V'}$ is injective. Suppose $\phi|_{V'}(v^*) = \phi|_{V'}(v')$. Then $[v^*] = [v']$, which means that $v^* - v' \in V$. But $v^* - v' \in V'$, and $V \cap V' = \{0\}$, since $W = V \oplus V'$. Therefore, $v^* - v' = 0$, so $v^* = v'$. Because $\phi|_{V'}$ is linear, and we have shown that it is bijective, it is a vector space isomorphism. Therefore, V' is isomorphic to W/V.

For the converse, we assume that V' and W/V are isomorphic under $\phi|_{V'}$. We show that $W = V \oplus V'$, by first showing that $W = V + V'$. Let $w \in W$. Then $\phi(w) \in W/V$. Because W/V is isomorphic to V', the vector $\phi(w)$ is uniquely identifiable with a vector in V' under the isomorphism $\phi|_{V'}$. In other words, we have $\phi^{-1}|_{V'}(\phi(w)) = v'$, so $\phi(w) = \phi|_{V'}(v')$, where $v' \in V'$. This shows that $[w] = [v']$, so $w + V = v' + V$. Therefore, we can write $w + v = v' + v^*$, where $v, v^* \in V$. Rearranging, we get $w = v' + (v^* - v)$. This shows that $W = V + V'$. To show that $V \cap V' = \{0\}$, let $w \in V \cap V'$. Then $\phi|_{V'}(w) = [w] \in W/V$. But $w \in V \cap V'$, so $[w] = [0]$. This implies that $w \in \ker(\phi|_{V'})$, and because $\phi|_{V'}$ is injective, $\ker(\phi|_{V'}) = \{0\}$. This shows that $w = 0$. We conclude that $W = V \oplus V'$. □

We note that in the preceding theorem and its proof, there is no mention of a basis for any space. Indeed, the construction of the quotient space W/V only depends on the spaces W and V. This fact makes it useful in the channel model to be considered in the next chapter.

Example 2.2. Consider the vector space $\mathbb{R}^2$, and a subspace U of $\mathbb{R}^2$, where

$$U = \{(x, y) \mid x = y\}.$$

The space U is the line through the origin with slope 1. Defining the relation $\sim$ on $\mathbb{R}^2$ as

$$(x, y) \sim (x', y') \Leftrightarrow (x - x', y - y') \in U,$$

we see that the cosets $(x, y) + U$ are lines parallel to U. We can then draw another line V through the origin, which is not parallel to U. This line intersects each of the cosets exactly once. This is shown in fig. 2.1. Therefore, we can identify V with $\mathbb{R}^2/U$, so $\mathbb{R}^2 = U \oplus V$.

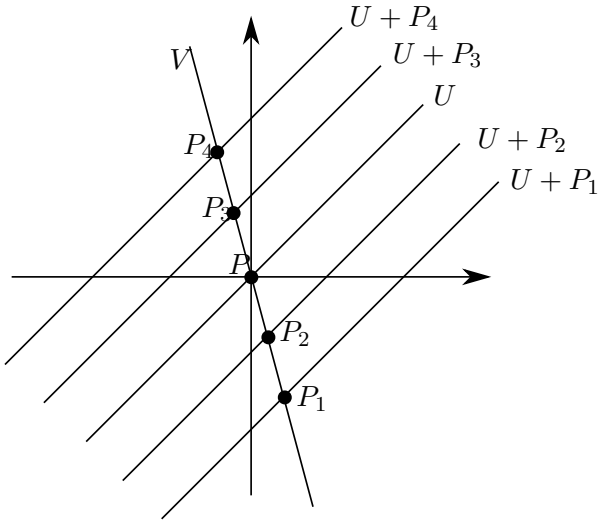

Figure 2.1: Quotient space example.

2.2 Distance Between Subspaces

In order to define the minimum distance of subspace codes, and to investigate their erasure and error correcting capabilities, we want a way to measure distance between subspaces. Define the function $d : \mathcal{P}(W) \times \mathcal{P}(W) \to \mathbb{Z}_+$ as

$$d(U, V) = \dim(U + V) - \dim(U \cap V).$$

Remark 2.3. Note that because

$$\dim(U + V) = \dim(U) + \dim(V) - \dim(U \cap V),$$

we can write $d(U, V)$ as

$$d(U, V) = \dim(U) + \dim(V) - 2\dim(U \cap V).$$

We have the following result, which was first shown in [KK08].

Proposition 2.4. *The function d is a metric on $\mathcal{P}(W)$.*

Proof. We verify the three requirements for being a metric. Let $U, V \in \mathcal{P}(W)$. Because $U \cap V$ is a subspace of $U + V$, we have $\dim(U + V) \geq \dim(U \cap V)$, and so $d(U, V) \geq 0$. Also, $d(U, V) = d(V, U)$, since $U + V = V + U$ and $U \cap V = V \cap U$. We now verify the triangle inequality. Let $T \in \mathcal{P}(W)$. We want to show that

$$d(U, V) \leq d(U, T) + d(T, V).$$

From the definition of d, this is equivalent to showing that

$$\begin{aligned}\dim(U) + \dim(V) - 2\dim(U \cap V) \leq \dim(U) + \dim(T) - 2\dim(U \cap T) \\ + \dim(T) + \dim(V) - 2\dim(T \cap V).\end{aligned}$$

We can rearrange this inequality to

$$\begin{aligned}0 &\leq 2\dim(T) + 2\dim(U \cap V) - 2\dim(U \cap T) - 2\dim(T \cap V) \\ &= 2\left(\dim(T) + \dim(U \cap V) - \dim(U \cap T) - \dim(T \cap V)\right).\end{aligned}$$

We want to show that the quantity inside the parenthesis is positive. Note that

$$\dim((U \cap T) + (T \cap V)) = \dim(U \cap T) + \dim(T \cap V) - \dim(U \cap T \cap V)$$

which can be rearranged to

$$\dim((U \cap T) + (T \cap V)) + \dim(U \cap T \cap V) = \dim(U \cap T) + \dim(T \cap V).$$

Because $(U \cap T) + (T \cap V)$ is a subspace of T, and $U \cap T \cap V$ is a subspace of $U \cap V$,

$$\begin{aligned}\dim((U \cap T) + (T \cap V)) \leq \dim(T), \\ \dim(U \cap T \cap V) \leq \dim(U \cap V),\end{aligned}$$

and we therefore have the inequality

$$\dim((U \cap T) + (T \cap V)) + \dim(U \cap T \cap V) \leq \dim(T) + \dim(U \cap V).$$

From this, we see that the triangle inequality holds, and therefore, d is a metric on $\mathcal{P}(W)$. □

2.3 Gaussian Coefficient

In this section, we will count the number of ℓ-dimensional subspaces of an N-dimensional vector space over $\mathbb{F}_q$. This result is used later, when we consider bounds on codes. The result is given in the following proposition. We first prove a lemma. The results in this section are adapted from [Cam00, ch. 1].

Lemma 2.5. *Let V be a ℓ-dimensional vector space over $\mathbb{F}_q$. The number of distinct bases $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_\ell\}$ of V equals*

$$(q^\ell - 1)(q^\ell - q)(q^\ell - q^2)\cdots(q^\ell - q^{\ell-1}), \tag{2.1}$$

where distinct refers to distinct up to permutation of basis vectors.

Proof. An ℓ-dimensional vector space over $\mathbb{F}_q$ contains q^ℓ vectors. For choosing the first basis vector $\mathbf{b}_1$, there are $q^\ell - 1$ choices, since the zero vector cannot be chosen. For the second vector $\mathbf{b}_2$, there are $q^\ell - q$ choices, since there are q vectors in the span of $\mathbf{b}_1$, and those vectors cannot be chosen. Continuing in this way, we see that the number of different choices is given in eq. (2.1). □

We now give a proof of the number of subspaces of a given vector space.

Proposition 2.6. *Let W be a N-dimensional vector space over $\mathbb{F}_q$. The number of distinct ℓ-dimensional subspaces of W is*

$$\frac{(q^N-1)(q^{N-1}-1)\cdots(q^{N-\ell+1}-1)}{(q^\ell-1)(q^{\ell-1}-1)\cdots(q-1)}. \tag{2.2}$$

Proof. We will count the number of subspaces of W of dimension ℓ. This means that we shall choose ℓ linearly independent vectors in W. The first vector can be chosen in $q^N - 1$ ways, since we cannot choose the zero vector. The next basis vector can be chosen in $q^N - q$ ways since we must not choose any vector in the first subspace, which contains q vectors. For the third vector, there are $q^N - q^2$, since there are q^N vectors in W, and q^2 of those are in the span of the previous two vectors. Continuing in this manner, we get that ℓ linearly independent vectors in W can be chosen in $(q^N-1)(q^N-q)(q^N-q^2)\cdots(q^N-q^{\ell-1})$ ways. To get the number of distince ℓ-dimensional subspaces of W, we must divide by the number of distinct bases for an ℓ-dimensional vector space over $\mathbb{F}_q$. From lemma 2.5, this number equals $(q^\ell-1)(q^\ell-q)(q^\ell-q^2)\cdots(q^\ell-q^{\ell-1})$. Therefore, the number of distinct ℓ-dimensional subspaces of W equals

$$\begin{aligned}
&\frac{(q^N-1)(q^N-q)(q^N-q^2)\cdots(q^N-q^{\ell-1})}{(q^\ell-1)(q^\ell-q)(q^\ell-q^2)\cdots(q^\ell-q^{\ell-1})} \\
&= \frac{(q^N-1)q(q^{N-1}-1)q^2(q^{N-2}-1)\cdots q^{\ell-1}(q^{N-\ell+1}-1)}{(q^\ell-1)q(q^{\ell-1}-1)q^2(q^{\ell-2}-1)\cdots q^{\ell-1}(q-1)} \\
&= \frac{(q^N-1)(q^{N-1}-1)\cdots(q^{N-\ell+1}-1)}{(q^\ell-1)(q^{\ell-1}-1)\cdots(q-1)}.
\end{aligned}$$

□

The number of ℓ-dimensional subspaces of a vector space W of dimension N over $\mathbb{F}_q$ is called the Gaussian coefficient of N over ℓ [KK08], [LW92, ch. 24].

Definition 2.7 (Gaussian Coefficient)**.** For $\ell \leq N$, we define the q-ary Gaussian coefficient $\begin{bmatrix} N \\ \ell \end{bmatrix}_q$ as

$$\begin{bmatrix} N \\ \ell \end{bmatrix}_q = \frac{(q^N - 1)(q^{N-1} - 1) \cdots (q^{N-\ell+1} - 1)}{(q^\ell - 1)(q^{\ell-1} - 1) \cdots (q - 1)}.$$

Later, we will need to upper bound the number of codewords in a subspace code. In that regard, the following proposition is used.

Proposition 2.8. *Let $q > 1$, and let $0 < \ell < n$, where ℓ and n are integers. The Gaussian coefficient satisfies the inequality*

$$\begin{bmatrix} N \\ \ell \end{bmatrix}_q < 4q^{\ell(N-\ell)}.$$

Proof. We can write the Gaussian coefficient as

$$\begin{aligned} \begin{bmatrix} N \\ \ell \end{bmatrix}_q &= q^{\ell(N-\ell)} \frac{(1 - q^{-N})(1 - q^{-N+1}) \cdots (1 - q^{-N+\ell-1})}{(1 - q^{-\ell})(1 - q^{-\ell+1}) \cdots (1 - q^{-1})} \\ &< q^{\ell(N-\ell)} \frac{1}{(1 - q^{-\ell})(1 - q^{-\ell+1}) \cdots (1 - q^{-1})} \\ &< q^{\ell(N-\ell)} \prod_{j=1}^{\infty} \frac{1}{(1 - q^{-j})}. \end{aligned}$$

Let $f(x) = \prod_{j=1}^{\infty} \frac{1}{(1-q^{-j})}$. From [LW92, ch. 15], the function $f(x)$ is increasing in $x \in \mathbb{R}$. Therefore,

$$\prod_{j=1}^{\infty} \frac{1}{(1 - q^{-j})} \leq \prod_{j=1}^{\infty} \frac{1}{(1 - 2^{-j})} < 4,$$

because increasing q means decreasing $\frac{1}{q}$. The rightmost inequality follows from the identity $\prod_{j=1}^{\infty} \frac{1}{(1-2^{-j})} = \frac{1}{Q_0}$, where $Q_0 \approx 0.288788095$ [KK08], [Ber80]. □

2.4 Linearized Polynomials

We will now look at a certain class of polynomials, called linearized polynomials, and some of their algebraic properties. These polynomials are used when defining certain subspace codes. We start by defining linearized polynomials.

Definition 2.9 (Linearized Polynomial)**.** Let $\mathbb{F}_q$ be a finite field, and let $\mathbb{F} = \mathbb{F}_{q^m}$ be an extension field of $\mathbb{F}_q$. A polynomial $L(x)$ over $\mathbb{F}$ is called a linearized polynomial if

$$L(x) = \sum_{i=0}^{d} a_i x^{q^i}, \tag{2.3}$$

where the coefficients a_i are in $\mathbb{F}$.

In the case when q is fixed, for a linearized polynomial $L(x)$ we will sometimes write $[i]$ instead of q^i, and eq. (2.3) then becomes $L(x) = \sum_{i=0}^{d} a_i x^{[i]}$.

Let $\mathscr{L}_{q^m}[X]$ be the set of linearized polynomials with coefficients in $\mathbb{F}_{q^m}$. We define two binary operations on $\mathscr{L}_{q^m}[X]$ and show that the resulting algebraic structure is a non-commutative ring with unity.

Proposition 2.10. *Let $\mathbb{F}_q$ be a finite field, and let $\mathbb{F} = \mathbb{F}_{q^m}$ be an extension field of $\mathbb{F}_q$. Let $L_1(x), L_2(x) \in \mathscr{L}_{q^m}[X]$, where*

$$L_1(x) = \sum_{i=0}^{d_1} a_i x^{[i]}, \quad L_2(x) = \sum_{j=0}^{d_2} b_j x^{[j]}.$$

Define $+$ on $\mathscr{L}_{q^m}[X]$ as follows:

$$L_1(x) + L_2(x) = \sum_{k=0}^{M} (a_k + b_k) x^{[k]}, \tag{2.4}$$

where $M = \max\{d_1, d_2\}$, and we set $a_{d_1+1}, \ldots, a_{d_2}$ equal to zero if $d_2 > d_1$, similarly for $b_{d_2+1}, \ldots, b_{d_1}$ if $d_1 > d_2$. Furthermore, define $\otimes$ on $\mathscr{L}_{q^m}[X]$ as follows:

$$L_1(x) \otimes L_2(x) = L_1(L_2(x)).$$

With these definitions, $\mathscr{L}_{q^m}[X]$ is a non-commutative ring with unity.

Proof. For the addition, we have that in eq. (2.4), $a_k + b_k \in \mathbb{F}$ because $\mathbb{F}$ is a field. Also, the $L_1(x) + L_2(x) = L_2(x) + L_1(x)$ because addition in a field is commutative. Associativity is also clear for the same reason. The additive identity is zero polynomial, which is trivially a linearized polynomial. Existence and uniqueness of an additive inverse for $L(x) \in \mathscr{L}_{q^m}[X]$ follows from the fact that field elements have additive inverses.

For the $\otimes$ operation, we have

$$\begin{aligned} L_1(x) \otimes L_2(x) &= L_1(L_2(x)) = L_1\left(\sum_{j=0}^{d_2} b_j x^{[j]}\right) \\ &= \sum_{i=0}^{d_1} a_i \left(\sum_{j=0}^{d_2} b_j x^{[j]}\right)^{[i]} = \sum_{i=0}^{d_1}\sum_{j=0}^{d_2} a_i b_j^{[i]} x^{[i+j]} = \sum_{k=0}^{d_1+d_2} c_k x^{[k]}, \end{aligned}$$

where $c_k = \sum_{\ell=0}^{k} a_\ell b_{k-\ell}^{[\ell]}$. Because $c_k \in \mathbb{F}$, $L_1(x) \otimes L_2(x) \in \mathscr{L}_{q^m}[X]$, and has degree $q^{d_1+d_2}$. The multiplicative identity is the linearized polynomial $\sum_{i=0}^{d_1} a_i x^{[i]}$, with $a_0 = 1$ and $a_i = 0$ for $i \geq 1$. That is, the multiplicative identity is the polynomial x. □

To show that $\otimes$ is a non-commutative operation in $\mathscr{L}_{q^m}[X]$, we give an example of two linearized polynomial which do not commute.

Example 2.11. Let $f(x), g(x) \in \mathscr{L}_{q^m}[X]$, where

$$f(x) = a_0 x^{[0]} + a_1 x^{[1]} \text{ and } g(x) = b_0 x^{[0]} + b_1 x^{[1]}.$$

Then $f(x) \otimes g(x) = \sum_{k=0}^{2} c_k x^{[k]}$, where

$$\begin{aligned} c_0 &= a_0 b_0^{[0]} = a_0 b_0, \\ c_1 &= a_0 b_1^{[0]} + a_1 b_0^{[1]} = a_0 b_1 + a_1 b_0^{[1]}, \\ c_2 &= a_0 b_2^{[0]} + a_1 b_1^{[1]} + a_2 b_0^{[2]} = a_1 b_1^{[1]}. \end{aligned}$$

On the other hand, $g(x) \otimes f(x) = \sum_{k=0}^{2} d_k x^{[k]}$, where

$$\begin{aligned} d_0 &= b_0 a_0^{[0]} = b_0 a_0, \\ d_1 &= b_0 a_1^{[0]} + b_1 a_0^{[1]} = b_0 a_1 + b_1 a_0^{[1]}, \\ d_2 &= b_0 a_2^{[0]} + b_1 a_1^{[1]} + b_2 a_0^{[2]} = b_1 a_1^{[1]}. \end{aligned}$$

We see that c_1 does not necessarily equal d_1, similarly for c_2 and d_2. Therefore, $\mathscr{L}_{q^m}[X]$ is a non-commutative ring.

For a linearized polynomial $f(x) \in \mathscr{L}_{q^m}[X]$, we use the notation $f^{\otimes n}(x)$ for $\underbrace{f(x) \otimes \cdots \otimes f(x)}_{n \text{ times}}$. With this notation, $f^{\otimes 1}(x) = f(x)$, and $f^{\otimes 0}(x) = x$. Also, we write $f(x) \equiv 0$ if $f(x)$ is the zero polynomial, and we write $f(x) \equiv g(x)$ if $f(x) - g(x) \equiv 0$.

Ordinary multiplication works in $\mathscr{L}_{q^m}[X]$ in special cases, as the following remark shows.

Remark 2.12. Let $f(x) \in \mathscr{L}_{q^m}[X]$, where

$$f(x) = \sum_{i=0}^{d} a_i x^{[i]}.$$

Then $f^q(x) = \underbrace{f(x) \cdots f(x)}_{q \text{ times}}$ is also a linearized polynomial. To see this, note that $(f(x) + g(x))^q = f(x)^q + g(x)^q$, for $f(x), g(x) \in \mathscr{L}_{q^m}[X]$. Therefore,

$$\begin{aligned} f^q(x) = (f(x))^q &= \left(\sum_{i=0}^{d} a_i x^{[i]} \right)^q \\ &= \sum_{i=0}^{d} a_i^q (x^{[i]})^q \\ &= \sum_{i=0}^{d} a_i^q x^{[i+1]}, \end{aligned}$$

which is a linearized polynomial.

Even though the ring $\mathscr{L}_{q^m}[X]$ is non-commutative, as was shown in ex. 2.11, if we only consider linearized polynomials over $\mathbb{F}_q$, the operation $\otimes$ is commutative.

Proposition 2.13. *The set $\mathscr{L}_q[X]$ consisting of linearized polynomials with coefficients in $\mathbb{F}_q$ is a commutative subring of $\mathscr{L}_{q^m}[X]$.*

Proof. It is clear that any linearized polynomial in $\mathscr{L}_q[X]$ is also contained in $\mathscr{L}_{q^m}[X]$, since $\mathbb{F}_q$ is a subfield of $\mathbb{F}_{q^m}$. Also, by adding two linearized polynomials in $\mathscr{L}_q[X]$, the resulting linearized polynomial is also in $\mathscr{L}_q[X]$, because the coefficients are from $\mathbb{F}_q$, and a field is closed under addition. For the $\otimes$ operation, let

$$f(x) = \sum_{i=0}^{d_1} a_i x^{[i]} \quad \text{and} \quad g(x) = \sum_{j=0}^{d_2} b_j x^{[j]}.$$

Then $f(x) \otimes g(x) = \sum_{k=0}^{d_1+d_2} c_k x^{[k]}$, while $g(x) \otimes f(x) = \sum_{k=0}^{d_1+d_2} d_k x^{[k]}$. Because

$$c_k = \sum_{\ell=0}^{k} a_\ell b_{k-\ell}^{[\ell]} = \sum_{\ell=0}^{k} a_\ell b_{k-\ell} \quad \text{and } d_k = \sum_{\ell=0}^{k} b_\ell a_{k-\ell}^{[\ell]} = \sum_{\ell=0}^{k} b_\ell a_{k-\ell},$$

it follows that $c_k = d_k$, because for $\alpha \in \mathbb{F}_q$, $\alpha^{[k]} = \alpha^{q^k} = \alpha$. □

One property of $\mathscr{L}_{q^m}[X]$ that will be used later is the fact that it does not contain any zero divisors.

Proposition 2.14. *The ring $\mathscr{L}_{q^m}[X]$ does not contain any zero divisors.*

Proof. Suppose

$$f(x) \otimes g(x) \equiv 0,$$

where $f(x), g(x) \in \mathscr{L}_{q^m}[X]$, with

$$f(x) = \sum_{i=0}^{d_1} a_i x^{[i]} \text{ and } g(x) = \sum_{j=0}^{d_2} b_j x^{[j]}.$$

We must show that at least one of these polynomials is the zero polynomial. Suppose $f(x)$ is not the zero polynomial. Let ℓ be the smallest index such that $a_\ell \neq 0$, thus we assume $a_0 = 0, a_1 = 0, \ldots, a_{\ell-1} = 0$. We must then show that $g(x) \equiv 0$, that is, $b_j = 0$, for $0 \leq j \leq d_2$. The proof is by induction on j. For $j = 0$, we have

$$\begin{aligned} c_\ell &= a_0 b_\ell^{[0]} + a_1 b_{\ell-1}^{[1]} + \cdots + a_{\ell-1} b_1^{[\ell-1]} + a_\ell b_0^{[\ell]} \\ &= a_\ell b_0^{[\ell]} = 0. \end{aligned}$$

Because $a_\ell \neq 0$, we have $b_0^{[\ell]} = 0$, and therefore, $b_0 = 0$, because $b_0 \in \mathbb{F}_{q^m}$. This establishes the base case of the induction proof.

For the induction step, assume that $b_k = 0$, we must show that this implies that $b_{k+1} = 0$. We have

$$\begin{aligned} c_{\ell+k+1} =& a_0 b_{\ell+k+1}^{[0]} + \cdots + a_{\ell-1} b_k^{[\ell-1]} \\ &+ a_\ell b_{k+1}^{[\ell]} + a_{\ell+1} b_k^{[\ell+1]} + \cdots + a_{\ell+k+1} b_0^{[\ell+k+1]} \\ =& a_\ell b_{k+1}^{[\ell]} = 0, \end{aligned}$$

and because $a_\ell \neq 0$, we have $b_{k+1}^{[\ell]} = 0$, and therefore, $b_{k+1} = 0$. By induction, $g(x)$ is the zero polynomial. Therefore, the ring $\mathscr{L}_{q^m}[X]$ does not contain any zero divisors. □

Corollary 2.15. *The ring $\mathscr{L}_q[X]$ is an integral domain.*

Proof. From prop. 2.13, $\mathscr{L}_q[X]$ is a commutative ring. Because $\mathscr{L}_{q^m}[X]$ does not contain any zero divisors, neither does $\mathscr{L}_q[X]$. □

We also define bivariate linearized polynomials as follows.

Definition 2.16 (Bivariate Linearized Polynomial)**.** Let $\mathbb{F}_q$ be a finite field, and let $\mathbb{F} = \mathbb{F}_{q^m}$ be an extension field of $\mathbb{F}_q$. A bivariate polynomial $L(x, y)$ over $\mathbb{F}$ is called a linearized polynomial if

$$L(x, y) = \sum_{i=0}^{d_1} a_i x^{[i]} + \sum_{j=0}^{d_2} b_j y^{[j]}.$$

Notice that a bivariate linearized polynomial has no terms of the form $x^{[i]}y^{[j]}$. We also define a n variate linearized polynomial $Q(x_1, \ldots, x_n)$ as

$$Q(x_1, \ldots, x_n) = \sum_{i_1=0}^{d_1} a_{1,i_1} x^{[i_1]} + \cdots + \sum_{i_n=0}^{d_n} a_{n,i_n} x^{[i_n]},$$

where the coefficients are in $\mathbb{F}_{q^m}$. The reason for the name linearized polynomial is that a linearized polynomial over $\mathbb{F}_{q^m}$ defines a linear map with respect to $\mathbb{F}_q$. This is shown in the next proposition.

Proposition 2.17. *Given a finite extension field $\mathbb{K}$ of $\mathbb{F} = \mathbb{F}_{q^m}$, and a linearized polynomial $L(x)$ over $\mathbb{F}$, the map*

$$L : \mathbb{K} \to \mathbb{K}, \quad \beta \mapsto L(\beta)$$

is linear over $\mathbb{F}_q$. This means that for all $\beta_1, \beta_2 \in \mathbb{K}$ and all $\lambda \in \mathbb{F}_q$, we have

$$\begin{aligned} L(\beta_1 + \beta_2) &= L(\beta_1) + L(\beta_2), \\ L(\lambda\beta_1) &= \lambda L(\beta_1). \end{aligned}$$

Proof. Let

$$L(x) = \sum_{i=0}^{d} a_i x^{[i]}$$

be a linearized polynomial over $\mathbb{F} = \mathbb{F}_{q^m}$, and let $\mathbb{K}$ be an extension field of $\mathbb{F}$. For the first condition of linearity, we have that

$$\begin{aligned} L(\beta_1) + L(\beta_2) &= \sum_{i=0}^{d} a_i \beta_1^{[i]} + \sum_{i=0}^{d} a_i \beta_2^{[i]} \\ &= \sum_{i=0}^{d} a_i (\beta_1^{[i]} + \beta_2^{[i]}) = \sum_{i=0}^{d} a_i (\beta_1 + \beta_2)^{[i]} \\ &= L(\beta_1 + \beta_2), \end{aligned}$$

where we have used that $\beta_1^{[i]} + \beta_2^{[i]} = (\beta_1 + \beta_2)^{[i]}$ holds for all elements in $\mathbb{K}$. To verify that L is scalar multiplication preserving, we have

$$\begin{aligned} L(\lambda\beta_1) &= \sum_{i=0}^{d} a_i (\lambda\beta_1)^{[i]} = \sum_{i=0}^{d} a_i \lambda^{[i]} \beta_1^{[i]} \\ &= \sum_{i=0}^{d} a_i \lambda \beta_1^{[i]} = \lambda \sum_{i=0}^{d} a_i \beta_1^{[i]} \\ &= \lambda L(\beta_1), \end{aligned}$$

since $\lambda^{[i]} = \lambda$ for all $\lambda \in \mathbb{F}_q$. Therefore, $L(x)$ defines a linear map over $\mathbb{F}_q$. □

Remark 2.18. The kernel of $L(x)$ is the set

$$\ker(L(x)) = \{\beta \in \mathbb{K} \mid L(\beta) = 0\}$$

is a subspace of $\mathbb{K}$, and the dimension of this subspace is k, where q^k is the degree of $L(x)$.

This next result shows that if two linearized polynomials agree on sufficiently many points, then they are equal.

Proposition 2.19. *Let $f(x), g(x)$ be linearized polynomials over $\mathbb{F} = \mathbb{F}_{q^m}$ of degree less than q^d, and let $\alpha_1, \alpha_2, \ldots, \alpha_d$ be linearly independent elements of $\mathbb{K}$, such that $f(\alpha_i) = g(\alpha_i)$, for $1 \leq i \leq d$, where $\mathbb{K}$ is an extension field of $\mathbb{F}$. Then $f(x) \equiv g(x)$.*

Proof. Let $h(x) = f(x) - g(x)$. Then $h(\alpha_i) = 0$, for all α_i, $1 \leq i \leq d$. Let

$$\gamma_1\alpha_1 + \cdots + \gamma_d\alpha_d,$$

be an arbitrary linear combination in $\mathbb{K}$, where $\gamma_i \in \mathbb{F}_q$. Because $f(x), g(x)$ have degree less than q^d, so does their difference $h(x)$. However, there are q^d linear combinations possible, because there are d linearly independent elements α_i, and the scalars γ_i are from $\mathbb{F}_q$. This implies that $h(x)$ has q^d zeros, but the degree of $h(x)$ is less than q^d, so $h(x)$ is the zero polynomial. We conclude that $h(x) \equiv 0$, so $f(x) - g(x) \equiv 0$, we therefore have $f(x) = g(x)$. □

The following proposition shows the connection between linearized polynomials and vector spaces over finite fields, and is based on [MS77, lem.21, p.119].

Proposition 2.20. *Let V be an n-dimensional subspace of $\mathbb{K}$, where $\mathbb{K}$ is an extension field of $\mathbb{F}_{q^m}$. Then the polynomial*

$$L(x) = \prod_{\beta \in V} (x - \beta)$$

is a monic linearized polynomial over $\mathbb{K}$. That is, there exits $\ell_0, \ldots, \ell_{n-1} \in \mathbb{K}$ such that

$$L(x) = x^{[n]} + \sum_{i=0}^{n-1} \ell_i x^{[i]} \tag{2.5}$$

Proof. Let $B = \{\beta_0, \beta_1, \ldots, \beta_{n-1}\}$ be a basis for V. We want to construct a polynomial of the form in eq. (2.5), which has $\beta_0, \beta_1, \ldots, \beta_{n-1}$ as roots. Consider the n equations in n unknowns

$$\beta_i^{[n]} + \sum_{j=0}^{n-1} \ell_j \beta_i^{[j]} = 0, \ i = 0, 1, \ldots, n-1,$$

where $\ell_0, \ldots, \ell_{n-1} \in \mathbb{K}$. We can rewrite this in matrix form as

$$\begin{bmatrix} \beta_0^{[0]} & \beta_0^{[1]} & \beta_0^{[2]} & \cdots & \beta_0^{[n-1]} \\ \beta_1^{[0]} & \beta_1^{[1]} & \beta_1^{[2]} & \cdots & \beta_1^{[n-1]} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \beta_{n-1}^{[0]} & \beta_{n-1}^{[1]} & \beta_{n-1}^{[2]} & \cdots & \beta_{n-1}^{[n-1]} \end{bmatrix} \begin{bmatrix} \ell_0 \\ \ell_1 \\ \vdots \\ \ell_{n-1} \end{bmatrix} = - \begin{bmatrix} \beta_0^{[n]} \\ \beta_1^{[n]} \\ \vdots \\ \beta_{n-1}^{[n]} \end{bmatrix}, \tag{2.6}$$

or more compactly, $\mathbf{B}\ell = -\beta$. We now want to show that $\mathbf{B}$ is invertible, because then the system of equations in (2.6) is consistent.

The vector space V contains q^n elements. But the field $\mathbb{F}_{q^n}$ can be regarded as a vector space of dimension n over $\mathbb{F}_q$, and so V and $\mathbb{F}_{q^n}$ are isomorphic as vector spaces. But $\mathbb{F}_{q^n} = \mathbb{F}_q[X]/\langle \pi(x) \rangle$, where $\pi(x)$ is an irreducible polynomial of degree n over $\mathbb{F}_q$. Let $A = \{1, \alpha, \alpha^2, \ldots, \alpha^{n-1}\}$ be a basis for V, where $\pi(\alpha) = 0$. We can write each element in A in terms of the elements in B, namely

$$\alpha^k = c_{k,0}\beta_0 + c_{k,1}\beta_1 + \cdots + c_{k,n-1}\beta_{n-1}, \ \ k = 0, 1, \ldots, n-1.$$

In matrix form, the above can be written as

$$\begin{bmatrix} 1 \\ \alpha \\ \alpha^2 \\ \vdots \\ \alpha_{n-1} \end{bmatrix} = \begin{bmatrix} c_{0,0} & c_{0,1} & \cdots & c_{0,n-1} \\ c_{1,0} & c_{1,1} & \cdots & c_{1,n-1} \\ \vdots & \vdots & \vdots & \vdots \\ c_{n-1,0} & c_{n-1,1} & \cdots & c_{n-1,n-1} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{n-1} \end{bmatrix},$$

or more succinctly, $\alpha = \mathbf{C}\beta$. Let $\alpha^k \in A$, and let $0 \leq j \leq n-1$. Then

$$\begin{aligned} (\alpha^k)^{[j]} &= (c_{k,0}\beta_0 + c_{k,1}\beta_1 + \cdots + c_{k,n-1}\beta_{n-1})^{[j]} \\ &= (c_{k,0}\beta_0)^{[j]} + (c_{k,1}\beta_1)^{[j]} + \cdots + (c_{k,n-1}\beta_{n-1})^{[j]} \\ &= c_{k,0}\beta_0^{[j]} + c_{k,1}\beta_1^{[j]} + \cdots + c_{k,n-1}\beta_{n-1}^{[j]}, \end{aligned}$$

because $c_{k,m} \in \mathbb{F}_q$, so $c_{k,m}^{[j]} = c_{k,m}^{q^j} = c_{k,m}$. Therefore, we have

$$\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ \alpha^{[0]} & \alpha^{[1]} & \alpha^{[2]} & \cdots & \alpha^{[n-1]} \\ (\alpha^2)^{[0]} & (\alpha^2)^{[1]} & (\alpha^2)^{[2]} & \cdots & (\alpha^2)^{[n-1]} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ (\alpha^{n-1})^{[0]} & (\alpha^{n-1})^{[1]} & (\alpha^{n-1})^{[2]} & \cdots & (\alpha^{n-1})^{[n-1]}. \end{bmatrix} = \mathbf{CB}.$$

Because this matrix is invertible (see [MS77, p.118]), so is $\mathbf{B}$. Therefore, there exist $\ell_0, \ell_1, \ldots, \ell_{n-1}$, such that the system defined in (2.6) is consistent. Therefore, $\beta_0, \beta_1, \ldots, \beta_{n-1}$ are roots in $L(x)$, where

$$L(x) = x^{[n]} + \sum_{i=0}^{k-1} \ell_i x^{[i]}.$$

Now, $L(x)$ is a linearized polynomial, so if $\beta_i, \beta_j \in B$ and $\lambda, \mu \in \mathbb{F}_q$, we have

$$L(\lambda\beta_i + \mu\beta_j) = L(\lambda\beta_i) + L(\mu\beta_j) = 0,$$

which shows that any linear combination of the β's is also a root of $L(x)$. Therefore, we can write

$$L(x) = \prod_{\beta \in V} (x - \beta).$$

□

2.5 Modules

In this section, we define an algebraic structure called a module, and look at some of the mathematical properties that such a structure has. The theory of modules is used later when we look at decoding of subspace codes. This material is adapted from [DF04]. In this section, we do not assume that the rings are commutative.

Definition 2.21 (Module)**.** Let R be a ring with unity 1 (but not necessarily commutative), and M be a non-empty set. A left R-module M is a set with a binary operation $+$ which makes $(M, +)$ an abelian group, and an action $\circ : R \times M \to M$ defined by $r \circ m = rm$. The action $\circ$ satisfies the following conditions for all $r, s \in R$ and all $m, m' \in M$:

1. $(r + s)m = rm + sm$,
2. $r(m + m') = rm + rm'$,
3. $(rs)m = r(sm)$,
4. $1m = m$.

In other words, an R-module M is a set which is an abelian group under addition, and where we can left multiply by "scalars" from the ring R. For a right R-module M, the action $\circ$ is defined as $m \circ r = mr$, for $r \in R$ and $m \in M$. If the ring R is commutative, then we can make a left R-module M into a right R-module by defining $mr = rm$, for $m \in M$ and $r \in R$. Also, if R is a field $\mathbb{F}$, then a $\mathbb{F}$-module M is a vector space over $\mathbb{F}$.

Definition 2.22 (Submodule)**.** Let R be a ring, and M be an R-module. A subset N of M is called a R-submodule is N is a subgroup of M, and which is closed under the action $\circ$, i.e. $r \circ n = rn \in N$, for all $r \in R$ and all $n \in N$.

We note that since M is an abelian group, so is any subgroup of M, and therefore, the above definition makes sense.

Proposition 2.23. *Let $M_1, \ldots, M_n$ be R-modules. Then the intersection $M_1 \cap \cdots \cap M_n$ is also an R-module.*

Example 2.24. Let G be an abelian group. Then G is a $\mathbb{Z}$-module. For $g \in G$ and $n \in \mathbb{Z}$, the action $\circ$ is defined as

$$ng = \begin{cases} \underbrace{g + \cdots + g}_{n \text{ times}} & \text{if } n > 0 \\ 0 & \text{if } n = 0 \\ \underbrace{-g - \cdots - g}_{n \text{ times}} & \text{if } n < 0 \end{cases}$$

Example 2.25. The ring R is an R-module over itself. This is because $(R, +)$ is an abelian group, and the action $\circ$ is the multiplication of ring elements.

Similarly for what is done for groups and rings, one can define structure-preserving maps between modules.

Definition 2.26 (Module homomorphism)**.** Let R be a ring, and let M and N be R-modules. A mapping $\phi : M \to N$ satisfying

1. $\phi(m + m') = \phi(m) + \phi(m')$,
2. $\phi(rm) = r\phi(m)$,

for all $m, m' \in M$ and all $r \in R$, is called an R-module homomorphism, or just module homomorphism for short.

Given an R-module homomorphism $\phi : M \to N$, it can be shown that the kernel $\ker(\phi)$ and image $\mathrm{im}(\phi)$ are R-submodules of M and N respectively. Note that if R is a field $\mathbb{F}$, then a $\mathbb{F}$ module homomorphism from M to N is a linear map.

Definition 2.27 (Free Module)**.** Let R be a ring with unity. An R-module M is called free on A, where $A \subseteq M$, if for every $m \in M$, there are unique ring elements $r_1, \ldots, r_n \in R$ and unique $a_1, \ldots, a_n \in A$ such that

$$m = r_1 a_1 + \cdots + r_n a_n.$$

The set A is called a basis for the module M.

The next result shows that when one has a non-empty set A, and a ring R, then one can construct a free R-module which has the elements in A as a basis.

Theorem 2.28. *Let A be a set, and let R be a ring with unity 1. Then there exists a free R-module on A.*

Proof. Let $F(A)$ be the set of all functions f mapping A to R, such that $f(a) = 0$ for all but finitely many $a \in A$. We define addition of two elements in $F(A)$ and multiplication of a $f \in F(A)$ by a $r \in R$ pointwise. For all $a \in A$, all $f, g \in F(A)$ and all $r \in R$, let

$$\begin{aligned}(f+g)(a) &= f(a) + g(a)\\ (rf)(a) &= r(f(a)).\end{aligned}$$

Note that $f(a), g(a) \in R$. We first show that $(F(A), +)$ is an abelian group. The additive identity in $F(A)$ is the zero function $0(a) = 0$, for all $a \in A$, since $f(a)+0(a) = f(a)$. For $f \in F(A)$, the additive inverse is $-f$, defined by $(-f)(a) = -f(a)$. We have $(f+(-f))(a) = f(a)+(-f)(a) = f(a)-f(a) = 0$. The addition is also commutative, since for $f, g \in F(A)$,

$$(f+g)(a) = f(a) + g(a) = g(a) + f(a) = (g+f)(a),$$

because addition of ring elements is commutative. Also, we have that for $r, s \in R$, and $f, g \in F(A)$,

$$\begin{aligned}(r+s)f(a) &= rf(a) + sf(a),\\ r(f(a)+g(a)) &= rf(a) + rg(a),\\ (rs)f(a) &= r(sf(a)), \text{ and}\\ 1f(a) &= f(a),\end{aligned}$$

from the distributive and associative laws in the ring R. This shows that $F(A)$ is an R-module.

We now show how to identify the set A with a subset of $F(A)$. Let $f_a \in F(A)$ be the function defined by

$$f_a(x) = \begin{cases}1, & \text{if } x = a,\\ 0, & \text{otherwise}\end{cases}$$

where 1 is the multiplicative identity in R. In this way, any $a \in A$ can be identified with a unique function $f_a \in F(A)$. This can be seen as follows. Let $F_A = \{f_{a_1}, f_{a_2}, f_{a_3}, \ldots\}$. Define a map $\Phi : A \to F_A$ by $a \mapsto f_a$. We show that Φ is a bijection. Suppose $\Phi(a) = \Phi(a')$, then $f_a = f_{a'}$. So

$$1 = f_a(a) = f_{a'}(a) = f_{a'}(a') = 1,$$

and so $a = a'$. Therefore, Φ is injective. For surjectivity, let $f_a \in F_A$. Then $a \in A$ satisfies $\Phi(a) = f_a$. We see that Φ is a bijection, so A can be identified with the set $F_A \subset F(A)$.

We now show that the set A is a basis for the module. Let $\{a_1, \ldots, a_n\} \subset A$. Then the function $f \in F(A)$, defined as

$$f(x) = \begin{cases} r_i, & \text{if } x = a_i, \\ 0, & \text{otherwise} \end{cases}$$

can be written uniquely as

$$r_1 a_1 + \cdots + r_n a_n.$$

For example, $f(a_j) = r_j f_{a_j} = r_j$, where f_{a_j} is identified with $a_j \in A$ using the bijection Φ. □

The last definition pertains to linear algebra.

Definition 2.29 (Linear Functional)**.** Let V be a vector space over a field $\mathbb{F}$. A linear mapping $f : V \to \mathbb{F}$ is called a linear functional.

One example of a linear functional is the evaluation mapping.

Example 2.30. Consider the polynomial ring $\mathbb{F}[x]$, where $\mathbb{F}$ is a field. The ring $\mathbb{F}[x]$ is also a vector space over $\mathbb{F}$, where the vector space operations are ordinary addition of polynomials and multiplication of a polynomial by a field element. Define a mapping $\mathsf{ev}_a : \mathbb{F}[x] \to \mathbb{F}$ by $\mathsf{ev}_a(f) = f(a)$, for $a \in \mathbb{F}$. Then ev_a is linear, because

$$\begin{aligned} \mathsf{ev}_a(f+g) = (f+g)(a) = f(a) + g(a) = \mathsf{ev}_a(f) + \mathsf{ev}_a(g), \text{ and} \\ \mathsf{ev}_a(\alpha f) = (\alpha f)(a) = \alpha f(a) = \alpha \mathsf{ev}_a(f), \end{aligned}$$

for $f \in \mathbb{F}[x]$ and $\alpha \in \mathbb{F}$. The kernel of ev_a is the set of polynomials which have a as a root.

Chapter 3

Coding for the Operator Channel

In this chapter, we start by defining subspace codes, and look at various code parameters. Minimum distance decoding of such codes is also examined, along with an upper bound on how many codewords a subspace code can have. Then, we introduce the Koetter-Kschischang kodes (KK-codes for short), which are defined using linearized polynomials. These codes were first defined in [KK08]. A decoding procedure of KK-codes is also considered.

3.1 The Operator Channel

We start this chapter by looking at a special type of communications channel, known as the operator channel. This channel was introduced in [KK08], and its input and output alphabet are the set of all subspaces of a vector space W. This set is denoted as $\mathcal{P}(W)$, and the set of all ℓ-dimensional subspaces of W is denoted $\mathcal{P}(W, \ell)$. In this set-up, there is one sender and one receiver. The sender wants to send an element V from $\mathcal{P}(W)$ to the receiver, who receives U. We assume that both the sender and receiver use the same alphabet $\mathcal{P}(W)$. The communications channel is called an operator channel, and the input V and output U are related as

$$U = \mathcal{H}_k(V) \oplus E. \tag{3.1}$$

In eq. (3.1), $\mathcal{H}_k(V)$ is an mapping which maps a subspace from W into another subspace of W as follows: If $k < \dim(V)$, then this mapping returns a randomly chosen k-dimensional subspace of V, otherwise it returns V. This is motivated from the discussion of the channel model in sec. 1.3. From this

definition, transmitting a subspace V through the channel, we get a space $\mathcal{H}_k(V)$, where the dimension is reduced by $\dim(V) - k$, compared to the dimension of V. We will call this quantity ρ, and say that the channel does ρ erasures. The space E is an error space, and corresponds to the errors done by the channel. We assume that E is a subspace of W. Let $\dim(E) = t$, then we say that the channel does t errors. Letting $k = \dim(U \cap V)$, we have that $\mathcal{H}_k(V)$ is isomorphic to $U \cap V$. So we can write $U = (U \cap V) \oplus E$, where E is isomorphic to the quotient space $U/(U \cap V)$ (see sec. 2.1). Recall from that section that the quotient space construction is independent of a basis chosen the spaces, which makes it useful in the channel model, since we do not prefer any particular basis for the input and output spaces.

3.2 Code Parameters

In this section, we formally define subspace codes, and introduce their parameters. We also look at dual codes of such codes, and treat the topic of minimum distance decoding.

Definition 3.1 (Subspace codes)**.** Let W be an N-dimensional vector space over $\mathbb{F}_q$, and consider $\mathcal{P}(W)$, the set of all subspaces of W. A subspace code $\mathcal{C}$ is a nonempty subset of $\mathcal{P}(W)$. The code $\mathcal{C}$ has parameters

$$\left[N, \ell(\mathcal{C}), \log_q(|\mathcal{C}|), D(\mathcal{C})\right],$$

where

$$\ell(\mathcal{C}) = \max_{X \in \mathcal{C}} \dim(X),$$

is the maximum dimension of any codeword in $\mathcal{C}$. The number of codewords in $\mathcal{C}$ is $|\mathcal{C}|$, and

$$D(\mathcal{C}) = \min_{\substack{X,Y \in \mathcal{C} \\ X \neq Y}} d(X, Y)$$

is the minimum distance of $\mathcal{C}$, where d is the subspace distance defined in def. 2.2.

We will write D instead of $D(\mathcal{C})$ when specifying the parameters of a code. Also, if all codewords in a code have the same dimension, the we write ℓ instead of $\ell(\mathcal{C})$. In this case, the code is called constant-dimensional, and the codewords are a subset of $\mathcal{P}(W, \ell)$. We now define normalized parameters for a subspace code $\mathcal{C}$. In the following, let $\mathcal{C}$ be a subspace code with parameters $\left[N, \ell, \log_q(|\mathcal{C}|), D\right]$. We first define the rate R of $\mathcal{C}$.

Definition 3.2. The rate R of a subspace code $\mathcal{C}$ is defined as

$$R = \frac{\log_q(|\mathcal{C}|)}{N\ell}. \tag{3.2}$$

Note that in order to send a message, this message is first mapped to a codeword $C \in \mathcal{C}$. The encoding is injective, that is, if $\mathbf{x}, \mathbf{y}$ are two messages and $\mathbf{x} \neq \mathbf{y}$, then $C(\mathbf{x}) \neq C(\mathbf{y})$. From the parametes of $\mathcal{C}$, in order to specify a codeword C (which is a subspace), the sender needs to send at least $N\ell$ symbols from $\mathbb{F}_q$. Any symbol sent in excess of $N\ell$ is redundant. We also define the normalized weight λ as

$$\lambda = \frac{\ell}{N},$$

and the normalized distance δ is defined as

$$\delta = \frac{D}{2\ell}.$$

Both λ and δ are constrained to the range $[0, 1]$. The range of λ follows from the fact that $\ell \leq N$. Because $\mathcal{C}$ is constant dimensional, the minimum distance of $\mathcal{C}$ is less than or equal to $\dim(X) + \dim(Y) = \ell + \ell = 2\ell$, for some $X, Y \in \mathcal{C}$.

Example 3.3. Let $W = \mathbb{F}_q^N$, and consider the matrices $[\mathbf{I}|\mathbf{A}_i]$, where $\mathbf{I}$ is the $\ell \times \ell$ identity matrix and $\mathbf{A}_i$ is a $\ell \times (N - \ell)$ matrix with entries from $\mathbb{F}_q$. Denote the row span of $[\mathbf{I}|\mathbf{A}_i]$ over $\mathbb{F}_q$ by $G(U_i)$. For $i \neq j$, let the matrices $\mathbf{A}_i$ and $\mathbf{A}_j$ be different in at least one position, we have that the matrices $[\mathbf{I}|\mathbf{A}_i]$ and $[\mathbf{I}|\mathbf{A}_j]$ are different in at least one row. This means that $G(U_i)$ and $G(U_j)$ intersect in a subspace of dimension at most $\ell - 1$. All spaces $G(U_i)$ are of dimension ℓ. Turning to the minimum distance, we have

$$\begin{aligned} d(G(U_i), G(U_j)) &\geq \ell + \ell - 2(\ell - 1) \\ &= 2\ell - 2(\ell - 1) = 2. \end{aligned}$$

Furthermore, because there are $q^{\ell(N-\ell)}$ different $\ell \times (N - \ell)$ matrices with entries from $\mathbb{F}_q$, the number of codewords is $q^{\ell(N-\ell)}$. So the set $\mathcal{C}$ consisting of the subspaces $G(U_i)$ is a constant dimensional subspace code with parameters $[N, \ell, \ell(N - \ell), 2]$. This code has normalized weight $\lambda = \frac{\ell}{N}$, and rate

$$R = \frac{\ell(N - \ell)}{N\ell} = \frac{N - \ell}{N} = 1 - \frac{\ell}{N} = 1 - \lambda.$$

The normalized distance of $\mathcal{C}$ is $\delta = \frac{D}{2\ell} = \frac{1}{\ell} = \frac{1}{\lambda N}$.

3.3 Dual Codes of Subspace Codes

In order to define the dual code $\mathcal{C}^\perp$, we first recall the following. Let $\mathbf{u}, \mathbf{v} \in \mathbb{F}_q^N$, and let

$$\langle \mathbf{u}, \mathbf{v} \rangle = \sum_{i=0}^{N} u_i v_i. \tag{3.3}$$

It can be shown that this is a valid inner product on $\mathbb{F}_q^N$. Let U be a subspace of W, and define $U^\perp$ as

$$U^\perp = \{\mathbf{v} \in W \mid \langle \mathbf{v}, \mathbf{u} \rangle = 0, \text{ for all } \mathbf{u} \in U\}.$$

The set $U^\perp$ is called the orthogonal complement of U. Using the above, we can define the dual code $\mathcal{C}^\perp$.

Definition 3.4 (Dual Code)**.** Let $\mathcal{C}$ be a subspace code, and $\langle \cdot, \cdot \rangle$ be the inner product on $\mathbb{F}_q^N$ defined in eq. (3.3). Then the dual code of $\mathcal{C}$ is the set of subspaces

$$\mathcal{C}^\perp = \{U^\perp | U \in \mathcal{C}\}.$$

We focus on the case where $\mathcal{C}$ is a constant dimensional code, with $\ell = \ell(\mathcal{C})$. It can be seen that $\mathcal{C}^\perp$ is a subspace code with parameters $\left[N, N-\ell, \log_q(|\mathcal{C}^\perp|), D^\perp\right]$. We show that a subspace code $\mathcal{C}$ and its dual code $\mathcal{C}^\perp$ have the same minimum distance.

Proposition 3.5. *Let $\mathcal{C}$ be a subspace code with minimum distance D. Then the dual code $\mathcal{C}^\perp$ also has minimum distance D.*

Proof. Let $U^\perp, V^\perp \in \mathcal{C}^\perp$. Using $U^\perp + V^\perp = (U \cap V)^\perp$ and $U^\perp \cap V^\perp = (U+V)^\perp$, we have

$$\begin{aligned} d(U^\perp, V^\perp) &= \dim(U^\perp + V^\perp) - \dim(U^\perp \cap V^\perp) \\ &= \dim((U \cap V)^\perp) - \dim((U+V)^\perp) \\ &= N - \dim(U \cap V) - (N - \dim(U+V)) \\ &= \dim(U+V) - \dim(U \cap V) \\ &= d(U, V). \end{aligned}$$

Therefore, the result follows. □

3.4 Minimum Distance Decoding

One way of decoding a received subspace is to find the codeword in $\mathcal{C}$ which is closest to it. This method is called minimum distance decoding. Using the distance function d defined in sec. 2.2, one can define a minimum distance decoder for subspace codes.

Definition 3.6 (Minimum Distance Decoder)**.** Let $\mathcal{C}$ be a subspace code, suppose $V \in \mathcal{C}$ is sent over an operator channel, which returns U. A minimum distance decoder takes as input the output U of the operator channel, and returns a space $V' \in \mathcal{C}$, which satisfies

$$d(U, V') \leq d(U, V''),$$

for all $V'' \in \mathcal{C}$.

Proposition 3.7. *Suppose we use a subspace code $\mathcal{C}$ with minimum distance D for transmission over an operator channel. Let $V \in \mathcal{C}$ be transmitted, and let*

$$U = \mathcal{H}_k(V) \oplus E$$

be received, where $k = \dim(V \cap U)$ and E is an error space, whose dimension is t. Let $\rho = (\ell(\mathcal{C}) - k)_+ = \max\{0, \ell(\mathcal{C}) - k\}$ (the positive part of $\ell(\mathcal{C}) - k$)[1] be the maximum number if erasures induced by the operator channel. Then if

$$2(t + \rho) < D, \tag{3.4}$$

a minimum distance decoder will correctly return the transmitted codeword V from the received space U.

Proof. Let $V' = \mathcal{H}_k(V)$. Then we have

$$d(V, U) \leq d(V, V') + d(V', U).$$

Because $d(V, V') \leq \rho$ and $d(V', U) \leq t$, we get the inequality

$$d(V, U) \leq \rho + t.$$

To show that the minimum distance decoder correctly decodes V from U as long as eq. (3.4) holds, let $T \neq V$ be another codeword in $\mathcal{C}$. We then have

$$D \leq d(V, T) \leq d(V, U) + d(U, T),$$

and combining with eq. (3.4), we have

$$\begin{aligned} d(U, V) &\leq \rho + t = 2(\rho + t) - (\rho + t) \\ &< D - (\rho + t) \leq D - d(V, U) \\ &\leq d(U, T). \end{aligned}$$

Therefore, the result follows. □

[1] Note that if $\ell(\mathcal{C}) < k$, then $\mathcal{H}_k(V)$ returns V, that is, $\rho = 0$.

In communicating over an operator channel, two special cases are when $k = \dim(W)$, which corresponds to no erasures, and $E = \{0\}$, which translates into no errors. We have the following result.

Corollary 3.8. *Suppose we use a subspace code $\mathcal{C}$ with minimum distance D for transmission over an operator channel. Let $V \in \mathcal{C}$ be the transmitted codeword. If*

$$U = \mathcal{H}_{\dim(W)}(V) \oplus E = V \oplus E,$$

is received, and $2t < D$, where $t = \dim(E)$, then a minimum distance decoder will correctly return V from U.

Also, if $E = \{0\}$, so

$$U = \mathcal{H}_k(V) \oplus E = \mathcal{H}_k(V),$$

and if $2\rho < D$, where $\rho = (\ell(\mathcal{C}) - k)_+$, then a minimum distance decoder will correctly return V from U.

We can rearrange $2t < D$ (the no-erasure case) to

$$t \leq \left\lfloor \frac{D-1}{2} \right\rfloor. \tag{3.5}$$

Compare this to the error correction of classical coding theory, where a code with minimum distance d can correct up to t errors, as long as $t \leq \left\lfloor \frac{d-1}{2} \right\rfloor$. One difference however is that erasures and errors are equally costly, which is in contrast to the situation in classical coding theory.

3.5 Singleton Bound

In this section, we present an upper bound on the number of codewords in a constant-dimensional subspace code $\mathcal{C}$. To do this, we define a puncturing operation on the codewords in $\mathcal{C}$. Let $\mathcal{C}$ be a subspace code with parameters $\left[N, \ell, \log_q(|\mathcal{C}|), D\right]$, where W is the ambient space and $\dim(W) = N$. Let $V \in \mathcal{C}$, so $\dim(V) = \ell$, and let W' be a subspace of W. Let $V' \in \mathcal{C}'$ be defined as follows. If $\dim(W' \cap V) = \ell - 1$, then let $V' = W' \cap V$, otherwise, let V' be some random $(\ell - 1)$-dimensional subspace of V. We call the code obtained in this way a punctured code $\mathcal{C}'$. We have the following result.

Proposition 3.9. *Let $\mathcal{C}$ be a subspace code with parameters $\left[N, \ell, \log_q(|\mathcal{C}|), D\right]$, where $D > 2$. Then the punctured code $\mathcal{C}'$ has parameters*

$$\left[N - 1, \ell - 1, \log_q(|\mathcal{C}'|), D'\right], \tag{3.6}$$

where $D' \geq D - 2$. Furthermore, $\mathcal{C}'$ has as least as many codewords as $\mathcal{C}$.

Proof. The parameters $N-1$ and $\ell-1$ follow from the way that $\mathcal{C}'$ is defined. We now verify that $D' \geq D-2$. Let $U', V' \in \mathcal{C}'$, and suppose $d(U', V') = D'$. Then $\dim(U') = \dim(V') = \ell - 1$, and we have

$$\begin{aligned} d(U', V') &= \dim(U') + \dim(V') - 2\dim(U' \cap V') \\ &= \ell - 1 + \ell - 1 - 2\dim(U' \cap V') \\ &= 2\ell - 2 - 2\dim(U' \cap V'). \end{aligned}$$

Now, because U' is a subspace of U and V' is a subspace of V, $U' \cap V'$ is a subspace of $U \cap V$, and so

$$\dim(U' \cap V') \leq \dim(U \cap V).$$

Therefore,

$$\begin{aligned} D' = d(U', V') &= 2\ell - 2 - 2\dim(U' \cap V'). \\ &\geq 2\ell - 2 - 2\dim(U \cap V) \\ &= 2\ell - 2\dim(U \cap V) - 2 \\ &= d(U, V) - 2 \\ &\geq D - 2. \end{aligned}$$

Also, the number of codewords in $\mathcal{C}'$ is larger than or equal to $\mathcal{C}$, because $d(U', V') \geq D - 2 > 0$. □

Remark 3.10. The minimum distance D of a $\left[N, \ell, \log_q(|\mathcal{C}|), D\right]$ subspace code $\mathcal{C}$ is even. Let $U, V \in \mathcal{C}$ be two codewords with minimum distance, i.e. $d(U, V) = D$. Then $D = \dim(U) + \dim(V) - 2\dim(U \cap V) = 2\ell - 2\dim(U \cap V) = 2(\ell - \dim(U \cap V))$, which is an even number.

We can now state and prove the Singleton bound, due to [KK08], for subspace codes.

Theorem 3.11 (Singleton bound)**.** *Let W be a N-dimensional vector space over $\mathbb{F}_q$, and let $\mathcal{C} \in \mathcal{P}(W, \ell)$ be a $\left[N, \ell, \log_q(|\mathcal{C}|), D\right]$ subspace code. Then the number of codewords in $\mathcal{C}$ is upper bounded by*

$$\begin{bmatrix} N - (D-2)/2 \\ \max\{\ell, N - \ell\} \end{bmatrix}_q.$$

Proof. From prop. 3.9, by puncturing $\mathcal{C}$, we get a $\left[N-1, \ell-1, \log_q(|\mathcal{C}'|), D'\right]$ code, where $D' \geq D - 2$. If we puncture $\mathcal{C}$ $(D-2)/2$ times, we get a $\left[N - (D-2)/2, \ell - (D-2)/2, \log_q(|\mathcal{C}^*|), D^*\right]$ code, where

$$D^* \geq D - 2((D-2)/2) = D - (D-2) = 2.$$

Note that from remark 3.10, the minimum distance D^* must be even. The code $\mathcal{C}^*$ is a subset of $\mathcal{P}(W^*, \ell^*)$, where $\dim(W^*) = N - (D-2)/2$, and $\ell^* = \ell - (D-2)/2$. From the discussion of the Gaussian coefficient (sec. 2.3),

$$|\mathcal{P}(W^*, \ell^*)| = \begin{bmatrix} N - (D-2)/2 \\ \ell - (D-2)/2 \end{bmatrix}_q.$$

From the identity $\begin{bmatrix} N \\ \ell \end{bmatrix}_q = \begin{bmatrix} N \\ N-\ell \end{bmatrix}_q$ of the Gaussian coefficients [Kon98], we have

$$\begin{bmatrix} N - (D-2)/2 \\ \ell - (D-2)/2 \end{bmatrix}_q = \begin{bmatrix} N - (D-2)/2 \\ N - (D-2)/2 - \ell + (D-2)/2 \end{bmatrix}_q = \begin{bmatrix} N - (D-2)/2 \\ N - \ell \end{bmatrix}_q,$$

so the number of codewords in $\mathcal{C}^*$ is upper bounded by $\begin{bmatrix} N-(D-2)/2 \\ N-\ell \end{bmatrix}_q$.

Turning to the dual code $\mathcal{C}^\perp$ of $\mathcal{C}^*$, we have the bound

$$|\mathcal{C}^\perp| \leq \begin{bmatrix} N - (D-2)/2 \\ \ell \end{bmatrix}_q,$$

so by combining these two inequalities, we see that the number of codewords in a subspace code is upper bounded by

$$\begin{bmatrix} N - (D-2)/2 \\ \max\{\ell, N-\ell\} \end{bmatrix}_q.$$

□

We can express the Singleton bound in the normalized parameters R, δ and λ. We suppose $\ell \leq N - \ell$, that is $\frac{\ell}{N} \leq \frac{N-\ell}{N}$. Because $\lambda = \frac{\ell}{N}$, $\lambda \leq \frac{1}{2}$. The case $\lambda \geq \frac{1}{2}$ reduces the the first case by considering dual codes. We then have the following.

Proposition 3.12. *Let $\mathcal{C} \subseteq \mathcal{P}(W, \ell)$, where $\dim(W) = N$, and $\delta = \frac{D}{2\ell}$. The rate R of $\mathcal{C}$ has the upper bound*

$$R \leq (1-\delta)(1-\lambda) + \frac{1}{\lambda N}(1 - \lambda + o(1)),$$

where $o(1)$ is a quantity that goes to zero as $N \to \infty$.

Proof. Because $\mathcal{C}$ contains at most $\begin{bmatrix} N \\ \ell \end{bmatrix}_q$ codewords, and by using the upper bound on the Gaussian coefficient from prop. 2.8, we have

$$\begin{aligned} R &\leq \frac{1}{N\ell} \log_q(|\mathcal{C}|) < \frac{1}{N\ell} \log_q(4q^{(N-\ell)(\ell - \frac{1}{2}(D-2))}) \\ &= \frac{1}{N\ell} \left(\log_q(4) + (N-\ell)\left(\ell - \frac{D-2}{2}\right)\right). \end{aligned}$$

The second term inside the parenthesis can be written as

$$(N-\ell)\left(\ell-\frac{D-2}{2}\right)=N\ell-\ell^2-\frac{DN}{2}+\frac{D\ell}{2}+\frac{\ell}{\lambda}-\ell.$$

We have

$$\begin{aligned}\frac{1}{N\ell}\left(N\ell-\ell^2-\frac{DN}{2}+\frac{D\ell}{2}+\frac{\ell}{\lambda}-\ell\right)&=1-\frac{\ell}{N}-\frac{D}{2\ell}+\frac{D}{2N}+\frac{1}{\lambda N}-\frac{1}{N}\\&=1-\lambda-\delta+\delta\lambda+\frac{1}{\lambda N}-\frac{1}{N}\\&=(1-\delta)(1-\lambda)+\frac{1}{\lambda N}(1-\lambda).\end{aligned}$$

Because $\frac{\log_q(4)}{N\ell}\to 0$, as $N\to\infty$, the result follows. □

3.6 Koetter-Kschischang Codes

In this section, we define the subspace codes as evaluation codes, similar to the way Reed-Solomon codes can be defined. In order to define the ambient space W, we start with the finite field $\mathbb{F}_q$. Let $\mathbb{F}=\mathbb{F}_{q^m}$ be an extension field of $\mathbb{F}_q$. We can regard $\mathbb{F}$ as an m-dimensional vector space over $\mathbb{F}_q$. Let $A=\{\alpha_1,\alpha_2,\ldots,\alpha_\ell\}\subseteq\mathbb{F}$ be a linearly independent set. We denote their span over $\mathbb{F}_q$ as $\langle A\rangle$, i.e. $\langle A\rangle=\mathrm{Span}_{\mathbb{F}_q}\{\alpha_1,\alpha_2,\ldots,\alpha_\ell\}$. Now let

$$W=\langle A\rangle\oplus\mathbb{F}=\{(\alpha,\beta)\mid\alpha\in\langle A\rangle,\beta\in\mathbb{F}\}.$$

Because $\dim_{\mathbb{F}_q}\langle A\rangle=\ell$ and $\dim_{\mathbb{F}_q}\mathbb{F}=m$, the dimension of the ambient space W is $\ell+m$. We want to construct codewords of subspace codes. To do this, we use the linearized polynomials from section 2.4, along with the vector space $\langle A\rangle$. Suppose the sender has k information symbols $u_i\in\mathbb{F}_{q^m}$ that are to be encoded. Let $\mathbf{u}=(u_0,u_1,\ldots,u_{k-1})\in\mathbb{F}_{q^m}^k$ be the information vector and define a linearized polynomial $f(x)$ as

$$f(x)=\sum_{i=0}^{k-1}u_ix^{q^i}=\sum_{i=0}^{k-1}u_ix^{[i]}.$$

The set of all linearized polynomials over $\mathbb{F}_{q^m}$ of degree q^{k-1} or less is denoted $\mathscr{L}_{q^m}^k[X]$. For each basis vector α_j in $\langle A\rangle$, let $\beta_j=f(\alpha_j)$. We have the following lemma.

Lemma 3.13. *If $\{\alpha_1,\alpha_2,\ldots,\alpha_\ell\}$ is a linearly independent set, and $\beta_j=f(\alpha_j)$, where $f(x)$ is a linearized polynomial of degree at most q^{k-1}, then the set*

$$\{(\alpha_1,\beta_1),(\alpha_2,\beta_2),\ldots,(\alpha_\ell,\beta_\ell)\}$$

is linearly independent over $\mathbb{F}_q$.

Proof. Suppose $\gamma_1(\alpha_1, \beta_1) + \cdots + \gamma_\ell(\alpha_\ell, \beta_\ell) = (0, 0)$. We want to show that $\gamma_j = 0$, for $1 \leq \gamma_j \leq \ell$. But because $\gamma_1\alpha_1 + \cdots + \gamma_\ell\alpha_\ell = 0$, and $\{\alpha_1, \alpha_2, \ldots, \alpha_\ell\}$ is a linearly independent set, we see that

$$\{(\alpha_1, \beta_1), (\alpha_2, \beta_2), \ldots, (\alpha_\ell, \beta_\ell)\}$$

is a linearly independent set. □

We also have the following lemma.

Lemma 3.14. *Suppose $\{(\alpha_1, \beta_1), \ldots, (\alpha_r, \beta_r)\} \subseteq W$ is a linearly independent set. If $\beta_j = f(\alpha_j)$ for $1 \leq j \leq r$ and some linearized polynomial f over $\mathbb{F} = \mathbb{F}_{q^m}$, then*

$$\{\alpha_1, \ldots, \alpha_r\}$$

is a linearly independent set.

Proof. Suppose $\gamma_1\alpha_1 + \cdots + \gamma_r\alpha_r = 0$. We want to show that $\gamma_i = 0$, for $1 \leq i \leq r$. We have

$$\sum_{i=1}^{r} \gamma_i(\alpha_i, \beta_i) = \sum_{i=1}^{r} (\gamma_i\alpha_i, \gamma_i\beta_i) = \sum_{i=1}^{r} (0, \gamma_i\beta_i).$$

Furthermore, $\beta_i = f(\alpha_i)$, so

$$\begin{aligned}\sum_{i=1}^{r} (0, \gamma_i\beta_i) = \sum_{i=1}^{r} (0, \gamma_i f(\alpha_i)) &= \sum_{i=1}^{r} (0, f(\gamma_i\alpha_i)) \\ &= (0, f(\sum_{i=1}^{r} \gamma_i\alpha_i)) = (0, f(0)) = (0, 0),\end{aligned}$$

where we used the fact that f is a linear map. We conclude that $\{\alpha_1, \ldots, \alpha_r\}$ is a linearly independent set. □

For constructing subspace codes, the following mapping is used.

Definition 3.15 (Evaluation Map)**.** Let $\mathbf{u} = (u_0, u_1, \ldots, u_{k-1}) \in \mathbb{F}_{q^m}^k$ be a message vector of length k. Define a linearized polynomial $f(x)$ as

$$f(x) = \sum_{i=0}^{k-1} u_i x^{q^i} = \sum_{i=0}^{k-1} u_i x^{[i]}.$$

For each basis vector α_j in $\langle A \rangle = \langle \alpha_1, \ldots, \alpha_\ell \rangle$, let $\beta_j = f(\alpha_j)$. Define a map $\mathsf{ev}_A : \mathscr{L}_{q^m}^k[X] \to \mathcal{P}(W, |A|)$ by

$$\mathsf{ev}_A(f) = \mathrm{Span}_{\mathbb{F}_q}\{(\alpha_1, f(\alpha_1)), \ldots, (\alpha_\ell, f(\alpha_\ell))\} \quad = \langle (\alpha_1, f(\alpha_1)), \ldots, (\alpha_\ell, f(\alpha_\ell)) \rangle .$$

This map is called evaluation of f at A.

The next lemma shows that if the dimension of $\langle A \rangle$ is sufficiently large, then a message polynomial is mapped to a unique subspace, using the ev_A mapping.

Lemma 3.16. *Consider the map ev_A defined in def. 3.15. If $\dim(A) = |A| \geq k$, then $\mathsf{ev}_A : \mathscr{L}_{q^m}^k[X] \to \mathcal{P}(W, |A|)$ is injective.*

Proof. Let $f(x), g(x) \in \mathscr{L}_q^k[X]$ satisfy $\mathsf{ev}_A(f) = \mathsf{ev}_A(g)$. Let $h(x) = f(x) - g(x)$. Then $h(\alpha_i) = f(\alpha_i) - g(\alpha_i) = 0$, for all $\alpha_i \in A$. But since h is a linearized polynomial, $h(\alpha) = 0$ for all $\alpha \in \langle A \rangle$. Therefore, $h(x)$ has $q^{|A|}$ zeros. However, the degree of $h(x)$ is at most q^{k-1}, so $h(x) \equiv 0$. Thus $f(x) \equiv g(x)$, which shows that ev_A is injective . □

Using the map ev_A, we show that the image of $\mathscr{L}_{q^m}^k[X]$ is a constant-dimensional subspace code. This code is called a KK-code[2].

Theorem 3.17. *Let $\mathsf{ev}_A : \mathscr{L}_{q^m}^k[X] \to \mathcal{P}(W, \ell)$, where $\ell = \dim(A)$. The image of $\mathscr{L}_{q^m}^k[X]$ under ev_A is a subspace code with parameters*

$$[\ell + m, \ell, mk, 2(\ell - k + 1)]\,.$$

Proof. Because the map ev_A sends linearized polynomials to subspaces, the image of $\mathscr{L}_{q^m}^k[X]$ under ev_A is a set of subspaces. We now verify the four parameters. Because the ambient space is W, and $W = \langle A \rangle \oplus \mathbb{F}$,

$$\dim(W) = \dim(\langle A \rangle \oplus \mathbb{F}) = \dim(\langle A \rangle) + \dim(\mathbb{F}) = \ell + m.$$

From lemma 3.13, each codeword has dimension ℓ, so the code is constant dimensional. Regarding the number of codewords, because the information vector contains k symbols from $\mathbb{F}_{q^m}$, there are $(q^m)^k = q^{mk}$ possible linearized polynomials which can be defined from these vectors. Because ev_A is injective (lemma 3.16), there are q^{mk} codewords, so the parameter mk follows. We now turn to the minimum distance. Let $f(x), g(x) \in \mathscr{L}_{q^m}^k[X]$, with $f(x) \neq g(x)$. From the injectivity of ev_A, we have $\mathsf{ev}_A(f) \neq \mathsf{ev}_A(g)$, so the codewords $U = \mathsf{ev}_A(f)$ and $V = \mathsf{ev}_A(g)$ are distinct. We will now find a lower bound on the distance between U and V, $d(U, V)$. We have

$$\begin{aligned} d(U, V) &= \dim(U) + \dim(V) - 2\dim(U \cap V) \\ &= \ell + \ell - 2\dim(U \cap V) \\ &= 2(\ell - \dim(U \cap V)). \end{aligned}$$

Suppose $\dim(U \cap V) = r$, so $(U \cap V) = \langle \beta_1', \ldots, \beta_r' \rangle$. We then have $f(a) - g(a) = 0$ for all $a \in \langle \beta_1', \ldots, \beta_r' \rangle$, where $f(x)$ and $g(x)$ each have degree at

[2] KK is short for Koetter-Kschischang

most q^{k-1}. Suppose $r \geq k$. Then both $f(x)$ and $g(x)$ have $q^r \geq q^k$ zeros, which implies that $f(x) - g(x) \equiv 0$, or $f(x) \equiv g(x)$, which is a contradiction to the assumption that $f(x)$ and $g(x)$ are distinct. So we must have $r \leq k-1$. Hence $\dim(U \cap V) \leq k-1$, and we get the lower bound

$$d(U,V) = 2(\ell - \dim(U \cap V)) \geq 2(\ell - k + 1).$$

□

Example 3.18. In this example, we show that the lower bound on the minimum distance, shown in the proof to the above theorem, is attainable. We do this by constructing two different linearized polynomials of degree at most q^{k-1}, which both have $k-1$ roots among the elements in $A = \{\alpha_1, \ldots, \alpha_\ell\}$. Consider the set $A' = \{\alpha_1, \ldots, \alpha_{k-1}\} \subset A$. This is a linearly independent subset of A. Then we use prop. 2.20 to form a monic linearized polynomial $L(x)$ over $\mathbb{F}_{q^m}$ of degree q^{k-1}. The polynomial $L(x)$ has the $k-1$ elements in A' as roots. Let $f, g \in \mathscr{L}^k_{q^m}[X]$, where

$$f(x) = f_0 x, \;\; g(x) = g_0 x,$$

and $f_0 \neq g_0$ and both nonzero. Because $f(x)$, $g(x)$ and $L(x)$ are linearized polynomials over $\mathbb{F}_{q^m}$, so are $f(L(x))$ and $g(L(x))$. Also, because the degree of $f(x)$ and $g(x)$ is $q^0 = 1$, the two polynomials $f(L(x))$ and $g(L(x))$ have degree q^{k-1}. Therefore, they are valid message polynomials. From the polynomials $f(L(x))$ and $g(L(x))$, we construct the two codewords $U = \mathsf{ev}_A(f(L(x)))$ and $V = \mathsf{ev}_A(g(L(x)))$ which are of the form

$$\begin{aligned}
\mathsf{ev}_A(f(L(x))) &= \langle (\alpha_1, 0), \ldots, (\alpha_{k-1}, 0), (\alpha_k, f(L(\alpha_k))), \ldots, (\alpha_n, f(L(\alpha_n))) \rangle \\
\mathsf{ev}_A(g(L(x))) &= \langle (\alpha_1, 0), \ldots, (\alpha_{k-1}, 0), (\alpha_k, g(L(\alpha_k))), \ldots, (\alpha_n, g(L(\alpha_n))) \rangle .
\end{aligned}$$

Because the ev_A mapping is injective, these two codewords are different. From the above, we see that $U \cap V = \langle (\alpha_1, 0), \ldots, (\alpha_{k-1}, 0) \rangle$. Therefore, the distance between these two codewords is

$$\begin{aligned}
d(U,V) &= \dim(U) + \dim(V) - 2\dim(U \cap V) \\
&= \ell + \ell - 2(k-1) \\
&= 2\ell - 2(k-1) = 2(\ell - (k-1))
\end{aligned}$$

The last result in this section shows that the KK-codes are nearly Singleton bound achieveing.

Proposition 3.19. *Let $\mathcal{C}$ be a subspace code with parameters*

$$[\ell + m, \ell, mk, 2(\ell - k + 1)] .$$

The rate R of $\mathcal{C}$ attains the upper bound in prop. 3.12 as $\ell + m \to \infty$.

Proof. From prop. 3.12, the upper bound on the rate of a subspace code is

$$R < \frac{1}{N\ell}\left(\log_q(4) + (N-\ell)\left(\ell - \frac{D-2}{2}\right)\right).$$

In our case, the code rate is $R = \frac{\log_q(|\mathcal{C}|)}{N\ell} = \frac{mk}{\ell(\ell+m)}$, which we denote R_{KK}. Substituting the parameters for $\mathcal{C}$, the above inequality becomes

$$\begin{aligned} R_{KK} &< \frac{1}{\ell(\ell+m)}\left(\log_q(4) + (\ell+m-\ell)\left(\ell - \frac{2(\ell-k+1)-2}{2}\right)\right) \\ &= \frac{1}{\ell(\ell+m)}\left(\log_q(4) + m(\ell-(\ell-k+1-1))\right) \\ &= \frac{1}{\ell(\ell+m)}\left(\log_q(4) + mk\right) \\ &= \frac{mk}{\ell(\ell+m)} + \frac{\log_q(4)}{\ell(\ell+m)} \\ &= R_{KK} + \frac{\log_q(4)}{\ell(\ell+m)}. \end{aligned}$$

Because $\frac{\log_q(4)}{\ell(\ell+m)} \to 0$, as $\ell + m \to \infty$, the result follows. □

3.7 Decoding KK-codes

We now describe the situation of decoding subspace codes. We suppose that the code $\mathcal{C}$ used has parameters $[\ell+m, \ell, mk, 2(\ell-k+1)]$. Also, we assume that the channel does ρ erasures and t errors. The transmitter wants to send a message, which is encoded using a codeword $V \in \mathcal{C}$, where $\dim(V) = \ell$. Suppose the receiver gets U, which is some subspace of W, and that $\dim(U \cap V) = \ell - \rho$. Because there are ρ erasures and t errors, $\dim(U) = \ell - \rho + t$. The distance between U and V is

$$\begin{aligned} d(U,V) &= \ell + \ell - \rho + t - 2(\ell - \rho) \\ &= 2\ell - \rho + t - 2\ell + 2\rho = \rho + t. \end{aligned}$$

At the receiver, minimum distance decoding is successful if and only if $2(t+\rho) < D$, where D is the minimum distance of the code (prop. 3.7). In our case, $D = 2(\ell - k + 1)$. Therefore, the receiver can decode if and only if

$$d(U,V) = \rho + t < \frac{D}{2} = \frac{2(\ell-k+1)}{2} = \ell - k + 1.$$

Let $r = \ell - \rho + t$, and suppose that the received subspace U is spanned by $\{(x_1, y_1), \ldots, (x_r, y_r)\}$, i.e.

$$U = \langle (x_1, y_1), \ldots, (x_r, y_r) \rangle .$$

Let $Q_x(x), Q_y(x) \in \mathscr{L}_{q^m}[X]$, where $\deg(Q_x(x)) = q^{\tau-1}$ and $\deg(Q_y(y)) = q^{\tau-k}$. In the next section, we describe an algorithm which, given a linearly independent set $\{(x_1, y_1), \ldots, (x_r, y_r)\}$ produces a bivariate polynomial $Q(x, y)$, with the property that

$$Q(x_i, y_i) = 0, \text{ for all } i \text{ , where } Q(x, y) = Q_x(x) + Q_y(y).$$

Suppose we have such a polynomial $Q(x, y)$. The condition $Q(x_i, y_i) = 0$, for $1 \leq i \leq r$ means that we have r homogeneous equations in $\tau - 1 + 1 + \tau - k + 1 = 2\tau - k + 1$ unknowns. If $r < 2\tau - k + 1$, then the system has a non-trivial solution.

Because $\mathcal{C}$ is a subspace code defined via an evaluation map ev_A, each codeword in $\mathcal{C}$ is of the form $\mathsf{ev}_A(f)$, where $f \in \mathscr{L}^k_{q^m}[X]$. So f is a linearized polynomial of degree at most q^{k-1}, and $Q(x, y)$ is a bivariate linearized polynomial (see def. 2.16). Therefore, $Q(x, f(x))$ is a linearized polynomial, and

$$Q(x, f(x)) = Q_x(x) + Q_y(f(x)) = Q_x(x) + Q_y(x) \otimes f(x).$$

The degree of $Q(x, f(x))$ is upper bounded by $q^{\tau-1}$. Now let $U \cap V = \langle (a_1, b_1), \ldots, (a_{\ell-\rho}, b_{\ell-\rho}) \rangle$. Then, because $U \cap V \subseteq U$, we have $Q(a_i, b_i) = 0$ for $1 \leq i \leq \ell - \rho$. But $U \cap V \subseteq V$, where V is the codeword sent, so $b_i = f(a_i)$. Therefore,

$$Q(a_i, b_i) = Q(a_i, f(a_i)) = 0,$$

and so $a_1, \ldots, a_{\ell-\rho}$ are roots of $Q(x, f(x))$. These roots are linearly independent, so they span a vector space of dimension $\ell - \rho$ over $\mathbb{F}_q$. This vector space contains $q^{\ell-\rho}$ vectors, which are roots in $Q(x, f(x))$. But $Q(x, f(x))$ has degree at most $q^{\tau-1}$, so if $q^{\ell-\rho} \geq q^{\tau-1}$, then $\ell - \rho \geq \tau - 1$. This implies that $Q(x, f(x))$ has more roots than its degree, which is only possible if $Q(x, f(x))$ is the zero polynomial.

Now, $Q(x, y) = Q_x(x) + Q_y(y)$, so $Q(x, f(x)) = Q_x(x) + Q_y(f(x))$. If $Q(x, f(x)) = 0$, then the polynomial $f(x)$ is a y-root of $Q(x, y)$. The goal is to find the y-root $f(x)$ which corresponds to the message polynomial.

Definition 3.20 (Decodable)**.** Let $Q(x, y) = Q_x(x) + Q_y(y)$, where $Q_x(x)$ and $Q_y(y)$ are linearized polynomials with degrees $\deg(Q_x(x)) = q^{\tau-1}$ and $\deg(Q_y(y)) = q^{\tau-k}$. Suppose the sender uses a subspace code $\mathcal{C}$ with parameters $[\ell + m, \ell, mk, 2(\ell - k + 1)]$. Also, assume that the operator channel imposes ρ erasures and t errors. We call a received space U decodable if and only if conditions 1 and 2 below are satisfied,

1. $r = \ell - \rho + t < 2\tau - k + 1$,

2. $\ell - \rho \geq \tau$,

where $\dim(U) = \ell - \rho$ and $\dim(U \cap V) = r$.

Condition 1 in def. 3.20 is the requirement of the interpolation step, that is, the existence of a non-zero interpolation polynomial. Condition 2 comes from the step of finding the message polynomial, also called the factorization step.

Proposition 3.21. *A received space U is decodable if and only if $\rho + t < \ell - k + 1$.*

Proof. Suppose U is decodable. Then from the two conditions of decodability in def. 3.20, we have

$$\begin{aligned} \ell - \rho + t &< 2\tau - k + 1 \\ &\leq 2(\ell - \rho) - k + 1, \end{aligned}$$

which implies $\rho + t < \ell - k + 1$.

Conversely, assume that $\rho + t < \ell - k + 1$. Then $t + k \leq \ell - \rho$. Adding $\ell - \rho$ to both sides of the inequality, we get

$$\ell - \rho + t + k = r + k \leq 2(\ell - \rho).$$

Since the receiver knows r, which is the dimension of the received space U, and the number of information symbols k, he can select $\tau = \left\lceil \frac{r+k}{2} \right\rceil$. This gives us $\frac{r+k}{2} \leq \tau$, and multiplying both sides by 2, we get the inequality $r + k \leq 2\tau < 2\tau + 1$, which can be rearranged to $r < 2\tau - k + 1$. Thus condition 1 in def. 3.20 is satisfied.

Because r and k are integers, so is $r + k$. If $r + k$ is even, then $\frac{r+k}{2}$ is an integer, and $\tau = \frac{r+k}{2}$ in this case. If $r + k$ is odd, then $\frac{r+k}{2} + \frac{1}{2}$ is an integer, and then $\tau = \frac{r+k}{2} + \frac{1}{2}$. Therefore, $\tau \leq \frac{r+k}{2} + \frac{1}{2}$, and we see that

$$\tau \leq \frac{2(\ell - \rho)}{2} + \frac{1}{2} = \ell - \rho + \frac{1}{2}.$$

But τ, ℓ and ρ are integers, so we get $\tau \leq \ell - \rho$. Therefore, condition 2 in def. 3.20 is satisfied. □

3.7.1 Division Algorithm for Linearized Polynomials

In this section, we present a division algorithm for linearized polynomials, due to [KK08]. This algorithm is used for finding the linearized polynomial corresponding to the sent message. Because the binary operation $\otimes$ is not commutative in $\mathscr{L}_{q^m}[X]$, there are two division algorithms, **RDiv** and **LDiv** for right and left division respectively. For the **LDiv** algorithm, all lines are identical to **RDiv**, except that

$$t(x) := \left(\frac{a_d}{b_e}\right)^{[m-e]} x^{[d-e]} \text{ is replaced by } t(x) := \frac{a_d}{(b_e)^{[d-e]}} x^{[d-e]} \text{ and}$$
$$a(x) - b(x) \otimes t(x) \text{ is replaced by } a(x) - t(x) \otimes b(x).$$

Also, $q_R(x)$ is replaced by $q_L(x)$, while $r_R(x)$ is replaced by $r_L(x)$. The

Algorithm 1 Division algorithm for linearized polynomials

RDiv$(a(x), b(x))$
Input: Two linearized polynomials $a(x), b(x)$, where $b(x) \not\equiv 0$
Output: A tuple $(q_R(x), r_R(x))$, where $q_R(x), r_R(x) \in \mathscr{L}_{q^m}[X]$, satisfying $a(x) = b(x) \otimes q_R(x) + r_R(x)$,
where either $r_R(x) \equiv 0$ or $\deg(r_R(x)) < \deg(b(x))$.
if $\deg(a(x)) < \deg(b(x))$ **then**
    return $(0, a(x))$.
else
    $d \leftarrow \deg(a(x))$.
    $e \leftarrow \deg(b(x))$.
    $a_d \leftarrow \mathrm{lc}(a(x)$.
    $b_e \leftarrow \mathrm{lc}(b(x)$.
    $t(x) \leftarrow \left(\frac{a_d}{b_e}\right)^{[m-e]} x^{[d-e]}$
    return $(t(x), 0)$+**RDiv**$(a(x) - b(x) \otimes t(x), b(x))$.
end if

algorithm is shown in alg. 1, and correctness is shown in prop. 3.22.

Proposition 3.22. *Given linearized polynomials $a(x)$ and $b(x)$, algorithm 1 returns linearized polynomials $q_R(x)$ and $r_R(x)$ satisfying*

$$a(x) = b(x) \otimes q_R(x) + r_R(x), \tag{3.7}$$

where $r_R(x) \equiv 0$ or $\deg(r_R(x)) < \deg(b(x))$.

Proof. The proof is by induction on $n = \deg(a(x)) - \deg(b(x))$. For the base case, if $n < 0$, then $\deg(a(x)) < \deg(b(x))$, and then we are in the if block. In this case, the algorithm returns $q_R(x) \equiv 0$ and $r_R(x) = a(x)$. For the

induction hypothesis, assume that when $n = \deg(a(x)) - \deg(b(x)) > 0$, then the algorithm returns $q_R(x)$ and $r_R(x)$ satisfying eq. (3.7), where $r_R(x) \equiv 0$ or $\deg(r_R(x)) < \deg(b(x))$. We must show that this implies the correctness of the algorithm for $n+1$. If the difference in degrees is $n+1$, then we are in the else block. Herein, the **RDiv** procedure is called with the arguments $a(x) - b(x) \otimes t(x)$ and $b(x)$. Now, the term with degree $[d]$ in $b(x) \otimes t(x)$ equals

$$b_e \left(\frac{a_d}{b_e}\right)^{[m]} x^{[d]},$$

and the term with degree $[d]$ in $a(x) - b(x) \otimes t(x)$ equals

$$\left(a_d - b_e \left(\frac{a_d}{b_e}\right)^{[m]}\right) x^{[d]} = \left(a_d - b_e \frac{a_d}{b_e}\right) x^{[d]} = 0,$$

since $\alpha^{[m]} = \alpha^{q^m} = \alpha = \alpha^{[0]}$ for $\alpha \in \mathbb{F}_{q^m}$. Because the leading terms of $a(x)$ and $b(x) \otimes t(x)$ cancel, the difference in degree between $a(x)$ and $a(x) - b(x) \otimes t(x)$ is n. Using the induction hypotheses, the **RDiv** procedure with arguments $a(x) - b(x) \otimes t(x)$ and $b(x)$ correcly returns $q_R(x)$ and $r_R(x)$ with the desired properties. By induction, the algorithm is correct for all n. □

Example 3.23. Here we give an example of the **RDiv** and **LDiv** algorithms. Let

$$f(x) = a_0 x^{[0]} + a_1 x^{[1]}, \text{ and } g(x) = b_0 x^{[0]}. \tag{3.8}$$

We first run the **RDiv** algorithm with inputs $f(x)$ and $g(x)$. If the first iteration, we are in the else block, and compute

$$t(x) = \left(\frac{a_1}{b_0}\right)^{[m-0]} x^{[1]} = \frac{a_1}{b_0} x^{[1]}.$$

Then, $a(x) - b(x) \otimes t(x)$ is equal to

$$\begin{aligned} &a_0 x^{[0]} + a_1 x^{[1]} - b_0 \left(\frac{a_1}{b_0} x^{[1]}\right)^{[0]} \\ &= a_0 x^{[0]} + \left(a_1 - b_0 \frac{a_1}{b_0}\right) x^{[1]} = a_0 x^{[0]}. \end{aligned}$$

The algorithm calls itself with parameters $a_0 x^{[0]}$ and $b_0 x^{[0]}$. Again, we are in the else block, and

$$t(x) = \left(\frac{a_0}{b_0}\right)^{[m-0]} x^{[0]} = \frac{a_0}{b_0} x^{[0]}.$$

Then, $a(x) - b(x) \otimes t(x) = a_0 x^{[0]} - b_0 \left(\frac{a_0}{b_0} x^{[0]}\right) = a_0 x^{[0]} - a_0 x^{[0]} = 0$. When the algorithm calls itself with parameters $0, b_0 x^{[0]}$, it goes to the if-block, and returns $q_R(x) \equiv 0$ and $r_R(x) = 0$.

We therefore have $q_R(x) = \frac{a_1}{b_0}x^{[1]} + \frac{a_0}{b_0}x^{[0]}$, and the product $g(x) \otimes q_R(x)$ equals $c_0x^{[0]} + c_1x^{[1]}$, where

$$c_0 = b_0 \left(\frac{a_0}{b_0}\right)^{[0]} = a_0$$
$$c_1 = b_0 \left(\frac{a_1}{b_0}\right)^{[0]} = a_1.$$

Therefore, $f(x) = g(x) \otimes q_R(x)$.

We now use the **LDiv** algorithm with the arguments $f(x)$ and $g(x)$ in eq. (3.8). In the first iteration, we are in the else block, and

$$t(x) = \frac{a_1}{b_0^{[1-0]}}x^{[1-0]} = \frac{a_1}{b_0^{[1]}}x^{[1]}.$$

In this case, $a(x) - t(x) \otimes b(x)$ equals

$$a_0x^{[0]} + a_1x^{[1]} - \frac{a_1}{b_0^{[1]}}\left(b_0x^{[0]}\right)^{[1]}$$
$$= a_0x^{[0]} + a_1x^{[1]} - \frac{a_1b_0^{[1]}}{b_0^{[1]}}x^{[1]} = a_0x^{[0]}.$$

Now the **LDiv** algorithm calls itself with arguments $a_0x^{[0]}$ and $b_0x^{[0]}$. Again, we are in the else block, and

$$t(x) = \frac{a_0}{b_0^{[0]}}x^{[0]}.$$

Then $a(x) - t(x) \otimes b(x)$ equals

$$a_0x^{[0]} - \frac{a_0}{b_0^{[0]}}\left(b_0x^{[0]}\right)^{[0]} = a_0x^{[0]} - a_0x^{[0]} = 0.$$

When the **LDiv** algorithm is called with parameters 0 and $b_0x^{[0]}$, it goes to the if block and returns $q_L(x) \equiv 0$ and $r_L(x) \equiv 0$. The quotient is $q_L(x) = \frac{a_1}{b_0^{[1]}}x^{[1]} + \frac{a_0}{b_0^{[0]}}x^{[0]}$. Then $q_L(x) \otimes g(x)$ equals

$$\left(\frac{a_1}{b_0^{[1]}}\left(b_0x^{[0]}\right)^{[1]} + \frac{a_0}{b_0^{[0]}}\left(b_0x^{[0]}\right)^{[0]}\right) = a_1x^{[1]} + a_0x^{[0]} = f(x).$$

In the next example, we use the field $\mathbb{F}_4$ and the results of the previous example.

Example 3.24. Consider the previous example, with $q = 2$, $m = 2$. We therefore consider the field $\mathbb{F}_4 = \{0, 1, \omega, \omega + 1\}$. Let

$$f(x) = x^2, \text{ and } g(x) = \omega x.$$

Using the results of the previous example, with $a_0 = 0$, $a_1 = 1$ and $b_0 = \omega$, we get

$$\begin{aligned} q_L(x) &= \frac{1}{\omega^2}x^2 = (\omega^2)^{-1}x^2 = (\omega+1)^{-1}x^2 = \omega x^2, \text{ and} \\ q_R(x) &= \frac{1}{\omega}x^2 = \omega^{-1}x^2 = (\omega+1)x^2. \end{aligned}$$

Clearly, $q_L(x) \neq q_R(x)$. However, we have

$$\begin{aligned} q_L(x) \otimes g(x) &= (\omega x^2) \otimes (\omega x) \\ &= \omega(\omega x)^2 = \omega\omega^2 x^2 \\ &= \omega(\omega+1)x^2 = \omega\omega^{-1}x^2 = x^2 = f(x), \end{aligned}$$

while

$$\begin{aligned} g(x) \otimes q_R(x) &= (\omega x) \otimes ((\omega+1)x^2) \\ &= \omega((\omega+1)x^2) = \omega(\omega+1)x^2 \\ &= \omega\omega^{-1}x^2 = x^2 = f(x). \end{aligned}$$

3.7.2 Interpolation Algorithm for Decoding KK-codes

In this section, we present an algorithm, due to [KK08], which can be used in the decoding of subspace codes. The algorihm finds an interpolation polynomial from a set of r linearly independent points. We start by defining the degree on the ring of bivariate linarized polynomials, which is used in proving the correctness of the algorithm. The algorithm, which is called **Interpolate**, can be used in conjunction with the **RDiv** algorithm presented earlier to decode KK-codes.

Let $f_x(x)$ and $f_y(y)$ be univariate linearized polynomials, with $\deg(f_x(x)) = q^{d_x(f)} = [d_x(f)]$ and $\deg(f_y(y)) = q^{d_y(g)} = [d_y(g)]$. The polynomial $f(x, y) = f_x(x) + f_y(y)$ is a bivariate linearized polynomial, i.e. it is of the form

$$f(x, y) = \sum_{i=0}^{d_x(f)} a_i x^{[i]} + \sum_{j=0}^{d_y(f)} b_j y^{[j]}.$$

Definition 3.25 ($(1, k-1)$-weighted degree)**.** Let $f(x, y) = f_x(x) + f_y(y)$, with $f_x(x), f_y(y)$ being linearized polynomials with degree as above. We define the $(1, k-1)$-weighted degree of $f(x, y)$ as

$$\deg_{1,k-1}(f(x, y)) = \max\{d_x(f), k - 1 + d_y(f)\}.$$

Suppose that the receiver has received a space $U \in \mathcal{P}(W)$, where $U = \langle(x_1, y_1), \ldots, (x_r, y_r)\rangle$. From the elements (x_i, y_i), we want to find a bivariate linearized polynomial $Q(x, y)$ which interpolates the basis elements of U, that is, $Q(x_i, y_i) = 0$, for $1 \leq i \leq r$. The interpolation algorithm is given in alg. 2.

We now explain the algorithm, and then prove its correctness. The algorithm is given a basis $\{(x_1, y_1), \ldots, (x_r, y_r)\}$ for a received subspace U. It first initializes the polynomials $f_0(x, y)$ and $f_1(x, y)$, and then iterates through the elements in the basis for U. At the beginning of each iteration, the algorithm evaluates the polynomials $f_0(x, y)$ and $f_1(x, y)$ in the current basis element for U. These evaluations are denoted Δ_0 and Δ_1 for $f_0(x, y)$ and $f_1(x, y)$ respectively. If $\Delta_0 = 0$ and $\Delta_1 \neq 0$, the algorithm updates $f_1(x, y)$ such that

$$f_1'(x, y) \leftarrow f_1^q(x, y) - \Delta_1^{q-1} f_1(x, y).$$

Note that $f_1'(x, y)$ has the current basis element for U as a root, and that the degree of $f_1(x, y)$ increases by q. In the current case, $f_0(x, y)$ is unchanged. If $\Delta_0 \neq 0$ and $\Delta_1 = 0$, then the algorithm does the same, but with the roles of $f_0(x, y)$ and $f_1(x, y)$ reversed. For the case $\Delta_0 = 0$ and $\Delta_1 = 0$, no updating is necessary.

The remaining case is $\Delta_0 \neq 0$ and $\Delta_1 \neq 0$, where both polynomials need to be updated. The algorithm then compares the $(1, k-1)$-degrees of $f_0(x, y)$ and $f_1(x, y)$. If $\deg_{1,k-1}(f_0(x, y)) \leq \deg_{1,k-1}(f_1(x, y))$, then it first sets $f_1'(x, y) := \Delta_1 f_0(x, y) - \Delta_0 f_1(x, y)$, and then $f_1'(x, y)$ has the current basis element as a root. Note that the degree of $f_1(x, y)$ does not increase. Furthermore, the assignment

$$f_0'(x, y) \leftarrow f_0^q(x, y) - \Delta_0^{q-1} f_0(x, y),$$

results in the polynomial $f_0'(x, y)$ having the current basis element as root, and the degree increases by q. For the case

$$\deg_{1,k-1}(f_0(x, y)) \geq \deg_{1,k-1}(f_1(x, y)),$$

the roles of $f_0(x, y)$ and $f_1(x, y)$ are reversed.

Finally, the algorithm returns either $f_0(x, y)$ or $f_1(x, y)$, depending on which one has smaller $(1, k-1)$-degree.

Algorithm 2 Interpolation of basis for subspace

Interpolate(U)
Input: Subspace U of W, where $U = \langle (x_1, y_1), \ldots, (x_r, y_r) \rangle$.
Output: Linearized bivariate polynomial $Q(x, y) = Q_x(x) + Q_y(y)$, satisfying $Q(x_i, y_i) = 0$, for $1 \leq i \leq r$.
Initialization: $f_0(x, y) \leftarrow x, f_1(x, y) \leftarrow y$.
for $i = 1 \rightarrow r$ **do**
    $\Delta_0 \leftarrow f_0(x_i, y_i)$
    $\Delta_1 \leftarrow f_1(x_i, y_i)$
    if $\Delta_0 = 0$ **then**
        $f_1(x, y) \leftarrow f_1^q(x, y) - \Delta_1^{q-1} f_1(x, y)$
    else if $\Delta_1 = 0$ **then**
        $f_0(x, y) \leftarrow f_0^q(x, y) - \Delta_0^{q-1} f_0(x, y)$
    else
        if $\deg_{1,k-1}(f_0) \leq \deg_{1,k-1}(f_1)$ **then**
            $f_1(x, y) \leftarrow \Delta_1 f_0(x, y) - \Delta_0 f_1(x, y)$
            $f_0(x, y) \leftarrow f_0^q(x, y) - \Delta_0^{q-1} f_0(x, y)$
        else
            $f_0(x, y) \leftarrow \Delta_1 f_0(x, y) - \Delta_0 f_1(x, y)$
            $f_1(x, y) \leftarrow f_1^q(x, y) - \Delta_1^{q-1} f_1(x, y)$
        end if
    end if
end for
if $\deg_{1,k-1}(f_1) < \deg_{1,k-1}(f_0)$ **then**
    return $f_1(x, y)$
else
    return $f_0(x, y)$
end if

In order to prove the correctness of algorithm 2, we define a order $\prec$ on the ring of bivariate linearized polynomials as follows. Let $f(x, y)$ and $g(x, y)$ be bivariate linearized polynomials.We say that $f(x, y) \prec g(x, y)$ if

$$\deg_{1,k-1}(f(x, y)) < \deg_{1,k-1}(g(x, y)).$$

On the other hand, if

$$\deg_{1,k-1}(f(x, y)) = \deg_{1,k-1}(g(x, y)),$$

then $f(x, y) \prec g(x, y)$ if both

$$d_y(f) + k - 1 < \deg_{1,k-1}(f(x, y)) \text{ and}$$
$$d_y(g) + k - 1 = \deg_{1,k-1}(g(x, y))$$

hold. If none of these hold, we say that $f(x,y)$ and $g(x,y)$ are not comparable.

Note that $\prec$ is not a total order on monomials of the form $x^{[i]}y^{[j]}$. For example, if

$$m_1(x,y) = x^{[2]}y^{[5]}, \text{ and } m_2(x,y) = x^{[3]}y^{[5]},$$

and $k = 4$, then

$$\begin{aligned}\deg_{1,k-1}(m_1(x,y)) &= \max\{2, 4-1+5\} = 8\\ \deg_{1,k-1}(m_2(x,y)) &= \max\{3, 4-1+5\} = 8,\end{aligned}$$

and so $\deg_{1,k-1}(m_1(x,y)) = \deg_{1,k-1}(m_2(x,y))$. However, for this choice of monomials, the condition

$$d_y(f) + k - 1 < \deg_{1,k-1}(m_1(x,y))$$

is not satisfied because $5 + 4 - 1 = 8 < 8$ is false. Therefore, $m_1(x,y)$ and $m_2(x,y)$ are incomparable under $\prec$. However, if we consider monomials of the form $x^{[i]}$ and $y^{[j]}$, and use the convention that the degree of the zero polynomial is $-\infty$, then $\prec$ gives a total order on these monomials. We show this in an example.

Example 3.26. In this example, we compare monomials of the form $x^{[i]}$ and $y^{[j]}$, and show that any pair of such monomials can be compared under the order $\prec$. Consider the monomials $x^{[i]}$ and $x^{[j]}$. Then

$$\begin{aligned}\deg_{1,k-1}(x^{[i]}) &= \max\{i, -\infty\} = i,\\ \deg_{1,k-1}(x^{[j]}) &= \max\{j, -\infty\} = j,\end{aligned}$$

so in this case, we can compare $x^{[i]}$ and $x^{[j]}$ under $\prec$: If $i < j$, then $x^{[i]} \prec x^{[j]}$, and if $i > j$, then $x^{[i]} \succ x^{[j]}$. For the case $i = j$, the monomials $x^{[i]}$ and $x^{[j]}$ are the same monomial.

For the monomials $y^{[i]}$ and $y^{[j]}$, we have

$$\begin{aligned}\deg_{1,k-1}(y^{[i]}) &= \max\{-\infty, k-1+i\} = k-1+i,\\ \deg_{1,k-1}(y^{[j]}) &= \max\{-\infty, k-1+j\} = k-1+j.\end{aligned}$$

In this case, if $i < j$, then $y^{[i]} \prec y^{[j]}$, and if $i > j$, then $y^{[i]} \succ y^{[j]}$. For the case $i = j$, the monomials $y^{[i]}$ and $y^{[j]}$ are the same monomial.

Now we consider $x^{[i]}$ and $y^{[j]}$. The $(1, k-1)$-degrees are

$$\begin{aligned}\deg_{1,k-1}(x^{[i]}) &= \max\{i, k-1-\infty\} = i,\\ \deg_{1,k-1}(y^{[j]}) &= \max\{-\infty, k-1+j\} = k-1+j.\end{aligned}$$

Here, if $i < k-1+j$, then $x^{[i]} \prec y^{[j]}$, and $i > k-1+j$ implies that $x^{[i]} \succ y^{[j]}$. Now, if $i = k-1+j$, then two monomials have the same $(1, k-1)$-degree, and so we check wheter the two conditions for this subcase are satisfied. We have

$$\begin{aligned} d_y(x^{[i]}) + k - 1 &= -\infty + k - 1 < k - 1 = \deg_{1,k-1}(x^{[i]}), \text{ and} \\ d_y(y^{[j]}) + k - 1 &= j + k - 1 = k - 1 + j = \deg_{1,k-1}(y^{[j]}). \end{aligned}$$

Because both of the above conditions are true, $x^{[i]} \prec y^{[j]}$ in this case.

As the following example shows, even when two monomials have the same $(1, k-1)$-degree, with one monomial being a power of x, while the other being a power of y, we can still compare them using $\prec$.

Example 3.27. Let $m_1(x, y) = x^{[k-1]}$, while $m_2(x, y) = y^{[0]}$. Then

$$\begin{aligned} \deg_{1,k-1}(m_1(x, y)) &= \max\{k - 1, -\infty\} = k - 1 \\ \deg_{1,k-1}(m_2(x, y)) &= \max\{-\infty, k - 1 + 0\} = k - 1, \end{aligned}$$

and so $\deg_{1,k-1}(m_1(x, y)) = \deg_{1,k-1}(m_2(x, y))$. We therefore proceed to check if the two conditions required in this case are satisfied. We have

$$\begin{aligned} d_y(m_1(x, y)) + k - 1 &= -\infty + k - 1 < k - 1, \text{ and} \\ d_y(m_2(x, y)) + k - 1 &= 0 + k - 1 = k - 1, \end{aligned}$$

both of which are true. So in this case, $m_1(x, y) \prec m_2(x, y)$, that is, $x^{[k-1]} \prec y^{[0]}$.

We note that we can also write the ordering on the monomials as follows. For $i, k \geq 0$

$$\begin{aligned} x^{[i]} \prec x^{[i+1]}, \;\; y^{[i]} \prec y^{[i+1]}, \\ x^{[i+k-1]} \prec y^{[i]} \prec x^{[i+k]}, \end{aligned}$$

where the last comparison follows because

$$\begin{aligned} \deg_{1,k-1}(x^{[i+k-1]}) &= \max\{i + k - 1, -\infty\} = i + k - 1, \\ \deg_{1,k-1}(y^{[i]}) &= \max\{-\infty, i + k - 1\} = i + k - 1, \\ \deg_{1,k-1}(x^{[i+k]}) &= \max\{i + k, -\infty\} = i + k. \end{aligned}$$

For comparing $x^{[i+k-1]}$ to $y^{[i]}$, we have

$$\begin{aligned} d_y(x^{[i+k-1]}) + k - 1 &= -\infty + k - 1 < i + k - 1, \\ d_y(y^{[i]}) + k - 1 &= i + k - 1 = i + k - 1. \end{aligned}$$

A linearized polynomial $f(x) = \sum_{i=0}^{d} a_i x^{[i]}$ is a linear combination of monomials, with coefficients in $\mathbb{F}_{q^m}$. Therefore, we can define the leading monomial of f under $\prec$ as the term whose monomial has highest weighted degree. We write the leading monomial of f under $\prec$ as $\mathrm{lm}_{\prec}(f)$. The coefficient of the leading monomial is denoted $\mathrm{lc}_{\prec}(f)$.

Lemma 3.28. *Let $f(x, y)$ and $g(x, y)$ be bivariate linearized polynomials over $\mathbb{F}_{q^m}$. If $f(x, y)$ and $g(x, y)$ are not comparable, then $\mathrm{lm}_{\prec}(f) = \mathrm{lm}_{\prec}(g)$.*

Proof. We prove the contrapositive. Suppose $\mathrm{lm}_{\prec}(f) \neq \mathrm{lm}_{\prec}(g)$. Because all monomials are comparable, the linearized polynomials $f(x, y)$ and $g(x, y)$ are comparable under $\prec$. □

Lemma 3.29. *Let $f(x, y)$ and $g(x, y)$ be bivariate linearized polynomials over $\mathbb{F}_{q^m}$ which are not comparable under $\prec$. Then, for some suitable $\gamma \in \mathbb{F}_{q^m}$, the polynomial*

$$h(x, y) = f(x, y) + \gamma g(x, y)$$

satisfies $h(x, y) \prec f(x, y)$ and $h(x, y) \prec g(x, y)$.

Proof. Suppose $f(x, y)$ and $g(x, y)$ are not comparable under $\prec$. Then $\mathrm{lm}_{\prec}(f) = \mathrm{lm}_{\prec}(g)$. Let $\gamma = \frac{\mathrm{lc}_{\prec}(f)}{\mathrm{lc}_{\prec}(g)}$ and consider

$$h(x, y) = f(x, y) + \gamma g(x, y).$$

Because the leading terms cancel, we have $\mathrm{lm}_{\prec}(h) \prec \mathrm{lm}_{\prec}(f) = \mathrm{lm}_{\prec}(g)$. □

Definition 3.30 (x-minimal and y-minimal)**.** Let $f(x, y)$ be a nonzero bivariate linearized polynomial, and $A = \{(x_1, y_1), \ldots, (x_r, y_r)\} \subseteq W$ be a linearly independent set. The polynomial f is called x-minimal with respect to A if $f(x, y)$ is a minimal polynomial under $\prec$ such that $f(x_i, y_i) = 0$ and $\mathrm{lm}_{\prec}(f) = x^{[d_x(f)]}$, and is called y-minimal with respect to A if $f(x, y)$ is a minimal polynomial under $\prec$ such that $f(x_i, y_i) = 0$ and $\mathrm{lm}_{\prec}(f) = y^{[d_y(f)]}$.

We use the previous lemmas to prove the correctness of the interpolation algorithm.

Theorem 3.31. *The polynomials $f_0(x, y)$ and $f_1(x, y)$ which are returned by the **Interpolate** algorithm (alg. 2) are x- and y-minimal (respectively), with respect to the basis points $(x_1, y_1), \ldots, (x_r, y_r)$ in U.*

Proof. The proof is by induction on the number of elements in the basis of U.

Base case: The polynomials $f_0(x, y) = x$ and $f_1(x, y) = y$ are x- and y-minimal (respectively) with respect to the empty set.

Induction step: For the induction hypothesis, we assume that f_0 and f_1 are x- and y-minimal (respectively) with respect to $(x_1, y_1), \ldots, (x_j, y_j)$. In order to show that f_0 and f_1 are x- and y-minimal when the next point (x_{j+1}, y_{j+1}) is considered, we first check that f_0 and f_1 vanish at this point. We have

$$\begin{aligned} f_1(x_{j+1}, y_{j+1}) &= f_1^q(x_{j+1}, y_{j+1}) - \Delta_1^{q-1} f_1(x_{j+1}, y_{j+1}) \\ &= f_1^q(x_{j+1}, y_{j+1}) - f_1^{q-1}(x_{j+1}, y_{j+1}) f_1(x_{j+1}, y_{j+1}) = 0, \end{aligned}$$

similarly for f_0.

To verify the next part of the definition of minimality (def. 3.30), we consider the four cases $\Delta_0 \neq 0, \Delta_1 \neq 0$, $\Delta_0 = 0, \Delta_1 \neq 0$, $\Delta_0 \neq 0, \Delta_1 = 0$ and $\Delta_0 = 0, \Delta_1 = 0$ in alg. 2 separately.

Case 1, $\Delta_0 \neq 0, \Delta_1 \neq 0$:

If $f_1(x, y) \prec f_0(x, y)$, then we let

$$f_0'(x, y) = \Delta_1 f_0(x, y) - \Delta_0 f_1(x, y).$$

Here $\mathrm{lm}_\prec(f_0') = \mathrm{lm}_\prec(f_0)$, so f_0' is x-minimal with respect to

$$\{(x_1, y_1), \ldots, (x_{j+1}, y_{j+1})\}.$$

For the next part, let

$$f_1'(x, y) = f_1^q(x, y) - \Delta_1^{q-1} f_1(x, y).$$

We show that $f_1'(x, y)$ is y-minimal with respect to $\{(x_1, y_1), \ldots, (x_{j+1}, y_{j+1})\}$. For contradiction, assume that $f_1'(x, y)$ is not y-minimal. Then there is a polynomial $f_1''(x, y)$, which is y-minimal with respect to $\prec$, such that $f_1''(x_i, y_i) = 0$, for $1 \leq i \leq j + 1$. Note that

$$\Delta_1 = f_1(x_{j+1}, y_{j+1}) \neq 0, \quad \text{while} \quad f_1''(x_{j+1}, y_{j+1}) = 0,$$

so the polynomials f_1 and f_1'' are not equal. However, the polynomials have the same leading monomial. This can be seen as follows. If $\mathrm{lm}_\prec(f_1'') \prec \mathrm{lm}_\prec(f_1)$, then f_1'' would be y-minimal on $\{(x_1, y_1), \ldots, (x_j, y_j)\}$, contradicting the y-minimality of f_1.

On the other hand, if $\mathrm{lm}_{\prec}(f_1) \prec \mathrm{lm}_{\prec}(f_1'')$, there are two possibilities when comparing $\mathrm{lm}_{\prec}(f_1')$ and $\mathrm{lm}_{\prec}(f_1'')$. This is because in the construction of f_1', the degree of f_1' increases by q. So if $\mathrm{lm}_{\prec}(f_1) = y^{[k]}$, then $\mathrm{lm}_{\prec}(f_1') = y^{[k+1]}$. If f_1' and f_1'' have the same leading monomial, we can use lemma 3.29 to form a new polynomial f_1''' which precedes both f_1' and f_1'', and would be zero on $\{(x_1, y_1), \ldots, (x_{j+1}, y_{j+1})\}$, contradicting the y-minimality of f_1''.

If $\mathrm{lm}_{\prec}(f_1') = y^{[k]}$, while $\mathrm{lm}_{\prec}(f_1'') = y^{[k+n]}, n \geq 1$, then f_1'' would not be y-minimal on $\{(x_1, y_1), \ldots, (x_j, y_j)\}$, and so, we must have $\mathrm{lm}_{\prec}(f_1) = \mathrm{lm}_{\prec}(f_1'')$. Therefore, we can use lemma 3.29 to form a polynomial $h(x, y)$, which satisfies

$$h(x, y) \prec f_1(x, y) \text{ and } h(x, y) \prec f_1''(x, y).$$

But then $h(x_i, y_i) = 0$, for $1 \leq i \leq j$ contradicting the assumption that $f_0(x, y)$ and $f_1(x, y)$ are x- and y-minimal (respectively) with respect to $\{(x_1, y_1), \ldots, (x_j, y_j)\}$. If $f_0(x, y) \prec f_1(x, y)$, then we can use the same argument.

Case 2, $\Delta_0 = 0, \Delta_1 \neq 0$:

In this case, $f_0'(x, y) = f_0(x, y)$, so f_0' is x-minimal because f_0 is. For updating $f_1(x, y)$, we have

$$f_1'(x, y) = f_1^q(x, y) - \Delta_1^{q-1} f_1(x, y),$$

and we must show that $f_1'(x, y)$ is y-minimal with respect to

$$\{(x_1, y_1), \ldots, (x_{j+1}, y_{j+1})\}.$$

For contradiction, assume that $f_1'(x, y)$ is not y-minimal. Then there is a polynomial $f_1''(x, y)$ different from $f_1(x, y)$, which satisfies $f_1''(x_i, y_i) = 0$, for $1 \leq i \leq j + 1$.Using the same reasoning as in case 1, the polynomials f_1'' and f_1 have the same leading monomial. We can then use lemma 3.29 to form a polynomial $h(x, y)$, which precedes both $f_1''(x, y)$ and $f_1(x, y)$ under $\prec$. If $h(x, y) \prec f_0(x, y)$, we get a contradiction to the assumption that $f_0(x, y)$ is x-minimal. If $\mathrm{lm}_{\prec}(h) = y^{[d_y(h)]}$, then $h(x, y) \prec f_1(x, y)$, which contradicts that $f_1(x, y)$ is y-minimal with respect to $\{(x_1, y_1), \ldots, (x_j, y_j)\}$. Because $h(x, y) = f_1''(x, y) + \gamma f_1(x, y)$, for some suitable $\gamma \in \mathbb{F}_{q^m}$, $h(x_{j+1}, y_{j+1}) \neq 0$. Recall that $f_0(x_{j+1}, y_{j+1}) = 0$, so $h(x, y)$ is not a $\otimes$-multiple of $f_0(x, y)$. Therefore, for some t, we can form the polynomial $h''(x, y)$ as (see lemma 3.29)

$$h''(x, y) = h(x, y) + \gamma \left[x^{[t]} \otimes f_0(x, y) \right],$$

where $h''(x, y) \prec h(x, y)$. The γ is chosen so that the leading terms cancel. Note that $x^{[t]} \otimes f_0(x, y)$ has the degree of all its terms increased by $[t]$,

compared to $f_0(x,y)$. Because $h''(x_{j+1}, y_{j+1}) \neq 0$, we repeat the above procedure, and arrive at a polynomial $\hat{h}(x,y)$. If $\mathrm{lm}_{\prec}(\hat{h}) = y^{[d_y(\hat{h})]}$, then we get a contradiction to the y-minimality of $f_1(x,y)$. If $\hat{h} \prec f_0$, we get a contradiction to the x-minimality of $f_0(x,y)$.

Case 3, $\Delta_0 \neq 0, \Delta_1 = 0$:

This case is proven in a similar way to case 2.

Case 4, $\Delta_0 = 0, \Delta_1 = 0$:

Here there is nothing to prove. □

3.7.3 Complexity Analysis of Interpolate

Because the degree of $f_0(x,y)$ and $f_1(x,y)$ only increases by at most q in each iteration of the algorithm, in the worst case the polynomials $f_0(x,y)$ and $f_1(x,y)$ have degree q^k, where k is the iteration number. This means that at the beginning of the k'th iteration, the polynomials have $2(k-1)$ terms each in the worst case. In order to compute Δ_0, observe that $\alpha^{q^k} = (\alpha^{q^{k-1}})^q$, so we can compute the powers of α iteratively. Therefore, when we evaluate $f_0(x,y)$ in (x_k, y_k), we need to evaluate

$$f(x_k, y_k) = \sum_{i=0}^{k-1} a_i x_k^{q^i} + \sum_{j=0}^{k-1} b_j y_k^{q^j},$$

which amounts to $2(k-1)+1$ additions, $2(k-1)$ multiplications, and $2(k-1)$ operations of raising to the q'th power. Assuming that these operations are roughly equally costly, computing Δ_0 requires $6(k-1)+1$ field operations in $\mathbb{F}_{q^m}$. Computing Δ_1 requires the same number of field operations as computing Δ_0, in the worst case.

Now suppose we are in the inner else block. Note that this block requires most computations in the for loop. For updating $f_1(x,y)$ according to

$$f_1(x,y) \leftarrow f_1^q(x,y) - \Delta_1^{q-1} f_1(x,y),$$

we need to do $2k$ operations of raising to the q'th power for computing $f_1^q(x,y)$. Computing $\Delta_1^{q-1} f_1(x,y)$ requires $2k$ multiplications, and we do $2k$ subtractions. In total, updating $f_1(x,y)$ requires $2k + 2k + 2k = 6k$ field operations in $\mathbb{F}_{q^m}$. We also need to update $f_1(x,y)$ according to

$$f_0(x,y) \leftarrow \Delta_1 f_0(x,y) - \Delta_0 f_1(x,y).$$

This requies $2k + 2k = 4k$ multiplications and $2k$ subtractions, yielding a total of $6k$ field operations.

Therefore, for iteration k, the algorithm does a total of $2 \cdot (6(k-1) + 1) + 6k + 6k = 24k - 10$ field operations in $\mathbb{F}_{q^m}$. Summing up over the r iterations, we get

$$\sum_{k=0}^{r}(24k - 10) = 24\sum_{k=0}^{r} k - 10r = 24\frac{r(r+1)}{2} - 10r$$

field operations. For large r, the expression above is dominated by r^2. In the worst case, the received space is the whole of W, which is the ambient space. Recalling that the dimension of W equals $\ell + m$, the algorithm has time complexity $O((\ell + m)^2)$. This is an advantage over ordinary Gaussian elimination, which requires $O(n^3)$ field operations for $n \times n$ linear systems.

3.7.4 Summary of the Decoding Procedure

We summarize the decoding procedure described. The polynomial returned by algorithm 2 is of minimal $(1, k-1)$ weighted degree $\tau - 1$. If a subspace $V \in \mathcal{C}$ of dimension ℓ is sent and a subspace U of W of dimension $\ell - \rho + t$ is received, then decoding can be done using the following steps.

1. Use the **Interpolate** algorithm (alg. 2) to find a bivariate linearized polynomial $Q(x, y) = Q_x(x) + Q_y(y)$ of minimal $(1, k-1)$ weighted degree which is zero on the points in U.

2. Divide $-Q_x(x)$ by $Q_y(y)$ using the **RDiv** algorithm to find a linearized polynomial $f(x)$ of degree at most q^{k-1} which satisfies

   $$-Q_x(x) \equiv Q_y(y) \otimes f(x).$$

   If no such polynomial $f(x)$ can be found, declare failure.

3. Identify $f(x)$ returned by **RDiv** with a codeword $\hat{V} \in \mathcal{C}$ if $d(U, \hat{V}) < \ell - k + 1$.

In the above, the linearized polynomial $f(x)$ is the message polynomial, that is, it has as coefficients the information symbols sent by the transmitter. Regarding step 3, recall from prop. 3.21 that a received space is decodable if and only if $\rho + t < \ell - k + 1$.

3.8 Examples

In this section, we present examples of transmission over the operator channel using subspace codes. These examples demonstrate the **Interpolate** and **RDiv** algorithms presented in previous sections. When doing computations in a finite field, we use vector notation for adding field elements, and exponential notation when multiplying elements. Since the fields used in the examples are $\mathbb{F}_2^3 \cong \mathbb{F}_8$ and $\mathbb{F}_2^4 \cong \mathbb{F}_{16}$, the addition and multiplication in those fields is shown below. These tables are taken from [LN83, p.546].

Vector notation	Exponential notation
001	0
010	1
100	2
101	3
111	4
011	5
110	6

Table 3.1: Elements of $\mathbb{F}_8 \backslash \{0\}$

Vector notation	Exponential notation
0001	0
0010	1
0100	2
1000	3
1001	4
1011	5
1111	6
0111	7
1110	8
0101	9
1010	10
1101	11
0011	12
0110	13
1100	14

Table 3.2: Elements of $\mathbb{F}_{16} \backslash \{0\}$

In table 3.1, the element (011) is written in exponential notation as α^5, where α is a primitive element in $\mathbb{F}_8$. For adding (010) to (101), we use vector notation. The result is (111). For multiplication, we use the exponential notation, so $\alpha^3 \alpha^2 = \alpha^{3+2} = \alpha^5$. Note that in the multiplication, when computing the exponent, we must reduce modulo the order of $\mathbb{F} \backslash \{0\}$. In the case of $\mathbb{F}_8$, we must reduce modulo 7 in the exponent, when doing multiplication.

Example 3.32. Using the parameters of the previous section, let $\ell = 2$, $m = 3$ and $q = 2$. The message symbols are taken from $\mathbb{F}_8 = \mathbb{F}_{2^3}$. We can regard $\mathbb{F}_8$ as a 3-dimensional vector space over $\mathbb{F}_2$, then we write it as $\mathbb{F}$. Let $A \subseteq \mathbb{F}$ be a linearly independent set, where

$$A = \{(011), (100)\}.$$

We use the canonical basis for $\mathbb{F}$ in this example, i.e.

$$\mathbb{F}_2^3 = \langle (100), (010), (001) \rangle .$$

From the definition of ambient space for subspace codes, the ambient space in this case is

$$W = \langle A \rangle \oplus \mathbb{F} = \{ (\alpha, \beta) \mid \alpha \in \langle A \rangle , \beta \in \mathbb{F} \}$$

One basis for W is thus

$$\{(011, 000), (100, 000), (000, 100), (000, 010), (000, 001)\}.$$

By taking $k = 1$, the message polynomial is of the form

$$f(x) = a_0 x,$$

where $a_0 \in \mathbb{F}$. Because there are 8 elements in $\mathbb{F}$, the sender can choose between 8 messages to send in each transmission. We now form the codewords. For $a_0 = 0$, $f(x) \equiv 0$, and so

$$V_0 = \langle (011, f(011)), (100, f(100)) \rangle = \langle (011, 000), (100, 000) \rangle$$

For the other codewords, let $a_0 = \alpha \in \mathbb{F}$. Then $f(x) = \alpha x$. To evaluate $f(x)$ in the basis for $\langle A \rangle$, we use the exponential notation in table 3.1. Then, we write (011) as α^3 and (100) as α^2. We therefore have

$$f(011) = f(\alpha^3) = \alpha\alpha^3 = \alpha^4, \text{ and } f(100) = f(\alpha^2) = \alpha\alpha^2 = \alpha^3.$$

Switching back to vector notation, α^4 can be written as (110), while α^3 can be written as (011). The codeword corresponding to $f(x) = \alpha x$ is therefore

$$V_2 = \langle (011, f(011)), (100, f(100)) \rangle = \langle (011, 110), (100, 011) \rangle .$$

We do the same for the other elements in $\mathbb{F}$. The possible codewords are therefore

$$\begin{aligned}
V_0 &= \langle (011, 000), (100, 000) \rangle , \\
V_1 &= \langle (011, 011), (100, 100) \rangle , \\
V_2 &= \langle (011, 110), (100, 011) \rangle , \\
V_3 &= \langle (011, 101), (100, 111) \rangle , \\
V_4 &= \langle (011, 111), (100, 110) \rangle , \\
V_5 &= \langle (011, 100), (100, 010) \rangle , \\
V_6 &= \langle (011, 001), (100, 101) \rangle , \\
V_7 &= \langle (011, 010), (100, 001) \rangle .
\end{aligned}$$

The code used in the transmission is a $[5, 2, 3, 4]$ subspace code $\mathcal{C}$, consisting of the codewords $V_0, V_1, \ldots, V_7$.

Now suppose the sender wants to send $a_0 = 0$. He then transmits the codeword V_0, which corresponds to $a_0 = 0$. We assume that the operator channel does 1 erasure and no errors. So $\rho = 1$ and $t = 0$. Suppose $V_1' = \mathcal{H}_1(V_1)$, where

$$V_1' = \langle (011, 000) \rangle .$$

Because no errors are assumed to have happend in the transmission , the receiver gets the space $V_1' \in \mathcal{P}(\mathbb{F})$. Also, the receiver knows $r = \dim(U_1) = 1$ and $k = 1$, so he can select $\tau = \left\lceil \frac{r+k}{2} \right\rceil = \left\lceil \frac{1+1}{2} \right\rceil = 1$.

The objective now is to form the interpolation polynomial $Q(x, y) = Q_x(x) + Q_y(y)$, satisfying the interpolation constraint

$$Q((011), (000)) = 0$$

and the degree constraints

$$\deg(Q_x(x)) \leq 2^{\tau-1} = 2^0 = 1, \deg(Q_y(y)) \leq 2^{\tau-k} = 2^0 = 1.$$

From the degree constraints, we see that the interpolation polynomial must be of the form $Q(x, y) = q_0 x + q_1 y$. Substituting the basis for V_1' and writing the basis elements in exponential notation, we have

$$Q(\alpha^3, 0) = q_0 \alpha^3 + q_1 0 = 0,$$

which implies that $q_0 = 0$ and q_1 can be any element from $\mathbb{F}$. We take $q_1 = \alpha^5$, so the interpolation polynomial becomes $Q(x, y) = \alpha^5 y$. We now find the message polynomial. Because $Q(x, f(x)) = Q_x(x) + Q_y(f(x)) = Q_y(f(x)) = 0$, we have $Q_y \otimes f(x) \equiv 0$. Because there are no zero divisors in $\mathscr{L}_{q^m}[X]$, we must have $f(x) \equiv 0$. Using this, the receiver decodes V_1' to the codeword V_1.

Example 3.33. This is a variation on the previous example. We use the same subspace code as in that example. Suppose the sender wants to send $a_0 = \alpha \in \mathbb{F}$. This corresponds to the codeword V_2, where

$$V_2 = \langle (011, 110), (100, 011) \rangle .$$

Assume that the operator channel does $\rho = 1$ erasure and $t = 0$ errors, the reciever gets the subspace

$$V_2' = \mathcal{H}_1(V_2) = \langle (011, 110) \rangle .$$

We now write the interpolation polynomial $Q(x, y)$.

$$Q(x, y) = q_0 \alpha^3 + q_1 \alpha^4 \equiv 0$$

which shows that $q_0 = \alpha^{-3} = \alpha^4$ and $q_1 = -\alpha^{-4} = \alpha^3$. Therefore, the interpolation polynomial becomes

$$Q(x, y) = Q_x(x) + Q_y(y) = \alpha^4 x - \alpha^3 y.$$

Substituting $f(x)$ for y, we have

$$\begin{aligned} Q(x, f(x)) = \alpha^4 x - \alpha^3 f(x) \equiv 0 \Rightarrow \\ \alpha^4 x = \alpha^3 f(x), \end{aligned}$$

from which we see that $f(x) = \alpha x$. The decoder therefore correctly decodes to the codeword V_2.

Example 3.34. This example is more substantial than the previous examples, in that we also use the **Interpolate** and **RDiv** algorithms. We set $k = 1$, so the message polynomial is of the form $f(x) = a_0 x$. In this example, the field used is $\mathbb{F}_{2^4} = \mathbb{F}_{16}$, which we denote by $\mathbb{F}$. This is a 4-dimensional vector space over $\mathbb{F}_2$, and a basis for this space is

$$\{(0001), (0010), (0100), (1000)\}.$$

We will use the exponential notation in table 3.2, so we write the elements in the basis for $\mathbb{F}$ as $1, \alpha, \alpha^2, \alpha^3$. Let $\langle A \rangle = \mathbb{F}$, then the ambient space is $W = \langle A \rangle \oplus \mathbb{F} = \mathbb{F} \oplus \mathbb{F}$. Suppose the sender wants to send to element $\alpha^2 \in \mathbb{F}$. The linearized polynomial is then $f(x) = \alpha^2 x$. The codeword formed is

$$\begin{aligned} V &= \left\langle (1, f(1)), (\alpha, f(\alpha)), (\alpha^2, f(\alpha^2)), (\alpha^3, f(\alpha^3)) \right\rangle \\ &= \left\langle (1, \alpha^2), (\alpha, \alpha^3), (\alpha^2, \alpha^4)), (\alpha^3, \alpha^5) \right\rangle. \end{aligned}$$

Suppose the operator channel does $\rho = 1$ erasure and $t = 1$ error. Then we assume that

$$V' = \mathcal{H}_3(V) = \left\langle (1, \alpha^2), (\alpha, \alpha^3), (\alpha^2, \alpha^4) \right\rangle, \text{ and } E = \langle (1, 1) \rangle.$$

The received space is then

$$U = \mathcal{H}_3(V) \oplus E = \left\langle (1, 1), (1, \alpha^2), (\alpha, \alpha^3), (\alpha^2, \alpha^4) \right\rangle.$$

We now run the **Interpolate** algorithm with input U.

At the initialization, the algorithm sets $f_0 := x$ and $f_1 := y$. The algorithm iterates through the basis elements for U.

Iteration $i = 1, (x_1, y_1) = (1, 1)$.

We have

$$\Delta_0 = 1, \ \Delta_1 = 1.$$

Because $\Delta_0 \neq 0$ and $\Delta_1 \neq 0$, we proceed to compare the degrees of f_0 and f_1:

$$\deg_{1,k-1}(f_0) = 0,\ \deg_{1,k-1}(f_1) = 1.$$

Because $\deg_{1,k-1}(f_0) \leq \deg_{1,k-1}(f_1)$, the updating is as follows:

$$\begin{aligned} f_0 &:= x^2 - x, \\ f_1 &:= x - y. \end{aligned}$$

Iteration $i = 2, (x_2, y_2) = (1, \alpha^2)$.

We have

$$\begin{aligned} \Delta_0 &= f_0(1, \alpha^2) = 1^2 - 1 = 0, \\ \Delta_1 &= f_1(1, \alpha^2) = 1 - \alpha^2 = \alpha^9 \neq 0. \end{aligned}$$

Because $\Delta_0 = 0$ and $\Delta_1 \neq 0$ we only update f_1 as follows:

$$\begin{aligned} f_1 &:= (x-y)^2 - (1-\alpha^2)(x-y) \\ &= x^2 + y^2 - \alpha^9(x-y) \end{aligned}$$

Iteration $i = 3, (x_3, y_3) = (\alpha, \alpha^3)$.

We have

$$\begin{aligned} \Delta_0 &= f_0(\alpha, \alpha^3) = \alpha^2 - \alpha = \alpha^{13} \neq 0, \\ \Delta_1 &= f_1(\alpha, \alpha^3) = \alpha^2 + \alpha^6 + \alpha^9(\alpha - \alpha^3) = \alpha \neq 0. \end{aligned}$$

Because $\Delta_0 \neq 0$ and $\Delta_1 \neq 0$, we proceed to compare the degrees of f_0 and f_1:

$$\deg_{1,k-1}(f_0) = 1,\ \deg_{1,k-1}(f_1) = 2.$$

As $\deg_{1,k-1}(f_0) \leq \deg_{1,k-1}(f_1)$, the updating is as follows:

$$\begin{aligned} f_0 &:= (x^2 + x)^2 + \alpha^{13}(x^2 + x) \\ &= x^4 + x^2 + \alpha^{13}x^2 + \alpha^{13}x \\ &= x^4 + \alpha^7 x^2 + \alpha^{13}x, \\ f_1 &:= \alpha(x^2 - x) + \alpha^{13}(x^2 + y^2 + \alpha^9(x + y)) \\ &= (\alpha + \alpha^{13})x^2 + (\alpha + \alpha^7)x + \alpha^{13}y^2 + \alpha^7 y \\ &= \alpha^2 x^2 + \alpha^9 x + \alpha^{13} y^2 + \alpha^7 y. \end{aligned}$$

Iteration $i = 4, (x_4, y_4) = (\alpha^2, \alpha^4)$.

We have

$$\Delta_0 = f_0(\alpha^2, \alpha^4) = \alpha^8 + \alpha^7\alpha^4 + \alpha^{13}\alpha^2 = \alpha^{12} + 1 = \alpha \neq 0,$$
$$\Delta_1 = f_1(\alpha^2, \alpha^4) = \alpha^2\alpha^4 + \alpha^9\alpha^2 + \alpha^{13}\alpha^8 + \alpha^7\alpha^4 = \alpha^6 + \alpha^6 = 0.$$

Because $\Delta_0 \neq 0$ and $\Delta_1 = 0$ we only update f_0 as follows:

$$\begin{aligned} f_1(x, y) &:= (x^4 + \alpha^7x^2 + \alpha^{13}x)^2 + \alpha(x^4 + \alpha^7x^2 + \alpha^{13}x) \\ &= x^8 + \alpha^{14}x^4 + \alpha^{26}x^2 + \alpha x^4 + \alpha^8x^2 + \alpha^{14}x \\ &= x^8 + (\alpha + \alpha^{14})x^4 + (\alpha^8 + \alpha^{26})x^2 + \alpha^{14}x \\ &= x^8 + \alpha^8x^4 + \alpha^{12}x^2 + \alpha^{14}x \end{aligned}$$

We now have the polynomials

$$f_0(x, y) = x^8 + \alpha^8x^4 + \alpha^{12}x^2 + \alpha^{14}x,$$
$$f_1(x, y) = \alpha^2x^2 + \alpha^9x + \alpha^{13}y^2 + \alpha^7y.$$

The $(1, k-1)$-degrees of these polynomials are

$$\deg_{1,k-1}(f_0) = \max\{4, 2 - 1 - \infty\} = 4,$$
$$\deg_{1,k-1}(f_1) = \max\{1, 2 - 1 + 1\} = 2.$$

Since $f_1(x, y)$ has lower $(1, k-1)$-degree than $f_0(x, y)$, the algorithm returns $f_1(x, y)$.

The interpolation polynomial is $Q(x, y) := f_1(x, y)$. We have $Q(x, y) = Q_x(x) + Q_y(y)$, where

$$Q_x(x) = \alpha^2x^2 + \alpha^9x,$$
$$Q_y(y) = \alpha^{13}y^2 + \alpha^7y.$$

The following computations show that the interpolation constraint is satisfied:

$$\begin{aligned} Q(1, 1) &= \alpha^2 + \alpha^9 + \alpha^{13} + \alpha^7 = \alpha^{12} + \alpha^{12} = 0 \\ Q(1, \alpha^2) &= \alpha^2 + \alpha^9 + \alpha^{13}\alpha^4 + \alpha^7\alpha^2 \\ &= \alpha^2 + \alpha^9 + \alpha^2 + \alpha^9 = 0 \\ Q(\alpha, \alpha^3) &= \alpha^2\alpha^2 + \alpha^9\alpha + \alpha^{13}\alpha^6 + \alpha^7\alpha^3 \\ &= \alpha^4 + \alpha^{10} + \alpha^4 + \alpha^{10} = 0 \\ Q(\alpha^2, \alpha^4) &= \alpha^2\alpha^4 + \alpha^9\alpha^2 + \alpha^{13}\alpha^8 + \alpha^7\alpha^4 \\ &= \alpha^6 + \alpha^{11} + \alpha^6 + \alpha^{11} = 0. \end{aligned}$$

For the degree constraint, we have $\tau = \left\lceil \frac{4+1}{2} \right\rceil = 3$, and so

$$\deg(Q_x(x)) = 2 \leq 2^{3-1} = 4,$$
$$\deg(Q_y(y)) = 2 \leq 2^{3-1} = 4.$$

We now run the **RDiv** algorithm with input $-Q_x(x)$ and $Q_y(y)$. Then, because $\deg(Q_x(x)) = \deg(Q_y(y))$, the algorithm computes $t(x)$:

$$t(x) = \left(\frac{a^2}{a^{13}}\right)^{[4-1]} x^{[1-1]} = (\alpha^{-11})^8 x = \alpha^2 x$$

Then

$$\begin{aligned} Q_x(x) - Q_y(t(x)) &= \alpha^2 x^2 + \alpha^9 x + \alpha^{13}(\alpha^2 x)^2 + \alpha^7(\alpha^2 x) \\ &= \alpha^2 x^2 + \alpha^9 x + \alpha^2 x^2 + \alpha^9 x = 0 \end{aligned}$$

Then we execute the statement **return**$(\alpha^2 x, 0)$+**RDiv**$(0, Q_y(y))$.

Now, because $\deg(0) = -\infty < \deg(Q_y(y)) = 2$, the algorithm returns $(0, 0)$.

This implies that $q_R(x) = \alpha^2 x$, and so the message polynomial is $f(x) = \alpha^2 x$.

Chapter 4

List Decoding of Subspace Codes

In the previous chapter, we looked at KK-codes and described their mathematical properties, along with a decoding algorithm for these codes. This algorithm was shown to return the transmitted codeword, as long as not too many erasures and errors occured in the transmission. In this chapter, we will describe a class of subspace codes which can be decoded by means of list-decoding. In list-decoding, the decoder produces a list of codewords containing the sent codeword. We will look at how the problem of finding an interpolation polynomial can be stated in the language of modules, along with an algorithm which finds the roots of a multivariate linearized polynomial. This algorithm is used in the list-decoding algorithms. The codes suitable for list decoding, along with supporting results, are from [MV12], while the introduction to list decoding is adapted from [Rot06].

4.1 List Decoding

In this section, we describe list decoding. Suppose we have a subspace code $\mathcal{C}$, and that the codeword $V \in \mathcal{C}$ is sent over the operator channel. The receiver gets a subspace $U \in \mathcal{P}(W)$, where W is the ambient space. In list-L decoding, the decoder takes the received subspace U and produces a list of codewords in $\mathcal{C}$. This size of the list is at most L. Decoding is considered successful if the list contains the sent codeword. One quantity describing a list decoder is the decoding radius. The decoding radius τ is defined to be the smallest number such that if $V \in \mathcal{C}$ was sent and $U \in \mathcal{P}(W)$ was received, then $d(U, V) \leq \tau$ implies that the codeword V is contained in the list.

4.2 Extension of KK-codes

Recall from sec. 3.6 that a codeword in a KK-code is of the form

$$V = \langle (\alpha_1, f(\alpha_1)), \ldots, (\alpha_\ell, f(\alpha_\ell)) \rangle$$

where $\alpha_1, \ldots, \alpha_\ell$ constitute a basis for $\langle A \rangle$ and $f(x) \in \mathscr{L}_{q^m}^k[X]$ is a linearized polynomial corresponding to the message that is to be encoded. In the previous chapter, a decoding algorithm for the decoding of KK-codes was described. This algorithm can be viewed as a list decoding algorithm, where the list size is 1. As is done in classical coding theory, one performs list-decoding in order to be able to correct more errors. List decoding can also be performed for subspace codes, when the list size is at most L, and includes the sent codeword. Using the KK-codes as a basis, and a list size of at most 2, one could define the codeword as

$$V = \left\langle (\alpha_1, f(\alpha_1), f^{\otimes 2}(\alpha_1)), \ldots, (\alpha_\ell, f(\alpha_\ell), f^{\otimes 2}(\alpha_\ell)) \right\rangle.$$

This codeword is sent over a operator channel. Suppose that not too many errors or erasures occur, and U is the space received. The receiver then constructs an interpolation polynomial of the form

$$Q(x, y_1, y_2) = Q_0(x) + Q_1(y_1) + Q_2(y_2).$$

satisfying certain degree constraints and an interpolation constraint. Then, the receiver finds all polynomials $f(x) \in \mathscr{L}_{q^m}^k[X]$ which are roots of the interpolation polynomial. Suppose that $f(x)$ is a root. Then the roots of the univariate linearized polynomial

$$Q(x, f(x), f^{\otimes 2}(x)) = Q_0(x) + Q_1(f(x)) + Q_2(f^{\otimes 2}(x)) \tag{4.1}$$

are in some field. If there are more roots that the degree, then $Q(x, f(x), f^{\otimes 2}(x))$ is identically zero. Then, the receiver attempts to reconstruct the message polynomial $f(x)$ from eq. (4.1). However, because the coefficients of $f(x)$ are in $\mathbb{F}_{q^m}$, there may be more then two solutions in $\mathscr{L}_{q^m}^k[X]$, which is demonstrated in example 4.1.

Example 4.1. Consider the equation

$$y^{\otimes 2} - x^{q^2} = 0 \tag{4.2}$$

which we want to solve for y. If we allow solutions from $\mathscr{L}_{q^m}^k[X]$, then there are more than two solutions. It is seen that $f(x) = ux^q$, where $u^{q+1} = 1$ is a solution, because

$$y^{\otimes 2} = (ux^q) \otimes (ux^q) = u^{q+1} x^{q^2} = x^{q^2}.$$

If m is even, then

$$(q+1)(q^{m-1} - q^{m-2} + q^{m-3} - \cdots + q - 1) = q^m - 1$$

and in this case, $q+1$ divides $q^m - 1$. Note that $\mathbb{F}_{q^m} \backslash \{0\} = \mathbb{F}_{q^m}^*$ is a cyclic group of order $q^m - 1$, where the group operation is multiplication. Now, even though the converse to the theorem of Lagrange doesn't hold in general, it does hold for cyclic groups. Therefore, because $q+1$ divides the order of $\mathbb{F}_{q^m}^*$, $\mathbb{F}_{q^m}^*$ has a subgroup of order $q+1$, generated by α. Every element in this subgroup is of the form α^k, for some integer k. Because $(\alpha^k)^{q+1} = (\alpha^{q+1}) = 1$, every element in the subgroup can be used as a coefficient u of x^{q^2} in $f(x)$. Therefore, there are $q+1$ possible solutions to eq. (4.2).

On the other hand, suppose we only consider solutions in $\mathscr{L}_q^k[X]$. Then $f(x) = ux^{q^2}$ is a solution, for every $u \in \mathbb{F}_q$ satisfying $u^{q+1} = 1$. But because $u \in \mathbb{F}_q$, $u^q = u$. Therefore, $u^{q+1} = u^q u = u^2 = 1$, which shows that there are at most two solutions in $\mathscr{L}_q^k[X]$ to eq. (4.2).

As the previous example demonstrates, we cannot consider the ring $\mathscr{L}_{q^m}[X]$, when constructing the message polynomial. This is because the list can contain too many codewords. However, we saw that $\mathscr{L}_q[X]$, which is a commutative ring, is a possible replacement. Therefore the codes to be considered are modified in order to enable list-L decoding. By considering the subring $\mathscr{L}_q[X]$ of $\mathscr{L}_{q^m}[X]$, the list size is upper bounded by L, as the following theorem shows.

Theorem 4.2. *Let $Q_0(x), Q_1(x), \ldots, Q_L(x)$ be linearized polynomials over $\mathbb{F}_{q^m}$, and suppose that at least one of these polynomials is nonzero. Then the equation*

$$\sum_{i=0}^{L} Q_i \otimes f^{\otimes i}(x) = 0 \tag{4.3}$$

has at most L solutions $f(x)$ in $\mathscr{L}_q^k[X]$.

Proof. The proof is by induction on L. For the base case, we have $L = 0$, and so eq. (4.3) reduces to

$$Q_0(x) \otimes f^{\otimes 0}(x) = Q_0(x) = 0. \tag{4.4}$$

By assumption, Q_0 is nonzero, so there is no solution in $\mathscr{L}_q^k[X]$ in this case.

We now turn to the induction step. For the induction hypothesis, suppose

$$\sum_{i=0}^{L-1} Q_i \otimes f^{\otimes i}(x) = 0$$

has at most $L-1$ solutions in $\mathscr{L}_q^k[X]$. Consider the equation

$$\sum_{i=0}^{L} Q_i \otimes f^{\otimes i}(x) = 0.$$

If this equation has no solution in $\mathscr{L}_q^k[X]$, we are done. Otherwise, suppose $f_0(x) \in \mathscr{L}_q^k[X]$ is a solution. We show that there are at most $L-1$ other solutions. We have

$$\sum_{i=0}^{L} Q_i \otimes f^{\otimes i}(x) - \sum_{i=0}^{L} Q_i \otimes f_0^{\otimes i}(x) = \sum_{i=0}^{L} Q_i \otimes \left(f^{\otimes i} - f_0^{\otimes i}\right) = 0. \tag{4.5}$$

We now rewrite $f^{\otimes i} - f_0^{\otimes i}$. Recall that $f^{\otimes 1} = f$ and $f^{\otimes 0} = x$.

$$\begin{aligned}
f^{\otimes i} - f_0^{\otimes i} =& f_0^{\otimes 0} \otimes f^{\otimes i} - f_0^{\otimes i} \otimes f^{\otimes 0} \\
=& f_0^{\otimes (i-1)} \otimes f^{\otimes 1} - f_0^{\otimes i} \otimes f^{\otimes 0} + f_0^{\otimes (i-2)} \otimes f^{\otimes 2} - f_0^{\otimes (i-1)} \otimes f^{\otimes 1} + \cdots + \\
& f_0^{\otimes 1} \otimes f^{\otimes (i-1)} - f_0^{\otimes 2} \otimes f^{\otimes (i-2)} + f_0^{\otimes 0} \otimes f^{\otimes i} - f_0^{\otimes 1} \otimes f^{\otimes (i-1)} \\
=& \Big(f_0^{\otimes (i-1)} \otimes f^{\otimes 0} + f_0^{\otimes (i-2)} \otimes f^{\otimes 1} + \cdots + \\
& f_0^{\otimes 1} \otimes f^{\otimes (i-2)} + f_0^{\otimes 0} \otimes f^{\otimes (i-1)}\Big) \otimes (f - f_0) \\
=& \left(\sum_{j=0}^{i-1} f_0^{\otimes (i-j-1)} \otimes f^{\otimes j}\right) \otimes (f - f_0),
\end{aligned}$$

where we have used that $\mathscr{L}_q[X]$ is a commutative ring. We can substitute the expression for $f^{\otimes i} - f_0^{\otimes i}$ into eq. (4.5). Then, we have

$$\sum_{i=1}^{L} Q_i \otimes \left(\sum_{j=0}^{i-1} (f_0^{\otimes (i-j-1)} \otimes f^{\otimes j}) \otimes (f - f_0)\right) = 0.$$

which, by associativity, implies

$$\left(\sum_{i=1}^{L} Q_i \otimes \sum_{j=0}^{i-1} (f_0^{\otimes (i-j-1)} \otimes f^{\otimes j})\right) \otimes (f - f_0) = 0.$$

Because $f \neq f_0$, $f - f_0$ is nonzero. Also, since there are no zero divisors in $\mathscr{L}_{q^m}[X]$, we must have

$$\sum_{i=1}^{L} Q_i \otimes \sum_{j=0}^{i-1} (f_0^{\otimes (i-j-1)} \otimes f^{\otimes j}) = 0.$$

Changing the order of summation, the above can be rewritten to

$$\sum_{j=0}^{L-1} \left(\sum_{i=j+1}^{L} Q_i \otimes f_0^{\otimes (i-j-1)}\right) \otimes f^{\otimes j} = 0.$$

Using the induction hypothesis, this equation has at most $L-1$ solutions in $\mathscr{L}_q^k[X]$. By induction, the result follows. □

Because the information symbols are from $\mathbb{F}_q$ rather than $\mathbb{F}_{q^m}$, the rate is reduced by a factor m. In order to compensate for this, a normal basis for $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$ is introduced.

Definition 4.3 (Normal Basis)**.** Let $\mathbb{F}_{q^m}$ be a extension field of $\mathbb{F}_q$, and let $\alpha \in \mathbb{F}_{q^m}$. Then the set

$$\{\alpha, \alpha^q, \alpha^{q^2}, \ldots, \alpha^{q^{m-1}}\},$$

is called a normal basis for $\mathbb{F}_{q^m}$, when regarded as a vector space over $\mathbb{F}_q$.

Any field extension $\mathbb{F}_{q^m}$ of $\mathbb{F}_q$ has a normal basis [LN83, thm. 2.35], and a normal basis is actually a basis for $\mathbb{F}_{q^m}$ as vector space over $\mathbb{F}_q$. Now, let $f(x)$ be a linearized polynomial over $\mathbb{F}_q$. Then, as $f(\alpha^{q^j}) = f(\alpha)^{q^j}$, once $f(\alpha)$ is received, all powers of it can be computed at the receiver. This means that they don't have to be transmitted, compensating for the aforementioned rate reduction.

4.3 Mahdavifar Vardy Codes

We will begin by describing the encoding procedure of a message, so that it can be list-decoded at the receiver. Let a finite field $\mathbb{F}_q$ and an extension field $\mathbb{F}_{q^m}$ be fixed. In the following, we sometimes write $[i]$ for q^i. Let $\alpha \in \mathbb{F}_{q^m}$ be an element which generates a normal basis for $\mathbb{F}_{q^m}$ as a vector space over $\mathbb{F}_q$. Let $\mathbf{u} = (u_0, u_1, \ldots, u_{k-1}) \in \mathbb{F}_q^k$ be the message that is to be encoded. We require that $k \leq m$, which is related to the injectivity requirement of the encoding. The first part is to form a linearized polynomial

$$f(x) = \sum_{i=0}^{k-1} u_i x^{q^i}.$$

Similar to what was done for KK-codes in the previous chapter, we define an evaluation mapping.

Definition 4.4. Let $f(x) \in \mathscr{L}_q^k[X]$ be the linearized polynomial whose coefficients are the components of the message vector $\mathbf{u}$, and let α be a generator for a normal basis for $\mathbb{F}_{q^m}$ as a vector space over $\mathbb{F}_q$. Define the mapping $\mathsf{ev}_{\alpha,L}$ as

$$\begin{aligned}&\mathsf{ev}_{\alpha,L} : \mathscr{L}_q^k[X] \to \mathcal{P}(W, 1), \quad \text{given by}\\ &\mathsf{ev}_{\alpha,L}(f) = \left\langle (\alpha, f(\alpha), f^{\otimes 2}(\alpha), \ldots, f^{\otimes L}(\alpha)) \right\rangle.\end{aligned}$$

We call this map evaluation of f at α.

The image of f under the mapping $\mathsf{ev}_{\alpha,L}$ is a subspace of W, where

$$W = \langle \alpha \rangle \oplus \underbrace{\mathbb{F}_{q^m} \oplus \cdots \oplus \mathbb{F}_{q^m}}_{L \text{ times}}$$
$$= \langle \alpha \rangle \oplus \left(\bigoplus_{i=1}^{L} \mathbb{F}_{q^m} \right).$$

The subspace $\mathsf{ev}_{\alpha,L}(f)$, which we denote V, has dimension 1, and the ambient space W is of dimension $1 + Lm$. Then, for each $\mathbf{u} \in \mathbb{F}_q^k$, we get a linearized polynomial $f(x) \in \mathscr{L}_q^k[X]$. The image of $\mathscr{L}_q^k[X]$ under $\mathsf{ev}_{\alpha,L}$ is a subspace code $\mathcal{C}$, and it has a special name.

Definition 4.5 (One-dimensional MV-code)**.** Consider the evaluation mapping as defined in def. 4.4. The image of $\mathscr{L}_q^k[X]$ under $\mathsf{ev}_{\alpha,L}$ is a subspace code, which we call a one-dimensional MV-code [1]. This code is denoted $\mathcal{C}_q(k, 1, m, L)$, where k is the length of the message vector, 1 is the dimension of $\langle \alpha \rangle$, m is the dimension of $\mathbb{F}_{q^m}$ as a vector space over $\mathbb{F}_q$, while L is the list size.

Later we will consider MV-codes where the codeword dimension is ℓ. Then the code is denoted $\mathcal{C}_q(k, \ell, m, L)$. The distance between two codewords is lower bounded by $2(\ell - \left\lceil \frac{k}{m} \right\rceil + 1)$ [Mah12], which will be shown in prop. 4.16 in sec. 4.5. Note that because the message polynomials of an MV-code are from $\mathscr{L}_q^k[X]$ instead of $\mathscr{L}_{q^m}^k[X]$, MV-codes contain fewer codewords than corrsponding KK-codes.

We can think of a packet as being a vector with entries in some finite field. The packet rate of a MV-code is then defined as follows.

Definition 4.6 (Packet Rate)**.** Let $\mathcal{C}_q(k, \ell, m, L)$ be an MV-code as defined in def. 4.5. The packet rate R^* of $\mathcal{C}$ is then defined as

$$R^* = \frac{\log_{q^m}(|\mathcal{C}|)}{\ell} = \frac{\log_q(|\mathcal{C}|)}{\ell m}.$$

Using the code defined in the previous section, the transmitter sends the codeword V through an operator channel. We assume that this operator channel does no erasure, since if one erasure occurs, all information is lost. We also assume that the error space E is of dimension t. Suppose U is received, where $\dim(U) = 1 + t$. It will be shown that if the following upper bound on the errors hold,

$$t < L - \frac{L(L+1)}{2} \frac{(k-1)}{m}, \tag{4.6}$$

[1] MV is short for Mahdavifar Vardy

then list decoding with list size smaller than or equal to L is successful. This last statement means that the decoder outputs a list of codewords of size at most L, and this list contains the transmitted codeword. As mentioned in the start of this chapter, list decoding comprises three steps. We describe these steps for the MV-codes of this section, and then prove the correctness of the method.

Computing the interpolation points

Suppose a space $U \in \mathcal{P}(W)$ is received, with $\dim(U) = t + 1$. The receiver the finds a basis for U consisting of the points

$$(x_i, y_{i,1}, \ldots, y_{i,L}),\ i = 1, 2, \ldots, t + 1.$$

From this basis, the interpolation points are formed, giving

$$\mathcal{P} = \{(x_i^{q^h}, y_{i,1}^{q^h}, \ldots, y_{i,L}^{q^h}) \mid 1 \leq i \leq t + 1,\ 0 \leq h \leq m - 1\}$$

The set $\mathcal{P}$ thus consists of $m(t + 1)$ points from W.

Constructing the interpolation polynomial

Using the points from the set $\mathcal{P}$ constructed in the previous step, the reciever forms a multivariate linearized polynomial $Q(x, y_1, \ldots, y_L)$, where

$$Q(x, y_1, \ldots, y_L) = Q_0(x) + Q_1(y_1) + \cdots + Q_L(y_L)$$

satisfies the degree constraints

$$\begin{aligned}
&\deg(Q_0) \leq q^{m-1},\\
&\deg(Q_1) \leq q^{m-(k-1)-1} = q^{m-k},\\
&\deg(Q_2) \leq q^{m-2(k-1)-1} = q^{m-2k+1},\\
&\qquad\vdots\\
&\deg(Q_L) \leq q^{m-L(k-1)-1} = q^{m-Lk+(L-1)},
\end{aligned}$$

and the interpolation constraint

$$Q(x, y_1, \ldots, y_L) = 0,\ \text{for all } (x, y_1, \ldots, y_L) \in \mathcal{P}.$$

Finding the message polynomial

Using the interpolation polynomial $Q(x, y_1, \ldots, y_L)$ found in the previous step, and knowledge of k, the receiver calls the **LRR** algorithm with parameters $(Q, k, 0)$. This algorithm is described in detail in section 4.7.

4.4 Correctness of List-L Decoding for MV-codes of Dimension One

In this section, we show that list-L decoding of MV-codes is possible, when the codewords have dimension 1. We want to prove two things. First, that the list size is upper bounded by L, and second, that the list includes the sent codeword. To do so, we first prove that the interpolation polynomial is non-trivial, given certain assumptions.

Lemma 4.7. *With t, L, k and m as in the previous section, if*

$$t < L - \frac{L(L+1)}{2}\frac{(k-1)}{m},$$

then there is a non-trivial multivariate linearized polynomial $Q(x, y_1, \ldots, y_L)$ satisfying the interpolation and degree constraints.

Proof. The interpolation constraint $Q(x, y_1, \ldots, y_L) = 0$ is a homogeneous system of equations. The number of equations is at most $m(t+1)$, because there are at most $m(t+1)$ elements in $\mathcal{P}$. From the degree constraints, the number of unknowns is

$$\begin{aligned}\sum_{i=0}^{L}(m-(k-1)i) &= (L+1)m - \sum_{i=0}^{L} i(k-1) \\ &= (L+1)m - (k-1)\frac{L(L+1)}{2}.\end{aligned}$$

A homogeneous system has a non-trivial solution as long as the number of equations is less than the number of unknowns. Therefore, there is a nontrivial solution as long as

$$m(t+1) < (L+1)m - (k-1)\frac{L(L+1)}{2},$$

which can be rearranged to

$$t < L - \frac{(k-1)}{m}\frac{L(L+1)}{2}.$$

□

Let $Q(x, y_1, \ldots, y_L)$ be the interpolation polynomial, and $f(x) \in \mathscr{L}_q^k[X]$ be the message polynomial. Define $E(x)$ as

$$E(x) = Q(x, f(x), f^{\otimes 2}(x), \ldots, f^{\otimes L}(x)) = \sum_{i=0}^{L} Q_i \otimes f^{\otimes i}(x).$$

Note that $E(x)$ is a univariate polynomial.

Lemma 4.8. *For $j = 0, 1, 2, \ldots, m-1$, we have $E(\alpha^{q^j}) = 0$.*

Proof. Because we assume that no erasures occur in the transmission, the received vector space U is a subspace of the sent codeword V, and so $(\alpha, f(\alpha), f^{\otimes 2}(\alpha), \ldots, f^{\otimes L}(\alpha)) \in U$. Now, any point $(x, y_1, \ldots, y_L) \in \mathcal{P}$ is a root of the interpolation polynomial. Also, because $(x^{q^j}, y_1^{q^j}, \ldots, y_L^{q^j}) \in \mathcal{P}$, $Q(x^{q^j}, y_1^{q^j}, \ldots, y_L^{q^j}) = 0$. Hence, we have

$$Q(\alpha^{q^j}, f(\alpha)^{q^j}, \ldots, f^{\otimes L}(\alpha)^{q^j}) = 0.$$

But because $f(x)$ is a linearized polynomial, $f(\alpha)^{q^j} = f(\alpha^{q^j})$. From this, we have

$$E(\alpha^{q^j}) = Q(\alpha^{q^j}, f(\alpha^{q^j}), \ldots, f^{\otimes L}(\alpha^{q^j})) = 0.$$

□

Corollary 4.9. *$E(x)$ is the all zero polynomial.*

Proof. Because $f(x) \in \mathscr{L}_q^k[X]$, $\deg(f) \leq q^{k-1}$. Also, for each $Q_i(x_i)$, we have $\deg(Q_i) \leq q^{m-(k-1)i-1}$. Therefore, the degree of $Q_i(f^{\otimes i})$ is

$$\deg(Q_i(f^{\otimes i})) \leq q^{k-1} q^{m-(k-1)i-1} = q^{m-1}.$$

From lemma 4.8, $E(x)$ has at least m linearly independent roots

$$\alpha, \alpha^q, \alpha^{q^2}, \ldots, \alpha^{q^{m-1}}.$$

Because $E(x)$ is a univariate linearized polynomial, these roots form a subspace which consists of q^m elements. Because the degree of $E(x)$ is at most q^{m-1}, $E(x)$ is the all zero polynomial. □

Theorem 4.10. *If the inequality*

$$t < L - \frac{L(L+1)}{2} \frac{(k-1)}{m} \tag{4.7}$$

holds, then the list decoder provides a list of size at most L, and this list includes the transmitted codeword.

Proof. Suppose (4.7) holds. Then from lemma 4.7, a non-trivial interpolation polynomial exists. Then,using corollary 4.9, $E(x)$ is identically zero, which implies that the message polynomial $f(x)$ is contained in the list. From theorem 4.2, the are at most L solutions in $\mathscr{L}_q^k[X]$ to $Q(x, y_1, \ldots, y_L) = 0$. Therefore, the list size is at most L. □

4.5 MV-codes with Arbitrary List Size and Codeword Dimension

In the previous section MV-codes were defined, where the codewords had dimension 1. One drawback of these codes is that they cannot correct one erasure. The codes in this section, due to [MV12], have codewords with dimension greater than one. This makes it possible to correct more erasures in the transmission. However, one-dimensional MV-codes are defined through an generator α for a normal basis of $\mathbb{F}_{q^m}$. So it is not beneficial to add another element β which generates a different normal basis for $\mathbb{F}_{q^m}$, as α already spans the entire field $\mathbb{F}_{q^m}$ [MV12]. Therefore, the elements $\alpha_1, \ldots, \alpha_\ell$ are defined from a generator for a normal basis for $\mathbb{F}_{q^m}$, and the space $\langle \alpha_1, \ldots, \alpha_\ell \rangle$ is then viewed as a subspace of an extension field $\mathbb{F}_{q^{m\ell}}$ of $\mathbb{F}_{q^m}$.

Lemma 4.11. *Suppose ℓ divides $q-1$. Then the equation*

$$x^\ell - 1 = 0$$

has ℓ solutions in $\mathbb{F}_q$.

Proof. If ℓ divides $q-1$, then $\mathbb{F}_q^*$ has a subgroup of order ℓ. The elements in this subgroup are roots of $x^\ell - 1$. □

Let $\mathbb{F} = \mathbb{F}_{q^{m\ell}}$ be an extension field of $\mathbb{F}_{q^m}$. From [LS87], there exists a primitive element $\gamma \in \mathbb{F}$, which generates a normal basis for $\mathbb{F}$ as a vector space over $\mathbb{F}_{q^m}$. We can then write

$$\begin{aligned} \mathbb{F} &= \left\langle \gamma^{(q^m)^0}, \gamma^{(q^m)^1}, \gamma^{(q^m)^2}, \ldots, \gamma^{(q^m)^{(\ell-1)}} \right\rangle \\ &= \left\langle \gamma, \gamma^{q^m} \gamma^{q^{2m}}, \ldots, \gamma^{q^{(\ell-1)m}} \right\rangle. \end{aligned}$$

Also, using lemma 4.11, let $e_1 = 1, e_2, \ldots, e_\ell$ be the ℓ solutions in $\mathbb{F}_q$ to the equation $x^\ell = 1$. The elements $\alpha_1, \alpha_2, \ldots, \alpha_\ell$ are defined as

$$\alpha_i = \gamma + e_i^{-1}\gamma^{q^m} + e_i^{-2}\gamma^{q^{2m}} + \cdots + e_i^{-(\ell-1)}\gamma^{q^{(\ell-1)m}}, \tag{4.8}$$

for $i = 1, 2, \ldots, \ell$. The elements α_i are used for evaluation of a message polynomial, to define a codeword.

Lemma 4.12. *Let $\alpha_1, \ldots, \alpha_\ell$ be defined as in eq. (4.8). Then $\alpha_1 \in \mathbb{F}_{q^m}$, and $\alpha_i^\ell \in \mathbb{F}_{q^m}$, for $i = 2, \ldots, \ell$.*

Proof. For $i = 1, 2, \ldots, \ell$, we have

$$\begin{aligned}
\alpha_i^{q^m} &= \left(\sum_{j=0}^{\ell-1} e_i^{-j}\gamma^{q^{mj}}\right)^{q^m} = \sum_{j=0}^{\ell-1}\left(e_i^{-j}\right)^{q^m}\left(\gamma^{q^{mj}}\right)^{q^m} \\
&= \sum_{j=0}^{\ell-1}\left(e_i^{q^m}\right)^{-j}\gamma^{q^{m(j+1)}} = \sum_{j=0}^{\ell-1} e_i^{-j}\gamma^{q^{m(j+1)}} \\
&= \gamma^{q^m} + e_i^{-1}\gamma^{q^{2m}} + \cdots + e_i^{-(\ell-1)}\gamma^{q^{\ell m}} \\
&= e_i\left(e_i^{-1}\gamma^{q^m} + e_i^{-2}\gamma^{q^{2m}} + \cdots + e_i^{-\ell}\gamma^{q^{\ell m}}\right) \\
&\overset{(1)}{=} e_i\left(\gamma + e_i^{-1}\gamma^{q^m} + e_i^{-2}\gamma^{q^{2m}} + \cdots + e_i^{-(\ell-1)}\gamma^{q^{(\ell-1)m}}\right) \\
&= e_i\alpha_i,
\end{aligned}$$

where we in (1) have used that $\gamma \in \mathbb{F}_{q^{m\ell}}$. From this, we get that $\alpha_1^{q^m} = e_i\alpha_1 = \alpha_1$, which implies that $\alpha_1 \in \mathbb{F}_{q^m}$. Also, for $i = 2, 3, \ldots, \ell$, $\left(\alpha_i^{\ell}\right)^{q^m} = \left(\alpha_i^{q^m}\right)^{\ell} = (e_i\alpha_i)^{q^m} = \alpha_i^{\ell}$, because e_i is a root of $x^{\ell} - 1$. Hence $\alpha_i^{\ell} \in \mathbb{F}_{q^m}$, for $i = 2, 3, \ldots, \ell$. □

Lemma 4.13. *Let $\mathbb{F} = \mathbb{F}_{q^{m\ell}}$ be an extension field over $\mathbb{F}_{q^m}$. Then the set*

$$Z = \{\alpha_i^{q^j} \mid 1 \le i \le \ell, 0 \le j \le m-1\}$$

is a basis for $\mathbb{F}$ as a vector space over $\mathbb{F}_q$, where the α_i are as defined in eq. (4.8).

Proof. Let

$$\begin{aligned}
A &= [\alpha_1, \alpha_2, \ldots, \alpha_\ell], \\
\Gamma &= \left[\gamma, \gamma^{q^m}, \ldots, \gamma^{q^{(\ell-1)m}}\right], \text{ and} \\
E &= \begin{bmatrix} 1 & e_1^{-1} & e_1^{-2} & \cdots & e_1^{-(\ell-1)} \\ 1 & e_2^{-1} & e_2^{-2} & \cdots & e_2^{-(\ell-1)} \\ & \vdots & & \vdots & \\ 1 & e_\ell^{-1} & e_\ell^{-2} & \cdots & e_\ell^{-(\ell-1)} \end{bmatrix}.
\end{aligned}$$

Notice that E is a Vandermonde matrix, and because the e_i^{-1}'s are distinct, E is invertible over $\mathbb{F}_q$. Also

$$\begin{aligned}
\Gamma E^{\top} &= \left[\gamma, \gamma^{q^m}, \ldots, \gamma^{q^{(\ell-1)m}}\right]\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ e_1^{-1} & e_2^{-1} & e_3^{-1} & \cdots & e_\ell^{-1} \\ & \vdots & & \vdots & \\ e_1^{-(\ell-1)} & e_2^{-(\ell-1)} & e_3^{-(\ell-1)} & \cdots & e_\ell^{-(\ell-1)} \end{bmatrix} \\
&= [\alpha_1, \alpha_2, \ldots, \alpha_\ell] = A.
\end{aligned}$$

So $A = \Gamma E^\top$, and because E is invertible, so is $E^\top$, and $A\left(E^\top\right)^{-1} = \Gamma = A\left(E^{-1}\right)^\top$. This means that each element in Γ is a linear combination of the elements in A, where the coefficients are in a column of the matrix $(E^{-1})^\top$. That is, for $0 \leq j \leq \ell - 1$,

$$\gamma^{q^{jm}} = \hat{e}_{1,j}\alpha_1 + \hat{e}_{2,j}\alpha_2 + \cdots + \hat{e}_{\ell,j}\alpha_\ell,$$

where $\hat{e}_{1,j}, \hat{e}_{2,j}, \ldots, \hat{e}_{\ell,j}$ are the elements of the j'th column of $(E^{-1})^\top$. Hence

$$\begin{aligned}\left(\gamma^{q^{jm}}\right)^{q^r} &= (\hat{e}_{1,j}\alpha_1 + \hat{e}_{2,j}\alpha_2 + \cdots + \hat{e}_{\ell,j}\alpha_\ell)^{q^r} \\ &= \hat{e}_{1,j}\alpha_1^{q^r} + \hat{e}_{2,j}\alpha_2^{q^r} + \cdots + \hat{e}_{\ell,j}\alpha_\ell^{q^r},\end{aligned}$$

where $0 \leq r \leq m - 1$. This implies that

$$\gamma^{q^{jm+r}} \in \left\langle \alpha_1^{q^r}, \alpha_2^{q^r}, \ldots, \alpha_\ell^{q^r} \right\rangle,$$

which means that $\gamma^{q^s} \in \langle Z \rangle$, for $0 \leq s \leq \ell m - 1$. Because γ is a primitive element of $\mathbb{F}$, the elements γ^{q^s} for $0 \leq s \leq \ell m - 1$ are distinct. Therefore, the elements of the set Z span $\mathbb{F}$ as a vector space over $\mathbb{F}_q$. But $\mathbb{F}$ has dimension ℓm over $\mathbb{F}_q$, and Z also contains ℓm elements. Hence Z is a basis for $\mathbb{F}$. □

4.5.1 Encoding

Let $\mathbf{u} = (u_0, \ldots, u_{k-1}) \in \mathbb{F}_q^k$ be the symbols to be transmitted. The sender forms a linearized polynomial $f(x)$, where

$$f(x) = \sum_{i=0}^{k-1} u_i x^{q^i}.$$

Similar to the one-dimensional case, we define an evaluation mapping for ℓ-dimensional MV-codes.

Definition 4.14. Let $f(x) \in \mathscr{L}_q^k[X]$ be the linearized polynomial whose coefficients are the components of the message vector $\mathbf{u}$, and let $\langle A \rangle = \langle \alpha_1, \ldots, \alpha_\ell \rangle$. Define the mapping $\mathsf{ev}_{A,L}$ as

$$\begin{aligned}&\mathsf{ev}_{A,L} : \mathscr{L}_q^k[X] \to \mathcal{P}(W, n), \text{ given by} \\ &\mathsf{ev}_{A,L}(f) = \left\langle (\alpha_i, f(\alpha_i), f^{\otimes 2}(\alpha_i), \ldots, f^{\otimes L}(\alpha_i)) \right\rangle, \qquad \text{for } i = 1, \ldots, \ell.\end{aligned}$$

We call this map evaluation of f at A.

The image of f under the mapping $\mathsf{ev}_{A,L}$ is a subspace of W, where

$$W = \langle A \rangle \oplus \underbrace{\mathbb{F}_{q^{m\ell}} \oplus \cdots \oplus \mathbb{F}_{q^{m\ell}}}_{L \text{ times}}$$
$$= \langle A \rangle \oplus \left(\bigoplus_{i=1}^{L} \mathbb{F}_{q^{m\ell}} \right).$$

This vector space W has dimension $\ell + Lm\ell$ over $\mathbb{F}_q$. A vector in W is of the form $(x, y_1, \ldots, y_L)$, where $x \in \langle A \rangle$ and $y_j \in \mathbb{F}_{q^{m\ell}}$, for $1 \leq j \leq L$.

The transmitter then encodes $f(x)$ using the evaluation mapping $\mathsf{ev}_{A,L}$. This produces a codeword V, where

$$V = \mathsf{ev}_{A,L}(f) = \left\langle \left(\alpha_i, f(\alpha_i), f^{\otimes 2}(\alpha_i), \ldots, f^{\otimes L}(\alpha_i) \right) \right\rangle, \text{ for } i = 1, \ldots, \ell.$$

Letting $v_i = (\alpha_i, f(\alpha_i), f^{\otimes 2}(\alpha_i), \ldots, f^{\otimes L}(\alpha_i))$, $i = 1, \ldots, \ell$, we can also write the codeword V as

$$V = \langle v_1, v_2, \ldots, v_\ell \rangle$$

The image of $\mathscr{L}_q^k[X]$ under $\mathsf{ev}_{A,L}$ is an MV-code, denoted $\mathcal{C}_q(k, \ell, m, L)$. It is required that the encoding is injective. As the following result shows, this happens if $k \leq \ell m$.

Proposition 4.15. *Consider the evaluation mapping $\mathsf{ev}_{A,L}$ defined in def. 4.14. If $k \leq \ell m$, then this mapping is injective.*

Proof. Let $f, g \in \mathscr{L}_q^k[X]$, with $\mathsf{ev}_{A,L}(f) = \mathsf{ev}_{A,L}(g)$. From the definition of a codeword in a MV-code, we have

$$f(\alpha_i) = g(\alpha_i), \text{ for all } \alpha_i \in A.$$

Because f and g are linearized polynomials over $\mathbb{F}_q$, we have $f(\alpha_i^{q^h}) = f(\alpha_i)^{q^h}$, and $g(\alpha_i^{q^h}) = g(\alpha_i)^{q^h}$ for all $0 \leq h \leq m-1$. Therefore, these two polynomials agree on $q^{\ell m}$ points. Let $h(x) = f(x) - g(x)$. Then h has $q^{\ell m}$ roots, but the degree of h is at most q^{k-1}. Because $k \leq \ell m$ by assumption, $h \equiv 0$, so $f \equiv g$. Therefore, $\mathsf{ev}_{A,L}$ is injective. □

The following result establishes a lower bound on the distance between two codewords in an MV-code.

Proposition 4.16. *Let $\mathcal{C}$ be an $\mathcal{C}_q(k, \ell, m, L)$ MV-code, and let U and V be two distinct codewords in this code. Then*

$$d(U, V) \geq 2 \left(\ell - \left\lceil \frac{k}{m} \right\rceil + 1 \right)$$

[Mah12].

Proof. We only prove result for list size $L = 1$. Extension to arbitrary list sizes is straightforward. Let $f, g \in \mathscr{L}_q^k[X]$ be two distinct message polynomials. The corresponding codewords are

$$U = \mathsf{ev}_{A,L}(f) = \langle (\alpha_i, f(\alpha_i)) \rangle$$
$$V = \mathsf{ev}_{A,L}(g) = \langle (\alpha_i, g(\alpha_i)) \rangle ,$$

for $1 \leq i \leq \ell$. Suppose $U \cap V$ has dimension r. Because $U \cap V \subseteq \langle \alpha_1, \ldots, \alpha_\ell \rangle \oplus \mathbb{F}_{q^{m\ell}}$, we have

$$U \cap V = \left\langle (\hat{\alpha}_1, \hat{\beta}_1), \ldots, (\hat{\alpha}_r, \hat{\beta}_r) \right\rangle ,$$

where $\{\hat{\alpha}_1, \ldots, \hat{\alpha}_r\} \subseteq \{\alpha_1, \ldots, \alpha_\ell\}$, and $\hat{\beta}_j = f(\hat{\alpha}_j) = g(\hat{\alpha}_j)$, for $1 \leq j \leq r$. Because f and g are linearized polynomials over $\mathbb{F}_q$, we have

$$f(\hat{\alpha}_j^{q^h}) = f(\hat{\alpha}_j)^{q^h}, g(\hat{\alpha}_j^{q^h}) = g(\hat{\alpha}_j)^{q^h},$$

where $0 \leq h \leq m-1$. From this, we see that f and g agree on q^{mr} points. Suppose $r \geq \left\lceil \frac{k}{m} \right\rceil$, and let $h = f - g$. Then h has q^{mr} roots, but the degree of h is at most q^{k-1}. So because

$$mr \geq m \left\lceil \frac{k}{m} \right\rceil \geq m \frac{k}{m} = k,$$

the polynomial h has more roots than its degree, and therefore, $h \equiv 0$. We then have $f \equiv g$, a contradiction to the assumption that f and g are distinct. Therefore, $r \leq \left\lceil \frac{k}{m} \right\rceil - 1$. The distance between U and V is

$$\begin{aligned} d(U,V) &= \dim(U) + \dim(V) - 2\dim(U \cap V) = 2(\ell - r) \\ &\geq 2\left(\ell - (\left\lceil \frac{k}{m} \right\rceil - 1)\right) \\ &= 2\left(\ell - \left\lceil \frac{k}{m} \right\rceil + 1\right). \end{aligned}$$

□

Now suppose the transmitter sends V over an operator channel, which does ρ erasures and t errors. Then the reciever gets a subspace $U \in \mathcal{P}(W)$, where $d = \dim(U) = \ell - \rho + t$.

4.5.2 Decoding

Here we describe the decoding of MV-codes, as defined in the previous subsection. The decoding is similar to the case where codewords have dimension one.

Finding the Interpolation Points

First we find a basis for U, denote it by

$$\{(x_i, y_{i,1}, \ldots, y_{i,L}) \mid 1 \leq i \leq d\}.$$

Then define, for $h = 0, 1, 2, \ldots, m-1$, the sets $\mathcal{P}_h$ as

$$\mathcal{P}_h = \{(x_i^{q^h}, y_{i,1}^{q^h}, \ldots, y_{i,L}^{q^h})\}$$

The set of interpolation points is the union of the $\mathcal{P}_h$'s. Denote this union by $\mathcal{P}$.

Constructing the Interpolation Polynomial

From the set $\mathcal{P}$ constructed in the previous step, define a multivariate linearized interpolation polynomial $Q(x, y_1, \ldots, y_L)$, where

$$Q(x, y_1, \ldots, y_L) = Q_0(x) + Q_1(y_1) + \cdots + Q_L(y_L),$$

where $Q(x, y_1, \ldots, y_L)$ satisfies the degree constraints

$$\begin{aligned}
&\deg(Q_0) \leq q^{\omega-1}, \\
&\deg(Q_1) \leq q^{\omega-(k-1)-1} = q^{\omega-k}, \\
&\deg(Q_2) \leq q^{\omega-2(k-1)-1} = q^{\omega-2k+1}, \\
&\qquad \vdots \\
&\deg(Q_L) \leq q^{\omega-L(k-1)-1} = q^{\omega-Lk+(L-1)},
\end{aligned}$$

where ω is a parameter to be defined later. Also, the interpolation constraint

$$Q(x, y_1, \ldots, y_L) = 0, \text{ for all } (x, y_1, \ldots, y_L) \in \mathcal{P},$$

is required to be satisfied by the interpolation polynomial.

Factorization Step

In this step, we use the Linearized Roth Ruckenstein algorithm, described in sec. 4.7 to find all roots in $\mathscr{L}_q^k[X]$ of degree at most q^{k-1} of the polynomial

$$Q(x, f(x), f^{\otimes 2}(x) \ldots, f^{\otimes L}(x)).$$

The coefficients of the polynomial $f(x)$ are then associated with the sent message.

4.5.3 Correctness of the Decoding Procedure

In this subsection, we prove that the list decoding of MV-codes described in the previous section is correct. The first results gives conditions on when there is a non-trivial interpolation polynomial.

Lemma 4.17. *By choosing*

$$\omega = \left\lceil \frac{md+1}{L+1} + \frac{1}{2}L(k-1) \right\rceil, \tag{4.9}$$

where the variables are defined as in the previous section, there exists a non-trivial multivariate linearized interpolation polynomial $Q(x, y_1, \ldots, y_L)$, satisfying the degree and interpolation constraints.

Proof. Because $\mathcal{P}$ contains dm elements, the interpolation constraint

$$Q(x, y_1, \ldots, y_L) = 0$$

corresponds to dm homogeneous equations. To count the number of unknowns, we look at the degree constraints. Then

$$\begin{aligned}\sum_{i=0}^{L}(\omega - (k-1)i) &= (L+1)\omega - (k-1)\sum_{i=0}^{L} i \\ &= (L+1)\omega - (k-1)\frac{L(L+1)}{2}.\end{aligned}$$

As long as a homogeneous system has fewer equations than unknowns, there exitsts a non-trivial solution. Therefore, if

$$md < (L+1)\omega - (k-1)\frac{L(L+1)}{2}$$

holds, then there is a non-trivial solution. Because all variables in the above equation are integers, we can add 1 to the left hand side to relax the strictness of the inequality

$$md + 1 \leq (L+1)\omega - (k-1)\frac{L(L+1)}{2} \tag{4.10}$$

Eq. (4.10) can be rearranged to

$$\frac{md+1}{L+1} + \frac{(k-1)L}{2} \leq \omega.$$

Therefore, the value of ω from eq. (4.9) guarantees the existence of a non-trivial interpolation polynomial. □

Lemma 4.18. *The span of $\mathcal{P}_h$ over $\mathbb{F}_q$,*

$$\langle \mathcal{P}_h \rangle = \left\langle (x_i^{q^h}, y_{i,1}^{q^h}, \ldots, y_{i,L}^{q^h}) \right\rangle,$$

where $i = 1, 2, \ldots, d$ and $h = 0, 1, 2, \ldots, m-1$, are disjoint.

Proof. We consider the first component x_i of an element in the span of $\mathcal{P}$. For $i = 1, 2, \ldots, d$,

$$x_i \in \langle \alpha_1, \alpha_2, \ldots, \alpha_\ell \rangle,$$

and because raising a field element to the q^h'th power is a linear operation, we have

$$x_i^{q^h} \in \left\langle \alpha_1^{q^h}, \alpha_2^{q^h}, \ldots, \alpha_\ell^{q^h} \right\rangle.$$

From lemma 4.13, these spans are disjoint as h varies. In other words, for $h \neq k$, the spans

$$\left\langle x_i^{q^h} \right\rangle \text{ and } \left\langle x_i^{q^k} \right\rangle$$

are disjoint. Therefore, the spans of the $\mathcal{P}_h$'s are also disjoint. □

In the next two results, it is shown that the list returned by the list decoding algorithm is of size at most L and contains the transmitted codeword. Let $Q(x, y_1, \ldots, y_L)$ be the interpolation polynomial found in the second step of the list decoding algorithm, and let $f(x)$ be the message polynomial. As in the one-dimensional case, define the polynomial $E(x)$ as

$$E(x) = Q(x, f(x), f^{\otimes 2}(x), \ldots, f^{\otimes L}(x)). \tag{4.11}$$

Lemma 4.19. *The linearized polynomial $E(x)$ has at least $(\ell - \rho)m$ linearly independent roots in $\mathbb{F}$.*

Proof. Let $V \in \mathcal{C}$ be the sent codeword, and $U \in \mathcal{P}(W)$ the received subspace. Let $U' = U \cap V$. Then $U' \subseteq U$, and $\dim(U') = \ell - \rho$. For any

$$(x, y_1, \ldots, y_L) \in U',$$

and any $h = 0, 1, \ldots, m-1$, we have

$$(x^{q^h}, y_1^{q^h}, \ldots, y_L^{q^h}) \in \langle \mathcal{P}_h \rangle.$$

This is because all basis elements of U' are in the span of $\mathcal{P}_h$, for some $h = 0, 1, \ldots, m-1$. Because $Q(x, y_1, \ldots, y_L)$ is a linearized polynomial,

$$Q(x^{q^h}, y_1^{q^h}, \ldots, y_L^{q^h}) = 0.$$

But since $(x, y_1, \ldots, y_L) \in U'$, $(x, y_1, \ldots, y_L) \in V$. Therefore, we can write this element as $(\beta, f(\beta), \ldots, f^{\otimes L}(\beta))$, where $\beta \in \langle \alpha_1, \ldots, \alpha_\ell \rangle$ and $f(x) \in \mathscr{L}_q^k[X]$. Raising to the q^h'th power, we have

$$(x^{q^h}, y_1^{q^h}, \ldots, y_L^{q^h}) = (\beta^{q^h}, f(\beta)^{q^h}, \ldots, f^{\otimes L}(\beta)^{q^h}).$$

Because $f(x) \in \mathscr{L}_q^k[X]$, $f(x)^{q^h} = f(x^{q^h})$, and so

$$(x^{q^h}, y_1^{q^h}, \ldots, y_L^{q^h}) = (\beta^{q^h}, f(\beta^{q^h}), \ldots, f^{\otimes L}(\beta^{q^h})).$$

From this, we see that because $\dim(U') = \ell - \rho$, and the $\mathcal{P}_h$'s are disjoint, there are at least $(\ell - \rho)m$ linearly independent roots of $E(x)$. □

Corollary 4.20. *If $\omega < (\ell - \rho)m$, then $E(x)$ is identically zero.*

Proof. Because $f(x) \in \mathscr{L}_q^k[X]$, $\deg(f) \leq q^{k-1}$, and so $\deg(f^{\otimes i}) \leq q^{(k-1)i}$. Also

$$\begin{aligned} \deg(Q_i \otimes f^{\otimes i}) &\leq q^{\omega-(k-1)i-1} q^{(k-1)i} \\ &= q^{\omega-1}, \end{aligned}$$

which implies that $\deg(E(x)) \leq q^{\omega-1}$. By lemma 4.19, $E(x)$ has at least $(\ell - \rho)m$ linearly independent roots. But the degree of $E(x)$ is at most $q^{\omega-1}$ from the previous calculations. Therefore, from the choice of ω, $E(x)$ has more roots than its degree, so $E(x)$ must be identically zero. □

Theorem 4.21. *Given a subspace $U \in \mathcal{P}(W)$ of dimension $\ell - \rho + t$, the list decoding algorithm described in this section outputs a list of codewords in $\mathcal{C}$ of size at most L, and this list contains the transmitted codeword, provided that the inequality*

$$L\rho + t \leq \ell L - \frac{L(L+1)}{2} \frac{(k-1)}{m} - \frac{1}{m} \tag{4.12}$$

holds.

Proof. From lemma 4.17, a non-trivial linearized interpolation polynomial $Q(x, y_1, \ldots, y_L)$ exists as long as

$$\omega \geq \frac{md+1}{L+1} + \frac{1}{2} L(k-1). \tag{4.13}$$

Using corollary 4.20, if

$$\frac{md+1}{L+1} + \frac{1}{2} L(k-1) \leq (\ell - \rho)m, \tag{4.14}$$

then $E(x)$ as defined in eq. (4.11) is identically zero. Then, as $d = \dim(U) = \ell - \rho + t$,

$$\frac{m(\ell - \rho + t) + 1}{L+1} + \frac{1}{2} L(k-1) \leq (\ell - \rho)m,$$

which can be rearranged to eq. (4.12). □

4.5.4 Rate of MV-codes

Recall from sec. 4.3 that the packet rate of a subspace code is defined as

$$R^* = \frac{\log_{q^m}(|\mathcal{C}|)}{\ell} = \frac{\log_q(|\mathcal{C}|)}{\ell m}.$$

From the degree constraint, we have the inequality

$$\ell m - (k-1)L - 1 \geq 0$$

which can be rearranged to

$$\frac{\ell m - 1}{k-1} \geq L.$$

For large k and m, the above can be approximated by $\frac{\ell m}{k} \geq L$, that is $\frac{1}{R^*} \geq L$. Then we have the upper bound on the packet rate

$$R^* \leq \frac{1}{L}.$$

From thm 4.21, the decoding radius is

$$\tau = \ell L - \frac{L(L+1)}{2}\frac{(k-1)}{m} - \frac{1}{m},$$

and normalizing by ℓ, the codeword dimension, the normalized decoding radius is

$$\tau_L = L - \frac{L(L+1)}{2}\frac{(k-1)}{\ell m} - \frac{1}{\ell m}, \tag{4.15}$$

which can be approximated by $L - \frac{L(L+1)}{2}R^*$. The KK-codes have packet rate

$$\frac{\log_{q^m}(q^{km})}{\ell} = \frac{k}{\ell}. \tag{4.16}$$

Also, the list-1 decoder given in [KK08] yields a normalized decoding radius of $\tau_{KK} = \frac{\ell-k+1}{\ell}$, which is approximately equal to $1 - \frac{k}{\ell}$. Therefore, we can express the normalized decoding radius as $\tau_{KK} = 1 - R^*$, which is also equal to τ_L, when $L = 1$. For an MV-code with parameters $\mathcal{C}_q(k, \ell, m, L)$, the packet rate is

$$\frac{\log_{q^m}((q^m)^{\frac{k}{m}})}{\ell} = \frac{k}{\ell m}, \tag{4.17}$$

and the one-dimensional MV-codes have packet rate $\frac{k}{m}$. As the list size increases, the normalized decoding radius is only higher than one for low packet rates. For example, the case $L = 2$, list decoding is only beneficial for rates less than $\frac{1}{3}$. In [MV11], the authors use the concept of multiplicity

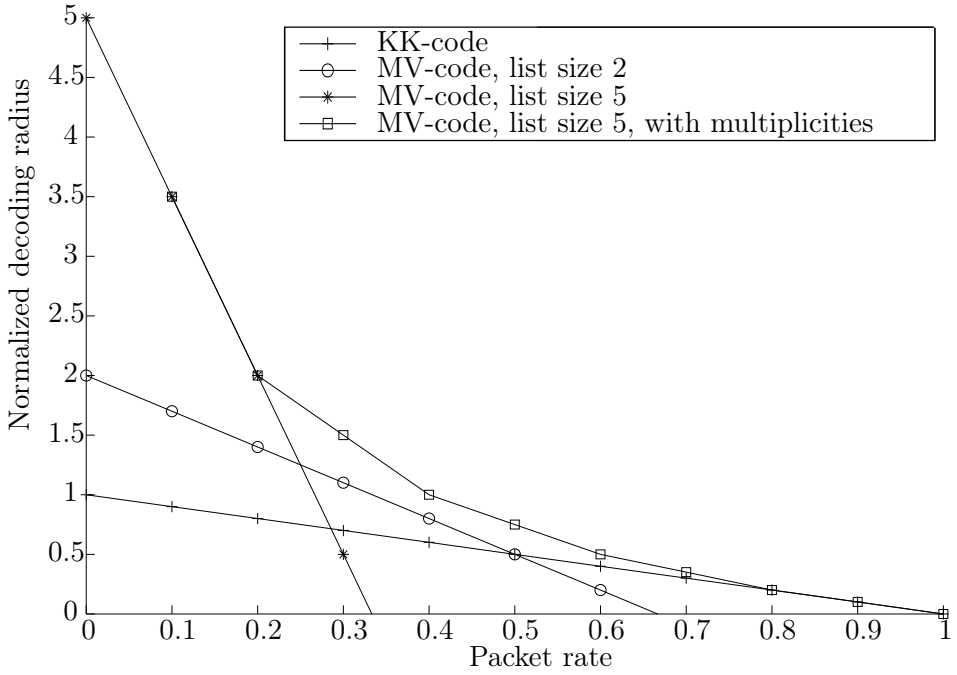

Figure 4.1: Rate comparison for KK- and MV-codes

of a root of a linearized polynomial to get a better decoding radius of MV-codes for high packet rates. More specifically, the authors define a mapping from $\mathscr{L}_q[X]$ to $\mathbb{F}_q[X]$, which is shown to be a ring-isomorphism. Using the notion of multiplicity in $\mathbb{F}_q[X]$, along with defining the formal derivative of a linearized polynomial, the decoding radius is increased, compared to using the MV-codes described in this chapter. For list size L, packet rate R^* and multiplicity r, the normalized decoding radius is then shown to be

$$\frac{2(L+1)}{r+1} - 1 - \frac{L(L+1)}{r(r+1)}R^*,$$

and the value of r that maximizes this expression is $\lceil LR^* \rceil$ [MV11]. The normalized decoding radius of the codes considered is shown in fig. 4.1.

4.6 Interpolation over Modules

Here we will look at an algorithm for dealing with the interpolation problem in decoding subspace codes. The material in this section makes use of the theory of modules in sec. 2.5. This algorihm was devised in [XYS11], on which this section is based.

Let $B = \{b_0, b_1, \ldots, b_L\}$ be an arbitrary set. Using thm. 2.28, we can construct a free $\mathscr{L}_{q^m}[X]$-module V with a basis $B = \{b_0, b_1, \ldots, b_L\}$. Recall that the ring $\mathscr{L}_{q^m}[X]$ is non-commutative with unity x. Then any element in V can be represented as

$$\begin{aligned} Q &= \ell_0(x) \otimes b_0 + \ell_1(x) \otimes b_1 + \cdots + \ell_L(x) \otimes b_L \\ &= \sum_{j=0}^{L} \ell_j(x) \otimes b_j, \end{aligned}$$

where $\ell_j(x) \in \mathscr{L}_{q^m}[X]$. Because we can write $\ell_j(x)$ as

$$\ell_j(x) = \sum_{i \geq 0} a_{i,j} x^{[i]},$$

the polynomial Q can be written as

$$Q = \sum_{j=0}^{L} \sum_{i \geq 0} a_{i,j} x^{[i]} \otimes b_j.$$

From the above, we see that we can write any $Q \in V$ as a $\mathbb{F}_{q^m}$-linear combination of elements of the form $x^{[i]} \otimes b_j$. Therefore, the set

$$M = \{x^{[i]} \otimes b_j \mid i \geq 0, 0 \leq j \leq L\}$$

is a basis for V as a vector space over $\mathbb{F}_{q^m}$.

Suppose there is a total ordering $\prec$ on M. That is, suppose that for any $m_i, m_j \in M$, we have that if $i < j$, then $m_i \prec m_j$. We can then represent any element $Q \in V$ as

$$\begin{aligned} Q &= a_0 m_0 + \cdots + a_J m_J \\ &= \sum_{j=0}^{J} a_j m_j, \end{aligned}$$

where $a_j \in \mathbb{F}_{q^m}$ and $m_j \in M$. The integer J is called the order of Q, or $\mathrm{ord}(Q)$ for short. The monomial m_J is called the leading monomial of Q, denoted $\mathrm{lm}_{\prec}(Q)$.

Using the ordering $\prec$, we can define an ordering on the elements in V. Let $Q, Q' \in V$. Then we say that

$$\begin{aligned} Q \prec_O Q' &\text{ if } \mathrm{ord}(Q) < \mathrm{ord}(Q'), \\ Q =_O Q' &\text{ if } \mathrm{ord}(Q) = \mathrm{ord}(Q'). \end{aligned}$$

We call an element $Q \in V$ minimum of V if $Q \prec_O Q'$ for all $Q' \in V \backslash \{Q\}$. We also define $\mathrm{ind}(\ell(x) \otimes b_j) = j$. For a $Q \in V$, we define $\mathrm{ind}(Q) = \mathrm{ind}(\mathrm{lm}_{\prec}(Q))$.

This last definition makes sense, because we can order the monomials in a given polynomial Q. Because all $Q \in V$ have their own unique index $\text{ind}(Q)$, we can define the set S_j as

$$S_j = \{Q \in V \mid \text{ind}(Q) = j\}.$$

From this, we can partition V according to the index, that is,

$$V = \bigcup_j S_j.$$

Remark 4.22. Let $Q(x, y_1, \ldots, y_L)$ be a multivariate linearized polynomial, and $x^{[i]} \in \mathscr{L}_{q^m}[X]$, for $i \geq 0$. If $\text{ord}(Q) = j$, then $\text{ord}(x^{[i]} \otimes Q) = j + i$. This is because

$$\begin{aligned} x^{[i]} \otimes Q &= x^{[i]} \otimes (Q_0(x) + Q_1(y_1) + \cdots + Q_L(y_L)) \\ &= (Q_0(x))^{[i]} + (Q_1(y_1))^{[i]} + \cdots + (Q_L(y_L))^{[i]}. \end{aligned}$$

The above also shows that if $\text{ind}(Q) = k$, then $\text{ind}(x^{[i]} \otimes Q) = k$.

From the above discussion, V is a vector space over $\mathbb{F}_{q^m}$. From section 2.5, a linear function from a vector space V to the underlying field of V is called a linear functional. In our case, we want to define mappings which evaluate an interpolation polynomial $Q \in V$ in the respective interpolation points.

Let C be the number of interpolation points, and define C linear functionals from V to $\mathbb{F}_{q^m}$. The k'th linear functional evaluates polynomials $Q \in V$ in the k'th interpolation point. So if we let (x_k, y_k) be the k'th interpolation point, then the corresponding linear functional is

$$D_k : V \to \mathbb{F}_{q^m} \quad D_k(Q) = Q(x_k, y_k).$$

If there are $L + 1$ basis elements for the module, then the k'th interpolation points is of the form $(x_k, y_{1,k}, \ldots, y_{L,k})$.

Now, the kernel for D_k is $\ker(D_k) = \{Q \in V \mid Q((x_k, y_k)) = 0\}$, so if $Q(x_k, y_k) = 0$, then $Q \in \ker(D_k)$, which we denote K_k. For a polynomial $Q \in V$ to interpolate all C points, we need $Q \in K_j$, for $j = 1, \ldots, C$. In other words, we need

$$Q \in K_1 \cap K_2 \cap \cdots \cap K_C,$$

and we denote this intersection $\overline{K}_C$. Furthermore, for an integer $i \leq C$, we denote

$$\overline{K}_i = K_1 \cap K_2 \cap \cdots \cap K_i.$$

We show that the kernel K_i is a $\mathscr{L}_{q^m}[X]$-submodule of V. The result is for the case of two basis elements $B = \{x, y\}$, extension of the result to a higher number of basis elements is straightforward.

Proposition 4.23. *Let V be a free $\mathscr{L}_{q^m}[X]$-module with basis $\{x, y\}$. The kernel K_i of the mapping D_i is a $\mathscr{L}_{q^m}[X]$-submodule of V.*

Proof. Because K_i is a subgroup of V, and V is abelian, so is K_i. We now show that K_i is closed under the action $\otimes$. Let $\ell(x) \in \mathscr{L}_{q^m}[X]$, and $Q(x, y) \in K_i$, where

$$\ell(x) = \sum_{i=0}^{n} a_i x^{[i]}, \quad Q(x, y) = Q_0(x) + Q_1(y),$$

and $Q_0, Q_1 \in \mathscr{L}_{q^m}[X]$. Then

$$\begin{aligned} \ell(x) \otimes Q(x, y) &= \sum_{i=0}^{n} a_i x^{[i]} \otimes (Q_0(x) + Q_1(y)) \\ &= \sum_{i=0}^{n} a_i \left(Q_0(x) + Q_1(y)\right)^{[i]}. \end{aligned}$$

But $D_i(Q(x, y)) = Q_0(x_i) + Q_1(y_i) = 0$. Hence

$$D_i\left(\ell(x) \otimes Q(x, y)\right) = \sum_{i=0}^{n} a_i 0^{[i]} = 0.$$

Therefore, $\ell(x) \otimes Q(x, y) \in K_i$, so K_i is a $\mathscr{L}_{q^m}[X]$-submodule of V. □

Given a set of C interpolation points, we want the polynomial $Q^* \in V$ which interpolates all these points, i.e. we want $Q^* \in \overline{K}_C$. Furthermore, we want Q^* to be minimal (with respect to $\prec_O$) among all polynomials $Q \in \overline{K}_C$.

Proposition 4.24. *The minimal element $Q \in \overline{K}_C$ is unique up to multiplication by elements in $\mathbb{F}_{q^m}$.*

Proof. Suppose $Q^*, Q' \in \overline{K}_C$ are minimal with respect to $\prec_O$. Then, for suitable choice of $\alpha, \beta \in \mathbb{F}_{q^m}$, the polynomial $Q = \alpha Q^* + \beta Q'$ precedes both Q^* and Q' under $\prec_O$. That is, we have $Q \prec_O Q^* =_O Q'$, a contradiction to the minimality of Q^* and Q'. □

Consider those polynomials $Q \in V$ which interpolate the first i interpolation points, that is, $Q \in \overline{K}_i$. Further, consider those which have index $\mathrm{ind}(Q) = j$. The set of all polynomial satisfying these two conditions is denoted $T_{i,j}$, that is,

$$T_{i,j} = \overline{K}_i \cap S_j.$$

Also, for the polynomials in $T_{i,j}$, define the minimal one (with respect to the ordering $\prec_O$) by $g_{i,j}$, so

$$g_{i,j} = \min_{g \in T_{i,j}} g.$$

Note that from prop. 4.24, $g_{i,j}$ is unique.

4.6.1 Interpolation Algorithm by Linearized Polynomials

Algorithm 3 General Interpolation by Linearized Polynomials

Input: Set of C interpolation points, basis $B = \{b_0, \ldots, b_L\}$ of an $\mathscr{L}_{q^m}[X]$-module, ordering $\prec$ on monomials in M.
Output: Linearized polynomial Q over $\mathbb{F}_{q^m}$, which is zero on the interpolation points and minimal with respect to $\prec_O$.

for $j = 0$ to L **do**
  $g_{0,j} \leftarrow b_j$
end for
for $i = 0$ to $C - 1$ **do**
  for $j = 0$ to L **do**
    $g_{i+1,j} \leftarrow g_{i,j}$
    $\Delta_{i+1,j} \leftarrow D_{i+1}(g_{i,j})$
  end for
  $J \leftarrow \{j \mid \Delta_{i+1,j} \neq 0\}$
  if $J \neq \emptyset$ **then**
    $j^* \leftarrow \arg\min_{j \in J}\{g_{i,j}\}$
    for $j \in J$ **do**
      if $j \neq j^*$ **then**
        $g_{i+1,j} \leftarrow \Delta_{i+1,j^*} g_{i,j} - \Delta_{i+1,j} g_{i,j^*}$
      else if $j = j^*$ **then**
        $g_{i+1,j} \leftarrow \Delta_{i+1,j}(x^{[1]} \otimes g_{i,j}) - D_{i+1}(x^{[1]} \otimes g_{i,j}) g_{i,j}$
      end if
    end for
  end if
end for
$Q^* \leftarrow \min_j g_{C,j}$
return Q^*

We now describe the algorithm, shown in alg. 3. Firstly, it iterates through the basis B for the module V. For the case of KK-codes, the basis is $B = \{x, y\}$ while for the MV-codes, the basis is $B = \{x, y_1, \ldots, y_L\}$, where L is the list size. The algorithm then iterates through the C interpolation points. Then each basis element in B is evaluated in each interpolation point.

Now, we want to update $g_{i+1,j}$ from $g_{i,j}$, where the order is fixed. The updated polynomial $g_{i+1,j}$ is required to interpolate the first $i+1$ interpolation points, and have index j. There are three different cases for updating $g_{i+1,j}$ from $g_{i,j}$.

1. If $g_{i,j} \in K_{i+1}$, then set $g_{i+1,j} := g_{i,j}$.

2. If $g_{i,j} \notin K_{i+1}$, then we consider all $g_{i,j}$'s which are not in K_{i+1}. Let g_{i,j^*} be the one with lowest order, which, by lemma 4.24, is unique. Note that the $g_{i,j}$'s under consideration are in K_i. For the elements $g_{i,j} \neq_O g_{i,j^*}$, set

$$g_{i+1,j} := D_{i+1}(g_{i,j^*})g_{i,j} - D_{i+1}(g_{i,j})g_{i,j^*}.$$

Then $\text{ord}(g_{i+1,j}) = \text{ord}(D_{i+1}(g_{i,j^*})g_{i,j} - D_{i+1}(g_{i,j})g_{i,j^*}) = \text{ord}(g_{i,j})$, so $g_{i+1,j} =_O g_{i,j}$. Note that $D_{i+1}(g_{i,j^*})$ and $D_{i+1}(g_{i,j})$ are not equal to zero.

3. From g_{i,j^*}, the element g_{i+1,j^*} is constructed as

$$g_{i+1,j^*} = D_{i+1}(g_{i,j^*})(x^{[1]} \otimes g_{i,j^*}) - D_{i+1}(x^{[1]} \otimes g_{i,j^*})g_{i,j^*}.$$

In this case, the order increases, that is $g_{i,j^*} \prec_O g_{i+1,j^*}$ (see remark 4.22).

Finally, the algorithm returns the polynomial Q with lowest order, and which interpolates the C interpolation points.

Proposition 4.25. *In all three cases listed above, the constructed $g_{i+1,j}$ is a minimum (with respect to $\prec_O$) in $T_{i+1,j}$.*

Proof. The proof is by induction on i. For the base case $i = 0$, the statement is trivially true. Suppose the statement holds for some i. To show that this implies that the statement holds for $i + 1$, there are three cases.

Consider first case 1. If $g_{i,j} \in \overline{K}_{i+1}$, then we set $g_{i+1,j} := g_{i,j}$. This implies that $g_{i+1,j} \in T_{i+1,j}$. But $T_{i+1,j} \subseteq T_{i,j}$, and therefore, because $g_{i,j}$ is a minimum in $T_{i+1,j}$, so is $g_{i+1,j}$.

For case 2, we have $g_{i,j^*} \neq_O g_{i,j}$. Because the minimum is unique by lemma 4.24, we have $g_{i,j^*} \prec_O g_{i,j}$.Then the constructed $g_{i+1,j}$ interpolates the $(i + 1)$'th point, since

$$\begin{aligned} D_{i+1}(g_{i+1,j}) &= D_{i+1}\left(D_{i+1}(g_{i,j^*})g_{i,j} - D_{i+1}(g_{i,j})g_{i,j^*}\right) \\ &= D_{i+1}(g_{i,j^*})D_{i+1}(g_{i,j}) - D_{i+1}(g_{i,j})D_{i+1}(g_{i,j^*}) \\ &= 0, \end{aligned}$$

because D_{i+1} is a linear map. This implies that $g_{i+1,j} \in \ker(D_{i+1}) = K_{i+1}$. Also, $\text{ind}(g_{i+1,j}) = \text{ind}(g_{i,j})$, from the way $g_{i+1,j}$ is defined. Therefore, $g_{i+1,j} \in S_j$, and so $g_{i+1,j} \in T_{i+1,j}$. Because $g_{i,j}, g_{i,j^*} \in \overline{K}_i$, $D_k(g_{i+1,j}) = 0$, for all $k \leq i$, and so $g_{i+1,j} \in T_{i,j}$. From the construction of $g_{i+1,j}$, $g_{i+1,j} =_O$

$g_{i,j}$. So $g_{i+1,j}$ is a minimum in $T_{i,j}$, and also a minimum in $T_{i+1,j}$, because $T_{i+1,j} \subseteq T_{i,j}$.

For case 3, we have

$$g_{i+1,j^*} = D_{i+1}(g_{i,j^*})(x^{[1]} \otimes g_{i,j^*}) - D_{i+1}(x^{[1]} \otimes g_{i,j^*})g_{i,j^*}.$$

As in case 2,

$$D_{i+1}(g_{i+1,j^*}) = D_{i+1}(g_{i,j^*})D_{i+1}(x^{[1]} \otimes g_{i,j^*}) - D_{i+1}(x^{[1]} \otimes g_{i,j^*})D_{i+1}(g_{i,j^*}) = 0,$$

and so $g_{i+1,j^*} \in K_{i+1}$. For $k \leq i$,

$$D_k(g_{i+1,j^*}) = D_{i+1}(g_{i,j^*})D_k(x^{[1]} \otimes g_{i,j^*}) - D_{i+1}(x^{[1]} \otimes g_{i,j^*})D_k(g_{i,j^*}) = 0,$$

which follows because K_i is a $\mathscr{L}_{q^m}[X]$-module (prop. 4.23), so because $g_{i,j}, g_{i,j^*} \in K_i$ and $x^{[i]} \in \mathscr{L}_{q^m}[X]$, then $x^{[i]} \otimes g_{i,j^*} \in K_i$, and therefore we also have $x^{[i]} \otimes g_{i,j^*} \in \overline{K}_i$. Also, $\overline{K}_i \subseteq K_i$, which implies that $g_{i+1,j^*} \in K_k$, for $k \leq i$, and so $g_{i+1,j^*} \in \overline{K}_i$.

Turning to the index of g_{i+1,j^*}, we have

$$\text{ind}(g_{i+1,j^*}) = \text{ind}(x^{[1]} \otimes g_{i,j^*}) = j^*,$$

because in $x^{[1]} \otimes g_{i,j^*}$, the degree of all terms in g_{i,j^*} is increased by $[1] = q$. This implies that $g_{i+1,j^*} \in K_{i+1} \cap S_{j^*} = T_{i+1,j^*}$. We show that g_{i+1,j^*} is minimal in T_{i+1,j^*}. This is done by contradiction. Suppose $f_{i+1,j^*} \in T_{i+1,j^*}$, with $f_{i+1,j^*} \prec_O g_{i+1,j^*}$. Because $T_{i+1,j^*} \subseteq T_{i,j^*}$, $f_{i+1,j^*} \in T_{i,j^*}$. But g_{i,j^*} is minimal in T_{i,j^*}, and so $\text{ord}(g_{i,j^*}) \leq \text{ord}(f_{i+1,j^*})$. From this,

$$\text{ord}(g_{i,j^*}) \leq \text{ord}(f_{i+1,j^*}) < \text{ord}(x^{[1]} \otimes g_{i,j^*}).$$

This is because if $\text{ord}(f_{i+1,j^*}) = \text{ord}(x^{[1]} \otimes g_{i,j^*})$, then $\text{ord}(f_{i+1,j^*}) = \text{ord}(g_{i+1,j^*})$, since $\text{ord}(g_{i+1,j^*}) = \text{ord}(x^{[1]} \otimes g_{i,j^*})$ which in turn contradicts the assumption that $f_{i+1,j^*} \prec_O g_{i+1,j^*}$. Since $\text{ind}(g_{i,j^*}) = \text{ind}(x^{[1]} \otimes g_{i,j^*})$, g_{i,j^*} and $x^{[1]} \otimes g_{i,j^*}$ are in S_{j^*}. Therefore, there is no $f_{i+1,j^*} \in S_{j^*}$ satisfying

$$\text{ord}(g_{i,j^*}) < \text{ord}(f_{i+1,j^*}) < \text{ord}(x^{[1]} \otimes g_{i,j^*}).$$

So we must have $f_{i+1,j^*} =_O g_{i,j^*}$. But then, using lemma 4.24, we can form the polynomial h as

$$h = \alpha f_{i+1,j^*} + \beta g_{i,j^*},$$

where $\alpha, \beta \in \mathbb{F}_{q^m}$, hence $h \prec_O g_{i,j^*}$. Also, because $f_{i+1,j^*}, g_{i,j^*} \in \overline{K}_i$, $h \in \overline{K}_i$. But $g_{i,j^*} \notin T_{i+1,j^*}$, so $h \notin \overline{K}_{i+1}$. We now have $h \in \overline{K}_i \backslash \overline{K}_{i+1}$, and $h \prec_O g_{i,j^*}$. However, this contradicts the assumption that g_{i,j^*} is minimal among all elements in $\overline{K}_i \backslash \overline{K}_{i+1}$. Therefore, g_{i+1,j^*} is a minimal element in T_{i+1,j^*}. $\square$

4.6.2 Interpolation Procedure in Decoding KK-codes

In this section, we use the theory described to find a bivariate linearized polynomial, which interpolates a set of points. Consider the set $\{x, y\}$. From this we construct a free $\mathscr{L}_{q^m}[X]$-module V with basis $\{x, y\}$, which by thm. 2.28 can be done. The elements Q in V are of the form

$$Q(x, y) = \ell_0(x) \otimes x + \ell_1(x) \otimes y.$$

Defining the operation $\otimes$ as $\ell(x) \otimes b = \ell(b)$, for $b \in \{x, y\}$, we can write $Q(x, y) = \ell_0(x) + \ell_1(y)$. The set V is also a vector space over $\mathbb{F}_{q^m}$, where

$$\begin{aligned} M &= \{x^{[i]} \otimes x, x^{[j]} \otimes y \mid i, j \geq 0\} \\ &= \{x^{[i]}, y^{[j]} \mid i, j \geq 0\} \end{aligned}$$

is a basis for V. We now define an ordering on M. If $i \geq 0$, then

$$\begin{aligned} &x^{[i]} \prec x^{[i+1]},\ y^{[i]} \prec y^{[i+1]}, \\ &x^{[i+k-1]} \prec y^{[i]} \prec x^{[i+k]}, \end{aligned} \tag{4.18}$$

where k is the number of information symbols.

Suppose there are n interpolation points (x_i, y_i), $i = 0, 1, \ldots, n-1$. The n linear functionals D_i are then defined as

$$D_i(Q) = Q(x_i, y_i).$$

An element Q is in the kernel of D_i, denoted K_i, if $Q(x_i, y_i) = 0$. From prop. 4.23, the kernel K_i is a $\mathscr{L}_{q^m}[X]$-submodule of V.

We now show that when $B = \{x, y\}$, the interpolation algorithm (alg. 3) reduces to the list-1 interpolation algorithm of KK-codes (alg. 2). The ordering $\prec$ defined as in (4.18) corresponds to the ordering induced by the $(1, k-1)$-degree comparison given in sec. 3.7.2. In the notation of this section, the polynomial $g_{i,0}$ is the polynomial which interpolates through the first i points, and its leading monomial (under $\prec$) is a power of x. Similarly, $g_{i,1}$ interpolates through the first i points, and its leading monomial is a power of y. In the terminology of sec. 3.7.2, $g_{i,0}$ is x-minimal and interpolates the first i points, while $g_{i,1}$ is y-minimal and interpolates the first i points.

Initially, the algorithm sets $g_{0,0} = x$ and $g_{0,1} = y$. For updating, the algorithm iterates through the C interpolation points, and sets

$$\begin{aligned} g_{i+1,0} &\leftarrow g_{i,0}, \quad \Delta_{i+1,0} \leftarrow D_{i+1}(g_{i,0}), \\ g_{i+1,1} &\leftarrow g_{i,1}, \quad \Delta_{i+1,1} \leftarrow D_{i+1}(g_{i,1}), \end{aligned}$$

where $0 \leq i \leq C-1$. The set J then consists of those indices where polynomials $g_{i,j}$, $j \in \{0,1\}$ evaluate to non-zero on the $(i+1)$'th interpolation point. There are three cases. The first case is $J = \{0\}$, which corresponds to the polynomial which evaluates to a non-zero value has leading monomial $x^{[k]}$, for some $k \geq 0$. Then the line

$$g_{i+1,0} \leftarrow \Delta_{i+1,0}\left(x^{[1]} \otimes g_{i,0}\right) - D_{i+1}\left(x^{[1]} \otimes g_{i,0}\right) g_{i,0}$$

is executed. This line corresponds to the line

$$f_0' \leftarrow f_0^q(x,y) - \Delta_0^{q-1} f_0(x,y)$$

in the KK case. Note that multiplying the right hand side in the above line by Δ_0, which is non-zero, we get

$$f_0''(x,y) \leftarrow \Delta_0 f_0^q(x,y) - \Delta_0^q f_0(x,y)$$

and this does not affect the result, that is, we can use either f_0' or f_0''. This is because the index is unchanged, as is the order. For the second case, we have $J = \{1\}$, which means that the polynomial has leading monomial $y^{[k]}$, $k \geq 0$. Then a reasoning similar to the first case applies.

For the case $J = \{0,1\}$, the algorithm finds the index j^* for which $g_{i,j}$ is smallest. This corresponds to comparing the $(1, k-1)$-degrees of $f_0(x,y)$ and $f_1(x,y)$. If $j^* = 0$, then this corresponds to $\deg_{1,k-1}(f_0) \leq \deg_{1,k-1}(f_1)$ in the KK algorithm. The lines

$$\begin{aligned} g_{i+1,1} &\leftarrow \Delta_{i,0} g_{i,1} - \Delta_{i+1,1} g_{i,0}, \\ g_{i+1,0} &\leftarrow \Delta_{i+1,0}\left(x^{[1]} \otimes g_{i,0}\right) - D_{i+1}\left(x^{[1]} \otimes g_{i,0}\right) g_{i,0} \end{aligned}$$

are then executed. This corresponds to the lines

$$\begin{aligned} f_1(x,y) &\leftarrow \Delta_1 f_0(x,y) - \Delta_0 f_1(x,y) \\ f_0(x,y) &\leftarrow f_0^q(x,y) - \Delta_0^{q-1} f_0(x,y) \end{aligned}$$

in the KK algorithm. Similarly, if $j^* = 1$, then $\deg_{1,k-1}(f_1) \leq \deg_{1,k-1}(f_0)$, and updating is then done accordingly.

Finally, the algorithm compares the $(1, k-1)$-degrees of $f_0(x,y)$ and $f_1(x,y)$. This corresponds to the statement $Q^* \leftarrow \min_j g_{C,j}$.

We conclude that in the case of $B = \{x, y\}$, the interpolation algorithm in alg. 3 reduces to the interpolation algorithm of KK-codes in alg. 2.

4.6.3 Interpolation Procedure in Decoding MV-codes

In this section, we discuss the application of algorithm 3 for solving the interpolation problem of MV-codes. We consider an MV-code with codeword dimension ℓ, with message vector of length k and list size L. The set $B = \{x, y_1, \ldots, y_L\}$ is chosen to form a free $\mathscr{L}_{q^{m\ell}}[X]$-module V. This module consists of multivariate linearized polynomials of the form $Q(x, y_1, \ldots, y_L)$. The action $\otimes$ is defined as

$$\ell(x) \otimes b_j = \ell(b_j)$$

for $\ell(x) \in \mathscr{L}_{q^{m\ell}}[X]$ and $b_j \in B$. An element $Q \in V$ can then be written as

$$Q(x, y_1, \ldots, y_L) = Q_0(x) + Q_1(y_1) + \cdots + Q_L(y_L),$$

where $Q_i(x) \in \mathscr{L}_{q^{m\ell}}[X]$. The set

$$\begin{aligned} M &= \{x^{[i]} \otimes b_j \mid i \geq 0, 0 \leq j \leq L\} \\ &= \{x^{[i_0]} \otimes x, x^{[i_1]} \otimes y_1, \ldots, x^{[i_L]} \otimes y_L \mid 0 \leq i_k\} \\ &= \{x^{[i_0]}, y_1^{[i_1]}, \ldots, y_L^{[i_L]} \mid 0 \leq i_k\} \end{aligned}$$

is a basis for V as a vector space over $\mathbb{F}_{q^{m\ell}}$. Then an ordering $\prec$ is defined on M as follows. For $0 \leq j \leq L$,

$$b_j^{[i]} \prec b_j^{[i+1]},$$

and if $0 \leq j < j' \leq L$ and $j(k-1) + i = (j+1)(k-1) + i'$, for $i < i'$, then

$$b_j^{[i]} \prec b_{j'}^{[i']}.$$

In the first case, the same basis element is compared for different exponents, while the second case deals with different basis elements. It can be shown that $\prec$ is a total order on M.

In the initial step of the algorithm, the $L+1$ polynomials are set as

$$g_{0,0} = x, g_{0,1} = y_1, \ldots, g_{0,L} = y_L.$$

For evaluating the interpolation polynomial in the given points, $(\ell + t)m$ linear functionals $D_i : V \to \mathbb{F}_{q^{m\ell}}$ are defined, where t is the dimension of the error space. The i'th linear functional has kernel K_i. From this, we want a non-zero polynomial $Q \in V$, which interpolates the $(\ell + t)m$ points. This is the same as requiring that

$$Q \in K_1 \cap K_2 \cap \cdots \cap K_{(\ell+t)m} = \overline{K}_{(\ell+t)m}.$$

By entension of prop. 4.23 to the case of $B = \{x, y_1, \ldots, y_L\}$, these kernels are $\mathscr{L}_{q^{m\ell}}[X]$-submodules of V. Furthermore, the element $Q \in V$ should be of lowest order. Since the order $\prec$ is a total order on the monomials in M, the index of a polynomial $\mathrm{ind}(Q)$ and the order $\mathrm{ord}(Q)$ are well-defined. Therefore, alg. 3 can be used to solve this problem.

4.7 Linearized Roth-Ruckenstein algorithm

In this section, we will look at the Linearized Roth-Ruckenstein algorithm (LRR)-algorithm from [MV12], which is a generalisation of the algorithm in [RR00]. This algorithm is used for finding roots of equations over the ring of linearized polynomials. In our case, we want all solutions $f(x)$ in $\mathscr{L}_q^k[X]$ of the equation

$$Q(x,y) = Q_0(x) + Q_1(x) \otimes y + Q_2(x) \otimes y^{\otimes 2} + \cdots + Q_L(x) \otimes y^{\otimes L} = 0. \quad (4.19)$$

In the above equation, each $Q_i(x)$ is a linearized polynomial over some extension field $\mathbb{F}$ of $\mathbb{F}_q$. We say that the polynomial $Q(x,y)$ is divisible by x^{q^s} if every term $Q_i(x)$ is divisible by x^{q^s}, for some integer s. We then write $Q_i'(x)$ for the result of this division, so $(Q_i'(x))^{q^s} = Q_i(x)$. We also write

$$Q_{\downarrow s}(x,y) = Q_0'(x) + Q_1'(x) \otimes y + Q_2'(x) \otimes y^{\otimes 2} + \cdots + Q_L'(x) \otimes y^{\otimes L}.$$

The algorithm is shown in alg. 4, and we now explain how it works.

Algorithm 4 Linearized Roth-Ruckenstein algorithm

LRR$(Q(x,y), k, \lambda)$
Input: Bivariate linearized polynomial $Q(x,y)$, $k \in \mathbb{N}$, and $\lambda \in \mathbb{N}_0$.
Output: Set of linearized polynomials over $\mathbb{F}_q$ of degree at most q^{k-1}, which are y-roots in $Q(x,y)$.
Global variables: Set $A \subseteq \mathscr{L}_q^k[X]$
Polynomial $g(x) = \sum_{i=0}^{k-1} u_i x^{q^i} \in \mathscr{L}_q^k[X]$.
if $\lambda = 0$ **then**
    $A \leftarrow \emptyset$
end if
$s \leftarrow$ largest integer such that x^{q^s} divides $Q(x,y)$.
$H(x,\gamma) \leftarrow \frac{1}{x} Q_{\downarrow s}(x, \gamma x)$.
$Z \leftarrow$ set of roots of $H(0,\gamma)$ in $\mathbb{F}_q$.
for each $\gamma \in Z$ **do**
    $u_\lambda \leftarrow \gamma$
    if $\lambda < k-1$ **then**
        LRR$(Q_{\downarrow s}(x, y^q + \gamma x), k, \lambda + 1)$
    else
        if $Q(x, u_{k-1}x) = 0$ **then**
            $A \leftarrow A \cup \{g(x)\}$
        end if
    end if
end for

The algorithm is given the parameters $Q(x,y)$, k and λ. Here, $Q(x,y)$ is a bivariate linearized polynomial over $\mathbb{F}$. The integer k means that all

solutions to $Q(x, f(x)) = 0$, where $f(x) \in \mathscr{L}_q^k[X]$, will have degree q^{k-1} or less. The last parameter λ is used to keep track of the level of recursion.

Each time the algorithm is called, it defines a set A and a polynomial $g(x)$. The set A is the set of roots of $Q(x, y)$ in $\mathscr{L}_q^k[X]$, and $g(x)$ is, at the end of a recursion descent, a root of $Q(x, y)$. Note that the first time the algorithm is called, the parameters are $Q(x, y)$ defined in eq. (4.19), $k > 0$ and $\lambda = 0$.

The first time, $\lambda = 0$, so the algorithm initializes the set A to the empty set. As long as $\lambda < k - 1$, the algorithm calls itself, where λ is increased by one each time. In the line

$$Z \leftarrow \text{ set of roots of } H(0, \gamma) \in \mathbb{F}_q,$$

we find all roots of the univariate polynomial $H(0, \gamma)$ in $\mathbb{F}_q$. There exist various algorithms for accomplishing this task.

For recursion level i, let for $i = 0, 1, 2, \ldots, k - 1$,

$$\begin{aligned} P_i(x, y) &\text{ be the value of } Q(x, y) \\ T_i(x, y) &\text{ be the value of } Q_{\downarrow s}(x, y) \\ H_i(x, y) &\text{ be the value of } H(x, \gamma) \end{aligned}$$

Then, for $i = 0$, we have $P_0(x, y) = Q(x, y) \neq 0$ by assumption. If $P_i(x, y)$ is non-zero, then $T_i(x, y)$ is non-zero. But then **LRR** is called with the arguments $T_i(x, y^q + \gamma x)$, k and $\lambda + 1$. So at recursion level $i + 1$, $P_{i+1}(x, y) = T_i(x, y^q + \gamma x)$, which is non-zero. By induction, P_i and T_i are non-zero for all $i = 0, 1, \ldots, k - 1$. Therefore, the exponent s in q^s is well-defined.

4.7.1 Correctness of the LRR algorithm

Here, we prove that the algorithm is correct, by a series of lemmas.

In the first lemma, it is shown that the set A only consists of the roots of $Q(x, y)$ in $\mathscr{L}_q^k[X]$.

Lemma 4.26. *Let A be the set computed by the call $\boldsymbol{LRR}(Q, k, 0)$. Then every element of A is a root of $Q(x, y)$ in $\mathscr{L}_q^k[X]$.*

Proof. Let

$$f(x) = u_0 x + u_1 x^q + \cdots + u_{k-1} x^{q^{k-1}} \in A.$$

For $i = 0, 1, 2, \ldots, k$, define $\phi_i(x)$ as

$$\phi_i(x) = u_i x + u_{i+1} x^q + \cdots + u_{k-1} x^{q^{k-i-1}}$$

In other words, define the $\phi_i(x)$'s as

$$\begin{aligned}
\phi_0(x) &= u_0 x + u_1 x^q + \cdots + u_{k-1} x^{q^{k-1}} \\
\phi_1(x) &= u_1 x + u_2 x^q + \cdots + u_{k-1} x^{q^{k-2}} \\
&\vdots \\
\phi_{k-2}(x) &= u_{k-2} x + u_{k-1} x^q \\
\phi_{k-1}(x) &= u_{k-1} x,
\end{aligned}$$

where $u_i \in \mathbb{F}_q$. Now,

$$\begin{aligned}
\phi_{i+1}(x)^q &= \left(u_{i+1} x + u_{i+2} x^q + \cdots + u_{k-1} x^{q^{k-1}} \right)^q \\
&= u_{i+1} x^q + u_{i+2} x^{q^2} + \cdots + u_{k-1} x^{q^k},
\end{aligned}$$

and so $\phi_{i+1}(x)^q + u_i x = \phi_i(x)$.

For recursion level i, let

$$P_i = Q, \quad \text{and } T_i = T.$$

We show that $\phi_i(x)$ is a root of P_i. The proof is backward induction on i. For the base case, $i = k - 1$. Here

$$\phi_i(x) = \phi_{k-1}(x) = u_{k-1} x,$$

and ϕ_{k-1} is a root of P_{k-1}. This is because by hypothesis, the algorithm is called with $\lambda = 0$, and so the expression $Q(x, u_{k-1} x) = 0$ is true.

Turning to the induction step, suppose that ϕ_{i+1} is a root of P_{i+1}. Then this implies that ϕ_i is a root of P_i, because

$$\begin{aligned}
P_i(x, \phi_i) &\overset{(1)}{=} T_i(x, \phi_i)^{q^s} \\
&= T_i(x, \phi_{i+1}^q + u_i x)^{q^s} \\
&\overset{(2)}{=} P_{i+1}(x, \phi_{i+1})^{q^s} \overset{(3)}{=} 0,
\end{aligned}$$

where (1) follows from the definition of T_i, (2) is by the definition of P_{i+1}, and for equality (3), we have used the induction hypothesis.

Now, if $i = 0$, $P_0(x, \phi_0) = Q(x, \phi_0) = Q(x, f(x)) = 0$, which shows that $f(x)$ is a root of $Q(x, y)$. Note that because $\gamma \in Z \subseteq \mathbb{F}_q$, the polynomial $f(x)$ is over $\mathbb{F}_q$. □

Lemma 4.27. *Let*

$$\begin{aligned} Q(x,y) &= Q_0(x) + Q_1(x) \otimes y + \cdots + Q_L(x) \otimes y^{\otimes L}, \\ f(x) &= f_0 x + f_1 x^q + \cdots + f_{k-1} x^{k-1} \in \mathscr{L}_q^k[X] \text{ and} \\ H(x,\gamma) &= \frac{1}{x} Q(x, \gamma x), \end{aligned}$$

where $\gamma \in \mathbb{F}_q$. Then the coefficient of x in $Q(x, f(x))$ equals $H(0, f_0)$.

Proof. Firstly, note that the coefficient of x in $f^{\otimes i}(x)$ is equal to f_0^i, and the x-term is $f_0^i x$. This implies that the coefficient of x in $Q(x, f(x))$ is equal to the coefficient of x in

$$Q_0(x) + Q_1(f_0 x) + Q_2(f_0^2 x) + \cdots + Q_L(f_0^L x),$$

which equals $xH(x, f_0)$. Note that the coefficient of x in $xH(x, f_0)$ equals the constant term in $H(x, f_0)$, which is $H(0, f_0)$. □

The next two lemmas constitute a converse to the previous lemma.

Lemma 4.28. *Let*

$$f(x) = u_0 x + u_1 x^q + \cdots + u_{k-1} x^{q^{k-1}}$$

be a root of $Q(x, y)$ in $\mathscr{L}_q^k[X]$. For $i = 0, 1, \ldots, k-1$ define $P_i(x, y)$ and $T_i(x, y)$ inductively as follows:

$$\begin{aligned} P_0(x,y) &= Q(x,y) \\ P_i(x,y) &= T_i(x,y)^{q^{s_i}} \\ P_{i+1}(x,y) &= T_i(x, y^q + f_i x), \end{aligned}$$

where s_i is the largest integer such that $x^{q^{s_i}}$ divides $P_i(x, y)$. Define

$$H_i(x,\gamma) = \frac{1}{x} T_i(x, \gamma x).$$

Then, for $i = 0, 1, \ldots, k-1$, the following holds

1. *$\phi_i(x) = f_i x + f_{i+1} x^q + \cdots + f_{k-1} x^{k-1-i}$ is a root of $P_i(x, y)$*
2. *$H_i(0, f_i) = 0$.*

Proof. We first prove part 1. This is by induction on i. For the base case, $i = 0$. Then

$$\phi_0(x) = f_0 x + f_1 x^q + \cdots + f_{k-1} x^{q^{k-1}} = f(x)$$

is a root of $P_0 = Q$ by hypothesis.

For the induction step, suppose $\phi_i(x)$ is a root of $P_i(x, y)$. We have

$$\begin{aligned}\phi_i(x) &= f_i x + f_{i+1} x^q + \cdots + f_{k-1} x^{q^{k-i-1}} \\ &= \left(f_{i+1} x + f_{i+2} x^q + \cdots + f_{k-1} x^{q^{k-i}}\right)^q + f_i x,\end{aligned}$$

that is, $\phi_i(x) = \phi_{i+1}^q + f_i x$. This shows that $\phi_{i+1}^q + f_i x$ is a root of $P_i(x, y)$, so

$$P_i(x, \phi_{i+1}^q + f_i x) = 0.$$

Let $y = \phi_{i+1}$, then ϕ_{i+1} is a root of $P_i(x, y^q + f_i x)$. Because $T_i(x, y^q + f_i x) = P_{i+1}(x, y)$, ϕ_{i+1} is a root of $P_{i+1}(x, y)$. By induction, the statement holds for all $i \geq 0$.

Now to part 2. We have

$$P_i(x, \phi_i(x)) = T_i(x, \phi_i(x))^{q^{s_i}} = 0,$$

where the first equality comes from the definition of T_i, while in the second equality, we have used part 1. We see that $T_i(x, \phi_i(x))^{q^{s_i}} = 0$ implies $T_i(x, \phi_i(x)) = 0$ because there are no zero divisors in $\mathscr{L}_{q^m}[X]$. Note that the x-term of $T_i(x, \phi_i(x))$ equals $H_i(0, f_i)$ (using lemma 4.27), and because $T_i(x, \phi_i(x)) = 0$ we have $H_i(0, f_i) = 0$. □

Using the previous lemma, the next lemma is a converse to lemma 4.26.

Lemma 4.29. *Let A be the set computed by calling $\boldsymbol{LRR}(Q, k, 0)$. Then every root of $Q(x, y)$ in $\mathscr{L}_q^k[X]$ is contained in A.*

Proof. Let

$$f(x) = f_0 x + f_1 x^q + \cdots + f_{k-1} x^{q^{k-1}}$$

be a root of $Q(x, y)$ in $\mathscr{L}_q^k[X]$. We define $H_i(x, y)$, $T_i(x, y)$ and $Q_i(x, y)$ as in lemma 4.28. We prove by induction on i that there is a recursion descent such that at recursion level i, **LRR** is called with the parameters (P_i, k, i). For the base case $i = 0$, we execute $\mathbf{LRR}(P_0, k, 0)$. Because $P_0 = Q$, the base case is correct. Turning to the induction step, suppose that at recursion level i, **LRR** is called with parameters (P_i, k, i). Then, using part 2 of lemma 4.28, $H_i(0, f_i) = 0$, and so f_i is one of the roots of $H_i(0, \gamma)$ in Z.

If $i < k-1$, then **LRR** is called with parameters $Q_{\downarrow s}(x, y^q + \gamma x) = T_i(x, y^q + f_i x)$, k and $\lambda + 1 = i + 1$. Because $T_i(x, y^q + f_i x) = P_{i+1}(x, y)$, the call to **LRR** is made with the arguments $(P_{i+1}, k, i+1)$.

Else, $i = k-1$, and then $P_{k-1}(x, f_{k-1}) = 0$. Because $P_{k-1}(x, f_{k-1}) = Q(x, u_{k-1}x) = 0$, the else step is executed, and so $g(x)$ is added to the set A. □

Theorem 4.30. *The algorithm **LRR** is correct.*

Proof. From lemma 4.26, we know that every element of A is a root of $Q(x, y)$, and from lemma 4.29, we have that every root of $Q(x, y)$ is contained in the set A. Therefore, the set A consists exactly of the roots of $Q(x, y)$ in $\mathscr{L}_q^k[X]$. □

Chapter 5

Conclusion

In this thesis, we have studied the problem of coding for the operator channel. One important class of subspace codes are the KK-codes. We looked at these codes in chapter 3, where various properties of these codes were shown. One interesting result is the fact that these codes asymptotically achieve the Singleton bound in the subspace metric. The issue of decoding was also addressed, where an ordering on monomials was used to establish correctness of the decoding procedure.

Subspace codes suitable for list decoding were also presented. These codes, called MV-codes, are an extension of KK-codes. We looked at the construction and error- and erasure capabilities of these codes. Compared to KK-codes, these codes contain fewer codewords, when the underlying finite field is fixed. The interpolation step in decoding subspace codes was considered in the language of modules.

We saw that MV-codes have better decoding radius than KK-codes for low packet rates. By introducing multiplicity on the ring of lineraized polynomials, we saw that MV-codes are better than KK-codes in terms of decoding radius for all rates. One question is whether it is possible to do list decoding of KK-codes without modifying them.

Bibliography

[ACLY00] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W.-H. Yeung. Network Information Flow. *IEEE Trans. Inf. Th.*, 46(4):1204–1216, 2000.

[Ber80] E. R. Berlekamp. The Technology of Error-Correcting Codes. *Proc. IEEE*, 68(5):564–593, 1980.

[Cam00] P. J. Cameron. Notes on classical groups. Lecture Notes, 2000.

[CLO07] D. Cox, J. Little, and D. O'Shea. *Ideals, Varieties, and Algorithms.* Springer, 2007.

[CLRS01] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms, 2nd edition.* MIT Press and McGraw-Hill, 2001.

[DF04] D. S. Dummit and R. M. Foote. *Abstract Algebra, 3rd edition.* John Wiley and Sons, 2004.

[FS07] C. Fragouli and E. Soljanin. *Network Coding Fundamentals.* now Publishers Inc, 2007.

[GT11] O. Geil and C. Thomsen. *Aspects of random network coding.* World Scientific, 2011.

[HK71] K. M. Hoffman and R. Kunze. *Linear Algebra, 2nd Edition.* Prentice Hall, 1971.

[HMK+06] T. Ho, M. Médard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong. A random linear network coding approach to multicast. *IEEE Trans. Inf. Th.*, 52:4413–4430, 2006.

[KK08] R. Koetter and F.R. Kschischang. Coding for Errors and Erasures in Random Network Coding. *IEEE Trans. Inf. Th.*, 54(8):3579–3591, 2008.

[KM03] R. Koetter and M. Médard. An algebraic approach to network coding. *IEEE/ACM Trans. Networking*, 11:782–795, 2003.

[Kon98] J. Konvalina. Generalized Binomial Coefficients and the Subset Subspace Problem. *Adv. Appl. Math*, 21:228–240, 1998.

[LN83] R. Lidl and H. Niederreiter. *Finite Fields*. Cambridge University Press, 1983.

[LS87] H.W. Lenstra and R.J. Schoof. Primitive Normal Bases for Finite Fields. *Math. Comp.*, 48(177):217–231, 1987.

[LW92] J. H. Van Lint and R. M. Wilson. *A Course in Combinatorics*. Cambridge University Press, 1992.

[LYC03] S.-Y. R. Li, R. W. Yeung, and N. Cai. Linear network coding. *IEEE Trans. Inf. Th.*, 49(2):371–381, February 2003.

[Mah12] H. Mahdavifar. Private communication, 2012.

[MS77] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error-Correcting Codes*. North Holland Publishing Co., 1977.

[MV11] H. Mahdavifar and A. Vardy. Algebraic List-decoding of Subspace Codes with Multiplicities. *Proceedings of the 49th annual Allerton Conference on Communications, Control and Computing*, pages 1430–1437, 2011.

[MV12] H. Mahdavifar and A. Vardy. Algebraic List-decoding of Subspace Codes. *arXiv preprint*, abs/1202.0338, 2012.

[Rot06] R. M. Roth. *Introduction to Coding Theory*. Cambridge University Press, 2006.

[RR00] R. M. Roth and G. Ruckenstein. Efficient Decoding of Reed-Solomon Codes Beyond Half the Minimum Distance. *IEEE Trans. Inf. Th.*, 46(1):246–257, 2000.

[Wes01] D. B. West. *Introduction to Graph Theory, 2nd edition*. Prentice Hall, 2001.

[XYS11] H. Xie, Z. Yan, and B. Suter. General Linearized Polynomial Interpolation and Its Applications. *arXiv preprint*, abs/1104.3886, 2011.