REPUBLIQUE DU CAMEROUN
Paix – Travail – Patrie
****
UNIVERSITE DE YAOUNDE I
FACULTE DES SCIENCES
DEPARTEMENT DE MATHÉMATIQUES
****
CENTRE DE RECHERCHE ET DE FORMATION DOCTORALE EN SCIENCES, TECHNOLOGIES ET GÉOSCIENCES
LABORATOIRE D'ALGÈBRE, GÉOMÉTRIE ET APPLICATIONS

REPUBLIC OF CAMEROUN
Peace – Work – Fatherland
****
UNIVERSITY OF YAOUNDE I
FACULTY OF SCIENCE
DEPARTMENT OF MATHEMATICS
****
POSTGRADUATE SCHOOL OF SCIENCE, TECHNOLOGY AND GEOSCIENCES
LABORATORY OF ALGEBRA, GEOMETRY AND APPLICATIONS

Rank-Metric Codes Over Finite Principal Ideal Rings and Applications in Wireless Communication Systems

THESIS
Submitted in partial fulfilment of the requirements for the award of Doctorat/Ph.D in Mathematics

Par : **TCHATCHIEM KAMCHE Hermann**
Master in Mathematics

Sous la direction de
MOUAHA Christophe
Associate Professor

Année Académique : 2019 - 2020

REPUBLIQUE DU CAMEROUN
Paix-Travail-Patrie
....
UNIVERSITE DE YAOUNDE I
FACULTE DES SCIENCES
B.P 812 Yaoundé
Tel / Fax: (237) 22 23 44 96

REPUBLIC OF CAMEROON
Peace-Work-Fatherland
....
THE UNIVERSITY OF YAOUNDE I
FACULTY OF SCIENCE
P.O box 812 Yaounde
Tel / Fax: (237) 22 23 44 96

DEPARTEMENT DE MATHEMATIQUES
DEPARTMENT OF MATHEMATICS

ATTESTATION DE CORRECTION DE LA THESE DE DOCTORAT/PH.D

Nous soussignés, membres du jury de soutenance de la thèse de Doctorat/Ph.D de Monsieur **TCHATCHIEM KAMCHE Hermann**, Matricule **07V914**, intitulée : « **Rank-Metric Codes Over Finite Principal Ideal Rings and Applications in Wireless Communication Systems**» soutenue le 28 Juillet 2020, attestons que toutes les corrections demandées par le jury de soutenance ont été effectuées.

En foi de quoi, la présente attestation lui est délivrée pour servir et valoir ce que de droit.

Yaoundé, le 09/08/2020

Président :	**BITJONG NDOMBOL, Professeur**	**Université de Yaoundé I**	________
Rapporteur :	**MOUAHA Christophe, Maître de Conférences**	**Université de Yaoundé I**	________
Membres :	**LELE Célestin, Professeur**	**Université de Dschang**	________
	NKUIMI JUGNIA Célestin, Maître de Conférences	**Université de Yaoundé I**	________
	TEMGOUA ALOMO Etienne Romuald, Maître de Conférences	**Université de Yaoundé I**	________
	NDJEYA Sélestin, Maître de Conférences	**Université de Yaoundé I**	________

RÉPUBLIQUE DU CAMEROUN
Paix-Travail-Patrie
****
UNIVERSITÉ DE YAOUNDÉ I
****
CENTRE DE RECHERCHE ET DE FORMATION DOCTORALE EN SCIENCES, TECHNOLOGIES ET GÉOSCIENCES
****
UNITÉ DE RECHERCHE ET DE FORMATION DOCTORALE EN MATHÉMATIQUES, INFORMATIQUE, BIOINFORMATIQUE ET APPLICATIONS

REPUBLIC OF CAMEROON
Peace-Work-Fatherland
****
THE UNIVERSITY OF YAOUNDÉ I
****
POSTGRADUATE SCHOOL OF SCIENCE, TECHNOLOGY AND GEOSCIENCES
****
RESEARCH AND TRAINING UNIT FOR DOCTORATE IN MATHEMATICS, COMPUTER SCIENCES AND APPLICATIONS

DÉPARTEMENT DE MATHÉMATIQUES
DEPARTMENT OF MATHEMATICS

LABORATOIRE D'ALGÈBRE, GÉOMÉTRIE ET APPLICATIONS
LABORATORY OF ALGEBRA, GEOMETRY AND APPLICATIONS

Rank-Metric Codes Over Finite Principal Ideal Rings and Applications in Wireless Communication Systems

THESIS
Submitted in partial fulfilment of the requirements for the award of Doctorat/Ph.D in Mathematics
Speciality: Algebra

By:
TCHATCHIEM KAMCHE Hermann
Registration number: 07V914
Master in Mathematics

Thesis defended on July 28, 2020 in front of the jury made up of:

President :	**BITJONG NDOMBOL,** *Professor*	**University of Yaoundé I;**
Reporter :	**MOUAHA Christophe,** *Associate Professor*	**University of Yaoundé I;**
Members :	**LELE Célestin,** *Professor*	**University of Dschang;**
	NKUIMI JUGNIA Célestin, *Associate Professor*	**University of Yaoundé I;**
	TEMGOUA ALOMO Etienne Romuald, *Associate Professor*	**University of Yaoundé I;**
	NDJEYA Sélestin, *Associate Professor*	**University of Yaoundé I.**

Academic Year: 2019/2020

Dedicated to my family

Acknowledgements

Firstly, I express my gratitude to my supervisor, Prof. Christophe Mouaha, who introduced me to algebra and coding theory. I also thank him for his availability, his rigor, his trust and his valuable advice.

I would like to thank the members of the Laboratory of Algebra, Geometry and Applications of the University of Yaoundé I and the members of ERAL (Equipe de Recherche en Algèbre et Logique). I especially thank Prof. Marcel Tonga, Prof. Célestin Nkuimi, Prof. Sélestin Ndjeya, Prof. Daniel Tieudjo, Prof. Célestin Lele, Prof. Etienne Romuald Temgoua Alomo, Dr. Maurice Kianpi, Dr. Michel Djiadeu Ngaha, Dr. Romain Nimpa Pefoukeu, Dr. Emmanuel Fouotsa, Dr. Alexandre Fotue Tabue, Dr. Hervé Talé Kalachi, Rostand Kuitché and Francis Nyamda for their useful remarks and suggestions. I also thank my friend Dr. Miradain Atontsa Nguemo for his comments and encouragement.

I warmly thank my wife, children, parents and the rest of my family for their encouragement and assistance.

Abstract

Rank-metric codes have been studied over finite fields and the applications have been given in network coding and cryptography. Recent works on nested-lattice-based network coding allow the construction of more efficient physical-layer network coding schemes with network coding over finite principal ideal rings. In this new algebraic approach, it is necessary to detect and correct errors introduced into the system.

In this thesis, it is shown that some results in the theory of rank-metric codes over finite fields can be extended to finite commutative principal ideal rings. More precisely, the rank metric is generalized and the rank-metric Singleton bound is established. The definition of Gabidulin codes is extended and it is shown that their properties are preserved. The theory of Gröbner bases is used to give the unique decoding, minimal list decoding, and error-erasure decoding algorithms of interleaved Gabidulin codes. These results are then applied in space-time codes and in random linear network coding as in the case of finite fields. Specifically, two existing encoding schemes of random linear network coding are combined to improve the error correction.

Keywords: finite principal ideal rings, Galois extensions, Gröbner bases, interleaved Gabidulin codes, random linear network coding, rank-metric codes, skew polynomials, space-time codes.

Résumé

Les codes en métrique rang ont été étudiés sur des corps finis et les applications ont été données en codage réseau et en cryptographie. Des travaux récents sur le codage réseau basé sur les réseaux de points emboîtés permettent de construire des schémas de codage réseau de couche physique plus efficaces avec un codage réseau sur les anneaux commutatifs finis principaux. Dans cette nouvelle approche algébrique, il est nécessaire de détecter et de corriger les erreurs introduites dans le système.

Dans cette thèse, il est montré que certains résultats de la théorie du codage en métrique rang sur les corps finis peuvent être étendus aux anneaux commutatifs finis principaux. Plus précisément, la métrique rang est généralisée et la borne de Singleton en métrique rang est établie. La définition des codes de Gabidulin est étendue et leurs propriétés sont préservées. La théorie des bases de Gröbner est utilisée pour donner des algorithmes de décodage unique, de décodage en liste minimal et de décodage d'erreur-effacement des codes de Gabidulin entrelacés. Ces résultats sont ensuite appliqués dans le codage spatio-temporel et dans le codage réseau linéaire aléatoire, comme dans le cas des corps finis. Plus précisément, deux systèmes du codage réseau linéaire aléatoire existants sont combinés pour améliorer la correction d'erreurs.

Mots clés: anneaux finis principaux, extensions de Galois, bases de Gröbner, codes de Gabidulin entrelacés, codage réseau linéaire aléatoire, codes en métrique rang, polynômes tordus, codes spatio-temporels.

Contents

Notations

Rings and modules

$\mathbb{F}_q$	Finite field of order q
$\mathbb{Z}_\eta$	The ring of integers modulo η
$\mathbb{Z}_\eta[i]$	The ring $\mathbb{Z}_\eta + i\mathbb{Z}_\eta$ where $i^2 = -1$
R	A finite commutative principal ideal ring
$a \mid b$	a divides b, i.e. $b = ca$ for some $c \in R$
$\mu_R(M)$	The minimum number of generators of the R-module M
$\langle \{u_j\}_{1\leq j\leq r} \rangle$	The R-submodule generated by $\{u_j\}_{1\leq j\leq r}$

Matrices

$R^{m\times n}$	The set of all $m \times n$ matrices with entries from R
$\mathbf{I}_k$	The $k \times k$ identity matrix
$row(\mathbf{A})$	The R-submodules generated by the row vectors of the matrix $\mathbf{A}$
$col(\mathbf{A})$	The R-submodules generated by the column vectors of the matrix $\mathbf{A}$
$diag(d_1, \ldots, d_r)$	A diagonal matrix
$rank(\mathbf{A})$	The rank of the matrix $\mathbf{A}$
$freerank(\mathbf{A})$	The free rank of the matrix $\mathbf{A}$

Galois extensions of finite principal ideal rings

$R \cong R_{(1)} \times \cdots \times R_{(\rho)}$	The decomposition of R as the product of local rings $R_{(i)}$
$\mathfrak{m}_{(i)}$	The maximal ideal of $R_{(i)}$
$\mathbb{F}_{q_{(i)}}$	The residue field of $R_{(i)}$, i.e. $R_{(i)}/\mathfrak{m}_{(i)}$
$\nu_{(i)}$	the nilpotency index of $\mathfrak{m}_{(i)}$
$S_{(i)}$	The Galois extension of $R_{(i)}$ of dimension m
$\mathfrak{M}_{(i)}$	The maximal ideal of $S_{(i)}$
$\sigma_{(i)}$	A generator of the Galois group of $S_{(i)}$
$S = S_{(1)} \times \cdots \times S_{(\rho)}$	The Galois extension of R of dimension m
$\sigma = \left(\sigma_{(i)}\right)_{1\leq i\leq \rho}$	A generator of the Galois group of S

Skew polynomials

$S[X,\sigma]$	The skew polynomial ring over S with automorphism σ
$S[X,\sigma]_{<k}$	The set of all skew polynomials of degree less than k
$f = f_0 + f_1 X + \cdots + f_n X^n$	An element of $S[X,\sigma]$, with $f_n \neq 0$
$\deg(f)$	The degree of f, i.e. n
$lm(f)$	The leading monomial of f, i.e. X^n
$lc(f)$	The leading coefficient of f, i.e. f_n
$lt(f)$	The leading term of f, i.e. $f_n X^n$
$f(b)$	The element $f_0 b + f_1 \sigma(b) + \cdots + f_n \sigma^n(b)$ where $b \in S$
$f(\mathbf{b})$	The vector $(f(b_1), \ldots, f(b_n))$ where $\mathbf{b} = (b_1, \ldots, b_n) \in S^n$
$\ker f$	The kernel of f, i.e. $\{x \in S : f(x) = 0\}$

Gröbner bases of modules over skew polynomials

$S[X,\sigma]^{\ell+1}$	The $\ell+1$-fold direct product of $S[X,\sigma]$
$\left(\mathbf{e}^{(0)}, \mathbf{e}^{(1)}, \ldots, \mathbf{e}^{(\ell)}\right)$	The canonical basis of $S[X,\sigma]^{\ell+1}$
$X^{\alpha}\mathbf{e}^{(l)}$	A monomial in $S[X,\sigma]^{\ell+1}$
$ind\left(X^{\alpha}\mathbf{e}^{(l)}\right)$	The index of $X^{\alpha}\mathbf{e}^{(l)}$, i.e. l
$X^{\alpha_1}\mathbf{e}^{(l_1)} \vert X^{\alpha_2}\mathbf{e}^{(l_2)}$	$X^{\alpha_1}\mathbf{e}^{(l_1)}$ divides $X^{\alpha_2}\mathbf{e}^{(l_2)}$, i.e. $l_1 = l_2$ and $\alpha_1 \leq \alpha_2$
$Mon\left(S[X,\sigma]^{\ell+1}\right)$	The set of monomials of $S[X,\sigma]^{\ell+1}$
$\succeq$	A monomial order on $Mon\left(S[X,\sigma]^{\ell+1}\right)$
$\mathbf{f} = \sum_{i=1}^{n} c_i X^{\alpha_i}\mathbf{e}^{(l_i)}$	An element of $S[X,\sigma]^{\ell+1}$, with $c_1 \neq 0$ and $X^{\alpha_1}\mathbf{e}^{(l_1)} \succ \cdots \succ X^{\alpha_n}\mathbf{e}^{(l_n)}$
$lm(\mathbf{f})$	The leading monomial of $\mathbf{f}$, i.e. $X^{\alpha_1}\mathbf{e}^{(l_1)}$
$lc(\mathbf{f})$	The leading coefficient of $\mathbf{f}$, i.e. c_1
$lt(\mathbf{f})$	The leading term of $\mathbf{f}$, i.e. $c_1 X^{\alpha_1}\mathbf{e}^{(l_1)}$
$\deg(\mathbf{f})$	The degree of $\mathbf{f}$, i.e. α_1
$\mathbf{f} \xrightarrow{F} \mathbf{h}$	f reduces to h by F in one step
$\mathbf{f} \xrightarrow{F}_{+} \mathbf{h}$	f reduces to h by F

Rank-metric codes

$\mathcal{M}$	A matrix rank code, i.e. a subset of $R^{m\times n}$
$d(\mathcal{M})$	The rank distance of a matrix rank code $\mathcal{M}$, i.e. $\min\{rank(\mathbf{A}-\mathbf{B}) :\ \mathbf{A},\ \mathbf{B} \in \mathcal{M},\ \mathbf{A} \neq \mathbf{B}\}$
$\mathcal{C}$	A vector rank code, i.e. a subset of S^n
$d(\mathcal{C})$	The rank distance of the vector rank code $\mathcal{C}$, i.e. $\min\{rank(\mathbf{u}-\mathbf{v}) :\ \mathbf{u},\ \mathbf{v} \in \mathcal{C},\ \mathbf{u} \neq \mathbf{v}\}$
$\mathcal{C}^{\perp}$	The dual of $\mathcal{C}$
$Gab_k(\mathbf{g})$	The Gabidulin code of length n, dimension k and support $\mathbf{g} \in \mathbf{S}^n$
$IGab_{\left(k^{(1)},\ldots,k^{(\ell)}\right)}\left(\mathbf{g}^{(1)}, \ldots, \mathbf{g}^{(\ell)}\right)$	An Interleaved Gabidulin code

Introduction

In a communication network, the transmitters can send information simultaneously to the receivers. These are represented by a matrix where rows consist of various information. Practically, it may happen some perturbations and the received signals be different from the transmitted ones. In such predicament, for securing the system against noises, one can use the rank-metric codes to detect and correct errors.

Rank-metric codes

Rank-metric codes [16] are codes for which each codeword is a matrix and the distance between two codewords is the rank of their difference. The most important family of rank-metric codes is that of Gabidulin codes [16], [24], [63]. They are optimal in the sense that they achieve the rank-metric Singleton bound. In [24], Gabidulin used the Galois extension to give the vector representation of rank-metric codes. He also gave a polynomial-time unique decoding algorithm of Gabidulin codes.

The length of a Gabidulin code is lower bounded by the degree of the Galois extension. To increase the code length, we can use an interleaved Gabidulin code [46] which is a direct sum of several Gabidulin codes. Another advantage of interleaved Gabidulin codes is the existence of polynomial-time decoding algorithms [46], [67], [79] that can decode beyond the error correction capability with high probability. Nowadays, rank-metric codes are used in space-time coding [48], public key cryptosystems [25] and random linear network coding [69].

Space-time codes based on rank-metric codes

A space-time code is a multiple-input/multiple-output transmit strategy for fading channels in point-to-point single-user scenarios. It was introduced in [74] by Tarokh et al. It combines the space diversity, provided by multiple antennas, and the time diversity to increase system capacity and reduce multipath fading. Among the performance criteria for space-time codes, we have the rank criterion [74] which states that in order to achieve the maximum diversity, the rank of the difference of two distinct codewords has to be maximal. On the other hand, for any space-time block code there is a tradeoff between the transmission rate and the transmit diversity gain [74], [47]. As in [37], a space-time block code that achieves this rate-diversity tradeoff will be called an optimal space-time block code. To construct these optimal codes, rank-metric codes can be used. Thus, in [48] Lusina et al. used rank-preserving map from finite fields to Gaussian integers to construct optimal space-time block codes from rank-metric codes over finite fields. In [2], Asif et al. used interleaved Gabidulin codes to construct space-time block codes and compared them to orthogonal space-time block codes. In [61], Puchinger et al. extended the works of Lusina et al. [48] to Eisenstein integers. They also proposed decoding scheme of space-time block codes using lattice-reduction-aided equalization and error-erasure decoding algorithm of Gabidulin codes. In [3], Augot et al. transposed the theory of rank metric and Gabidulin codes to the case of fields of characteristic zero.

Rank-metric codes in random linear network coding

A random linear network coding is a technique that can be used to disseminate information in networks and improve the performance of communication systems. In the transmission model for end-to-end coding over finite fields, the channel equation is given by $\mathbf{Y} = \mathbf{AX} + \mathbf{E}$, where $\mathbf{X}$ is the transmitted matrix whose rows are packets transmitted by the source node; $\mathbf{Y}$ is the received matrix whose rows are the packets received by the sink node; $\mathbf{A}$ is a transfer matrix corresponding to the overall linear transformation applied by intermediate nodes of the network and $\mathbf{E}$ is an error matrix whose rows are linear combinations of corrupt packets injected in the network. Random matrices $\mathbf{A}$ and $\mathbf{E}$ are unknown to the destination. The problem is to recover the transmitted codeword $\mathbf{X}$ from the received matrix $\mathbf{Y}$.

Since linear network coding is vector-space preserving, Kötter and Kschischang [38] suggested the use of a basis of a vector space as the rows of the transmitted matrix. They defined a distance function between subspaces, constructed a family of constant-dimension subspace codes and the decoding algorithm. In [69] Silva et al. used the lifted rank-metric codes to show that minimum distance decoding of constant-dimension subspace codes can be reformulated as a generalized decoding problem for rank-metric codes. They then gave an error-erasure decoding algorithm of Gabidulin codes to solve the problem of error control in random linear network coding.

Network coding over finite principal ideal rings

A principal ideal ring is a ring in which any ideal is generated by one element. In a digital modulation system, some signal constellation sets can be represented by a finite principal ideal ring. In particular [22], if η is some positive integer then the signal constellation set of the η^2-ary square quadrature amplitude modulation is represented by the ring $\mathbb{Z}_\eta[i] = \mathbb{Z}_\eta + i\mathbb{Z}_\eta$ where $i^2 = -1$ and $\mathbb{Z}_\eta$ is the ring of integers modulo η. The works on nested-lattice-based network coding [51], [22] allow the construction of more efficient physical-layer network coding schemes with network coding over finite principal ideal rings. Motivated by this algebraic approach, space-time codes and random linear network coding were studied in the specific cases of principal ideal rings.

In [37], Kiran and Rajan extended the definition of Gabidulin codes to Galois rings and used a rank-preserving map to construct an optimal space-time block code. In [44], Liu et al. defined the notion of $\sum_o$-rank over the ring $\mathbb{Z}_{2^k}[i]$ and used it to construct the rank metric space-time codes for the 2^{2k} quadrature and amplitude modulated. The works of Silva et al. [70] and Nóbrega et al. [54] were extended respectively in [21] and [53] to finite chain rings. The works of Kötter and Kschischang [38], and Gorla and Ravagnani [30] were extended in [31] to finite principal ideal rings.

Note that the works of [31], [21] and [53] allow to improve the error correction in random linear network coding over finite principal ideal rings. As in the case of finite fields, another method that one can use is rank-metric codes. Thus, in this thesis we focus on a problem raised by Frank R. Kschischang which consists of studying properties of rank-metric codes likely to be preserved over finite principal ideal rings. The resolution of this problem will allow to give the encoding and decoding schemes for random linear network coding over finite principal ideal rings. Moreover, an optimal space-time block code will be constructed for all digital modulation systems whose signal constellation set is algebraically represented [22] by a finite principal ideal ring.

Our contribution

To extend rank-metric codes to finite principal ideal rings, we first extend the rank metric using the Smith normal form of a matrix. We then use the Galois extensions to prove that Gabidulin codes can be extended to finite principal ideal rings and that their properties are preserved. As in [46], we show that collaborative decoding of interleaved Gabidulin codes can be translated to the problem of reconstruction of skew polynomials. Analogous to [41], the theory of Gröbner bases is used to give an iterative algorithm to solve this reconstruction problem. The solutions of this problem allow us to give the unique decoding, minimal list decoding, and error-erasure decoding algorithms of interleaved Gabidulin codes. We then apply these results to space-time coding and random linear network coding. Specifically, we show that there is a rank-preserving map from a finite principal ideal ring to a complex signal set and we use it to construct an optimal space-time block code. We combine the encoding and decoding schemes of [69] and [70] to improve the error correction in random linear network coding..

Organization of the thesis

In Chapter 1, we recall some properties of matrices and modules over principal ideal rings. We show that the rank metric can be extended to principal ideal rings. We use the Galois extensions of finite principal ideal rings to give the vector representation of matrices. We also show that some properties of linearized polynomials over finite fields can be generalized to finite principal ideal rings. We review some facts about the theory of Gröbner bases of modules over skew polynomials.

In Chapter 2, we establish the rank-metric Singleton bound and prove that Gabidulin codes achieve this bound as in the case of finite fields. We describe the interleaved

Gabidulin codes, give the key equation and the algorithm to solve it. The decoding algorithms are given.

In Chapter 3, the applications in space-time codes and in random linear network coding are given.

We then present our conclusions and future research directions.

Chapter I

Preliminaries

In this chapter, we give mathematical tools that we will use to extend some results in rank-metric codes over finite commutative principal ideal rings. This chapter is organized as follows.

In Section 1.1, we describe finite chain rings and use the structure theorem for finite commutative rings to show that any finite commutative principal ideal ring can be decomposed as a direct sum of finite chain rings.

In Section 1.2, we define the Smith normal form and give a method to compute it in finite commutative principal ideal rings. We also show how to use the Smith normal form to solve a linear system of equations.

In Section 1.3, we use the Smith normal form to show that the rank metric can be extended to principal ideal rings.

In Section 1.4, we construct the Galois extension of finite principal ideal rings and use it to give the vector representation of matrices.

In Section 1.5, we show that some properties of linearized polynomials can be extended to finite principal ideal rings. We also give some properties of Gröbner bases of modules over skew polynomials that we will use to solve the key equation.

Throughout this thesis, by ring we mean a commutative ring with identity element, ring homomorphisms are assumed to be unitary, and all modules are unital. Unless otherwise specified, we assume that R is a finite principal ideal ring. An element $u \in R$ is called a **unit** if $uv = 1$ for some $v \in R$. Let $a, b \in R$, we say that a **divides** b, denoted $a|b$, if $b = ca$ for some $c \in R$. The set of all $m \times n$ matrices with entries from R will be denoted by $R^{m\times n}$. The $k \times k$ identity matrix is denoted by $\mathbf{I}_k$. Let $\mathbf{A} \in R^{m\times n}$, we denote by $row(\mathbf{A})$ and $col(\mathbf{A})$ the R-submodules generated by the row and column vectors of $\mathbf{A}$, respectively.

1.1 Finite chain rings

Definition 1.1 *[49] A* ***chain ring*** *is a ring whose ideals are linearly ordered by inclusion. A* ***local ring*** *is a ring with exactly one maximal ideal.*

Proposition 1.2 *[49] A finite ring is a chain ring if and only if it is a local principal ideal ring.*

Example 1.3 *Examples of finite chain rings are the ring $\mathbb{Z}_{p^k}$, p is a prime, and the ring $\mathbb{Z}_{2^k}[i]$, whose maximal ideals are $p\mathbb{Z}_{p^k}$ and $(1+i)\mathbb{Z}_{2^k}[i]$, respectively. Other examples of construction of finite chain rings using the ring of algebraic integers are given in [37].*

In a finite chain ring, every ideal is a power of the maximal ideal. More specifically we have the following:

Proposition 1.4 *[49]Assume that R is a finite chain ring, π a generator of its maximal ideal, ν the nilpotency index of π, i.e., the smallest positive integer such that $\pi^{\nu} = 0$. Then, every ideal of R is of the form $\pi^i R$, for $i = 0, \ldots, \nu$, and for all $a \in R$ there is a unique $i \in \{0, \ldots, \nu\}$ and a unit $u \in R$ such that $a = \pi^i u$.*

If $a = \pi^i u$ as in Proposition 1.4, then the integer i is denoted by $\nu_\pi(a)$. Thus, for all $a, b \in R$, a divides b if and only if $\nu_\pi(a) \leq \nu_\pi(b)$.

Definition 1.5 *1) A **Galois ring** of characteristic p^n and rank r, denoted by $GR(p^n, r)$, is the ring $\mathbb{Z}_{p^n}[X]/(f)$, where $f \in \mathbb{Z}_{p^n}[X]$ is a monic polynomial of degree r, irreducible modulo p and (f) denotes the ideal generated by f.*

*2) A polynomial $g(X) = X^s + p(a_{s-1}X^{s-1} + \cdots + a_1 X + a_0) \in GR(p^n, r)[X]$, where a_0 is a unit in $GR(p^n, r)$ is called an **Eisenstein polynomial** over $GR(p^n, r)$.*

Proposition 1.6 *[49] The Galois ring $GR(p^n, r)$ is a finite chain ring whose the maximal ideal is $pGR(p^n, r)$.*

The following theorem give a characterization of finite chain rings.

Theorem 1.7 *[49, Theorem XVII.5] Assume that R is a finite chain ring, ν the nilpotency index of the maximal ideal $\mathfrak{m}$ of R, the characteristic of R is p^n and $\mathbb{F}_{p^r} = R/\mathfrak{m}$. Then, there exist integers t and s such that*

$$R \cong GR(p^n, r)[X]/\left(g(X), p^{n-1}X^t\right)$$

where $t = \nu - (n-1)s > 0$ and $g(X)$ is an Eisenstein polynomial of degree s over $GR(p^n, r)$. Conversely, any such quatient ring is a finite chain ring.

The structure theorem for finite commutative rings [49, Theorem VI.2] says that each finite ring can be decomposed as a direct sum of finite local rings. Therefore, each finite principal ideal ring can be decomposed as a direct sum of finite chain rings. More specifically, we have the following:

Theorem 1.8 *[49, Theorem VI.2] There exist a positive integer ρ and finite chain rings $R_{(i)}$, for $i = 1, \ldots, \rho$, such that the finite principal ideal ring R is isomorphic to $R_{(1)} \times \cdots \times R_{(\rho)}$. Furthermore, this decomposition is unique up to permutation of direct summands.*

Example 1.9 *Let* $R = \mathbb{Z}_{12} = \mathbb{Z}/12\mathbb{Z}$, $R_{(1)} = \mathbb{Z}/3\mathbb{Z}$, $R_{(2)} = \mathbb{Z}/4\mathbb{Z}$. *The map*

$$\Phi : R \to R_{(1)} \times R_{(2)}$$

given by

$$x + 12\mathbb{Z} \longmapsto (x + 3\mathbb{Z}, x + 4\mathbb{Z})$$

is a ring isomorphism. The inverse morphism Φ^{-1} *is defined by*

$$(x + 3\mathbb{Z}, y + 4\mathbb{Z}) \longmapsto xe_1 + ye_2,$$

where $e_1 = 4 + 12\mathbb{Z}$ *and* $e_2 = 9 + 12\mathbb{Z}$.

1.2 Smith normal form

In [71], Smith proved that each matrix with integer coefficients can be reduced by elementary transformations into a diagonal matrix such that each diagonal element is a divisor of the next one. In [34], Kaplansky studied the rings in which this result can be generalized, especially the principal ideal rings. In [72], Storjohann gave an algorithm for computing the Smith normal form over principal ideal rings and its complexity. Each finite principal ideal ring can be decomposed as a direct sum of finite chain rings. Thus, one can also use the simple method given in the proof of [29, Theorem 1.1.12.] to compute the Smith normal form over finite chain rings. As in the proof of [9, Theorem 15.9], one can then compute the Smith normal form over finite principal ideal rings. The Smith normal form allow to solve a system of linear equations over principal ideal rings [12], [52]. As other application, we will use the Smith normal form to show that the rank metric can be extended to principal ideal rings.

1.2.1 Description

Definition 1.10 *[9] A matrix* $\mathbf{D} = (d_{i,j}) \in R^{m\times n}$ *is called a* ***diagonal matrix*** *if* $d_{i,j} = 0$ *whenever* $i \neq j$. *A diagonal matrix* $\mathbf{D} = (d_{i,j}) \in R^{m\times n}$ *can be written as* $\mathbf{D} = diag\,(d_1, \ldots, d_r)$, *where* $r = min\{n, m\}$, *and* $d_i = d_{i,i}$, *for* $i = 1, ..., r$.

Remark 1.11 *If* $m \leq n$, *then*

$$diag\,(d_1, \ldots, d_r) = \begin{pmatrix} d_1 & 0 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & d_2 & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots & & \vdots \\ 0 & \cdots & 0 & d_r & 0 & \cdots & 0 \end{pmatrix}$$

If $m \geq n$, then

$$diag\left(d_1, \ldots, d_r\right) = \begin{pmatrix} d_1 & 0 & \cdots & 0 \\ 0 & d_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & d_r \\ 0 & \cdots & 0 & 0 \\ \vdots & & \vdots & \vdots \\ 0 & \cdots & 0 & 0 \end{pmatrix}$$

By [9, Theorem 15.24], we have the following:

Theorem 1.12 *For all matrix $\mathbf{A} \in R^{m\times n}$, there are two invertible matrices $\mathbf{P}$, $\mathbf{Q}$, and a diagonal matrix $\mathbf{D} = diag\left(d_1, d_2, \ldots, d_r\right)$, satisfying the divisibility relations $d_1|d_2|\ldots|d_r$, such that $\mathbf{A} = \mathbf{PDQ}$. The elements $d_1, d_2, \ldots, d_r$ are unique up to associates.*

Definition 1.13 *The matrix $\mathbf{D}$, in Theorem 1.12, is called a **Smith normal form** of $\mathbf{A}$.*

1.2.2 Computing the Smith normal form over finite chain rings

We will give the steps that allow to compute the Smith normal form over finite chain rings. Assume that R is a finite chain ring, π a generator of its maximal ideal. Let $\mathbf{A} = (a_{i,j}) \in R^{m\times n}$. To compute the Smith normal form of $\mathbf{A}$ we can use the following steps given in the proof of [29, Theorem 1.1.12.].

1) **Choosing a pivot**

- Multiplying by permutation matrices as necessary, we may assume that

$$\alpha_1 := \nu_\pi\left(a_{1,1}\right) \leq \ \nu_\pi\left(a_{i,j}\right)$$

for all i, j.

- Multiplying the first row by a unit, we may assume that $a_{1,1} = \pi^{\alpha_1}$.

$$\begin{pmatrix} \pi^{\alpha_1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{pmatrix}$$

2) **Eliminating entries**

- Using elementary row and column operations as necessary, we can assume that $a_{1,j} = a_{i,1} = 0$ for $i, j \geq 2$.

$$\begin{pmatrix} \pi^{\alpha_1} & 0 & \cdots & 0 \\ 0 & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & & \vdots \\ 0 & a_{m,2} & \cdots & a_{m,n} \end{pmatrix}$$

3) **Iteration**

- Apply induction to the submatrix of $\mathbf{A}$ obtained by deleting the first row and column.

Note that the invertible matrices $\mathbf{P}$, $\mathbf{Q}$ such that $\mathbf{PAQ} = \mathbf{D}$ where $\mathbf{D}$ is a Smith normal form of $\mathbf{A}$ can be computed simultaneously by applying the same row operations on the matrix $\mathbf{I}_m$ and the same column operations on the matrix $\mathbf{I}_n$.

Example 1.14 *Let*

$$\mathbf{A} = \begin{pmatrix} 0 & 2 & 0 & 0 \\ 0 & 2 & 0 & 2 \\ 3 & 2 & 0 & 2 \end{pmatrix}$$

be a matrix with coefficients in $\mathbb{Z}_4$.

Step 0: *initialization*

$\mathbf{D} = \mathbf{A}$, $\mathbf{P} = \mathbf{I}_3$, $\mathbf{Q} = \mathbf{I}_4$.

Step 1: $\mathbf{L}_1 \longleftrightarrow \mathbf{L}_3$ *(exchange the first row with last row)*

$$\mathbf{D} = \begin{pmatrix} 3 & 2 & 0 & 2 \\ 0 & 2 & 0 & 2 \\ 0 & 2 & 0 & 0 \end{pmatrix}, \quad \mathbf{P} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \mathbf{Q} = \mathbf{I}_4.$$

Step 2: $\mathbf{L}_1 \longleftarrow 3\mathbf{L}_1$ *(multiplying the first row by* 3*)*

$$\mathbf{D} = \begin{pmatrix} 1 & 2 & 0 & 2 \\ 0 & 2 & 0 & 2 \\ 0 & 2 & 0 & 0 \end{pmatrix}, \quad \mathbf{P} = \begin{pmatrix} 0 & 0 & 3 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \mathbf{Q} = \mathbf{I}_4$$

Step 3: $\mathbf{C}_2 \longleftarrow \mathbf{C}_2 - 2\mathbf{C}_1$; $\mathbf{C}_4 \longleftarrow \mathbf{C}_4 - 2\mathbf{C}_1$ *(column operations)*

$$\mathbf{D} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 2 \\ 0 & 2 & 0 & 0 \end{pmatrix}, \quad \mathbf{P} = \begin{pmatrix} 0 & 0 & 3 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \quad \mathbf{Q} = \begin{pmatrix} 1 & 2 & 0 & 2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Step 4: $\mathbf{L}_3 \longleftarrow \mathbf{L}_3 - \mathbf{L}_2$

$$\mathbf{D} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 2 \\ 0 & 0 & 0 & 2 \end{pmatrix}, \quad \mathbf{P} = \begin{pmatrix} 0 & 0 & 3 \\ 0 & 1 & 0 \\ 1 & 3 & 0 \end{pmatrix}, \quad \mathbf{Q} = \begin{pmatrix} 1 & 2 & 0 & 2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Step 5: $\mathbf{C}_4 \longleftarrow \mathbf{C}_4 - \mathbf{C}_2$

$$\mathbf{D} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix}, \quad \mathbf{P} = \begin{pmatrix} 0 & 0 & 3 \\ 0 & 1 & 0 \\ 1 & 3 & 0 \end{pmatrix}, \quad \mathbf{Q} = \begin{pmatrix} 1 & 2 & 0 & 0 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Step 6: $\mathbf{C}_3 \longleftrightarrow \mathbf{C}_4$

$$\mathbf{D} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \end{pmatrix}, \quad \mathbf{P} = \begin{pmatrix} 0 & 0 & 3 \\ 0 & 1 & 0 \\ 1 & 3 & 0 \end{pmatrix}, \quad \mathbf{Q} = \begin{pmatrix} 1 & 2 & 0 & 0 \\ 0 & 1 & 3 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Thus, $\mathbf{D}$ *is a Smith normal form of* $\mathbf{A}$ *and* $\mathbf{PAQ} = \mathbf{D}$.

1.2.3 Computing the Smith normal form over finite principal ideal rings

By Theorem 1.8, there is a ring isomorphism $\Phi : R \longrightarrow R_{(1)} \times \cdots \times R_{(\rho)}$. Let $\Phi_i : R \longrightarrow R_{(i)}$ be the composition of Φ and the i-th projection map $R_{(1)} \times \cdots \times R_{(\rho)} \longrightarrow R_{(i)}$, for $i = 1, \ldots, \rho$. We extend Φ coefficient-by-coefficient as a map from $R^{m\times n}$ to $R_{(1)}^{m\times n} \times \cdots \times R_{(\rho)}^{m\times n}$. We also extend Φ_i coefficient-by-coefficient as a map from $R^{m\times n}$ to $R_{(i)}^{m\times n}$. As in the proof of [9, Theorem 15.9], we have the following:

Proposition 1.15 *Let* $\mathbf{A} \in R^{m\times n}$. *Set* $\mathbf{A}_{(i)} := \Phi_i(\mathbf{A}) \in R_{(i)}^{m\times n}$ *, for* $i = 1, \ldots, \rho$. *Let* $\mathbf{D}_{(i)} \in R_{(i)}^{m\times n}$ *be a Smith normal form of* $\mathbf{A}_{(i)}$ *and let the invertible matrices* $\mathbf{P}_{(i)}$, $\mathbf{Q}_{(i)}$ *with coefficients in* $R_{(i)}$ *such that* $\mathbf{A}_{(i)} = \mathbf{P}_{(i)}\mathbf{D}_{(i)}\mathbf{Q}_{(i)}$, *for* $i = 1, \ldots, \rho$. *Set*

$$\mathbf{D} = \Phi^{-1}\left(\left(\mathbf{D}_{(1)}, \ldots, \mathbf{D}_{(\rho)}\right)\right),$$

$$\mathbf{P} = \Phi^{-1}\left(\left(\mathbf{P}_{(1)}, \ldots, \mathbf{P}_{(\rho)}\right)\right),$$

and

$$\mathbf{Q} = \Phi^{-1}\left(\left(\mathbf{Q}_{(1)}, \ldots, \mathbf{Q}_{(\rho)}\right)\right).$$

Then, the matrices $\mathbf{P}$, $\mathbf{Q}$ *are invertible,* $\mathbf{A} = \mathbf{PDQ}$, *and* $\mathbf{D}$ *is a Smith normal form of* $\mathbf{A}$.

Thus, the computation of the Smith normal form over finite principal ideal rings is reduced to the computation over finite chain rings.

Example 1.16 *Consider the isomorphism* $\Phi : R \longrightarrow R_{(1)} \times R_{(2)}$ *given in Example 1.9. Let*

$$\mathbf{A} = \begin{pmatrix} 8 & 10 & 4 & 4 \\ 4 & 2 & 8 & 2 \\ 11 & 6 & 0 & 6 \end{pmatrix}$$

be a matrix with coefficients in R. *The image of* $\mathbf{A}$ *in* $R_{(1)}$ *is*

$$\mathbf{A}_{(1)} = \begin{pmatrix} 2 & 1 & 1 & 1 \\ 1 & 2 & 2 & 2 \\ 2 & 0 & 0 & 0 \end{pmatrix}$$

and the image of $\mathbf{A}$ *in* $R_{(2)}$ *is*

$$\mathbf{A}_{(2)} = \begin{pmatrix} 0 & 2 & 0 & 0 \\ 0 & 2 & 0 & 2 \\ 3 & 2 & 0 & 2 \end{pmatrix}.$$

By Example 1.14, $\mathbf{P}_{(2)}\mathbf{A}_{(2)}\mathbf{Q}_{(2)} = \mathbf{D}_{(2)}$ *where*

$$\mathbf{D}_{(2)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \end{pmatrix}, \quad \mathbf{P}_{(2)} = \begin{pmatrix} 0 & 0 & 3 \\ 0 & 1 & 0 \\ 1 & 3 & 0 \end{pmatrix}, \quad \mathbf{Q}_{(2)} = \begin{pmatrix} 1 & 2 & 0 & 0 \\ 0 & 1 & 3 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

We also have $\mathbf{P}_{(1)}\mathbf{A}_{(1)}\mathbf{Q}_{(1)} = \mathbf{D}_{(1)}$ *where*

$$\mathbf{D}_{(1)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{P}_{(1)} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}, \quad \mathbf{Q}_{(1)} = \begin{pmatrix} 2 & 2 & 0 & 0 \\ 0 & 2 & 2 & 0 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Using Φ^{-1}*, we get* $\mathbf{PAQ} = \mathbf{D}$ *where*

$$\mathbf{D} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 6 & 0 \end{pmatrix}, \quad \mathbf{P} = \begin{pmatrix} 4 & 0 & 3 \\ 4 & 3 & 8 \\ 1 & 7 & 0 \end{pmatrix}, \quad \mathbf{Q} = \begin{pmatrix} 5 & 2 & 0 & 0 \\ 0 & 5 & 11 & 0 \\ 0 & 0 & 4 & 5 \\ 0 & 0 & 9 & 4 \end{pmatrix}.$$

1.2.4 System of linear equations

As in [12] and [52], we will show how to use the Smith normal form to solve a system of linear equations in R. A general system of m linear equations with n unknowns can be written as

$$\begin{cases} a_{1,1}x_1 + \cdots + a_{1,n}x_n = b_1 \\ a_{2,1}x_1 + \cdots + a_{2,n}x_n = b_2 \\ \qquad\qquad \vdots \\ a_{m,1}x_1 + \cdots + a_{m,n}x_n = b_m \end{cases} \tag{1.1}$$

where $x_1, \ldots, x_n$ are the unknowns, $a_{1,1}, a_{1,2} \ldots, a_{m,n}$ are the coefficients of the system, and $b_1, \ldots, b_m$ are the constant terms.

Equation (1.1) is equivalent to a matrix equation of the form

$$\mathbf{Ax} = \mathbf{b} \tag{1.2}$$

where

$$\mathbf{A} = \begin{pmatrix} a_{1,1} & \cdots & a_{1,n} \\ \vdots & \ddots & \vdots \\ a_{m,1} & \cdots & a_{m,n} \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \text{ and } \mathbf{b} = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}.$$

Let $\mathbf{D} = diag\,(d_1, \ldots, d_r)$ be a Smith normal form of $\mathbf{A}$ and the invertible matrices $\mathbf{P}$, $\mathbf{Q}$ such that $\mathbf{PAQ} = \mathbf{D}$. Then, Equation (1.2) is equivalent to

$$\mathbf{Dy} = \mathbf{c}$$

where $\mathbf{y} = \mathbf{Q}^{-1}\mathbf{x}$ and $\mathbf{c} = \mathbf{Pb}$.

Thus, the necessary and sufficient conditions such that Equation (1.1) has a solution are as follows :

$$d_i \text{ must divide } c_i, \text{ for } i = 1, \ldots, r, \text{ and } \; c_i = 0, \text{ for } i > r.$$

Example 1.17 *Let* $\mathbf{A}$ *be the matrix given in Example 1.16. Consider the equation*

$$\mathbf{Ax} = \mathbf{b} \tag{1.3}$$

where

$$\mathbf{b} = \begin{pmatrix} 2 \\ 4 \\ 7 \end{pmatrix}$$

Since $\mathbf{PAQ} = \mathbf{D}$ *where*

$$\mathbf{D} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 6 & 0 \end{pmatrix}, \quad \mathbf{P} = \begin{pmatrix} 4 & 0 & 3 \\ 4 & 3 & 8 \\ 1 & 7 & 0 \end{pmatrix}, \quad \mathbf{Q} = \begin{pmatrix} 5 & 2 & 0 & 0 \\ 0 & 5 & 11 & 0 \\ 0 & 0 & 4 & 5 \\ 0 & 0 & 9 & 4 \end{pmatrix},$$

Equation (1.3) is equivalent to

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 6 & 0 \end{pmatrix} \mathbf{y} = \begin{pmatrix} 5 \\ 4 \\ 6 \end{pmatrix} \tag{1.4}$$

where $\mathbf{y} = \mathbf{Q}^{-1}\mathbf{x}$. *A solution of (1.4) is* $(5, 2, 1, 0)$. *Thus, a solution of (1.3) is* $(5, 9, 4, 9)$.

1.3 Rank metric

In field theory, the rank of a matrix defines a group-norm in the matrix space of the same size. In this subsection, we use the Smith normal form to extend this property to principal ideal rings. Let M be a finitely generated R-module. Let $\{\mathbf{a}_1, \dots, \mathbf{a}_r\}$ be a subset of M. The R-submodule of M generated by $\{\mathbf{a}_1, \dots, \mathbf{a}_r\}$ is denoted by $\langle \mathbf{a}_1, \dots, \mathbf{a}_r \rangle_R$. Recall that $\{\mathbf{a}_1, \dots, \mathbf{a}_r\}$ is **linearly independent** over R if whenever $\alpha_1 \mathbf{a}_1 + \cdots + \alpha_r \mathbf{a}_r = \mathbf{0}$ for some $\alpha_1, \dots, \alpha_r \in R$, then $\alpha_1 = 0, \dots, \alpha_r = 0$. If $\{\mathbf{a}_1, \dots, \mathbf{a}_r\}$ is linearly independent, then we say that it is a **free base** of the free module $\langle \mathbf{a}_1, \dots, \mathbf{a}_r \rangle_R$. As in [9, page 190] we use the following notation.

Notation 1.18 *Let* M *be a finitely generated* R*-module. The smallest number of elements in* M *which generate* M *as an* R*-module is denoted by* $\mu_R(M)$. *If* $M = \{0\}$, *then we set* $\mu_R(M) = 0$.

Lemma 1.19 *[43] Let* F *be a finitely generated free* R*-module and* $\{e_1, \dots, e_n\}$ *be a free basis of* F. *Then,* $\mu_R(F) = n$ *and any generating set of* F *consisting of* n *elements is a free basis of* F.

Proposition 1.20 *Let* M *be a finitely generated* R*-module,* $\mu_R(M) := r_M$, *and let* N *be a submodule of* M, $\mu_R(N) := r_N$. *Then,* $r_N \leq r_M$ *and there is a generating set* $\{u_i\}_{1 \leq i \leq r_M}$ *of* M *and* r_N *scalars* $d_1, \dots, d_{r_N}$ *of* R *such that* $\{d_i u_i\}_{1 \leq i \leq r_N}$ *generates* N, *with* $d_1 | d_2 | \dots | d_{r_N}$. *Furthermore, if* M *is a free module then* $\{u_i\}_{1 \leq i \leq r_M}$ *is a free basis of* M.

Proof. Let $\{y_i\}_{1\leq i\leq r_N}$ be a generating set of N and $\{x_i\}_{1\leq i\leq r_M}$ be a generating set of M. Then, since N is a submodule of M, there is a matrix $\mathbf{A}\in R^{r_M\times r_N}$ such that

$$(y_1,\ldots,y_{r_N})=(x_1,\ldots,x_{r_M})\,\mathbf{A}.$$

Let $\mathbf{D}=diag\,(d_1,\ldots,d_r)$ be a Smith normal form of $\mathbf{A}$ and $\mathbf{P}$, $\mathbf{Q}$ be the invertible matrices such that $\mathbf{A}=\mathbf{PDQ}$. Set

$$(u_1,\ldots u_{r_M})=(x_1,\ldots,x_{r_M})\,\mathbf{P}$$

and

$$(v_1,\ldots,v_{r_N})=(y_1,\ldots,y_{r_N})\,\mathbf{Q}^{-1}.$$

Then $\{u_i\}_{1\leq i\leq r_M}$ and $\{v_i\}_{1\leq i\leq r_N}$ are respectively the generating sets of M and N, and we have $v_i=d_iu_i$, for $i=1,\ldots,r$. Thus, $r=r_N\leq r_M$. If M is a free module, then $\{u_i\}_{1\leq i\leq r_M}$ is a free basis of M, by Lemma 1.19. ■

Note that if N and N' are two submodules of a finitely generated R-module, then $\mu_R\,(N+N')\leq\mu_R\,(N)+\mu_R\,(N')$. Thus, the minimum number of generators of a module over a principal ideal ring has several properties similar to the dimension of vector spaces. Therefore, analogous to the case of fields, we give the following definition.

Definition 1.21 *(Rank of matrix). Let $\mathbf{A}\in R^{m\times n}$.*

*(i) The **rank** of $\mathbf{A}$, denoted by $rank_R\,(\mathbf{A})$, or simply by $rank\,(\mathbf{A})$, is the number $\mu_R\,(col\,(\mathbf{A}))$.*

*(ii) The **free rank** of $\mathbf{A}$, denoted by $freerank_R\,(\mathbf{A})$ or simply by $freerank\,(\mathbf{A})$, is the maximum of the ranks of free R-submodules of $col\,(\mathbf{A})$.*

Lemma 1.22 *[9, Lemma 15.12] Suppose $I_1,\ldots,I_n$ are ideals in R such that*

$$I_1+\cdots+I_n\neq R.$$

Then

$$\mu_R\,(R/I_1\times\cdots\times R/I_n)=n.$$

Lemma 1.23 *[9, Theorem 15.33] Let M be a finitely generated R-module. Then*

$$M\cong(R/a_1R)\times\cdots\times(R/a_nR)$$

with $a_1R\subset a_2R\subset\cdots\subset a_nR$. Furthermore, if no summand R/a_iR is zero here, then this decomposition is unique.

Proposition 1.24 *Let $\mathbf{A}\in R^{m\times n}\backslash\{\mathbf{0}\}$ and $\mathbf{D}=diag\,(d_1,\ldots,d_r)$ be a Smith normal form of $\mathbf{A}$. Then,*

$$col\,(\mathbf{A})\cong row\,(\mathbf{A}),$$

$$rank\,(\mathbf{A})=\max\{i\in\{1,\ldots,r\}:d_i\neq 0\},$$

and

$$freerank\,(\mathbf{A})=\max\{i\in\{1,\ldots,r\}:d_i\text{ is a unit}\}.$$

Proof. Let $\mathbf{P}$ and $\mathbf{Q}$ be the invertible matrices such that $\mathbf{A} = \mathbf{PDQ}$. Set

$$s = \max\{i \in \{1, \ldots, r\} : d_i \neq 0\},$$

and

$$M = d_1 R \times \cdots \times d_s R.$$

Then,

$$row\,(\mathbf{A}) = row\,(\mathbf{DQ}) \cong M$$

and

$$col\,(\mathbf{A}) = col\,(\mathbf{PD}) \cong M.$$

Since R is a principal ideal ring, there is $c_i \in R$ such that $d_i R \cong R/c_i R$, for $i = 1, \ldots, s$. As $d_1|d_2|\ldots|d_s$, we have $c_1 R \subset c_2 R \subset \cdots \subset c_s R$. Thus, by Lemma 1.22, $\mu_R(M) = s$.

Let

$$t = \max\{i \in \{1, \ldots, r\} : d_i \text{ is a unit}\}.$$

Assume that $t \neq 0$. Then $c_i = 0$, for $i = 1, \ldots, t$, so

$$col\,(\mathbf{A}) \cong R^t \times (R/c_{t+1}R) \times \cdots \times (R/c_s R).$$

Let F be a free submodule of $col\,(\mathbf{A})$ such that

$$u := \mu_R(F) = freerank\,(col\,(\mathbf{A})).$$

Then, since R is a Frobenius ring, F is an injective module [43]. So, $col\,(\mathbf{A}) = F \oplus N$ where N is a submodule of $col\,(\mathbf{A})$. By Lemma 1.23,

$$N \cong (R/b_1 R) \times \cdots \times (R/b_v R)$$

with $b_v|b_{v-1}|\cdots|b_1$. Thus,

$$col\,(\mathbf{A}) \cong R^u \times (R/b_1 R) \times \cdots \times (R/b_v R).$$

Consequently $t = u$, by Lemma 1.23. ■

Corollary 1.25 *Let $\mathbf{A} \in R^{m \times n}$. We have $rank_R(\mathbf{A}) = \mu_R(row\,(\mathbf{A}))$ and $freerank_R(\mathbf{A})$ is the maximum of the ranks of free R-submodules of $row\,(\mathbf{A})$.*

Example 1.26 *If $\mathbf{A}$ is the matrix given in Example 1.16, then $rank\,(\mathbf{A}) = 3$ and $freerank\,(\mathbf{A}) = 1$.*

Remark 1.27 *In linear algebra over fields, the rank-nullity theorem states that the sum of the rank of a matrix and the dimension of its right kernel is equal to the number of its columns. Using the definition of rank given in Definition 1.21, this property is not true in general over finite principal ideal rings, due to zero divisors. Indeed, let $\mathbb{Z}_6$ be the ring of integers modulo 6 and*

$$\mathbf{A} = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$$

be a matrix with coefficients in $\mathbb{Z}_6$. The right kernel of $\mathbf{A}$ is generated by the vectors $(3, 0)$ and $(0, 3)$. By Theorem 1.29, $rank\,(\mathbf{A}) = 2$. Thus, the rank-nullity theorem can not be applied to the matrix $\mathbf{A}$.

Proposition 1.28 *(Rank Decompositions). Let* $\mathbf{E} \in R^{m\times n}$, $rank(\mathbf{E}) = t$.

1) There are $\mathbf{A} \in R^{m\times t}$, $rank(\mathbf{A}) = t$, *and* $\mathbf{B} \in R^{t\times n}$, $freerank(\mathbf{B}) = t$, *such that* $\mathbf{E} = \mathbf{AB}$.

2) There are $\mathbf{A}' \in R^{m\times t}$, $freerank(\mathbf{A}') = t$, *and* $\mathbf{B}' \in R^{t\times n}$, $rank(\mathbf{B}') = t$, *such that* $\mathbf{E} = \mathbf{A}'\mathbf{B}'$.

Proof. Let $\mathbf{D} = diag(d_1, \ldots, d_r)$ be a Smith normal form of $\mathbf{E}$ and $\mathbf{P}$, $\mathbf{Q}$ be the invertible matrices such that $\mathbf{E} = \mathbf{PDQ}$.

1) Set

$$\mathbf{D} = \begin{pmatrix} \mathbf{D}_1 & \mathbf{D}_2 \end{pmatrix}$$

where $\mathbf{D}_1$ and $\mathbf{D}_2$ are the submatrices of $\mathbf{D}$ of sizes $m \times t$, and $m \times (n-t)$, respectively. Set

$$\mathbf{Q} = \begin{pmatrix} \mathbf{Q}_1 \\ \mathbf{Q}_2 \end{pmatrix}$$

where $\mathbf{Q}_1$ and $\mathbf{Q}_2$ are the submatrices of $\mathbf{Q}$ of sizes $t \times n$, and $(n-t) \times n$, respectively. Then

$$\mathbf{E} = \mathbf{AB}$$

where $\mathbf{A} = \mathbf{PD}_1$ and $\mathbf{B} = \mathbf{Q}_1$. By Proposition 1.24, $rank(\mathbf{A}) = t$ and $freerank(\mathbf{B}) = t$.

2) This result can be proved as above using the column decomposition of $\mathbf{P}$. ■

The following theorem extends the notion of rank metric to principal ideal rings.

Theorem 1.29 *The map* $R^{m\times n} \to \mathbb{N}$ *given by* $\mathbf{A} \mapsto rank(\mathbf{A})$ *is a group-norm, i.e.,*

(i) for all $\mathbf{A} \in R^{m\times n}$, $rank(\mathbf{A}) = 0$ *if and only if* $\mathbf{A} = \mathbf{0}$;

(ii) for all $\mathbf{A} \in R^{m\times n}$, $rank(-\mathbf{A}) = rank(\mathbf{A})$;

(iii) for all $\mathbf{A}, \mathbf{B} \in R^{m\times n}$,

$$rank(\mathbf{A} + \mathbf{B}) \leq rank(\mathbf{A}) + rank(\mathbf{B}).$$

Proof. (i) and (ii) are straightforward. Proof of (iii): let $\mathbf{A}, \mathbf{B} \in R^{m\times n}$, then

$$col(\mathbf{A} + \mathbf{B}) \subset col(\mathbf{A}) + col(\mathbf{B}).$$

Hence, by Proposition 1.20,

$$\mu_R(col(\mathbf{A} + \mathbf{B})) \leq \mu_R(col(\mathbf{A}) + col(\mathbf{B})).$$

But by the definition of μ_R, we have

$$\mu_R(col(\mathbf{A}) + col(\mathbf{B})) \leq \mu_R(col(\mathbf{A})) + \mu_R(col(\mathbf{B})).$$

Thus, $rank(\mathbf{A} + \mathbf{B}) \leq rank(\mathbf{A}) + rank(\mathbf{B})$. ■

Corollary 1.30 *The map* $R^{m\times n} \times R^{m\times n} \to \mathbb{N}$ *given by* $(\mathbf{A}, \mathbf{B}) \mapsto rank(\mathbf{A} - \mathbf{B})$ *is a metric.*

Remark 1.31 *In general, freerank does not satisfy conditions (i) and (iii) of Theorem 1.29.*

1.4 Galois extensions of finite principal ideal rings

In [24], Gabidulin used Galois extensions of finite fields to give a vector representation of rank-metric codes. In [5], Auslander and Goldman introduced the notion of Galois extension of commutative rings. In [14], Chase, Harrison, and Rosenberg generalized the classical Galois correspondence theorem from fields to commutative rings. In [28], Ganske and McDonald studied the Galois theory of finite local commutative rings. In this section, we show that every finite principal ideal rings admits the Galois extension of any order. We then use this result to give a vector representation of matrices as in the case of finite fields.

1.4.1 Galois extensions

Definition 1.32 *[17] Let F be a ring extension of a ring K and let G be a finite group of automorphisms of F. The ring F is called a* ***Galois extension*** *of K with Galois group G if :*

(i) $F^G = K$, where $F^G = \{x \in F : \tau(x) = x, \ \forall \tau \in G \}$;

(ii) for each maximal ideal M of F and for each $\tau \in G \setminus \{id_G\}$ there is an $x \in F$ with $\tau(x) - x \notin M$.

By Theorem 1.8, $R \cong R_{(1)} \times \cdots \times R_{(\rho)}$. In the following, we identify R with $R_{(1)} \times \cdots \times R_{(\rho)}$. Let $i \in \{1, \ldots, \rho\}$, we denote by $\mathfrak{m}_{(i)}$ the maximal ideal of $R_{(i)}$, $\mathbb{F}_{q_{(i)}} = R_{(i)}/\mathfrak{m}_{(i)}$ its residue field and $\nu_{(i)}$ the nilpotency index of $\mathfrak{m}_{(i)}$. We denote the natural projection $R_{(i)} \to \mathbb{F}_{q_{(i)}}$ by $\psi_{(i)}$. We extend $\psi_{(i)}$ coefficient-by-coefficient to polynomials over $R_{(i)}$.

Let m be a nonzero positive integer. Let $i \in \{1, \ldots, \rho\}$ and $h_{(i)} \in R_{(i)}[X]$ be a monic polynomial of degree m such that $\psi_{(i)}\left(h_{(i)}\right)$ is irreducible in $\mathbb{F}_{q_{(i)}}[X]$. Set $S_{(i)} = R_{(i)}[X] / \left(h_{(i)}\right)$, where $\left(h_{(i)}\right)$ denotes the ideal generated by $h_{(i)}$. By [49], $S_{(i)}$ is a free local Galois extension of $R_{(i)}$ of $R_{(i)}$-dimension m, with the maximal ideal $\mathfrak{M}_{(i)} = \mathfrak{m}_{(i)} S_{(i)}$, where the Galois group is cyclic of order m, generated by a power map $\sigma_{(i)} : \alpha_{(i)} \mapsto \alpha_{(i)}^{q_{(i)}}$ on a suitable primitive element $\alpha_{(i)}$. Moreover, $\mathbb{F}_{q_{(i)}^m} = S_{(i)}/\mathfrak{M}_{(i)}$.

Set $S = S_{(1)} \times \cdots \times S_{(\rho)}$ and $\sigma = \left(\sigma_{(i)}\right)_{1 \leq i \leq \rho}$. Let $G_R(S)$ be the group generated by σ.

Proposition 1.33 *With the above notations, the ring S is a Galois extension of R with Galois group $G_R(S)$.*

Proof. Let $\theta = \left(\theta_{(i)}\right)_{1 \leq i \leq \rho} \in G_R(S)$ and $x = \left(x_{(i)}\right)_{1 \leq i \leq \rho} \in S$ such that $\theta(x) = x$. Then, for $i = 0, \ldots, \rho$, $\theta_{(i)}\left(x_{(i)}\right) = x_{(i)}$, consequently $x_{(i)} \in R_{(i)}$, thus $S^{G_R(S)} = R$. Let $\tau = \left(\tau_{(i)}\right)_{1 \leq i \leq \rho} \in G_R(S) \setminus \{id\}$ and let M be a maximal ideal of S, then there is $i_0 \in \{1, \ldots, \rho\}$ such that

$$M = S_{(1)} \times \cdots \times S_{(i_0 - 1)} \times M_{(i_0)} \times S_{(i_0+1)} \times \cdots \times S_{(\rho)},$$

where $M_{(i_0)}$ is a maximal ideal of $S_{(i_0)}$. Since $\tau_{(i_0)} \neq id$ and $S_{(i_0)}$ is the Galois extension of $R_{(i_0)}$, there is $x_{(i_0)} \in S_{(i_0)}$ such that $\tau_{(i_0)}\left(x_{(i_0)}\right) - x_{(i_0)} \notin M_{(i_0)}$. Set $y = \left(y_{(i)}\right)_{1 \leq i \leq \rho}$ where $y_{(i_0)} = x_{(i_0)}$ and $y_{(i)} = 0$ if $i \neq i_0$, then we have $\tau(y) - y \notin M$. ■

Remark 1.34 *1) Since $S_{(i)}$ is a free $R_{(i)}$-module of rank m, then S is a free R-module of rank m.*

2) Since $R_{(i)}$ is a finite chain ring, then $S_{(i)}$ is also a finite chain ring.

3) Since $S_{(i)}$ is a finite chain ring, then S is a finite principal ideal ring.

Proposition 1.35 *[15, Theorem 3.2.] There is a monic polynomial $h \in R[X]$ of degree m such that $S \cong R[X]/(h)$.*

Example 1.36 *Consider the isomorphism $\Phi : R \longrightarrow R_{(1)} \times R_{(2)}$ given in Example 1.9. We will construct a Galois extension of R of dimension 4. Set*

$$h_{(1)} = X^4 + 2X^3 + 2 \in R_{(1)}[X],$$

$$h_{(2)} = X^4 + 2X^2 + 3X + 1 \in R_{(2)}[X],$$

$$S_{(1)} = R_{(1)}[X]/\left(h_{(1)}\right),$$

$$S_{(2)} = R_{(2)}[X]/\left(h_{(2)}\right),$$

$$\alpha_{(1)} = X + \left(h_{(1)}\right),$$

$$\alpha_{(2)} = X + \left(h_{(2)}\right).$$

Let the maps $\sigma_{(1)} : S_{(1)} \to S_{(1)}$ given by $\sigma_{(1)}(x) = x^3$, for all $x \in S_{(1)}$, and $\sigma_{(2)} : S_{(2)} \to S_{(2)}$ given by $\alpha_{(2)} \mapsto \alpha_{(2)}^2$, that is, for all

$$x = x_0 + x_1\alpha_{(2)} + x_2\alpha_{(2)}^2 + x_3\alpha_{(2)}^3 \in S_{(2)},$$

where $x_0,\ x_1,\ x_2,\ x_3 \in R_{(2)}$,

$$\sigma_{(2)}(x) = x_0 + x_1\alpha_{(2)}^2 + x_2\alpha_{(2)}^4 + x_3\alpha_{(2)}^6.$$

Then, $S_{(1)} \times S_{(2)}$ is a Galois extension of $R_{(1)} \times R_{(2)}$ where the Galois group is generated by $\left(\sigma_{(1)}, \sigma_{(2)}\right)$. We extend Φ^{-1} coefficient-by-coefficient to $R_{(1)}[X] \times R_{(2)}[X]$. Set

$$h = \Phi^{-1}\left(h_{(1)}, h_{(2)}\right) = X^4 + 8X^3 + 6X^2 + 3X + 5,$$

$$S = R[X]/(h),$$

$$\alpha = X + (h).$$

Then, by [15, Theorem 3.2.], $S \cong S_{(1)} \times S_{(2)}$, $S_{(1)} \cong 4S$, $S_{(2)} \cong 9S$ and $S = 4S \oplus 9S$. Thus, S is a Galois extension of R where the Galois group is generated by a power map $\alpha \mapsto 4\alpha^3 + 9\alpha^2$.

1.4.2 Vector representation of matrices

In this subsection, we define the group-norm in S^n that will allow to give an R-isomorphic isometry between S^n and $R^{m\times n}$.

Definition 1.37 *Let $\mathbf{u} = (u_1, \ldots, u_n) \in S^n$. By considering S as R-module, the number $\mu_R(\langle\{u_1, \ldots, u_n\}\rangle)$ is called the* ***rank*** *of $\mathbf{u}$ and denoted by $rank_R(\mathbf{u})$ or simply by $rank(\mathbf{u})$. Where $\langle\{u_1, \ldots, u_n\}\rangle$ denotes the R-submodule of S generated by $\{u_1, \ldots, u_n\}$.*

Remark 1.38 *Using the same arguments as in the proof of Theorem 1.29, we can show that the map $rank : S^n \to \mathbb{N}$ given by $\mathbf{u} \mapsto rank(\mathbf{u})$ is a group-norm.*

The following proposition gives a relation between Definition 1.21 and Definition 1.37. Let $(\beta_1, \ldots, \beta_m)$ be a free basis of S as R-module. Consider $\mathbf{a} = (a_1, \ldots, a_n) \in S^n$. For $j = 1, \ldots, n$, a_j can be written as

$$a_j = \sum_{1\leq i\leq m} a_{i,j}\beta_i,$$

where $a_{i,j} \in R$. The matrix

$$\mathbf{A_a} := (a_{i,j})_{1\leq i\leq m,\ 1\leq j\leq n}$$

is the matrix representation of $\mathbf{a}$ in the basis $(\beta_1, \ldots, \beta_m)$ over R.

Proposition 1.39 *With the above notations, the map $S^n \to R^{m\times n}$ given by $\mathbf{a} \mapsto \mathbf{A_a}$ is an R-isomorphic isometry between the normed spaces $(S^n, rank)$ and $(R^{m\times n}, rank)$.*

Proof. Let $\mathbf{a}, \mathbf{b} \in S^n$ and $\lambda \in R$. We have

$$\mathbf{a} = (\beta_1, \ldots, \beta_m)\,\mathbf{A_a}$$

and

$$\mathbf{b} = (\beta_1, \ldots, \beta_m)\,\mathbf{A_b}.$$

Therefore,

$$\mathbf{A_{a+\lambda b}} = \mathbf{A_a} + \lambda\mathbf{A_b}.$$

We now prove that $rank(\mathbf{a}) = rank(\mathbf{A_a})$. Let $r = rank(\mathbf{a})$, then by Proposition 1.20, there are r scalars $d_1, \ldots, d_r$ of R such that $\{d_i\beta_i\}_{1\leq i\leq r}$ generates $\langle\{a_1, \ldots, a_n\}\rangle$, with $d_1|d_2|\ldots|d_r$. Thus, there are $\mathbf{B} \in R^{n\times r}$ and $\mathbf{C} \in R^{r\times n}$ such that

$$(a_1, \ldots, a_n) = (d_1\beta_1, \ldots, d_r\beta_r)\,\mathbf{B}$$

and

$$(d_1\beta_1, \ldots, d_r\beta_r) = (a_1, \ldots, a_n)\,\mathbf{C}.$$

Let $\mathbf{D} \in R^{m\times r}$ such that

$$\mathbf{D} = \begin{pmatrix} d_1 & 0 & \cdots & 0 \\ 0 & d_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & d_r \\ 0 & \cdots & 0 & 0 \\ \vdots & & \vdots & \vdots \\ 0 & \cdots & 0 & 0 \end{pmatrix}$$

We have $\mathbf{A_a} = \mathbf{DB}$ and $\mathbf{D} = \mathbf{A_a C}$. Consequently,

$$col\,(\mathbf{A_a}) = col\,(\mathbf{DB}) \subset col\,(\mathbf{D})$$

and

$$col\,(\mathbf{D}) = col\,(\mathbf{A_a C}) \subset col\,(\mathbf{A_a})\,.$$

Thus, $col\,(\mathbf{A_a}) = col\,(\mathbf{D})$ and, by Proposition 1.24, $rank\,(\mathbf{A_a}) = r$. ■

Proposition 1.28 can be interpreted in vector representation as follows.

Proposition 1.40 *Let $\mathbf{u} \in S^n$, $rank\,(\mathbf{u}) = t$.*

1) There are $\mathbf{a} \in S^t$, $rank\,(\mathbf{a}) = t$, and $\mathbf{B} \in R^{t\times n}$, $freerank\,(\mathbf{B}) = t$, such that $\mathbf{u} = \mathbf{aB}$.

2) There are $\mathbf{a}' \in S^t$, $freerank\,(\mathbf{a}') = t$, and $\mathbf{B}' \in R^{t\times n}$, $rank\,(\mathbf{B}') = t$, such that $\mathbf{u} = \mathbf{a}'\mathbf{B}'$.

1.5 Skew polynomials

In [58], Ore introduced the notion of skew polynomials. He then gave a relation between skew polynomials and linearized polynomials in [57]. In [24], Gabidulin used linearized polynomials to give the encoding and decoding schemes of Gabidulin codes. In this section, we show that some properties of linearized polynomials over finite fields [57] can be generalized to finite principal ideal rings.

1.5.1 Definitions and properties

In the following, we give the definition of skew polynomials over S with automorphism σ without derivation.

Definition 1.41 *The* ***skew polynomial ring*** *over S with automorphism σ, denoted by $S[X,\sigma]$, is the ring of all polynomials in $S[X]$ under the usual addition of polynomials, and the multiplication is defined by the basic rule $Xa = \sigma(a)X$, for all $a \in S$, and extended to all elements of $S[X]$ by associativity and distributivity.*

Let $f = f_0 + f_1X + \cdots + f_nX^n \in S[X,\sigma]$ with $f_n \neq 0$, then n is called the **degree** of f, X^n the **leading monomial** of f, f_n the **leading coefficient** of f, f_nX^n the **leading term** of f, denoted $\deg(f)$, $lm\,(f)$, $lc\,(f)$ and $lt\,(f)$ respectively. If $f = 0$, then we put

$\deg(0) := -\infty$, $lm(0) := 0$, $lc(0) := 0$ and $lt(0) := 0$. The skew polynomial f is called **monic** if $lc(f) = 1$. We denote by $S[X,\sigma]_{<k}$ the set of all skew polynomials of degree less than k. As in the case of classical polynomials, we have the following:

Proposition 1.42 *[11] For all f and g in $S[X,\sigma]$, we have $\deg(f+g) \leq \max\{\deg(f), \deg(g)\}$ and $\deg(fg) \leq \deg(f) + \deg(g)$. Furthermore, if the leading coefficients of g is a unit, then $\deg(fg) = \deg(f) + \deg(g)$ and there exist unique polynomials q, q', r and r' in $S[X,\sigma]$ such that $f = qg + r$ (**right division**) and $f = gq' + r'$ (**left division**) with $\deg(r) < \deg(g)$ and $\deg(r') < \deg(g)$.*

McDonald gave the relation between skew polynomials and linear endomorphisms over finite fields in [49, Corollary II.16]. By [17, Chapter III, Proposition 1.2.], this result can be extended as follows.

Proposition 1.43 *The map:*

$$S[X,\sigma] \longrightarrow Hom_R(S,S)$$

given by

$$\sum_{0\leq 1\leq n} a_i X^i \longmapsto \sum_{0\leq 1\leq n} a_i \sigma^i$$

is a homomorphism of R-algebras. It induces an isomorphism of R-algebras:

$$S[X,\sigma]/(X^m - 1) \cong Hom_R(S,S).$$

Note that if $R = \mathbb{F}_q$, then $S = \mathbb{F}_{q^m}$ and $\sigma(x) = x^q$, for all $x \in \mathbb{F}_{q^m}$. Thus, we now prove that some results in [57] can be extended to finite principal ideal rings.

Notation 1.44 *Let $f = f_0 + f_1 X + \cdots + f_n X^n \in S[X,\sigma]$, $b \in S$ and $\mathbf{b} = (b_1, \ldots, b_n) \in S^n$.*

1. *The element $f_0 b + f_1 \sigma(b) + \cdots + f_n \sigma^n(b)$ will be denoted by $f(b)$.*
2. *The **kernel** of f is $\ker f := \{x \in S : f(x) = 0\}$.*
3. *The vector $(f(b_1), \ldots, f(b_n))$ will be denoted by $f(\mathbf{b})$.*

As $S = S_{(1)} \times \cdots \times S_{(\rho)}$ and $\mathfrak{M}_{(i)} = \mathfrak{m}_{(i)} S_{(i)}$, we have the following Lemma.

Lemma 1.45 *Let $y \in S$. If $\{y\}$ is linearly independent over R, then y is a unit.*

Proof. Suppose that $\{y\}$ is linearly independent over R and y is not a unit. Set $y = \left(y_{(i)}\right)_{1\leq i\leq \rho}$ where $y_{(i)} \in S_{(i)}$. Since y is not a unit, then there is $i_0 \in \{1, \ldots, \rho\}$ such that $y_{(i_0)}$ is not a unit. Consequently, $y_{(i_0)} \in \mathfrak{M}_{(i_0)} = \mathfrak{m}_{(i_0)} S_{(i_0)}$ and there is $0 \neq b_{(i_0)} \in \mathfrak{m}_{(i_0)}^{\nu_{(i_0)}-1}$ such that $b_{(i_0)} y_{(i_0)} = 0$. Set $b = \left(\beta_{(i)}\right)_{1\leq i\leq \rho}$ where $\beta_{(i_0)} = b_{(i_0)}$ and $\beta_{(i)} = 0$ if $i \neq i_0$. Then $by = 0$, which is impossible because $\{y\}$ is linearly independent over R. ■

Analogous to [57], we have the following two propositions.

Proposition 1.46 *Let $\{u_j\}_{1\leq j\leq r}$ be a subset of S, which is linearly independent over R . Then, there is a monic skew polynomial $f \in S[X,\sigma]$ of degree r such that $\ker f = \langle\{u_j\}_{1\leq j\leq r}\rangle$, where $\langle\{u_j\}_{1\leq j\leq r}\rangle$ denotes the R-submodule of S generated by $\{u_j\}_{1\leq j\leq r}$.*

Proof. We prove by induction on $k \in \{1,\ldots,r\}$. Set $f_1 = X - \sigma(u_1)\, u_1^{-1}$. Let $x \in S$, then $x \in \ker f_1$ iff $f_1(x) = 0$ iff $\sigma(x) - \sigma(u_1)\, u_1^{-1} x = 0$ iff $\sigma\left(u_1^{-1} x\right) = u_1^{-1} x$ iff $u_1^{-1} x \in R$ iff $x \in \langle\{u_1\}\rangle$. Thus $\ker f_1 = \langle\{u_1\}\rangle$. Let $k \in \{1,\ldots,r-1\}$. Assume that there is a monic polynomial $f_k \in S[X,\sigma]$ of degree k such that $\ker f_k = \langle\{u_j\}_{1\leq j\leq k}\rangle$. We claim that $f_k(u_{k+1})$ is a unit. Indeed, let $a \in R$ such that $a f_k(u_{k+1}) = 0$ then $a u_{k+1} \in \ker f_k = \langle\{u_i\}_{1\leq j\leq k}\rangle$, consequently, $a = 0$ because $\{u_j\}_{1\leq j\leq k+1}$ is R-linear independent. Thus by lemma 1.45, $f_k(u_{k+1})$ is a unit. Set $f_{k+1} = \left(X - \sigma\left(f_k(u_{k+1})\right) f_k(u_{k+1})^{-1}\right) \times f_k$, then $\deg(f_{k+1}) = k+1$ and $\{u_j\}_{1\leq j\leq k+1} \subset \ker f_{k+1}$. Let $x \in \ker f_{k+1}$, then $f_{k+1}(x) = 0$, i.e. $\sigma(f_k(x)) - \sigma\left(f_k(u_{k+1})\right) f_k(u_{k+1})^{-1} f_k(x) = 0$, i.e. $\sigma\left(f_k(u_{k+1})^{-1} f_k(x)\right) = f_k(u_{k+1})^{-1} f_k(x)$, i.e. $f_k(u_{k+1})^{-1} f_k(x) \in R$, i.e. there is $\lambda \in R$ such that $f_k(u_{k+1})^{-1} f_k(x) = \lambda$, i.e. $x - \lambda u_{k+1} \in \ker f_k$, i.e. $x \in \langle\{u_j\}_{1\leq j\leq k+1}\rangle$. Hence, $\ker f_{k+1} = \langle\{u_j\}_{1\leq j\leq k+1}\rangle$. ■

Proposition 1.47 *Let $\{u_j\}_{1\leq j\leq r}$ be a subset of S. Then, the matrix $\left(\sigma^i(u_j)\right)_{0\leq i\leq r-1,\ 1\leq j\leq r}$ is invertible if and only if $\{u_j\}_{1\leq j\leq r}$ is linearly independent over R.*

Proof. Assume that $\{u_j\}_{1\leq j\leq r}$ is linearly independent over R. Let $i \in \{1,\ldots,r\}$. By Proposition 1.46, there is a monic skew polynomial $T_i \in S[X,\sigma]$ of degree $r-1$ such that $\ker T_i = \langle\{u_j\}_{1\leq j\leq r, j\neq i}\rangle$. Using the same arguments as in the proof of Proposition 1.46, we can show that $T_i(u_i)$ is a unit. Set $T_i(u_i)^{-1} T_i(X) = \sum_{0\leq j\leq r-1} v_{i,j} X^j$, where $v_{i,j} \in S$, then the matrix $(v_{i,j})_{1\leq i\leq r,\ 0\leq j\leq r-1}$ is the inverse of the matrix $\left(\sigma^i(u_j)\right)_{0\leq i\leq r-1,\ 1\leq j\leq r}$.

Conversely, assume that $\left(\sigma^i(u_j)\right)_{0\leq i\leq r-1,\ 1\leq j\leq r}$ is invertible. Let $\lambda_1,\ldots,\lambda_r$ be the elements of R such that $\lambda_1 u_1 + \cdots + \lambda_r u_r = 0$. Then, we have $\lambda_1 \sigma^i(u_1) + \cdots + \lambda_r \sigma^i(u_r) = 0$, for $i = 0,\ldots r-1$. Consequently, $\lambda_1 = \cdots = \lambda_r = 0$. ■

Corollary 1.48 *Let $\{u_j\}_{1\leq j\leq r}$ be a subset of S, which is linearly independent over R and let $V \in S[X,\sigma]$ be a monic skew polynomial of degree r such that $\ker V = \langle\{u_j\}_{1\leq j\leq r}\rangle$. Let $P \in S[X,\sigma]$. Then, $P(u_j) = 0$, for $j = 1,\ldots,r$, if and only if there is $Q \in S[X,\sigma]$ such that $P = QV$.*

Proof. Let Q be the quotient and W be the remainder of the right Euclidean division of P by V in $S[X,\sigma]$. Then, $P(u_j) = 0$, for $j = 1,\ldots,r$, if and only if $W(u_j) = 0$, for $j = 1,\ldots,r$, if and only if $W = 0$, because $deg\,(W) < r$ and the matrix $\left(\sigma^i(u_j)\right)_{0\leq i\leq r-1,\ 1\leq j\leq r}$ is invertible. ■

A direct consequence of Proposition 1.46 and Proposition 1.40 is the following:

Proposition 1.49 *Let $\mathbf{w} = (w_i)_{1\leq i\leq n} \in S^n$, $rank\,(\mathbf{w}) = r$. Then, there is a monic skew polynomial $P \in S[X,\sigma]$ of degree r such that $P(\mathbf{w}) = \mathbf{0}$.*

As in the case of finite fields [57], the following proposition gives the link between the degree of a skew polynomial and the rank of its kernel.

Proposition 1.50 *Let $P = a_0 + a_1X + \cdots + a_\eta X^\eta \in S[X,\sigma]$ such that a_{i_0} is a unit for some $i_0 \in \{0,\ldots,\eta\}$. Then, $rank\,(\ker P) \leq \deg(P)$.*

Proof. Suppose that $\deg(P) < rank\,(\ker P)$. Set $r = rank\,(\ker P)$, then by Proposition 1.20 there is a free basis $\{b_i\}_{1\leq i\leq m}$ of S and the scalars $\lambda_1,\ldots,\lambda_r$ of R such that $\{\lambda_i b_i\}_{1\leq i\leq r}$ generates $\ker P$, with $\lambda_1|\lambda_2|\ldots|\lambda_r$. We then have $\lambda_r P(b_i) = 0$, for $i = 1,\ldots,r$. Hence, by Corollary 1.48, $\lambda_r P = 0$. This is clearly impossible because $\lambda_r \neq 0$ and a_{i_0} is a unit. Thus, $rank\,(\ker P) \leq \deg(P)$. ■

Remark 1.51 *In Proposition 1.50, if all coefficients of P are non-units, then we can have $\deg(P) < rank\,(\ker P)$. Indeed, let $R = \mathbb{Z}_4$, $S = R[z]/(z^2+z+1)$ and $a = z + (z^2+z+1)$. Then, S is a Galois extension of R where the Galois group is generated by a power map $\sigma : a \mapsto a^2$. Set $P = 2X - 2 \in S[X,\sigma]$. Then, $\ker P$ is generated by 1 and $2a$. Thus, all coefficients of P are non-units and $\deg(P) < rank\,(\ker P)$.*

Proposition 1.47 and Proposition 1.50 are some of the main results that allow to extend the properties of Gabudulin codes to finite principal ideal rings. Note that if one of the automorphisms $\sigma_{(i)}$ is not a generator of the respective Galois group, then the ring S is not a Galois extension of R with Galois group $G_R(S)$ and therefore, as in [3], Proposition 1.47 and Proposition 1.50 will not be true in general. Indeed, consider the following:

Example 1.52 *Consider the finite field $\mathbb{F}_2$ and the Galois extention $\mathbb{F}_{2^4} = \mathbb{F}_2[z]/(z^4+z^3+1)$, set $a = z + (z^4+z^3+1)$ and let $\theta = \left(\theta_{(1)},\theta_{(2)}\right)$ be the map from $\mathbb{F}_{2^4}\times\mathbb{F}_{2^4}$ to $\mathbb{F}_{2^4}\times\mathbb{F}_{2^4}$, where $\theta_{(1)}(x) = x^2$ and $\theta_{(2)}(x) = x^4$ for all x in $\mathbb{F}_{2^4}$. The map θ is an $\mathbb{F}_2\times\mathbb{F}_2$-automorphism of $\mathbb{F}_{2^4}\times\mathbb{F}_{2^4}$ and we have $\theta^2 = \left(\theta_{(1)}^2, id\right)$.*

1) Let G be the group generated by θ. The set $\mathbb{F}_{2^4}\times\{0\}$ is a maximal ideal of $\mathbb{F}_{2^4}\times\mathbb{F}_{2^4}$ and for all $x \in \mathbb{F}_{2^4}\times\mathbb{F}_{2^4}$ we have $x - \theta^2(x) \in \mathbb{F}_{2^4}\times\{0\}$. Thus, by Definition 1.32, $\mathbb{F}_{2^4}\times\mathbb{F}_{2^4}$ is not a Galois extension of $\mathbb{F}_2\times\mathbb{F}_2$ with the group G.

2) Set $\mathbf{a} = (a,a)$ and $\mathbf{1} = (1,1)$. Then $\left\{\mathbf{1},\mathbf{a},\mathbf{a}^2\right\}$ is linearly independent over $\mathbb{F}_2\times\mathbb{F}_2$. By [20, Corollary 2.8], the matrix

$$
\begin{aligned}
\mathbf{M} &= \begin{pmatrix} \mathbf{1} & \mathbf{a} & \mathbf{a}^2 \\ \theta(\mathbf{1}) & \theta(\mathbf{a}) & \theta(\mathbf{a}^2) \\ \theta^2(\mathbf{1}) & \theta^2(\mathbf{a}) & \theta^2(\mathbf{a}^2) \end{pmatrix} \\
&= \begin{pmatrix} (1,1) & (a,a) & (a^2,a^2) \\ (1,1) & (a^2,a^4) & (a^4,a^8) \\ (1,1) & (a^4,a) & (a^8,a^2) \end{pmatrix}
\end{aligned}
$$

is not invertible because the rows of the matrix

$$
\begin{pmatrix} 1 & a & a^2 \\ 1 & a^4 & a^8 \\ 1 & a & a^2 \end{pmatrix}
$$

are not linearly independent.

3) Let $P = X - (1,1)$ in $(\mathbb{F}_{2^4}\times\mathbb{F}_{2^4})[X,\theta]$. $\ker P$ is generated by $(1,1)$ and $(0,a+a^4)$. Thus, $rank\,(\ker P) > \deg(P)$.

1.5.2 Gröbner bases of modules over skew polynomials

Gröbner bases are a mathematical tool that allows to solve several problems in the set of polynomials. It was introduced by Buchberger in his Ph.D thesis [10]. Nowadays, Gröbnes bases have many applications, especially in the coding theory. Indeed, in [23], Fitzpatrick used this theory to give an iterative method for decoding alternate codes. In [41], Kuijper and Trautmann adopted this iterative method to give a parametrization approach to the list decoding algorithm of Gabidulin codes. The theory of Gröbnes bases has been generalized over rings. Thus, in [33], Jiménez and Lezama studied the theory of Gröbner bases of modules over skew Poincaré–Birkhoff–Witt extension. In this subsection, we recall some results in this theory that we will use to solve the key equation.

Given a positive integer ℓ, we denote by $S[X,\sigma]^{\ell+1}$ the $\ell+1$-fold direct product of $S[X,\sigma]$. For all $\mathbf{u} \in S[X,\sigma]^{\ell+1}$, the l-th component of $\mathbf{u}$ is denoted by $u^{(l)}$, for $l \in \{0,\dots,\ell\}$, i.e. $\mathbf{u} = \left(u^{(0)}, u^{(1)}, \dots, u^{(\ell)}\right)$. We consider $S[X,\sigma]^{\ell+1}$ as a left $S[X,\sigma]$-module where addition is defined componentwise and for $a \in S[X,\sigma]$ and $\mathbf{u} \in S[X,\sigma]^{\ell+1}$, $a\mathbf{u} = \left(au^{(0)}, au^{(1)}, \dots, au^{(\ell)}\right)$. We denote by $\mathbf{e}^{(0)} = (1,0,\dots,0)$, $\mathbf{e}^{(1)} = (0,1,0,\dots,0)$, $\dots$, $\mathbf{e}^{(\ell)} = (0,\dots,0,1)$ the canonical basis of $S[X,\sigma]^{\ell+1}$. A **monomial** in $S[X,\sigma]^{\ell+1}$ is an element of the form $X^\alpha \mathbf{e}^{(l)}$ where $\alpha \in \mathbb{N}$ and $l \in \{0,\dots,\ell\}$. The set of monomials of $S[X,\sigma]^{\ell+1}$ will be denoted by $Mon\left(S[X,\sigma]^{\ell+1}\right)$. If $X^\alpha \mathbf{e}^{(l)} \in Mon\left(S[X,\sigma]^{\ell+1}\right)$, then l is called the **index** of $X^\alpha \mathbf{e}^{(l)}$ and denoted by $ind\left(X^\alpha \mathbf{e}^{(l)}\right)$. Let $X^{\alpha_1}\mathbf{e}^{(l_1)}, X^{\alpha_2}\mathbf{e}^{(l_2)} \in Mon\left(S[X,\sigma]^{\ell+1}\right)$, we say that $X^{\alpha_1}\mathbf{e}^{(l_1)}$ **divides** $X^{\alpha_2}\mathbf{e}^{(l_2)}$, denoted $X^{\alpha_1}\mathbf{e}^{(l_1)} | X^{\alpha_2}\mathbf{e}^{(l_2)}$, if $l_1 = l_2$ and there is $\beta \in \mathbb{N}$ such that $\alpha_2 = \alpha_1 + \beta$. We will say that any monomial $X^\alpha \mathbf{e}^{(l)} \in Mon\left(S[X,\sigma]^{\ell+1}\right)$ divides the null vector $\mathbf{0}$.

Definition 1.53 *A **monomial order** on $Mon\left(S[X,\sigma]^{\ell+1}\right)$ is a total order $\succeq$ satisfying the following two conditions:*

(i) $X^\beta\left(X^\alpha \mathbf{e}^{(l)}\right) \succeq X^\alpha \mathbf{e}^{(l)}$, for all $X^\alpha \mathbf{e}^{(l)} \in Mon\left(S[X,\sigma]^{\ell+1}\right)$ and every $\beta \in \mathbb{N}$;

(ii) if $X^{\alpha_2}\mathbf{e}^{(l_2)} \succeq X^{\alpha_1}\mathbf{e}^{(l_1)}$, then $X^\beta\left(X^{\alpha_2}\mathbf{e}^{(l_2)}\right) \succeq X^\beta\left(X^{\alpha_1}\mathbf{e}^{(l_1)}\right)$ for all $X^{\alpha_1}\mathbf{e}^{(l_1)}, X^{\alpha_2}\mathbf{e}^{(l_2)} \in Mon\left(S[X,\sigma]^{\ell+1}\right)$ and every $\beta \in \mathbb{N}$.

If $X^{\alpha_2}\mathbf{e}^{(l_2)} \succeq X^{\alpha_1}\mathbf{e}^{(l_1)}$ and $X^{\alpha_2}\mathbf{e}^{(l_2)} \neq X^{\alpha_1}\mathbf{e}^{(l_1)}$ we will write $X^{\alpha_2}\mathbf{e}^{(l_2)} \succ X^{\alpha_1}\mathbf{e}^{(l_1)}$. $X^{\alpha_1}\mathbf{e}^{(l_1)} \preceq X^{\alpha_2}\mathbf{e}^{(l_2)}$ means that $X^{\alpha_2}\mathbf{e}^{(l_2)} \succeq X^{\alpha_1}\mathbf{e}^{(l_1)}$.

Remark 1.54 *By [39, Chapter 0, Section 17, Lemma 15] a monomial order on $Mon\left(S[X,\sigma]^{\ell+1}\right)$ is a well order. Note that the condition (iii) of [33, Definition 15.] is given so that a monomial order is a well order. So, in this specific case we do not need this condition.*

We fix a monomial order $\succeq$ on the monomials of $S[X,\sigma]^{\ell+1}$. Let $\mathbf{f} \in S[X,\sigma]^{\ell+1} \setminus \{\mathbf{0}\}$, then $\mathbf{f}$ can be written uniquely as $\mathbf{f} = \sum_{i=1}^n c_i X^{\alpha_i}\mathbf{e}^{(l_i)}$ where $n \in \mathbb{N}$, $c_i \in S$, for $i = 1,\dots,n$, $c_1 \neq 0$ and $X^{\alpha_1}\mathbf{e}^{(l_1)} \succ \cdots \succ X^{\alpha_n}\mathbf{e}^{(l_n)}$. We define:

- $lm(\mathbf{f}) := X^{\alpha_1}\mathbf{e}^{(l_1)}$ as the **leading monomial** of $\mathbf{f}$;
- $lc(\mathbf{f}) := c_1$ as the **leading coefficient** of $\mathbf{f}$;
- $lt(\mathbf{f}) := c_1 X^{\alpha_1}\mathbf{e}^{(l_1)}$as the **leading term** of $\mathbf{f}$;

- $\deg(\mathbf{f}) := \alpha_1$ as the **degree** of $\mathbf{f}$.

For $\mathbf{f} = \mathbf{0}$ we define $lt(\mathbf{0}) := \mathbf{0}$, $lm(\mathbf{0}) := \mathbf{0}$, $lc(\mathbf{0}) := 0$ and extend $\succeq$ to $Mon\left(S[X,\sigma]^{\ell+1}\right) \cup \{\mathbf{0}\}$ by $X^{\alpha}\mathbf{e}^{(l)} \succ \mathbf{0}$ for all $X^{\alpha}\mathbf{e}^{(l)} \in Mon\left(S[X,\sigma]^{\ell+1}\right)$. Let $W \subset S[X,\sigma]^{\ell+1}$, we write $lt(W)$ for $\{lt(\mathbf{w}) \ : \mathbf{w} \in W\}$ and the submodule of $S[X,\sigma]^{\ell+1}$ generated by W is denoted by $\langle W \rangle$.

As in [33], we give the definition of the reduction process in $S[X,\sigma]^{\ell+1}$.

Definition 1.55 *Let F be a finite set of nonzero vectors of $S[X,\sigma]^{\ell+1}$ and let $\mathbf{f},\mathbf{h} \in S[X,\sigma]^{\ell+1}$, we say that $\mathbf{f}$ **reduces** to $\mathbf{h}$ by F in **one step**, denoted $\mathbf{f} \xrightarrow{F} \mathbf{h}$, if there exist elements $\mathbf{f}_1,\ldots,\mathbf{f}_t \in F$ and $r_1,\ldots,r_t \in S$ such that:*

(i) $lm(\mathbf{f}_i)\,|lm(\mathbf{f})$, for $i = 1,\ldots,t$, i.e., there exist $\alpha_i \in \mathbb{N}$ such that $lm(\mathbf{f}) = X^{\alpha_i} lm(\mathbf{f}_i)$;

(ii) $lc(\mathbf{f}) = r_1\sigma^{\alpha_1}(lc(\mathbf{f}_1)) + \cdots + r_t\sigma^{\alpha_t}(lc(\mathbf{f}_t))$;

(iii) $\mathbf{h} = \mathbf{f} - \sum_{i=1}^{t} r_i X^{\alpha_i}\mathbf{f}_i$.

*We say that $\mathbf{f}$ **reduces** to $\mathbf{h}$ by F, denoted $\mathbf{f} \xrightarrow{F}_{+} \mathbf{h}$, if and only if there exist vectors $\mathbf{h}_1,\ldots,\mathbf{h}_{t-1} \in S[X,\sigma]^{\ell+1}$ such that*

$$\mathbf{f} \xrightarrow{F} \mathbf{h}_1 \xrightarrow{F} \mathbf{h}_2 \xrightarrow{F} \cdots \xrightarrow{F} \mathbf{h}_{t-1} \xrightarrow{F} \mathbf{h}.$$

*$\mathbf{f}$ is **reduced** also called minimal w.r.t. F if $\mathbf{f} = \mathbf{0}$ or there is no one step reduction of $\mathbf{f}$ by F , i.e., one of the first two conditions of Definition 1.55 fails. Otherwise, we will say that $\mathbf{f}$ is **reducible** w.r.t. F. If $\mathbf{f} \xrightarrow{F}_{+} \mathbf{h}$ and $\mathbf{h}$ is reduced w.r.t. F, then we say that $\mathbf{h}$ is a **remainder** for $\mathbf{f}$ w.r.t. F.*

Remark 1.56 *With the notations of the Definition 1.55, we have the following remarks:*

(a) if $\mathbf{f} \xrightarrow{F} \mathbf{h}$, then $lm(\mathbf{f}) \succ lm(\mathbf{h})$ and $\mathbf{f} - \mathbf{h} \in \langle F \rangle$;

(b) by definition we will assume that $\mathbf{0} \xrightarrow{F} \mathbf{0}$.

By [33, Theorem 23.], we have the following proposition.

Proposition 1.57 *Let $F = \{\mathbf{f}_1,\ldots,\mathbf{f}_t\}$ be a set of nonzero vectors of $S[X,\sigma]^{\ell+1}$ and let $\mathbf{f} \in S[X,\sigma]^{\ell+1}$, then there exist $q_1,\ldots,q_t \in S[X,\sigma]$ and the reduced vector $\mathbf{h} \in S[X,\sigma]^{\ell+1}$ w.r.t. F such that $\mathbf{f} \xrightarrow{F}_{+} \mathbf{h}$ and*

$$\mathbf{f} = q_1\mathbf{f}_1 + \cdots + q_t\mathbf{f}_t + \mathbf{h}$$

with

$$lm(\mathbf{f}) = \max\{lm(q_1)\,lm(\mathbf{f}_1), \ldots, lm(q_t)\,lm(\mathbf{f}_t), lm(\mathbf{h})\}.$$

Definition 1.58 *[33] (a) Let M be a nonzero submodule of $S[X,\sigma]^{\ell+1}$ and let G be a non empty finite subset of nonzero vectors of M, we say that G is a **Gröbner basis** for M if each element $\mathbf{0} \neq \mathbf{f} \in M$ is reducible w.r.t. G. We will say that $\{\mathbf{0}\}$ is a Gröbner basis for $M = 0$.*

(b) A set $G \subset S[X,\sigma]^{\ell+1}$ is called a Gröbner basis provided that G is a Gröbner basis for $\langle G \rangle$.

By [33, Theorem 26.], we have the following:

Proposition 1.59 *Let M be a nonzero submodule of $S[X,\sigma]^{\ell+1}$ and let G be a non empty finite subset of nonzero vectors of M. Then, the following conditions are equivalent.*

(i) G is a Gröbner basis for M.

(ii) For any vector $\mathbf{f} \in S[X,\sigma]^{\ell+1}$, $\mathbf{f} \in M$ if and only if $\mathbf{f} \xrightarrow{G}_{+} \mathbf{0}$.

(iii) For any $\mathbf{f} \in M$ there exist $\mathbf{g}_1, \ldots, \mathbf{g}_t \in G$ such that $lm(\mathbf{g}_j) \,|\, lm(\mathbf{f})$, for $j = 1, \ldots, t$, i.e., there exist $\alpha_j \in \mathbb{N}$ such that $lm(\mathbf{g}_j) = X^{\alpha_j} lm(\mathbf{f})$, and $lc(\mathbf{f}) \in \langle \sigma^{\alpha_1}(lc(\mathbf{g}_1)), \ldots, \sigma^{\alpha_t}(lc(\mathbf{g}_t)) \rangle$.

By Proposition 1.55 and Proposition 1.59, we have the following:

Proposition 1.60 *Let M be a submodule of $S[X,\sigma]^{\ell+1}$ and let $G = \{\mathbf{g}_1, \ldots, \mathbf{g}_t\} \subset M$. If G is a Gröbner basis for M then for all $\mathbf{f} \in M$ there exist $q_1, \ldots, q_t \in S[X,\sigma]$ such that*

$$\mathbf{f} = q_1 \mathbf{g}_1 + \cdots + q_t \mathbf{g}_t$$

with

$$lm(\mathbf{f}) = \max\{lm(q_1)\, lm(\mathbf{g}_1), \ldots, lm(q_t)\, lm(\mathbf{g}_t)\}.$$

According to [33, Corollary 31.], we have the following:

Proposition 1.61 *Let M be a nonzero submodule of $S[X,\sigma]^{\ell+1}$. Then, M has a Gröbner basis.*

Chapter II

Rank-metric codes over finite principal ideal rings

Recall that rank-metric codes are codes for which each codeword is a matrix and the distance between two codewords is the rank of their difference. In this chapter, we show that some results in rank-metric codes can be extended to finite principal ideal rings. These results are given as follows.

In Section 2.1, we give the two representations of rank-metric codes and we prove that the rank-metric Singleton bound can be extended to finite principal ideal rings.

In Section 2.2, we extend the definition of Gabidulin codes and prove that their properties are preserved.

In Section 2.3, we give some properties of interleaved Gabidulin codes. We show that collaborative decoding of interleaved Gabidulin codes can be translated to the problem of reconstruction of skew polynomials. We use the theory of Gröbner bases to give an iterative algorithm to solve this reconstruction problem.

In Section 2.4, we give the unique decoding, minimal list decoding, and error-erasure decoding algorithms of interleaved Gabidulin codes.

2.1 Matrix and vector representations of rank-metric codes

Analogous to the case of finite fields [16], [24], [63], we give the following definitions.

In matrix representation, **rank codes** are defined as subsets of a normed space $(R^{m\times n},\ rank)$, where the norm of a matrix $\mathbf{A}\in R^{m\times n}$ is the rank of $\mathbf{A}$ over R. The **rank distance** between two matrices $\mathbf{A}$ and $\mathbf{B}$ is the rank of their difference, i.e $rank\,(\mathbf{A}-\mathbf{B})$. The **rank distance of a matrix rank code** $\mathcal{M}\subset R^{m\times n}$ is defined as the minimal pairwise distance:

$$d\,(\mathcal{M})=\min\left\{rank\,(\mathbf{A}-\mathbf{B}):\ \mathbf{A},\ \mathbf{B}\in\mathcal{M},\ \mathbf{A}\neq\mathbf{B}\right\}.$$

A matrix rank code $\mathcal{M}\subset R^{m\times n}$ is said R-linear if it is a submodule of $R^{m\times n}$.

In vector representation, rank codes are defined as subsets of a normed S-module space $(S^{n},\ rank)$, where the norm of a vector $\mathbf{u}\in S^{n}$ is the rank of $\mathbf{u}$. The **rank distance** between two vectors $\mathbf{u}$ and $\mathbf{v}$ is the rank of their difference, i.e $rank\,(\mathbf{u}-\mathbf{v})$. The **rank distance of a vector rank code** $\mathcal{C} \subset S^n$ is defined as the minimal pairwise distance:

$$d\,(\mathcal{C}) = \min\left\{rank\,(\mathbf{u}-\mathbf{v}):\ \mathbf{u},\ \mathbf{v} \in \mathcal{C},\ \mathbf{u} \neq \mathbf{v}\right\}.$$

A vector rank code $\mathcal{C} \subset S^n$ is called **linear** if it is a submodule of S-module S^n, furthermore if $\mathcal{C}$ is a free submodule of S^n then $\mathcal{C}$ is called a **free rank code**.

Let $\mathcal{C} \subset S^n$ be a linear rank code. The number $\mu_S\,(\mathcal{C})$, denoted by $rank_S\,(\mathcal{C})$ or simply by $rank\,(\mathcal{C})$, is called the **rank** of $\mathcal{C}$. A **generator matrix** of $\mathcal{C}$ is a $rank(\mathcal{C}) \times n$ matrix over S whose rows generate $\mathcal{C}$. The **inner product** of two vectors $\mathbf{u} = (u_1, \ldots, u_n) \in S^n$ and $\mathbf{v} = (v_1, \ldots, v_n) \in S^n$ is defined by

$$\mathbf{u} \cdot \mathbf{v} = u_1 v_1 + \cdots + u_n v_n.$$

The **dual** of $\mathcal{C}$ is the submodule of S^n defined by

$$\mathcal{C}^{\perp} = \left\{\mathbf{u} \in S^n:\ \mathbf{u} \cdot \mathbf{v} = 0, \quad \text{for every } \mathbf{v} \in \mathcal{C}\right\}.$$

A **parity-check matrix** of $\mathcal{C}$ is a generator matrix of $\mathcal{C}^{\perp}$.

Note that by Proposition 1.39, there exists a relation between the matrix representation and the vector representation. As in the case of finite fields [16], [24], [63], the following proposition establishes the rank-metric Singleton bound.

Proposition 2.1 *(Singleton bound)*
Let $\mathcal{M} \subset R^{m\times n}$ be a rank code of rank distance d, then

$$|\mathcal{M}| \leq |R|^{\min\{m(n-d+1),\ n(m-d+1)\}}$$

where $|\mathcal{M}|$ and $|R|$ denote the cardinality of $\mathcal{M}$ and R respectively.

Proof. Since the minimal distance of $\mathcal{M}$ is d, no two distinct code matrices $\mathbf{A}_1,\ \mathbf{A}_2 \in \mathcal{M}$ have the same first $n-(d-1)$ columns. For, otherwise, we have $rank\,(\mathbf{A}_1 - \mathbf{A}_2) \leq d-1$, which contradicts the minimality of d. So, $|\mathcal{M}| \leq |R|^{m(n-(d-1))}$. Using the same argument for the rows of two distinct code matrices of $\mathcal{M}$, we also have $|\mathcal{M}| \leq |R|^{n(m-(d-1))}$. Consequently, $|\mathcal{M}| \leq |R|^{\min\{m(n-(d-1)),\ n(m-(d-1))\}}$. ■

Definition 2.2 *If $\mathcal{M} \subset R^{m\times n}$ and $\mathcal{C} \subset S^n$ be the rank codes of rank distance d such that $|\mathcal{M}| = |\mathcal{C}| = |R|^{\min\{m(n-d+1),\ n(m-d+1)\}}$, we say that $\mathcal{M}$ and $\mathcal{C}$ are **Maximum Rank Distance codes**, or, **MRD codes**.*

In finite fields, Gabidulin codes are MRD codes [16], [24], [63]. We will prove that this property extends to finite principal ideal rings.

2.2 Gabidulin codes

Let $\mathbf{g} = (g_1, \ldots, g_n) \in S^n$, such that $\{g_1, \ldots, g_n\}$ is linearly independent over R. Let k be an integer such that $0 < k \leq n$.

Definition 2.3 *(Gabidulin Codes)*
A ***Gabidulin code*** $Gab_k(\mathbf{g})$ *of length* n*, dimension* k *and support* $\mathbf{g}$ *is the* S*-module given by:*

$$Gab_k(\mathbf{g}) = \{f(\mathbf{g}) : f \in S[X, \sigma]_{<k}\}.$$

Proposition 2.4 *The Gabidulin code* $Gab_k(\mathbf{g})$ *is a free rank code of rank* k *with a generator matrix*

$$\mathbf{G} = \begin{pmatrix} \sigma^0(g_1) & \cdots & \sigma^0(g_n) \\ \vdots & \ddots & \vdots \\ \sigma^{k-1}(g_1) & \cdots & \sigma^{k-1}(g_n) \end{pmatrix}.$$

Proof. Let $\mathbf{c} = (c_1, \ldots, c_n) \in Gab_k(\mathbf{g})$. Then, there is $f = f_0 + f_1 X + \cdots f_{k-1} X^{k-1}$ in $S[X, \sigma]$ such that $\mathbf{c} = f(\mathbf{g})$, i.e.

$$\begin{cases} c_1 = f_0 \sigma^0(g_1) + f_1 \sigma(g_1) + \cdots f_{k-1} \sigma^{k-1}(g_1) \\ \vdots \\ c_n = f_0 \sigma^0(g_n) + f_1 \sigma(g_n) + \cdots f_{k-1} \sigma^{k-1}(g_n) \end{cases},$$

i.e.

$$(c_1, \ldots, c_n) = (f_0, \ldots, f_{k-1}) \begin{pmatrix} \sigma^0(g_1) & \cdots & \sigma^0(g_n) \\ \vdots & \ddots & \vdots \\ \sigma^{k-1}(g_1) & \cdots & \sigma^{k-1}(g_n) \end{pmatrix}$$

Thus, the rows of $\mathbf{G}$ generate $Gab_k(\mathbf{g})$. By Proposition 1.47 and [20, Corollary 2.8], the rows of $\mathbf{G}$ are linearly independent over S, hence $Gab_k(\mathbf{g})$ is a free code of rank k. ■

Theorem 2.5 *(a) The rank distance,* d*, of* $Gab_k(\mathbf{g})$ *is given by* $d = n - k + 1$*.*

(b) $Gab_k(\mathbf{g})$ *is an MRD code.*

Proof. (a) Since $n \leq m$ and $Gab_k(\mathbf{g})$ is a free code of rank k, we have $d \leq n - k + 1$, by Proposition 2.1. Let $\mathbf{c} \in Gab_k(\mathbf{g})$ such that $rank(\mathbf{c}) = d$. Then, there is $f \in S[X, \sigma]_{<k}$, such that $\mathbf{c} = f(\mathbf{g})$. By Proposition 1.49, there is a monic skew polynomial $P \in S[X, \sigma]$, $\deg(P) = d$, such that $P(\mathbf{c}) = \mathbf{0}$. Consequently, $(Pf)(\mathbf{g}) = \mathbf{0}$. As $Pf \neq 0$, we have $n \leq \deg(Pf)$, by Corollary 1.48. But $\deg(Pf) = \deg(P) + \deg(f) \leq d + k - 1$.

(b) As $n \leq m$, $d = n - k + 1$ and $Gab_k(\mathbf{g})$ is a free code of rank k, then $Gab_k(\mathbf{g})$ an MRD code. ■

As in the case of finite fields, the next theorem shows that the dual of a Gabidulin code is also a Gabidulin code.

Theorem 2.6 *Let $(\gamma_{i,j})_{1\leq i\leq n,1\leq j\leq n}$ be the inverse of the matrix $(\sigma^i(g_j))_{0\leq i\leq n-1,1\leq j\leq n}$. Set*

$$h_i := \sigma^{-n+k+1}(\gamma_{i,n}), \qquad i = 1, \ldots, n.$$

Then, the family $\{h_1, \ldots, h_n\}$ is linearly independent over R and a parity-check matrix of $Gab_k(\mathbf{g})$ is

$$\mathbf{H} = \begin{pmatrix} \sigma^0(h_1) & \cdots & \sigma^0(h_n) \\ \vdots & \ddots & \vdots \\ \sigma^{n-k-1}(h_1) & \cdots & \sigma^{n-k-1}(h_n) \end{pmatrix}.$$

Proof. The product of the two matrices $(\sigma^i(g_j))_{0\leq i\leq n-1,1\leq j\leq n}$ and $(\sigma^{1-n+j}(\gamma_{i,n}))_{1\leq i\leq n,0\leq j\leq n-1}$ is a lower unitriangular matrix. Thus, the matrix $(\sigma^{1-n+j}(\gamma_{i,n}))_{1\leq i\leq n,0\leq j\leq n-1}$ is invertible. Therefore, by Proposition 1.47, $\{\gamma_{1,n}, \ldots, \gamma_{n,n}\}$ is linearly independent over R. Consequently, $\{h_1, \ldots, h_n\}$ is linearly independent over R. Thus, the rows of the matrix $\mathbf{H}$ are linearly independent over S and $\mathbf{G}\mathbf{H}^T = \mathbf{0}$. Since $Gab_k(\mathbf{g})$ is a free code of length n and rank k, by [20, Proposition 2.9], $Gab_k(\mathbf{g})^\perp$ is a free code of rank $n-k$. Consequently, $\mathbf{H}$ is a parity-check matrix of $Gab_k(\mathbf{g})$. ■

In [45], Loidreau showed that decoding of Gabidulin codes can be translated to the problem of reconstruction of skew polynomials. In the input of decoding algorithm given in [45, page 40], it is assumed that the rank of the error is less than or equal to the error-correcting capability of the code. But in practice, the receiver does not know the rank of the error. In [4], Augot et al. gave a similar algorithm without this condition. We will prove that [4, Algorithm 2] can be extended to finite principal ideal rings.

For the remainder of this section, let $t_0 := \lfloor (n-k)/2 \rfloor$ be the error correction capability of the Gabidulin code $Gab_k(\mathbf{g})$. Similarly to [45, Proposition 1 and Proposition 2], we give the following:

Lemma 2.7 *Let $\mathbf{y} \in S^n$ be a received word of the Gabidulin code $Gab_k(\mathbf{g})$. Assume that there is $f \in S[X,\sigma]_{<k}$ such that $rank(\boldsymbol{y} - f(\mathbf{g})) \leq t_0$. Then, the following linear equation*

$$\begin{pmatrix} \mathbf{A}_1 & \mathbf{A}_2 \end{pmatrix} \begin{pmatrix} \mathbf{u}^T \\ \mathbf{v}^T \end{pmatrix} = \begin{pmatrix} \sigma^{t_0}(y_1) \\ \vdots \\ \sigma^{t_0}(y_n) \end{pmatrix} \tag{2.1}$$

with unknowns $\mathbf{u} = (u_0, \ldots, u_{k+t_0-1})$ and $\mathbf{v} = (v_0, \ldots, v_{t_0-1})$ has a solution, where

$$\mathbf{A}_1 = \begin{pmatrix} \sigma^0(g_1) & \cdots & \sigma^{k+t_0-1}(g_1) \\ \vdots & \ddots & \vdots \\ \sigma^0(g_n) & \cdots & \sigma^{k+t_0-1}(g_n) \end{pmatrix}$$

and

$$\mathbf{A}_2 = \begin{pmatrix} -\sigma^0(y_1) & \cdots & -\sigma^{t_0-1}(y_1) \\ \vdots & \ddots & \vdots \\ -\sigma^0(y_n) & \cdots & -\sigma^{t_0-1}(y_n) \end{pmatrix}.$$

Moreover, if $\mathbf{u} = (u_0, \ldots, u_{k+t_0-1})$ *and* $\mathbf{v} = (v_0, \ldots, v_{t_0-1})$ *are a solution of this equation, then* $U = Vf$ *where* $U = u_0 + u_1X + \cdots + u_{k+t_0-1}X^{k+t_0-1}$ *and* $V = v_0 + v_1X + \cdots + v_{t_0-1}X^{t_0-1} + X^{t_0}$.

Proof. Set $t = rank\,(\boldsymbol{y} - f\,(\mathbf{g}))$. By Proposition 1.49, there is a monic skew polynomials $W \in S[X,\sigma]$ of degree t such that $W\,(\boldsymbol{y} - f\,(\mathbf{g})) = \mathbf{0}$. Therefore, $X^{t_0-t}W\,(\boldsymbol{y}) = X^{t_0-t}W\,(f\,(\mathbf{g}))$. Set $X^{t_0-t}Wf = u_0 + u_1X + \cdots + u_{k+t_0-1}X^{k+t_0-1}$ and $X^{t_0-t}W = v_0 + v_1X + \cdots + v_{t_0-1}X^{t_0-1} + X^{t_0}$. Then, $\mathbf{u} = (u_0, \ldots, u_{k+t_0-1})$ and $\mathbf{v} = (v_0, \ldots, v_{t_0-1})$ are a solution of (2.1).

Now, let $\mathbf{u} = (u_0, \ldots, u_{k+t_0-1})$ and $\mathbf{v} = (v_0, \ldots, v_{t_0-1})$ be a solution of (2.1). Set $U = u_0 + u_1X + \cdots + u_{k+t_0-1}X^{k+t_0-1}$ and $V = v_0 + v_1X + \cdots + v_{t_0-1}X^{t_0-1} + X^{t_0}$. Then, we have $V\,(\boldsymbol{y}) = U\,(\mathbf{g})$. Since $rank\,(\boldsymbol{y} - f\,(\mathbf{g})) \le t_0$, we also have $rank\,(V\,(\boldsymbol{y} - f\,(\mathbf{g}))) \le t_0$, that is, $rank\,((U - Vf)\,(\mathbf{g})) \le t_0$. Thus, By Proposition 1.49, there is a monic skew polynomial $L \in S[X,\sigma]_{<t_0+1}$ such that $(L\,(U - Vf))\,(\mathbf{g}) = \mathbf{0}$. As $\deg\,(L\,(U - Vf)) \le 2t_0 + k - 1 \le n - 1$, by Corollary 1.48, $L\,(U - Vf) = 0$. Since L is monic, we have $U - Vf = 0$. ■

Lemma 2.7 allows to obtain Algorithm 1.

Algorithm 1: Decoding Gabidulin codes up to half the minimum distance

---

Input: a received word $\mathbf{y} \in S^n$ of the Gabidulin code $Gab_k\,(\mathbf{g})$.

Output: $f \in S[X,\sigma]_{<k}$ such that $rank\,(\boldsymbol{y} - f\,(\mathbf{g})) \le \lfloor (n-k)/2 \rfloor$ or "decoding failure".

Solve linear equation (2.1)
if *(2.1) has no solution* **then**
  return *"decoding failure"*
else
  Set $U = u_0 + u_1X + \cdots + u_{k+t_0-1}X^{k+t_0-1}$ and $V = v_0 + v_1X + \cdots + v_{t_0-1}X^{t_0-1} + X^{t_0}$ where $\mathbf{u} = (u_0, \ldots, u_{k+t_0-1})$ and $\mathbf{v} = (v_0, \ldots, v_{t_0-1})$ are a solution of (2.1).
  Compute the quotient Q and the remainder P on the left Euclidean division of U by V in $S[X,\sigma]$.
  if $P \neq 0$ **then**
    return *"decoding failure"*
  else
    return Q

Theorem 2.8 *Let* $\mathbf{y} \in S^n$ *be a received word of the Gabidulin code* $Gab_k\,(\mathbf{g})$. *Let* $f \in S[X,\sigma]$. *Then, Algorithm 1 returns* f *if and only if* $\deg\,(f) < k$ *and* $rank\,(\boldsymbol{y} - f\,(\mathbf{g})) \le t_0$.

Proof. Assume that Algorithm 1 returns f , then $U = Vf$ where U and V are as in Algorithm 1. Since $\deg\,(U) \le k + t_0 - 1$, we have $\deg\,(f) < k$. As $V\,(\boldsymbol{y}) = U\,(\mathbf{g})$, we also have $V\left(\boldsymbol{y}-f\left(\mathbf{g}\right)\right)=\mathbf{0}$. Thus, by Proposition 1.50, $rank\left(\boldsymbol{y}-f\left(\mathbf{g}\right)\right)\leq t_0$. The converse is given by Lemma 2.7. ■

Recall that one can use the Smith normal form to solve (2.1). Thus, an implementation and a simulation example of Algorithm 1 are given in Appendix A. In the next section we will show that one can also use the iterative method similarly to [41].

2.3 Interleaved Gabidulin codes

Recall that an interleaved Gabidulin code is a direct sum of several Gabidulin codes [46], [67]. In this subsection, we give the properties of interleaved Gabidulin codes, establish a key equation and give an algorithm to solve it.

2.3.1 Definition and properties

Let $l\in\{1,\ldots,\ell\}$. Let $n^{(l)}$ and $k^{(l)}$ be the integers such that $0<k^{(l)}\leq n^{(l)}\leq m$.
Let $\mathbf{g}^{(l)}=\left(g_1^{(l)},\ldots,g_{n^{(l)}}^{(l)}\right)$, where $\{g_1^{(l)},\ldots,g_{n^{(l)}}^{(l)}\}$ is a R-linear independent subset of S. The rank distance of $Gab_{k^{(l)}}\left(\mathbf{g}^{(l)}\right)$ is denoted by $d^{(l)}$. The concatenation of ℓ vectors $\mathbf{c}^{(1)}\in S^{n^{(1)}},\ldots,\mathbf{c}^{(\ell)}\in S^{n^{(\ell)}}$ is denoted by $\left(\mathbf{c}^{(1)}\cdots\mathbf{c}^{(\ell)}\right)\in S^{n^{(1)}+\cdots+n^{(\ell)}}$.

Definition 2.9 *An* ***interleaved Gabidulin code****,* $IGab_{\left(k^{(1)},\ldots,k^{(\ell)}\right)}\left(\mathbf{g}^{(1)},\ldots,\mathbf{g}^{(\ell)}\right)$*, is the set*

$$\left\{\left(\mathbf{c}^{(1)}\cdots\mathbf{c}^{(\ell)}\right)\ :\mathbf{c}^{(l)}\ \in Gab_{k^{(l)}}\left(\mathbf{g}^{(l)}\right),\ l=1,\ldots,\ell\right\}.$$

We observe that if $\ell=1$ then an interleaved Gabidulin code is a Gabidulin code.

Proposition 2.10 *The interleaved Gabidulin code* $IGab_{\left(k^{(1)},\ldots,k^{(\ell)}\right)}\left(\mathbf{g}^{(1)},\ldots,\mathbf{g}^{(\ell)}\right)$ *is a free linear rank code of rank* $k^{(1)}+\cdots+k^{(\ell)}$ *and rank distance* $\min_{l\in\{1,\ldots,\ell\}}\left\{d^{(l)}\right\}$.

Proof. The generator matrix of $IGab_{\left(k^{(1)},\ldots,k^{(\ell)}\right)}\left(\mathbf{g}^{(1)},\ldots,\mathbf{g}^{(\ell)}\right)$ is on the form $diag\left(\mathbf{G}^{(1)},\ldots,\mathbf{G}^{(\ell)}\right)$, where $\mathbf{G}^{(l)}$ is a generator matrix of $Gab_{k^{(l)}}\left(\mathbf{g}^{(l)}\right)$, for $l=1,\ldots,\ell$. Thus, $IGab_{\left(k^{(1)},\ldots,k^{(\ell)}\right)}\left(\mathbf{g}^{(1)},\ldots,\mathbf{g}^{(\ell)}\right)$ is a free linear rank code of rank $k^{(1)}+\cdots+k^{(\ell)}$.

Let $l_0\in\{1,\ldots,\ell\}$ such that $d^{(l_0)}=\min_{l\in\{1,\ldots,\ell\}}\left\{d^{(l)}\right\}$. Then, there is $\mathbf{c}^{(l_0)}\in Gab_{k^{(l_0)}}\left(\mathbf{g}^{(l_0)}\right)$ such that $rank\left(\mathbf{c}^{(l_0)}\right)=d^{(l_0)}$. Let $\mathbf{x}=\left(\mathbf{x}^{(1)}\cdots\mathbf{x}^{(\ell)}\right)\in S^{n^{(1)}+\cdots+n^{(\ell)}}$ defined by $\mathbf{x}^{(l_0)}=\mathbf{c}^{(l_0)}$ and $\mathbf{x}^{(l)}=\mathbf{0}$ if $l\in\{1,\ldots,\ell\}\setminus\{l_0\}$. Then, $\mathbf{x}\in IGab_{\left(k^{(1)},\ldots,k^{(\ell)}\right)}\left(\mathbf{g}^{(1)},\ldots,\mathbf{g}^{(\ell)}\right)$ and $rank\left(\mathbf{x}\right)=d^{(l_0)}$. Let $\mathbf{c}=\left(\mathbf{c}^{(1)}\ldots\mathbf{c}^{(\ell)}\right)\in IGab_{\left(k^{(1)},\ldots,k^{(\ell)}\right)}\left(\mathbf{g}^{(1)},\ldots,\mathbf{g}^{(\ell)}\right)\setminus\{\mathbf{0}\}$, then there is $l_1\in\{1,\ldots,\ell\}$ such that $\mathbf{c}^{(l_1)}\neq\mathbf{0}$. Consequently, $d^{(l_0)}\leq d^{(l_1)}\leq rank\left(\mathbf{c}^{(l_1)}\right)\leq rank\left(\mathbf{c}\right)$. Thus, the rank distance of $IGab_{\left(k^{(1)},\ldots,k^{(\ell)}\right)}\left(\mathbf{g}^{(1)},\ldots,\mathbf{g}^{(\ell)}\right)$ is $d^{(l_0)}$. ■

Corollary 2.11 *If* $k^{(l)}=k^{(1)}$ *and* $n^{(l)}=m$*, for* $l=1,\ldots,\ell$*, then* $IGab_{\left(k^{(1)},\ldots,k^{(\ell)}\right)}\left(\mathbf{g}^{(1)},\ldots,\mathbf{g}^{(\ell)}\right)$ *is an MRD code.*

Proof. Assume that $k^{(l)} = k^{(1)}$ and $n^{(l)} = m$, for $l = 1, \dots, \ell$. We have

$$\begin{aligned} \left| IGab_{\left(k^{(1)},\dots,k^{(\ell)}\right)}\left(g^{(1)}, \dots, g^{(\ell)}\right) \right| &= \left| S^{k^{(1)}+\cdots+k^{(\ell)}} \right| \\ &= \left| S^{\ell k^{(1)}} \right| \\ &= \left| S^{\ell\left(n^{(1)}-d^{(1)}+1\right)} \right| \\ &= \left| R^{m\ell\left(n^{(1)}-d^{(1)}+1\right)} \right| \\ &= |R|^{m\ell\left(m-d^{(1)}+1\right)} \end{aligned}$$

■

Notation 2.12 *Recall that for* $\mathbf{U} \in S[X,\sigma]^{\ell+1}$*, the l-th component of* $\mathbf{U}$ *is denoted by* $U^{(l)}$*, for* l *in* $\{0, \dots, \ell\}$*, i.e.* $\mathbf{U} = \left(U^{(0)}, \dots, U^{(\ell)}\right)$*. In order to simplify the notations, the element* $\left(A^{(1)}, \dots, A^{(\ell)}\right)$ *in* $S[X,\sigma]^{\ell}$ *is denoted by* $\hat{\mathbf{A}}$*.*

For the remainder of this section, let $\mathbf{y} = \left(\mathbf{y}^{(1)} \cdots \mathbf{y}^{(\ell)}\right) \in S^{n^{(1)}+\cdots+n^{(\ell)}}$ be a received word of the interleaved Gabidulin code $IGab_{\left(k^{(1)},\dots,k^{(\ell)}\right)}\left(\mathbf{g}^{(1)}, \dots, \mathbf{g}^{(\ell)}\right)$. The following theorem is the analogue of [41, Theorem 12].

Theorem 2.13 *Let* $\tau \in \mathbb{N}$*. Then, the following statements are equivalent.*

(i) There is $\mathbf{c} \in IGab_{\left(k^{(1)},\dots,k^{(\ell)}\right)}\left(\mathbf{g}^{(1)}, \dots, \mathbf{g}^{(\ell)}\right)$ *such that* $rank\,(\mathbf{y} - \mathbf{c}) \leq \tau$*.*

(ii) There is $\mathbf{U} \in S[X,\sigma]^{\ell+1}$ *such that:*

1) $U^{(0)}\left(\mathbf{y}^{(l)}\right) = U^{(l)}\left(\mathbf{g}^{(l)}\right)$*, for* $l = 1, \dots, \ell$*;*

2) $\deg\left(U^{(l)}\right) - k^{(l)} \leq \deg\left(U^{(0)}\right) - 1$*, for* $l = 1, \dots, \ell$*;*

3) $U^{(0)}$ *is monic;*

4) $\deg\left(U^{(0)}\right) \leq \tau$*;*

5) the remainder of the left Euclidean division of $U^{(l)}$ *by* $U^{(0)}$ *is equal to zero, for* $l = 1, \dots, \ell$*.*

Proof. Assume there is $\mathbf{c} \in IGab_{\left(k^{(1)},\dots,k^{(\ell)}\right)}\left(\mathbf{g}^{(1)}, \dots, \mathbf{g}^{(\ell)}\right)$ such that $rank\,(\mathbf{y} - \mathbf{c}) \leq \tau$. Let $f^{(l)} \in S[X,\sigma]_{<k^{(l)}}$, $l = 1, \dots, \ell$, such that $\mathbf{c} = \left(f^{(1)}\left(\mathbf{g}^{(1)}\right) \cdots f^{(\ell)}\left(\mathbf{g}^{(\ell)}\right)\right)$. Then, by Proposition 1.49, there exists a monic skew polynomial $U^{(0)} \in S[X,\sigma]$ of degree $rank\,(\mathbf{y} - \mathbf{c})$ such that, for $l = 1, \dots, \ell$, $U^{(0)}\left(\mathbf{y}^{(l)} - f^{(l)}\left(\mathbf{g}^{(l)}\right)\right) = \mathbf{0}$, i.e., $U^{(0)}\left(\mathbf{y}^{(l)}\right) = \left(U^{(0)}f^{(l)}\right)\left(\mathbf{g}^{(l)}\right)$. Set $U^{(l)} = U^{(0)}f^{(l)}$, for $l = 1, \dots, \ell$, then $\left(U^{(0)}, \dots, U^{(\ell)}\right)$ verifies the five conditions of Theorem 2.13 (ii).

Conversely, assume there is $\mathbf{U} \in S[X,\sigma]^{\ell+1}$ verifying the five conditions of Theorem 2.13 (ii). Let $l \in \{1, \dots, \ell\}$ and let $f^{(l)}$ be the quotient of the left Euclidean division of $U^{(l)}$ by $U^{(0)}$, then $U^{(l)} = U^{(0)}f^{(l)}$. As $\deg\left(U^{(l)}\right) - k^{(l)} \leq \deg\left(U^{(0)}\right) - 1$, we have $\deg\left(f^{(l)}\right) \leq k^{(l)} - 1$. Since $U^{(0)}\left(\mathbf{y}^{(l)}\right) = U^{(l)}\left(\mathbf{g}^{(l)}\right)$, we also have $U^{(0)}\left(\mathbf{y}^{(l)} - f^{(l)}\left(\mathbf{g}^{(l)}\right)\right) = \mathbf{0}$. Thus, by Proposition 1.50,

$$rank\left(\mathbf{y} - \left(f^{(1)}\left(\mathbf{g}^{(1)}\right) \cdots f^{(\ell)}\left(\mathbf{g}^{(\ell)}\right)\right)\right) \leq \deg\left(U^{(0)}\right) \leq \tau.$$

■

Definition 2.14 *(the key equation)*

We say that $\mathbf{U} \in S[X,\sigma]^{\ell+1}$ *is a solution of the key equation if :*

- $U^{(0)}\left(\mathbf{y}^{(l)}\right) = U^{(l)}\left(\mathbf{g}^{(l)}\right)$, *for* $l = 1,\ldots,\ell$;
- $\deg\left(U^{(l)}\right) - k^{(l)} \leq \deg\left(U^{(0)}\right) - 1$, *for* $l = 1,\ldots,\ell$.
- $U^{(0)}$ *is monic;*

A solution $\mathbf{U}$ *is called minimal if* $\deg\left(U^{(0)}\right)$ *is minimal.*

In finite fields, the resolution of the key equation given in Definition 2.14 is equivalent to the problem of multi-sequence generalized linear skew-feedback shift register introduced in [60]. In [60], Puchinger et al. solved this problem using row reduction. We will solve the key equation using the iterative method introduced in [23], because it is easy to extend this method to modules and finite rings (see, for example [42], [56], [13], [80], [1], [41], [40]). Note that in [8], Bartz and Wachter-Zeh used this iterative method for decoding interleaved subspace and Gabidulin codes, because its complexity is better than Gaussian elimination. Further, it allows to compute a minimal Gröbner basis for the interpolation module.

2.3.2 Iterative solving the key equation

Similar to [41], [1], we give an iterative algorithm that allows to solve the key equation. Recall that the elements a and b in S are said to be associated if $b = ua$ for some unit $u \in S$.

Notation 2.15 *Since associatedness is an equivalence relation on* S*, we denote*

- the equivalence class of $a \in S$ *by* $[a]$*;*

- a complete set of representatives of the equivalence classes by $[S]$*, without loss of generality, assume that* $1 \in [S]$*;*

- and let $[S]^* := [S]\backslash\{0\}$.

As $S = S_{(1)} \times \cdots \times S_{(\rho)}$, where $S_{(j)}$ is a finite chain ring and a generator of its maximal ideal is in $R_{(j)}$, we have the following:

Lemma 2.16 *For all* $a \in S$*,* a *and* $\sigma(a)$ *are associated.*

Proof. Let $\pi_{(j)}$ be a generator of the maximal ideal of $R_{(j)}$ for $j = 1,\ldots,\rho$. Then $\pi_{(j)}$ be a generator of the maximal ideal of $S_{(j)}$. So, for all $a = \left(a_{(1)},\ldots,a_{(\rho)}\right) \in S$, there exist a unit $u_{(j)} \in S_{(j)}$ and $i_{(j)} \in \mathbb{N}$ such that $a_{(j)} = u_{(j)}\pi_{(j)}^{i_{(j)}}$, for $j = 1,\ldots,\rho$. Therefore $a = uv$ where $u = \left(u_{(1)},\ldots,u_{(\rho)}\right)$ and $v = \left(\pi_{(1)}^{i_{(1)}},\ldots,\pi_{(\rho)}^{i_{(\rho)}}\right)$. Since $v \in R$, we have $\sigma(a) = \sigma(u)\,v$. Thus a and $\sigma(a)$ are associated because u is a unit in S. ■

Notation 2.17 *Let* $\mathbf{y} = \left(\mathbf{y}^{(1)}\cdots\mathbf{y}^{(\ell)}\right) \in S^{n^{(1)}+\cdots+n^{(\ell)}}$ *be a received word of the interleaved Gabidulin code* $IGab_{\left(k^{(1)},\ldots,k^{(\ell)}\right)}\left(\mathbf{g}^{(1)},\ldots,\mathbf{g}^{(\ell)}\right)$. *Set* $\mathbf{g} = \left(\mathbf{g}^{(1)}\cdots\mathbf{g}^{(\ell)}\right)$. *We denote by* $M[\mathbf{y},\mathbf{g}]$ *the set of all* $\mathbf{U}$ *in* $S[X,\sigma]^{\ell+1}$ *such that* $U^{(0)}\left(\mathbf{y}^{(l)}\right) = U^{(l)}\left(\mathbf{g}^{(l)}\right)$, *for* $l = 1,\ldots,\ell$, *that is,* $U^{(0)}\left(y_i^{(l)}\right) = U^{(l)}\left(g_i^{(l)}\right)$, *for* $l = 1,\ldots,\ell$ *and* $i = 1,\ldots,n^{(l)}$.

The set $M[\mathbf{y},\mathbf{g}]$ is a $S[X,\sigma]$-submodule of $S[X,\sigma]^{\ell+1}$ and by Definition 2.14, all the solutions of the key equation are in $M[\mathbf{y},\mathbf{g}]$. Therefore, to find these solutions, just find a Gröbner basis for $M[\mathbf{y},\mathbf{g}]$ with a monomial order $\succeq$ that we will specify later. To compute a Gröbner basis for $M[\mathbf{y},\mathbf{g}]$, we will use the iterative method described in [56].

Notation 2.18 *Set* $n^{(0)} := 0$. *We define by induction the subsets* $M[\mathbf{y},\mathbf{g}]_{(l,i)}$ *as follows:* $M[\mathbf{y},\mathbf{g}]_{(0,0)} = S[X,\sigma]^{\ell+1}$ *and for all* $(l,i) \in \{1,\ldots,\ell\} \times \left\{1,\ldots,n^{(l)}\right\}$, $M[\mathbf{y},\mathbf{g}]_{(l,i)}$ *is the set of all* $\mathbf{U}$ *in* $M[\mathbf{y},\mathbf{g}]_{(\underline{l},\underline{i})}$ *such that* $U^{(0)}\left(y_i^{(l)}\right) = U^{(l)}\left(g_i^{(l)}\right)$, *where*

$$(\underline{l},\underline{i}) = \begin{cases} \left(l-1, n^{(l-1)}\right) & \text{if } i=1 \\ (l, i-1) & \text{else} \end{cases}$$

We have $M[\mathbf{y},\mathbf{g}]_{(0,0)} \supset M[\mathbf{y},\mathbf{g}]_{(1,1)} \supset \cdots \supset M[\mathbf{y},\mathbf{g}]_{(1,n^{(1)})} \supset M[\mathbf{y},\mathbf{g}]_{(2,1)} \supset \cdots \supset M[\mathbf{y},\mathbf{g}]_{(2,n^{(2)})} \supset \cdots \supset M[\mathbf{y},\mathbf{g}]_{(\ell,1)} \supset \cdots \supset M[\mathbf{y},\mathbf{g}]_{(\ell,n^{(\ell)})} = M[\mathbf{y},\mathbf{g}]$. Note that as in [1] a Gröbner basis for $S[X,\sigma]^{\ell+1}$ is $\mathcal{B}_{(0,0)} := \left\{s\mathbf{e}^{(r)}\right\}_{0\leq r\leq \ell,\ s\in[S]^*}$. So, we will compute a Gröbner basis, $\mathcal{B} = \left\{\mathbf{V}_{(r,s)}\right\}_{0\leq r\leq \ell,\ s\in[S]^*}$, for $M[\mathbf{y},\mathbf{g}]$ which has the same properties as $\mathcal{B}_{(0,0)}$, that is, for all (r,s), $ind(lm\left(\mathbf{V}_{(r,s)}\right)) = r$, $lc\left(\mathbf{V}_{(r,s)}\right) \in [s]$, and $\deg\left(\mathbf{V}_{(r,s)}\right)$ is minimal among the degree of all $\mathbf{U} \in M[\mathbf{y},\mathbf{g}]$ with $ind(lm(\mathbf{U})) = r$, $lc(\mathbf{U}) \in [s]$.

Let $(l,i) \in \{1,\ldots,\ell\} \times \left\{1,\ldots,n^{(l)}\right\}$. Assume that $M[\mathbf{y},\mathbf{g}]_{(\underline{l},\underline{i})}$ has a Gröbner basis $\mathcal{B}_{(\underline{l},\underline{i})} = \left\{\mathbf{V}_{(r,s)}\right\}_{0\leq r\leq \ell,\ s\in[S]^*}$ such that for all (r,s), $ind(lm\left(\mathbf{V}_{(r,s)}\right)) = r$, $lc\left(\mathbf{V}_{(r,s)}\right) \in [s]$, and $\deg\left(\mathbf{V}_{(r,s)}\right)$ is minimal among the degree of all $\mathbf{U} \in M[\mathbf{y},\mathbf{g}]_{(\underline{l},\underline{i})}$ with $ind(lm(\mathbf{U})) = r$, $lc(\mathbf{U}) \in [s]$.

- Let $\mathcal{J}_{(r,s)}$ be the set of all $(r',s') \in \{0,\ldots,\ell\} \times [S]^*$ such that $lm\left(\mathbf{V}_{(r',s')}\right) \prec lm\left(\mathbf{V}_{(r,s)}\right)$.
- Let $D_{(l,i)} : M[\mathbf{y},\mathbf{g}]_{(\underline{l},\underline{i})} \longrightarrow S$ be defined as

$$D_{(l,i)}(\mathbf{U}) = U^{(0)}\left(y_i^{(l)}\right) - U^{(l)}\left(g_i^{(l)}\right).$$

- The discrepancy of $\mathbf{V}_{(r,s)}$ is given by

$$\Delta_{(r,s)} := D_{(l,i)}\left(\mathbf{V}_{(r,s)}\right).$$

- Let $b_{(r,s)} \in S$ such that

$$\sigma\left(\Delta_{(r,s)}\right) - b_{(r,s)}\Delta_{(r,s)} = 0.$$

Lemma 2.19 *With the above notations,*
(a) $D_{(l,i)}$ *is an* S*-module homomorphism;*
(b) $M[\mathbf{y},\mathbf{g}]_{(l,i)} = \left\{\mathbf{U} \in M[\mathbf{y},\mathbf{g}]_{(\underline{l},\underline{i})} : D_{(l,i)}(\mathbf{U}) = 0\right\}$;
(c) $\left(X - b_{(r,s)}\right)\mathbf{V}_{(r,s)} \in M[\mathbf{y},\mathbf{g}]_{(l,i)}$.

Using a Gröbner basis, $\left\{\mathbf{V}_{(r,s)}\right\}_{0\leq r\leq \ell,\ s\in[S]^*}$, for $M[\mathbf{y},\mathbf{g}]_{(\underline{l},\underline{i})}$, we now show how one can compute a Gröbner basis for $M[\mathbf{y},\mathbf{g}]_{(l,i)}$. Let $\left\{\mathbf{V}'_{(r,s)}\right\}_{0\leq r\leq \ell,\ s\in[S]^*} \subset S[X,\sigma]^{\ell+1}$ be defined as :

- if $\Delta_{(r,s)} = 0$ then
$$\mathbf{V}'_{(r,s)} := \mathbf{V}_{(r,s)} \tag{2.2}$$

- if $\Delta_{(r,s)} \neq 0$ and there exist $\theta_{(r';s')} \in S$, $(r', s') \in \mathcal{J}_{(r,s)}$ such that
$$\Delta_{(r,s)} + \sum_{(r',s') \in \mathcal{J}_{(r,s)}} \theta_{(r',s')} \Delta_{(r',s')} = 0 \tag{2.3}$$

then
$$\mathbf{V}'_{(r,s)} := \mathbf{V}_{(r,s)} + \sum_{(r',s') \in \mathcal{J}_{(r,s)}} \theta_{(r',s')} \mathbf{V}_{(r',s')} \tag{2.4}$$

- otherwise,
$$\mathbf{V}'_{(r,s)} := \left(X - b_{(r,s)}\right) \mathbf{V}_{(r,s)} \tag{2.5}$$

Proposition 2.20 *Let* $\left\{\mathbf{V}'_{(r,s)}\right\}_{0 \le r \le \ell,\ s \in [S]^*}$ *be the subset of* $S[X, \sigma]^{\ell+1}$ *computed using (2.2), (2.4) and (2.5). Then,* $\left\{\mathbf{V}'_{(r,s)}\right\}_{0 \le r \le \ell,\ s \in [S]^*}$ *is a Gröbner basis for* $M[\mathbf{y}, \mathbf{g}]_{(l,i)}$ *and for all* (r, s)*,* $ind(lm\left(\mathbf{V}'_{(r,s)}\right)) = r$*,* $lc\left(\mathbf{V}'_{(r,s)}\right) \in [s]$*, and* $\deg\left(\mathbf{V}'_{(r,s)}\right)$ *is minimal among the degree of all* $\mathbf{U} \in M[\mathbf{y}, \mathbf{g}]_{(l,i)}$ *with* $ind(lm\,(\mathbf{U})) = r$*,* $lc\,(\mathbf{U}) \in [s]$*.*

Proof. By the definition of $\mathbf{V}'_{(r,s)}$, we have $\mathbf{V}'_{(r,s)} \in M[\mathbf{y}, \mathbf{g}]_{(l,i)}$, $ind(lm\left(\mathbf{V}'_{(r,s)}\right)) = r$, $lc\left(\mathbf{V}'_{(r,s)}\right) \in [s]$. We now prove that $\deg\left(\mathbf{V}'_{(r,s)}\right)$ is minimal among the degree of all $\mathbf{U} \in M[\mathbf{y}, \mathbf{g}]_{(l,i)}$ with $ind(lm\,(\mathbf{U})) = r$, $lc\,(\mathbf{U}) \in [s]$. If $\mathbf{V}'_{(r,s)}$ is defined as in (2.2) or (2.4), then the result follows. Assume that $\mathbf{V}'_{(r,s)}$ is defined as in (2.5) and that there is $\mathbf{W} \in M[\mathbf{y}, \mathbf{g}]_{(l,i)}$ such that $ind(lm\,(\mathbf{W})) = r$, $lc\,(\mathbf{W}) \in [s]$ and $\deg(\mathbf{W}) < \deg\left(\mathbf{V}'_{(r,s)}\right)$. Then, since $\mathbf{W} \in M[\mathbf{y}, \mathbf{g}]_{(\underline{l},\underline{i})}$ and $\deg\left(\mathbf{V}'_{(r,s)}\right) = \deg\left(\mathbf{V}_{(r,s)}\right) + 1$, we have $\deg(\mathbf{W}) = \deg\left(\mathbf{V}_{(r,s)}\right)$. Therefore, as $lc\,(\mathbf{W}) \in [s]$ and $lc\left(\mathbf{V}_{(r,s)}\right) \in [s]$, there is $a \in S$ such that $lm\left(\mathbf{V}_{(r,s)} - a\mathbf{W}\right) \prec lm\left(\mathbf{V}_{(r,s)}\right)$. Consequently, by Proposition 1.60, we have
$$\mathbf{V}_{(r,s)} - a\mathbf{W} = \sum_{(r',s')\ \in \mathcal{J}_{(r,s)}} h_{(r',s')} \mathbf{V}_{(r',s')}$$

where $h_{(r',s')} \in S[X, \sigma]$. By the right Euclidean division of $h_{(r',s')}$ by $X - b_{(r',s')}$ there exist $Q_{(r',s')} \in S[X, \sigma]$ and $\lambda_{(r',s')} \in S$ such that
$$h_{(r',s')} = Q_{(r',s')}\left(X - b_{(r',s')}\right) + \lambda_{(r',s')}.$$

Hence, we have
$$\mathbf{V}_{(r,s)} - a\mathbf{W} = \sum_{(r',s')\ \in \mathcal{J}_{(r,s)}} Q_{(r',s')}\left(X - b_{(r',s')}\right) \mathbf{V}_{(r',s')} + \sum_{(r',s')\ \in \mathcal{J}_{(r,s)}} \lambda_{(r',s')} \mathbf{V}_{(r',s')}.$$

Consequently, by Lemma 2.19,
$$D_{(l,i)}\left(\mathbf{V}_{(r,s)}\right) = \sum_{(r',s') \in \mathcal{J}_{(r,s)}} \lambda_{(r',s')} D_{(l,i)}\left(\mathbf{V}_{(r',s')}\right)$$

This contradicts the definition of $\mathbf{V}'_{(r,s)}$. Thus, the result follows.

Now we prove that $\left\{\mathbf{V}'_{(r,s)}\right\}_{0\leq r\leq \ell,\ s\in[S]^*}$ is a Gröbner basis for $M[\mathbf{y},\mathbf{g}]_{(l,i)}$. Let $\mathbf{U} \in M[\mathbf{y},\mathbf{g}]_{(l,i)}$, $r = ind(lm\,(\mathbf{U}))$, $s \in [S]^*$ such that $lc\,(\mathbf{U}) \in [s]$ and $\alpha = \deg(\mathbf{U}) - \deg\left(\mathbf{V}'_{(r,s)}\right)$. Then, $lm\,(\mathbf{U}) = X^{\alpha} lm\left(\mathbf{V}'_{(r,s)}\right)$ and $lc\,(\mathbf{U}) \in \left\langle \sigma^{\alpha}\left(lc\left(\mathbf{V}'_{(r,s)}\right)\right)\right\rangle$. Thus, by Proposition 1.59, the result follows. ■

Proposition 2.20 justifies Algorithm 2.

Algorithm 2: a Gröbner basis of the key equation

Input: a received vector $\mathbf{y} = \left(\mathbf{y}^{(1)}\cdots\mathbf{y}^{(\ell)}\right) \in S^{n^{(1)}+\cdots+n^{(\ell)}}$ of the interleaved Gabidulin code $IGab_{\left(k^{(1)},\ldots,k^{(\ell)}\right)}\left(\mathbf{g}^{(1)},\ldots,\mathbf{g}^{(\ell)}\right)$.

Output: a Gröbner basis $\left\{\mathbf{V}_{(r,s)}\right\}_{0\leq r\leq\ell,\ s\in[S]^*}$ for the module $M[\mathbf{y},\mathbf{g}]$.

$\mathcal{J} \leftarrow \{0,\ldots,\ell\}\times[S]^*$
for $(r,s)\in\mathcal{J}$ **do**
  $\mathbf{V}_{(r,s)} \leftarrow s\mathbf{e}^{(r)}$
for $l \leftarrow 1$ ***to*** ℓ **do**
  for $i \leftarrow 1$ ***to*** $n^{(l)}$ **do**
    for $(r,s)\in\mathcal{J}$ **do**
      $\Delta_{(r,s)} \leftarrow V^{(0)}_{(r,s)}\left(y_i^{(l)}\right) - V^{(l)}_{(r,s)}\left(g_i^{(l)}\right)$
    for $(r,s)\in\mathcal{J}$ **do**
      if $\Delta_{(r,s)} = 0$ **then**
        $\mathbf{V}'_{(r,s)} \leftarrow \mathbf{V}_{(r,s)}$
      else
        if *there exists a nonempty* $\mathcal{J}'\subset\mathcal{J}$ *such that for* $(r',s')\in\mathcal{J}'$, $lm\left(\mathbf{V}_{(r',s')}\right) \prec lm\left(\mathbf{V}_{(r,s)}\right)$ *and* $\Delta_{(r,s)} + \sum_{(r',s')\in\mathcal{J}'}\theta_{(r',s')}\Delta_{(r',s')} = 0$ *for some* $\theta_{(r',s')}\in S$, **then**
          $\mathbf{V}'_{(r,s)} \leftarrow \mathbf{V}_{(r,s)} + \sum_{(r',s')\in\mathcal{J}'}\theta_{(r',s')}\mathbf{V}_{(r',s')}$
        else
          $\mathbf{V}'_{(r,s)} \leftarrow \left(X - b_{(r,s)}\right)\mathbf{V}_{(r,s)}$ where $b_{(r,s)}$ is an element of S such that $\sigma\left(\Delta_{(r,s)}\right) - b_{(r,s)}\Delta_{(r,s)} = 0$.
    for $(r,s)\in\mathcal{J}$ **do**
      $\mathbf{V}_{(r,s)} \leftarrow \mathbf{V}'_{(r,s)}$
return $\left\{\mathbf{V}_{(r,s)}\right\}_{0\leq r\leq\ell,\ s\in[S]^*}$

Remark 2.21 *Since $S = S_{(1)}\times\cdots\times S_{(\rho)}$, where $S_{(j)}$ is a finite chain ring, the equation (2.3) is easy to solve in $S_{(j)}$. Indeed, in $S_{(j)}$ this equation is equivalent to: $\Delta_{(r',s')}$ divides $\Delta_{(r,s)}$ for some (r',s') in $\mathcal{J}_{(r,s)}$. Thus, analogous to [13, Algorithm VI.5], it is easy to compute a Gröbner basis of Algorithm 2 in $S_{(j)}[X,\sigma_{(j)}]^{\ell+1}$, and then to apply the "strong join" method described in [55] to obtain a Gröbner basis in $S[X,\sigma]^{\ell+1}$.*

Note that the monomial order of Algorithm 2 is not specified. We now define a monomial order that will allow to give the solutions of the key equation.

Definition 2.22 *Set $k^{(0)} := 1$. The relation $\preceq_{\left(k^{(0)},\ldots,k^{(\ell)}\right)}$ is defined on the monomial of $S[X,\sigma]^{\ell+1}$ by:*

$$X^{\alpha_1}\mathbf{e}^{(l_1)} \preceq_{\left(k^{(0)},\ldots,k^{(\ell)}\right)} X^{\alpha_2}\mathbf{e}^{(l_2)}$$

if and only if $\alpha_1 - k^{(l_1)} < \alpha_2 - k^{(l_2)}$ or [$\alpha_1 - k^{(l_1)} = \alpha_2 - k^{(l_2)}$ and $l_1 \geq l_2$].

By [64, Theorem 29], the relation $\preceq_{\left(k^{(0)},\ldots,k^{(\ell)}\right)}$ is a monomial order.

Proposition 2.23 *The vector $\mathbf{U} \in M[\mathbf{y},\mathbf{g}]$ is a solution of the key equation if and only if, w.r.t. $\preceq_{\left(k^{(0)},\ldots,k^{(\ell)}\right)}$, ind($lm\left(\mathbf{U}\right)) = 0$ and $lc\left(\mathbf{U}\right) = 1$.*

Proof. Let $\mathbf{U} \in M[\mathbf{y},\mathbf{g}]$, then $\mathbf{U}$ is a solution of the key equation if and only if, $U^{(0)}$ is monic and $\deg\left(U^{(l)}\right) - k^{(l)} \leq \deg\left(U^{(0)}\right) - 1$, for $l = 1,\ldots,\ell$, that is, w.r.t. $\preceq_{\left(k^{(0)},\ldots,k^{(\ell)}\right)}$, *ind*($lm\left(\mathbf{U}\right)) = 0$ and $lc\left(\mathbf{U}\right) = 1$. ■

Now, we can apply Proposition 1.60 to obtain all the solutions of the key equation.

Theorem 2.24 *Let $\left\{\mathbf{V}_{(r,s)}\right\}_{0\leq r\leq\ell,\ s\in[S]^*}$ be a Gröbner basis for $M[\mathbf{y},\mathbf{g}]$ obtained by Algorithm 2 w.r.t. $\preceq_{\left(k^{(0)},\ldots,k^{(\ell)}\right)}$. Set $\alpha_{(r,s)} := \deg\left(V^{(r)}_{(r,s)}\right)$.*

(a) The vector $\mathbf{V}_{(0,1)}$ is a minimal solution of the key equation.

(b) All solution $\mathbf{U}$ of the key equation can be written as

$$\mathbf{U} = \sum_{0\leq r\leq\ell,\ s\in[S]^*} w_{(r,s)}\mathbf{V}_{(r,s)}$$

where $w_{(r,s)} \in S[X,\sigma]$, $w_{(0,1)}$ is monic, for all $s \in [S]^*\setminus\{1\}$,

$$\deg\left(w_{(0,s)}\right) + \alpha_{(0,s)} < \deg\left(w_{(0,1)}\right) + \alpha_{(0,1)}$$

and for all $(r,s) \in \{1,\ldots,\ell\} \times [S]^*$,

$$\deg\left(w_{(r,s)}\right) + \alpha_{(r,s)} - k^{(r)} \leq \deg\left(w_{(0,1)}\right) + \alpha_{(0,1)} - k^{(0)}.$$

Proof. (a) By construction of $\mathbf{V}_{(0,1)}$ and by Proposition 2.23, $\mathbf{V}_{(0,1)}$ is a minimal solution.

(b) Let $\mathbf{U}$ be a solution of the key equation. Then, $\mathbf{U} \in M[\mathbf{y},\mathbf{g}]$ and, by Proposition 2.23, *ind*($lm\left(\mathbf{U}\right)) = 0$, $lc\left(\mathbf{U}\right) = 1$, w.r.t. $\preceq_{\left(k^{(0)},\ldots,k^{(\ell)}\right)}$. Let $\alpha = \deg\left(\mathbf{U}\right) - \deg\left(\mathbf{V}_{(0,1)}\right)$, then

$$lm\left(\mathbf{U} - X^{\alpha}\mathbf{V}_{(0,1)}\right) \prec_{\left(k^{(0)},\ldots,k^{(\ell)}\right)} lm\left(\mathbf{U}\right).$$

Therefore since $\mathbf{U} - X^{\alpha}\mathbf{V}_{(0,1)} \in M[\mathbf{y},\mathbf{g}]$, by Proposition 1.60,

$$\mathbf{U} - X^{\alpha}\mathbf{V}_{(0,1)} = \sum_{0\leq r\leq\ell,\ s\in[S]^*} h_{(r,s)}\mathbf{V}_{(r,s)},$$

where $h_{(r,s)} \in S[X,\sigma]$ and

$$lm\left(\mathbf{U} - X^{\alpha}\mathbf{V}_{(0,1)}\right) = \max_{0\leq r\leq \ell,\ s\in[S]^*} \left\{lm\left(h_{(r,s)}\right) lm\left(\mathbf{V}_{(r,s)}\right)\right\}.$$

Set $w_{(0,1)} = X^{\alpha} + h_{(0,1)}$ and $w_{(r,s)} = h_{(r,s)}$ if $(r,s) \neq (0,1)$. Then,

$$\mathbf{U} = \sum_{0\leq r\leq \ell,\ s\in[S]^*} w_{(r,s)}\mathbf{V}_{(r,s)},$$

$w_{(0,1)}$ is monic,

$$lm\left(\mathbf{U}\right) = lm\left(w_{(0,1)}\right) lm\left(\mathbf{V}_{(0,1)}\right)$$

and for all $(r,s) \neq (0,1)$,

$$lm\left(w_{(r,s)}\right) lm\left(\mathbf{V}_{(r,s)}\right) \prec_{\left(k^{(0)},\ldots,k^{(\ell)}\right)} lm\left(\mathbf{U}\right).$$

As $ind(lm\left(\mathbf{V}_{(r,s)}\right)) = r$, we have

$$lm\left(w_{(r,s)}\right) lm\left(\mathbf{V}_{(r,s)}\right) = X^{\deg\left(w_{(r,s)}\right)+\deg\left(V^{(r)}_{(r,s)}\right)}\mathbf{e}^{(r)}.$$

Thus, the result follows. ■

2.4 Decoding algorithms of interleaved Gabidulin codes

In this section, we use the solutions of the key equation to give the minimal list decoding, unique decoding, and error-erasure decoding algorithms of interleaved Gabidulin codes.

2.4.1 Minimal list decoding

In [41], Kuijper and Trautmann used an iterative parametrization approach to give a minimal list decoding algorithm of Gabidulin codes over finite fields. In this subsection, we show that this algorithm can be generalized to interleaved Gabidulin codes over finite principal ideal rings.

Definition 2.25 *Let a received word* $\mathbf{y} \in S^{n^{(1)}+\cdots+n^{(\ell)}}$ *of the interleaved Gabidulin code* $IGab_{\left(k^{(1)},\ldots,k^{(\ell)}\right)}\left(\mathbf{g}^{(1)},\ldots,\mathbf{g}^{(\ell)}\right)$. ***Minimal list decoding*** *consists to find the value of*

$$t_{\min} := \min_{\mathbf{c}\in IGab_{\left(k^{(1)},\ldots,k^{(\ell)}\right)}\left(\mathbf{g}^{(1)},\ldots,\mathbf{g}^{(\ell)}\right)} \left\{rank\left(\mathbf{y}-\mathbf{c}\right)\right\} \tag{2.6}$$

as well as all codewords $\mathbf{c} \in IGab_{\left(k^{(1)},\ldots,k^{(\ell)}\right)}\left(\mathbf{g}^{(1)},\ldots,\mathbf{g}^{(\ell)}\right)$ *such that*

$$rank\left(\mathbf{y}-\mathbf{c}\right) = t_{\min}.$$

Theorem 2.13 and Theorem 2.24 justify Algorithm 3 of minimal list decoding.

Algorithm 3: Minimal list decoding

Input: a received word $\mathbf{y} = \left(\mathbf{y}^{(1)} \cdots \mathbf{y}^{(\ell)}\right) \in S^{n^{(1)}+\cdots+n^{(\ell)}}$ of the interleaved Gabidulin code $IGab_{\left(k^{(1)},\ldots,k^{(\ell)}\right)}\left(\mathbf{g}^{(1)},\ldots,\mathbf{g}^{(\ell)}\right)$.

Output: A list of $\hat{\mathbf{f}} \in S[X,\sigma]_{<k^{(1)}} \times \cdots \times S[X,\sigma]_{<k^{(\ell)}}$ such that $rank\left(\mathbf{y} - \left(f^{(1)}\left(\mathbf{g}^{(1)}\right) \cdots f^{(\ell)}\left(\mathbf{g}^{(\ell)}\right)\right)\right)$ is minimal.

Compute a Gröbner basis $\left\{\mathbf{V}_{(r,s)}\right\}_{0\le r\le \ell,\ s\in[S]^*}$ for the module $M[\mathbf{y},\mathbf{g}]$ as in Algorithm 2 w.r.t. $\preceq_{\left(k^{(0)},\ldots,k^{(\ell)}\right)}$

$\alpha_{(r,s)} \leftarrow \deg\left(V_{(r,s)}^{(r)}\right)$

list $\leftarrow \emptyset$

$j \leftarrow 0$

while *list* $= \emptyset$ **do**

  Compute the set $\mathcal{U}$ of all $\mathbf{U} = \sum_{0\le r\le \ell,\ s\in[S]^*} w_{(r,s)}\mathbf{V}_{(r,s)}$ where $w_{(r,s)} \in S[X,\sigma]$, $w_{(0,1)}$ is monic, $\deg\left(w_{(0,1)}\right) = j$, $\deg\left(w_{(0,s)}\right) + \alpha_{(0,s)} < j + \alpha_{(0,1)}$, for all $s \in [S]^* \setminus \{1\}$, and $\deg\left(w_{(r,s)}\right) + \alpha_{(r,s)} - k^{(r)} \le j + \alpha_{(0,1)} - k^{(0)}$, for all $(r,s) \in \{1,\ldots,\ell\} \times [S]^*$

  foreach $\mathbf{U} \in \mathcal{U}$ **do**

    for $l \leftarrow 1$ ***to*** ℓ **do**

      Compute the quotient $Q^{(l)}$ and the remainder $P^{(l)}$ on the left Euclidean division of $U^{(l)}$ by $U^{(0)}$ in $S[X,\sigma]$

    if *for all* $l \in \{1,\ldots,\ell\}$, $P^{(l)} = 0$ **then**

      list $\leftarrow$ *list* $\cup \{\hat{\mathbf{Q}}\}$

  $j \leftarrow j+1$

return *list*

In general, the list size of minimal list decoding might be greater than one. In the next subsection, we give a sufficient condition so that the list size is one and a decoding algorithm in this case.

2.4.2 Unique decoding beyond the error correction capability

Let $t_0 := \left\lfloor \left(\min_{l\in\{1,\ldots,\ell\}}\left\{d^{(l)}\right\} - 1\right)/2 \right\rfloor$ be the error correction capability of the interleaved Gabidulin code $IGab_{\left(k^{(1)},\ldots,k^{(\ell)}\right)}\left(\mathbf{g}^{(1)},\ldots,\mathbf{g}^{(\ell)}\right)$ and let $\mathbf{y} = \left(\mathbf{y}^{(1)} \cdots \mathbf{y}^{(\ell)}\right)$ be a received word. We may have $t_{\min} \le t_0$ or $t_0 < t_{\min}$. Moreover, if $t_{\min} \le t_0$, then the list size of minimal list decoding is one. The next lemma give a necessary and sufficient condition so that $t_{\min} \le t_0$.

Lemma 2.26 *Let* $\mathbf{U}$ *be a minimal solution of the key equation and* $\hat{\mathbf{f}} \in S[X,\sigma]_{<k^{(1)}} \times \cdots \times S[X,\sigma]_{<k^{(\ell)}}$. *The following statements are equivalent.*

(i) $rank\left(\mathbf{y} - \left(f^{(1)}\left(\mathbf{g}^{(1)}\right) \cdots f^{(\ell)}\left(\mathbf{g}^{(\ell)}\right)\right)\right) \le t_0$.

(ii) It holds both that:

1) $\deg\left(U^{(0)}\right) \leq t_0$*;*
2) $U^{(l)} = U^{(0)} f^{(l)}$*, for* $l = 1, \ldots, \ell$*.*

Proof. By Theorem 2.13, (ii)$\Longrightarrow$ (i).

Proof that (i)$\Longrightarrow$ (ii). Assume that $rank\left(\mathbf{y} - \left(f^{(1)}\left(\mathbf{g}^{(1)}\right) \cdots f^{(\ell)}\left(\mathbf{g}^{(\ell)}\right)\right)\right) \leq t_0$. Then, by Theorem 2.13, there is $\left(W^{(0)}, W^{(1)}, \ldots, W^{(\ell)}\right) \in S[X,\sigma]^{\ell+1}$ verifying the five conditions of Theorem 2.13 (ii), with $\tau = t_0$. Thus, since $\mathbf{U}$ is minimal, we have $\deg\left(U^{(0)}\right) \leq \deg\left(W^{(0)}\right) \leq t_0$. Set

$$\boldsymbol{\varepsilon} = \left(\boldsymbol{\varepsilon}^{(1)} \cdots \boldsymbol{\varepsilon}^{(\ell)}\right) = \mathbf{y} - \left(f^{(1)}\left(\mathbf{g}^{(1)}\right) \cdots f^{(\ell)}\left(\mathbf{g}^{(\ell)}\right)\right).$$

As

$$U^{(0)}\left(\mathbf{y}^{(l)}\right) = U^{(l)}\left(\mathbf{g}^{(l)}\right),$$

we have

$$U^{(0)}\left(\boldsymbol{\varepsilon}^{(l)}\right) = \left(U^{(l)} - U^{(0)} \times f^{(l)}\right)\left(\mathbf{g}^{(l)}\right).$$

But, since

$$rank\left(\left(\boldsymbol{\varepsilon}^{(1)} \cdots \boldsymbol{\varepsilon}^{(\ell)}\right)\right) \leq t_0,$$

we also have

$$rank\left(\left(U^{(0)}\left(\boldsymbol{\varepsilon}^{(1)}\right) \cdots U^{(0)}\left(\boldsymbol{\varepsilon}^{(\ell)}\right)\right)\right) \leq t_0.$$

Consequently, by Proposition 1.49, there exists a monic skew polynomial $T \in S[X,\sigma]_{<t_0+1}$ such that for $l = 1, \ldots, \ell$,

$$T\left(U^{(0)}\left(\boldsymbol{\varepsilon}^{(l)}\right)\right) = \mathbf{0}$$

i.e.,

$$\left(T \times \left(U^{(l)} - U^{(0)} \times f^{(l)}\right)\right)\left(\mathbf{g}^{(l)}\right) = \mathbf{0}.$$

But $\{g_i^{(l)}\}_{1 \leq i \leq n^{(l)}}$ is R-linear independent and $\deg\left(T\left(U^{(l)} - U^{(0)} \times f^{(l)}\right)\right) < n^{(l)}$, thus using Corollary 1.48 we have

$$T \times \left(U^{(l)} - U^{(0)} \times f^{(l)}\right) = 0.$$

Therefore, since T is a monic polynomial, we have

$$U^{(l)} - U^{(0)} \times f^{(l)} = 0.$$

$\blacksquare$

Lemma 2.26 shows that if the rank of the error is at most the error correction capability, then every minimal solution of the key equation allows to recover the transmitted codeword. We use this property to give the unique decoding method beyond the error correction capability.

Lemma 2.27 *Assume there is $\hat{\mathbf{f}} \in S[X,\sigma]_{<k^{(1)}} \times \cdots \times S[X,\sigma]_{<k^{(\ell)}}$ such that for every minimal solution, $\mathbf{U}$, of the key equation we have $U^{(l)} = U^{(0)} f^{(l)}$, for $l = 1, \ldots, \ell$. Then, $\hat{\mathbf{f}}$ is the unique element in $S[X,\sigma]_{<k^{(1)}} \times \cdots \times S[X,\sigma]_{<k^{(\ell)}}$ such that*

$$rank\left(\mathbf{y} - \left(f^{(1)}\left(\mathbf{g}^{(1)}\right) \cdots f^{(\ell)}\left(\mathbf{g}^{(\ell)}\right)\right)\right) = t_{\min}$$

where $t_{\min}$ is defined as in (2.6).

Proof. We show first that in this condition, $t_{\min}$ is equal to the degree of a minimal solution of the key equation. Let $\mathbf{U}$ be a minimal solution of the key equation and let t be the degree of $U^{(0)}$. Then, by the definition of $t_{\min}$ and by Theorem 2.13, we have $t \leq t_{\min}$. By the assumption, we have $U^{(l)} = U^{(0)} f^{(l)}$, for $l = 1, \ldots, \ell$. Therefore, by Theorem 2.13, we also have $t_{\min} \leq t$. Thus, $t_{\min} = t$.

Now, let $\hat{\mathbf{b}} \in S[X,\sigma]_{<k^{(1)}} \times \cdots \times S[X,\sigma]_{<k^{(\ell)}}$ such that

$$rank\left(\mathbf{y} - \left(b^{(1)}\left(\mathbf{g}^{(1)}\right) \cdots b^{(\ell)}\left(\mathbf{g}^{(\ell)}\right)\right)\right) = t_{\min}.$$

Then, by Proposition 1.49, there exists a monic skew polynomial $W \in S[X,\sigma]$ of degree $t_{\min}$ such that, for $l = 1, \ldots, \ell$, $W\left(\mathbf{y}^{(l)} - b^{(l)}\left(\mathbf{g}^{(l)}\right)\right) = \mathbf{0}$. Therefore, $\left(W, Wb^{(1)}, \ldots, Wb^{(\ell)}\right)$ is a minimal solution of the key equation. Thus $b^{(l)} = f^{(l)}$, for $l = 1, \ldots, \ell$. ■

Lemma 2.27 gives a sufficient condition so that the list size of minimal list decoding is one. The following lemma gives a Gröbner basis interpretation of this condition.

Lemma 2.28 *Let $\left\{\mathbf{V}_{(r,s)}\right\}_{0 \leq r \leq \ell,\ s \in [S]^*}$ be a Gröbner basis for $M[\mathbf{y},\mathbf{g}]$ obtained by Algorithm 2 w.r.t. $\preceq_{\left(k^{(0)},\ldots,k^{(\ell)}\right)}$. Set $\alpha_{(r,s)} := \deg\left(V_{(r,s)}^{(r)}\right)$. Let $Q_{(0,1)}^{(l)}$ be the quotient and $P_{(0,1)}^{(l)}$ be the remainder of the left Euclidean division of $V_{(0,1)}^{(l)}$ by $V_{(0,1)}^{(0)}$ in $S[X,\sigma]$. The following statements are equivalent.*

(i) There is $\hat{\mathbf{f}} \in S[X,\sigma]_{<k^{(1)}} \times \cdots \times S[X,\sigma]_{<k^{(\ell)}}$ such that for every minimal solution, $\mathbf{U}$, of the key equation we have $U^{(l)} = U^{(0)} f^{(l)}$, for $l = 1, \ldots, \ell$.

(ii) The Gröbner basis $\left\{\mathbf{V}_{(r,s)}\right\}_{0 \leq r \leq \ell,\ s \in [S]^*}$ has the following properties:

1) $P_{(0,1)}^{(l)} = 0$, for $l = 1, \ldots, \ell$;

2) $\alpha_{(0,1)} - k^{(0)} < \alpha_{(r,s)} - k^{(r)}$, for all $r \in \{1, \ldots, \ell\}$ and $s \in [S]^*$;

3) $V_{(0,s)}^{(l)} = V_{(0,s)}^{(0)} Q_{(0,1)}^{(l)}$, for all $l \in \{1, \ldots, \ell\}$ and $s \in [S]^* \setminus \{1\}$.

Proof. (i)$\Longrightarrow$ (ii):
1) Since $\mathbf{V}_{(0,1)}$ is a minimal solution of the key equation, we have $V_{(0,1)}^{(l)} = V_{(0,1)}^{(0)} f^{(l)}$, for $l = 1, \ldots, \ell$. Consequently, $Q_{(0,1)}^{(l)} = f^{(l)}$ and $P_{(0,1)}^{(l)} = 0$, for $l = 1, \ldots, \ell$.
2) Suppose there are $r \in \{1, \ldots, \ell\}$ and $s \in [S]^*$ such that $\alpha_{(r,s)} - k^{(r)} \leq \alpha_{(0,1)} - k^{(0)}$. Then, $\mathbf{V}_{(0,1)} + \mathbf{V}_{(r,s)}$ is a minimal solution of the key equation. Consequently, we have $V_{(0,1)}^{(r)} + V_{(r,s)}^{(r)} = \left(V_{(0,1)}^{(0)} + V_{(r,s)}^{(0)}\right) f^{(r)}$. Since $V_{(0,1)}^{(r)} = V_{(0,1)}^{(0)} f^{(r)}$, we then have $V_{(r,s)}^{(r)} = V_{(r,s)}^{(0)} f^{(r)}$. Hence, $\deg\left(V_{(r,s)}^{(r)}\right) = \deg\left(V_{(r,s)}^{(0)} f^{(r)}\right)$, i.e., $\deg\left(V_{(r,s)}^{(r)}\right) \leq \deg\left(V_{(r,s)}^{(0)}\right) + k^{(r)} - 1$ which is absurd because w.r.t. $\preceq_{\left(k^{(0)},\ldots,k^{(\ell)}\right)}$, $ind(lm\left(\mathbf{V}_{(r,s)}\right)) = r$.
3) Let $s \in [S]^* \setminus \{1\}$. Since $\deg\left(\mathbf{V}_{(0,s)}\right)$ is minimal among the degree of all $\mathbf{U} \in M[\mathbf{y},\mathbf{g}]$

with $ind(lm(\mathbf{U}))=0$, $lc(\mathbf{U})\in[s]$, then we have $\alpha_{(0,s)}\leq\alpha_{(0,1)}$. If $\alpha_{(0,s)}<\alpha_{(0,1)}$, then $\mathbf{V}_{(0,1)}+\mathbf{V}_{(0,s)}$ is a minimal solution of the key equation and consequently we have $V^{(l)}_{(0,s)}=V^{(0)}_{(0,s)}f^{(l)}$. If $\alpha_{(0,s)}=\alpha_{(0,1)}$, then $\mathbf{V}_{(0,1)}+\mathbf{V}_{(0,s)}-lc\left(V^{(0)}_{(0,s)}\right)\mathbf{V}_{(0,1)}$ is a minimal solution of the key equation and therefore we have $V^{(l)}_{(0,s)}=V^{(0)}_{(0,s)}f^{(l)}$.

(ii)$\Longrightarrow$ (i): Let $\mathbf{U}$ be a minimal solution of the key equation. Then, by Theorem 2.24,

$$\mathbf{U}=\sum_{0\leq r\leq\ell,\ s\in[S]^*}w_{(r,s)}\mathbf{V}_{(r,s)}$$

where $w_{(r,s)}\in S[X,\sigma]$, $w_{(0,1)}=1$, for all $s\in[S]^*\setminus\{1\}$,

$$\deg\left(w_{(0,s)}\right)+\alpha_{(0,s)}<\alpha_{(0,1)}$$

and for all $(r,s)\in\{1,\ldots,\ell\}\times[S]^*$,

$$\deg\left(w_{(r,s)}\right)+\alpha_{(r,s)}-k^{(r)}\leq\alpha_{(0,1)}-k^{(0)}.$$

Let $(r,s)\in\{1,\ldots,\ell\}\times[S]^*$, then $w_{(r,s)}=0$ because $\alpha_{(0,1)}-k^{(0)}<\alpha_{(r,s)}-k^{(r)}$. Therefore $U^{(l)}=U^{(0)}Q^{(l)}_{(0,1)}$, for $l=1,\ldots,\ell$, because $V^{(l)}_{(0,s)}=V^{(0)}_{(0,s)}Q^{(l)}_{(0,1)}$, for $l=1,\ldots,\ell$ and $s\in[S]^*$. ■

The previous lemmas allow to give Algorithm 4. We have the following theorem.

Theorem 2.29 *(a) If there is $\hat{\mathbf{f}}\in S[X,\sigma]_{<k^{(1)}}\times\cdots\times S[X,\sigma]_{<k^{(\ell)}}$ such that $rank\left(\mathbf{y}-\left(f^{(1)}\left(\mathbf{g}^{(1)}\right)\cdots f^{(\ell)}\left(\mathbf{g}^{(\ell)}\right)\right)\right)\leq t_0$, then Algorithm 4 returns $\hat{\mathbf{f}}$.*

(b) If Algorithm 4 returns $\hat{\mathbf{f}}$, then it is the unique element in $S[X,\sigma]_{<k^{(1)}}\times\cdots\times S[X,\sigma]_{<k^{(\ell)}}$ such that $rank\left(\mathbf{y}-\left(f^{(1)}\left(\mathbf{g}^{(1)}\right)\cdots f^{(\ell)}\left(\mathbf{g}^{(\ell)}\right)\right)\right)=t_{\min}$.

Proof. (a) Since $\mathbf{V}_{(0,1)}$ is a minimal solution of the key equation, then, by Lemma 2.26, there is $\hat{\mathbf{f}}\in S[X,\sigma]_{<k^{(1)}}\times\cdots\times S[X,\sigma]_{<k^{(\ell)}}$ such that

$$rank\left(\mathbf{y}-\left(f^{(1)}\left(\mathbf{g}^{(1)}\right)\cdots f^{(\ell)}\left(\mathbf{g}^{(\ell)}\right)\right)\right)\leq t_0$$

if and only if $\alpha_{(0,1)}\leq t_0$ and $P^{(l)}_{(0,1)}=0$, for $l=1,\ldots,\ell$.

(b) This result is a direct consequence of Lemma 2.27 and Lemma 2.28. ■

Recall that we may have $t_{\min}\leq t_0$ or $t_0<t_{\min}$. Thus, Algorithm 4 can uniquely decode beyond the error correction capability. The following example is given as an illustration.

Example 2.30 *Let $R=\mathbb{Z}_4$, $S=R[z]/(z^4+2z^2+3z+1)$ and $a=z+(z^4+2z^2+3z+1)$. Then, S is a Galois extension of R where the Galois group is generated by a power map $\sigma:a\mapsto a^2$. Set $\mathbf{g}^{(1)}=\mathbf{g}^{(2)}=(1,a,a^2,a^3)$,*

$\mathbf{y}^{(1)}=(3a^3+2a^2+2,a^2+2a,a^3+2,2a^3+2a^2+3a+3)$,

$\mathbf{y}^{(2)}=(a^2+2a+3,2a^3+a^2+2a+3,a^3+a^2+2a+3,2a^3+3)$.

We consider the received word $\mathbf{y}=\begin{pmatrix}\mathbf{y}^{(1)} & \mathbf{y}^{(2)}\end{pmatrix}$ of the interleaved Gabidulin code $IGab_{(1,1)}\left(\mathbf{g}^{(1)},\mathbf{g}^{(2)}\right)$. Using SageMathCloud [65], Algorithm 4 returns $\left(f^{(1)},f^{(2)}\right)$ where $f^{(1)}=2a^3+3a$ and $f^{(2)}=3a^2+2a+1$. Therefore, the error vector is

$$\boldsymbol{\varepsilon}=\mathbf{y}-\begin{pmatrix}f^{(1)}\left(\mathbf{g}^{(1)}\right) & f^{(2)}\left(\mathbf{g}^{(2)}\right)\end{pmatrix}$$

and $rank(\boldsymbol{\varepsilon})=2>t_0=1$. For more details, see Appendix A.

Algorithm 4: Unique decoding

Input: a received word $\mathbf{y} = \left(\mathbf{y}^{(1)} \cdots \mathbf{y}^{(\ell)}\right) \in S^{n^{(1)}+\cdots+n^{(\ell)}}$ of the interleaved Gabidulin code $IGab_{\left(k^{(1)},\ldots,k^{(\ell)}\right)}\left(\mathbf{g}^{(1)},\ldots,\mathbf{g}^{(\ell)}\right)$.

Output: "decoding failure" or the element $\hat{\mathbf{f}}$ in $S[X,\sigma]_{<k^{(1)}} \times \cdots \times S[X,\sigma]_{<k^{(\ell)}}$ such that for every minimal solution, $\mathbf{U}$, of the key equation we have $U^{(l)} = U^{(0)} f^{(l)}$, for $l = 1, \ldots, \ell$.

$t_0 \leftarrow \left\lfloor \left(\min_{l \in \{1,\ldots,\ell\}} \left\{d^{(l)}\right\} - 1\right)/2 \right\rfloor$

Compute a Gröbner basis $\left\{\mathbf{V}_{(r,s)}\right\}_{0 \le r \le \ell,\ s \in [S]^*}$ for the module $M[\mathbf{y},\mathbf{g}]$ as in Algorithm 2 w.r.t. $\preceq_{\left(k^{(0)},\ldots,k^{(\ell)}\right)}$

$\alpha_{(r,s)} \leftarrow \deg\left(V_{(r,s)}^{(r)}\right)$

if *there is* $r \in \{1,\ldots,\ell\}$ *and* $s \in [S]^*$ *such that* $\alpha_{(r,s)} - k^{(r)} \le \alpha_{(0,1)} - k^{(0)}$ **then**

  return *"decoding failure"*

for $l \leftarrow 1$ ***to*** ℓ **do**

  Compute the quotient $Q_{(0,1)}^{(l)}$ and the remainder $P_{(0,1)}^{(l)}$ on the left Euclidean division of $V_{(0,1)}^{(l)}$ by $V_{(0,1)}^{(0)}$ in $S[X,\sigma]$.

if *there is* $l \in \{1,\ldots,\ell\}$ *such that* $P_{(0,1)}^{(l)} \neq 0$ **then**

  return *"decoding failure"*

else

  if $\alpha_{(0,1)} \le t_0$ **then**

    return $\hat{\mathbf{Q}}_{(0,1)}$

  else

    if *there is* $l \in \{1,\ldots,\ell\}$ *and* $s \in [S]^* \setminus \{1\}$ *such that* $V_{(0,s)}^{(l)} \neq V_{(0,s)}^{(0)} Q_{(0,1)}^{(l)}$ **then**

      return *"decoding failure"*

    else

      return $\hat{\mathbf{Q}}_{(0,1)}$

Remark 2.31 *In finite fields, Sidorenko et al. [68] gave an algorithm for decoding interleaved Gabidulin codes beyond the error correction capability and an upper bound of the failure probability. We implemented Algorithm 4 and compared it to [68, Algorithm 4] (see Appendix A). We observed that these two algorithms fail in the same cases. This coincidence is probably due to the fact that, in [68, Algorithm 4], Sidorenko et al. computed the error span polynomial using shift-register synthesis. We also compute the same error span polynomial using Gröbner bases. Thus, it would be interesting to see if there exists the connection between a two algorithms.*

2.4.3 Error-Erasure Decoding

As in [79], we define row and column erasures of interleaved Gabidulin codes. We then show that errors and erasures decoding of an interleaved Gabidulin code is reduced to errors decoding of another interleaved Gabidulin code.

Let $\mathbf{y} = \left(\mathbf{y}^{(1)} \dots \mathbf{y}^{(\ell)}\right) \in S^{n^{(1)}+\dots+n^{(\ell)}}$ be a received vector for a transmitted codeword $\left(f^{(1)}\left(\mathbf{g}^{(1)}\right) \cdots f^{(\ell)}\left(\mathbf{g}^{(\ell)}\right)\right)$ of the interleaved Gabidulin code $IGab_{\left(k^{(1)},\dots,k^{(\ell)}\right)}\left(\mathbf{g}^{(1)},\dots,\mathbf{g}^{(\ell)}\right)$.

Assume that the error vector

$$\boldsymbol{\varepsilon} = \left(\mathbf{y}^{(1)} \dots \mathbf{y}^{(\ell)}\right) - \left(f^{(1)}\left(\mathbf{g}^{(1)}\right) \cdots f^{(\ell)}\left(\mathbf{g}^{(\ell)}\right)\right) \tag{2.7}$$

is decomposed into

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^{(E)} + \boldsymbol{\varepsilon}^{(R)} + \boldsymbol{\varepsilon}^{(C)} \tag{2.8}$$

where

- $\boldsymbol{\varepsilon}^{(E)}$, called the full error , is unknown, $rank\left(\boldsymbol{\varepsilon}^{(E)}\right) = t^{(E)}$;
- $\boldsymbol{\varepsilon}^{(R)}$, called the **row erasure** , can be expressed in the form

$$\boldsymbol{\varepsilon}^{(R)} = \left(\mathbf{a}^{(R,1)}\mathbf{B}^{(R,1)} \cdots \mathbf{a}^{(R,\ell)}\mathbf{B}^{(R,\ell)}\right)$$

with $\mathbf{a}^{(R,l)} \in S^{t^{(R,l)}}$is known, $rank\left(\mathbf{a}^{(R,l)}\right) = t^{(R,l)}$, and $\mathbf{B}^{(R,l)} \in R^{t^{(R,l)}\times n^{(l)}}$ is unknown, for $l = 1,\dots,\ell$;

- $\boldsymbol{\varepsilon}^{(C)}$, called the **column erasure** , can be expressed in the form

$$\boldsymbol{\varepsilon}^{(C)} = \left(\mathbf{a}^{(C,1)}\mathbf{B}^{(C,1)} \cdots \mathbf{a}^{(C,\ell)}\mathbf{B}^{(C,\ell)}\right)$$

with $\mathbf{a}^{(C,l)} \in S^{t^{(C,l)}}$is unknown, $\mathbf{B}^{(C,l)} \in R^{t^{(C,l)}\times n^{(l)}}$ is known, $freerank\left(\mathbf{B}^{(C,l)}\right) = t^{(C,l)}$, for $l = 1,\dots,\ell$.

By Proposition 1.49, there are the monic skew polynomials $P^{(R,l)} \in S[X,\sigma]$ of degree $t^{(R,l)}$ such that $P^{(R,l)}\left(\mathbf{a}^{(R,l)}\right) = \mathbf{0}$, for $l = 1,\dots,\ell$.

By [20, Proposition 2.9], there are the free column matrices $\mathbf{F}^{(C,l)} \in R^{n^{(l)}\times\left(n^{(l)}-t^{(C,l)}\right)}$ such that $\mathbf{B}^{(R,l)}\mathbf{F}^{(C,l)} = \mathbf{0}$, for $l = 1,\dots,\ell$.

Theorem 2.32 *With the above notations, the relation (2.7) can be transformed into*

$$\boldsymbol{\varepsilon}' = \left(\mathbf{y}'^{(1)} \dots \mathbf{y}'^{(\ell)}\right) - \left(f'^{(1)}\left(\mathbf{g}'^{(1)}\right) \cdots f'^{(\ell)}\left(\mathbf{g}'^{(\ell)}\right)\right)$$

where $\mathbf{y}'^{(l)} = P^{(R,l)}\left(\mathbf{y}^{(l)}\right)\mathbf{F}^{(C,l)}$, $\mathbf{g}'^{(l)} = \mathbf{g}^{(l)}\mathbf{F}^{(C,l)}$, $f'^{(l)} = P^{(R,l)}f^{(l)}$, *for* $l = 1,\dots,\ell$, *and* $rank\left(\boldsymbol{\varepsilon}'\right) \le t^{(E)}$.

Proof. Set $\boldsymbol{\varepsilon}^{(E)} = \left(\boldsymbol{\varepsilon}^{(E,1)} \cdots \boldsymbol{\varepsilon}^{(E,\ell)}\right)$ where $\boldsymbol{\varepsilon}^{(E,l)} \in S^{n^{(l)}}$, for $l = 1,\dots,\ell$. Then, by (2.7) and (2.8), we have

$$\boldsymbol{\varepsilon}^{(E,l)} + \boldsymbol{\varepsilon}^{(R,l)} + \boldsymbol{\varepsilon}^{(C,l)} = \mathbf{y}^{(l)} - f^{(l)}\left(\mathbf{g}^{(l)}\right), \text{ for } l = 1,\dots,\ell.$$

Let $l \in \{1,\dots,\ell\}$. Since $\boldsymbol{\varepsilon}^{(R,l)} = \mathbf{a}^{(R,l)}\mathbf{B}^{(R,l)}$ and $P^{(R,l)}\left(\mathbf{a}^{(R,l)}\right) = \mathbf{0}$, we have

$$P^{(R,l)}\left(\boldsymbol{\varepsilon}^{(E,l)}\right) + P^{(R,l)}\left(\boldsymbol{\varepsilon}^{(C,l)}\right) = P^{(R,l)}\left(\mathbf{y}^{(l)} - f^{(l)}\left(\mathbf{g}^{(l)}\right)\right)$$

i.e.,

$$P^{(R,l)}\left(\boldsymbol{\varepsilon}^{(E,l)}\right) + P^{(R,l)}\left(\mathbf{a}^{(C,l)}\right)\mathbf{B}^{(C,l)} = P^{(R,l)}\left(\mathbf{y}^{(l)} - f^{(l)}\left(\mathbf{g}^{(l)}\right)\right) \tag{2.9}$$

because $\varepsilon^{(C,l)} = \mathbf{a}^{(C,l)}\mathbf{B}^{(C,l)}$. If we right multiply both sides of (2.9) by $\mathbf{F}^{(C,l)}$ we get

$$\varepsilon'^{(E,l)} = \mathbf{y}'^{(l)} - f'^{(l)}\left(\mathbf{g}'^{(l)}\right)$$

where $\varepsilon'^{(E,l)} = P^{(R,l)}\left(\varepsilon^{(E,l)}\right)\mathbf{F}^{(C,l)}$.

Set $\varepsilon' = \left(\varepsilon'^{(E,1)}\cdots\varepsilon'^{(E,\ell)}\right)$, then

$$\varepsilon' = \left(\mathbf{y}'^{(1)}\ldots\mathbf{y}'^{(\ell)}\right) - \left(f'^{(1)}\left(\mathbf{g}'^{(1)}\right)\cdots f'^{(\ell)}\left(\mathbf{g}^{(\ell)}\right)\right).$$

As $rank\left(\left(\varepsilon^{(E,1)}\cdots\varepsilon^{(E,\ell)}\right)\right) = t^E$, we have $rank\left(\varepsilon'^{(E,1)}\cdots\varepsilon'^{(E,\ell)}\right) \leq t^E$. ■

Set $k'^{(l)} = k^{(l)} + t^{(R,l)}$, $n'^{(l)} = n^{(l)} - t^{(C,l)}$ and assume that $k'^{(l)} \leq n'^{(l)}$, for $l = 1, \ldots, \ell$. Then, according to Theorem 2.32, the error and erasure decoding of the interleaved Gabidulin code $IGab_{\left(k^{(1)},\ldots,k^{(\ell)}\right)}\left(\mathbf{g}^{(1)}, \ldots, \mathbf{g}^{(\ell)}\right)$ is reduced to the error decoding of the interleaved Gabidulin code $IGab_{\left(k'^{(1)},\ldots,k'^{(\ell)}\right)}\left(\mathbf{g}'^{(1)}, \ldots, \mathbf{g}'^{(\ell)}\right)$. In particular we have the following:

Corollary 2.33 *With the above notations, if*

$$2t^{(E)} \leq \min_{1\leq l\leq \ell}\left\{n^{(l)} - \left(k^{(l)} + t^{(R,l)} + t^{(C,l)}\right)\right\}$$

then the transmitted massage i.e., $f^{(1)}, \ldots, f^{(\ell)}$, can be recovered.

Proof. Assume that $2t^{(E)} \leq \min_{1\leq l\leq \ell}\left\{n^{(l)} - \left(k^{(l)} + t^{(R,l)} + t^{(C,l)}\right)\right\}$.

Then

$$2t^{(E)} \leq d' - 1,$$

where d' is the rank distance of the interleaved Gabidulin code $IGab_{\left(k'^{(1)},\ldots,k'^{(\ell)}\right)}\left(\mathbf{g}'^{(1)}, \ldots, \mathbf{g}'^{(\ell)}\right)$. Hence, we can use Algorithm 4 to determine $f'^{(1)}, \ldots, f'^{(\ell)}$ and then use the left Euclidean division of $f'^{(l)}$ by $P^{(R,l)}$ to determine $f^{(l)}$ for $l = 1, \ldots, \ell$. ■

As in [26], [69], [68], [7], simultaneous correction of errors and erasures allow to recover the transmitted codeword in random linear network coding. As an illustration, see subsection 3.3.

Chapter III

Applications

As mentioned in the introduction, rank-metric codes have several applications. In this chapter, we use encoding and decoding schemes of interleaved Gabidulin codes to detect and correct errors in wireless communication systems. Specifically in space-time coding and in random linear network coding. This chapter is organized as follows.

In Section 3.1, we give the discrete baseband wireless communication system model.

In Section 3.2, we recall the performance criteria for space-time block codes, and use rank-metric codes to construct optimal space-time block codes.

In Section 3.3, we combine two existing network coding schemes and prove that the problem of decoding random linear network codes can be reformulated as an error-erasure decoding problem for rank-metric codes.

3.1 Overview of wireless communication systems

3.1.1 Basic elements of a wireless communication system

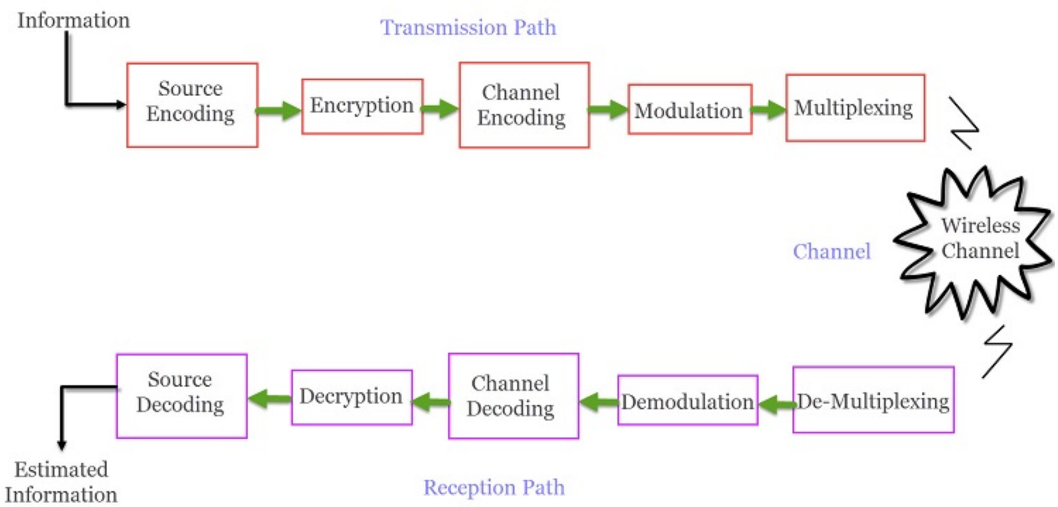

Figure 3.1: Basic elements of a wireless communication system [18]

Wireless communication involves transfer of information without any physical connection between two or more points [75]. Wireless communication system can be divided into three elements [18]: the transmitter, the channel and the receiver (See Figure 3.1).

The transmission path of a wireless communication system consists of :

- **source coding** (data compression) is the process of encoding the information using lesser number of bits than the uncoded version of the information [78];

- **encryption** is the process of encoding a message or information in such a way that only authorized parties can access it and those who are not authorized cannot [19];

- **channel coding** attempts to add redundancy to the data to make it more reliable (which reduces data rate) and therefore more robust against the channel noise [78];

- **modulation** is the process whereby message information is embedded into a radio frequency carrier [73];

- **multiplexing** is a technique by which multiple analog signals or digital data streams are combined into a single signal to be transmitted over a shared medium [50].

The channel carries the signal, but will usually distort it. The receive path reconstructs the source signal using the inverse operations of the transmission path. In the next subsections, we will show how information is modulated and transmitted.

In the following, most of the definitions and results are from [59], [76], [73], [77].

3.1.2 Digital modulation

A real-valued emitted signal $s(t)$, with a frequency content concentrated in a narrow band of frequencies near the carrier frequency f_c (**bandpass signal**), can be written as

$$s(t) = a\left(t\right)\cos\left(2\pi f_c t + \theta\left(t\right)\right)$$

where $a\left(t\right)$ and $\theta\left(t\right)$ represent respectively the envelope and phase of $s(t)$. In complex notation, $s(t)$ can be written as

$$\begin{aligned} s(t) &= a\left(t\right)\cos\left(2\pi f_c t + \theta\left(t\right)\right) \\ &= Re\left(a\left(t\right)e^{i(2\pi f_c t + \theta(t))}\right) \\ &= Re\left(\tilde{s}(t)e^{i2\pi f_c t}\right), \end{aligned}$$

where

$$\tilde{s}(t) = a\left(t\right)e^{i\theta(t)}$$

and $Re\left(\cdot\right)$ denotes the real part operation. The signal $\tilde{s}(t)$ is called the **complex envelope** or **complex baseband representation** of the bandpass signal $s(t)$.

Digital modulation is the process of mapping a digital sequence to signals for transmission over a communication channel. In **linear modulation** , the baseband complex envelope can be written as

$$\tilde{s}(t) = \sum_n a_n p\left(t - nT_s\right),$$

where a_n are the transmitted symbols, $p\left(t\right)$ is the pulse shape and T_s represents the duration symbol. The complex symbols a_n take its values into a set of M complex

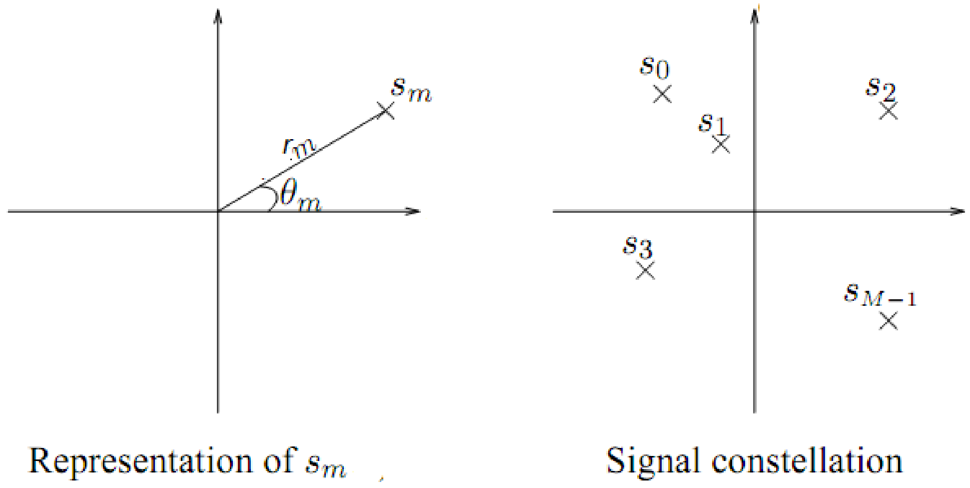

Figure 3.2: The complex plane representation of the signal constellation [77].

numbers $\{s_1, s_1, \ldots, s_M\}$ called **constellation** representing a particular modulation. In polar coordinates, we have $s_m = r_m e^{i\theta_m}$, $1 \le m \le M$ (See Figure 3.2).

Some commonly used signal constellations are:

- Pulse Amplitude Modulation (PAM). Information only in amplitude:

$$\theta_m = 0 \text{ and } r_m = (2m - 1 - M)\,\tfrac{d}{2}, \quad m = 0, \ldots, M-1$$

- Phase Modulation or Phase Shift Keying (PSK). Information only in phase:

$$\theta_m = \tfrac{2\pi m}{M} \text{ and } r_m = r, \quad m = 0, \ldots, M-1$$

- Quadrature Amplitude Modulation (QAM). Information in phase and amplitude.

In [22], the η^2-ary square quadrature amplitude modulation is algebraically represented by the ring $\mathbb{Z}_\eta[i] = \mathbb{Z}_\eta + i\mathbb{Z}_\eta$, where $i^2 = -1$ and $\mathbb{Z}_\eta$ is the ring of integers modulo η. For example, the Quadrature Phase-Shift Keying (QPSK) is algebraically represented by the ring $\mathbb{Z}_2[i] = \{0, 1, i, 1+i\}$ (See Figure 3.3).

2-Ary digits	QPSK	Complex representation
11	$\sqrt{2}\cos\left(2\pi f_c t + \frac{\pi}{4}\right)$	$\sqrt{2}e^{\frac{\pi}{4}i} = 1 + i$
10	$\sqrt{2}\cos\left(2\pi f_c t - \frac{\pi}{4}\right)$	$\sqrt{2}e^{-\frac{\pi}{4}i} = 1 - i$
01	$\sqrt{2}\cos\left(2\pi f_c t + \frac{3\pi}{4}\right)$	$\sqrt{2}e^{\frac{3\pi}{4}i} = -1 + i$
00	$\sqrt{2}\cos\left(2\pi f_c t - \frac{3\pi}{4}\right)$	$\sqrt{2}e^{\frac{3\pi}{4}i} = -1 - i$

3.1.3 Discrete time baseband representation of multipart propagation

When the signal is modulated, it is transmitted over a wireless channel. Due to refraction, reflection and diffraction in a wireless communication environment, the propagation of the

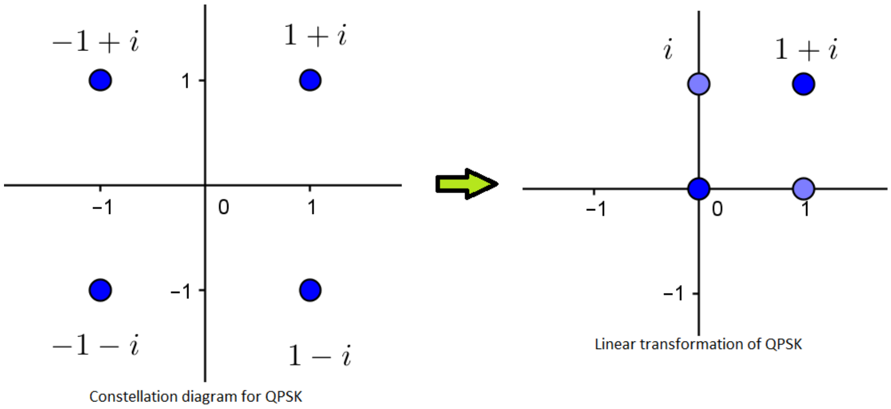

Figure 3.3: The ring representation of QPSK: $\mathbb{Z}_2[i] = \{0, 1, i, 1 + i\}$.

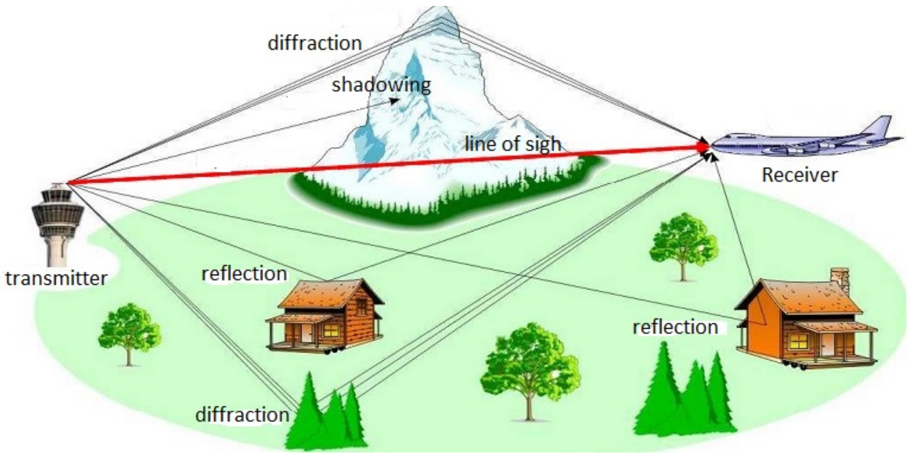

Figure 3.4: multipath propagation [32].

signal transmitted by the source reaches the receiver side by different paths (See Figure 3.4). This multipath propagation causes constructive and destructive interference, and phase shifting of the signal. Thus, each n-th path received signal is associated with a corresponding attenuation factor $\alpha_n(t)$ and the propagation delay $\tau_n(t)$. Therefore, if $s(t)$ is the bandpass transmitted signal then, using the principle of superposition, the bandpass received signal may be expressed in the form

$$r(t) = \sum_n \alpha_n(t)\, s(t - \tau_n(t)) + w(t)$$

where $w(t)$ is the additive noise. According to the central limit theorem, we may assume that $w(t)$ is a white Gaussian noise process.

A channel is said to be **frequency-nonselective channel**, or **flat fading** if the bandwidth of the transmitted signal is much smaller than the coherence bandwidth of the channel. In this case, the baseband received signal $\tilde{r}(t)$ can be expressed in the form

$$\tilde{r}(t) = C(t)\, \tilde{s}(t) + \tilde{w}(t) \tag{3.1}$$

where $C(t)$ is the **complex channel gain**. Due to the multipath propagation, we may assume that $C(t)$ is modeled as a zero-mean complex-valued Gaussian random process (Rayleigh channel model).

If the time variations of the complex channel gain are very slow within a time interval $0 \leq t \leq T$, when T is the symbol interval, then Equation (3.1) may be simply expressed as

$$\tilde{r}(t) = C\tilde{s}(t) + \tilde{w}(t), \quad 0 \leq t \leq T \tag{3.2}$$

where C is constant within the time interval $0 \leq t \leq T$. In this case, we call the channel a **slowly fading channel**. Next, consider time to be discrete, where t_k denotes the time at which the k-th symbol $\tilde{x}_k := \tilde{x}(t_k)$ is transmitted. In a discrete time baseband, (3.2) become

$$\tilde{r}_k = C\tilde{s}_k + \tilde{w}_k,$$

where $\tilde{r}_k := \tilde{r}(t_k)$ and $\tilde{w}_k = \tilde{w}(t_k)$.

3.1.4 Multiple-input, multiple-output channel

To reduce multipath fading and increase system capacity, we can use multiple-input and multiple-output (MIMO) antenna systems (See Figures 3.5 and 3.6).

By [35], Mobile operators have implemented 2×2 MIMO in their LTE 4G networks for a number of years and are now beginning to deploy 4×4 MIMO to meet increased data demands.

We will denote the number of transmit and receive antennas in the complex domain by m_t and m_r, respectively. We consider a discrete-time complex baseband model of a flat-fading MIMO channel with additive white Gaussian noise. A block-fading channel is assumed, i.e., the channel matrix is constant over the whole block of n_c data symbols.

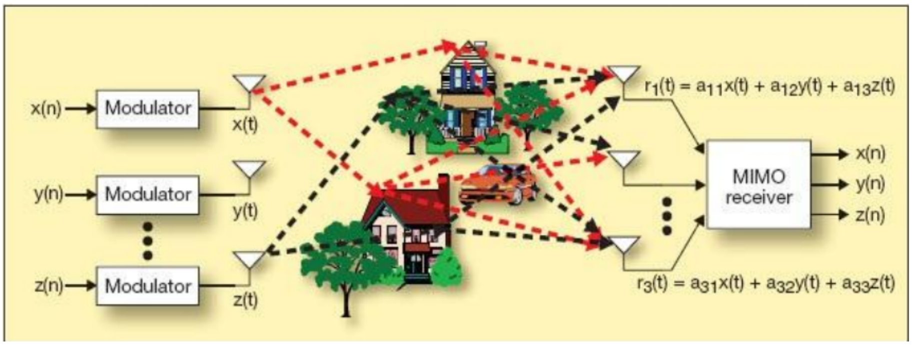

Figure 3.5: MIMO channel [36].

Figure 3.6: 4x4 MIMO [35].

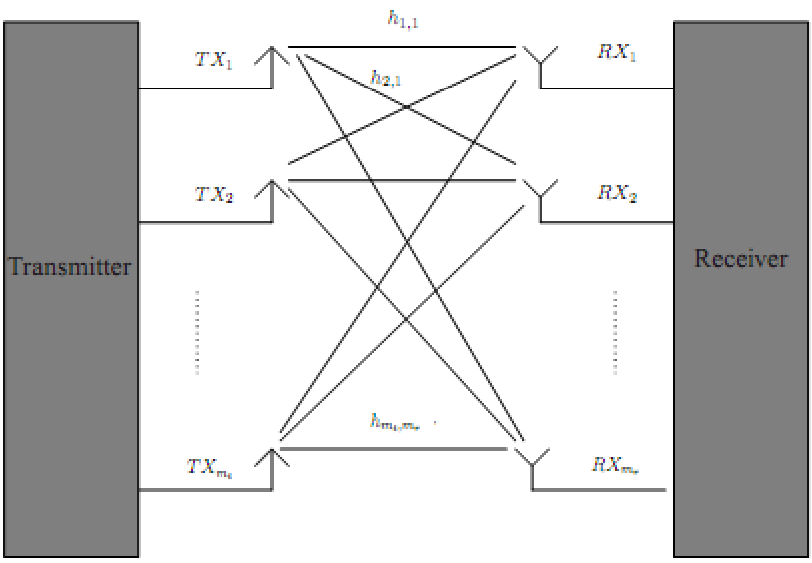

Figure 3.7: MIMO model with m_t transmit antennas and m_r receive antennas [6].

The complex channel gain between the l-th transmit antenna and the i-th receive antenna is denoted $h_{i,l}$ (See Figure 3.7).

Let $x_{l,j}$ be the j-th data symbol transmitted from the l-th transmit antenna. Then the j-th data symbol received at the i-th antenna can be expressed as:

$$y_{i,j} = \sum_{1 \leq l \leq n_c} h_{i,l} x_{l,j} + n_{i,j} \tag{3.3}$$

where $n_{i,j}$ is a noise term. In matrix representation, (3.3) become

$$\mathbf{Y} = \mathbf{HX} + \mathbf{N}$$

where $\mathbf{Y} = (y_{i,j})$, $\mathbf{H} = (h_{i,l})$, $\mathbf{X} = (x_{l,j})$ and $\mathbf{N} = (n_{i,j})$.

In the next section, we show how to detect and correct errors in the MIMO channel.

3.2 Space-time block codes

3.2.1 Performance criteria for space-time block codes

A **space-time block code** $\mathcal{C}_{ST}$ is a set of codeword matrices $\mathbf{X}$ over $\mathbb{C}$ of size $m_t \times n_c$. The entries of each of the codeword matrices are drawn from a transmission symbol alphabet set (or signal constellation) $\mathcal{A}$. Let E_s be the average energy of the signal constellation. The constellation points are scaled by a factor of $\sqrt{E_s}$ such that the average energy of the constellation points is 1. We assume that received matrix $\mathbf{Y} \in \mathbb{C}^{m_r \times n_c}$ is decomposed into

$$\mathbf{Y} = \sqrt{E_s}\mathbf{HX} + \mathbf{N}$$

where:

· $\mathbf{X} \in \mathcal{C}_{ST}$ is the sent codeword.

· $\mathbf{H} \in \mathbb{C}^{l \times n}$ is the channel matrix, which is known at the receiver (perfect channel state information), and whose entries are independent and identically distributed (i.i.d.), complex circularly symmetric Gaussian random variables with zero mean and unit variance.

· $\mathbf{N} \in \mathbb{C}^{l \times m}$ represents the additive white noise, which is unknown at the receiver, and whose entries are i.i.d, complex circularly symmetric Gaussian random variables with zero mean and variance N_0.

When $\mathbf{Y}$ is received, **maximum likelihood decoder** consists to find $\widehat{\mathbf{X}} \in \mathcal{C}_{ST}$ such that

$$\left\| \mathbf{Y} - \sqrt{E_s}\mathbf{H}\widehat{\mathbf{X}} \right\|_F = \min_{\mathbf{X} \in \mathcal{C}_{ST}} \left\| \mathbf{Y} - \sqrt{E_s}\mathbf{HX} \right\|_F$$

where $\|\cdot\|_F$ is the Frobenius norm. Maximum likelihood decoding fails if $\mathbf{X}$ is transmitted and $\mathbf{X} \neq \widehat{\mathbf{X}}$. Thus, the pairwise error probability that $\widehat{\mathbf{X}}$ is selected when $\mathbf{X}$ is transmitted, for any given channel matrix realization $\mathbf{H}$, is

$$P\left(\mathbf{X} \to \widehat{\mathbf{X}} \mid \mathbf{H}\right) := P\left(\left\| \mathbf{Y} - \sqrt{E_s}\mathbf{H}\widehat{\mathbf{X}} \right\|_F \leq \left\| \mathbf{Y} - \sqrt{E_s}\mathbf{HX} \right\|_F\right)$$

The following theorem give the upper-bound on the pairwise error probability.

Theorem 3.1 *[74]*
We have

$$P\left(\mathbf{X} \to \widehat{\mathbf{X}} \mid \mathbf{H}\right) \leq \left(\prod_{i=1}^{r} \lambda_i\right)^{-m_r} (E_s/4N_0)^{-m_r \times r}$$

where
· $r = rank\left(\mathbf{X} - \widehat{\mathbf{X}}\right)$
· $\prod_{i=1}^{r} \lambda_i$ *is a product of nonzero eigenvalues of* $\left(\mathbf{X} - \widehat{\mathbf{X}}\right)\left(\mathbf{X} - \widehat{\mathbf{X}}\right)^H$, *with* $(\cdot)^H$ *is the Hermitian transpose operation.*

To minimize the maximum pairwise error probability, the following two criteria were derived [74]:

Rank criterion: the minimum rank r of $\mathbf{X} - \widehat{\mathbf{X}}$ taken over all distinct codeword pairs is the **transmit diversity gain** and should be maximized.

Determinant criterion: the minimum of $\prod_{i=1}^{r} \lambda_i$ taken over all distinct codeword pairs is the **coding gain** and must be maximized.

For any space-time block code there is a tradeoff between the transmission rate and the transmit diversity gain [74], [47]. Specifically, using the same arguments as in the proof of Proposition 2.1, we can show the following proposition.

Proposition 3.2 *(Rate-Diversity Tradeoff) For any space-time code $\mathcal{C}_{ST}$,*

$$R_{\mathcal{C}_{ST}} \leq m_t - d_{\mathcal{C}_{ST}} + 1$$

*where $R_{\mathcal{C}_{ST}}$ is the **rate** of $\mathcal{C}_{ST}$,*

$$R_{\mathcal{C}_{ST}} := \frac{1}{n_c} \log_{|\mathcal{A}|} |\mathcal{C}_{ST}|$$

*and $d_{\mathcal{C}_{ST}}$ is the **transmit diversity gain** of $\mathcal{C}_{ST}$,*

$$d_{\mathcal{C}_{ST}} := \min\left\{rank\left(\mathbf{X} - \mathbf{X}'\right) : \ \mathbf{X},\ \mathbf{X}' \in \mathcal{C}_{ST},\ \mathbf{X} \neq \mathbf{X}'\right\}$$

As in [37], a space-time block code that achieves this rate-diversity tradeoff will be called an **optimal space-time block code**.

3.2.2 Space-time block codes from codes over finite principal ideal rings

In this subsection, we generalize to finite principal ideal rings the methods of [48], [44], [37], [61] in the construction of space-time block codes. More precisely, we show that there is a rank-preserving map from a finite principal ideal ring to a complex signal set and we use it to construct space-time block codes that are optimal under the rate-diversity tradeoff [74], [47], [37].

Let T be a principal ideal ring such that there exists a surjective ring homomorphism $\varphi : T \to R$. Let φ^* be a section of φ, i.e., a map from R to T such that $\varphi \circ \varphi^* = id_R$. The extension of φ (resp., φ^*) coefficient-by-coefficient to the set of matrix $T^{m\times n}$ (resp., $R^{m\times n}$) is also denoted by φ (resp., φ^*). As an example, we may have $T = \mathbb{Z}[i]$, $R = \mathbb{Z}[i]/\eta\mathbb{Z}[i]$, where η is some positive integer, $\varphi(x) = x + \eta\mathbb{Z}[i]$ and $\varphi^*(a + bi + \eta\mathbb{Z}[i]) = (a \bmod \eta) + (b \bmod \eta)\, i$, for all $x \in \mathbb{Z}[i]$, $a \in \mathbb{Z}$, $b \in \mathbb{Z}$.

Lemma 3.3 *Let $\mathbf{A} \in T^{m\times n}$. Then,*

$$rank_R(\varphi(\mathbf{A})) \leq rank_T(\mathbf{A}).$$

Proof. Let $r = rank_T(\mathbf{A})$ and $\{\mathbf{b}_1, \ldots, \mathbf{b}_r\}$ be a generating set of $col(\mathbf{A})$. Then, $\{\varphi(\mathbf{b}_1), \ldots, \varphi(\mathbf{b}_r)\}$ is a generating set of $col(\varphi(\mathbf{A}))$. Consequently, $rank_R(\varphi(\mathbf{A})) \leq rank_T(\mathbf{A})$. ■

Theorem 3.4 *Let $\mathcal{M} \subset R^{m\times n}$ be a rank code of rank distance d and let d' be the rank distance of $\varphi^*(\mathcal{M})$, then $d \leq d'$. Moreover, if $\mathcal{M}$ is an MRD code, then $d = d'$.*

Proof. Let $\varphi^*(\mathbf{M}_1)$, $\varphi^*(\mathbf{M}_2) \in \varphi^*(\mathcal{M})$ such that $\varphi^*(\mathbf{M}_1) \neq \varphi^*(\mathbf{M}_2)$. Then, $\mathbf{M}_1 \neq \mathbf{M}_2$ and by Lemma 3.3,

$$\begin{aligned} rank_T(\varphi^*(\mathbf{M}_1) - \varphi^*(\mathbf{M}_2)) &\geq rank_R(\varphi(\varphi^*(\mathbf{M}_1) - \varphi^*(\mathbf{M}_2))) \\ &\geq d. \end{aligned}$$

Thus, $d \leq d'$.

Assume that $\mathcal{M}$ is an MRD code. Then,

$$|\varphi^*(\mathcal{M})| = |\mathcal{M}| = |R|^{\min\{m(n-d+1),\ n(m-d+1)\}} \tag{3.4}$$

Using the same arguments as in the proof of Proposition 2.1, we can show that

$$|\varphi^*(\mathcal{M})| \leq |\varphi^*(R)|^{\min\{m(n-d'+1),\ n(m-d'+1)\}} \tag{3.5}$$

It follows from (3.4) and (3.5) that $d' \leq d$. ■

By the previous theorem, we can use an MRD code in R to construct an MRD code in T. The following example is a generalization of [48], [2].

Example 3.5 *Since $S \cong R[X]/(h)$ where h is a monic polynomial, set $h = a_0 + a_1X + \cdots + a_{m-1}X^{m-1} + X^m$, $\alpha = X + (h)$ and $\mathbf{g} = (\alpha, \alpha^2, \ldots, \alpha^m)$. Then, the Gabidulin code $Gab_1(\mathbf{g})$ is a free S-linear rank code generated by $\mathbf{g}$. Thus, $Gab_1(\mathbf{g})$ is a free R-linear rank code generated by $\{\mathbf{g}, \alpha\mathbf{g}, \ldots, \alpha^{m-1}\mathbf{g}\}$. The matrix representation of $\mathbf{g}$ in the basis $(1, \alpha, \ldots, \alpha^{m-1})$ is*

$$\mathbf{A}_{\mathbf{g}} = \begin{pmatrix} 0 & 0 & \cdots & 0 & -a_0 \\ 1 & 0 & \cdots & 0 & -a_1 \\ 0 & 1 & \cdots & 0 & -a_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & -a_{m-1} \end{pmatrix}$$

and the matrix representation of $\alpha^i \mathbf{g}$ is $\mathbf{A}_{\mathbf{g}}^{i+1}$ for $i = 1, \ldots, m-1$. Therefore, the matrix representation of $Gab_1(\mathbf{g})$ is a R-linear rank code generated by $\left\{\mathbf{A}_{\mathbf{g}}^i\right\}_{1 \leq i \leq m}$. Its image in T is an MRD code of rank distance m. Moreover, all codeword have the full rank. By Proposition 2.10, the interleaved Gabidulin code $IGab_{\left(k^{(1)}, \ldots, k^{(\ell)}\right)}\left(\mathbf{g}^{(1)}, \ldots, \mathbf{g}^{(\ell)}\right)$ with $k^{(l)} = 1$ and $\mathbf{g}^{(l)} = (\alpha, \alpha^2, \ldots, \alpha^m)$, for $l = 1, \ldots, \ell$, have the same proprieties. Thus, we can use it to construct optimal space-time block code in T.

The construction of space-time codes using rank metric codes allows to achieve the rate-diversity tradeoff. Another advantage lies in the decoding algorithm. In MIMO channel, additive white Gaussian noise suggests the decoding of space-time codes using maximum likelihood decoding. But, the complexity of maximum likelihood decoding increases exponentially as the code length increases. To reduce the complexity, in [61], Puchinger et al. combined lattice-reduction-aided equalization techniques and error-erasure decoding algorithm of Gabidulin codes to decode space-time codes. Recall that in our construction of space-time codes, we used the linear labeling method introduced in [22]. The linear labeling allows to decode space-time codes using a new linear receiver architecture called integer-forcing linear receiver, recently proposed in [81] (see, for example [66]). The advantages of the integer-forcing linear receiver compared to lattice-reduction-aided equalization techniques have been given, for example, in [81] and [66]. Thus, it would be interesting to study the decoding of space-time codes using the combination of the integer-forcing linear receiver and the decoding algorithms of interleaved Gabidulin codes.

3.3 Decoding of random linear network codes over finite principal ideal rings

In this section, we consider random linear network coding over finite principal ideal rings. To improve the error correction, we combine the encoding schemes of [69] and [70], that is, we consider that the transmitted matrix is represented by the matrix $\mathbf{X} = \begin{pmatrix} \mathbf{0}_{m \times \beta_0} & \mathbf{I}_m & \mathbf{M} \end{pmatrix}$ where $\mathbf{M}$ is a code matrix of some matrix code $\mathcal{M} \subset R^{m \times n}$. The channel equation is given by

$$\mathbf{Y} = \mathbf{A}\mathbf{X} + \mathbf{E} \tag{3.6}$$

where the transfer matrix $\mathbf{A} \in R^{m_r \times m}$ and $rank(\mathbf{E}) := \beta$. Recall that the random matrices $\mathbf{A}$ and $\mathbf{E}$ are unknown to the destination and the problem is to recover the transmitted matrix $\mathbf{X}$ from the received matrix $\mathbf{Y}$. As in [69] and [26], we will show that this problem can be reformulated as an error-erasure decoding problem for rank-metric codes.

When the matrix $\mathbf{Y}$ is received, the Smith normal form is used to successively transform the decoding problem into error-erasure decoding. In the following, we give these transformations.

3.3.1 First transformation

Set

$$\mathbf{Y} = \begin{pmatrix} \mathbf{Y}_0 & \mathbf{Y}_1 & \mathbf{Y}_2 \end{pmatrix},$$

where $\mathbf{Y}_0$, $\mathbf{Y}_1$ and $\mathbf{Y}_2$ are submatrices of $\mathbf{Y}$ of sizes $m_r \times \beta_0$, $m_r \times m$ and $m_r \times n$, respectively. Set $freerank(\mathbf{Y}_0) := \alpha_{0f}$. Then, using the Smith normal form, there exist the invertible matrices $\mathbf{P}$, $\mathbf{Q}$ and the diagonal matrix $\mathbf{D}_2$ such that

$$\mathbf{PY_0Q} = \begin{pmatrix} \mathbf{I}_{\alpha_{0f}} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_2 \end{pmatrix}.$$

Set

$$\widetilde{\mathbf{Q}} = \begin{pmatrix} \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{m+n} \end{pmatrix}.$$

and

$$\mathbf{P} = \begin{pmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \end{pmatrix}$$

where $\mathbf{P}_1$ and $\mathbf{P}_2$ are the submatrices of $\mathbf{P}$ of sizes $\alpha_{0f} \times m_r$, and $(m_r - \alpha_{0f}) \times m_r$, respectively. If we multiply both sides of (3.6) by $\mathbf{P}$ and $\widetilde{\mathbf{Q}}$ we get the following:

Lemma 3.6 *With the above notations,*

$$\mathbf{Y}' = \mathbf{A}' \begin{pmatrix} \mathbf{I}_m & \mathbf{M} \end{pmatrix} + \mathbf{E}' \tag{3.7}$$

where $\mathbf{Y}' = \mathbf{P}_2 \begin{pmatrix} \mathbf{Y}_1 & \mathbf{Y}_2 \end{pmatrix}$, $\mathbf{A}' = \mathbf{P}_2\mathbf{A}$ *and* $\mathbf{E}'$ *is a matrix with* $rank(\mathbf{E}') := \beta' \leq \beta - \alpha_{0f}$.

Proof. Set

$$\mathbf{E} = \begin{pmatrix} \mathbf{E}_0 & \mathbf{E}_1 & \mathbf{E}_2 \end{pmatrix},$$

where $\mathbf{E}_0$, $\mathbf{E}_1$ and $\mathbf{E}_2$ are submatrices of $\mathbf{E}$ of sizes $m_r \times \beta_0$, $m_r \times m$ and $m_r \times n$, respectively.

If we multiply both sides of (3.6) by $\mathbf{P}$ and $\widetilde{\mathbf{Q}}$ we get

$$\begin{pmatrix} \mathbf{I}_{\alpha_{0f}} & \mathbf{0} & \mathbf{P}_1\mathbf{Y}_1 & \mathbf{P}_1\mathbf{Y}_2 \\ \mathbf{0} & \mathbf{D}_2 & \mathbf{P}_2\mathbf{Y}_1 & \mathbf{P}_2\mathbf{Y}_2 \end{pmatrix} = \mathbf{PA} \begin{pmatrix} \mathbf{0}_{m \times \beta_0} & \mathbf{I}_m & \mathbf{M} \end{pmatrix} + \widetilde{\mathbf{E}}$$

where

$$\widetilde{\mathbf{E}} = \mathbf{PE}\widetilde{\mathbf{Q}}.$$

Consequently,

$$\widetilde{\mathbf{E}} = \begin{pmatrix} \mathbf{I}_{\alpha_{0f}} & \mathbf{0} & \mathbf{P}_1\mathbf{E}_1 & \mathbf{P}_1\mathbf{E}_2 \\ \mathbf{0} & \mathbf{D}_2 & \mathbf{P}_2\mathbf{E}_1 & \mathbf{P}_2\mathbf{E}_2 \end{pmatrix}.$$

Set $\mathbf{E}' = \begin{pmatrix} \mathbf{P}_2\mathbf{E}_1 & \mathbf{P}_2\mathbf{E}_2 \end{pmatrix}$ and $rank(\mathbf{E}') := \beta'$, then $\beta' \leq rank\left(\widetilde{\mathbf{E}}\right) - \alpha_{0f}$ and

$$\begin{pmatrix} \mathbf{Y}_1' & \mathbf{Y}_2' \end{pmatrix} = \mathbf{A}' \begin{pmatrix} \mathbf{I}_m & \mathbf{M} \end{pmatrix} + \mathbf{E}'$$

■

3.3.2 Second transformation

Set $m'_r := m_r - \alpha_{0f}$ and

$$\mathbf{Y}' := \begin{pmatrix} \mathbf{Y}'_1 & \mathbf{Y}'_2 \end{pmatrix}.$$

where $\mathbf{Y}'_1$ and $\mathbf{Y}'_2$ are submatrices of $\mathbf{Y}'$ of sizes $m'_r \times m$ and $m'_r \times n$, respectively.

Set $rank\left(\mathbf{Y}'_1\right) := \alpha_1$, $freerank\left(\mathbf{Y}'_1\right) := \alpha_{1f}$. Using the Smith normal form, there exist the invertible matrices $\mathbf{P}'$, $\mathbf{Q}'$ and the diagonal matrix $\mathbf{D}' = diag\left(d_1, \ldots, d_r\right)$, with $d_1 = \cdots = d_{\alpha_{1f}} = 1$, such that

$$\mathbf{P}'\mathbf{Y}'_1\mathbf{Q}' = \mathbf{D}'.$$

Using Proposition 1.28, if we decompose $\mathbf{E}'$ as in [26, Eq. (29)] then we get the following:

Lemma 3.7 *With the above notations,*

$$\mathbf{Y}''_2 = \mathbf{D}'\mathbf{M}' + \mathbf{E}''. \tag{3.8}$$

where $\mathbf{Y}''_2 = \mathbf{P}'\mathbf{Y}'_2$, $\mathbf{M}' = \mathbf{Q}'^{-1}\mathbf{M}$ and $\mathbf{E}''$ is a matrix with $rank\left(\mathbf{E}''\right) \leq \beta'$.

Proof. As $rank\left(\mathbf{E}'\right) = \beta'$, by Proposition 1.28,

$$\mathbf{E}' = \mathbf{B}'\mathbf{Z}',$$

where $\mathbf{B}'$ is a $m'_r \times \beta'$ matrix, $rank\left(\mathbf{B}'\right) = \beta'$, and $\mathbf{Z}'$ is a $\beta' \times (m+n)$ matrix.

Set $\mathbf{Z}' = \begin{pmatrix} \mathbf{Z}'_1 & \mathbf{Z}'_2 \end{pmatrix}$ where $\mathbf{Z}'_1$ and $\mathbf{Z}'_2$ are submatrices of $\mathbf{Z}'$ of sizes $\beta' \times m$ and $\beta' \times n$, respectively. By (3.7) we have

$$\mathbf{Y}'_1 = \mathbf{A}' + \mathbf{B}'\mathbf{Z}'_1$$

and

$$\mathbf{Y}'_2 = \mathbf{A}'\mathbf{M} + \mathbf{B}'\mathbf{Z}'_2.$$

Consequently,

$$\mathbf{Y}'_2 = \mathbf{Y}'_1\mathbf{M} + \mathbf{B}'\left(\mathbf{Z}'_2 - \mathbf{Z}'_1\mathbf{M}\right).$$

If we multiply the above equation by $\mathbf{P}'$, then we have

$$\mathbf{Y}''_2 = \mathbf{D}'\mathbf{M}' + \mathbf{E}'',$$

where $\mathbf{E}'' = \mathbf{P}'\mathbf{B}'\left(\mathbf{Z}'_2 - \mathbf{Z}'_1\mathbf{M}'\right)$ and $rank\left(\mathbf{E}''\right) \leq rank\left(\mathbf{B}'\right) = \beta'$. ∎

3.3.3 Third transformation

Set

$$\mathbf{D}' = \begin{pmatrix} \mathbf{D}'_1 \\ \mathbf{0} \end{pmatrix}$$

and

$$\mathbf{Y}''_2 = \begin{pmatrix} \mathbf{Y}''_{21} \\ \mathbf{Y}''_{22} \end{pmatrix}$$

where $\mathbf{D}'_1$ is the submatrix of $\mathbf{D}'$ of sizes $\alpha_1 \times m$, $\mathbf{Y}''_{21}$ and $\mathbf{Y}''_{22}$ are submatrices of $\mathbf{Y}''_2$ of sizes $\alpha_1 \times n$ and $(m'_r - \alpha_1) \times n$, respectively.

Let $\alpha_{22f} := freerank\,(\mathbf{Y}''_{22})$. If $\alpha_{22f} \neq 0$ then, using the Smith normal form, there is a $\alpha_{22f} \times (m'_r - \alpha_1)$ matrix $\mathbf{U}$, such that the free rank of the matrix $\mathbf{Y}'''_{22} := \mathbf{U}\mathbf{Y}''_{22}$ is α_{22f}.

Let $\widehat{\mathbf{Y}}_{22}$ be the matrix defined by $\widehat{\mathbf{Y}}_{22} := \mathbf{Y}'''_{22}$ if $\alpha_{22f} \neq 0$ and $\widehat{\mathbf{Y}}_{22}$ is a $1 \times n$ zero matrix else.

Let $\mathbf{D}''_1$ be the $m \times m$ matrix and $\mathbf{Y}'''_{21}$ be the $m \times n$ matrix obtained respectively from the matrices $\mathbf{D}'_1$ and $\mathbf{Y}''_{21}$ by inserting all-zero rows below the last row if $\alpha_1 \leq m$ and by deleting the $\alpha_1 - m$ last rows else.

Set $\widehat{\mathbf{D}}_1 := \mathbf{Q}'\,(\mathbf{D}''_1 - \mathbf{I}_m)$ and $\widehat{\mathbf{Y}}_{21} := \mathbf{Q}'\mathbf{Y}'''_{21}$. Note that, $\widehat{\mathbf{D}}_1 = \mathbf{0}$ if $\alpha_{1f} \geq m$ and $rank\left(\widehat{\mathbf{D}}_1\right) \leq m - \alpha_{1f}$ else. We have the following:

Theorem 3.8 *With the above notations, the matrix $\widehat{\mathbf{Y}}_{21}$ can be decomposed into*

$$\widehat{\mathbf{Y}}_{21} = \mathbf{M} + \widehat{\mathbf{D}}_1\mathbf{W}_1 + \mathbf{W}_2\widehat{\mathbf{Y}}_{22} + \widehat{\mathbf{E}},$$

where $\mathbf{M}$ is the transmitted codeword, the matrices $\mathbf{W}_1$, $\mathbf{W}_2$ and $\widehat{\mathbf{E}}$ are unknown, $rank\left(\widehat{\mathbf{E}}\right) \leq \beta - \alpha_{0f} - \alpha_{22f}$.

Proof. Set

$$\mathbf{E}'' = \begin{pmatrix} \mathbf{E}''_1 \\ \mathbf{E}''_2 \end{pmatrix},$$

where $\mathbf{E}''_1$ and $\mathbf{E}''_2$ are submatrices of $\mathbf{E}''$ of sizes $\alpha_1 \times n$ and $(m'_r - \alpha_1) \times n$, respectively. By (3.8), we have

$$\begin{pmatrix} \mathbf{Y}''_{21} \\ \mathbf{Y}''_{22} \end{pmatrix} = \begin{pmatrix} \mathbf{D}'_1 \\ \mathbf{0} \end{pmatrix} \mathbf{M}' + \begin{pmatrix} \mathbf{E}''_1 \\ \mathbf{E}''_2 \end{pmatrix}.$$

Thus,

$$\mathbf{Y}''_{21} = \mathbf{D}'_1\mathbf{M}' + \mathbf{E}''_1 \tag{3.9}$$

and

$$\mathbf{Y}''_{22} = \mathbf{E}''_2.$$

- Assume that $freerank\,(\mathbf{Y}''_{22}) \neq 0$. As $\mathbf{Y}'''_{22} = \mathbf{U}\mathbf{Y}''_{22}$, set $\mathbf{E}''' := \begin{pmatrix} \mathbf{I}_{\alpha_1} & \mathbf{0} \\ \mathbf{0} & \mathbf{U} \end{pmatrix} \mathbf{E}''$.

Then,
$rank\,(\mathbf{E}''') \leq rank\,(\mathbf{E}'') \leq \beta'$ and $\mathbf{E}''' = \begin{pmatrix} \mathbf{E}''_1 \\ \mathbf{Y}'''_{22} \end{pmatrix}$. Since $freerank\,(\mathbf{Y}'''_{22}) = \alpha_{22f}$, by [20, Proposition 2.11], there are $(n - \alpha_{22f}) \times n$ matrix $\mathbf{Y}_3$, $n\times$ $(n - \alpha_{22f})$ matrix $\mathbf{F}_1$ and $n \times \alpha_{22f}$ matrix $\mathbf{F}_2$ such that

$$\begin{pmatrix} \mathbf{Y}_3 \\ \mathbf{Y}'''_{22} \end{pmatrix} \begin{pmatrix} \mathbf{F}_1 & \mathbf{F}_2 \end{pmatrix} = \mathbf{I}_n.$$

As

$$\begin{aligned}\mathbf{I}_n &= \begin{pmatrix} \mathbf{F}_1 & \mathbf{F}_2 \end{pmatrix} \begin{pmatrix} \mathbf{Y}_3 \\ \mathbf{Y}'''_{22} \end{pmatrix} \\ &= \mathbf{F}_1 \mathbf{Y}_3 + \mathbf{F}_2 \mathbf{Y}'''_{22},\end{aligned}$$

we have

$$\mathbf{E}''_1 = \mathbf{E}''_1 \mathbf{F}_1 \mathbf{Y}_3 + \mathbf{E}''_1 \mathbf{F}_2 \mathbf{Y}'''_{22},$$

that is,

$$\mathbf{E}''_1 = \mathbf{E}_3 + \mathbf{E}_4 \mathbf{Y}'''_{22}, \tag{3.10}$$

where $\mathbf{E}_3 = \mathbf{E}''_1 \mathbf{F}_1 \mathbf{Y}_3$ and $\mathbf{E}_4 = \mathbf{E}''_1 \mathbf{F}_2$. Moreover, since

$$\mathbf{E}''' \begin{pmatrix} \mathbf{F}_1 & \mathbf{F}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{E}''_1 \mathbf{F}_1 & \mathbf{E}''_1 \mathbf{F}_2 \\ \mathbf{0} & \mathbf{I}_{\alpha_{22f}} \end{pmatrix},$$

we have, $rank\,(\mathbf{E}_3) \leq rank\,(\mathbf{E}''_1 \mathbf{F}_1) = rank\,(\mathbf{E}''') - \alpha_{22f} \leq \beta' - \alpha_{22f}$. By (3.9) and (3.10),

$$\mathbf{Y}''_{21} = \mathbf{D}'_1 \mathbf{M}' + \mathbf{E}_4 \mathbf{Y}'''_{22} + \mathbf{E}_3.$$

Let $\mathbf{E}'_4$ be the $m \times \alpha_{22f}$ matrix and $\mathbf{E}'_3$ be the $m \times n$ matrix obtained respectively from matrices $\mathbf{E}_4$ and $\mathbf{E}_3$ by inserting all-zero rows below the last row if $\alpha_1 \leq m$ and by deleting the $\alpha_1 - m$ last rows else. Then,

$$\mathbf{Y}'''_{21} = \mathbf{D}''_1 \mathbf{M}' + \mathbf{E}'_4 \mathbf{Y}'''_{22} + \mathbf{E}'_3. \tag{3.11}$$

If we left multiply both sides of (3.11) by $\mathbf{Q}'$ we get

$$\widehat{\mathbf{Y}}_{21} = \mathbf{M} + \widehat{\mathbf{D}}_1 \mathbf{W}_1 + \mathbf{W}_2 \widehat{\mathbf{Y}}_{22} + \widehat{\mathbf{E}}.$$

where $\mathbf{W}_1 = \mathbf{M}'$, $\mathbf{W}_2 = \mathbf{Q}' \mathbf{E}'_4$ and $\widehat{\mathbf{E}} = \mathbf{Q}' \mathbf{E}'_3$.

- Assume that $freerank\,(\mathbf{Y}_{22}) = 0$. Then, by (3.9), we have

$$\widehat{\mathbf{Y}}_{21} = \mathbf{M} + \widehat{\mathbf{D}}_1 \mathbf{W}_1 + \widehat{\mathbf{E}},$$

where $\mathbf{W}_1$ is defined as above and $\widehat{\mathbf{E}} = \mathbf{Q}' \mathbf{E}_5$, where $\mathbf{E}_5$ is the $m \times n$ matrix obtained from the matrix $\mathbf{E}''_1$ by inserting all-zero rows below the last row if $\alpha_1 \leq m$ or by deleting the $\alpha_1 - m$ last rows else. ■

Theorem 3.8 and Corollary 2.33 imply the following result.

Corollary 3.9 *With the above notations, assume that $\mathcal{M}$ is the matrix representation of an interleaved Gabidulin code of rank distance d. If $rank\left(\widehat{\mathbf{D}}_1\right) + rank\left(\widehat{\mathbf{Y}}_{22}\right) + 2rank\left(\widehat{\mathbf{E}}\right) \leq d - 1$, then the transmitted codeword can be recovered.*

3.3.4 Application example

The following example is computed using SageMathCloud [65]. For more details, see Appendix A.

Example 3.10 *Let* $R = \mathbb{Z}_8$, $S = R[z] / (z^5 + 4z^3 + 7z^2 + 2z + 7)$ *and* $a = z + (z^5 + 4z^3 + 7z^2 + 2z + 7)$. *Then* S *is a Galois extension of* R *where the Galois group is generated by a power map* $\sigma : a \mapsto a^2$. *Set* $\mathbf{g}^{(1)} = \mathbf{g}^{(2)} = (a, a^2, a^3, a^4, a^5)$; $f^{(1)} = 1+2a+3a^2+5a^3$; $f^{(2)} = 1+4a+7a^2+2a^3+5a^4$; $\mathbf{c}^{(1)} = f^{(1)}\left(\mathbf{g}^{(1)}\right)$; $\mathbf{c}^{(2)} = f^{(2)}\left(\mathbf{g}^{(2)}\right)$. *Then* $\left(\begin{array}{cc} \mathbf{c}^{(1)} & \mathbf{c}^{(2)} \end{array}\right)$ *is a codeword of the interleaved Gabidulin code* $IGab_{(1,1)}\left(\mathbf{g}^{(1)}, \mathbf{g}^{(2)}\right)$. *Let*

$$\mathbf{M} = \left(\begin{array}{cc} \mathbf{M}_1 & \mathbf{M}_2 \end{array}\right)$$

where $\mathbf{M}_1$ *and* $\mathbf{M}_2$ *are respectively the matrix representations of* $\mathbf{c}^{(1)}$ *and* $\mathbf{c}^{(2)}$ *in the basis* $(1, a, a^2, a^3, a^4)$.

The transmitted matrix is

$$\mathbf{X} = \left(\begin{array}{ccc} \mathbf{0}_{5\times 2} & \mathbf{I}_5 & \mathbf{M} \end{array}\right)$$

Assume that

$$\mathbf{A} = \left(\begin{array}{ccccc} 5 & 6 & 6 & 3 & 3 \\ 3 & 2 & 7 & 1 & 0 \\ 4 & 6 & 0 & 6 & 7 \\ 4 & 1 & 2 & 1 & 0 \\ 1 & 4 & 5 & 6 & 2 \\ 2 & 5 & 7 & 5 & 0 \\ 4 & 4 & 1 & 3 & 1 \end{array}\right)$$

and

$$\mathbf{E} = \mathbf{BZ}$$

where

$$\mathbf{B} = \left(\begin{array}{ccc} 6 & 4 & 2 \\ 4 & 5 & 5 \\ 2 & 5 & 4 \\ 6 & 7 & 6 \\ 3 & 7 & 2 \\ 2 & 7 & 1 \\ 6 & 0 & 7 \end{array}\right)$$

and

$$\mathbf{Z} = \left(\begin{array}{ccccccccccccccccc} 0 & 7 & 7 & 0 & 6 & 3 & 3 & 1 & 5 & 2 & 6 & 7 & 4 & 3 & 4 & 1 & 2 \\ 0 & 0 & 7 & 5 & 2 & 4 & 5 & 2 & 3 & 0 & 3 & 0 & 4 & 5 & 5 & 6 & 5 \\ 6 & 3 & 0 & 5 & 5 & 7 & 2 & 3 & 7 & 0 & 4 & 3 & 5 & 1 & 5 & 2 & 5 \end{array}\right)$$

The received matrix is

$$\mathbf{Y} = \mathbf{AX} + \mathbf{BZ}.$$

By Theorem 3.8, there are the matrices $\mathbf{W}_1$, $\mathbf{W}_2$ *and* $\widehat{\mathbf{E}}$ *such that*

$$\widehat{\mathbf{Y}}_{21} = \mathbf{M} + \widehat{\mathbf{D}}_1\mathbf{W}_1 + \mathbf{W}_2\widehat{\mathbf{Y}}_{22} + \widehat{\mathbf{E}} \tag{3.12}$$

with rank $\left(\widehat{\mathbf{E}}\right) \leq 1$, *where*

$$\widehat{\mathbf{Y}}_{21} = \begin{pmatrix} 0 & 6 & 5 & 4 & 5 & 7 & 3 & 6 & 4 & 4 \\ 5 & 7 & 5 & 1 & 3 & 5 & 6 & 7 & 4 & 6 \\ 0 & 2 & 4 & 7 & 3 & 5 & 2 & 1 & 0 & 3 \\ 7 & 1 & 7 & 3 & 5 & 7 & 5 & 1 & 2 & 1 \\ 5 & 7 & 3 & 6 & 4 & 0 & 2 & 2 & 0 & 1 \end{pmatrix}$$

$$\widehat{\mathbf{D}}_1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 6 \\ 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 7 \\ 0 & 0 & 0 & 0 & 7 \end{pmatrix}$$

and

$$\widehat{\mathbf{Y}}_{22} = \begin{pmatrix} 0 & 7 & 6 & 2 & 1 & 6 & 7 & 5 & 5 & 1 \end{pmatrix}$$

The vector representation of (3.12) in the basis $(1, a, a^2, a^3, a^4)$ *is*

$$\mathbf{y} = \mathbf{c} + a^{(R)}\mathbf{B}^{(R)} + \mathbf{a}^{(C)}\mathbf{B}^{(C)} + \boldsymbol{\varepsilon}^{(E)}$$

where $\mathbf{y}$, $\mathbf{c}$, $\mathbf{a}^{(C)}$, $\boldsymbol{\varepsilon}^{(E)}$ *are respectively the vector representations of* $\widehat{\mathbf{Y}}_{21}$, $\mathbf{M}$, $\mathbf{W}_2$, $\widehat{\mathbf{E}}$ *and* $\mathbf{B}^{(C)} = \widehat{\mathbf{Y}}_{22}$, $\mathbf{B}^{(R)}$ *is the last row of* $\mathbf{W}_1$, $a^{(R)} = 7a^4 + 7a^3 + 4a^2 + 6a + 4$.

Set

$$\mathbf{y} = \begin{pmatrix} \mathbf{y}^{(1)} & \mathbf{y}^{(2)} \end{pmatrix}$$

where $\mathbf{y}^{(1)} \in S^5$ *and* $\mathbf{y}^{(2)} \in S^5$. *Then*

$$\mathbf{y}^{(1)} = \mathbf{c}^{(1)} + a^{(R)}\mathbf{B}^{(R,1)} + \mathbf{a}^{(C)}\mathbf{B}^{(C,1)} + \boldsymbol{\varepsilon}^{(E,1)}$$

$$\mathbf{y}^{(2)} = \mathbf{c}^{(2)} + a^{(R)}\mathbf{B}^{(R,2)} + \mathbf{a}^{(C)}\mathbf{B}^{(C,2)} + \boldsymbol{\varepsilon}^{(E,2)}$$

Let

$$P^{(R)} = X + 5a^4 + a^3 + 6a^2 + 2a + 2,$$

$$\mathbf{F}^{(R,1)} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 7 & 6 & 2 & 0 \\ 1 & 2 & 7 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

and

$$\mathbf{F}^{(R,2)} = \begin{pmatrix} 1 & 5 & 5 & 1 \\ 7 & 3 & 3 & 6 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

Then, $P^{(R)}\left(a^{(R)}\right)=0$, $\mathbf{B}^{(C,1)}\mathbf{F}^{(R,1)}=\mathbf{0}$ *and* $\mathbf{B}^{(C,2)}\mathbf{F}^{(R,2)}=\mathbf{0}$.

Set $\mathbf{y}'^{(l)}=P^{(R)}\left(\mathbf{y}^{(l)}\right)\mathbf{F}^{(C,l)}$, $\mathbf{g}'^{(l)}=\mathbf{g}^{(l)}\mathbf{F}^{(C,l)}$, $\mathbf{c}'^{(l)}=P^{(R,l)}\left(\mathbf{c}^{(l)}\right)\mathbf{F}^{(C,l)}$, *for* $l\in\{1,2\}$. *Thus, by Theorem 2.32, there is* $\boldsymbol{\varepsilon}'\in S^8$ *such that*

$$\left(\begin{array}{cc}\mathbf{y}'^{(1)} & \mathbf{y}'^{(2)}\end{array}\right)=\left(\begin{array}{cc}\mathbf{c}'^{(1)} & \mathbf{c}'^{(2)}\end{array}\right)+\varepsilon'$$

where $rank\left(\boldsymbol{\varepsilon}'\right)\leq 1$.

When we apply Algorithm 4 for the received word $\left(\begin{array}{cc}\mathbf{y}'^{(1)} & \mathbf{y}'^{(2)}\end{array}\right)$ *of the interleaved Gabidulin code* $IGab_{(2,2)}\left(\mathbf{g}'^{(1)},\mathbf{g}'^{(2)}\right)$, *it returns* $\left(f'^{(1)},f'^{(2)}\right)$ *where* $f'^{(1)}=(7a^4+5a^3+5a+1)X+4a^4+3a^3+4a+1$ *and* $f'^{(2)}=(5a^4+7a^3+5a^2+4a+6)X+2a^4+5a^3+3a^2+5a$. *The left Euclidean division of* $f'^{(1)}$ *and* $f'^{(2)}$ *by* $P^{(R)}$ *gives respectively* $f^{(1)}$ *and* $f^{(2)}$.

Conclusion and perspectives

Conclusion

We have studied some properties of rank-metric codes that are extended from the case of finite fields to finite principal ideal rings. We have first generalized the rank metric and established the rank-metric Singleton bound. As in the case of finite fields, we have shown that Gabidulin codes achieve this bound and the dual of a Gabidulin code is also a Gabidulin code. We have proved that collaborative decoding of interleaved Gabidulin codes can be translated to the problem of reconstruction of skew polynomials. We have used the theory of Gröbner bases of modules over skew polynomials to give the unique decoding, minimal list decoding, and error-erasure decoding algorithms of interleaved Gabidulin codes. Specifically, we have given an iterative algorithm that can uniquely decode interleaved Gabidulin codes beyond the error correction capability. We have also shown that the errors and erasures decoding of an interleaved Gabidulin code is reduced to errors decoding of another interleaved Gabidulin code. These codes are then applied in space-time coding and in random linear network coding. More precisely, we have shown that there is a rank-preserving map from a finite principal ideal ring to a complex signal set and we have used it to construct an optimal space-time block code. Using the lifting construction, we have shown that the decoding problem for random linear network coding over finite principal ideal rings can be reformulated as an error-erasure decoding problem for rank-metric codes.

Perspectives

The complexity of the algorithms. In our algorithms, we have used some operations on skew polynomials (addition, multiplication, Euclidean division, evaluation, ...). In [62], Puchinger and Wachter-Zeh gave fast algorithms for operations on linearized polynomials using normal bases. Since the Galois extension of finite principal ideal rings admits a normal basis [14], in our future work, we will first extend the results of [62] to finite principal ideal rings, then we will give the complexity of our algorithms.

The failure probability of unique decoding algorithm. As we specified in Remark 2.31, in our future work, we will investigate the connection between Algorithm 4 and [68, Algorithm 4]. This will allow us to give the upper bound of the failure probability of Algorithm 4.

Decoding space-time codes using rank metric codes. As we specified in Subsection 3.2.2, in our future work, we will study the decoding of space-time codes using the combination of the integer-forcing linear receiver and the decoding algorithms of interleaved Gabidulin codes.

Generalization of other properties. We have shown that some properties of rank-metric codes can be extended over finite principal ideal rings. In our future work, we will see if this is the case for other properties, such as packing properties, covering properties, MacWilliams Identity [27].

Cryptography based on rank-metric codes. In [25], Gabidulin et al. proposed a cryptosystem using rank-metric codes over finite fields. In finite principal ideal rings we have zero divisors that can be used to improve the cryptosystem. So, in our future work, we will study the work of [25] over finite principal ideal rings.

Index

Bibliography

[1] M. A. Armand. List decoding of generalized Reed-Solomon codes over commutative rings. *IEEE transactions on information theory*, 51(1):411–419, 2005.

[2] H. M. Asif, B. Honary, and M. T. Hamayun. Gaussian integers and interleaved rank codes for space–time block codes. *International Journal of Communication Systems*, 30(1), 2015.

[3] D. Augot, P. Loidreau, and G. Robert. Rank metric and Gabidulin codes in characteristic zero. In *Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on*, pages 509–513. IEEE, 2013.

[4] D. Augot, P. Loidreau, and G. Robert. Generalized Gabidulin codes over fields of any characteristic. *Designs, Codes and Cryptography*, 86(8):1807–1848, 2018.

[5] M. Auslander and O. Goldman. The Brauer group of a commutative ring. *Transactions of the American Mathematical Society*, 97(3):367–409, 1960.

[6] B. Badic. *Space-time block coding for multiple antenna systems.* PhD thesis, Vienna University of Technology, 2005.

[7] H. Bartz and V. Sidorenko. Improved syndrome decoding of lifted L-interleaved Gabidulin codes. *Designs, Codes and Cryptography*, 87(2-3):547–567, 2019.

[8] H. Bartz and A. Wachter-Zeh. Efficient decoding of interleaved subspace and Gabidulin codes beyond their unique decoding radius using Gröbner bases. *Advances in Mathematics of Communications*, 12(4):773–804, 2018.

[9] W. C. Brown. *Matrices over commutative rings.* Marcel Dekker, Inc., 1993.

[10] B. Buchberger. *An algorithm for finding a basis for the residue class ring of a zero-dimensional polynomial ideal.* PhD thesis, Ph. D. thesis, University of Innsbruck, Austria, 1965.

[11] J. L. Bueso, J. Gómez-Torrecillas, and A. Verschoren. *Algorithmic methods in non-commutative algebra: Applications to quantum groups.* Kluwer Academic Publishers, Dordrecht, 2003.

[12] A. Butson and B. Stewart. Systems of linear congruences. *Canadian Journal of Mathematics*, 7:358–368, 1955.

[13] E. Byrne and P. Fitzpatrick. Hamming metric decoding of alternant codes over Galois rings. *IEEE Transactions on Information Theory*, 48(3):683–694, 2002.

[14] S. U. Chase, D. K. Harrison, and A. Rosenberg. *Galois theory and cohomology of commutative rings*, volume 52. American Mathematical Soc., 1969.

[15] A. A. De Andrade and R. Palazzo Jr. Construction and decoding of BCH codes over finite commutative rings. *Linear Algebra and Its Applications*, 286(1-3):69–85, 1999.

[16] P. Delsarte. Bilinear forms over a finite field, with applications to coding theory. *Journal of Combinatorial Theory, Series A*, 25(3):226–241, 1978.

[17] F. DeMeyer and E. Ingraham. *Separable algebras over commutative rings*. Lecture Notes in Mathematics. Springer-Verlag, Berlin, Heidelberg, New York, 1 edition, 1971.

[18] ELECTRONICS HUB. Wireless communication: Introduction, types and applications. `https://www.electronicshub.org/wireless-communication-introduction-types-applications`. Accessed: 2019-09-15.

[19] D. Experts. *Guide to RRB Junior Engineer Stage II Civil & Allied Engineering 3rd Edition*. Disha Publications, 2019.

[20] Y. Fan, S. Ling, and H. Liu. Matrix product codes over finite commutative Frobenius rings. *Designs, codes and cryptography*, 71(2):201–227, 2014.

[21] C. Feng, R. W. Nóbrega, F. R. Kschischang, and D. Silva. Communication over finite-chain-ring matrix channels. *IEEE Transactions on Information Theory*, 60(10):5899–5917, 2014.

[22] C. Feng, D. Silva, and F. R. Kschischang. An algebraic approach to physical-layer network coding. *IEEE Transactions on Information Theory*, 59(11):7576–7596, 2013.

[23] P. Fitzpatrick. On the key equation. *IEEE Transactions on Information Theory*, 41(5):1290–1302, 1995.

[24] E. M. Gabidulin. Theory of codes with maximum rank distance. *Problemy Peredachi Informatsii*, 21(1):3–16, 1985.

[25] E. M. Gabidulin, A. Paramonov, and O. Tretjakov. Ideals over a non-commutative ring and their application in cryptology. In *Workshop on the Theory and Application of Cryptographic Techniques*, pages 482–489. Springer, 1991.

[26] E. M. Gabidulin, N. I. Pilipchuk, and M. Bossert. Decoding of random network codes. *Problems of information transmission*, 46(4):300–320, 2010.

[27] M. Gadouleau. *Algebraic codes for random linear network coding.* PhD thesis, 2009.

[28] G. Ganske and B. McDonald. Finite local rings. *The Rocky Mountain Journal of Mathematics*, pages 521–540, 1973.

[29] D. Goldschmidt. *Algebraic functions and projective curves*, volume 215 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 2003.

[30] E. Gorla and A. Ravagnani. Partial spreads in random network coding. *Finite Fields and Their Applications*, 26:104–115, 2014.

[31] E. Gorla and A. Ravagnani. An algebraic framework for end-to-end physical-layer network coding. *IEEE Transactions on Information Theory*, 64(6):4480–4495, 2018.

[32] K. Hofbauer and G. Kubin. Aeronautical voice radio channel modelling and simulation-a tutorial review. In *Proceedings of the 2nd International Conference on Research in Air Transportation (ICRAT 2006)*, 2006.

[33] H. Jiménez and O. Lezama. Gröbner bases for modules over sigma-PBW extensions. *Acta Mathematica Academiae Paedagogicae Nyıregyháziensis*, 31(3), 2015.

[34] I. Kaplansky. Elementary divisors and modules. *Transactions of the American Mathematical Society*, 66(2):464–491, 1949.

[35] KATREIN. Spacing out...getting the most out of mimo with proper antenna spacing. `https://www.kathreinusa.com`, 2017.

[36] Keysight Technologies. Addressing multi-channel synchronization and calibration for mimo and beamforming applications. `https://www.keysight.com`, 2014.

[37] T. Kiran and B. S. Rajan. Optimal STBCs from codes over Galois rings. In *Personal Wireless Communications (CPWC), 2005 IEEE International Conference on*, pages 120–124. IEEE, 2005.

[38] R. Koetter and F. R. Kschischang. Coding for errors and erasures in random network coding. *IEEE Transactions on Information theory*, 54(8):3579–3591, 2008.

[39] E. R. Kolchin. *Differential algebra and algebraic groups*, volume 54. Academic press, 1973.

[40] M. Kuijper and R. Pinto. An iterative algorithm for parametrization of shortest length linear shift registers over finite chain rings. *Designs, Codes and Cryptography*, 83(2):283–305, 2017.

[41] M. Kuijper and A.-L. Trautmann. Iterative list-decoding of Gabidulin codes via Gröbner based interpolation. In *Information Theory Workshop (ITW), 2014 IEEE*, pages 581–585. IEEE, 2014.

[42] V. L. Kurakin. The Berlekamp–Massey algorithm over finite rings, modules and bimodules. *Diskretnaya Matematika*, 10(4):3–34, 1998.

[43] T.-Y. Lam. *Lectures on modules and rings.* Graduate Texts in Mathematics 189. Springer-Verlag New York, 1 edition, 1999.

[44] Y. Liu, M. P. Fitz, and O. Y. Takeshita. A rank criterion for QAM space-time codes. *IEEE Transactions on Information Theory*, 48(12):3062–3079, 2002.

[45] P. Loidreau. A Welch–Berlekamp like algorithm for decoding Gabidulin codes. In *Proceedings of the 4th International Workshop on Coding and Cryptography (WCC'2005)*, pages 36–45. Springer, Berlin, Heidelberg, 2006.

[46] P. Loidreau and R. Overbeck. Decoding rank errors beyond the error-correction capability. *in International Workshop on Algebraic and Combinatorial Coding Theory (ACCT 2006)*, pages 168–190, 2006.

[47] H.-f. Lu and P. V. Kumar. Rate-diversity tradeoff of space-time codes with fixed alphabet and optimal constructions for PSK modulation. *IEEE Transactions on Information Theory*, 49(10):2747–2751, 2003.

[48] P. Lusina, E. Gabidulin, and M. Bossert. Maximum rank distance codes as space-time codes. *IEEE Transactions on Information Theory*, 49(10):2757–2760, 2003.

[49] B. R. McDonald. *Finite rings with identity*, volume 28. Marcel Dekker Incorporated, 1974.

[50] A. Mohammadi and F. Ghannouchi. *RF Transceiver Design for MIMO Wireless Communications.* Lecture Notes in Electrical Engineering. Springer Berlin Heidelberg, 2012.

[51] B. Nazer and M. Gastpar. Compute-and-forward: Harnessing interference through structured codes. *IEEE Transactions on Information Theory*, 57(10):6463–6486, 2011.

[52] A. A. Nechaev. Finite rings with applications. *Handbook of Algebra*, 5:213–320, 2008.

[53] R. W. Nóbrega, C. Feng, D. Silva, and B. F. Uchôa-Filho. On multiplicative matrix channels over finite chain rings. In *Network Coding (NetCod), 2013 International Symposium on*, pages 1–6. IEEE, 2013.

[54] R. W. Nóbrega, B. F. Uchôa-Filho, and D. Silva. On the capacity of multiplicative finite-field matrix channels. In *Information Theory Proceedings (ISIT), 2011 IEEE International Symposium on*, pages 341–345. IEEE, 2011.

[55] G. H. Norton and A. Sălăgean. Gröbner bases and products of coefficient rings. *Bulletin of the Australian Mathematical Society*, 65(1):145–152, 2002.

[56] H. O'Keeffe and P. Fitzpatrick. Gröbner basis solutions of constrained interpolation problems. *Linear algebra and its applications*, 351:533–551, 2002.

[57] O. Ore. On a special class of polynomials. *Transactions of the American Mathematical Society*, 35(3):559–584, 1933.

[58] O. Ore. Theory of non-commutative polynomials. *Annals of mathematics*, pages 480–508, 1933.

[59] J. G. Proakis and M. Salehi. *Digital communications*, volume 4. McGraw-hill New York, 2001.

[60] S. Puchinger, J. R. né Nielsen, W. Li, and V. Sidorenko. Row reduction applied to decoding of rank-metric and subspace codes. *Designs, Codes and Cryptography*, 82(1-2):389–409, 2017.

[61] S. Puchinger, S. Stern, M. Bossert, and R. F. Fischer. Space-time codes based on rank-metric codes and their decoding. In *Wireless Communication Systems (ISWCS), 2016 International Symposium on*, pages 125–130. IEEE, 2016.

[62] S. Puchinger and A. Wachter-Zeh. Fast operations on linearized polynomials and their applications in coding theory. *Journal of Symbolic Computation*, 89:194–215, 2018.

[63] R. M. Roth. Maximum-rank array codes and their application to crisscross error correction. *IEEE transactions on Information Theory*, 37(2):328–336, 1991.

[64] C. Rust and G. J. Reid. Rankings of partial derivatives. In *Proceedings of the 1997 international symposium on Symbolic and algebraic computation*, pages 9–16. ACM, 1997.

[65] I. SageMath. *SageMathCloud Online Computational Mathematics*, 2019. SageMath-Cloud https://cloud. sagemath. com.

[66] A. Sakzad, J. Harshan, and E. Viterbo. Integer-forcing MIMO linear receivers based on lattice reduction. *IEEE transactions on wireless communications*, 12(10):4905–4915, 2013.

[67] V. Sidorenko and M. Bossert. Decoding interleaved Gabidulin codes and multisequence linearized shift-register synthesis. In *Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on*, pages 1148–1152. IEEE, 2010.

[68] V. Sidorenko, L. Jiang, and M. Bossert. Skew-feedback shift-register synthesis and decoding interleaved Gabidulin codes. *IEEE Transactions on Information Theory*, 57(2):621–632, 2011.

[69] D. Silva, F. R. Kschischang, and R. Koetter. A rank-metric approach to error control in random network coding. *IEEE transactions on information theory*, 54(9):3951–3967, 2008.

[70] D. Silva, F. R. Kschischang, and R. Kotter. Communication over finite-field matrix channels. *IEEE Transactions on Information Theory*, 56(3):1296–1305, 2010.

[71] H. J. S. Smith. On systems of linear indeterminate equations and congruences. *Philosophical transactions of the royal society of london*, 151:293–326, 1861.

[72] A. Storjohann. *Algorithms for matrix canonical forms.* PhD thesis, ETH Zurich, 2000.

[73] G. Stüber. *Principles of Mobile Communication.* Springer International Publishing, 2017.

[74] V. Tarokh, N. Seshadri, and A. R. Calderbank. Space-time codes for high data rate wireless communication: Performance criterion and code construction. *IEEE transactions on information theory*, 44(2):744–765, 1998.

[75] T. Tran. *Wireless Communication: Learn to Wireless Communication.* 2019.

[76] D. Tse and P. Viswanath. *Fundamentals of wireless communication.* Cambridge university press, 2005.

[77] V. Venugopal. Lecture notes in communication systems. `http://vvv.ece.illinois.edu/ece459/handouts/notes.pdf`, 2000.

[78] M. Viswanathan. Simulation of digital communication systems using matlab. *Mathuranathan Viswanathan at Smashwords*, 2013.

[79] A. Wachter-Zeh and A. Zeh. List and unique error-erasure decoding of interleaved Gabidulin codes with interpolation techniques. *Designs, codes and cryptography*, 73(2):547–570, 2014.

[80] H. Xie, Z. Yan, and B. W. Suter. General linearized polynomial interpolation and its applications. In *2011 International Symposium on Networking Coding*, pages 1–4. IEEE, 2011.

[81] J. Zhan, B. Nazer, U. Erez, and M. Gastpar. Integer-forcing linear receivers. *IEEE Transactions on Information Theory*, 60(12):7661–7685, 2014.

Appendix A: SAGE Implementation

We implemented in SageMathCloud the algorithms that we gave in the manuscript. We also gave more details in the examples.

RankMetricCodesOverFinitePIR504.sagews

Author	**Hermann Tchatchiem Kamche**
Date	**2019-09-21T00:58:33**
Project	**161292cf-d91b-443f-99ea-49c42e2f0fa9**
Location	[RankMetricCodesOverFinitePIR504.sagews]()
Original file	[RankMetricCodesOverFinitePIR504.sagews]()


```
#########################################
#   Implementation of "Rank-Metric Codes
#    Over Finite Principal Ideal Rings
#         and Applications"
#########################################
#
# In the implementation, we assume that the ring `R` is the integer modulo ring `Z_n`.
#  Implematation is done in SageMathCloud (https://cocalc.com/)
#
# H. Tchatchiem Kamche (tchatchiemh@yahoo.fr) and C. Mouaha (cmouaha@yahoo.fr)
#
#  Contents
# I. Galois extension
# II. Decomposition of an element in Finite chain rings
# III. Smith Normal Form and Rank Metric
# IV. Skew polynomials
# V. Vector representation of matrices
# VI. Unique decoding gabidulin codes using Smith normal form
# VII. Computing a Grobner basis
# VIII. Unique decoding beyond the error correction capability
# IX. Comparison of unique decoding interleaved Gabidulin codes
# X. Decoding of random linear network codes
#
#
# I. Galois extension
#
# The ring `Z_n` is isomorphic to the product of the rings of integer modulo a power
# of a prime number. Thus, to construct the Galois extension of `Z_n`, it suffices
# to construct that of `Z_ {p ^ nu}` where `p` a prime and `nu` is a positive integer.
# We will construct a Galois extension of `Z_ {p ^ nu}` such that the multiplicative
# order of `a` is `p ^ m-1`, where` a` is a generator of the Galois extension and `m`
# is the dimension of the Galois extension. Therefore, the Galois group will be generated
# by a power map `a |--> a ^ p`.
#
# I.1. Program
#
def HenselLiftOfPrimitivePolynomial(p,nu,m):
    """
    Input: `p` the characteristic of the residue field,
    `m` the dimension of the Galois extension,
    `nu` the nilpotency index.
    Output: a monic polynomial `h` in `Z_ {p ^ nu}[z]` of degree `m`
    such that `h`  divises `z^(p^m-1)-1` and
    the projection of `h` in `GF(p)[z]` is a primitive polynomial.
    """
    Zpz.<z>=QQ[]
    Hensel=Zpz(z^(p^m-1)-1).hensel_lift(p, nu)
    Conway=conway_polynomial(p,m)
    Fpz.<z>=GF(p)[]
    i=0
    while  Fpz(Conway)<>Fpz(Hensel[i]) :
        i=i+1
    return Hensel[i]
#
# I.2. Example
#
```

```
# We will construct a Galois Extension of `Z_12` of dimension `4`.
# Set `R12=Z_12`, `R3=Z_3` and `R4=Z_4`. The map `R3xR4 --> R12`
# given by `(x,y) |--> (4*x+9*y)` is an isomorphim. Let `S3=R3[a3]=R3[z]/(h3)`
# and `S4=R4[a4]=R4[z]/(h4)` be the Galois extension of `R3` and `R4` such that
# the Galois groups are respectively generated by the power maps
#  `sigma3: a3 |--> a3 ^ 3` and `sigma4: a4 |--> a4 ^ 2`
# Since `R3[z]xR4[z]` is somorphic to `R12[z]`, the image of `(h3,h4)` in `R12[z]`
# is `h12:=4*h3+9*h4`. Set `S12:=R12[a12]=R12[z]/(h12)`. Them `S12` is a Galois Extension
# of `R12` where the Galois group is generated by the power map
#  `sigma12: a12 |--> 4*a12 ^ 3+9*a12^2`
#
R12=Integers(12)
R3=Integers(3)
p3=3
nu3=1
R3z.<z>=R3[]
R4=Integers(4)
p4=2
nu4=2
R4z.<z>=R4[]
m12=4
h3=R3z(HenselLiftOfPrimitivePolynomial(p3,nu3,m12))
h4=R4z(HenselLiftOfPrimitivePolynomial(p4,nu4,m12))
R12z.<z12>=R12[]
h12=R12[`z`](4*R12[`z`](h3)+9*R12[`z`](h4))
S12.<a12>=R12z.quotient(h12)
b12=4*a12 ^ 3+9*a12^2
sigma12 = S12.hom([b12])
c12=S12.random_element()
print "h3","=",h3
""
print "h4","=",h4
""
print "h12","=", h12
""
print "sigma12 :", sigma12
""
print "c12","=", c12
""
print "sigma12(c12)","=", sigma12(c12)
""
print (sigma12^m12)(c12)==c12
```

```
h3 = z^4 + 2*z^3 + 2

''

h4 = z^4 + 2*z^2 + 3*z + 1

''

h12 = z^4 + 8*z^3 + 6*z^2 + 3*z + 5

''

sigma12 : Ring endomorphism of Univariate Quotient Polynomial Ring in a12 over Ring of integers modulo 12 wi
modulus z12^4 + 8*z12^3 + 6*z12^2 + 3*z12 + 5
  Defn: a12 |--> 4*a12^3 + 9*a12^2

''

c12 = 2*a12^3 + 10*a12^2 + 9

''

sigma12(c12) = 6*a12^3 + 6*a12^2 + 2*a12 + 7

''

True
```

```
# II. Decomposition of an element in Finite chain rings
#
# Whem `R=` Z_ {p ^nu}`, them `S` is a finite chain ring whose the maximal
# ideal is generated by `p`. Thus, any element `u` in `S` can by decomposed in to
```

```
# `u:=p^j*v` where `v` is a unit and `0<=j<=nu`
#
# II.1. Program
#
def ValuationOf(u,p,nu):
    """
    Input: `u:=p^j*v` where `v` is a unit
    Output: `j`
    """
    S=parent(u)
    i=0;
    while S((p^i)*u)<>S(0) :
        i=i+1
    return nu-i
#
def NormOf(u,p,nu):
    """
    Input: `u:=p^j*v` where v is a unit
    Output: `p^j`
    """
    S=parent(u)
    i=0;
    while S((p^i)*u)<>S(0) :i=i+1;
    return p^(nu-i)
#
def UnitOf(u,p,nu):
    """
    Input: `u:=p^j*v`
    where `v` is a unit in the ring `S= Z_ {p ^nu}[a]`
    Output: `v`
    """
    S=parent(u)
    a=S.gen()
    v=S(1)
    if S(u)==S(0):
        v=1
    else :
        w=ZZ[`z`](S(u).lift())//NormOf(u,p,nu)
        v=w(a)
    return S(v)
#
# II.2. Example
#
R9=Integers(9)
p9=3
nu9=2
m9=3
R9z.<z>=R9[]
h9=R9z(HenselLiftOfPrimitivePolynomial(p9,nu9,m9))
S9.<a9>=R9z.quotient(h9)
sigma9 = S9.hom([a9^p9])
S9x.<X> = S9['X',sigma9]
u9=S9.random_element()
print u9
""
print NormOf(u9,p9,nu9)
""
print UnitOf(u9,p9,nu9)
""
print u9==S9(NormOf(u9,p9,nu9)*UnitOf(u9,p9,nu9))
```

```
4*a9^2 + a9 + 1
''
1
''
4*a9^2 + a9 + 1
''
True
```

```
# III. Smith Normal Form and Rank Metric
#
# III.1. Smith Normal Form and Rank Metric over `Z_n`
#
# The Smith Normal Form are implemented in SageMath in the ring `Z`.
# We will use it to compute the Smith Normal Form in `Z_n`.
#
# III.1.1. Program
def SmithNormalFormOf(A):
    """
    Input: a matrix `A`
    Output: [D,P,Q,af]
    Where `af` is a freerank of `A`, `D=diag(d_1,…,d_r)` is a
    Smith normal form of `A` such that `d_1=1`, . . ., `d_af=1`,
    and `P`, `Q` are the invertible matrices such that `D=PAQ`.
    """
    R=A.base_ring()
    mu=R.order()
    L=matrix(ZZ,A)
    D=matrix(R,L.smith_form()[0])
    P=matrix(R,L.smith_form()[1])
    Q=matrix(R,L.smith_form()[2])
    af=0
    r=min(D.nrows(),D.ncols())
    u0=R(1)
    while af<r and R(D[af,af]).is_unit() :
        u0=ZZ(D[af,af])
        u1=xgcd(u0,mu)[1]
        u2=R(u1)
        D[af,af]=u2*D[af,af]
        for j in [0..P.nrows()-1]: P[af,j]=u2*P[af,j]
        af=af+1
    return [D,P,Q,af]
#
def RankOf(A):
    R=A.base_ring()
    ar=0
    D=SmithNormalFormOf(A)[0]
    r=min(D.nrows(),D.ncols())
    while ar<r and R(D[ar,ar])<>R(0) :
        ar=ar+1
    return ar
#
def FreeRankOf(A):
    return SmithNormalFormOf(A)[3]
#
#
# III.1.2. Example
# The following example is given in our manuscript.
#
A12=matrix(R12,[
[8, 10, 4, 4],
[4, 2, 8, 2],
[11, 6, 0, 6]
])
D12=SmithNormalFormOf(A12)[0]
P12=SmithNormalFormOf(A12)[1]
Q12=SmithNormalFormOf(A12)[2]
view("A12","=",A12)
""
view("D12","=",D12)
""
view("P12","=",P12)
""
view("Q12","=",Q12)
""
view(D12==P12*A12*Q12)
""
view("rank(A12)","=",RankOf(A12))
""
view("freerank(A12)","=",FreeRankOf(A12))

```

$$A12 = \begin{pmatrix} 8 & 10 & 4 & 4 \\ 4 & 2 & 8 & 2 \\ 11 & 6 & 0 & 6 \end{pmatrix}$$

''

$$D12 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 6 & 0 \end{pmatrix}$$

''

$$P12 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 10 & 0 \end{pmatrix}$$

''

$$Q12 = \begin{pmatrix} 11 & 6 & 0 & 0 \\ 0 & 2 & 1 & 2 \\ 0 & 1 & 0 & 1 \\ 2 & 7 & 11 & 6 \end{pmatrix}$$

''

True

''

rank(A12) = 3

''

freerank(A12) = 1

```
# III.2. Skmith Normal Form and Rank Metric over `Z_n[a]`
#
# The ring `Z_n` is isomorphic to the product of the rings of integer modulo a power
# of a prime number. Thus, to compute the Smith Normal Form in `Z_n[a]`, it suffices
# to compute in `Z_ {p ^ nu}[a]` where `p` a prime and `nu` is a positive integer.
# We use the simple method given in the proof of [Goldschmidt, 2006, Theorem 1.1.12.].
#
# III.2.2. Program
def PivotOf03(A,i1,j1,p,nu,S,ma,na):
    k0=i1; h0=j1; v0=nu; v1=0
    k=i1; h=j1; ti=j1; tj=j1
    PivotIsUnit=false
    while PivotIsUnit==false and ti<ma:
        h=j1; tj=j1
        while PivotIsUnit==false and tj<na:
            v1=ValuationOf(S(A[k,h]),p,nu)
            if v1==0 :
                PivotIsUnit=true
                h0=h
                k0=k
                v0=v1
            else:
                if v1<v0 :
                    h0=h
                    k0=k
                    v0=v1
            tj=tj+1
            h=h+1
        ti=ti+1
        k=k+1
    return [k0,h0,v0]
#
def SmithNormalFormOf2(A1,p,nu):
    """
    Input: a matrix `A1`
    Output: [D,P,Q]
    Where `D=diag(d_1,…,d_r)` is a Smith normal form of `A1`
    such that `d_1=1`, . . ., `d_af=1`, where `af` is a freerank of `A1`
    and `P`, `Q` are the invertible matrices such that `D=PA1Q`.
    """
```

```
    S=A1.base_ring()
    ma=A1.nrows()
    na=A1.ncols()
    A=matrix(S,A1)
    ra=min(ma,na)
    i0=0; j0=0; vv0=0;l0=0
    P=identity_matrix(S,ma)
    Q=identity_matrix(S,na)
    for l in [0..ra-2]:
        [i0,j0,vv0]=PivotOf03(A,l,l,p,nu,S,ma,na)
        A.swap_rows(l,i0)
        P.swap_rows(l,i0)
        A.swap_columns(l,j0)
        Q.swap_columns(l,j0)
        ul=S(InverseOf(UnitOf(S(A[l,l]),p,nu)))
        vl=vv0
        for i in [l..ma-1]:
            A[i,l]=ul*A[i,l]
        for i in [0..na-1]:
            Q[i,l]=ul*Q[i,l]
        for j in [l+1..na-1]:
            wc= S(-UnitOf(S(A[l,j]),p,nu)*p^(ValuationOf(S(A[l,j]),p,nu)-vl))
            A[l,j]=S(0)
            for i in [l+1..ma-1]:
                A[i,j]=S(A[i,j]+wc*A[i,l])
            for i in [0..na-1]:
                Q[i,j]=S(Q[i,j]+wc*Q[i,l])
        for i in [l+1..ma-1]:
            wr= S(-UnitOf(S(A[i,l]),p,nu)*p^(ValuationOf(S(A[i,l]),p,nu)-vl))
            A[i,l]=S(0)
            for j in [l+1..na-1]:
                A[i,j]=S(A[i,j]+wr*A[l,j])
            for j in [0..ma-1]:
                P[i,j]=S(P[i,j]+wr*P[l,j])
    if ma>ra:
        l=ra-1
        [i0,j0,vv0]=PivotOf03(A,l,l,p,nu,S,ma,na)
        A.swap_rows(l,i0)
        P.swap_rows(l,i0)
        ul=S(InverseOf(UnitOf(S(A[l,l]),p,nu)))
        vl=vv0
        A[l,l]=ul*A[l,l]
        for j in [0..ma-1]:
            P[l,j]=ul*P[l,j]
        for i in [l+1..ma-1]:
            wr= S(-UnitOf(S(A[i,l]),p,nu)*p^(ValuationOf(S(A[i,l]),p,nu)-vl))
            A[i,l]=S(0)
            for j in [0..ma-1]:
                P[i,j]=S(P[i,j]+wr*P[l,j])
    if na>ra:
        l=ra-1
        [i0,j0,vv0]=PivotOf03(A,l,l,p,nu,S,ma,na)
        A.swap_columns(l,j0)
        Q.swap_columns(l,j0)
        ul=S(InverseOf(UnitOf(S(A[l,l]),p,nu)))
        vl=vv0
        A[l,l]=ul*A[l,l]
        for i in [0..na-1]:
            Q[i,l]=ul*Q[i,l]
        for j in [l+1..na-1]:
            wc= S(-UnitOf(S(A[l,j]),p,nu)*p^(ValuationOf(S(A[l,j]),p,nu)-vl))
            A[l,j]=S(0)
            for i in [0..na-1]:
                Q[i,j]=S(Q[i,j]+wc*Q[i,l])
    if (na>ra)==False and (ma>ra)==False:
        l=ra-1
        ul=S(InverseOf(UnitOf(S(A[l,l]),p,nu)))
        A[l,l]=ul*A[l,l]
        for i in [0..na-1]:
            Q[i,l]=ul*Q[i,l]
    return [A,P,Q]
#
```

```
def RankOf2(A,p,nu):
    S=A.base_ring()
    ar=0
    D=SmithNormalFormOf2(A,p,nu)[0]
    r=min(D.nrows(),D.ncols())
    while ar<r and S(D[ar,ar])<>S(0) :
        ar=ar+1
    return ar
#
def FreeRankOf2(A,p,nu):
    S=A.base_ring()
    D=SmithNormalFormOf2(A,p,nu)[0]
    af=0
    r=min(D.nrows(),D.ncols())
    u=S(1)
    while af<r and S(D[af,af])==S(1) :
        af=af+1
    return af
#

# IV. Skew polynomials
#
# Skew polynomials are implemented in SageMath.
# We will give some functions that are not implemented.
#
# IV.1. Program
#
def LeftDivisionOf(f,g,sigma,m):
    """
    Input: the skew polynomials `f` and `g` in `Sx=S[X,sigma]`
    such that `g` is monic
    `m` the order of `sigma`.
    Output: [q,r], such that `f=g*q+r` and `deg(r)<deg(g)`
    """
    Sx=parent(f)
    q=Sx(0)
    r=f
    c=Sx(0)
    d1=Sx(g).degree()
    d2=m-d1
    while r<>Sx(0) and d1<=Sx(r).degree():
        t=Sx(r).degree()-d1
        c=((sigma^(d2))(Sx(r).leading_coefficient()))*X^t
        q=Sx(q+c)
        r=Sx(r-g*c)
    return [q,r]
#
def InverseOf(u):
    """
    Input: `u` an inverse element in `S=R[a]`
    Output: the inverse of `u`
    """
    S=parent(u)
    Rz=S.cover_ring()
    R=Rz.base_ring()
    P=S(u).charpoly(z)
    mu=R.order()
    d0=ZZ(P[0])
    d1=xgcd(d0,mu)[1]
    d2=R(d1)
    Q=ZZ[`z`](P)
    v=ZZ[`z`]((Q-Q[0])*ZZ(d2))//z
    return S(-v(u))
#
def MinimalSkewPolynomialOf(v,sigma) :
    """
    Input: `v` a list of elements in `S=R[a]`
    which are linearly independent over `R`
    Output: the monic skew polynomial in `Sx=S[X,sigma]`
    such that the kernel is generated by the elements of `v`
    """
    S=parent(v[0])
```

```
    Sx.<X> = S['X',sigma]
    P=Sx(1)
    for u in v:
        P=Sx((P.operator_eval(u)*X-sigma(P.operator_eval(u)))*P)
    P=InverseOf(Sx(P).leading_coefficient())*P
    return P
#
# IV.2. Example
#
S12x.<X> = S12['X',sigma12]
f12=S12x.random_element(degree=4)
g12=S12x.random_element(degree=3,monic=True)
[q12,r12]=LeftDivisionOf(f12,g12,sigma12,m12)
print "S12x :", S12x
""
print "f12","=", f12
""
print "g12","=", g12
""
print "q12","=", q12
""
print "r12","=", r12
""
print f12==g12*q12+r12
""
P12=MinimalSkewPolynomialOf([1+2*a12^3,6*a12+a12^4],sigma12)
print "P12","=", P12
""
print [P12.operator_eval(1+2*a12^3),P12.operator_eval(6*a12+a12^4)]
```

```
S12x : Skew Polynomial Ring in X over Univariate Quotient Polynomial Ring in a12 over Ring of integers modul
12 with modulus z12^4 + 8*z12^3 + 6*z12^2 + 3*z12 + 5 twisted by a12 |--> 4*a12^3 + 9*a12^2

''

f12 = (11*a12^3 + 7*a12^2 + 5*a12 + 10)*X^4 + (11*a12^3 + 4*a12^2 + 3*a12 + 3)*X^3 + (5*a12^3 + 10*a12^2 +
11*a12 + 9)*X^2 + (10*a12^3 + 2*a12^2 + 6*a12 + 1)*X + 4*a12^3 + 6*a12^2 + 4

''

g12 = X^3 + (9*a12^3 + 4*a12^2)*X^2 + (6*a12^3 + 8*a12^2 + 7*a12 + 8)*X + 4*a12^3 + 8*a12^2 + 7*a12 + 6

''

q12 = (11*a12^3 + 5*a12 + 5)*X + 11*a12^3 + 9*a12^2 + 6*a12 + 11

''

r12 = (7*a12^3 + 3*a12^2 + 6*a12 + 7)*X^2 + (3*a12^3 + 6*a12^2 + 11*a12 + 7)*X + 3*a12^3 + 10*a12 + 11

''

True

''

P12 = X^2 + (4*a12^3 + 11*a12^2 + 9*a12 + 7)*X + 2*a12^3 + a12^2 + 3*a12 + 4

''

[0, 0]
```

```
#
# V. Vector representation of matrices
#
# V.1. Program
#
def CoefficientOf(u):
    """
    Input:`u` in `S=R[a]`
    Output: the list of coefficent of `u`
    in the basis `(1,a,...,a^(m-1))`
    """
    S=parent(u)
    Rz=S.cover_ring()
    a=S.gen()
```

```
    m=S(a).charpoly(Rz.gen()).degree()
    u1=S(u).lift()
    u2=[u1[i] for i in [0..Rz(u1).degree()]]
    u3=[0 for i in [0..m-Rz(u1).degree()-2]]
    return u2+u3
#
def MatrixRepresentationOf(v):
    """
    Input:`v` a list with coefficient in `S=R[a]`
    Output: the matrix representation of `v` in the
    ring `R` relative to the basis `(1,a,...,a^(m-1))`
    """
    S=parent(v[0])
    Rz=S.cover_ring()
    R=Rz.base_ring()
    a=S.gen()
    m=S(a).charpoly(Rz.gen()).degree()
    return matrix(R,len(v),m,[CoefficientOf(v[j]) for j in [0..len(v)-1]]).transpose()
#
def VectorRepresentationOf(V,S):
    """
    Input:`V` a matrix of `m` rows with coefficient in `R`
    Output: the vector representation of `V` in the
    ring `S=R[a]` relative to the basis `(1,a,...,a^(m-1))`
    """
    a=S.gen()
    Rz=S.cover_ring()
    R=Rz.base_ring()
    m=S(a).charpoly(Rz.gen()).degree()
    Bs=matrix(S,1,m,[a^i for i in [0..m-1]])
    v=Bs*V
    return [v[0,i] for i in [0..v.ncols()-1]]
#
# V.2. Example
#
V12=random_matrix(R12,m12,4)
v12=VectorRepresentationOf(V12,S12)
U12=MatrixRepresentationOf(v12)
print V12
""
print v12
""
print v12[0]
""
print CoefficientOf(v12[0])
""
print U12==V12
```

```
[2 4 3 4]
[1 1 1 8]
[3 6 8 8]
[9 8 4 3]
''
[9*a12^3 + 3*a12^2 + a12 + 2, 8*a12^3 + 6*a12^2 + a12 + 4, 4*a12^3 + 8*a12^2 + a12 + 3, 3*a12^3 + 8*a12^2 +
8*a12 + 4]
''
9*a12^3 + 3*a12^2 + a12 + 2
''
[2, 1, 3, 9]
''
True
```

```
# VI. Unique decoding gabidulin codes using Smith normal form
#
# We implement the decoding algorithm of Gabidulin codes
# over the Galois exention of the rings of integer modulo a power
```

```
# of a prime number using the Smith normal form.
#
# VI.1. Program
#
def VandermondeMatrixOf(v,s,sigma):
    S=parent(v[0])
    lv=len(v)
    Vand=[[S(0) for j in [0..lv-1]] for i in [0..s-1]]
    for i in [0..s-1]:
        for j in [0..lv-1]:
            Vand[i][j]=S((sigma^(i))(v[j]))
    return Vand
#
def UniqueDecodingGabUsingSmithNormalForm(g,y,k,p,nu,m,sigma):
    S=parent(g[0])
    Sx.<X> = S['X',sigma]
    n=len(g)
    t0=floor((n-k)/2)
    A_1=(matrix(S,VandermondeMatrixOf(g,k+t0,sigma))).transpose()
    A_2=(matrix(S,VandermondeMatrixOf(y,t0,sigma))).transpose()
    A=block_matrix([[A_1,A_2]])
    Y=matrix(S,n,1,[(sigma^t0)(y[i]) for i in [0..n-1]])
    [D,P,Q]=SmithNormalFormOf2(A,p,nu)
    Y_2=P*Y
    v_1=[ValuationOf(D[i][i],p,nu) for i in [0..k+2*t0-1]]+[nu for i in [k+2*t0..n-1] ]
    v_2=[ValuationOf(Y_2[i][0],p,nu) for i in [0..n-1]]
    if (v_1<=v_2)==false:
        return 'decoding failure'
    else:
        Y_3=matrix(S,n,1,[(p^(v_2[i]-v_1[i]))*UnitOf(Y_2[i][0],p,nu) for i in [0..n-1]])
        Y_4=Q*Y_3[0:k+2*t0]
        Y_5=list((Y_4.transpose())[0])
        U=Sx(Y_5[0:k+t0])
        V=Sx(X^t0-Sx(Y_5[k+t0:k+2*t0]))
        [f_out,r_out]=LeftDivisionOf(U,V,sigma,m)
        if r_out<>Sx(0):
            return 'decoding failure'
        else:
            return f_out
#
# VI.2. Example
#
p25=5    # the characteristic of the residue field
nu25=2   # the nilpotency index
m25=6    # the degree of Galois extension
n25=5    # the length of Gabidulin code
k25=3    # dimensions of Gabidulin code
t25=1    # the rank of error
R25=Integers(ZZ(p25^nu25))    # base ring
R25z.<z>=R25[]
h25=R25z(HenselLiftOfPrimitivePolynomial(p25,nu25,m25))
S25.<a25>=R25z.quotient(h25)    # Galois extension of base ring
sigma25 = S25.hom([a25^p25])    # a genarator of Galois group
S25x.<X> = S25['X',sigma25]    # skew polynomial ring
g25=[S25(a25^i) for i in [0..n25-1]]    # the support of Gabidulin code
f25=S25x.random_element(degree=k25-1)
c25=[f25.operator_eval(g25[i]) for i in [0..n25-1]]
A25=random_matrix(R25,m25,t25)
B25=random_matrix(R25,t25,n25)
E25=A25*B25
e25=VectorRepresentationOf(E25,S25)
y25=[(c25[i]+e25[i]) for i in [0..n25-1]]
f25_out25=UniqueDecodingGabUsingSmithNormalForm(g25,y25,k25,p25,nu25,m25,sigma25)
S25x(f25_out25)==S25x(f25)
```

```
True
```

```
#
# VII. Computing a Grobner basis
#
# Recall that the ring `Z_n` is isomorphic to the product of integer rings modulo a power
# of a prime number. The linear equation is easy to solve in the finite chain rings.
```

```
# Thus, in this section, we will show how to compute a Grobner basis of the key equation
# in the Galois extension of `Z_ {p ^ nu}. To obtain a Grobner basis in the Galois extension
# of `Z_n`, one can use the "strong join" method described in (Norton et al., 2002)
# Assume that  `R` is the ring ` Z_ {p ^nu} `.
# Then, the set of associated relation classes of `S = R [a]` is
# `[S] = {0,1, p, p ^2, ..., p ^{nu-1}}`.
# For `0<=r<=ell` and `p ^{i}` is in `[S] ^{*}`, the pair `(r, p ^{i})`
# used to index the vector in the Grobner bases is replaced by `j = r * nu+ i`.
# Note that in this case, `r` is the quotient and` i` is the remainder
# of the Euclidean division of `j` by` nu`.
# The following algorithm is similar to that of
# (Byrne and Fitzpatrick 2002, algorithm VI.5)
#
def GrobnerBasis(g,y,k,p,nu,m,sigma):
    """
    Input: `g`` a list of the supports of Gabidulin codes
    `y` a received word of the interleaved Gabidulin code
    `k=[1,k^{(1)},...,k^{(\ell)}]`a list of the dimensions of Gabidulin codes
    Output: a Grobner basis of the key equation
    """
    S=parent(g[0][0])
    Sx.<X> = S['X',sigma]
    ell=len(g)
    n=[len(g[l]) for l in [0..ell-1]]
    V=[[Sx(0) for l in [0..ell]] for j in [0..nu*(ell+1)-1]]
    def WeightOrderOf(V,i,j,nu,k):
        l1=i//nu
        l2=j//nu
        w1=Sx(V[i][l1]).degree()-k[l1]
        w2=Sx(V[j][l2]).degree()-k[l2]
        if w1 < w2:
            return true
        else :
            if w1==w2 and l1 > l2:
                return true
            else:
                return false
    for j in [0..nu*(ell+1)-1]:
        V[j][j//nu]=Sx(p^(j%nu))
    for l in [1..ell]:
        for i in [0..n[l-1]-1]:
            W=[[Sx(0) for r in [0..ell]] for j in [0..nu*(ell+1)-1]]
            D=[S(0) for j in [0..nu*(ell+1)-1]]
            for j in [0..nu*(ell+1)-1]:
                D[j]=Sx(V[j][0]).operator_eval(S(y[l-1][i]))-Sx(V[j][l]).operator_eval(S(g[l-1][i]))
            for j in [0..nu*(ell+1)-1]:
                update=false
                if D[j]==S(0):
                    W[j]=[V[j][b] for b in [0..ell]]
                    update=true
                    continue
                t=0
                while ZZ(t)<=ZZ(nu*(ell+1)-1) and update==false :
                    vt=ValuationOf(D[t],p,nu)
                    vj=ValuationOf(D[j],p,nu)
                    if vt<=vj and WeightOrderOf(V,t,j,nu,k):
                        ut=UnitOf(D[t],p,nu)
                        uj=UnitOf(D[j],p,nu)
                        for b in [0..ell]:
                            W[j][b]=Sx(ut*(V[j][b])-(p^(vj-vt))*uj*(V[t][b]))
                        update=true
                        break
                    t=t+1
                if update==false:
                    W[j]=[Sx((UnitOf(D[j],p,nu)*X-sigma(UnitOf(D[j],p,nu)))*(V[j][b])) for b in [0..ell]]
            V=W
    V[0]=[InverseOf(Sx(V[0][0]).leading_coefficient())*(V[0][b]) for b in [0..ell]]
    return V
#
# VII.2. Example
#
g9=[[S9(1), a9, a9^2],[a9^3,a9^5]]
```

```
y9=[[1+2*a9,a9, a9^2],[3*a9^3,a9^5]]
k9=[1,2,1]
V9=GrobnerBasis(g9,y9,k9,p9,nu9,m9,sigma9)
ell9=len(g9)
for j in [0..nu9*(ell9+1)-1]:
    print V9[j]
```

```
[X + 8*a9^2 + 8*a9 + 6, (2*a9^2 + 2*a9 + 2)*X^2 + (8*a9^2 + 6*a9 + 8)*X + 5*a9^2 + 3*a9 + 6, 3]
[(3*a9^2 + 6*a9 + 3)*X + 6*a9^2, (3*a9^2 + 3*a9 + 6)*X^2 + 3*X + 3*a9^2 + 6*a9 + 6, 0]
[5*a9^2 + 8*a9 + 1, (3*a9^2 + 7*a9 + 5)*X^2 + (a9 + 4)*X + 6*a9^2, (4*a9^2 + 7*a9 + 8)*X + 7*a9^2 + 5*a9 + 6
[6*a9^2 + 6, 6*X^2 + (6*a9^2 + 6*a9 + 6)*X, (3*a9^2 + 3)*X + 6*a9]
[0, 0, (8*a9^2 + 7*a9 + 3)*X^2 + (3*a9^2 + 7*a9 + 2)*X + a9^2 + 2*a9 + 8]
[0, 0, (6*a9^2 + 3*a9)*X^2 + (3*a9 + 6)*X + 3*a9^2 + 6*a9 + 6]
```

```
#
# VIII. Unique decoding beyond the error correction capability
#
# VIII.1. Program
#
def UniqueDecodingIGabUsingGrobnerBasis(g,y,k,p,nu,m,sigma):
    """
    Input: `g` a list of the supports of Gabidulin codes
    `y` a received word of the interleaved Gabidulin code
    `k=k=[1,k^{(1)},...,k^{(\ell)}]` a list of the dimensions of Gabidulin codes
    Output: "decoding failure" or the element `\mathbf{\hat{f}}` such that
    for every minimal solution, `\mathbf{U}`, of the key equation we have
    `U^{(l)}=U^{(0)}*f^{(l)}`  for `l=1,...,\ell`.
    """
    S=parent(g[0][0])
    Sx.<X> = S['X',sigma]
    ell=len(g)
    n=[len(g[l]) for l in [0..ell-1]]
    t0=min((n[i]-k[i+1])//2 for i in [0..ell-1])
    V=GrobnerBasis(g,y,k,p,nu,m,sigma)
    Alpha=[V[j][j//nu].degree() for j in [0..nu*(ell+1)-1]]
    b1=nu
    while b1<=nu*(ell+1)-1 and Alpha[0]-k[0]< Alpha[b1]-k[b1//nu]:
        b1=b1+1
    if b1<=nu*(ell+1)-1 :
        return 'decoding failure'
    QP=[LeftDivisionOf(V[0][l],V[0][0],sigma,m) for l in [1..ell]]
    b2=0
    while b2<ell and Sx(QP[b2][1])==Sx(0):
        b2=b2+1
    if b2<ell :
        return 'decoding failure'
    else :
        if Alpha[0]<=t0 :
            return [QP[l][0] for l in [0..ell-1]]
        else:
            b3=1
            while b3<nu and [Sx(V[b3][l]) for l in [1..ell]]==[Sx((V[b3][0])*(QP[l-1][0])) for l in [1..ell]
                b3=b3+1
            if b3<nu :
                return 'decoding failure'
            else :
                return [QP[l][0] for l in [0..ell-1]]
#
# VIII.2. Example
#
UniqueDecodingIGabUsingGrobnerBasis(g9,y9,k9,p9,nu9,m9,sigma9)
```

```
'decoding failure'
```

```
#
# VII.3. Example
# The following example is given in our manuscript.
#
m4=4
nu2=2
```

```
S4.<a4>=R4z.quotient(h4)
sigma4 = S4.hom([a4^p4])
S4x.<X> = S4['X',sigma4]
g4_1=[S4(1), a4, a4^2,a4^3]
g4_2=[S4(1), a4, a4^2,a4^3]
y4_1=[3*a4^3+2*a4^2+2,a4^2+2*a4,a4^3+2,2*a4^3+2*a4^2+3*a4+3]
y4_2=[a4^2+2*a4+3,2*a4^3+a4^2+2*a4+3,a4^3+a4^2+2*a4+3,2*a4^3+3]
k4=[1,1,1]
g4=[g4_1,g4_2]
y4=[y4_1,y4_2]
[f4_1,f4_2]=UniqueDecodingIGabUsingGrobnerBasis(g4,y4,k4,p4,nu4,m4,sigma4)
e4_1=[S4(y4_1[i]-f4_1.operator_eval(g4_1[i])) for i in [0..m4-1]]
e4_2=[S4(y4_2[i]-f4_2.operator_eval(g4_2[i])) for i in [0..m4-1]]
e4=e4_1+e4_2
E4=MatrixRepresentationOf(e4)
print  "h4","=", h4
""
print  "f4_1","=", f4_1
""
print  "f4_2","=", f4_2
""
print "e4","=", e4
""
print  "RankOf(e4)","=", RankOf(E4)
""
print E4
```

```
h4 = z^4 + 2*z^2 + 3*z + 1

''

f4_1 = 2*a4^3 + 3*a4

''

f4_2 = 3*a4^2 + 2*a4 + 1

''

e4 = [a4^3 + 2*a4^2 + a4 + 2, 2*a4^2 + 2, 2*a4^3 + 2*a4^2 + 2*a4 + 2, 2*a4^2 + 2, 2*a4^2 + 2, 3*a4^3 + 3*a4^
+ a4 + 3, 3*a4^3 + 2*a4^2 + 3*a4 + 2, 3*a4^3 + a4^2 + a4 + 1]

''

RankOf(e4) = 2

''

[2 2 2 2 2 3 2 1]
[1 0 2 0 0 1 3 1]
[2 2 2 2 2 3 2 1]
[1 0 2 0 0 3 3 3]
```

```
# VIII.3. Failure probability of unique decoding interleaved Gabidulin codes
#
# We give Failure probability  of above example
#
n4=4
ell4=2
t4=2 # the rank of error
k4_b=1
def FailureProbability2(N4):
    """
    Input: `N4` number of simulations
    Output: `N4_1/N4` where N4_1 is the number of "decoding failure".
    """
    N4_1=0
    for j in [0..N4-1]:
        f4=[S4x.random_element(degree=k4_b-1) for _ in [0..ell4-1]]
        c4=[[f4[l].operator_eval(g4_1[i]) for i in [0..n4-1]] for l in [0..ell4-1]]
        A4=random_matrix(R4,m4,t4)
        B4=random_matrix(R4,t4,ell4*n4)
        E4_b=A4*B4
        t4_b=RankOf2(matrix(S4,E4_b),p4,nu4)
```

```
        while t4_b<>t4:
            A4=random_matrix(R4,m4,t4)
            B4=random_matrix(R4,t4,ell4*n4)
            E4_b=A4*B4
            t4_b=RankOf2(matrix(S4,E4_b),p4,nu4)
        e4_b=[matrix(S4,[[a4^i for i in [0..m4-1]]])*matrix(S4,E4_b[:,n4*l:n4*(l+1)]) for l in [0..ell4-1] ]
        y4_b=[[S4(c4[l][i]+e4_b[l][0][i]) for i in [0..n4-1]] for l in [0..ell4-1]]
        f4_out=UniqueDecodingIGabUsingGrobnerBasis(g4,y4_b,k4,p4,nu4,m4,sigma4)
        if f4_out=='decoding failure':
            N4_1=N4_1+1
    N4_2=RR(N4_1/N4)
    return N4_2
#
N4=100
FailureProbability2(N4)
```

```
0.0800000000000000
```

```
# IX. Comparison of unique decoding interleaved Gabidulin codes
#
# We compare our decoding algorithm of interleaved Gabidulin codes
# to the decoding algorithm of [Sidorenko et al., 2011]
# in the case of finite fields.
#
# IX.1. Unique decoding interleaved Gabidulin codes using skew-feedback shift register synthesis
#
# We implement the decoding algorithm of interleaved Gabidulin codes
# of [Sidorenko et al., 2011]
#
def SkewFeedbackShiftRegisterSynthesisOf3(s,sigma):
    S=parent(s[0][0]) # finite  field
    Sx.<X> = S['X',sigma] # Skew Polynomial ring
    L=len(s) # number of sequences
    Nl=[len(s[l]) for l in [0..L-1]] # length of sequences
    N=max(Nl)  #  maximum length of sequences
    u=[N-Nl[l] for l in [0..L-1]]
    v=[Sx(1),0]   # initialization of connection polynomial and the shift register length
    b=[[Sx(0),0,u[l]] for l in [0..L-1] ]  # initialization  of auxiliary variables
    dl=[S(1) for l in [0..L-1]] # initialization  of discrepancy
    for n in [1..N]:
        for l in [0..L-1]:
            if n>v[1]+u[l] :
                d=S(sum([Sx(v[0])[j]*((sigma^j)(s[l][n-1-j-u[l]])) for j in [0..v[1]]]))
                if S(d)<>S(0):
                    if n-v[1]<=b[l][2]-b[l][1]:
                        v[0]=Sx(v[0]-d*(X^(n-b[l][2]))*(dl[l]^-1)*b[l][0])
                    else :
                        b0=v[0]
                        b1=v[1]
                        v[0]=Sx(v[0]-d*(X^(n-b[l][2]))*(dl[l]^-1)*b[l][0])
                        v[1]=b[l][1]+n-b[l][2]
                        b[l]=[Sx(b0),b1,n]
                        dl[l]=d
    return [v]+[b]
#
def ParityCheckMatrixOf(g,k,m,sigma):
    S=parent(g[0])
    n=len(g)
    G_0=VandermondeMatrixOf(g,n,sigma)
    G_1=matrix(S,G_0)
    H_1=G_1^-1
    h=[S((sigma^(m-n+k+1))(H_1[i,n-1])) for i in [0..n-1]]
    H=VandermondeMatrixOf(h,n-k,sigma)
    return H
#
def ErrorLocationErrorValueDecoding(h,y,k,m,sigma):
    """
    Input: `y` a received word of the interleaved Gabidulin code
    `h` the first row of a parity check matrix of Gabidulin code
    `k` the dimensions of Gabidulin codes
```

```
    `m` the degree of Galois extension
    """
    S=parent(h[0]) # finite  field
    p=S.characteristic()
    a=S.gen()
    Sx.<X> = S['X',sigma] # Skew Polynomial ring
    ell=len(y) # number of sequences
    n=len(h)
    # Compute syndromes
    H=matrix(S,VandermondeMatrixOf(h,n-k,sigma))
    s=[list((matrix(S,[y[l]])*(H.transpose()))[0]) for l in [0..ell-1] ]
    # Compute Shift-Register Synthesis
    LSSR=SkewFeedbackShiftRegisterSynthesisOf3(s,sigma)
    N=n-k
    z=max([0,LSSR[0][1]-N])
    epsilon=sum([max([0,LSSR[1][l][2]-LSSR[1][l][1]-z-(N-LSSR[0][1])]) for l in [0..ell-1]])
    if epsilon <> 0 :
        return 'decoding failure'
    else :
        # Find a basis for the root space of connection polynomial
        Vx=LSSR[0][0]
        t=LSSR[0][1]
        ImVx=[Vx.operator_eval(a^i) for i in [0..m-1]]
        MVx=MatrixRepresentationOf(ImVx)
        KerMVx=MVx.right_kernel()
        tau=KerMVx.dimension()
        if tau<>t:
            return 'decoding failure'
        else:
            if t==0:
                return y
            else:
                BasisKerMVx1=KerMVx.basis()
                BasisKerMVx2=(matrix(GF(p),[list(BasisKerMVx1[i]) for i in [0..tau-1]])).transpose()
                RootSpaceVx=VectorRepresentationOf(BasisKerMVx2,S)
                # Solve '(41)'
                A1=matrix(S,VandermondeMatrixOf(RootSpaceVx,tau,sigma^(m-1)))
                A2=A1^-1
                TranOfs=[matrix(S,tau,1,[(sigma^(m-j))(s[l][j]) for j in [0..tau-1]]) for l in [0..ell-1]]
                F1=[A2*TranOfs[l] for l in [0..ell-1]]
                F2=[list(F1[l].transpose()[0]) for l in [0..ell-1]]
                F3=[MatrixRepresentationOf(F2[l]) for l in [0..ell-1]]
                # Solve '(40)'
                Mh1=MatrixRepresentationOf(h)
                Mh2=block_matrix([[Mh1,identity_matrix(GF(p),m)]])
                Mh3=Mh2.echelon_form()
                Mh4=Mh3[:,n:]
                B1=[Mh4*F3[l] for l in [0..ell-1]]
                B2=[B1[l][n:,:] for l in [0..ell-1]]
                B3=matrix(GF(p),m-n,tau)
                if [B2[l]==B3 for l in [0..ell-1] ]<>[True  for l in [0..ell-1]]:
                    return 'decoding failure'
                else:
                    B5=[(B1[l][:n,:]).transpose() for l in [0..ell-1]]
                    e_out=[list((matrix(S,[RootSpaceVx])*B5[l])[0]) for l in [0..ell-1] ]
                    c_out=[[S(y[l][i]-e_out[l][i]) for i in [0..n-1]] for l in [0..ell-1]]
                    return c_out

# IX.3. Simulation results of Comparison
#
p3=5    # the characteristic of finite field
m3=6    # the degree of Galois extension
k3=2    # dimensions of Gabidulin codes
n3=6    # the length of Gabidulin code
t3=3    # the rank of error
ell3=3  # interleaving order
R3z.<z> = GF(p3)[]
Conway=R3z(conway_polynomial(p3,m3))
S3.<a3>=R3z.quotient(Conway)    # Galois extension of 'GF(P3)'
sigma3 = S3.hom([a3^p3])    # a genarator of Galois group
S3x.<X> =S3['X',sigma3]    # skew polynomial ring
```

```
g3=[a3^i for i in [0..n3-1]] # the support of Gabidulin code
h3=ParityCheckMatrixOf(g3,k3,m3,sigma3)[0]    # the first row of a parity check matrix of Gabidulin code
g3_2=[g3 for l in [0..ell3-1]]
k3_2=[1]+[k3 for l in [0..ell3-1]]
f3=[S3x.random_element(degree=k3-1) for l in [0..ell3-1]]
c3=[[f3[l].operator_eval(g3[i]) for i in [0..n3-1]] for l in [0..ell3-1]]
E3=random_matrix(GF(p3), m3, n3*ell3,algorithm='echelonizable', rank=t3)
e3=[matrix(S3,[[a3^i for i in [0..m3-1]]])*matrix(S3,E3[:,n3*l:n3*(l+1)]) for l in [0..ell3-1] ]
y3=[[S3(c3[l][i]+e3[l][0][i]) for i in [0..n3-1]] for l in [0..ell3-1]]
f3_out=UniqueDecodingIGabUsingGrobnerBasis(g3_2,y3,k3_2,p3,1,m3,sigma3)
c3_out=ErrorLocationErrorValueDecoding(h3,y3,k3,m3,sigma3)
f3_out==f3
c3_out==c3
```

```
True

True
```

```
#
# X. Decoding of random linear network codes
#
# X.1. Program
#
def RedimensionOf(L,mt):
    """
    Input: a matrix `L` with coefficents in the ring `R`
    Output: the matrix of `mt` rows obtained from the matrix `L`
    by inserting all zero rows below the last row if `L.nrows()<=mt`
    or by deleting the `L.nrows()-mt` last rows else,
    where `mt`is the row  size of the transmitted matrix
     """
    R=L.base_ring()
    ar=L.nrows()
    if mt<=ar : L1=L[0:mt,:]
    else:
        L2=matrix(R,mt-ar,L.ncols())
        L1=block_matrix([[L],[L2]])
    return L1
#
def SuccessiveTransformationOf(mt,b0,n,Y):
    """
    Input:The row  size `mt` of the transmitted matrix.
    The column  size`b0` of the zero matrix
    and the column size `n` of a code matrix
    using in the transmitted matrix.
    A received matrix `Y` with coefficents in the ring `R`.
    Output: `[Yh_21,Dh_1,Yh_22]` such that
    `Yh_21=M+Dh_1*W_1+W_2*Yh_22+Eh` where `M` is a code matrix
    """
    R=Y.base_ring()
    # First transformation
    Y_0=Y[:,0:b0]
    a_f0=FreeRankOf(Y_0)
    P_2=SmithNormalFormOf(Y_0)[1][a_f0:,:]
    Y1=P_2*Y[:,b0:]
    # Second transformation
    m1r=Y.nrows()-a_f0
    Y1_1=Y1[:,:mt]
    Y1_2=Y1[:,mt:mt+n]
    a_f1=FreeRankOf(Y1_1)
    a_1=RankOf(Y1_1)
    [D1,P1,Q1]=SmithNormalFormOf(Y1_1)[0:3]
    Y2_2=P1*Y1_2
    # Third transformation
    D1_1=D1[:a_1,:]
    Y2_21=Y2_2[:a_1,:]
    Y2_22=Y2_2[a_1:,:]
    a_f22=FreeRankOf(Y2_22)
    if a_f22==0:
        Yh_22=matrix(R,1,n)
    else :
        Yh_22=SmithNormalFormOf(Y2_22)[1][:a_f22,:]*Y2_22
```

```
    D2_1=RedimensionOf(D1_1,mt)
    Y3_21=RedimensionOf(Y2_21,mt)
    Dh_1=Q1*(D2_1-identity_matrix(mt,mt))
    Yh_21=Q1*Y3_21
    return [Yh_21,Dh_1,Yh_22]
#
# X.2. Example
#
R30=Integers(30)
n30=12
mt30=7
br30=3
b030=3
mr30=10
M30=matrix(R30,mt30,n30)
Xt30=block_matrix([[matrix(R30,mt30,b030),identity_matrix(R30,mt30),M30]])
A30=random_matrix(R30,mr30,mt30)
B30=random_matrix(R30,mr30,br30)
Z30=random_matrix(R30,br30,b030+mt30+n30)
Y30=A30*Xt30+B30*Z30
T30=SuccessiveTransformationOf(mt30,b030,n30,Y30)
Yh30_21=T30[0]
Dh30_1=T30[1]
Yh30_22=T30[2]

print Xt30
""
print A30
""
print B30
""
print Z30
""
print Y30
""
print Yh30_21
""
print Dh30_1
""
print Yh30_22
```

```
[0 0 0|1 0 0 0 0 0 0|0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0|0 1 0 0 0 0 0|0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0|0 0 1 0 0 0 0|0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0|0 0 0 1 0 0 0|0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0|0 0 0 0 1 0 0|0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0|0 0 0 0 0 1 0|0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0|0 0 0 0 0 0 1|0 0 0 0 0 0 0 0 0 0 0 0]
''

[22 15 15  5  6 17 17]
[ 8 26  0 16 12 27 16]
[22 20  1  1 13 22 12]
[24  0 23 11 23  3  2]
[ 5 26  2 23 26  7 25]
[ 4 25  7 23 20  8 26]
[14 14  3 15  0  2 21]
[18 13 27  9  9 23 13]
[ 2  9  4  3 29 23 25]
[ 0 18 19 12 26 18 11]
''

[ 1  2 24]
[ 7 15  0]
[18 28 17]
[ 3 11  0]
[14 17  0]
[22  3 16]
[ 1 13  0]
```

```
[27 10 14]
[ 9 16 16]
[15 25 26]
''

[11 22  7  0 23 10  6 29 27 13  3 22 23 20  4 18 25  2 24 14  1  2]
[ 0  2  2  2  5 15 15 22  2 28  2  7 14  5  5  2 26 28 26 21  2 26]
[15  8  0 27  9  7  8  1 12 15 14  7 23 25 25 17 29 15 13 26  7 20]
''

[11  8 11 14 24 13 23 13  6 26 13 24  3  0 14 10 23 28 28 20 23 24]
[17  4 19  8 22 25 13  5  6 17 21 19 11  5 13  6 25 14 18 23  7 14]
[ 3 18  2 27  7  0  5 28 18 25 18 21 27 25  7  9 21 25  1 22 13 24]
[ 3 28 13 16  4  8 14 22 16 19  1 23 13 25  7 16  1 14 28  3 25 22]
[ 4 12 12  9 13  7  2 26 29 23 16  7 20  5 21 16 12 24 28 13 18 20]
[ 2 18 10 22  0 24 28 20 20  6 26 17 16 15 23 14 12  8  4  7 20 22]
[11 18  3 10 12 28  6 15 25  8 29 23 25 25  9 14  3  6  2 17 27 10]
[27  6 29 26  0  5 13  6 10 14 27 12  3 10 28 24 21  4 10 22 25 24]
[ 9 28  5 16 20 26  5 28 10 20 13  2 19  0  6 16 25 16  0  8  3  4]
[15 18  5  2  2  6 25 17  5  6  9 27  3 25 25 12  9 10 28  1  7  0]
''

[22 19  2 15  1 28 24 16  8 29 10 14]
[22 19  2 15  1 28 24 16  8 29 10 14]
[14 23  4 15 17 26 18  2 16 13 20 28]
[10 10 20  0 10 10  0 10 20 20 10 20]
[ 6 27  6 15  3 24 12 18 24 27  0 12]
[ 6 27  6 15  3 24 12 18 24 27  0 12]
[ 8 11 28 15 29  2  6 14 22  1 20 16]
''

[ 0  0  0  0  0  0 15]
[ 0  0  0  0  0  0  5]
[ 0  0  0  0  0  0  0]
[ 0  0  0  0  0  0  0]
[ 0  0  0  0  0  0 20]
[ 0  0  0  0  0  0 10]
[ 0  0  0  0  0  0  0]
''

[0 0 0 0 0 0 0 0 0 0 0 0]
```

```
#
# X.3. Example
# The following example is given in our manuscript.
#
R8=Integers(8)
p8=2
nu8=3
m8=5
R8z.<z>=R8[]
h8=R8z(z^5+4*z^3+7*z^2+2*z+7)
S8.<a8>=R8z.quotient(h8)
sigma8 = S8.hom([a8^p8])
S8x.<X> = S8['X',sigma8]
n8_1=5
n8_2=5
gt8_1=[a8^i for i in [1..n8_1]]
gt8_2=[a8^i for i in [1..n8_2]]
f8_1=S8x(1+2*a8+3*a8^2+5*a8^3)
f8_2=S8x(1+4*a8+7*a8^2+2*a8^3+5*a8^4)
c8_1=[f8_1.operator_eval(gt8_1[i]) for i in [0..n8_1-1]]
c8_2=[f8_2.operator_eval(gt8_2[i]) for i in [0..n8_2-1]]
M8_1=MatrixRepresentationOf(c8_1)
M8_2=MatrixRepresentationOf(c8_2)
M8=block_matrix([[M8_1,M8_2]])
n8=n8_1+n8_2
```

```
mt8=M8.nrows()
b08=2
Xt8=block_matrix([[matrix(R8,mt8,b08),identity_matrix(R8,mt8),M8]])
br8=3
mr8=7
A8=matrix(R8,[
[5, 6, 6, 3,3],
[3, 2, 7, 1, 0],
[4, 6, 0, 6, 7],
[4, 1, 2, 1, 0],
[1, 4, 5, 6, 2],
[2, 5, 7, 5, 0],
[4, 4, 1, 3, 1]
])
B8=matrix(R8,[
[6, 4, 2],
[4, 5, 5],
[2, 5, 4],
[6, 7, 6],
[3, 7, 2],
[2, 7, 1],
[6, 0, 7]
])
Z8=matrix(R8,[
[0, 7, 7, 0, 6, 3, 3, 1, 5, 2, 6, 7, 4, 3, 4, 1, 2],
[0, 0, 7, 5, 2, 4, 5, 2, 3, 0, 3, 0, 4, 5, 5, 6, 5],
[6, 3, 0, 5, 5, 7, 2, 3, 7, 0, 4, 3, 5, 1, 5, 2, 5]
])
Y8=A8*Xt8+B8*Z8
#
# Successive transformations
#
T8=SuccessiveTransformationOf(mt8,b08,n8,Y8)
Yh8_21=T8[0]
Dh8_1=T8[1]
Yh8_22=T8[2]
#view("Xt8","=",Xt8)
#view("Y8","=",Y8)
#view("Yh8_21","=",Yh8_21)
#view("Dh8_1","=",Dh8_1)
#view("Yh8_22","=",Yh8_22)
print Yh8_21
""
print Dh8_1
""
print Yh8_22
```

```
[0 6 5 4 5 7 3 6 4 4]
[5 7 5 1 3 5 6 7 4 6]
[0 2 4 7 3 5 2 1 0 3]
[7 1 7 3 5 7 5 1 2 1]
[5 7 3 6 4 0 2 2 0 1]
''

[0 0 0 0 4]
[0 0 0 0 6]
[0 0 0 0 4]
[0 0 0 0 7]
[0 0 0 0 7]
''

[0 7 6 2 1 6 7 5 5 1]
```

```
#
# Error-Erasure Decoding
#
SNFh8_22_1=SmithNormalFormOf(Yh8_22[:,0:n8_1])
Fh8_22_1=SNFh8_22_1[2][:,SNFh8_22_1[3]:]
F8c_1=Fh8_22_1
SNFh8_22_2=SmithNormalFormOf(Yh8_22[:,n8_1:n8])
```

```
Fh8_22_2=SNFh8_22_2[2][:,SNFh8_22_2[3]:]
F8c_2=Fh8_22_2
ah8_1=RankOf(Dh8_1)
SNFh8_1=SmithNormalFormOf(Dh8_1)
vh8_1=VectorRepresentationOf(( SNFh8_1[1]^-1)[:,:ah8_1],S8)
Pr8=MinimalSkewPolynomialOf(vh8_1,sigma8)
print F8c_1
" "
print F8c_2
" "
print Pr8

```

```
[0 0 0 1]
[7 6 2 0]
[1 2 7 0]
[0 1 0 0]
[1 0 0 0]
' '
[1 5 5 1]
[7 3 3 6]
[0 0 1 0]
[0 1 0 0]
[1 0 0 0]
' '
X + 5*a8^4 + a8^3 + 6*a8^2 + 2*a8 + 2
```

```
g8_new_1=matrix(S8,1,n8_1,[gt8_1])*F8c_1
g8_new_2=matrix(S8,1,n8_2,[gt8_2])*F8c_2
g8_new=[list(g8_new_1[0]), list(g8_new_2[0])]
yh8_21_1=VectorRepresentationOf(Yh8_21[:,:n8_1],S8)
yh8_21_2=VectorRepresentationOf(Yh8_21[:,n8_1:n8_1+n8_2],S8)
y8_new_1=matrix(S8,1,n8_1,[Pr8.operator_eval(yh8_21_1[i]) for i in [0..n8_1-1]])*F8c_1
y8_new_2=matrix(S8,1,n8_2,[Pr8.operator_eval(yh8_21_2[i]) for i in [0..n8_2-1]])*F8c_2
y8_new=[list(y8_new_1[0]), list(y8_new_2[0])]
k8_new=[1,1+Pr8.degree(),1+Pr8.degree()]
Out8=UniqueDecodingIGabUsingGrobnerBasis(g8_new,y8_new,k8_new,p8,nu8,m8,sigma8)
Out8
```

```
[(7*a8^4 + 5*a8^3 + 5*a8 + 1)*X + 4*a8^4 + 3*a8^3 + 4*a8 + 1, (5*a8^4 + 7*a8^3 + 5*a8^2 + 4*a8 + 6)*X + 2*a8
+ 5*a8^3 + 3*a8^2 + 5*a8]
```

```
print LeftDivisionOf(Out8[0],Pr8,sigma8,m8)
print LeftDivisionOf(Out8[1],Pr8,sigma8,m8)
print LeftDivisionOf(Out8[0],Pr8,sigma8,m8)[0]==f8_1
print LeftDivisionOf(Out8[1],Pr8,sigma8,m8)[0]==f8_2
```

```
[5*a8^3 + 3*a8^2 + 2*a8 + 1, 0]

[5*a8^4 + 2*a8^3 + 7*a8^2 + 4*a8 + 1, 0]

True

True
```

```
#
# X.4. Example
# In this example, the matrices A, B, Z are random.
#
R32=Integers(32)
p32=2
nu32=5
m32=8
R32z.<z>=R32[]
h32=R32z(HenselLiftOfPrimitivePolynomial(p32,nu32,m32))
S32.<a32>=R32z.quotient(h32)
sigma32 = S32.hom([a32^p32])
S32x.<X> = S32['X',sigma32]
```

```
n32_1=8
n32_2=8
n32_3=8
k32_1=2
k32_2=2
k32_3=2
gt32_1=[a32^i for i in [1..n32_1]]
gt32_2=[a32^i for i in [1..n32_2]]
gt32_3=[a32^i for i in [1..n32_3]]
f32_1=S32x.random_element(degree=k32_1-1)
f32_2=S32x.random_element(degree=k32_2-1)
f32_3=S32x.random_element(degree=k32_3-1)
c32_1=[f32_1.operator_eval(gt32_1[i]) for i in [0..n32_1-1]]
c32_2=[f32_2.operator_eval(gt32_2[i]) for i in [0..n32_2-1]]
c32_3=[f32_3.operator_eval(gt32_3[i]) for i in [0..n32_3-1]]
M32_1=MatrixRepresentationOf(c32_1)
M32_2=MatrixRepresentationOf(c32_2)
M32_3=MatrixRepresentationOf(c32_3)
M32=block_matrix([[M32_1,M32_2,M32_3]])
n32=n32_1+n32_2+n32_3
mt32=M32.nrows()
b032=4
br32=7
mr32=12
Xt32=block_matrix([[matrix(R32,mt32,b032),identity_matrix(R32,mt32),M32]])
A32=random_matrix(R32,mr32,mt32)
B32=random_matrix(R32,mr32,br32)
Z32=random_matrix(R32,br32,b032+mt32+n32)
Y32=A32*Xt32+B32*Z32
print f32_1
""
print f32_2
""
print f32_3
""
print Xt32
""
print A32
""
print B32
""
print Z32
""
print Y32
```

```
(22*a32^7 + 30*a32^6 + 18*a32^5 + 13*a32^4 + 13*a32^3 + 31*a32^2 + a32 + 26)*X + 9*a32^7 + 26*a32^6 + 24*a32
+ 19*a32^4 + 28*a32^3 + 22*a32^2 + 13

''

(28*a32^7 + 4*a32^6 + 29*a32^5 + 17*a32^4 + 6*a32^3 + 11*a32^2 + 10*a32 + 5)*X + 19*a32^7 + 28*a32^6 +
17*a32^5 + 4*a32^3 + 24*a32^2 + 14*a32

''

(14*a32^7 + 18*a32^6 + 7*a32^5 + 6*a32^4 + 24*a32^3 + 31*a32^2 + 19*a32 + 25)*X + 6*a32^7 + 9*a32^6 + 20*a32
+ 2*a32^4 + 31*a32^3 + 9*a32 + 9

''

[ 0  0  0  0| 1  0  0  0  0  0  0  0|17 21 13 16 14  7 26 13 25  3 26 15  1 17 12  1  0 13  7 20  2 24  1 16
[ 0  0  0  0| 0  1  0  0  0  0  0  0| 9 19 29 11 21 14 30  0  6 14 16 18  0 20 23  8 27 17 17  6 11 29 28 24
[ 0  0  0  0| 0  0  1  0  0  0  0  0|25  0 23 19 15 23 25 17 30 13 22 26 21 27 21 22  6 12 11  2 24 15 19  2
[ 0  0  0  0| 0  0  0  1  0  0  0  0|10 26  5  3 26 19 10 25 27 17  1 10 18 16  4  6 25 21 26  0 16 24 19 18
[ 0  0  0  0| 0  0  0  0  1  0  0  0| 8 22 18 30 24 22  2  6 22  2 10 29 19 20  5 14 24  1 29  3 28 13  6  9
[ 0  0  0  0| 0  0  0  0  0  1  0  0|20 30 19  2 25 14  7  1  8 16 13 25 19 29  1 22  0 17 16 16 15 23  4 16
[ 0  0  0  0| 0  0  0  0  0  0  1  0|11 10 28 30  5 17 16 22 20 17  8  5 18 11 27  8  6  9 15 26  5 25 31 25
[ 0  0  0  0| 0  0  0  0  0  0  0  1|20 29 24 19 16  2 13 25  5  7 11 29 16  4 17  8  4 16 18 20 31 23  6  6

''

[ 5 27 29 19 31 24 26 27]
[31  5 21  1  4  5 27 27]
```

```
[18 25  7 25  3 29  4 15]
[ 8 19 23  1 20 21 15 24]
[ 4  3 31  1  6 29 28  3]
[ 3 12 20 20 26  8  4 15]
[14  1  9 27  7 20 15  2]
[31 20 21 18 14 14  9 20]
[11 24 22 20 10  4 12  9]
[ 2 15 21 31  2 15 25 29]
[18  8  0 30 19  7  8  9]
[20 31 14 15 14 10 31 27]
''

[29 29  0 20 26 10 18]
[ 3 29 16 19  9  8 28]
[ 2 20  3 14 25 13 14]
[17 18 20 18  2  0  5]
[25 29 23 28  4 26 12]
[13 29  9  0 26 27  8]
[ 8 22 15 25 16  0 21]
[18 15 13 28 16  6 23]
[ 7 15 13 28  2 15 30]
[10  6 25 18 16  8 29]
[ 4 30  8 20  2 12  2]
[ 6  0 10 31 13 11 22]
''

[ 6  3  4 10 23 10  1 10 16 31 25  2  3  9 26 24 17  9 31 17 28 25  6 12 21  9  1  6 25 10 15 27 20 15  1 31
[ 4 22 19  9  6 28 14 31 15 10 26 18  9 10  6 12 27  2 29  9 27  7  7  5 24  1 24  8 14 30 12  6 27 11 12  5
[ 4 30 25 10  4 28  5 10  8 18 15 13 12 17  6 16  8  8 30  3 24  5  9 24 15 26 25 31 14 28  2  3 19 28 24 21
[ 7 10 17 20  6 30 30  9  2 20 26 23  0  0 24 30  2 29 22 10  4 24  5  2 30 17 26 23  4 30  6 29 22  0 24 13
[ 8  2  2 31 18 22 25 10 31 22 28 21  7 11 26 22 19 22 29 17 22 17 17 24 15 16 23 22 16 17  4 24  6 20  8 19
[ 4 16  9 18 12 15 23 17  5 10 19 15  0  6 21 19  9 12 19 28 28 31 12  5 20  5 29 26 16 10  0  3 22 11  6 28
[18  2 21  2 11 15 13  4 30 16 24 12  2 29 20 13  7  1  2  7 25 24 24  7 28 29  3 28 25 30 15 10 28  4 26 22
''

[10 21 23 21 24  1  2  2 30 13 15 27  3 16 30  8 26 20 28  4  0 26  9  2 12 16 29 17 15  2  7  0  9  4  4  5
[11 15 12 30 26 11 29 31 20 22 10 21  1 27 21 14  3 18 10  6 10  4 20 29  3 15  5 22  5  4 17  8 23 11 23  7
[ 2  2 10 11 16 16  6  4 15 25 26 28 25  6 19 19  1 10 23 18 22  4  1 14 29  9  0 21  4 11 13 17  8 14 22 21
[ 6 25 13  4 18 24 23 11 10  4 28  6  2 15  6 14 30 11 23 15 30 18  0 31 29 13  8 22  0 30 21 17 17 15 28  6
[10 19 20 29 25  7 31 18 31 16 14 24 12 14  0 14 21 25 21 11 29  0 17 27 28  4  0 13 13 11  3 30  9  2 23 17
[18  7 11 29 16 31  3  2  2  9 27 31  1 10  8  6  3 18 14  5  9  6 19 10 31  2  4 14  5 25 18  7 11  6  0 11
[13  2 27 26  3  6 31  0  1 26 22 12  7 13  1  0 12 11 21  4  6  0 11 27  3 16 30 28 29  8 19  3  9 10 31 30
[22 12 31 23 24 19  7 23 15 24  6 17 22 21 28 20 14 22 16  7 19 23  2  6 11 15 22 20 11 18 10 29  9 25  7  9
[ 6 13 27  7  4  5  9 20  8 15  1  5 27 16 18  0 15 20 14 23 11 30 11 28 11 30  4  6  1 21  0 21 19  2  8  7
[ 0 30  6 22 27 30 13 21  6 11 10  4  9  6 26 21  7  6 15 22 31 17 24  0 15  4 29  5 29  8 20 17 23  5 27  6
[ 0  0  0 16  4 14  8 20 19 19  4 23  2 23 31 31 21  8  8 20 27  5  6 21 22 17 19 26 28  4 21  9 14 17  0 11
[ 5 10 12 17  0  2  8 27 26 16 30 16 25 11 11  6 24  3 21 25 10 29 20 21 11 19 13 17 10 10 14 10  3 20  2 11
```

```
#
# Successive transformations
#
T32=SuccessiveTransformationOf(mt32,b032,n32,Y32)
Yh32_21=T32[0]
Dh32_1=T32[1]
Yh32_22=T32[2]
print Yh32_21
""
print Dh32_1
""
print Yh32_22
```

```
[ 7 30 17 10 31  2 15  3 20  2 21  9  7 29  8 11 31 11  3 13 12 17 11 31]
[20 29 12 12 10  6  2 18  8  9 12 10  8  4 30 13 16 16 26 29  2 17 20 14]
[ 6 16 24  0  1 22 28  9  3 30  8  6 19 12 23 13 16 13 19 31 26 12 16  6]
[ 7 21  1 22  3 30 25 15 26 11 13 28  6 13 13 20 23 30 28 31 10 21 25  2]
```

```
[14  8 20 12 22 26 18 18 22  2 28  9 11 22  1 24 10 31 13 23 28  9  4 23]
[31 23  4  9  4 28 27 15 20 16 30 31 23  2  3 11 29 24  4  2 27 25 31 25]
[16 31 17 11  4  7  0 12  8 19 19 27 18  2 27  1 13  2  3 28 29 11 10 26]
[13 20  5 20 29 26 19  3 27  7  4  7  4 23 29  5 13 13 10  0 23 15 17 13]
''
[ 0  0  0  0  0  0  0 21]
[ 0  0  0  0  0  0  0 14]
[ 0  0  0  0  0  0  0 16]
[ 0  0  0  0  0  0  0 18]
[ 0  0  0  0  0  0  0 18]
[ 0  0  0  0  0  0  0 17]
[ 0  0  0  0  0  0  0 27]
[ 0  0  0  0  0  0  0 13]
''
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
```

```
#
# Error-Erasure Decoding
#
SNFh32_22_1=SmithNormalFormOf(Yh32_22[:,0:n32_1])
Fh32_22_1=SNFh32_22_1[2][:,SNFh32_22_1[3]:]
if Fh32_22_1==matrix(R32,Fh32_22_1.nrows(),Fh32_22_1.ncols()):
     F32c_1=identity_matrix(R32,n32_1)
else:
    F32c_1=Fh32_22_1
#
SNFh32_22_2=SmithNormalFormOf(Yh32_22[:,n32_1:n32_1+n32_2])
Fh32_22_2=SNFh32_22_2[2][:,SNFh32_22_2[3]:]
if Fh32_22_2==matrix(R32,Fh32_22_2.nrows(),Fh32_22_2.ncols()):
     F32c_2=identity_matrix(R32,n32_2)
else:
    F32c_2=Fh32_22_2
#
SNFh32_22_3=SmithNormalFormOf(Yh32_22[:,n32_1+n32_2:n32_1+n32_2+n32_3])
Fh32_22_3=SNFh32_22_3[2][:,SNFh32_22_3[3]:]
if Fh32_22_3==matrix(R32,Fh32_22_3.nrows(),Fh32_22_3.ncols()):
     F32c_3=identity_matrix(R32,n32_3)
else:
    F32c_3=Fh32_22_3
#
ah32_1=RankOf(Dh32_1)
if ah32_1 ==0:
    Pr32=S32x(1)
else:
    SNFh32_1=SmithNormalFormOf(Dh32_1)
    vh32_1=VectorRepresentationOf(( SNFh32_1[1]^-1)[:,:ah32_1],S32)
    Pr32=MinimalSkewPolynomialOf(vh32_1,sigma32)
#
print F32c_1
" "
print F32c_2
" "
print F32c_3
" "
print Pr32
```

```
[0 0 0 0 0 0 0 1]
[0 0 0 0 0 0 1 0]
[0 0 0 0 0 1 0 0]
[0 0 0 0 1 0 0 0]
[0 0 0 1 0 0 0 0]
[0 0 1 0 0 0 0 0]
[0 1 0 0 0 0 0 0]
[1 0 0 0 0 0 0 0]
' '
```

```
[0 0 0 0 0 0 0 1]
[0 0 0 0 0 0 1 0]
[0 0 0 0 0 1 0 0]
[0 0 0 0 1 0 0 0]
[0 0 0 1 0 0 0 0]
[0 0 1 0 0 0 0 0]
[0 1 0 0 0 0 0 0]
[1 0 0 0 0 0 0 0]
' '

[0 0 0 0 0 0 0 1]
[0 0 0 0 0 0 1 0]
[0 0 0 0 0 1 0 0]
[0 0 0 0 1 0 0 0]
[0 0 0 1 0 0 0 0]
[0 0 1 0 0 0 0 0]
[0 1 0 0 0 0 0 0]
[1 0 0 0 0 0 0 0]
' '

X + 9*a32^7 + 3*a32^6 + 31*a32^5 + 12*a32^3 + 8*a32^2 + 2*a32 + 7
```

```
g32_new_1=matrix(S32,1,n32_1,[gt32_1])*F32c_1
g32_new_2=matrix(S32,1,n32_2,[gt32_2])*F32c_2
g32_new_3=matrix(S32,1,n32_3,[gt32_3])*F32c_3
g32_new=[list(g32_new_1[0]),list(g32_new_2[0]),list(g32_new_3[0])]
yh32_21_1=VectorRepresentationOf(Yh32_21[:,:n32_1],S32)
yh32_21_2=VectorRepresentationOf(Yh32_21[:,n32_1:n32_1+n32_2],S32)
yh32_21_3=VectorRepresentationOf(Yh32_21[:,n32_1+n32_2:n32_1+n32_2+n32_3],S32)
y32_new_1=matrix(S32,1,n32_1,[Pr32.operator_eval(yh32_21_1[i]) for i in [0..n32_1-1]])*F32c_1
y32_new_2=matrix(S32,1,n32_2,[Pr32.operator_eval(yh32_21_2[i]) for i in [0..n32_2-1]])*F32c_2
y32_new_3=matrix(S32,1,n32_3,[Pr32.operator_eval(yh32_21_3[i]) for i in [0..n32_3-1]])*F32c_3
y32_new=[list(y32_new_1[0]),list(y32_new_2[0]),list(y32_new_3[0])]
k32_new=[1,k32_1+Pr32.degree(),k32_2+Pr32.degree(),k32_3+Pr32.degree()]
Out32=UniqueDecodingIGabUsingGrobnerBasis(g32_new,y32_new,k32_new,p32,nu32,m32,sigma32)
Out32
```

```
[(14*a32^7 + 3*a32^6 + 28*a32^5 + 28*a32^4 + 25*a32^3 + 18*a32^2 + 16*a32 + 5)*X^2 + (25*a32^7 + 6*a32^6 +
22*a32^5 + 5*a32^4 + 23*a32^3 + 31*a32^2 + 29*a32 + 30)*X + 25*a32^7 + 14*a32^6 + 22*a32^5 + 12*a32^4 +
17*a32^3 + 31*a32^2 + 13*a32 + 30, (18*a32^7 + 29*a32^6 + 9*a32^5 + 23*a32^4 + 19*a32^3 + 20*a32^2 + 10*a32
22)*X^2 + (31*a32^7 + 19*a32^6 + 23*a32^5 + 2*a32^4 + 30*a32^3 + 22*a32^2 + 27*a32 + 8)*X + 13*a32^7 +
24*a32^6 + 5*a32^5 + 22*a32^4 + 17*a32^3 + 12*a32^2 + 22*a32 + 20, (8*a32^7 + 13*a32^6 + 5*a32^5 + 4*a32^4 +
12*a32^3 + 22*a32^2 + 30*a32 + 1)*X^2 + (26*a32^7 + 31*a32^6 + 9*a32^5 + 18*a32^4 + 16*a32^3 + 21*a32^2 +
16*a32 + 14)*X + 15*a32^7 + 10*a32^6 + 22*a32^5 + 30*a32^4 + 30*a32^3 + 13*a32^2 + 21*a32 + 13]
```

```
if Out32=='decoding failure' : print "'decoding failure'"
else:
    print LeftDivisionOf(Out32[0],Pr32,sigma32,m32)==[f32_1,S32x(0)]
    print LeftDivisionOf(Out32[1],Pr32,sigma32,m32)==[f32_2,S32x(0)]
    print LeftDivisionOf(Out32[2],Pr32,sigma32,m32)==[f32_3,S32x(0)]
```

```
True
True
True
```

Appendix B: Publication

The main results obtained in this thesis were the subject of an article which was published in "IEEE Transactions on Information Theory", one of the best journals specialized in coding theory.

H. T. Kamche and C. Mouaha, "Rank-Metric Codes Over Finite Principal Ideal Rings and Applications," IEEE Transactions on Information Theory, vol. 65, no. 12, pp. 7718-7735, Dec. 2019.

Rank-Metric Codes Over Finite Principal Ideal Rings and Applications

Hermann Tchatchiem Kamche and Christophe Mouaha

***Abstract*— In this paper, it is shown that some results in the theory of rank-metric codes over finite fields can be extended to finite commutative principal ideal rings. More precisely, the rank metric is generalized and the rank-metric Singleton bound is established. The definition of Gabidulin codes is extended and it is shown that its properties are preserved. The theory of Gröbner bases is used to give the unique decoding, minimal list decoding, and error-erasure decoding algorithms of interleaved Gabidulin codes. These results are then applied in space-time codes and in random linear network coding as in the case of finite fields. Specifically, two existing encoding schemes of random linear network coding are combined to improve the error correction.**

***Index Terms*— Finite principal ideal rings, Galois extension, Gröbner bases, interleaved Gabidulin codes, random linear network coding, rank-metric codes, skew polynomials, space-time codes.**

I. Introduction

In a communication network, the transmitters can send information simultaneously to the receivers. These are represented by a matrix where rows consist of various information. Practically, it may happen some perturbations and the received signals be different from the transmitted ones. In such predicament, for securing the system against noises, one can use the rank-metric codes to detect and correct errors.

A. Rank-Metric Codes

Rank-metric codes [1] are codes whose each codeword is a matrix and the distance between two codewords is the rank of their difference. The most important family of rank-metric codes is that of Gabidulin codes [1]–[3]. They are optimal in the sense that they achieve the rank-metric Singleton bound. In [2], Gabidulin used the Galois extension to give the vector representation of rank-metric codes. He also gave a polynomial-time unique decoding algorithm of Gabidulin codes.

The length of a Gabidulin code is lower bounded by the degree of the Galois extension. To increase the code length, we can use an interleaved Gabidulin code [4] which is a direct sum of several Gabidulin codes. Another advantage of interleaved Gabidulin codes is the existence of polynomial-time decoding algorithms [4]–[6] that can decode beyond the error correction capability with high probability. Nowadays, rank-metric codes are used in space-time coding [7], public key cryptosystems [8] and random linear network coding [9].

B. Space-time codes based on rank-metric codes

A space-time code is a multiple-input/multiple-output transmit strategy for fading channels in point-to-point single-user scenarios. It was introduced in [10] by Tarokh et al. It combines the space diversity, provided by multiple antennas, and the time diversity to increase system capacity and reduce multipath fading. Among the performance criteria for space-time codes, we have the rank criterion [10] which states that in order to achieve the maximum diversity, the rank of the difference of two distinct codewords has to be maximal. On the other hand, for any space-time block code there is a tradeoff between the transmission rate and the transmit diversity gain [10], [11]. As in [12], a space-time block code that achieves this rate-diversity tradeoff will be called an optimal space-time block code. To construct these optimal codes, rank-metric codes can be used. Thus, in [7] Lusina et al. used rank-preserving map from finite fields to Gaussian integers to construct optimal space-time block codes from rank-metric codes over finite fields. In [13], Asif et al. used interleaved Gabidulin codes to construct space-time block codes and compared them to orthogonal space-time block codes. In [14], Puchinger et al. extended the works of Lusina *et al.* [7] to Eisenstein integers. They also proposed decoding scheme of space-time block codes using lattice-reduction-aided equalization and error-erasure decoding algorithm of Gabidulin codes. In [15], Augot et al. transposed the theory of rank metric and Gabidulin codes to the case of fields of characteristic zero.

C. Rank-Metric Codes in Random Linear Network Coding

A random linear network coding is a technique that can be used to disseminate information in networks and improve the performance of communication systems. In the transmission model for end-to-end coding over finite fields, the channel equation is given by $\mathbf{Y} = \mathbf{AX} + \mathbf{E}$, where $\mathbf{X}$ is the transmitted matrix whose rows are packets transmitted by the source node; $\mathbf{Y}$ is the received matrix whose rows are the packets received by the sink node; $\mathbf{A}$ is a transfer matrix corresponding to the overall linear transformation applied by intermediate nodes

Manuscript received August 8, 2018; revised June 8, 2019; accepted July 29, 2019. Date of publication August 6, 2019; date of current version November 20, 2019.

H. Tchatchiem Kamche is with the Department of Mathematics, Faculty of Science, University of Yaoundé I, Cameroon (e-mail: tchatchiemh@yahoo.fr).

C. Mouaha is with the Department of Mathematics, Higher Teacher Training School, University of Yaoundé I, Cameroon (e-mail: cmouaha@yahoo.fr).

Communicated by F. Oggier, Associate Editor for Coding Theory.

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIT.2019.2933520

of the network and $\mathbf{E}$ is an error matrix whose rows are linear combinations of corrupt packets injected in the network. Random matrices $\mathbf{A}$ and $\mathbf{E}$ are unknown to the destination. The problem is to recover the transmitted codeword $\mathbf{X}$ from the received matrix $\mathbf{Y}$.

Since linear network coding is vector-space preserving, Kötter and Kschischang [16] suggested the use of a basis of a vector space as the rows of the transmitted matrix. They defined a distance function between subspaces, constructed a family of constant-dimension subspace codes and the decoding algorithm. In [9] Silva et al. used the lifted rank-metric codes to show that minimum distance decoding of constant-dimension subspace codes can be reformulated as a generalized decoding problem for rank-metric codes. They then gave an error-erasure decoding algorithm of Gabidulin codes to solve the problem of error control in random linear network coding.

D. Network Coding Over Finite Principal Ideal Rings

A principal ideal ring is a ring in which any ideal is generated by one element. In a digital modulation system, some signal constellation sets can be represented by a finite principal ideal ring. In particular [17], if η is some positive integer then the signal constellation set of the η^2-ary square quadrature amplitude modulation is represented by the ring $\mathbb{Z}_\eta[i] = \mathbb{Z}_\eta + i\mathbb{Z}_\eta$ where $i^2 = -1$ and $\mathbb{Z}_\eta$ is the ring of integers modulo η. The works on nested-lattice-based network coding [17], [18] allow the construction of more efficient physical-layer network coding schemes with network coding over finite principal ideal rings. Motivated by this algebraic approach, space-time codes and random linear network coding were studied in the specific cases of principal ideal rings.

In [12], Kiran and Rajan extended the definition of Gabidulin codes to Galois rings and used a rank-preserving map to construct an optimal space-time block code. In [19], Liu et al. defined the notion of $\sum_o$-rank over the ring $\mathbb{Z}_{2^k}[i]$ and used it to construct the rank metric space-time codes for the 2^{2k} quadrature and amplitude modulated. The works of Silva *et al.* [20] and Nóbrega *et al.* [21] were extended respectively in [22] and [23] to finite chain rings. The works of Kötter and Kschischang [16], and Gorla and Ravagnani [24] were extended in [25] to finite principal ideal rings.

Note that the works of [22], [25] and [23] allow to improve the error correction in random linear network coding over finite principal ideal rings. As in the case of finite fields, another method that one can use is rank-metric codes. Thus, in this paper we focus on a problem raised by Frank R. Kschischang which consists of studying properties of rank-metric codes likely to be preserved over finite principal ideal rings. The resolution of this problem will allow to give the encoding and decoding schemes for random linear network coding over finite principal ideal rings. Moreover, an optimal space-time block code will be constructed for all digital modulation systems whose signal constellation set is algebraically represented [17] by a finite principal ideal ring.

E. Our Contribution

To extend rank-metric codes to finite principal ideal rings, we first extend the rank metric using the Smith normal form of a matrix. We then use the Galois extensions to prove that Gabidulin codes can be extended to finite principal ideal rings and that its properties are preserved. As in [4], we show that collaborative decoding of interleaved Gabidulin codes can be translated to the problem of reconstruction of skew polynomials. Analogous to [26], the theory of Gröbner bases is used to give an iterative algorithm to solve this reconstruction problem. The solutions of this problem allow us to give the unique decoding, minimal list decoding, and error-erasure decoding algorithms of interleaved Gabidulin codes. We then apply these results to space-time coding and random linear network coding. Specifically, we show that there is a rank-preserving map from a finite principal ideal ring to a complex signal set and we use it to construct an optimal space-time block code. We combine the encoding and decoding schemes of [9] and [20] to improve the error correction in random linear network coding.

F. Structure of the Paper

In Section II, we set basic notations and review some facts about skew polynomials. In Section III, we show that the rank metric can be extended to principal ideal rings. We establish the rank-metric Singleton bound and prove that Gabidulin codes achieve this bound as in the case of finite fields. In Section IV, we describe the interleaved Gabidulin codes, give the key equation and an algorithm to solve it. The decoding algorithms are given in Section V. The applications in space-time codes and in random linear network coding are given in Section VI. We present our conclusions in Section VII.

II. Preliminaries

A. Smith Normal Form

Throughout this paper, by ring we mean a commutative ring with identity element, ring homomorphisms are assumed to be unitary, and all modules are unital. Unless otherwise specified, we assume that R is a finite principal ideal ring.

An element $u \in R$ is called a unit if $uv = 1$ for some $v \in R$. Let $a, b \in R$, we say that a divides b, denoted $a|b$, if $b = ca$ for some $c \in R$. The set of all $m \times n$ matrices with entries from R will be denoted by $R^{m\times n}$. The $k \times k$ identity matrix is denoted by $\mathbf{I}_k$. Let $\mathbf{A} \in R^{m\times n}$, we denote by $row(\mathbf{A})$ and $col(\mathbf{A})$ the R-submodules generated by the row and column vectors of $\mathbf{A}$, respectively.

A matrix $\mathbf{D} = (d_{i,j}) \in R^{m\times n}$ is called a diagonal matrix if $d_{i,j} = 0$ whenever $i \neq j$. A diagonal matrix $\mathbf{D} = (d_{i,j}) \in R^{m\times n}$ can be written as $\mathbf{D} = diag(d_1, \ldots, d_r)$, where $r = min\{n, m\}$, and $d_i = d_{i,i}$, for $i = 1, ..., r$. By [27, Theorem 15.24], for all matrix $\mathbf{A} \in R^{m\times n}$, there are two invertible matrices $\mathbf{P}$, $\mathbf{Q}$ and a diagonal matrix $\mathbf{D} = diag(d_1, d_2, \ldots, d_r)$ satisfying the divisibility relations $d_1|d_2| \ldots |d_r$, such that $\mathbf{A} = \mathbf{PDQ}$. The elements $d_1, d_2, \ldots, d_r$ are unique up to associates and the matrix $\mathbf{D}$ is called a Smith normal form of $\mathbf{A}$.

***Example** 2.1:* Let $R = \mathbb{Z}_{12}$. Set

$$\mathbf{A} = \begin{pmatrix} 8 & 10 & 4 & 4 \\ 4 & 2 & 8 & 2 \\ 11 & 6 & 0 & 6 \end{pmatrix}.$$

Using SageMathCloud [28], we compute a Smith normal form of $\mathbf{A}$, and we get

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 6 & 0 \end{pmatrix}.$$

In [29], Storjohann gave an algorithm for computing the Smith normal form over principal ideal rings and its complexity. As in [30] and [31], one can use the Smith normal form to solve a linear system of equations over principal ideal rings.

B. Finite Chain Rings

A local ring is a ring with exactly one maximal ideal. A chain ring is a ring whose ideals are linearly ordered by inclusion. It is known (see, e.g., [32]) that a finite ring is a chain ring if and only if it is a local principal ideal ring. Therefore, by the structure theorem of finite commutative rings [32, Theorem VI.2], each finite principal ideal ring can be decomposed as a direct sum of finite chain rings.

Examples of finite chain rings are the ring $\mathbb{Z}_{p^k}$, p is a prime, and the ring $\mathbb{Z}_{2^k}[i]$, whose the maximal ideals are $p\mathbb{Z}_{p^k}$ and $(1+i)\mathbb{Z}_{2^k}[i]$, respectively. Other examples of construction of finite chain rings using the ring of algebraic integers are given in [12]. The characterization of finite chain rings is given in [32, Theorem XVII.5].

In a finite chain ring, every ideal is a power of the maximal ideal. More specifically, assume that R is a finite chain ring, π a generator of its maximal ideal, ν the nilpotency index of π, i.e., the smallest positive integer such that $\pi^\nu = 0$. Then, every ideal of R is of the form $\pi^i R$, for $i = 0, \dots, \nu$, and for all $a \in R\backslash\{0\}$ there is a unique $i \in \{0, \dots, \nu - 1\}$ and a unit $u \in R$ such that $a = \pi^i u$.

Thus, to compute the Smith normal form over finite chain rings, one can also use the simple method given in the proof of [33, Theorem 1.1.12.]. As in the proof of [27, Theorem 15.9], one can then compute the Smith normal form over finite principal ideal rings.

C. Galois Extension of Finite Principal Ideal Rings

Let ρ be the positive integer such that

$$R \cong R_{(1)} \times \cdots \times R_{(\rho)},$$

where $R_{(i)}$ is a finite chain ring, for $i = 1, \dots, \rho$. Using this isomorphism, we identify R with $R_{(1)} \times \cdots \times R_{(\rho)}$. Let $i \in \{1, \dots, \rho\}$, we denote by $\mathfrak{m}_{(i)}$ the maximal ideal of $R_{(i)}$, $\mathbb{F}_{q_{(i)}} = R_{(i)}/\mathfrak{m}_{(i)}$ its residue field and $\nu_{(i)}$ the nilpotency index of $\mathfrak{m}_{(i)}$. We denote the natural projection $R_{(i)} \to \mathbb{F}_{q_{(i)}}$ by $\psi_{(i)}$. We extend $\psi_{(i)}$ coefficient-by-coefficient to polynomials over $R_{(i)}$. Let m be a nonzero positive integer. Let $i \in \{1, \dots, \rho\}$ and $h_{(i)} \in R_{(i)}[X]$ be a monic polynomial of degree m such that $\psi_{(i)}\left(h_{(i)}\right)$ is irreducible in $\mathbb{F}_{q_{(i)}}[X]$. Set

$$S_{(i)} = R_{(i)}[X]/\left(h_{(i)}\right),$$

where $\left(h_{(i)}\right)$ denotes the ideal generated by $h_{(i)}$. By [32], $S_{(i)}$ is a free local Galois extension of $R_{(i)}$ of $R_{(i)}$-dimension m, with the maximal ideal $\mathfrak{M}_{(i)} = \mathfrak{m}_{(i)} S_{(i)}$, where the Galois group is cyclic of order m, generated by a power map $\sigma_{(i)} : \alpha_{(i)} \mapsto \alpha_{(i)}^{q_{(i)}}$ on the suitable primitive element $\alpha_{(i)}$. Moreover, $\mathbb{F}_{q_{(i)}^m} = S_{(i)}/\mathfrak{M}_{(i)}$. Set

$$S = S_{(1)} \times \cdots \times S_{(\rho)}$$

and $\sigma = \left(\sigma_{(i)}\right)_{1 \le i \le \rho}$. Let $G_R(S)$ be the group generated by σ, then by [34, Proposition 1.2(5), pp.80], S is a Galois extension of R with the Galois group $G_R(S)$. Since $R_{(i)}$ is a finite chain ring and $S_{(i)}$ is a free $R_{(i)}$-module of rank m, then S is a finite principal ideal ring and it is a free R-module of rank m. Note that by [35, Theorem 3.2.], there is a monic polynomial $h \in R[X]$ of degree m such that $S \cong R[X]/(h)$.

***Example** 2.2:* Let $R = \mathbb{Z}_{12}$. By the Chinese remainder theorem [27, page 175], we have $R \cong R_{(1)} \times R_{(2)}$ where $R_{(1)} = \mathbb{F}_3$ and $R_{(2)} = \mathbb{Z}_4$. Set $S_{(1)} = \mathbb{F}_{3^4}$, $h_{(2)} = X^4 + 2X^2 + 3X + 1$, $S_{(2)} = R_{(2)}[X]/\left(h_{(2)}\right)$, $\alpha_{(2)} = X + \left(h_{(2)}\right)$. Let the maps $\sigma_{(1)} : S_{(1)} \to S_{(1)}$ given by $\sigma_{(1)}(x) = x^3$, for all $x \in S_{(1)}$, and $\sigma_{(2)} : S_{(2)} \to S_{(2)}$ given by $\alpha_{(2)} \mapsto \alpha_{(2)}^2$, that is, for all $x = x_0 + x_1\alpha_{(2)} + x_2\alpha_{(2)}^2 + x_3\alpha_{(2)}^3 \in S_{(2)}$, where $x_0, x_1, x_2, x_3 \in R_{(2)}$, $\sigma_{(2)}(x) = x_0 + x_1\alpha_{(2)}^2 + x_2\alpha_{(2)}^4 + x_3\alpha_{(2)}^6$. Then $S_{(1)} \times S_{(2)}$ is a Galois extension of $R_{(1)} \times R_{(2)}$ where the Galois group is generated by $\left(\sigma_{(1)}, \sigma_{(2)}\right)$.

D. Skew Polynomials

In this subsection, we show that some properties of linearized polynomials over finite fields [36] can be generalized to finite principal ideal rings. Let $S[X, \sigma]$ be the set of all (skew) polynomials $a_0 + a_1X + \cdots + a_nX^n$, where $n \in \mathbb{N}$, $a_i \in S$, for $i = 0, \dots, n$, and X is an indeterminate. The addition in $S[X, \sigma]$ is defined to be the usual addition of polynomials and the multiplication is defined by the basic rule $Xa = \sigma(a)X$, for all $a \in S$, and extended to all elements of $S[X, \sigma]$ by associativity and distributivity. The set $S[X, \sigma]$ with the above operations forms a ring called the skew polynomial ring over S with automorphism σ.

Let $f = f_0 + f_1X + \cdots + f_nX^n \in S[X, \sigma]$ with $f_n \neq 0$, then n is called the degree of f, X^n the leading monomial of f, f_n the leading coefficient of f, f_nX^n the leading term of f, denoted $\deg(f)$, $lm(f)$, $lc(f)$ and $lt(f)$ respectively. If $f = 0$, then we put $\deg(0) := -\infty$, $lm(0) := 0$, $lc(0) := 0$ and $lt(0) := 0$. The skew polynomial f is called monic if $lc(f) = 1$. We denote by $S[X, \sigma]_{<k}$ the set of all skew polynomials of degree less than k.

It has been proved (see, e.g., [37]) that for all f and g in $S[X, \sigma]$, we have $\deg(f + g) \le \max\{\deg(f), \deg(g)\}$ and $\deg(fg) \le \deg(f) + \deg(g)$. Furthermore, if the leading coefficients of g is a unit, then $\deg(fg) = \deg(f) + \deg(g)$ and there exist unique polynomials q, q', r and r' in $S[X, \sigma]$ such that $f = qg + r$ (right division) and $f = gq' + r'$ (left division) with $\deg(r) < \deg(g)$ and $\deg(r') < \deg(g)$.

Note that if $R = \mathbb{F}_q$, then $S = \mathbb{F}_{q^m}$ and $\sigma(x) = x^q$, for all $x \in \mathbb{F}_{q^m}$. Thus, we now prove that some results in [36] can be extended to finite principal ideal rings.

***Notation** 2.3:* Let $f = f_0 + f_1X + \cdots + f_nX^n \in S[X, \sigma]$, $b \in S$ and $\mathbf{b} = (b_1, \dots, b_n) \in S^n$.

1) The element $f_0b + f_1\sigma(b) + \cdots + f_n\sigma^n(b)$ will be denoted by $f(b)$.
2) The kernel of f is $\ker f := \{x \in S : f(x) = 0\}$.

3) The vector $(f(b_1), \ldots, f(b_n))$ will be denoted by $f(\mathbf{b})$.

As $S = S_{(1)} \times \cdots \times S_{(\rho)}$ and $\mathfrak{M}_{(i)} = \mathfrak{m}_{(i)} S_{(i)}$, we have the following Lemma.

***Lemma** 2.4:* Let $y \in S$. If $\{y\}$ is linearly independent over R, then y is a unit.

Proof: Suppose that $\{y\}$ is linearly independent over R and y is not a unit. Set $y = \left(y_{(i)}\right)_{1 \le i \le \rho}$ where $y_{(i)} \in S_{(i)}$. Since y is not a unit, then there is $i_0 \in \{1, \ldots, \rho\}$ such that $y_{(i_0)}$ is not a unit. Consequently, $y_{(i_0)} \in \mathfrak{M}_{(i_0)}$. As $\mathfrak{M}_{(i_0)} = \mathfrak{m}_{(i_0)} S_{(i_0)}$, there is $0 \ne b_{(i_0)} \in \mathfrak{m}_{(i_0)}^{\nu_{(i_0)}-1}$ such that $b_{(i_0)} y_{(i_0)} = 0$. Set $b = \left(\beta_{(i)}\right)_{1 \le i \le \rho}$ where $\beta_{(i_0)} = b_{(i_0)}$ and $\beta_{(i)} = 0$ if $i \ne i_0$. Then $by = 0$, which is impossible because $\{y\}$ is linearly independent over R. ■

Analogous to [36], we have the following two propositions.

***Proposition** 2.5:* Let $\{u_j\}_{1 \le j \le r}$ be a subset of S, which is linearly independent over R. Then, there is a monic skew polynomial $f \in S[X, \sigma]$ of degree r such that $\ker f = \left\langle \{u_j\}_{1 \le j \le r} \right\rangle$, where $\left\langle \{u_j\}_{1 \le j \le r} \right\rangle$ denotes the R-submodule of S generated by $\{u_j\}_{1 \le j \le r}$.

Proof: We prove by induction on $k \in \{1, \ldots, r\}$. Set $f_1 = X - \sigma(u_1) u_1^{-1}$, we have $\ker f_1 = \langle \{u_1\} \rangle$. Let $k \in \{1, \ldots, r-1\}$. Assume there is a monic polynomial $f_k \in S[X, \sigma]$ of degree k such that $\ker f_k = \left\langle \{u_j\}_{1 \le j \le k} \right\rangle$. We claim that $f_k(u_{k+1})$ is a unit. Indeed, let $a \in R$ such that $a f_k(u_{k+1}) = 0$, then $a u_{k+1} \in \ker f_k = \left\langle \{u_i\}_{1 \le j \le k} \right\rangle$. Consequently, $a = 0$ because $\{u_j\}_{1 \le j \le k+1}$ is R-linear independent. Therefore, by Lemma 2.4, $f_k(u_{k+1})$ is a unit. Set $f_{k+1} = \left(X - \sigma\left(f_k(u_{k+1})\right) f_k(u_{k+1})^{-1}\right) \times f_k$, then $\deg(f_{k+1}) = k+1$ and $\ker f_{k+1} = \left\langle \{u_j\}_{1 \le j \le k+1} \right\rangle$ ■

***Proposition** 2.6:* Let $\{u_j\}_{1 \le j \le r}$ be a subset of S. Then, the matrix $\left(\sigma^i(u_j)\right)_{0 \le i \le r-1,\ 1 \le j \le r}$ is invertible if and only if $\{u_j\}_{1 \le j \le r}$ is linearly independent over R.

Proof: Assume that $\{u_j\}_{1 \le j \le r}$ is linearly independent over R. Let $i \in \{1, \ldots, r\}$. By Proposition 2.5, there is a monic skew polynomial $T_i \in S[X, \sigma]$ of degree $r-1$ such that $\ker T_i = \left\langle \{u_j\}_{1 \le j \le r, j \ne i} \right\rangle$. Using the same arguments as in the proof of Proposition 2.5, we can show that $T_i(u_i)$ is a unit. Set $T_i(u_i)^{-1} T_i(X) = \sum_{0 \le j \le r-1} v_{i,j} X^j$, where $v_{i,j} \in S$, then the matrix $\left(v_{i,j}\right)_{1 \le i \le r,\ 0 \le j \le r-1}$ is the inverse of the matrix $\left(\sigma^i(u_j)\right)_{0 \le i \le r-1,\ 1 \le j \le r}$.

Conversely, assume that $\left(\sigma^i(u_j)\right)_{0 \le i \le r-1,\ 1 \le j \le r}$ is invertible. Let $\lambda_1, \ldots, \lambda_r$ be the elements of R such that $\lambda_1 u_1 + \cdots + \lambda_r u_r = 0$. Then, we have $\lambda_1 \sigma^i(u_1) + \cdots + \lambda_r \sigma^i(u_r) = 0$, for $i = 0, \ldots, r-1$. Consequently, $\lambda_1 = \cdots = \lambda_r = 0$. ■

From the preceding proposition, we get the following corollary.

***Corollary** 2.7:* Let $\{u_j\}_{1 \le j \le r}$ be a subset of S, which is linearly independent over R and let $V \in S[X, \sigma]$ be a monic skew polynomial of degree r such that $\ker V = \left\langle \{u_j\}_{1 \le j \le r} \right\rangle$. Let $P \in S[X, \sigma]$. Then, $P(u_j) = 0$, for $j = 1, \ldots, r$, if and only if there is $Q \in S[X, \sigma]$ such that $P = QV$.

E. Gröbner Bases of Modules Over Skew Polynomials

In [38], Jiménez and Lezama studied the theory of Gröbner bases of modules over skew Poincaré–Birkhoff–Witt extension. In this subsection, we recall some results in this theory that we will use to solve the key equation.

Let ℓ be a positive integer, we denote by $S[X, \sigma]^{\ell+1}$ the $\ell+1$-fold direct product of $S[X, \sigma]$. For all $\mathbf{u} \in S[X, \sigma]^{\ell+1}$, the l-th component of $\mathbf{u}$ is denoted by $u^{(l)}$, for $l \in \{0, \ldots, \ell\}$, i.e. $\mathbf{u} = \left(u^{(0)}, u^{(1)}, \ldots, u^{(\ell)}\right)$. We consider $S[X, \sigma]^{\ell+1}$ as a left $S[X, \sigma]$-module where addition is defined componentwise and for $a \in S[X, \sigma]$ and $\mathbf{u} \in S[X, \sigma]^{\ell+1}$, $a\mathbf{u} = \left(au^{(0)}, au^{(1)}, \ldots, au^{(\ell)}\right)$. We denote by $\mathbf{e}^{(0)} = (1, 0, \ldots, 0)$, $\mathbf{e}^{(1)} = (0, 1, 0, \ldots, 0)$, $\ldots$, $\mathbf{e}^{(\ell)} = (0, \ldots, 0, 1)$ the canonical basis of $S[X, \sigma]^{\ell+1}$. A monomial in $S[X, \sigma]^{\ell+1}$ is an element of the form $X^\alpha \mathbf{e}^{(l)}$ where $\alpha \in \mathbb{N}$ and $l \in \{0, \ldots, \ell\}$. The set of monomials of $S[X, \sigma]^{\ell+1}$ will be denoted by $Mon\left(S[X, \sigma]^{\ell+1}\right)$. If $X^\alpha \mathbf{e}^{(l)} \in Mon\left(S[X, \sigma]^{\ell+1}\right)$, then l is called the index of $X^\alpha \mathbf{e}^{(l)}$ and denoted by $ind\left(X^\alpha \mathbf{e}^{(l)}\right)$. Let $X^{\alpha_1} \mathbf{e}^{(l_1)}, X^{\alpha_2} \mathbf{e}^{(l_2)} \in Mon\left(S[X, \sigma]^{\ell+1}\right)$, we say that $X^{\alpha_1} \mathbf{e}^{(l_1)}$ divides $X^{\alpha_2} \mathbf{e}^{(l_2)}$, denoted $X^{\alpha_1} \mathbf{e}^{(l_1)} | X^{\alpha_2} \mathbf{e}^{(l_2)}$, if $l_1 = l_2$ and there is $\beta \in \mathbb{N}$ such that $\alpha_2 = \alpha_1 + \beta$. We will say that any monomial $X^\alpha \mathbf{e}^{(l)} \in Mon\left(S[X, \sigma]^{\ell+1}\right)$ divides the null vector $\mathbf{0}$.

***Definition** 2.8:* A monomial order on $Mon\left(S[X, \sigma]^{\ell+1}\right)$ is a total order $\succeq$ satisfying the following two conditions:

(i) $X^\beta \left(X^\alpha \mathbf{e}^{(l)}\right) \succeq X^\alpha \mathbf{e}^{(l)}$, for all $X^\alpha \mathbf{e}^{(l)} \in Mon\left(S[X, \sigma]^{\ell+1}\right)$ and every $\beta \in \mathbb{N}$;

(ii) if $X^{\alpha_2} \mathbf{e}^{(l_2)} \succeq X^{\alpha_1} \mathbf{e}^{(l_1)}$, then

$$X^\beta \left(X^{\alpha_2} \mathbf{e}^{(l_2)}\right) \succeq X^\beta \left(X^{\alpha_1} \mathbf{e}^{(l_1)}\right)$$

for all $X^{\alpha_1} \mathbf{e}^{(l_1)}$, $X^{\alpha_2} \mathbf{e}^{(l_2)} \in Mon\left(S[X, \sigma]^{\ell+1}\right)$ and every $\beta \in \mathbb{N}$.

If $X^{\alpha_2} \mathbf{e}^{(l_2)} \succeq X^{\alpha_1} \mathbf{e}^{(l_1)}$ and $X^{\alpha_2} \mathbf{e}^{(l_2)} \ne X^{\alpha_1} \mathbf{e}^{(l_1)}$ we will write $X^{\alpha_2} \mathbf{e}^{(l_2)} \succ X^{\alpha_1} \mathbf{e}^{(l_1)}$.

$X^{\alpha_1} \mathbf{e}^{(l_1)} \preceq X^{\alpha_2} \mathbf{e}^{(l_2)}$ means that $X^{\alpha_2} \mathbf{e}^{(l_2)} \succeq X^{\alpha_1} \mathbf{e}^{(l_1)}$.

***Remark** 2.9:* By [39, Chapter 0, Section 17, Lemma 15] a monomial order on $Mon\left(S[X, \sigma]^{\ell+1}\right)$ is a well order. Note that the condition (iii) of [38, Definition 15.] is given so that a monomial order is a well order. So, in this restricted specific case we do not need this condition.

We fix a monomial order $\succeq$ on the monomials of $S[X, \sigma]^{\ell+1}$. Let $\mathbf{f} \in S[X, \sigma]^{\ell+1} \setminus \{\mathbf{0}\}$, then $\mathbf{f}$ can be written uniquely as $\mathbf{f} = \sum_{i=1}^n c_i X^{\alpha_i} \mathbf{e}^{(l_i)}$ where $n \in \mathbb{N}$, $c_i \in S$, for $i = 1, \ldots, n$, $c_1 \ne 0$ and $X^{\alpha_1} \mathbf{e}^{(l_1)} \succ \cdots \succ X^{\alpha_n} \mathbf{e}^{(l_n)}$. We define:

- $lm(\mathbf{f}) := X^{\alpha_1} \mathbf{e}^{(l_1)}$ as the leading monomial of $\mathbf{f}$;
- $lc(\mathbf{f}) := c_1$ as the leading coefficient of $\mathbf{f}$;
- $lt(\mathbf{f}) := c_1 X^{\alpha_1} \mathbf{e}^{(l_1)}$ as the leading term of $\mathbf{f}$;
- $\deg(\mathbf{f}) := \alpha_1$ as the degree of $\mathbf{f}$.

For $\mathbf{f} = \mathbf{0}$ we define $lt(\mathbf{0}) := \mathbf{0}$, $lm(\mathbf{0}) := \mathbf{0}$, $lc(\mathbf{0}) := 0$ and extend $\succeq$ to $Mon\left(S[X, \sigma]^{\ell+1}\right) \cup \{\mathbf{0}\}$ by $X^\alpha \mathbf{e}^{(l)} \succ \mathbf{0}$ for all $X^\alpha \mathbf{e}^{(l)} \in Mon\left(S[X, \sigma]^{\ell+1}\right)$. According to [38, Theorem 26.], we give the following:

***Definition** 2.10:* Let M be a nonzero submodule of $S[X, \sigma]^{\ell+1}$ and let G be a non empty finite subset of nonzero vectors of M, we say that G is a Gröbner basis for M if for all $\mathbf{f} \in M$ there exist $\mathbf{g}_1, \ldots, \mathbf{g}_t \in G$ such that $lm\left(\mathbf{g}_j\right) | lm(\mathbf{f})$, for $j = 1, \ldots, t$, i.e., there exist $\alpha_j \in \mathbb{N}$ such that $lm(\mathbf{f}) = X^{\alpha_j} lm\left(\mathbf{g}_j\right)$, and $lc(\mathbf{f}) \in \langle \sigma^{\alpha_1}(lc(\mathbf{g}_1)), \ldots, \sigma^{\alpha_t}(lc(\mathbf{g}_t)) \rangle$. We will say that $\{\mathbf{0}\}$ is a Gröbner basis for $M = \{\mathbf{0}\}$.

By [38, Theorem 23.] and [38, Theorem 26.], we have the following:

***Proposition** 2.11:* Let M be a submodule of $S[X,\sigma]^{\ell+1}$ and let $G=\{\mathbf{g}_1,\ldots,\mathbf{g}_t\}\subset M$. If G is a Gröbner basis for M then for all $\mathbf{f}\in M$ there exist $q_1,\ldots,q_t\in S[X,\sigma]$ such that $\mathbf{f}=q_1\mathbf{g}_1+\cdots+q_t\mathbf{g}_t$ with

$$lm(\mathbf{f})=\max\{lm(q_1)\,lm(\mathbf{g}_1),\ldots,lm(q_t)\,lm(\mathbf{g}_t)\}.$$

III. Rank-Metric Codes Over Principal Ideal Rings

In this section, as in the case of finite fields, we give the two representations of rank codes [40]: matrix representation and vector representation. We establish the rank-metric Singleton bound. We extend the definition of Gabidulin codes and prove that its properties are preserved.

A. Rank Metric

In field theory, the rank of a matrix defines a group-norm in the matrix space of the same size. We extend this property to principal ideal rings. As in [27, page 190] we use the following notation.

***Notation** 3.1:* Let M be a finitely generated R-module. The smallest number of elements in M which generate M as an R-module is denoted by $\mu_R(M)$. If $M=\{0\}$, then we set $\mu_R(M)=0$.

By [41], if F is a finitely generated free R-module and $\{e_1,\ldots,e_n\}$ is a free basis of F, i.e., a linearly independent generating set, then $\mu_R(F)=n$ and any generating set of F consisting of n elements is a free basis of F. Using the Smith normal form, we have the following proposition.

***Proposition** 3.2:* Let M be a finitely generated R-module, $\mu_R(M)=r_M$, and let N be a submodule of M, $\mu_R(N)=r_N$. Then, $r_N\le r_M$ and there is a generating set $\{u_i\}_{1\le i\le r_M}$ of M and r_N scalars $d_1,\ldots,d_{r_N}$ of R such that $\{d_iu_i\}_{1\le i\le r_N}$ generates N, with $d_1|d_2|\ldots|d_{r_N}$. Furthermore, if M is a free module then $\{u_i\}_{1\le i\le r_M}$ is a free basis of M.

Note that if N and N' are two submodules of a finitely generated R-module, then $\mu_R\left(N+N'\right)\le\mu_R(N)+\mu_R\left(N'\right)$. Thus, the minimum number of generators of a module over a principal ideal ring has several properties similar to the dimension of vector spaces. Therefore, analogous to the case of fields, we give the following definition.

***Definition** 3.3:* (Rank of matrix). Let $\mathbf{A}\in R^{m\times n}$.

(i) The rank of $\mathbf{A}$, denoted by $rank_R(\mathbf{A})$, or simply by $rank(\mathbf{A})$, is the number $\mu_R(col(\mathbf{A}))$.

(ii) The free rank of $\mathbf{A}$, denoted by $freerank_R(\mathbf{A})$, or simply by $freerank(\mathbf{A})$, is the maximum of the ranks of free R-submodules of $col(\mathbf{A})$.

Using the Smith normal form and [27, Theorem 15.33], we have the following proposition.

***Proposition** 3.4:* Let $\mathbf{A}\in R^{m\times n}\setminus\{\mathbf{0}\}$ and $\mathbf{D}=diag(d_1,\ldots,d_r)$ be a Smith normal form of $\mathbf{A}$. Then,

$$col(\mathbf{A})\cong row(\mathbf{A}),$$

$$rank(\mathbf{A})=\max\{i\in\{1,\ldots,r\}:d_i\neq 0\},$$

and

$$freerank(\mathbf{A})=\max\{i\in\{1,\ldots,r\}:d_i\text{ is a unit}\}.$$

***Corollary** 3.5:* Let $\mathbf{A}\in R^{m\times n}$. We have

$$rank_R(\mathbf{A})=\mu_R(row(\mathbf{A}))$$

and $freerank_R(\mathbf{A})$ is the maximum of the ranks of free R-submodules of $row(\mathbf{A})$.

***Example** 3.6:* If $\mathbf{A}$ is the matrix given in Example 2.1, then $rank(\mathbf{A})=3$ and $freerank(\mathbf{A})=1$.

***Remark** 3.7:* In linear algebra over fields, the rank-nullity theorem states that the sum of the rank of a matrix and the dimension of its right kernel is equal to the number of its columns. Using the definition of rank given in Definition 3.3, this property is not true in general over finite principal ideal rings, due to zero divisors. Indeed, let $\mathbb{Z}_6$ be the ring of integers modulo 6 and

$$\mathbf{A}=\begin{pmatrix}2&0\\0&2\end{pmatrix}$$

be a matrix with coefficients in $\mathbb{Z}_6$. The right kernel of $\mathbf{A}$ is generated by the vectors $(3,0)$ and $(0,3)$. By Proposition 3.4, $rank(\mathbf{A})=2$. Thus, the rank-nullity theorem can not be applied to the matrix $\mathbf{A}$.

Using the Smith normal form, we have the following proposition.

***Proposition** 3.8:* (Rank Decompositions). Let $\mathbf{E}\in R^{m\times n}$, $rank(\mathbf{E})=t$.

1) There are $\mathbf{A}\in R^{m\times t}$, $rank(\mathbf{A})=t$, and $\mathbf{B}\in R^{t\times n}$, $freerank(\mathbf{B})=t$, such that $\mathbf{E}=\mathbf{AB}$.
2) There are $\mathbf{A}'\in R^{m\times t}$, $freerank\left(\mathbf{A}'\right)=t$, and $\mathbf{B}'\in R^{t\times n}$, $rank\left(\mathbf{B}'\right)=t$, such that $\mathbf{E}=\mathbf{A}'\mathbf{B}'$.

The following theorem extends the notion of rank metric to principal ideal rings.

***Theorem** 3.9:* The map $R^{m\times n}\to\mathbb{N}$ given by $\mathbf{A}\mapsto rank(\mathbf{A})$ is a group-norm, i.e.,

(i) for all $\mathbf{A}\in R^{m\times n}$, $rank(\mathbf{A})=0$ if and only if $\mathbf{A}=\mathbf{0}$;
(ii) for all $\mathbf{A}\in R^{m\times n}$, $rank(-\mathbf{A})=rank(\mathbf{A})$;
(iii) for all $\mathbf{A},\ \mathbf{B}\in R^{m\times n}$,

$$rank(\mathbf{A}+\mathbf{B})\le rank(\mathbf{A})+rank(\mathbf{B}).$$

Proof: The proof is similar to that in the case of fields if we replace the dimension of the vector space by the minimum number of generators of a module. ■

***Remark** 3.10:* In general, freerank does not satisfy conditions (i) and (iii) of Theorem 3.9.

B. Vector Representation of Matrices

In this subsection, we define the group-norm in S^n that will allow to give an R-isomorphic isometry between S^n and $R^{m\times n}$.

***Definition** 3.11:* Let $\mathbf{u}=(u_1,\ldots,u_n)\in S^n$. By considering S as R-module, the number $\mu_R(\langle\{u_1,\ldots,u_n\}\rangle)$ is called the rank of $\mathbf{u}$ and denoted by $rank_R(\mathbf{u})$ or simply by $rank(\mathbf{u})$.

***Remark** 3.12:* Using the same arguments as in the proof of Theorem 3.9, we can show that the map $rank:S^n\to\mathbb{N}$ given by $\mathbf{u}\mapsto rank(\mathbf{u})$ is a group-norm.

The following proposition gives a relation between Definition 3.3 and Definition 3.11. Let $(\beta_1, \ldots, \beta_m)$ be a free basis of S as R-module. Consider $\mathbf{a} = (a_1, \ldots, a_n) \in S^n$. For $j = 1, \ldots, n$, a_j can be written as $a_j = \sum_{1 \le i \le m} a_{i,j}\beta_i$, where $a_{i,j} \in R$. The matrix $\mathbf{A_a} := \left(a_{i,j}\right)_{1\le i\le m,\ 1\le j\le n}$ is the matrix representation of $\mathbf{a}$ in the basis $(\beta_1, \ldots, \beta_m)$ over R. Analogous to [40], we have the following:

***Proposition** 3.13:* With the above notations, the map $S^n \to R^{m\times n}$ given by $\mathbf{a} \mapsto \mathbf{A_a}$ is an R-isomorphic isometry between the normed spaces $(S^n, rank)$ and $\left(R^{m\times n}, rank\right)$.

Proposition 3.8 can be interpreted in vector representation as follows.

***Proposition** 3.14:* Let $\mathbf{u} \in S^n$, $rank\,(\mathbf{u}) = t$.

1) There are $\mathbf{a} \in S^t$, $rank\,(\mathbf{a}) = t$, and $\mathbf{B} \in R^{t\times n}$, $freerank\,(\mathbf{B}) = t$, such that $\mathbf{u} = \mathbf{aB}$.
2) There are $\mathbf{a}' \in S^t$, $freerank\,(\mathbf{a}') = t$, and $\mathbf{B}' \in R^{t\times n}$, $rank\,(\mathbf{B}') = t$, such that $\mathbf{u} = \mathbf{a}'\mathbf{B}'$.

A direct consequence of Proposition 2.5 and Proposition 3.14 is the following:

***Proposition** 3.15:* Let $\mathbf{w} = (w_i)_{1\le i\le n} \in S^n$, $rank\,(\mathbf{w}) = r$. Then, there is a monic skew polynomial $P \in S[X, \sigma]$ of degree r such that $P\,(\mathbf{w}) = \mathbf{0}$.

As in the case of finite fields [36], the following proposition gives the link between the degree of a skew polynomial and the rank of its kernel.

***Proposition** 3.16:* Let $P = a_0 + a_1X + \cdots + a_\eta X^\eta \in S[X, \sigma]$ such that a_{i_0} is a unit for some $i_0 \in \{0, \ldots, \eta\}$. Then, $rank\,(\ker P) \le \deg\,(P)$.

Proof: Suppose that $\deg\,(P) < rank\,(\ker P)$. Set $r = rank\,(\ker P)$, then by Proposition 3.2 there is a free basis $\{b_i\}_{1\le i\le m}$ of S and the scalars $\lambda_1, \ldots, \lambda_r$ in R such that $\{\lambda_i b_i\}_{1\le i\le r}$ generates $\ker P$, with $\lambda_1|\lambda_2|\ldots|\lambda_r$. We then have $\lambda_r P\,(b_i) = 0$, for $i = 1, \ldots, r$. Hence, by Corollary 2.7, $\lambda_r P = 0$. This is clearly impossible because $\lambda_r \ne 0$ and a_{i_0} is a unit. Thus, $rank\,(\ker P) \le \deg\,(P)$. ■

***Remark** 3.17:* In Proposition 3.16, if all coefficients of P are non-units, then we can have $\deg\,(P) < rank\,(\ker P)$. Indeed, let $R = \mathbb{Z}_4$, $S = R\,[z]/\left(z^2 + z + 1\right)$ and $a = z + \left(z^2 + z + 1\right)$. Then, S is a Galois extension of R where the Galois group is generated by a power map $\sigma : a \mapsto a^2$. Set $P = 2X - 2 \in S\,[X, \sigma]$. Then, $\ker P$ is generated by 1 and $2a$. Thus, all coefficients of P are non-units and $\deg\,(P) < rank\,(\ker P)$.

***Remark** 3.18:* Proposition 2.6 and Proposition 3.16 are some of the main results that allow to extend the properties of Gabudulin codes to finite principal ideal rings. Note that if one of the automorphisms $\sigma_{(i)}$ is not a generator of the respective Galois group, then the ring S is not a Galois extension of R with Galois group $G_R\,(S)$ and therefore, as in [15], Proposition 2.6 and Proposition 3.16 will not be true in general. Indeed, consider the following example.

***Example** 3.19:* Let the finite field $\mathbb{F}_2$ and the Galois extension $\mathbb{F}_{2^4} = \mathbb{F}_2[z]/\left(z^4 + z^3 + 1\right)$. Set $a = z + \left(z^4 + z^3 + 1\right)$. Let $\theta = \left(\theta_{(1)}, \theta_{(2)}\right)$ be the map from $\mathbb{F}_{2^4} \times \mathbb{F}_{2^4}$ to $\mathbb{F}_{2^4} \times \mathbb{F}_{2^4}$, where $\theta_{(1)}\,(x) = x^2$ and $\theta_{(2)}\,(x) = x^4$ for all x in $\mathbb{F}_{2^4}$. The map θ is an $\mathbb{F}_2 \times \mathbb{F}_2$-automorphism of $\mathbb{F}_{2^4} \times \mathbb{F}_{2^4}$ and we have $\theta^2 = \left(\theta^2_{(1)}, id\right)$.

1) Let G be the group generated by θ. The set $\mathbb{F}_{2^4} \times \{0\}$ is a maximal ideal of $\mathbb{F}_{2^4} \times \mathbb{F}_{2^4}$ and for all $x \in \mathbb{F}_{2^4} \times \mathbb{F}_{2^4}$ we have $x - \theta^2\,(x) \in \mathbb{F}_{2^4} \times \{0\}$. Thus, by [34, Proposition 1.2(5), pp.80], $\mathbb{F}_{2^4} \times \mathbb{F}_{2^4}$ is not a Galois extension of $\mathbb{F}_2 \times \mathbb{F}_2$ with the group G.

2) Set $\mathbf{a} = (a, a)$ and $\mathbf{1} = (1, 1)$. Then $\left\{\mathbf{1}, \mathbf{a}, \mathbf{a}^2\right\}$ is linearly independent over $\mathbb{F}_2 \times \mathbb{F}_2$. Set

$$\mathbf{M} = \begin{pmatrix} \mathbf{1} & \mathbf{a} & \mathbf{a}^2 \\ \theta\,(\mathbf{1}) & \theta\,(\mathbf{a}) & \theta\left(\mathbf{a}^2\right) \\ \theta^2\,(\mathbf{1}) & \theta^2\,(\mathbf{a}) & \theta^2\left(\mathbf{a}^2\right) \end{pmatrix} = \begin{pmatrix} (1,1) & (a,a) & \left(a^2, a^2\right) \\ (1,1) & \left(a^2, a^4\right) & \left(a^4, a^8\right) \\ (1,1) & \left(a^4, a\right) & \left(a^8, a^2\right) \end{pmatrix}$$

By [42, Corollary 2.8], the matrix $\mathbf{M}$ is not invertible because the rows of the matrix

$$\begin{pmatrix} 1 & a & a^2 \\ 1 & a^4 & a^8 \\ 1 & a & a^2 \end{pmatrix}$$

are not linearly independent.

3) Let $P = X - (1, 1)$ in $\left(\mathbb{F}_{2^4} \times \mathbb{F}_{2^4}\right)[X, \theta]$. The set $\ker P$ is generated by $(1, 1)$ and $\left(0, a + a^4\right)$. Thus, $rank\,(\ker P) > \deg\,(P)$.

C. Matrix and Vector Representation of Rank-Metric Codes

Analogous to the case of finite fields [1]–[3], we give the following definitions.

In matrix representation, rank codes are defined as subsets of a normed space $\left(R^{m\times n},\ rank\right)$, where the norm of a matrix $\mathbf{A} \in R^{m\times n}$ is the rank of $\mathbf{A}$ over R. The rank distance between two matrices $\mathbf{A}$ and $\mathbf{B}$ is the rank of their difference $rank\,(\mathbf{A} - \mathbf{B})$. The rank distance of a matrix rank code $\mathcal{M} \subset R^{m\times n}$ is defined as the minimal pairwise distance:

$$d\,(\mathcal{M}) = \min\left\{rank\,(\mathbf{A} - \mathbf{B}) :\ \mathbf{A},\ \mathbf{B} \in \mathcal{M},\ \mathbf{A} \ne \mathbf{B}\right\}.$$

A matrix rank code $\mathcal{M} \subset R^{m\times n}$ is called R-linear if $\mathcal{M}$ is a submodule of $R^{m\times n}$.

In vector representation, rank codes are defined as subsets of a normed S-module space $(S^n,\ rank)$, where the norm of a vector $\mathbf{u} \in S^n$ is the rank of $\mathbf{u}$. The rank distance of two vectors $\mathbf{u}$ and $\mathbf{v}$ is the rank of their difference $rank\,(\mathbf{u} - \mathbf{v})$. The rank distance of a vector rank code $\mathcal{C} \subset S^n$ is defined as the minimal pairwise distance:

$$d\,(\mathcal{C}) = \min\left\{rank\,(\mathbf{u} - \mathbf{v}) :\ \mathbf{u},\ \mathbf{v} \in \mathcal{C},\ \mathbf{u} \ne \mathbf{v}\right\}.$$

A vector rank code $\mathcal{C} \subset S^n$ is called linear if $\mathcal{C}$ is a submodule of S-module S^n, furthermore if $\mathcal{C}$ is a free submodule of S^n then $\mathcal{C}$ is called a free rank code.

Let $\mathcal{C} \subset S^n$ be a linear rank code. The number $\mu_S\,(\mathcal{C})$, denoted by $rank_S\,(\mathcal{C})$ or simply by $rank\,(\mathcal{C})$, is called the rank of $\mathcal{C}$. A generator matrix of $\mathcal{C}$ is a $rank(\mathcal{C}) \times n$ matrix over S whose rows generate $\mathcal{C}$. The inner product of two vectors $\mathbf{u} = (u_1, \ldots, u_n) \in S^n$ and $\mathbf{v} = (v_1, \ldots, v_n) \in S^n$ is defined by

$$\mathbf{u} \cdot \mathbf{v} = u_1v_1 + \cdots + u_nv_n.$$

The dual of $\mathcal{C}$ is the submodule of S^n defined by

$$\mathcal{C}^{\perp} = \left\{\mathbf{u} \in S^n : \mathbf{u} \cdot \mathbf{v} = 0, \quad \text{for every } \mathbf{v} \in \mathcal{C}\right\}.$$

A parity-check matrix of $\mathcal{C}$ is a generator matrix of $\mathcal{C}^{\perp}$.

Note that by Proposition 3.13, there exists a relation between the matrix representation and the vector representation. As in the case of finite fields [1]–[3], the following proposition establishes the rank-metric Singleton bound.

***Proposition** 3.20:* (Singleton bound)

Let $\mathcal{M} \subset R^{m\times n}$ be a rank code of rank distance d, then

$$|\mathcal{M}| \leq |R|^{\min\{m(n-d+1),\ n(m-d+1)\}}$$

where $|\mathcal{M}|$ and $|R|$ denote the cardinality of $\mathcal{M}$ and R respectively.

Proof: The proof is similar to that in the case of finite fields, see e.g. [43, Theorem 1]. ■

***Definition** 3.21:* Let $\mathcal{M} \subset R^{m\times n}$ and $\mathcal{C} \subset S^n$ be the rank codes of rank distance d such that

$$|\mathcal{M}| = |\mathcal{C}| = |R|^{\min\{m(n-d+1),\ n(m-d+1)\}},$$

then we say that $\mathcal{M}$ and $\mathcal{C}$ are maximum rank distance codes, or, MRD codes.

In finite fields, Gabidulin codes are MRD codes [1]–[3]. We will prove that this property extends to finite principal ideal rings.

D. Gabidulin Codes

Let $\mathbf{g} = (g_1, \ldots, g_n) \in S^n$, such that $\{g_1, \ldots, g_n\}$ is linearly independent over R. Let k be an integer such that $0 < k \leq n$.

***Definition** 3.22:* (Gabidulin Codes)

A Gabidulin code $Gab_k(\mathbf{g})$ of length n, dimension k and support $\mathbf{g}$ is the S-module given by:

$$Gab_k(\mathbf{g}) = \{f(\mathbf{g}) : f \in S[X,\sigma]_{<k}\}.$$

***Proposition** 3.23:* The Gabidulin code $Gab_k(\mathbf{g})$ is a free rank code of rank k with a generator matrix

$$\mathbf{G} = \begin{pmatrix} \sigma^0(g_1) & \cdots & \sigma^0(g_n) \\ \vdots & \ddots & \vdots \\ \sigma^{k-1}(g_1) & \cdots & \sigma^{k-1}(g_n) \end{pmatrix}.$$

Proof: The rows of $\mathbf{G}$ generate $Gab_k(\mathbf{g})$. By Proposition 2.6 and [42, Corollary 2.8], the rows of $\mathbf{G}$ are linearly independent over S, thus $Gab_k(\mathbf{g})$ is a free code of rank k. ■

***Theorem** 3.24:* (a) The rank distance, d, of $Gab_k(\mathbf{g})$ is given by $d = n - k + 1$.

(b) $Gab_k(\mathbf{g})$ is an MRD code.

Proof: Using Corollary 2.7 and Proposition 3.15, the proof is similar to that of [44, Proposition 7.]. ■

***Theorem** 3.25:* Let $\left(\gamma_{i,j}\right)_{1\leq i\leq n, 1\leq j\leq n}$ be the inverse of the matrix $\left(\sigma^i\left(g_j\right)\right)_{0\leq i\leq n-1, 1\leq j\leq n}$. Set

$$h_i := \sigma^{-n+k+1}\left(\gamma_{i,n}\right), \qquad i = 1, \ldots, n.$$

Then, the family $\{h_1, \ldots, h_n\}$ is linearly independent over R and a parity-check matrix of $Gab_k(\mathbf{g})$ is

$$\mathbf{H} = \begin{pmatrix} \sigma^0(h_1) & \cdots & \sigma^0(h_n) \\ \vdots & \ddots & \vdots \\ \sigma^{n-k-1}(h_1) & \cdots & \sigma^{n-k-1}(h_n) \end{pmatrix}.$$

Proof: The product of the two matrices $\left(\sigma^i\left(g_j\right)\right)_{0\leq i\leq n-1, 1\leq j\leq n}$ and $\left(\sigma^{1-n+j}\left(\gamma_{i,n}\right)\right)_{1\leq i\leq n, 0\leq j\leq n-1}$ is a lower unitriangular matrix. Thus, the matrix $\left(\sigma^{1-n+j}\left(\gamma_{i,n}\right)\right)_{1\leq i\leq n, 0\leq j\leq n-1}$ is invertible. Therefore, by Proposition 2.6, $\{\gamma_{1,n}, \ldots, \gamma_{n,n}\}$ is linearly independent over R. Consequently, $\{h_1, \ldots, h_n\}$ is linearly independent over R. Thus, the rows of the matrix $\mathbf{H}$ are linearly independent over S and $\mathbf{G}\mathbf{H}^T = \mathbf{0}$. Since $Gab_k(\mathbf{g})$ is a free code of length n and the rank k, by [42, Proposition 2.9], $Gab_k(\mathbf{g})^{\perp}$ is a free code of rank $n-k$. Consequently, $\mathbf{H}$ is a parity-check matrix of $Gab_k(\mathbf{g})$. ■

In [45], Loidreau showed that decoding of Gabidulin codes can be translated to the problem of reconstruction of skew polynomials. In the input of decoding algorithm given in [45, page 40], it is assumed that the rank of the error is less than or equal to the error-correcting capability of the code. But in practice, the receiver does not know the rank of the error. In [44], Augot et al. gave a similar algorithm without this condition. We will prove that [44, Algorithm 2] can be extended to finite principal ideal rings.

For the remainder of this section, let $t_0 := \lfloor (n-k)/2 \rfloor$ be the error correction capability of the Gabidulin code $Gab_k(\mathbf{g})$. Similarly to [45, Proposition 1 and Proposition 2], we give the following:

***Lemma** 3.26:* Let $\mathbf{y} \in S^n$ be a received word of the Gabidulin code $Gab_k(\mathbf{g})$. Assume that there is $f \in S[X,\sigma]_{<k}$ such that $rank(\mathbf{y} - f(\mathbf{g})) \leq t_0$. Then, the following linear equation

$$\begin{pmatrix} \mathbf{A}_1 & \mathbf{A}_2 \end{pmatrix} \begin{pmatrix} \mathbf{u}^T \\ \mathbf{v}^T \end{pmatrix} = \begin{pmatrix} \sigma^{t_0}(y_1) \\ \vdots \\ \sigma^{t_0}(y_n) \end{pmatrix} \tag{1}$$

with unknowns $\mathbf{u} = \left(u_0, \ldots, u_{k+t_0-1}\right)$ and $\mathbf{v} = \left(v_0, \ldots, v_{t_0-1}\right)$ has a solution, where

$$\mathbf{A}_1 = \begin{pmatrix} \sigma^0(g_1) & \cdots & \sigma^{k+t_0-1}(g_1) \\ \vdots & \ddots & \vdots \\ \sigma^0(g_n) & \cdots & \sigma^{k+t_0-1}(g_n) \end{pmatrix}$$

and

$$\mathbf{A}_2 = \begin{pmatrix} -\sigma^0(y_1) & \cdots & -\sigma^{t_0-1}(y_1) \\ \vdots & \ddots & \vdots \\ -\sigma^0(y_n) & \cdots & -\sigma^{t_0-1}(y_n) \end{pmatrix}.$$

Moreover, if $\mathbf{u} = \left(u_0, \ldots, u_{k+t_0-1}\right)$ and $\mathbf{v} = \left(v_0, \ldots, v_{t_0-1}\right)$ are a solution of this equation, then $U = Vf$ where $U = u_0 + u_1X + \cdots + u_{k+t_0-1}X^{k+t_0-1}$ and $V = v_0 + v_1X + \cdots + v_{t_0-1}X^{t_0-1} + X^{t_0}$.

Proof: Set $t = rank(\mathbf{y} - f(\mathbf{g}))$. By Proposition 3.15, there is a monic skew polynomials $W \in S[X,\sigma]$ of degree t such that $W(\mathbf{y} - f(\mathbf{g})) = \mathbf{0}$. Therefore, $X^{t_0-t}W(\mathbf{y}) =$

$X^{t_0-t}W(f(\mathbf{g}))$. Set $X^{t_0-t}Wf = u_0 + u_1X + \cdots + u_{k+t_0-1}X^{k+t_0-1}$ and $X^{t_0-t}W = v_0+v_1X+\cdots+v_{t_0-1}X^{t_0-1}+X^{t_0}$. Then, $\mathbf{u} = \left(u_0, \ldots, u_{k+t_0-1}\right)$ and $\mathbf{v} = \left(v_0, \ldots, v_{t_0-1}\right)$ are a solution of (1).

Now, let $\mathbf{u} = \left(u_0, \ldots, u_{k+t_0-1}\right)$ and $\mathbf{v} = \left(v_0, \ldots, v_{t_0-1}\right)$ be a solution of (1). Set $U = u_0 + u_1X + \cdots + u_{k+t_0-1}X^{k+t_0-1}$ and $V = v_0 + v_1X + \cdots + v_{t_0-1}X^{t_0-1} + X^{t_0}$. Then, we have $V(\mathbf{y}) = U(\mathbf{g})$. Since $rank(\mathbf{y} - f(\mathbf{g})) \leq t_0$, we also have $rank(V(\mathbf{y} - f(\mathbf{g}))) \leq t_0$, that is, $rank((U - Vf)(\mathbf{g})) \leq t_0$. Thus, By Proposition 3.15, there is a monic skew polynomial $L \in S[X,\sigma]_{<t_0+1}$ such that $(L(U - Vf))(\mathbf{g}) = \mathbf{0}$. As $\deg(L(U - Vf)) \leq 2t_0 + k - 1 \leq n - 1$, by Corollary 2.7, $L(U - Vf) = 0$. Since L is monic, we have $U - Vf = 0$. ■

Lemma 3.26 allows to give Algorithm 1.

Algorithm 1 Decoding Gabidulin Codes up to Half the Minimum Distance

Input: a received word $\mathbf{y} \in S^n$ of the Gabidulin code $Gab_k(\mathbf{g})$.
Output: $f \in S[X,\sigma]_{<k}$ such that $rank(\mathbf{y} - f(\mathbf{g})) \leq \lfloor (n-k)/2 \rfloor$ or "decoding failure".

Solve linear equation (1)
if *(1) has no solution* **then**
  return *"decoding failure"*
else
  Set $U = u_0 + u_1X + \cdots + u_{k+t_0-1}X^{k+t_0-1}$ and $V = v_0 + v_1X + \cdots + v_{t_0-1}X^{t_0-1} + X^{t_0}$ where $\mathbf{u} = \left(u_0, \ldots, u_{k+t_0-1}\right)$ and $\mathbf{v} = \left(v_0, \ldots, v_{t_0-1}\right)$ are a solution of (1).
  Compute the quotient Q and the remainder P on the left Euclidean division of U by V in $S[X,\sigma]$.
  if $P \neq 0$ **then**
    return *"decoding failure"*
  else
    return Q

***Theorem** 3.27:* Let $\mathbf{y} \in S^n$ be a received word of the Gabidulin code $Gab_k(\mathbf{g})$. Let $f \in S[X,\sigma]$. Then, Algorithm 1 returns f if and only if $\deg(f) < k$ and $rank(\mathbf{y} - f(\mathbf{g})) \leq t_0$.

Proof: Assume that Algorithm 1 returns f, then $U = Vf$ where U and V are as in Algorithm 1. Since $\deg(U) \leq k + t_0 - 1$, we have $\deg(f) < k$. As $V(\mathbf{y}) = U(\mathbf{g})$, we also have $V(\mathbf{y} - f(\mathbf{g})) = \mathbf{0}$. Thus, by Proposition 3.16, $rank(\mathbf{y} - f(\mathbf{g})) \leq t_0$. The converse is given by Lemma 3.26. ■

Recall that one can use the Smith normal form to solve (1). In the next section we will show that one can also use the iterative method similarly to [26].

IV. Interleaved Gabidulin Codes

Recall that an interleaved Gabidulin code is a direct sum of several Gabidulin codes. In this section, we give the properties of interleaved Gabidulin codes, establish a key equation and give an algorithm to solve it.

A. Description

Let $l \in \{1, \ldots, \ell\}$. Let $n^{(l)}$ and $k^{(l)}$ be the integers such that $0 < k^{(l)} \leq n^{(l)} \leq m$.

Let $\mathbf{g}^{(l)} = \left(g_1^{(l)}, \ldots, g_{n^{(l)}}^{(l)}\right)$, where $\{g_1^{(l)}, \ldots, g_{n^{(l)}}^{(l)}\}$ is a R-linear independent subset of S. The rank distance of $Gab_{k^{(l)}}\left(\mathbf{g}^{(l)}\right)$ is denoted by $d^{(l)}$. The concatenation of ℓ vectors $\mathbf{c}^{(1)} \in S^{n^{(1)}}, \ldots, \mathbf{c}^{(\ell)} \in S^{n^{(\ell)}}$ is denoted by $\left(\mathbf{c}^{(1)} \cdots \mathbf{c}^{(\ell)}\right) \in S^{n^{(1)}+\cdots+n^{(\ell)}}$.

***Definition** 4.1:* An interleaved Gabidulin code, $IGab_{\left(k^{(1)},\ldots,k^{(\ell)}\right)}\left(\mathbf{g}^{(1)}, \ldots, \mathbf{g}^{(\ell)}\right)$, is the set

$$\left\{\left(\mathbf{c}^{(1)} \cdots \mathbf{c}^{(\ell)}\right) : \mathbf{c}^{(l)} \in Gab_{k^{(l)}}\left(\mathbf{g}^{(l)}\right),\ l = 1, \ldots, \ell\right\}.$$

We observe that if $\ell = 1$ then an interleaved Gabidulin code is a Gabidulin code.

***Proposition** 4.2:* The interleaved Gabidulin code $IGab_{\left(k^{(1)},\ldots,k^{(\ell)}\right)}\left(\mathbf{g}^{(1)}, \ldots, \mathbf{g}^{(\ell)}\right)$ is a free linear rank code of rank $k^{(1)} + \cdots + k^{(\ell)}$ and rank distance $\min_{l\in\{1,\ldots,\ell\}}\left\{d^{(l)}\right\}$.

Proof: The proof is similar to that of [46, Lemma 2.17]. ■

***Corollary** 4.3:* If $k^{(l)} = k^{(1)}$ and $n^{(l)} = m$, for $l = 1, \ldots, \ell$, then $IGab_{\left(k^{(1)},\ldots,k^{(\ell)}\right)}\left(\mathbf{g}^{(1)}, \ldots, \mathbf{g}^{(\ell)}\right)$ is an MRD code.

***Notation** 4.4:* Recall that for $\mathbf{U} \in S[X,\sigma]^{\ell+1}$, the l-th component of $\mathbf{U}$ is denoted by $U^{(l)}$, for l in $\{0, \ldots, \ell\}$, i.e. $\mathbf{U} = \left(U^{(0)}, \ldots, U^{(\ell)}\right)$. In order to simplify the notations, the element $\left(A^{(1)}, \ldots, A^{(\ell)}\right)$ in $S[X,\sigma]^{\ell}$ is denoted by $\hat{\mathbf{A}}$.

For the remainder of this section, let $\mathbf{y} = \left(\mathbf{y}^{(1)} \cdots \mathbf{y}^{(\ell)}\right) \in S^{n^{(1)}+\cdots+n^{(\ell)}}$ be a received word of the interleaved Gabidulin code $IGab_{\left(k^{(1)},\ldots,k^{(\ell)}\right)}\left(\mathbf{g}^{(1)}, \ldots, \mathbf{g}^{(\ell)}\right)$. The following theorem is the analogue of [26, Theorem 12].

***Theorem** 4.5:* Let $\tau \in \mathbb{N}$. Then, the following statements are equivalent.

(i) There is $\mathbf{c} \in IGab_{\left(k^{(1)},\ldots,k^{(\ell)}\right)}\left(\mathbf{g}^{(1)}, \ldots, \mathbf{g}^{(\ell)}\right)$ such that $rank(\mathbf{y} - \mathbf{c}) \leq \tau$.

(ii) There is $\mathbf{U} \in S[X,\sigma]^{\ell+1}$ such that:
1) $U^{(0)}\left(\mathbf{y}^{(l)}\right) = U^{(l)}\left(\mathbf{g}^{(l)}\right)$, for $l = 1, \ldots, \ell$;
2) $\deg\left(U^{(l)}\right) - k^{(l)} \leq \deg\left(U^{(0)}\right) - 1$, for $l = 1, \ldots, \ell$;
3) $U^{(0)}$ is monic;
4) $\deg\left(U^{(0)}\right) \leq \tau$;
5) the remainder of the left Euclidean division of $U^{(l)}$ by $U^{(0)}$ is equal to zero, for $l = 1, \ldots, \ell$.

Proof: Using Proposition 3.16 and Proposition 3.15, the proof is similar to that of [26, Theorem 12] and [4]. ■

***Definition** 4.6:* (**the key equation**)

We say that $\mathbf{U} \in S[X,\sigma]^{\ell+1}$ is a solution of the key equation if :

- $U^{(0)}\left(\mathbf{y}^{(l)}\right) = U^{(l)}\left(\mathbf{g}^{(l)}\right)$, for $l = 1, \ldots, \ell$;
- $\deg\left(U^{(l)}\right) - k^{(l)} \leq \deg\left(U^{(0)}\right) - 1$, for $l = 1, \ldots, \ell$.
- $U^{(0)}$ is monic;

A solution $\mathbf{U}$ is called minimal if $\deg\left(U^{(0)}\right)$ is minimal.

In finite fields, the resolution of the key equation given in Definition 4.6 is equivalent to the problem of multi-sequence generalized linear skew-feedback shift register introduced in [47]. In [47], Puchinger et al. solved this problem using row reduction. We will solve the key equation using the iterative method introduced in [48], because it is easy to extend this method to modules and finite rings [49]–[51]. Note that in [52], Bartz and Wachter-Zeh used this iterative method for decoding interleaved subspace and Gabidulin codes, because its complexity is better than Gaussian elimination. Further, it allows to compute a minimal Gröbner basis for the interpolation module.

B. Iterative Solving the key Equation

Similar to [26], [50], we give an iterative algorithm that allows to solve the key equation. Recall that the elements a and b in S are said to be associated if $b = ua$ for some unit $u \in S$.

Notation 4.7: Since associatedness is an equivalence relation on S,

- the equivalent class of $a \in S$ is denoted by $[a]$;
- a complete set of representatives of the equivalence classes is denoted by $[S]$, without loss of generality, assume that $1 \in [S]$;
- we denote by $[S]^* := [S]\setminus\{0\}$.

As $S = S_{(1)} \times \cdots \times S_{(\rho)}$, where $S_{(j)}$ is a finite chain ring and a generator of its maximal ideal is in $R_{(j)}$, we have the following:

Lemma 4.8: For all $a \in S$, a and $\sigma(a)$ are associated.

Notation 4.9: Let $\mathbf{y} = \left(\mathbf{y}^{(1)} \cdots \mathbf{y}^{(\ell)}\right) \in S^{n^{(1)}+\cdots+n^{(\ell)}}$ be a received word of the interleaved Gabidulin code $IGab_{(k^{(1)},\ldots,k^{(\ell)})}\left(\mathbf{g}^{(1)},\ldots,\mathbf{g}^{(\ell)}\right)$. Set $\mathbf{g} = \left(\mathbf{g}^{(1)} \cdots \mathbf{g}^{(\ell)}\right)$. We denote by $M[\mathbf{y},\mathbf{g}]$ the set of all $\mathbf{U}$ in $S[X,\sigma]^{\ell+1}$ such that $U^{(0)}\left(\mathbf{y}^{(l)}\right) = U^{(l)}\left(\mathbf{g}^{(l)}\right)$, for $l = 1,\ldots,\ell$, that is, $U^{(0)}\left(y_i^{(l)}\right) = U^{(l)}\left(g_i^{(l)}\right)$, for $l = 1,\ldots,\ell$ and $i = 1,\ldots,n^{(l)}$.

The set $M[\mathbf{y},\mathbf{g}]$ is a $S[X,\sigma]$-submodule of $S[X,\sigma]^{\ell+1}$ and by Definition 4.6, all the solutions of the key equation are in $M[\mathbf{y},\mathbf{g}]$. Therefore, to find these solutions, just find a Gröbner basis for $M[\mathbf{y},\mathbf{g}]$ with a monomial order $\succeq$ that we will specify later. To compute a Gröbner basis for $M[\mathbf{y},\mathbf{g}]$, we will use the iterative method described in [49].

Notation 4.10: Set $n^{(0)} := 0$. We define by induction the subsets $M[\mathbf{y},\mathbf{g}]_{(l,i)}$ as following:
$M[\mathbf{y},\mathbf{g}]_{(0,0)} = S[X,\sigma]^{\ell+1}$ and for all $(l,i) \in \{1,\ldots,\ell\} \times \left\{1,\ldots,n^{(l)}\right\}$, $M[\mathbf{y},\mathbf{g}]_{(l,i)}$ is the set of all $\mathbf{U}$ in $M[\mathbf{y},\mathbf{g}]_{(\underline{l},\underline{i})}$ such that $U^{(0)}\left(y_i^{(l)}\right) = U^{(l)}\left(g_i^{(l)}\right)$, where

$$(\underline{l},\underline{i}) = \begin{cases} \left(l-1, n^{(l-1)}\right) \text{ if } i = 1 \\ (l, i-1) \text{ else} \end{cases}$$

We have $M[\mathbf{y},\mathbf{g}]_{(0,0)} \supset M[\mathbf{y},\mathbf{g}]_{(1,1)} \supset \cdots \supset M[\mathbf{y},\mathbf{g}]_{(1,n^{(1)})} \supset M[\mathbf{y},\mathbf{g}]_{(2,1)} \supset \cdots \supset M[\mathbf{y},\mathbf{g}]_{(2,n^{(2)})} \supset \cdots \supset M[\mathbf{y},\mathbf{g}]_{(\ell,1)} \supset \cdots \supset M[\mathbf{y},\mathbf{g}]_{(\ell,n^{(\ell)})} = M[\mathbf{y},\mathbf{g}]$. Note that as in [50] a Gröbner basis for $S[X,\sigma]^{\ell+1}$ is $\mathcal{B}_{(0,0)} := \left\{s\mathbf{e}^{(r)}\right\}_{0\le r\le\ell,\ s\in[S]^*}$. So, we will compute a Gröbner basis, $\mathcal{B} = \left\{\mathbf{V}_{(r,s)}\right\}_{0\le r\le\ell,\ s\in[S]^*}$, for $M[\mathbf{y},\mathbf{g}]$ which has the same properties as $\mathcal{B}_{(0,0)}$, that is, for all (r,s), $ind(lm\left(\mathbf{V}_{(r,s)}\right)) = r$, $lc\left(\mathbf{V}_{(r,s)}\right) \in [s]$, and $\deg\left(\mathbf{V}_{(r,s)}\right)$ is minimal among the degree of all $\mathbf{U} \in M[\mathbf{y},\mathbf{g}]$ with $ind(lm(\mathbf{U})) = r$, $lc(\mathbf{U}) \in [s]$.

Let $(l,i) \in \{1,\ldots,\ell\} \times \left\{1,\ldots,n^{(l)}\right\}$. Assume that $M[\mathbf{y},\mathbf{g}]_{(\underline{l},\underline{i})}$ has a Gröbner basis $\mathcal{B}_{(\underline{l},\underline{i})} = \left\{\mathbf{V}_{(r,s)}\right\}_{0\le r\le\ell,\ s\in[S]^*}$ such that for all (r,s), $ind(lm\left(\mathbf{V}_{(r,s)}\right)) = r$, $lc\left(\mathbf{V}_{(r,s)}\right) \in [s]$, and $\deg\left(\mathbf{V}_{(r,s)}\right)$ is minimal among the degree of all $\mathbf{U} \in M[\mathbf{y},\mathbf{g}]_{(\underline{l},\underline{i})}$ with $ind(lm(\mathbf{U})) = r$, $lc(\mathbf{U}) \in [s]$.

- Let $\mathcal{J}_{(r,s)}$ be the set of all $(r',s') \in \{0,\ldots,\ell\} \times [S]^*$ such that $lm\left(\mathbf{V}_{(r',s')}\right) \prec lm\left(\mathbf{V}_{(r,s)}\right)$.
- Let $D_{(l,i)} : M[\mathbf{y},\mathbf{g}]_{(\underline{l},\underline{i})} \longrightarrow S$ be defined as

$$D_{(l,i)}(\mathbf{U}) = U^{(0)}\left(y_i^{(l)}\right) - U^{(l)}\left(g_i^{(l)}\right).$$

- The discrepancy of $\mathbf{V}_{(r,s)}$ is given by

$$\Delta_{(r,s)} := D_{(l,i)}\left(\mathbf{V}_{(r,s)}\right).$$

- Let $b_{(r,s)} \in S$ such that

$$\sigma\left(\Delta_{(r,s)}\right) - b_{(r,s)}\Delta_{(r,s)} = 0.$$

Lemma 4.11: With the above notations,

(a) $D_{(l,i)}$ is an S-module homomorphism;
(b) $M[\mathbf{y},\mathbf{g}]_{(l,i)} = \left\{\mathbf{U} \in M[\mathbf{y},\mathbf{g}]_{(\underline{l},\underline{i})} : D_{(l,i)}(\mathbf{U}) = 0\right\}$;
(c) $\left(X - b_{(r,s)}\right)\mathbf{V}_{(r,s)} \in M[\mathbf{y},\mathbf{g}]_{(l,i)}$.

Using a Gröbner basis, $\left\{\mathbf{V}_{(r,s)}\right\}_{0\le r\le\ell,\ s\in[S]^*}$, for $M[\mathbf{y},\mathbf{g}]_{(\underline{l},\underline{i})}$, we now show how one can compute a Gröbner basis for $M[\mathbf{y},\mathbf{g}]_{(l,i)}$. Let $\left\{\mathbf{V}'_{(r,s)}\right\}_{0\le r\le\ell,\ s\in[S]^*} \subset S[X,\sigma]^{\ell+1}$ be defined as :

- if $\Delta_{(r,s)} = 0$ then

$$\mathbf{V}'_{(r,s)} := \mathbf{V}_{(r,s)} \tag{2}$$

- if $\Delta_{(r,s)} \neq 0$ and there exist $\theta_{(r';s')} \in S$, $(r',s') \in \mathcal{J}_{(r,s)}$ such that

$$\Delta_{(r,s)} + \sum_{(r',s')\in\mathcal{J}_{(r,s)}} \theta_{(r',s')}\Delta_{(r',s')} = 0 \tag{3}$$

then

$$\mathbf{V}'_{(r,s)} := \mathbf{V}_{(r,s)} + \sum_{(r',s')\in\mathcal{J}_{(r,s)}} \theta_{(r',s')}\mathbf{V}_{(r',s')} \tag{4}$$

- otherwise,

$$\mathbf{V}'_{(r,s)} := \left(X - b_{(r,s)}\right)\mathbf{V}_{(r,s)} \tag{5}$$

Proposition 4.12: Let $\left\{\mathbf{V}'_{(r,s)}\right\}_{0\le r\le\ell,\ s\in[S]^*}$ be the subset of $S[X,\sigma]^{\ell+1}$ computed using (2), (4) and (5). Then, $\left\{\mathbf{V}'_{(r,s)}\right\}_{0\le r\le\ell,\ s\in[S]^*}$ is a Gröbner basis for $M[\mathbf{y},\mathbf{g}]_{(l,i)}$ and for all (r,s), $ind(lm\left(\mathbf{V}'_{(r,s)}\right)) = r$, $lc\left(\mathbf{V}'_{(r,s)}\right) \in [s]$, and $\deg\left(\mathbf{V}'_{(r,s)}\right)$ is minimal among the degree of all $\mathbf{U} \in M[\mathbf{y},\mathbf{g}]_{(l,i)}$ with $ind(lm(\mathbf{U})) = r$, $lc(\mathbf{U}) \in [s]$.

Proof: By the definition of $\mathbf{V}'_{(r,s)}$, we have $\mathbf{V}'_{(r,s)} \in M[\mathbf{y},\mathbf{g}]_{(l,i)}$, $ind(lm\left(\mathbf{V}'_{(r,s)}\right)) = r$, $lc\left(\mathbf{V}'_{(r,s)}\right) \in [s]$. We now prove that $\deg\left(\mathbf{V}'_{(r,s)}\right)$ is minimal among the degree of all $\mathbf{U} \in M[\mathbf{y},\mathbf{g}]_{(l,i)}$ with $ind(lm(\mathbf{U})) = r$, $lc(\mathbf{U}) \in [s]$. If $\mathbf{V}'_{(r,s)}$ is defined as in (2) or (4), then the result follows. Assume that $\mathbf{V}'_{(r,s)}$ is defined as in (5) and that there is $\mathbf{W} \in M[\mathbf{y},\mathbf{g}]_{(l,i)}$ such that $ind(lm(\mathbf{W})) = r$, $lc(\mathbf{W}) \in [s]$ and $\deg(\mathbf{W}) < \deg\left(\mathbf{V}'_{(r,s)}\right)$. Then, since $\mathbf{W} \in M[\mathbf{y},\mathbf{g}]_{(\underline{l},\underline{i})}$ and $\deg\left(\mathbf{V}'_{(r,s)}\right) = \deg\left(\mathbf{V}_{(r,s)}\right) + 1$, we have $\deg(\mathbf{W}) =$

$\deg\left(\mathbf{V}_{(r,s)}\right)$. Therefore, as $lc\,(\mathbf{W}) \in [s]$ and $lc\left(\mathbf{V}_{(r,s)}\right) \in [s]$, there is $a \in S$ such that

$$lm\left(\mathbf{V}_{(r,s)} - a\mathbf{W}\right) \prec lm\left(\mathbf{V}_{(r,s)}\right).$$

Consequently, by Proposition 2.11, we have

$$\mathbf{V}_{(r,s)} - a\mathbf{W} = \sum_{(r',s')\ \in \mathcal{J}_{(r,s)}} h_{(r',s')}\mathbf{V}_{(r',s')}$$

where $h_{(r',s')} \in S[X,\sigma]$. By the right Euclidean division of $h_{(r',s')}$ by $X - b_{(r',s')}$ there exist $Q_{(r',s')} \in S[X,\sigma]$ and $\lambda_{(r',s')} \in S$ such that

$$h_{(r',s')} = Q_{(r',s')}\left(X - b_{(r',s')}\right) + \lambda_{(r',s')}.$$

Hence, we have

$$\begin{aligned}\mathbf{V}_{(r,s)} - a\mathbf{W} = &\sum_{(r',s')\ \in \mathcal{J}_{(r,s)}} Q_{(r',s')}\left(X - b_{(r',s')}\right)\mathbf{V}_{(r',s')} \\ &+ \sum_{(r',s')\ \in \mathcal{J}_{(r,s)}} \lambda_{(r',s')}\mathbf{V}_{(r',s')}.\end{aligned}$$

Consequently, by Lemma 4.11,

$$D_{(l,i)}\left(\mathbf{V}_{(r,s)}\right) = \sum_{(r',s') \in \mathcal{J}_{(r,s)}} \lambda_{(r',s')} D_{(l,i)}\left(\mathbf{V}_{(r',s')}\right)$$

This contradicts the definition of $\mathbf{V}'_{(r,s)}$. Thus, the result follows.

Now we prove that $\left\{\mathbf{V}'_{(r,s)}\right\}_{0 \le r \le \ell,\ s \in [S]^*}$ is a Gröbner basis for $M[\mathbf{y},\mathbf{g}]_{(l,i)}$. Let $\mathbf{U} \in M[\mathbf{y},\mathbf{g}]_{(l,i)}$, $r = ind(lm\,(\mathbf{U}))$, $s \in [S]^*$ such that $lc\,(\mathbf{U}) \in [s]$ and $\alpha = \deg(\mathbf{U}) - \deg\left(\mathbf{V}'_{(r,s)}\right)$. Then,

$$lm\,(\mathbf{U}) = X^{\alpha} lm\left(\mathbf{V}'_{(r,s)}\right)$$

and

$$lc\,(\mathbf{U}) \in \left\langle \sigma^{\alpha}\left(lc\left(\mathbf{V}'_{(r,s)}\right)\right)\right\rangle.$$

Thus, the result follows. ■

Proposition 4.12 justifies Algorithm 2.

Remark 4.13: Since $S = S_{(1)} \times \cdots \times S_{(\rho)}$, where $S_{(j)}$ is a finite chain ring, the equation (3) is easy to solve in $S_{(j)}$. Indeed, in $S_{(j)}$ this equation is equivalent to: $\Delta_{(r',s')}$ divides $\Delta_{(r,s)}$ for some $\left(r',s'\right)$ in $\mathcal{J}_{(r,s)}$. Thus, analogous to [53, Algorithm VI.5], it is easy to compute a Gröbner basis of Algorithm 2 in $S_{(j)}[X,\sigma_{(j)}]^{\ell+1}$, and then to apply the "strong join" method described in [54] to obtain a Gröbner basis in $S[X,\sigma]^{\ell+1}$.

Note that the monomial order of Algorithm 2 is not specified. We now define a monomial order that will allow to give the solutions of the key equation.

Definition 4.14: Set $k^{(0)} := 1$. The relation $\preceq_{\left(k^{(0)},\ldots,k^{(\ell)}\right)}$ is defined on the monomial of $S[X,\sigma]^{\ell+1}$ by:

$$X^{\alpha_1}\mathbf{e}^{(l_1)} \preceq_{\left(k^{(0)},\ldots,k^{(\ell)}\right)} X^{\alpha_2}\mathbf{e}^{(l_2)}$$

if and only if $\alpha_1 - k^{(l_1)} < \alpha_2 - k^{(l_2)}$ or [$\alpha_1 - k^{(l_1)} = \alpha_2 - k^{(l_2)}$ and $l_1 \ge l_2$].

By [55, Theorem 29], the relation $\preceq_{\left(k^{(0)},\ldots,k^{(\ell)}\right)}$ is a monomial order.

Algorithm 2 A Gröbner Basis of the key Equation

Input: a received vector $\mathbf{y} = \left(\mathbf{y}^{(1)} \cdots \mathbf{y}^{(\ell)}\right) \in S^{n^{(1)}+\cdots+n^{(\ell)}}$ of the interleaved Gabidulin code $IGab_{\left(k^{(1)},\ldots,k^{(\ell)}\right)}\left(\mathbf{g}^{(1)},\ldots,\mathbf{g}^{(\ell)}\right)$.
Output: a Gröbner basis $\left\{\mathbf{V}_{(r,s)}\right\}_{0 \le r \le \ell,\ s \in [S]^*}$ for the module $M[\mathbf{y},\mathbf{g}]$.

$\mathcal{J} \leftarrow \{0,\ldots,\ell\} \times [S]^*$
for $(r,s) \in \mathcal{J}$ **do**
  $\mathbf{V}_{(r,s)} \leftarrow s\mathbf{e}^{(r)}$
for $l \leftarrow 1$ ***to*** ℓ **do**
  for $i \leftarrow 1$ ***to*** $n^{(l)}$ **do**
    for $(r,s) \in \mathcal{J}$ **do**
      $\Delta_{(r,s)} \leftarrow V^{(0)}_{(r,s)}\left(y^{(l)}_i\right) - V^{(l)}_{(r,s)}\left(g^{(l)}_i\right)$
    for $(r,s) \in \mathcal{J}$ **do**
      if $\Delta_{(r,s)} = 0$ **then**
        $\mathbf{V}'_{(r,s)} \leftarrow \mathbf{V}_{(r,s)}$
      else
        if *there exists a nonempty* $\mathcal{J}' \subset \mathcal{J}$ *such that for* $\left(r',s'\right) \in \mathcal{J}'$, $lm\left(\mathbf{V}_{(r',s')}\right) \prec lm\left(\mathbf{V}_{(r,s)}\right)$ *and* $\Delta_{(r,s)} + \sum_{(r',s') \in \mathcal{J}'} \theta_{(r',s')}\Delta_{(r',s')} = 0$ *for some* $\theta_{(r',s')} \in S$, **then**
          $\mathbf{V}'_{(r,s)} \leftarrow \mathbf{V}_{(r,s)} + \sum_{(r',s') \in \mathcal{J}'} \theta_{(r',s')}\mathbf{V}_{(r',s')}$
        else
          $\mathbf{V}'_{(r,s)} \leftarrow \left(X - b_{(r,s)}\right)\mathbf{V}_{(r,s)}$ where $b_{(r,s)}$ is an element of S such that $\sigma\left(\Delta_{(r,s)}\right) - b_{(r,s)}\Delta_{(r,s)} = 0$.
    for $(r,s) \in \mathcal{J}$ **do**
      $\mathbf{V}_{(r,s)} \leftarrow \mathbf{V}'_{(r,s)}$
return $\left\{\mathbf{V}_{(r,s)}\right\}_{0 \le r \le \ell,\ s \in [S]^*}$

Proposition 4.15: The vector $\mathbf{U} \in M[\mathbf{y},\mathbf{g}]$ is a solution of the key equation if and only if, w.r.t. $\preceq_{\left(k^{(0)},\ldots,k^{(\ell)}\right)}$, $ind(lm\,(\mathbf{U})) = 0$ and $lc\,(\mathbf{U}) = 1$.

Now, we can apply Proposition 2.11 to obtain all the solutions of the key equation.

Theorem 4.16: Let $\left\{\mathbf{V}_{(r,s)}\right\}_{0 \le r \le \ell,\ s \in [S]^*}$ be a Gröbner basis for $M[\mathbf{y},\mathbf{g}]$ obtained by Algorithm 2 w.r.t. $\preceq_{\left(k^{(0)},\ldots,k^{(\ell)}\right)}$. Set $\alpha_{(r,s)} := \deg\left(V^{(r)}_{(r,s)}\right)$.

(a) The vector $\mathbf{V}_{(0,1)}$ is a minimal solution of the key equation.

(b) All solution $\mathbf{U}$ of the key equation can be written as

$$\mathbf{U} = \sum_{0 \le r \le \ell,\ s \in [S]^*} w_{(r,s)}\mathbf{V}_{(r,s)}$$

where $w_{(r,s)} \in S[X,\sigma]$, $w_{(0,1)}$ is monic, for all $s \in [S]^* \setminus \{1\}$,

$$\deg\left(w_{(0,s)}\right) + \alpha_{(0,s)} < \deg\left(w_{(0,1)}\right) + \alpha_{(0,1)}$$

and for all $(r,s) \in \{1,\ldots,\ell\} \times [S]^*$,

$$\deg\left(w_{(r,s)}\right) + \alpha_{(r,s)} - k^{(r)} \le \deg\left(w_{(0,1)}\right) + \alpha_{(0,1)} - k^{(0)}.$$

Proof: (a) By construction of $\mathbf{V}_{(0,1)}$ and by Proposition 4.15, $\mathbf{V}_{(0,1)}$ is a minimal solution.

(b) Let $\mathbf{U}$ be a solution of the key equation. Then, $\mathbf{U} \in M[\mathbf{y},\mathbf{g}]$ and, by Proposition 4.15, $ind(lm(\mathbf{U})) = 0$, $lc(\mathbf{U}) = 1$, w.r.t. $\preceq_{(k^{(0)},\ldots,k^{(\ell)})}$. Let

$$\alpha = \deg(\mathbf{U}) - \deg\left(\mathbf{V}_{(0,1)}\right),$$

then $lm\left(\mathbf{U} - X^{\alpha}\mathbf{V}_{(0,1)}\right) \prec_{(k^{(0)},\ldots,k^{(\ell)})} lm(\mathbf{U})$. Therefore since $\mathbf{U} - X^{\alpha}\mathbf{V}_{(0,1)} \in M[\mathbf{y},\mathbf{g}]$, by Proposition 2.11,

$$\mathbf{U} - X^{\alpha}\mathbf{V}_{(0,1)} = \sum_{0\le r\le \ell,\ s\in[S]^*} h_{(r,s)}\mathbf{V}_{(r,s)},$$

where $h_{(r,s)} \in S[X,\sigma]$ and

$$lm\left(\mathbf{U} - X^{\alpha}\mathbf{V}_{(0,1)}\right) = \max_{0\le r\le \ell,\ s\in[S]^*}\left\{lm\left(h_{(r,s)}\right) lm\left(\mathbf{V}_{(r,s)}\right)\right\}.$$

Set $w_{(0,1)} = X^{\alpha} + h_{(0,1)}$ and $w_{(r,s)} = h_{(r,s)}$ if $(r,s) \neq (0,1)$. Then, the result follows. ■

V. Decoding Algorithms of Interleaved Gabidulin Codes

In this section, we use the solutions of the key equation to give the minimal list decoding, unique decoding, and error-erasure decoding algorithms of interleaved Gabidulin codes.

A. Minimal List Decoding

In [26], Kuijper and Trautmann used an iterative parametrization approach to give a minimal list decoding algorithm of Gabidulin codes over finite fields. In this subsection, we show that this algorithm can be generalized to interleaved Gabidulin codes over finite principal ideal rings.

***Definition** 5.1:* Let a received word $\mathbf{y} \in S^{n^{(1)}+\cdots+n^{(\ell)}}$ of the interleaved Gabidulin code $IGab_{(k^{(1)},\ldots,k^{(\ell)})}\left(\mathbf{g}^{(1)},\ldots,\mathbf{g}^{(\ell)}\right)$. Minimal list decoding consists to find the value of

$$t_{\min} := \min_{\mathbf{c}\in IGab_{(k^{(1)},\ldots,k^{(\ell)})}\left(\mathbf{g}^{(1)},\ldots,\mathbf{g}^{(\ell)}\right)} \{rank(\mathbf{y}-\mathbf{c})\} \qquad (6)$$

as well as all codewords $\mathbf{c} \in IGab_{(k^{(1)},\ldots,k^{(\ell)})}\left(\mathbf{g}^{(1)},\ldots,\mathbf{g}^{(\ell)}\right)$ such that $rank(\mathbf{y}-\mathbf{c}) = t_{\min}$.

Theorem 4.5 and Theorem 4.16 justify Algorithm 3 of minimal list decoding.

In general, the list size of minimal list decoding might be greater than one. In the next subsection, we give a sufficient condition so that the list size is one and a decoding algorithm in this case.

B. Unique Decoding Beyond the Error Correction Capability

Let $t_0 := \left\lfloor \left(\min_{l\in\{1,\ldots,\ell\}}\left\{d^{(l)}\right\} - 1\right)/2 \right\rfloor$ be the error correction capability of the interleaved Gabidulin code $IGab_{(k^{(1)},\ldots,k^{(\ell)})}\left(\mathbf{g}^{(1)},\ldots,\mathbf{g}^{(\ell)}\right)$ and let $\mathbf{y} = \left(\mathbf{y}^{(1)}\cdots\mathbf{y}^{(\ell)}\right)$ be a received word. We may have $t_{\min} \le t_0$ or $t_0 < t_{\min}$. Moreover, if $t_{\min} \le t_0$, then the list size of minimal list decoding is one. The next lemma give a necessary and sufficient condition so that $t_{\min} \le t_0$.

***Lemma** 5.2:* Let $\mathbf{U}$ be a minimal solution of the key equation and $\hat{\mathbf{f}} \in S[X,\sigma]_{<k^{(1)}} \times \cdots \times S[X,\sigma]_{<k^{(\ell)}}$. The following statements are equivalent.

Algorithm 3 Minimal List Decoding

Input: a received word $\mathbf{y} = \left(\mathbf{y}^{(1)}\cdots\mathbf{y}^{(\ell)}\right) \in S^{n^{(1)}+\cdots+n^{(\ell)}}$ of the interleaved Gabidulin code $IGab_{(k^{(1)},\ldots,k^{(\ell)})}\left(\mathbf{g}^{(1)},\ldots,\mathbf{g}^{(\ell)}\right)$.

Output: A list of $\hat{\mathbf{f}} \in S[X,\sigma]_{<k^{(1)}} \times \cdots \times S[X,\sigma]_{<k^{(\ell)}}$ such that $rank\left(\mathbf{y} - \left(f^{(1)}\left(\mathbf{g}^{(1)}\right)\cdots f^{(\ell)}\left(\mathbf{g}^{(\ell)}\right)\right)\right)$ is minimal.

Compute a Gröbner basis $\left\{\mathbf{V}_{(r,s)}\right\}_{0\le r\le\ell,\ s\in[S]^*}$ for the module $M[\mathbf{y},\mathbf{g}]$ as in Algorithm 2 w.r.t. $\preceq_{(k^{(0)},\ldots,k^{(\ell)})}$

$\alpha_{(r,s)} \leftarrow \deg\left(V^{(r)}_{(r,s)}\right)$

$list \leftarrow \emptyset$

$j \leftarrow 0$

while $list = \emptyset$ **do**

  Compute the set $\mathcal{U}$ of all $\mathbf{U} = \sum_{0\le r\le\ell,\ s\in[S]^*} w_{(r,s)}\mathbf{V}_{(r,s)}$ where $w_{(r,s)} \in S[X,\sigma]$, $w_{(0,1)}$ is monic, $\deg\left(w_{(0,1)}\right) = j$, $\deg\left(w_{(0,s)}\right) + \alpha_{(0,s)} < j + \alpha_{(0,1)}$, for all $s \in [S]^* \setminus \{1\}$, and $\deg\left(w_{(r,s)}\right) + \alpha_{(r,s)} - k^{(r)} \le j + \alpha_{(0,1)} - k^{(0)}$, for all $(r,s) \in \{1,\ldots,\ell\} \times [S]^*$

  foreach $\mathbf{U} \in \mathcal{U}$ **do**

    for $l \leftarrow 1$ ***to*** ℓ **do**

      Compute the quotient $Q^{(l)}$ and the remainder $P^{(l)}$ on the left Euclidean division of $U^{(l)}$ by $U^{(0)}$ in $S[X,\sigma]$

    if *for all* $l \in \{1,\ldots,\ell\}$, $P^{(l)} = 0$ **then**

      $list \leftarrow list \cup \{\hat{\mathbf{Q}}\}$

  $j \leftarrow j + 1$

return $list$

(i) $rank\left(\mathbf{y} - \left(f^{(1)}\left(\mathbf{g}^{(1)}\right)\cdots f^{(\ell)}\left(\mathbf{g}^{(\ell)}\right)\right)\right) \le t_0$.

(ii) It holds both that:

1) $\deg\left(U^{(0)}\right) \le t_0$;
2) $U^{(l)} = U^{(0)}f^{(l)}$, for $l = 1,\ldots,\ell$.

Proof: By Theorem 4.5, (ii)$\Longrightarrow$ (i).

The proof that (i)$\Longrightarrow$ (ii) is similar to that of [15, Proposition 8]. ■

Lemma 5.2 shows that if the rank of the error is at most the error correction capability, then every minimal solution of the key equation allows to recover the transmitted codeword. We use this property to give the unique decoding method beyond the error correction capability.

***Lemma** 5.3:* Assume there is $\hat{\mathbf{f}} \in S[X,\sigma]_{<k^{(1)}} \times \cdots \times S[X,\sigma]_{<k^{(\ell)}}$ such that for every minimal solution, $\mathbf{U}$, of the key equation we have $U^{(l)} = U^{(0)}f^{(l)}$, for $l = 1,\ldots,\ell$. Then, $\hat{\mathbf{f}}$ is the unique element in $S[X,\sigma]_{<k^{(1)}} \times \cdots \times S[X,\sigma]_{<k^{(\ell)}}$ such that

$$rank\left(\mathbf{y} - \left(f^{(1)}\left(\mathbf{g}^{(1)}\right)\cdots f^{(\ell)}\left(\mathbf{g}^{(\ell)}\right)\right)\right) = t_{\min}$$

where $t_{\min}$ is defined as in (6).

Proof: We show first that in this condition, $t_{\min}$ is equal to the degree of a minimal solution of the key equation. Let $\mathbf{U}$ be a minimal solution of the key equation and let t be the degree of $U^{(0)}$. Then, by the definition of $t_{\min}$ and by Theorem 4.5, we have $t \leq t_{\min}$. By the assumption, we have $U^{(l)} = U^{(0)} f^{(l)}$, for $l = 1, \dots, \ell$. Therefore, by Theorem 4.5, we also have $t_{\min} \leq t$. Thus, $t_{\min} = t$.

Now, let $\hat{\mathbf{b}} \in S[X,\sigma]_{<k^{(1)}} \times \cdots \times S[X,\sigma]_{<k^{(\ell)}}$ such that $rank\left(\mathbf{y} - \left(b^{(1)}\left(\mathbf{g}^{(1)}\right) \cdots b^{(\ell)}\left(\mathbf{g}^{(\ell)}\right)\right)\right) = t_{\min}$. Then, by Proposition 3.15, there exists a monic skew polynomial $W \in S[X,\sigma]$ of degree $t_{\min}$ such that, for $l = 1, \dots, \ell$, $W\left(\mathbf{y}^{(l)} - b^{(l)}\left(\mathbf{g}^{(l)}\right)\right) = \mathbf{0}$. Therefore, $\left(W, Wb^{(1)}, \dots, Wb^{(\ell)}\right)$ is a minimal solution of the key equation. Thus $b^{(l)} = f^{(l)}$, for $l = 1, \dots, \ell$. ∎

Lemma 5.3 gives a sufficient condition so that the list size of minimal list decoding is one. The following lemma gives a Gröbner basis interpretation of this condition.

Lemma 5.4: Let $\left\{\mathbf{V}_{(r,s)}\right\}_{0 \leq r \leq \ell,\ s \in [S]^*}$ be a Gröbner basis for $M[\mathbf{y}, \mathbf{g}]$ obtained by Algorithm 2 w.r.t. $\preceq_{\left(k^{(0)},\dots,k^{(\ell)}\right)}$. Set $\alpha_{(r,s)} := \deg\left(V^{(r)}_{(r,s)}\right)$. Let $Q^{(l)}_{(0,1)}$ be the quotient and $P^{(l)}_{(0,1)}$ be the remainder of the left Euclidean division of $V^{(l)}_{(0,1)}$ by $V^{(0)}_{(0,1)}$ in $S[X,\sigma]$. The following statements are equivalent.

(i) There is $\hat{\mathbf{f}} \in S[X,\sigma]_{<k^{(1)}} \times \cdots \times S[X,\sigma]_{<k^{(\ell)}}$ such that for every minimal solution, $\mathbf{U}$, of the key equation we have $U^{(l)} = U^{(0)} f^{(l)}$, for $l = 1, \dots, \ell$.

(ii) The Gröbner basis $\left\{\mathbf{V}_{(r,s)}\right\}_{0 \leq r \leq \ell,\ s \in [S]^*}$ has the following properties:

1) $P^{(l)}_{(0,1)} = 0$, for $l = 1, \dots, \ell$;
2) $\alpha_{(0,1)} - k^{(0)} < \alpha_{(r,s)} - k^{(r)}$, for all $r \in \{1, \dots, \ell\}$ and $s \in [S]^*$;
3) $V^{(l)}_{(0,s)} = V^{(0)}_{(0,s)} Q^{(l)}_{(0,1)}$, for all $l \in \{1, \dots, \ell\}$ and $s \in [S]^* \setminus \{1\}$.

Proof: (i)$\Longrightarrow$ (ii):

1) Since $\mathbf{V}_{(0,1)}$ is a minimal solution of the key equation, we have $V^{(l)}_{(0,1)} = V^{(0)}_{(0,1)} f^{(l)}$, for $l = 1, \dots, \ell$. Consequently, $Q^{(l)}_{(0,1)} = f^{(l)}$ and $P^{(l)}_{(0,1)} = 0$, for $l = 1, \dots, \ell$.

2) Suppose there are $r \in \{1, \dots, \ell\}$ and $s \in [S]^*$ such that $\alpha_{(r,s)} - k^{(r)} \leq \alpha_{(0,1)} - k^{(0)}$. Then, $\mathbf{V}_{(0,1)} + \mathbf{V}_{(r,s)}$ is a minimal solution of the key equation. Consequently, we have $V^{(r)}_{(0,1)} + V^{(r)}_{(r,s)} = \left(V^{(0)}_{(0,1)} + V^{(0)}_{(r,s)}\right) f^{(r)}$. Since $V^{(r)}_{(0,1)} = V^{(0)}_{(0,1)} f^{(r)}$, we then have $V^{(r)}_{(r,s)} = V^{(0)}_{(r,s)} f^{(r)}$. Hence, $\deg\left(V^{(r)}_{(r,s)}\right) = \deg\left(V^{(0)}_{(r,s)} f^{(r)}\right)$, i.e., $\deg\left(V^{(r)}_{(r,s)}\right) \leq \deg\left(V^{(0)}_{(r,s)}\right) + k^{(r)} - 1$ which is absurd because w.r.t. $\preceq_{\left(k^{(0)},\dots,k^{(\ell)}\right)}$, $ind\left(lm\left(\mathbf{V}_{(r,s)}\right)\right) = r$.

3) Let $s \in [S]^* \setminus \{1\}$. Since $\deg\left(\mathbf{V}_{(0,s)}\right)$ is minimal among the degree of all $\mathbf{U} \in M$ with $ind(lm(\mathbf{U})) = 0$, $lc(\mathbf{U}) \in [s]$, then we have $\alpha_{(0,s)} \leq \alpha_{(0,1)}$. If $\alpha_{(0,s)} < \alpha_{(0,1)}$, then $\mathbf{V}_{(0,1)} + \mathbf{V}_{(0,s)}$ is a minimal solution of the key equation and consequently we have $V^{(l)}_{(0,s)} = V^{(0)}_{(0,s)} f^{(l)}$. If $\alpha_{(0,s)} = \alpha_{(0,1)}$, then $\mathbf{V}_{(0,1)} + \mathbf{V}_{(0,s)} - lc\left(V^{(0)}_{(0,s)}\right)\mathbf{V}_{(0,1)}$ is a minimal solution of the key equation and therefore we have $V^{(l)}_{(0,s)} = V^{(0)}_{(0,s)} f^{(l)}$.

(ii)$\Longrightarrow$ (i): Let $\mathbf{U}$ be a minimal solution of the key equation. Then, by Theorem 4.16,

$$\mathbf{U} = \sum_{0 \leq r \leq \ell,\ s \in [S]^*} w_{(r,s)} \mathbf{V}_{(r,s)}$$

where $w_{(r,s)} \in S[X,\sigma]$, $w_{(0,1)} = 1$, for all $s \in [S]^* \setminus \{1\}$,

$$\deg\left(w_{(0,s)}\right) + \alpha_{(0,s)} < \alpha_{(0,1)}$$

and for all $(r,s) \in \{1, \dots, \ell\} \times [S]^*$,

$$\deg\left(w_{(r,s)}\right) + \alpha_{(r,s)} - k^{(r)} \leq \alpha_{(0,1)} - k^{(0)}.$$

Let $(r,s) \in \{1, \dots, \ell\} \times [S]^*$, then $w_{(r,s)} = 0$ because $\alpha_{(0,1)} - k^{(0)} < \alpha_{(r,s)} - k^{(r)}$. Therefore $U^{(l)} = U^{(0)} Q^{(l)}_{(0,1)}$, for $l = 1, \dots, \ell$, because $V^{(l)}_{(0,s)} = V^{(0)}_{(0,s)} Q^{(l)}_{(0,1)}$, for $l = 1, \dots, \ell$ and $s \in [S]^*$. ∎

The previous lemmas allow to give Algorithm 4.

Algorithm 4 Unique Decoding

Input: a received word $\mathbf{y} = \left(\mathbf{y}^{(1)} \cdots \mathbf{y}^{(\ell)}\right) \in S^{n^{(1)}+\cdots+n^{(\ell)}}$ of the interleaved Gabidulin code $IGab_{\left(k^{(1)},\dots,k^{(\ell)}\right)}\left(\mathbf{g}^{(1)}, \dots, \mathbf{g}^{(\ell)}\right)$.

Output: "decoding failure" or the element $\hat{\mathbf{f}}$ in $S[X,\sigma]_{<k^{(1)}} \times \cdots \times S[X,\sigma]_{<k^{(\ell)}}$ such that for every minimal solution, $\mathbf{U}$, of the key equation we have $U^{(l)} = U^{(0)} f^{(l)}$, for $l = 1, \dots, \ell$.

```
t_0 ← ⌊(min_{l∈{1,...,ℓ}} {d^(l)} − 1)/2⌋
Compute a Gröbner basis {V_(r,s)}_{0≤r≤ℓ, s∈[S]*} for the
  module M[y, g] as in Algorithm 2 w.r.t. ⪯_(k^(0),...,k^(ℓ))
α_(r,s) ← deg(V^(r)_(r,s))
if there is r ∈ {1,...,ℓ} and s ∈ [S]* such that
  α_(r,s) − k^(r) ≤ α_(0,1) − k^(0) then
  return "decoding failure"
for l ← 1 to ℓ do
  Compute the quotient Q^(l)_(0,1) and the remainder P^(l)_(0,1)
  on the left Euclidean division of V^(l)_(0,1) by V^(0)_(0,1) in
  S[X, σ].
if there is l ∈ {1,...,ℓ} such that P^(l)_(0,1) ≠ 0 then
  return "decoding failure"
else
  if α_(0,1) ≤ t_0 then
    return Q̂_(0,1)
  else
    if there is l ∈ {1,...,ℓ} and s ∈ [S]* \ {1} such
    that V^(l)_(0,s) ≠ V^(0)_(0,s) Q^(l)_(0,1) then
      return "decoding failure"
    else
      return Q̂_(0,1)
```

We have the following theorem.

Theorem 5.5: (a) If there is $\hat{\mathbf{f}} \in S[X,\sigma]_{<k^{(1)}} \times \cdots \times S[X,\sigma]_{<k^{(\ell)}}$ such that $rank\left(\mathbf{y} - \left(f^{(1)}\left(\mathbf{g}^{(1)}\right) \cdots f^{(\ell)}\left(\mathbf{g}^{(\ell)}\right)\right)\right) \leq t_0$, then Algorithm 4 returns $\hat{\mathbf{f}}$.

(b) If Algorithm 4 returns $\hat{\mathbf{f}}$, then it is the unique element in $S[X,\sigma]_{<k^{(1)}} \times \cdots \times S[X,\sigma]_{<k^{(\ell)}}$ such that $rank\left(\mathbf{y} - \left(f^{(1)}\left(\mathbf{g}^{(1)}\right) \cdots f^{(\ell)}\left(\mathbf{g}^{(\ell)}\right)\right)\right) = t_{\min}$.

Proof: (a) Since $\mathbf{V}_{(0,1)}$ is a minimal solution of the key equation, then, by Lemma 5.2, there is $\hat{\mathbf{f}} \in S[X,\sigma]_{<k^{(1)}} \times \cdots \times S[X,\sigma]_{<k^{(\ell)}}$ such that $rank\left(\mathbf{y} - \left(f^{(1)}\left(\mathbf{g}^{(1)}\right) \cdots f^{(\ell)}\left(\mathbf{g}^{(\ell)}\right)\right)\right) \le t_0$ if and only if $\alpha_{(0,1)} \le t_0$ and $P^{(l)}_{(0,1)} = 0$, for $l = 1, \ldots, \ell$.

(b) This result is a direct consequence of Lemma 5.3 and Lemma 5.4. ■

Recall that we may have $t_{\min} \le t_0$ or $t_0 < t_{\min}$. Thus, Algorithm 4 can uniquely decode beyond the error correction capability. The following example is given as an illustration.

***Example** 5.6:* Let

$$R = \mathbb{Z}_4, \quad S = R[z]/\left(z^4 + 2z^2 + 3z + 1\right)$$

and $a = z + \left(z^4 + 2z^2 + 3z + 1\right)$. Then, S is a Galois extension of R where the Galois group is generated by a power map $\sigma : a \mapsto a^2$. Set $\mathbf{g}^{(1)} = \mathbf{g}^{(2)} = (1, a, a^2, a^3)$,

$$\begin{aligned}\mathbf{y}^{(1)} &= (3a^3 + 2a^2 + 2, a^2 + 2a, \\ &\quad a^3 + 2, 2a^3 + 2a^2 + 3a + 3) \\ \mathbf{y}^{(2)} &= (a^2 + 2a + 3, 2a^3 + a^2 + 2a + 3, \\ &\quad a^3 + a^2 + 2a + 3, 2a^3 + 3).\end{aligned}$$

We consider the received word $\mathbf{y} = \left(\mathbf{y}^{(1)} \;\; \mathbf{y}^{(2)}\right)$ of the interleaved Gabidulin code $IGab_{(1,1)}\left(\mathbf{g}^{(1)}, \mathbf{g}^{(2)}\right)$. Using SageMathCloud [28], Algorithm 4 returns $\left(f^{(1)}, f^{(2)}\right)$ where $f^{(1)} = 2a^3 + 3a$ and $f^{(2)} = 3a^2 + 2a + 1$. Therefore, the error vector is $\boldsymbol{\varepsilon} = \mathbf{y} - \left(f^{(1)}\left(\mathbf{g}^{(1)}\right) \;\; f^{(2)}\left(\mathbf{g}^{(2)}\right)\right)$ and $rank(\boldsymbol{\varepsilon}) = 2 > t_0 = 1$.

***Remark** 5.7:* In finite fields, Sidorenko et al. [56] gave an algorithm for decoding interleaved Gabidulin codes beyond the error correction capability and an upper bound of the failure probability. We implemented Algorithm 4 and compared it to [56, Algorithm 4]. We observed that these two algorithms fail in the same cases. Thus, it would be interesting to study if there exists the connection between the two algorithms.

C. Error-Erasure Decoding

As in [6], we define row and column erasures of interleaved Gabidulin codes. We then show that errors and erasures decoding of an interleaved Gabidulin code is reduced to errors decoding of another interleaved Gabidulin code.

Let $\mathbf{y} = \left(\mathbf{y}^{(1)} \ldots \mathbf{y}^{(\ell)}\right) \in S^{n^{(1)} + \cdots + n^{(\ell)}}$ be a received vector for a transmitted codeword $\left(f^{(1)}\left(\mathbf{g}^{(1)}\right) \cdots f^{(\ell)}\left(\mathbf{g}^{(\ell)}\right)\right)$ of the interleaved Gabidulin code $IGab_{\left(k^{(1)},\ldots,k^{(\ell)}\right)}\left(\mathbf{g}^{(1)}, \ldots, \mathbf{g}^{(\ell)}\right)$.

Assume that the error vector

$$\boldsymbol{\varepsilon} = \left(\mathbf{y}^{(1)} \ldots \mathbf{y}^{(\ell)}\right) - \left(f^{(1)}\left(\mathbf{g}^{(1)}\right) \cdots f^{(\ell)}\left(\mathbf{g}^{(\ell)}\right)\right) \tag{7}$$

is decomposed into

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^{(E)} + \boldsymbol{\varepsilon}^{(R)} + \boldsymbol{\varepsilon}^{(C)} \tag{8}$$

where

· $\boldsymbol{\varepsilon}^{(E)}$, called the full error, is unknown, $rank\left(\boldsymbol{\varepsilon}^{(E)}\right) = t^{(E)}$;

· $\boldsymbol{\varepsilon}^{(R)}$, called the row erasure, can be expressed in the form

$$\boldsymbol{\varepsilon}^{(R)} = \left(\mathbf{a}^{(R,1)}\mathbf{B}^{(R,1)} \ldots \mathbf{a}^{(R,\ell)}\mathbf{B}^{(R,\ell)}\right)$$

with $\mathbf{a}^{(R,l)} \in S^{t^{(R,l)}}$ is known, $rank\left(\mathbf{a}^{(R,l)}\right) = t^{(R,l)}$, and $\mathbf{B}^{(R,l)} \in R^{t^{(R,l)} \times n^{(l)}}$ is unknown, for $l = 1, \ldots, \ell$;

· $\boldsymbol{\varepsilon}^{(C)}$, called the column erasure, can be expressed in the form

$$\boldsymbol{\varepsilon}^{(C)} = \left(\mathbf{a}^{(C,1)}\mathbf{B}^{(C,1)} \ldots \mathbf{a}^{(C,\ell)}\mathbf{B}^{(C,\ell)}\right)$$

with $\mathbf{a}^{(C,l)} \in S^{t^{(C,l)}}$ is unknown, $\mathbf{B}^{(C,l)} \in R^{t^{(C,l)} \times n^{(l)}}$ is known, $freerank\left(\mathbf{B}^{(C,l)}\right) = t^{(C,l)}$, for $l = 1, \ldots, \ell$.

By Proposition 3.15, there are the monic skew polynomials $P^{(R,l)} \in S[X,\sigma]$ of degree $t^{(R,l)}$ such that $P^{(R,l)}\left(\mathbf{a}^{(R,l)}\right) = \mathbf{0}$, for $l = 1, \ldots, \ell$.

By [42, Proposition 2.9], there are the free column matrices $\mathbf{F}^{(C,l)} \in R^{n^{(l)} \times \left(n^{(l)} - t^{(C,l)}\right)}$ such that $\mathbf{B}^{(R,l)}\mathbf{F}^{(C,l)} = \mathbf{0}$, for $l = 1, \ldots, \ell$.

***Theorem** 5.8:* With the above notations, the relation (7) can be transformed into

$$\boldsymbol{\varepsilon}' = \left(\mathbf{y}'^{(1)} \ldots \mathbf{y}'^{(\ell)}\right) - \left(f'^{(1)}\left(\mathbf{g}'^{(1)}\right) \cdots f'^{(\ell)}\left(\mathbf{g}'^{(\ell)}\right)\right)$$

where $\mathbf{y}'^{(l)} = P^{(R,l)}\left(\mathbf{y}^{(l)}\right)\mathbf{F}^{(C,l)}$, $\mathbf{g}'^{(l)} = \mathbf{g}^{(l)}\mathbf{F}^{(C,l)}$, $f'^{(l)} = P^{(R,l)}f^{(l)}$, for $l = 1, \ldots, \ell$, and $rank\left(\boldsymbol{\varepsilon}'\right) \le t^{(E)}$.

Proof: Set $\boldsymbol{\varepsilon}^{(E)} = \left(\boldsymbol{\varepsilon}^{(E,1)} \ldots \boldsymbol{\varepsilon}^{(E,\ell)}\right)$ where $\boldsymbol{\varepsilon}^{(E,l)} \in S^{n^{(l)}}$, for $l = 1, \ldots, \ell$. Then, by (7) and (8), we have

$$\boldsymbol{\varepsilon}^{(E,l)} + \boldsymbol{\varepsilon}^{(R,l)} + \boldsymbol{\varepsilon}^{(C,l)} = \mathbf{y}^{(l)} - f^{(l)}\left(\mathbf{g}^{(l)}\right), \text{ for } l = 1, \ldots, \ell.$$

Let $l \in \{1, \ldots, \ell\}$. Since $\boldsymbol{\varepsilon}^{(R,l)} = \mathbf{a}^{(R,l)}\mathbf{B}^{(R,l)}$ and $P^{(R,l)}\left(\mathbf{a}^{(R,l)}\right) = \mathbf{0}$, we have

$$P^{(R,l)}\left(\boldsymbol{\varepsilon}^{(E,l)}\right) + P^{(R,l)}\left(\boldsymbol{\varepsilon}^{(C,l)}\right) = P^{(R,l)}\left(\mathbf{y}^{(l)} - f^{(l)}\left(\mathbf{g}^{(l)}\right)\right)$$

i.e.,

$$\begin{aligned}&P^{(R,l)}\left(\boldsymbol{\varepsilon}^{(E,l)}\right) \\ &+P^{(R,l)}\left(\mathbf{a}^{(C,l)}\right)\mathbf{B}^{(C,l)} \quad = P^{(R,l)}\left(\mathbf{y}^{(l)} - f^{(l)}\left(\mathbf{g}^{(l)}\right)\right)\end{aligned} \tag{9}$$

because $\boldsymbol{\varepsilon}^{(C,l)} = \mathbf{a}^{(C,l)}\mathbf{B}^{(C,l)}$. If we right multiply both sides of (9) by $\mathbf{F}^{(C,l)}$ we get

$$\boldsymbol{\varepsilon}'^{(E,l)} = \mathbf{y}'^{(l)} - f'^{(l)}\left(\mathbf{g}'^{(l)}\right)$$

where $\boldsymbol{\varepsilon}'^{(E,l)} = P^{(R,l)}\left(\boldsymbol{\varepsilon}^{(E,l)}\right)\mathbf{F}^{(C,l)}$.

Set $\boldsymbol{\varepsilon}' = \left(\boldsymbol{\varepsilon}'^{(E,1)} \ldots \boldsymbol{\varepsilon}'^{(E,\ell)}\right)$, then

$$\boldsymbol{\varepsilon}' = \left(\mathbf{y}'^{(1)} \ldots \mathbf{y}'^{(\ell)}\right) - \left(f'^{(1)}\left(\mathbf{g}'^{(1)}\right) \cdots f'^{(\ell)}\left(\mathbf{g}^{(\ell)}\right)\right).$$

As $rank\left(\left(\boldsymbol{\varepsilon}^{(E,1)} \ldots \boldsymbol{\varepsilon}^{(E,\ell)}\right)\right) = t^E$,
we have $rank\left(\boldsymbol{\varepsilon}'^{(E,1)} \ldots \boldsymbol{\varepsilon}'^{(E,\ell)}\right) \le t^E$. ■

Set $k'^{(l)} = k^{(l)} + t^{(R,l)}$, $n'^{(l)} = n^{(l)} - t^{(C,l)}$ and assume that $k'^{(l)} \le n'^{(l)}$, for $l = 1, \ldots, \ell$. Then, according to Theorem 5.8, the error and erasure decoding of the interleaved Gabidulin code $IGab_{\left(k^{(1)},\ldots,k^{(\ell)}\right)}\left(\mathbf{g}^{(1)}, \ldots, \mathbf{g}^{(\ell)}\right)$ is reduced to the error decoding of the interleaved Gabidulin code $IGab_{\left(k'^{(1)},\ldots,k'^{(\ell)}\right)}\left(\mathbf{g}'^{(1)}, \ldots, \mathbf{g}'^{(\ell)}\right)$. In particular we have the following:

***Corollary** 5.9:* With the above notations, If

$$2t^{(E)} \le \min_{1 \le l \le \ell}\left\{n^{(l)} - \left(k^{(l)} + t^{(R,l)} + t^{(C,l)}\right)\right\}$$

then the transmitted massage i.e., $f^{(1)}, \ldots, f^{(\ell)}$, can recover.

Proof: Assume that

$$2t^{(E)} \le \min_{1 \le l \le \ell}\left\{n^{(l)} - \left(k^{(l)} + t^{(R,l)} + t^{(C,l)}\right)\right\}.$$

Then,

$$2t^{(E)} \leq d' - 1,$$

where d' is the rank distance of the interleaved Gabidulin code $IGab_{(k'^{(1)},\ldots,k'^{(\ell)})}\left(\mathbf{g}'^{(1)},\ldots,\mathbf{g}'^{(\ell)}\right)$. Hence, we can use Algorithm 4 to determine $f'^{(1)},\ldots,f'^{(\ell)}$ and then use the left Euclidean division of $f'^{(l)}$ by $P^{(R,l)}$ to determine $f^{(l)}$ for $l = 1,\ldots,\ell$. ∎

As in [9], [57], [58], simultaneous correction of errors and erasures allows to recover the transmitted codeword in random linear network coding. As an illustration, see subsection VI-B.

VI. Applications

A. Space-Time Block Codes From Codes Over Finite Principal Ideal Rings

A space-time block code is a finite set of complex matrices of the same size. Recall that the rank criterion [10] for space-time block codes states that, in order to achieve the maximum diversity, the rank of the difference of two distinct codewords has to be maximal. In this subsection, we generalize to finite principal ideal rings the methods of [7], [12], [14], [19] in the construction of space-time block codes. More precisely, we show that there is a rank-preserving map from a finite principal ideal ring to a complex signal set and we use it to construct space-time block codes that are optimal under the rate-diversity tradeoff [10]–[12].

Let T be a principal ideal ring such that there exists a surjective ring homomorphism $\varphi : T \to R$. Let φ^* be a section of φ, i.e., a map from R to T such that $\varphi \circ \varphi^* = id_R$. The extension of φ (resp., φ^*) coefficient-by-coefficient to the set of matrix $T^{m\times n}$ (resp., $R^{m\times n}$) is also denoted by φ (resp., φ^*). As an example, we may have $T = \mathbb{Z}[i]$, $R = \mathbb{Z}[i]/\eta\mathbb{Z}[i]$, where η is some positive integer, $\varphi(x) = x + \eta\mathbb{Z}[i]$ and $\varphi^*(a + bi + \eta\mathbb{Z}[i]) = (a \bmod \eta) + (b \bmod \eta)\, i$, for all $x \in \mathbb{Z}[i]$, $a \in \mathbb{Z}$, $b \in \mathbb{Z}$.

***Lemma** 6.1:* Let $\mathbf{A} \in T^{m\times n}$. Then,

$$rank_R(\varphi(\mathbf{A})) \leq rank_T(\mathbf{A}).$$

Proof: Let $r = rank_T(\mathbf{A})$ and $\{\mathbf{b}_1,\ldots,\mathbf{b}_r\}$ be a generating set of $col(\mathbf{A})$. Then, $\{\varphi(\mathbf{b}_1),\ldots,\varphi(\mathbf{b}_r)\}$ is a generating set of $col(\varphi(\mathbf{A}))$. Consequently, $rank_R(\varphi(\mathbf{A})) \leq rank_T(\mathbf{A})$. ∎

***Theorem** 6.2:* Let $\mathcal{M} \subset R^{m\times n}$ be a rank code of rank distance d and let d' be the rank distance of $\varphi^*(\mathcal{M})$, then $d \leq d'$. Moreover, if $\mathcal{M}$ is an MRD code, then $d = d'$.

Proof: Let $\varphi^*(\mathbf{M}_1)$, $\varphi^*(\mathbf{M}_2) \in \varphi^*(\mathcal{M})$ such that $\varphi^*(\mathbf{M}_1) \neq \varphi^*(\mathbf{M}_2)$. Then, $\mathbf{M}_1 \neq \mathbf{M}_2$ and by Lemma 6.1, $rank_T(\varphi^*(\mathbf{M}_1) - \varphi^*(\mathbf{M}_2))$ is greater than or equal to $rank_R(\varphi(\varphi^*(\mathbf{M}_1) - \varphi^*(\mathbf{M}_2)))$. But,

$$rank_R\left(\varphi\left(\varphi^*(\mathbf{M}_1) - \varphi^*(\mathbf{M}_2)\right)\right) \geq d.$$

Thus, $d \leq d'$.

Assume that $\mathcal{M}$ is an MRD code. Then,

$$\left|\varphi^*(\mathcal{M})\right| = |\mathcal{M}| = |R|^{\min\{m(n-d+1),\ n(m-d+1)\}} \tag{10}$$

Using the same arguments as in the proof of Proposition 3.20, we can show that

$$\left|\varphi^*(\mathcal{M})\right| \leq \left|\varphi^*(R)\right|^{\min\left\{m\left(n-d'+1\right),\ n\left(m-d'+1\right)\right\}} \tag{11}$$

It follows from (10) and (11) that $d' \leq d$. ∎

By the previous theorem, we can use an MRD code in R to construct an MRD code in T. The following example is a generalization of [7], [13].

***Example** 6.3:* Since $S \cong R[X]/(h)$ where h is a monic polynomial, set $h = a_0 + a_1X + \cdots + a_{m-1}X^{m-1} + X^m$, $\alpha = X + (h)$ and $\mathbf{g} = \left(\alpha,\alpha^2,\ldots,\alpha^m\right)$. Then, the Gabidulin code $Gab_1(\mathbf{g})$ is a free S-linear rank code generated by $\mathbf{g}$. Thus, $Gab_1(\mathbf{g})$ is a free R-linear rank code generated by $\left\{\mathbf{g},\alpha\mathbf{g},\ldots,\alpha^{m-1}\mathbf{g}\right\}$. The matrix representation of $\mathbf{g}$ in the basis $\left(1,\alpha,\ldots,\alpha^{m-1}\right)$ is

$$\mathbf{A_g} = \begin{pmatrix} 0 & 0 & \cdots & 0 & -a_0 \\ 1 & 0 & \cdots & 0 & -a_1 \\ 0 & 1 & \cdots & 0 & -a_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & -a_{m-1} \end{pmatrix}$$

and the matrix representation of $\alpha^i\mathbf{g}$ is $\mathbf{A}_\mathbf{g}^{i+1}$ for $i = 1,\ldots,m-1$. Therefore, the matrix representation of $Gab_1(\mathbf{g})$ is a R-linear rank code generated by $\left\{\mathbf{A}_\mathbf{g}^i\right\}_{1\leq i\leq m}$. Its image in T is an MRD code of rank distance m. Moreover, all codeword have the full rank. By Proposition 4.2, the interleaved Gabidulin code $IGab_{(k^{(1)},\ldots,k^{(\ell)})}\left(\mathbf{g}^{(1)},\ldots,\mathbf{g}^{(\ell)}\right)$ with $k^{(l)} = 1$ and $\mathbf{g}^{(l)} = \left(\alpha,\alpha^2,\ldots,\alpha^m\right)$, for $l = 1,\ldots,\ell$, have the same proprieties. Thus, we can use it to construct optimal space-time block code in T.

B. Decoding of Random Linear Network Codes Over Finite Principal Ideal Rings

In this subsection, we consider random linear network coding over finite principal ideal rings. To improve the error correction, we combine the encoding schemes of [9] and [20], that is, we consider that the transmitted matrix is represented by the matrix $\mathbf{X} = \begin{pmatrix}\mathbf{0}_{m\times\beta_0} & \mathbf{I}_m & \mathbf{M}\end{pmatrix}$ where $\mathbf{M}$ is a code matrix of some matrix code $\mathcal{M} \subset R^{m\times n}$. The channel equation is given by

$$\mathbf{Y} = \mathbf{AX} + \mathbf{E} \tag{12}$$

where the transfer matrix $\mathbf{A} \in R^{m_r\times m}$ and $rank(\mathbf{E}) := \beta$. Recall that the random matrices $\mathbf{A}$ and $\mathbf{E}$ are unknown to the destination and the problem is to recover the transmitted matrix $\mathbf{X}$ from the received matrix $\mathbf{Y}$. As in [9] and [57], we will show that this problem can be reformulated as an error-erasure decoding problem for rank-metric codes.

When the matrix $\mathbf{Y}$ is received, the Smith normal form is used to successively transform the decoding problem into error-erasure decoding. In the following, we give these transformations.

1) First Transformation: Set

$$\mathbf{Y} = \begin{pmatrix}\mathbf{Y}_0 & \mathbf{Y}_1 & \mathbf{Y}_2\end{pmatrix},$$

where $\mathbf{Y}_0$, $\mathbf{Y}_1$ and $\mathbf{Y}_2$ are submatrices of $\mathbf{Y}$ of sizes $m_r \times \beta_0$, $m_r \times m$ and $m_r \times n$, respectively. Set $freerank(\mathbf{Y}_0) := \alpha_{0f}$.

Then, using the Smith normal form, there exist the invertible matrices $\mathbf{P}$, $\mathbf{Q}$ and the diagonal matrix $\mathbf{D}_2$ such that

$$\mathbf{P}\mathbf{Y}_0\mathbf{Q} = \begin{pmatrix} \mathbf{I}_{\alpha_{0f}} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_2 \end{pmatrix}.$$

Set

$$\widetilde{\mathbf{Q}} = \begin{pmatrix} \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{m+n} \end{pmatrix}$$

and

$$\mathbf{P} = \begin{pmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \end{pmatrix}$$

where $\mathbf{P}_1$ and $\mathbf{P}_2$ are the submatrices of $\mathbf{P}$ of sizes $\alpha_{0f} \times m_r$, and $(m_r - \alpha_{0f}) \times m_r$, respectively. If we multiply both sides of (12) by $\mathbf{P}$ and $\widetilde{\mathbf{Q}}$ we get the following:

***Lemma** 6.4:* With the above notations,

$$\mathbf{Y}' = \mathbf{A}' \begin{pmatrix} \mathbf{I}_m & \mathbf{M} \end{pmatrix} + \mathbf{E}' \tag{13}$$

where $\mathbf{Y}' = \mathbf{P}_2 \begin{pmatrix} \mathbf{Y}_1 & \mathbf{Y}_2 \end{pmatrix}$, $\mathbf{A}' = \mathbf{P}_2\mathbf{A}$ and $\mathbf{E}'$ is a matrix with $rank\left(\mathbf{E}'\right) := \beta' \leq \beta - \alpha_{0f}$.

2) Second Transformation: Set $m_r' := m_r - \alpha_{0f}$ and

$$\mathbf{Y}' := \begin{pmatrix} \mathbf{Y}_1' & \mathbf{Y}_2' \end{pmatrix}.$$

where $\mathbf{Y}_1'$ and $\mathbf{Y}_2'$ are submatrices of $\mathbf{Y}'$ of sizes $m_r' \times m$ and $m_r' \times n$, respectively.

Set $rank\left(\mathbf{Y}_1'\right) := \alpha_1$, $freerank\left(\mathbf{Y}_1'\right) := \alpha_{1f}$. Using the Smith normal form, there exist the invertible matrices $\mathbf{P}'$, $\mathbf{Q}'$ and the diagonal matrix $\mathbf{D}' = diag\,(d_1, \ldots, d_r)$, with $d_1 = \cdots = d_{\alpha_{1f}} = 1$, such that

$$\mathbf{P}'\mathbf{Y}_1'\mathbf{Q}' = \mathbf{D}'.$$

Using Proposition 3.8, if we decompose $\mathbf{E}'$ as in [57, Eq. (29)] then we get the following:

***Lemma** 6.5:* With the above notations,

$$\mathbf{Y}_2'' = \mathbf{D}'\mathbf{M}' + \mathbf{E}''. \tag{14}$$

where $\mathbf{Y}_2'' = \mathbf{P}'\mathbf{Y}_2'$, $\mathbf{M}' = \mathbf{Q}'^{-1}\mathbf{M}$ and $\mathbf{E}''$ is a matrix with $rank\left(\mathbf{E}''\right) \leq \beta'$.

3) Third Transformation: Set

$$\mathbf{D}' = \begin{pmatrix} \mathbf{D}_1' \\ \mathbf{0} \end{pmatrix}$$

and

$$\mathbf{Y}_2'' = \begin{pmatrix} \mathbf{Y}_{21}'' \\ \mathbf{Y}_{22}'' \end{pmatrix}$$

where $\mathbf{D}_1'$ is the submatrix of $\mathbf{D}'$ of sizes $\alpha_1 \times m$, $\mathbf{Y}_{21}''$ and $\mathbf{Y}_{22}''$ are submatrices of $\mathbf{Y}_2''$ of sizes $\alpha_1 \times n$ and $\left(m_r' - \alpha_1\right) \times n$, respectively.

Let $\alpha_{22f} := freerank\left(\mathbf{Y}_{22}''\right)$. If $\alpha_{22f} \neq 0$ then, using the Smith normal form, there is a $\alpha_{22f} \times \left(m_r' - \alpha_1\right)$ matrix $\mathbf{U}$, such that the free rank of the matrix $\mathbf{Y}_{22}''' := \mathbf{U}\mathbf{Y}_{22}''$ is α_{22f}.

Let $\widehat{\mathbf{Y}}_{22}$ be the matrix defined by $\widehat{\mathbf{Y}}_{22} := \mathbf{Y}_{22}'''$ if $\alpha_{22f} \neq 0$ and $\widehat{\mathbf{Y}}_{22}$ is a $1 \times n$ zero matrix else.

Let $\mathbf{D}_1''$ be the $m \times m$ matrix and $\mathbf{Y}_{21}'''$ be the $m \times n$ matrix obtained respectively from the matrices $\mathbf{D}_1'$ and $\mathbf{Y}_{21}''$ by inserting all-zero rows below the last row if $\alpha_1 \leq m$ and by deleting the $\alpha_1 - m$ last rows else.

Set $\widehat{\mathbf{D}}_1 := \mathbf{Q}'\left(\mathbf{D}_1'' - \mathbf{I}_m\right)$ and $\widehat{\mathbf{Y}}_{21} := \mathbf{Q}'\mathbf{Y}_{21}'''$. Note that, $\widehat{\mathbf{D}}_1 = \mathbf{0}$ if $\alpha_{1f} \geq m$ and $rank\left(\widehat{\mathbf{D}}_1\right) \leq m - \alpha_{1f}$ else. We have the following:

***Theorem** 6.6:* With the above notations, the matrix $\widehat{\mathbf{Y}}_{21}$ can be decomposed into

$$\widehat{\mathbf{Y}}_{21} = \mathbf{M} + \widehat{\mathbf{D}}_1\mathbf{W}_1 + \mathbf{W}_2\widehat{\mathbf{Y}}_{22} + \widehat{\mathbf{E}},$$

where $\mathbf{M}$ is the transmitted codeword, the matrices $\mathbf{W}_1$, $\mathbf{W}_2$ and $\widehat{\mathbf{E}}$ are unknown, $rank\left(\widehat{\mathbf{E}}\right) \leq \beta - \alpha_{0f} - \alpha_{22f}$.

Proof: Set

$$\mathbf{E}'' = \begin{pmatrix} \mathbf{E}_1'' \\ \mathbf{E}_2'' \end{pmatrix},$$

where $\mathbf{E}_1''$ and $\mathbf{E}_2''$ are submatrices of $\mathbf{E}''$ of sizes $\alpha_1 \times n$ and $\left(m_r' - \alpha_1\right) \times n$, respectively. By (14), we have

$$\begin{pmatrix} \mathbf{Y}_{21}'' \\ \mathbf{Y}_{22}'' \end{pmatrix} = \begin{pmatrix} \mathbf{D}_1' \\ \mathbf{0} \end{pmatrix} \mathbf{M}' + \begin{pmatrix} \mathbf{E}_1'' \\ \mathbf{E}_2'' \end{pmatrix}.$$

Thus,

$$\mathbf{Y}_{21}'' = \mathbf{D}_1'\mathbf{M}' + \mathbf{E}_1'' \tag{15}$$

and

$$\mathbf{Y}_{22}'' = \mathbf{E}_2''.$$

- Assume that $freerank\left(\mathbf{Y}_{22}''\right) \neq 0$. As $\mathbf{Y}_{22}''' = \mathbf{U}\mathbf{Y}_{22}''$, set $\mathbf{E}''' := \begin{pmatrix} \mathbf{I}_{\alpha_1} & \mathbf{0} \\ \mathbf{0} & \mathbf{U} \end{pmatrix} \mathbf{E}''$. Then,

$rank\left(\mathbf{E}'''\right) \leq rank\left(\mathbf{E}''\right) \leq \beta'$ and $\mathbf{E}''' = \begin{pmatrix} \mathbf{E}_1'' \\ \mathbf{Y}_{22}''' \end{pmatrix}$. Since $freerank\left(\mathbf{Y}_{22}'''\right) = \alpha_{22f}$, by [42, Proposition 2.11], there are $\left(n - \alpha_{22f}\right) \times n$ matrix $\mathbf{Y}_3$, $n \times \left(n - \alpha_{22f}\right)$ matrix $\mathbf{F}_1$ and $n \times \alpha_{22f}$ matrix $\mathbf{F}_2$ such that

$$\begin{pmatrix} \mathbf{Y}_3 \\ \mathbf{Y}_{22}''' \end{pmatrix} \begin{pmatrix} \mathbf{F}_1 & \mathbf{F}_2 \end{pmatrix} = \mathbf{I}_n.$$

As

$$\begin{aligned} \mathbf{I}_n &= \begin{pmatrix} \mathbf{F}_1 & \mathbf{F}_2 \end{pmatrix} \begin{pmatrix} \mathbf{Y}_3 \\ \mathbf{Y}_{22}''' \end{pmatrix} \\ &= \mathbf{F}_1\mathbf{Y}_3 + \mathbf{F}_2\mathbf{Y}_{22}''', \end{aligned}$$

we have

$$\mathbf{E}_1'' = \mathbf{E}_1''\mathbf{F}_1\mathbf{Y}_3 + \mathbf{E}_1''\mathbf{F}_2\mathbf{Y}_{22}''',$$

that is,

$$\mathbf{E}_1'' = \mathbf{E}_3 + \mathbf{E}_4\mathbf{Y}_{22}''', \tag{16}$$

where $\mathbf{E}_3 = \mathbf{E}_1''\mathbf{F}_1\mathbf{Y}_3$ and $\mathbf{E}_4 = \mathbf{E}_1''\mathbf{F}_2$. Moreover, since

$$\mathbf{E}''' \begin{pmatrix} \mathbf{F}_1 & \mathbf{F}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{E}_1''\mathbf{F}_1 & \mathbf{E}_1''\mathbf{F}_2 \\ \mathbf{0} & \mathbf{I}_{\alpha_{22f}} \end{pmatrix},$$

we have,

$$rank\,(\mathbf{E}_3) \leq rank\left(\mathbf{E}_1''\mathbf{F}_1\right) = rank\left(\mathbf{E}'''\right) - \alpha_{22f} \leq \beta' - \alpha_{22f}.$$

By (15) and (16),

$$\mathbf{Y}''_{21} = \mathbf{D}'_1\mathbf{M}' + \mathbf{E}_4\mathbf{Y}'''_{22} + \mathbf{E}_3.$$

Let $\mathbf{E}'_4$ be the $m \times \alpha_{22f}$ matrix and $\mathbf{E}'_3$ be the $m \times n$ matrix obtained respectively from matrices $\mathbf{E}_4$ and $\mathbf{E}_3$ by inserting all-zero rows below the last row if $\alpha_1 \leq m$ and by deleting the $\alpha_1 - m$ last rows else. Then,

$$\mathbf{Y}'''_{21} = \mathbf{D}''_1\mathbf{M}' + \mathbf{E}'_4\mathbf{Y}'''_{22} + \mathbf{E}'_3. \tag{17}$$

If we left multiply both sides of (17) by $\mathbf{Q}'$ we get

$$\widehat{\mathbf{Y}}_{21} = \mathbf{M} + \widehat{\mathbf{D}}_1\mathbf{W}_1 + \mathbf{W}_2\widehat{\mathbf{Y}}_{22} + \widehat{\mathbf{E}}.$$

where $\mathbf{W}_1 = \mathbf{M}'$, $\mathbf{W}_2 = \mathbf{Q}'\mathbf{E}'_4$ and $\widehat{\mathbf{E}} = \mathbf{Q}'\mathbf{E}'_3$.

• Assume that $freerank\,(\mathbf{Y}_{22}) = 0$. Then, by (15), we have

$$\widehat{\mathbf{Y}}_{21} = \mathbf{M} + \widehat{\mathbf{D}}_1\mathbf{W}_1 + \widehat{\mathbf{E}},$$

where $\mathbf{W}_1$ is defined as above and $\widehat{\mathbf{E}} = \mathbf{Q}'\mathbf{E}_5$, where $\mathbf{E}_5$ is the $m \times n$ matrix obtained from the matrix $\mathbf{E}''_1$ by inserting all-zero rows below the last row if $\alpha_1 \leq m$ or by deleting the $\alpha_1 - m$ last rows else. ■

Theorem 6.6 and Corollary 5.9 imply the following result.

***Corollary** 6.7:* With the above notations, assume that $\mathcal{M}$ is the matrix representation of an interleaved Gabidulin code of rank distance d. If

$$rank\left(\widehat{\mathbf{D}}_1\right) + rank\left(\widehat{\mathbf{Y}}_{22}\right) + 2rank\left(\widehat{\mathbf{E}}\right) \leq d - 1,$$

then the transmitted codeword can be recovered.

***Example** 6.8:* See Appendix.

VII. Conclusion

We have studied some properties of rank-metric codes that are extended from the case of finite fields to finite principal ideal rings. We have first generalized the rank metric and established the rank-metric Singleton bound. As in the case of finite fields, we have shown that Gabidulin codes achieve this bound and that collaborative decoding of interleaved Gabidulin codes can be translated to the problem of reconstruction of skew polynomials. We have used the theory of Gröbner bases of modules over skew polynomials to give the unique decoding, minimal list decoding, and error-erasure decoding algorithms of interleaved Gabidulin codes. These codes are then applied in space-time coding and in random linear network coding. Specifically, we have shown that there is a rank-preserving map from a finite principal ideal ring to a complex signal set and we have used it to construct an optimal space-time block code. Using the lifting construction, we have shown that the decoding problem for random linear network coding over finite principal ideal rings can be reformulated as an error-erasure decoding problem for rank-metric codes.

Analogous to the case of finite fields, we have given an iterative algorithm that can uniquely decode interleaved Gabidulin codes beyond the error correction capability. It would be interesting to study the complexity and the failure probability of this algorithm.

Appendix
Example

The following example exemplifies the application to random linear network codes from Section VI-B. It was computed in SageMathCloud [28].

Let $R = \mathbb{Z}_8$, $S = R[z]/\left(z^5 + 4z^3 + 7z^2 + 2z + 7\right)$ and $a = z + \left(z^5 + 4z^3 + 7z^2 + 2z + 7\right)$. Then S is a Galois extension of R where the Galois group is generated by a power map $\sigma : a \mapsto a^2$. Set $\mathbf{g}^{(1)} = \mathbf{g}^{(2)} = \left(a, a^2, a^3, a^4, a^5\right)$; $f^{(1)} = 1 + 2a + 3a^2 + 5a^3$; $f^{(2)} = 1 + 4a + 7a^2 + 2a^3 + 5a^4$; $\mathbf{c}^{(1)} = f^{(1)}\left(\mathbf{g}^{(1)}\right)$; $\mathbf{c}^{(2)} = f^{(2)}\left(\mathbf{g}^{(2)}\right)$. Then $\left(\mathbf{c}^{(1)} \quad \mathbf{c}^{(2)}\right)$ is a codeword of the interleaved Gabidulin code $IGab_{(1,1)}\left(\mathbf{g}^{(1)}, \mathbf{g}^{(2)}\right)$. Let

$$\mathbf{M} = \begin{pmatrix}\mathbf{M}_1 & \mathbf{M}_2\end{pmatrix}$$

where $\mathbf{M}_1$ and $\mathbf{M}_2$ are respectively the matrix representations of $\mathbf{c}^{(1)}$ and $\mathbf{c}^{(2)}$ in the basis $\left(1, a, a^2, a^3, a^4\right)$.

The transmitted matrix is

$$\mathbf{X} = \begin{pmatrix}\mathbf{0}_{5\times 2} & \mathbf{I}_5 & \mathbf{M}\end{pmatrix}.$$

Assume that

$$\mathbf{A} = \begin{pmatrix} 5 & 6 & 6 & 3 & 3 \\ 3 & 2 & 7 & 1 & 0 \\ 4 & 6 & 0 & 6 & 7 \\ 4 & 1 & 2 & 1 & 0 \\ 1 & 4 & 5 & 6 & 2 \\ 2 & 5 & 7 & 5 & 0 \\ 4 & 4 & 1 & 3 & 1 \end{pmatrix}$$

and

$$\mathbf{E} = \mathbf{BZ}$$

where

$$\mathbf{B} = \begin{pmatrix} 6 & 4 & 2 \\ 4 & 5 & 5 \\ 2 & 5 & 4 \\ 6 & 7 & 6 \\ 3 & 7 & 2 \\ 2 & 7 & 1 \\ 6 & 0 & 7 \end{pmatrix}$$

and

$$\mathbf{Z} = \begin{pmatrix}\mathbf{Z}_1 & \mathbf{Z}_2\end{pmatrix}$$

with

$$\mathbf{Z}_1 = \begin{pmatrix} 0 & 7 & 7 & 0 & 6 & 3 & 3 & 1 & 5 \\ 0 & 0 & 7 & 5 & 2 & 4 & 5 & 2 & 3 \\ 6 & 3 & 0 & 5 & 5 & 7 & 2 & 3 & 7 \end{pmatrix}$$

and

$$\mathbf{Z}_2 = \begin{pmatrix} 2 & 6 & 7 & 4 & 3 & 4 & 1 & 2 \\ 0 & 3 & 0 & 4 & 5 & 5 & 6 & 5 \\ 0 & 4 & 3 & 5 & 1 & 5 & 2 & 5 \end{pmatrix}.$$

The received matrix is

$$\mathbf{Y} = \mathbf{AX} + \mathbf{BZ}.$$

By Theorem 6.6, there are the matrices $\mathbf{W}_1$, $\mathbf{W}_2$ and $\widehat{\mathbf{E}}$ such that

$$\widehat{\mathbf{Y}}_{21} = \mathbf{M} + \widehat{\mathbf{D}}_1\mathbf{W}_1 + \mathbf{W}_2\widehat{\mathbf{Y}}_{22} + \widehat{\mathbf{E}} \tag{18}$$

with $rank\left(\widehat{\mathbf{E}}\right) \leq 1$, where

$$\widehat{\mathbf{Y}}_{21} = \begin{pmatrix} 0 & 6 & 5 & 4 & 5 & 7 & 3 & 6 & 4 & 4 \\ 5 & 7 & 5 & 1 & 3 & 5 & 6 & 7 & 4 & 6 \\ 0 & 2 & 4 & 7 & 3 & 5 & 2 & 1 & 0 & 3 \\ 7 & 1 & 7 & 3 & 5 & 7 & 5 & 1 & 2 & 1 \\ 5 & 7 & 3 & 6 & 4 & 0 & 2 & 2 & 0 & 1 \end{pmatrix}$$

$$\widehat{\mathbf{D}}_1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 6 \\ 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 7 \\ 0 & 0 & 0 & 0 & 7 \end{pmatrix}$$

and

$$\widehat{\mathbf{Y}}_{22} = \begin{pmatrix} 0 & 7 & 6 & 2 & 1 & 6 & 7 & 5 & 5 & 1 \end{pmatrix}.$$

The vector representation of (18) in the basis $\left(1, a, a^2, a^3, a^4\right)$ is

$$\mathbf{y} = \mathbf{c} + a^{(R)}\mathbf{B}^{(R)} + \mathbf{a}^{(C)}\mathbf{B}^{(C)} + \boldsymbol{\varepsilon}^{(E)}$$

where $\mathbf{y}$, $\mathbf{c}$, $\mathbf{a}^{(C)}$, $\boldsymbol{\varepsilon}^{(E)}$ are respectively the vector representations of $\widehat{\mathbf{Y}}_{21}$, $\mathbf{M}$, $\mathbf{W}_2$, $\widehat{\mathbf{E}}$ and $\mathbf{B}^{(C)} = \widehat{\mathbf{Y}}_{22}$, $\mathbf{B}^{(R)}$ is the last row of $\mathbf{W}_1$, $a^{(R)} = 7a^4 + 7a^3 + 4a^2 + 6a + 4$.

Set

$$\mathbf{y} = \begin{pmatrix} \mathbf{y}^{(1)} & \mathbf{y}^{(2)} \end{pmatrix}$$

where $\mathbf{y}^{(1)} \in S^5$ and $\mathbf{y}^{(2)} \in S^5$. Then

$$\mathbf{y}^{(1)} = \mathbf{c}^{(1)} + a^{(R)}\mathbf{B}^{(R,1)} + \mathbf{a}^{(C)}\mathbf{B}^{(C,1)} + \boldsymbol{\varepsilon}^{(E,1)}$$

$$\mathbf{y}^{(2)} = \mathbf{c}^{(2)} + a^{(R)}\mathbf{B}^{(R,2)} + \mathbf{a}^{(C)}\mathbf{B}^{(C,2)} + \boldsymbol{\varepsilon}^{(E,2)}.$$

Let

$$P^{(R)} = X + 5a^4 + a^3 + 6a^2 + 2a + 2,$$

$$\mathbf{F}^{(R,1)} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 7 & 6 & 2 & 0 \\ 1 & 2 & 7 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

and

$$\mathbf{F}^{(R,2)} = \begin{pmatrix} 1 & 5 & 5 & 1 \\ 7 & 3 & 3 & 6 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}.$$

Then, $P^{(R)}\left(a^{(R)}\right) = 0$, $\mathbf{B}^{(C,1)}\mathbf{F}^{(R,1)} = \mathbf{0}$ and $\mathbf{B}^{(C,2)}\mathbf{F}^{(R,2)} = \mathbf{0}$.

Set $\mathbf{y}'^{(l)} = P^{(R)}\left(\mathbf{y}^{(l)}\right)\mathbf{F}^{(C,l)}$, $\mathbf{g}'^{(l)} = \mathbf{g}^{(l)}\mathbf{F}^{(C,l)}$, $\mathbf{c}'^{(l)} = P^{(R,l)}\left(\mathbf{c}^{(l)}\right)\mathbf{F}^{(C,l)}$, for $l \in \{1, 2\}$. Thus, by Theorem 5.8, there is $\boldsymbol{\varepsilon}' \in S^8$ such that

$$\begin{pmatrix} \mathbf{y}'^{(1)} & \mathbf{y}'^{(2)} \end{pmatrix} = \begin{pmatrix} \mathbf{c}'^{(1)} & \mathbf{c}'^{(2)} \end{pmatrix} + \boldsymbol{\varepsilon}'$$

where $rank\left(\boldsymbol{\varepsilon}'\right) \leq 1$.

When we apply Algorithm 4 for the received word $\begin{pmatrix} \mathbf{y}'^{(1)} & \mathbf{y}'^{(2)} \end{pmatrix}$ of the interleaved Gabidulin code $IGab_{(2,2)}\left(\mathbf{g}'^{(1)}, \mathbf{g}'^{(2)}\right)$, it returns $\left(f'^{(1)}, f'^{(2)}\right)$ where $f'^{(1)} = (7a^4 + 5a^3 + 5a + 1)X + 4a^4 + 3a^3 + 4a + 1$ and $f'^{(2)} = (5a^4 + 7a^3 + 5a^2 + 4a + 6)X + 2a^4 + 5a^3 + 3a^2 + 5a$. The left Euclidean division of $f'^{(1)}$ and $f'^{(2)}$ by $P^{(R)}$ gives respectively $f^{(1)}$ and $f^{(2)}$.

Acknowledgment

The authors would like to thank the reviewers for their useful remarks and suggestions.

References

[1] P. Delsarte, "Bilinear forms over a finite field, with applications to coding theory," *J. Combinat. Theory, A*, vol. 25, no. 3, pp. 226–241, 1978.

[2] È. M. Gabidulin, "Theory of codes with maximum rank distance," *Problemy Peredachi Informatsii*, vol. 21, no. 1, pp. 3–16, 1985.

[3] R. M. Roth, "Maximum-rank array codes and their application to crisscross error correction," *IEEE Trans. Inf. Theory*, vol. 37, no. 2, pp. 328–336, Mar. 1991.

[4] P. Loidreau and R. Overbeck, "Decoding rank errors beyond the error-correction capability," in *Proc. Int. Workshop Algebraic Combinat. Coding Theory*, Sep. 2006, pp. 168–190.

[5] V. Sidorenko and M. Bossert, "Decoding interleaved Gabidulin codes and multisequence linearized shift-register synthesis," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2010, pp. 1148–1152.

[6] A. Wachter-Zeh and A. Zeh, "List and unique error-erasure decoding of interleaved Gabidulin codes with interpolation techniques," *Des., Codes Cryptogr.*, vol. 73, no. 2, pp. 547–570, 2014.

[7] P. Lusina, E. Gabidulin, and M. Bossert, "Maximum rank distance codes as space-time codes," *IEEE Trans. Inf. Theory*, vol. 49, no. 10, pp. 2757–2760, Oct. 2003.

[8] E. M. Gabidulin, A. V. Paramonov, and O. V. Tretjakov, "Ideals over a non-commutative ring and their application in cryptology," in *Proc. Workshop Theory Appl. Cryptograph. Techn. (EUROCRYPT)*, in Lecture Notes in Computer Science, vol. 547. Berlin, Germany: Springer-Verlag, 1991, pp. 482–489.

[9] D. Silva, F. R. Kschischang, and R. Koetter, "A rank-metric approach to error control in random network coding," *IEEE Trans. Inf. Theory*, vol. 54, no. 9, pp. 3951–3967, Sep. 2008.

[10] V. Tarokh, N. Seshadri, and A. R. Calderbank, "Space-time codes for high data rate wireless communication: Performance criterion and code construction," *IEEE Trans. Inf. Theory*, vol. 44, no. 2, pp. 744–765, Mar. 1998.

[11] H.-F. Lu and P. V. Kumar, "Rate-diversity tradeoff of space-time codes with fixed alphabet and optimal constructions for PSK modulation," *IEEE Trans. Inf. Theory*, vol. 49, no. 10, pp. 2747–2751, Oct. 2003.

[12] T. Kiran and B. S. Rajan, "Optimal STBCs from codes over Galois rings," in *Proc. IEEE Int. Conf. Pers. Wireless Commun.*, Jan. 2005, pp. 120–124.

[13] H. M. Asif, B. Honary, and M. T. Hamayun, "Gaussian integers and interleaved rank codes for space–time block codes," *Int. J. Commun. Syst.*, vol. 30, no. 1, 2017, Art. no. e2943.

[14] S. Puchinger, S. Stern, M. Bossert, and R. F. H. Fischer, "Space-time codes based on rank-metric codes and their decoding," in *Proc. Int. Symp. Wireless Commun. Syst. (ISWCS)*, Sep. 2016, pp. 125–130.

[15] D. Augot, P. Loidreau, and G. Robert, "Rank metric and Gabidulin codes in characteristic zero," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 2013, pp. 509–513.

[16] R. Koetter and F. R. Kschischang, "Coding for errors and erasures in random network coding," *IEEE Trans. Inf. Theory*, vol. 54, no. 8, pp. 3579–3591, Aug. 2008.

[17] C. Feng, D. Silva, and F. Kschischang, "An algebraic approach to physical-layer network coding," *IEEE Trans. Inf. Theory*, vol. 59, no. 11, pp. 7576–7596, Nov. 2013.

[18] B. Nazer and M. Gastpar, "Compute-and-forward: Harnessing interference through structured code," *IEEE Trans. Inf. Theory*, vol. 57, no. 10, pp. 6463–6486, Oct. 2011.

[19] Y. Liu, M. P. Fitz, and O. Y. Takeshita, "A rank criterion for QAM space-time codes," *IEEE Trans. Inf. Theory*, vol. 48, no. 12, pp. 3062–3079, Dec. 2002.

[20] D. Silva, F. R. Kschischang, and R. Kotter, "Communication over finite-field matrix channels," *IEEE Trans. Inf. Theory*, vol. 56, no. 3, pp. 1296–1305, Mar. 2010.

[21] R. W. Nóbrega, B. F. Uchôa-Filho, and D. Silva, "On the capacity of multiplicative finite-field matrix channels," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul./Aug. 2011, pp. 341–345.

[22] C. Feng, R. W. Nóbrega, F. R. Kschischang, and D. Silva, "Communication over finite-chain-ring matrix channels," *IEEE Trans. Inf. Theory*, vol. 60, no. 10, pp. 5899–5917, Oct. 2014.

[23] R. W. Nóbrega, C. Feng, D. Silva, and B. F. Uchôa-Filho, "On multiplicative matrix channels over finite chain rings," in *Proc. Int. Symp. Netw. Coding (NetCod)*, Jun. 2013, pp. 1–6.

[24] E. Gorla and A. Ravagnani, "Partial spreads in random network coding," *Finite Fields Appl.*, vol. 26, pp. 104–115, Mar. 2014.

[25] E. Gorla and A. Ravagnani, "An algebraic framework for end-to-end physical-layer network coding," *IEEE Trans. Inf. Theory*, vol. 64, no. 6, pp. 4480–4495, Jun. 2018.

[26] M. Kuijper and A.-L. Trautmann, "Iterative list-decoding of Gabidulin codes via Gröbner based interpolation," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Nov. 2014, pp. 581–585.

[27] W. C. Brown, *Matrices Over Commutative Rings*. New York, NY, USA: Marcel Dekker, 1993.

[28] I. SageMath, *SageMathCloud Online Computational Mathematics*, 2019. [Online]. Available: https://cloud.sagemath.com/

[29] A. Storjohann, "Algorithms for matrix canonical forms," Ph.D. dissertation, ETH Zürich, Zürich, Switzerland, 2000.

[30] A. T. Butson and B. Stewart, "Systems of linear congruences," *Can. J. Math.*, vol. 7, pp. 358–368, 1955. doi: 10.4153/CJM-1955-039-2.

[31] A. A. Nechaev, "Finite rings with applications," in *Handbook of Algebra*, vol. 5. Amsterdam, The Netherlands: North Holland, 2008, pp. 213–320.

[32] B. R. McDonald, *Finite Rings With Identity*, vol. 28. New York, NY, USA: Marcel Dekker, 1974.

[33] D. M. Goldschmidt, *Algebraic Functions and Projective Curves* (Graduate Texts in Mathematics), vol. 215. New York, NY, USA: Springer-Verlag, 2003.

[34] F. DeMeyer and E. Ingraham, *Separable Algebras Over Commutative Rings* (Lecture Notes in Mathematics), vol. 181. Berlin, Germany: Springer-Verlag, 1971.

[35] A. A. De Andrade and R. Palazzo, Jr., "Construction and decoding of BCH codes over finite commutative rings," *Linear Algebra Appl.*, vol. 286, nos. 1–3, pp. 69–85, Jan. 1999.

[36] O. Ore, "On a special class of polynomials," *Trans. Amer. Math. Soc.*, vol. 35, no. 3, pp. 559–584, Jul. 1933.

[37] J. L. Bueso, J. Gómez-Torrecillas, and A. Verschoren, *Algorithmic Methods in Non-Commutative Algebra. Applications to Quantum Groups*. Dordrecht, The Netherlands: Kluwer, 2003.

[38] H. Jiménez and O. Lezama, "Gröbner bases for modules over σ-PBW extensions," *Acta Math. Academiae Paedagogicae Nyiregyháziensis*, vol. 31, no. 3, pp. 39–66, 2015.

[39] E. R. Kolchin, *Differential Algebra & Algebraic Groups*, vol. 54. New York, NY, USA: Academic, 1973.

[40] E. M. Gabidulin, "Rank-metric codes and applications," Moscow Inst. Phys. Technol., State Univ., Dolgoprudny, Russia. [Online]. Available: http://iitp.ru/upload/content/839/Gabidulin.pdf

[41] T. Y. Lam, *Lectures on modules and rings* (Graduate Texts in Mathematics), 1st ed. New York, NY, USA: Springer-Verlag, 1999.

[42] Y. Fan, S. Ling, and H. Liu, "Matrix product codes over finite commutative Frobenius rings," *Des., Codes Cryptogr.*, vol. 71, no. 2, pp. 201–227, 2014.

[43] H.-F. Lu and P. V. Kumar, "A unified construction of space-time codes with optimal rate-diversity tradeoff," *IEEE Trans. Inf. Theory*, vol. 51, no. 5, pp. 1709–1730, May 2005.

[44] D. Augot, P. Loidreau, and G. Robert, "Generalized Gabidulin codes over fields of any characteristic," *Des., Codes Cryptogr.*, vol. 86, no. 8, pp. 1807–1848, Aug. 2018.

[45] P. Loidreau, "A Welch–Berlekamp like algorithm for decoding Gabidulin codes," in *Proc. 4th Int. Workshop Coding Cryptogr. (WCC)*, in Lecture Notes in Computer Science, vol. 3969. Berlin, Germany: Springer, 2006, pp. 36–45.

[46] A. Wachter-Zeh, "Decoding of block and convolutional codes in rank metric," Ph.D. dissertation, Ulm Univ., Ulm, Germany, Univ. Rennes 1, Rennes, France, 2013. [Online]. Available: https://tel.archives-ouvertes.fr/tel-01056746

[47] S. Puchinger, J. R. né Nielsen, W. Li, and V. Sidorenko, "Row reduction applied to decoding of rank-metric and subspace codes," *Des., Codes Cryptogr.*, vol. 82, nos. 1–2, pp. 389–409, 2017.

[48] P. Fitzpatrick, "On the key equation," *IEEE Trans. Inf. Theory*, vol. 41, no. 5, pp. 1290–1302, Sep. 1995.

[49] H. O'Keeffe and P. Fitzpatrick, "Gröbner basis solutions of constrained interpolation problems," *Linear Algebra Appl.*, vols. 351–352, pp. 533–551, Aug. 2002.

[50] M. A. Armand, "List decoding of generalized Reed–Solomon codes over commutative rings," *IEEE Trans. Inf. Theory*, vol. 51, no. 1, pp. 411–419, Jan. 2005.

[51] M. Kuijper and R. Pinto, "An iterative algorithm for parametrization of shortest length linear shift registers over finite chain rings," *Des., Codes Cryptogr.*, vol. 83, no. 2, pp. 283–305, 2017.

[52] H. Bartz and A. Wachter-Zeh, "Efficient decoding of interleaved subspace and Gabidulin codes beyond their unique decoding radius using Gröbner bases," *Adv. Math. Commun.*, vol. 12, no. 4, pp. 773–804, 2018.

[53] E. Byrne and P. Fitzpatrick, "Hamming metric decoding of alternant codes over Galois rings," *IEEE Trans. Inf. Theory*, vol. 48, no. 3, pp. 683–694, Mar. 2002.

[54] G. H. Norton and A. Sălăgean, "Gröbner bases and products of coefficient rings," *Bull. Austral. Math. Soc.*, vol. 65, no. 1, pp. 145–152, Feb. 2002.

[55] C. J. Rust and G. J. Reid, "Rankings of partial derivatives," in *Proc. Int. Symp. Symbolic Algebr. Comput.*, Jul. 1997, pp. 9–16.

[56] V. Sidorenko, L. Jiang, and M. Bossert, "Skew-feedback shift-register synthesis and decoding interleaved Gabidulin codes," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 621–632, Feb. 2011.

[57] E. M. Gabidulin, N. I. Pilipchuk, and M. Bossert, "Decoding of random network codes," *Problems Inf. Transmiss.*, vol. 46, pp. 300–320, Dec. 2010.

[58] H. Bartz and V. Sidorenko, "Improved syndrome decoding of lifted L-interleaved Gabidulin codes," *Des., Codes Cryptogr.*, vol. 87, nos. 2–3, pp. 547–567, 2019.

Hermann Tchatchiem Kamche received the B.A. and M.Sc. degrees in mathematics from the University of Yaoundé I, Yaoundé, Cameroon, in 2008 and 2010, respectively.

He is currently working toward the Ph.D. degree at the University of Yaoundé I. His research interests include cryptography, channel coding and network coding.

Christophe Mouaha received the B.S. and M.S. degrees in mathematics from the University of Yaoundé I, Yaoundé, Cameroon, in 1983 and 1985, respectively and the Ph.D. degree in 1988 from the University of Marseille II, Marseille, France, in the field of algebraic coding theory.

Currently, he is a Lecturer at the Higher Teacher Training School, University of Yaoundé I. His current scientific interests include coding theory and cryptography.