

# Iterative List-Decoding of Gabidulin Codes via Gröbner Based Interpolation

Margreta Kuijper and Anna-Lena Trautmann

Department of Electrical and Electronic Engineering, University of Melbourne, Australia.

**Abstract**—We show how Gabidulin codes can be list decoded by using an iterative parametrization approach. For a given received word, our decoding algorithm processes its entries one by one, constructing four polynomials at each step. This then yields a parametrization of interpolating solutions for the data so far. From the final result a list of all codewords that are closest to the received word with respect to the rank metric is obtained.

## I. INTRODUCTION

Over the last decade there has been increased interest in Gabidulin codes, mainly because of their relevance to network coding [5], [20]. Gabidulin codes are optimal rank-metric nonbinary codes over a field  $\mathbb{F}_q^m$  (where  $q$  is a prime power). They were first derived by Gabidulin in [3] and independently by Delsarte in [2]. These codes can be seen as the  $q$ -analog of Reed-Solomon codes, using  $q$ -linearized polynomials instead of arbitrary polynomials. They are optimal in the sense that they are not only MDS codes with respect to the Hamming metric, but also achieve the Singleton bound with respect to the rank metric and are thus MRD codes. They are not only of interest in network coding but also in space-time coding [11], crisscross error correction [15] and distributed storage [18].

The decoding of Gabidulin codes has obtained a fair amount of attention in the literature, starting with work on decoding inside the unique decoding radius in [3], [4] and more recently [10], [14], [16], [17], [21]. Decoding beyond the unique decoding radius was investigated in e.g. [9], [5], [12], [24], [25]. Related work on list-decoding of lifted Gabidulin codes can be found in [22].

Using the close resemblance between Reed-Solomon codes and Gabidulin codes, the paper [10] translates Gabidulin decoding into a set of polynomial interpolation conditions. Essentially, this setup is also used in the papers [5], [25] that present iterative algorithms that perform Gabidulin list decoding with a list size of 1. In this paper we present an iterative algorithm that bears similarity to the ones in [10], [5], [25] but yields *all* closest codewords rather than just one. The latter is due to our parametrization approach.

The paper is structured as follows. In the next section we present several preliminaries on  $q$ -linearized polynomials, Gabidulin codes, the rank metric and we recall the polynomial interpolation conditions from [10]. We also detail the iterative construction of the  $q$ -annihilator polynomial and the  $q$ -Lagrange polynomial. Section II closes with several preliminaries on Gröbner bases. In Section III we reformulate the Gabidulin list decoding requirements in terms of a module represented by four  $q$ -linearized polynomials. In Section IV we present the algorithm and our main result which details how the algorithm yields a list of all closest message polynomials. We conclude this paper in Section V.

ALT is also with the Department of Electrical and Computer Systems Engineering, Monash University. She was supported by Swiss National Science Foundation Fellowship no. 147304.

## II. PRELIMINARIES

### A. $q$ -linearized polynomials

Let  $q$  be a prime power and let  $\mathbb{F}_q$  denote the finite field with  $q$  elements. It is well-known that there always exists a primitive element  $\alpha$  of the extension field  $\mathbb{F}_{q^m}$ , such that  $\mathbb{F}_{q^m} \cong \mathbb{F}_q[\alpha]$ . Moreover,  $\mathbb{F}_{q^m}$  is isomorphic (as a vector space) to the vector space  $\mathbb{F}_q^m$ . One then easily gets the isomorphic description of matrices over the base field  $\mathbb{F}_q$  as vectors over the extension field, i.e.  $\mathbb{F}_q^{m \times n} \cong \mathbb{F}_{q^m}^n$ . Since we will work with matrices over different underlying fields we denote the rank of a matrix  $X$  over  $\mathbb{F}_q$  by  $\text{rank}_q(X)$ .

For some vector  $(v_1, \dots, v_n) \in \mathbb{F}_{q^m}^n$  we denote the  $k \times n$  Moore matrix by

$$M_k(v_1, \dots, v_n) := \begin{pmatrix} v_1 & v_2 & \dots & v_n \\ v_1^{[1]} & v_2^{[1]} & \dots & v_n^{[1]} \\ \vdots & \vdots & \ddots & \vdots \\ v_1^{[k-1]} & v_2^{[k-1]} & \dots & v_n^{[k-1]} \end{pmatrix},$$

where  $[i] := q^i$ . A  $q$ -linearized polynomial over  $\mathbb{F}_{q^m}$  is defined to be of the form

$$f(x) = \sum_{i=0}^n a_i x^{[i]}, \quad a_i \in \mathbb{F}_{q^m},$$

where  $n$  is called the  $q$ -degree of  $f(x)$ , assuming that  $a_n \neq 0$ , denoted by  $\text{qdeg}(f)$ . This class of polynomials was first studied by Ore in [13]. One can easily check that  $f(x_1 + x_2) = f(x_1) + f(x_2)$  and  $f(\lambda x_1) = \lambda f(x_1)$  for any  $x_1, x_2 \in \mathbb{F}_{q^m}$  and  $\lambda \in \mathbb{F}_q$ , hence the name *linearized*. The set of all  $q$ -linearized polynomials over  $\mathbb{F}_{q^m}$  is denoted by  $\mathcal{L}_q(x, q^m)$ . This set forms a non-commutative ring with the normal addition  $+$  and composition  $\circ$  of polynomials. Because of the non-commutativity, products and quotients of elements of  $\mathcal{L}_q(x, q^m)$  have to be specified as being “left” or “right” products or quotients. To not be mistaken with the standard division, we call the inverse of the composition *symbolic division*. I.e.  $f(x)$  is symbolically divisible by  $g(x)$  with right quotient  $m(x)$  if

$$g(x) \circ m(x) = g(m(x)) = f(x).$$

Efficient algorithms for all these operations (left and right symbolic multiplication and division) exist and can be found e.g. in [5].

**Lemma 1** (cf. [8] Thm. 3.50). *Let  $f(x) \in \mathcal{L}_q(x, q^m)$  and  $\mathbb{F}_{q^s}$  be the smallest extension field of  $\mathbb{F}_{q^m}$  that contains all roots of  $f(x)$ . Then the set of all roots of  $f(x)$  forms a  $\mathbb{F}_q$ -linear vector space in  $\mathbb{F}_{q^s}$ .*

**Lemma 2** ([8] Thm. 3.52). *Let  $U$  be a  $\mathbb{F}_q$ -linear subspace of  $\mathbb{F}_{q^m}$ . Then  $\prod_{g \in U} (x - g)$  is an element of  $\mathcal{L}_q(x, q^m)$ .*

Note that, if  $g_1, \dots, g_n$  is a basis of  $U$ , one can rewrite

$$\prod_{g \in U} (x - g) = \lambda \det(M_{t+1}(g_1, \dots, g_n, x))$$

for some constant  $\lambda \in \mathbb{F}_{q^m}$ . We call this polynomial the  $q$ -annihilator polynomial of  $U$ , denoted by  $\Pi_{(g_1, g_2, \dots, g_n)}(x)$ . Clearly its  $q$ -degree equals  $n$ .

We also have a notion of  $q$ -Lagrange polynomial: Let  $\mathbf{g} = (g_1, \dots, g_n)$  and  $\mathbf{r} = (r_1, \dots, r_n)$ . Define the matrix  $\mathfrak{D}_i(\mathbf{g}, x)$  as  $M_n(g_1, \dots, g_n, x)$  without the  $i$ -th column. We define the  $q$ -Lagrange polynomial as

$$\Lambda_{\mathbf{g}, \mathbf{r}}(x) := \sum_{i=1}^n (-1)^{n-i} r_i \frac{\det(\mathfrak{D}_i(\mathbf{g}, x))}{\det(M_n(\mathbf{g}))} \in \mathbb{F}_{q^m}[x].$$

It can be easily verified that the above polynomial is  $q$ -linearized and that  $\Lambda_{\mathbf{g}, \mathbf{r}}(g_i) = r_i$  for  $i = 1, \dots, n$ .

Note that, although not under the same name, the previous two polynomials were also defined in e.g. [23].

In the following we will use matrix composition, which is defined analogously to matrix multiplication:

$$\begin{bmatrix} a(x) & b(x) \\ c(x) & d(x) \end{bmatrix} \circ \begin{bmatrix} e(x) & f(x) \\ g(x) & h(x) \end{bmatrix} := \begin{bmatrix} a(e(x)) + b(g(x)) & a(f(x)) + b(h(x)) \\ c(e(x)) + d(g(x)) & c(f(x)) + d(h(x)) \end{bmatrix}.$$

We can recursively construct the  $q$ -annihilator and the  $q$ -Lagrange polynomial as follows.

**Proposition 3.** *Let  $g_1, \dots, g_n \in \mathbb{F}_{q^m}$  be linearly independent and  $r_1, \dots, r_n \in \mathbb{F}_{q^m}$ . Define*

$$\Pi_1(x) := x^q - g_1^{q-1}x \quad , \quad \Lambda_1(x) := \frac{r_1}{g_1}x,$$

and for  $i = 1, \dots, n-1$

$$\begin{bmatrix} \Pi_{i+1}(x) \\ \Lambda_{i+1}(x) \end{bmatrix} := \begin{bmatrix} x^q - \Pi_i(g_{i+1})^{q-1}x & 0 \\ -\frac{\Lambda_i(g_{i+1}) - r_{i+1}}{\Pi_i(g_{i+1})}x & x \end{bmatrix} \circ \begin{bmatrix} \Pi_i(x) \\ \Lambda_i(x) \end{bmatrix}.$$

Then for  $i = 1, \dots, n$  we have  $\Pi_i(x) = \Pi_{(g_1, g_2, \dots, g_i)}(x)$  and  $\Lambda_i(x) = \Lambda_{(g_1, g_2, \dots, g_i), (r_1, \dots, r_i)}(x)$ .

*Proof:* We prove this by induction on  $i$ . The theorem clearly holds for  $i = 1$ . Suppose that the theorem holds for a value of  $i$  with  $1 \leq i < n$ . By definition  $\Pi_{i+1}(x) = \Pi_i(x)^q - \Pi_i(g_{i+1})^{q-1}\Pi_i(x)$ , so that (using the induction hypothesis)  $\Pi_{i+1}(x)$  is a monic  $q$ -linearized polynomial of  $q$ -degree  $i+1$  such that for  $1 \leq j \leq i+1$  we have  $\Pi_{i+1}(g_j) = 0$ . It follows that then  $\Pi_{i+1}(x)$  must coincide with  $\Pi_{(g_1, g_2, \dots, g_{i+1})}(x)$ .

We next show that the formula for  $\Lambda_{i+1}(x)$  yields the  $q$ -Lagrange polynomial at level  $i+1$ . Assume that  $\Lambda_i(x)$  is the  $q$ -Lagrange polynomial at level  $i$  and look at  $\Lambda_{i+1}(x)$ , which is  $q$ -linearized since  $\Lambda_i(x)$  and  $\Pi_i(x)$  are  $q$ -linearized. As  $\text{qdeg}(\Pi_i(x)) = i > \text{qdeg}(\Lambda_i(x))$  it holds that  $\text{qdeg}(\Lambda_{i+1}(x)) = i$ . Furthermore, because  $\Pi_i(g_j) = 0$  for  $j = 1, \dots, i$  and  $\Lambda_i(g_j) = r_j$  for  $j = 1, \dots, i$ ,

$$\Lambda_{i+1}(g_j) = \Lambda_i(g_j) = r_j, \quad \text{and}$$

$$\Lambda_{i+1}(g_{i+1}) = \Lambda_i(g_{i+1}) - \frac{\Lambda_i(g_{i+1}) - r_{i+1}}{\Pi_i(g_{i+1})}\Pi_i(g_{i+1}) = r_{i+1}.$$

Therefore,  $\Lambda_{i+1}(x)$  evaluates in the same values as  $\Lambda_{(g_1, \dots, g_{i+1}), (r_1, \dots, r_{i+1})}(x)$  for  $g_1, \dots, g_{i+1}$ . Because of the linearity of both these polynomials they evaluate in the same values for all elements of  $\langle g_1, \dots, g_{i+1} \rangle$ , and as the  $g_i$  are linearly independent, these are  $q^{i+1}$  many values. Since the degree of both polynomials is  $q^i < q^{i+1}$ , it follows that they must be the same polynomial.  $\blacksquare$

## B. Gabidulin codes

Let  $g_1, \dots, g_n \in \mathbb{F}_{q^m}$  be linearly independent over  $\mathbb{F}_q$ . We define a *Gabidulin code*  $C \subseteq \mathbb{F}_{q^m}^n$  as the linear block code with generator matrix  $M_k(g_1, \dots, g_n)$ . Using the isomorphic matrix representation we can interpret  $C$  as a matrix code in  $\mathbb{F}_q^{m \times n}$ . The *rank distance*  $d_R$  on  $\mathbb{F}_q^{m \times n}$  is defined by

$$d_R(X, Y) := \text{rank}_q(X - Y) \quad , \quad X, Y \in \mathbb{F}_q^{m \times n}$$

and analogously for the isomorphic extension field representation. It holds that the code  $C$  constructed before has dimension  $k$  over  $\mathbb{F}_{q^m}$  and minimum rank distance (over  $\mathbb{F}_q$ )  $n - k + 1$ . One can easily see by the shape of the parity check and the generator matrices that an equivalent definition of the code is

$$C = \{(f(g_1), \dots, f(g_n)) \in \mathbb{F}_{q^m}^n \mid f(x) \in \mathcal{L}_q(x, q^m)_{<k}\},$$

where  $\mathcal{L}_q(x, q^m)_{<k} := \{f(x) \in \mathcal{L}_q(x, q^m), \text{qdeg}(f(x)) < k\}$ . For more information on bounds and constructions of rank-metric codes the interested reader is referred to [3].

Consider a received word  $\mathbf{r} = (r_1, \dots, r_n) \in \mathbb{F}_{q^m}^n$  as the sum  $\mathbf{r} = \mathbf{c} + \mathbf{e}$ , where  $\mathbf{c} = (c_1, \dots, c_n) \in C$  is a codeword and  $\mathbf{e} = (e_1, \dots, e_n) \in \mathbb{F}_{q^m}^n$  is the error vector. We now recall the polynomial interpolation setup from [10] via a more general formulation in the next theorem.

**Theorem 4** ([7], [10]). *Let  $f(x) \in \mathcal{L}_q(x, q^m)$ ,  $\text{qdeg}(f(x)) < k$  and  $c_i = f(g_i)$  for  $i = 1, \dots, n$ . Then  $d_R(\mathbf{c}, \mathbf{r}) = t$  if and only if there exists a  $D(x) \in \mathcal{L}_q(x, q^m)$ , such that  $\text{qdeg}(D(x)) = t$  and*

$$D(r_i) = D(f(g_i)) \quad \forall i \in \{1, \dots, n\}.$$

Furthermore, this  $D(x)$  is unique.

**Remark 5.** The previous theorem states that the roots of  $D(x)$  form a vector space of degree  $t$  which is equal to the span of  $e_1, \dots, e_n$  (for this note that  $e_i = f(g_i) - r_i$ ). This is why  $D(x)$  is also called the *error span polynomial* (cf. e.g. [19]). The analogy in the classical Hamming metric set-up is the *error locator polynomial*, whose roots indicate the locations of the errors, and whose degree equals the number of errors.

## C. Gröbner bases

We will now recall some definitions and results on Gröbner bases of  $\mathcal{L}_q(x, q^m)^2$ , since we will need them later on in this paper. Elements of  $\mathcal{L}_q(x, q^m)^2$  are of the form

$$[f(x) \ g(x)] = f(x)e_1 + g(x)e_2$$

where  $f(x) = \sum f_i x^i, g(x) = \sum g_i x^i \in \mathcal{L}_q(x, q^m)$  and  $e_1, e_2$  are the two unit vectors of length 2.

**Definition 6.** The  $(k_1, k_2)$ -weighted  $q$ -degree of  $[f(x) \ g(x)]$  is defined as  $\max\{k_1 + \text{qdeg}(f(x)), k_2 + \text{qdeg}(g(x))\}$ .

The monomials of  $[f(x) \ g(x)]$  are of the form  $x^{[i]}e_1$  and  $x^{[j]}e_2$  for all  $i$  such that  $f_i \neq 0$  and  $j$  such that  $g_j \neq 0$ , respectively.

**Definition 7.** The *term-over-position (TOP) monomial order* is defined as

$$x^{[i_1]}e_{j_1} < x^{[i_2]}e_{j_2} : \iff i_1 < i_2 \text{ or } [i_1 = i_2 \text{ and } j_1 < j_2].$$

The  $(k_1, k_2)$ -weighted TOP monomial order is defined as

$$x^{[i_1]}e_{j_1} <_{(k_1, k_2)} x^{[i_2]}e_{j_2} : \iff$$

$$i_1 + k_{j_1} < i_2 + k_{j_2} \text{ or } [i_1 + k_{j_1} = i_2 + k_{j_2} \text{ and } j_1 < j_2].$$

We can order all monomials of an element  $V \in \mathcal{L}_q(x, q^m)^2$  in decreasing order with respect to the (weighted or non-weighted) TOP monomial order. Rename them such that  $x^{[i_1]}e_{j_1} > x^{[i_2]}e_{j_2} > \dots$ . Then

- 1) the *leading monomial*  $\text{lm}(V) = x^{[i_1]}$  is the greatest monomial of  $V$ .
- 2) the *leading position*  $\text{lpos}(V) = j_1$  is the vector coordinate of the leading monomial.

**Definition 8.** Let  $M \subseteq \mathcal{L}_q(x, q^m)^2$  be a left module. A subset  $B \subset M$  is called a *Gröbner basis* of  $M$  if the leading monomials of  $B$  span a left module that contains all leading monomials of  $M$ .

In the context of this paper we are only interested in modules with a basis consisting of two vectors, say  $b_1, b_2 \in \mathcal{L}_q(x, q^m)^2$ . It can be easily seen from Definition 8 that such a basis  $\{b_1, b_2\}$  is a Gröbner basis if and only if  $\text{lpos}(b_1) \neq \text{lpos}(b_2)$ . In fact, for this restricted special case such a basis coincides with a *minimal* Gröbner basis (see e.g. [6]).

### III. ITERATIVE CONSTRUCTION OF THE INTERPOLATION MODULE

For the remainder of the paper let  $g_1, \dots, g_n \in \mathbb{F}_{q^m}$  be linearly independent over  $\mathbb{F}_q$  and let  $M_k(g_1, \dots, g_n)$  be the generator matrix of the Gabidulin code  $C \subseteq \mathbb{F}_{q^m}^n$ . Let  $\mathbf{r} = (r_1, \dots, r_n) \in \mathbb{F}_{q^m}^n$  be the received word and denote  $\mathbf{g} = (g_1, \dots, g_n)$ . Furthermore we need the following fact.

**Lemma 9** ([7]). *Let  $L(x) \in \mathcal{L}_q(x, q^m)$ , such that  $L(g_i) = 0$  for all  $i$ . Then*

$$\exists H(x) \in \mathcal{L}_q(x, q^m) : L(x) = H(x) \circ \prod_{g \in \langle g_1, \dots, g_n \rangle} (x - g).$$

In the following we abbreviate the row span of a (polynomial) matrix  $A$  by  $\text{rs}(A)$ .

**Definition 10.** The *interpolation module*  $\mathfrak{M}(\mathbf{r})$  for  $\mathbf{r}$  is defined as the left submodule of  $\mathcal{L}_q(x, q^m)$ , given by

$$\mathfrak{M}(\mathbf{r}) := \text{rs} \begin{bmatrix} \Pi_{\mathbf{g}}(x) & 0 \\ -\Lambda_{\mathbf{g}, \mathbf{r}}(x) & x \end{bmatrix}.$$

We identify any  $[f(x) \ g(x)] \in \mathfrak{M}(\mathbf{r})$  with the bivariate linearized  $q$ -polynomial  $Q(x, y) = f(x) + g(y)$ . It was shown in our recent paper [7] that the name interpolation module is justified for  $\mathfrak{M}(\mathbf{r})$ :

**Theorem 11** ([7]).  *$\mathfrak{M}(\mathbf{r})$  consists exactly of all  $Q(x, y) = f(x) + g(y)$  with  $f(x), g(x) \in \mathcal{L}_q(x, q^m)$ , such that  $Q(g_i, r_i) = 0$  for  $i = 1, \dots, n$ .*

The following statements can also be found in our recent paper [7]:

**Theorem 12.** *The elements  $[N(x) \ -D(x)]$  of  $\mathfrak{M}(\mathbf{r})$  that fulfill*

- 1)  $\text{qdeg}(N(x)) \leq t + k - 1$ ,
- 2)  $\text{qdeg}(D(x)) = t$ ,
- 3)  $N(x)$  is symbolically divisible on the right by  $D(x)$ , i.e. there exists  $f(x) \in \mathcal{L}_q(x, q^m)$  such that  $D(f(x)) = N(x)$ ,

are in one-to-one correspondence with the codewords of rank distance  $t$  to  $\mathbf{r}$ .

Therefore, list decoding within rank radius  $t$  is equivalent to finding all elements  $[N(x) \ -D(x)]$  in  $\mathfrak{M}(\mathbf{r})$  with  $(0, k-1)$ -weighted  $q$ -degree less than or equal to  $t + k - 1$  and  $\text{qdeg}(N(x)) \leq \text{qdeg}(D(x)) + k - 1$ , such that  $N(x)$  is symbolically divisible on the right by  $D(x)$ . It follows that, to find all closest codewords to a given  $\mathbf{r} \in \mathbb{F}_{q^m}^n$ , we need to find all elements  $[N(x) \ -D(x)] \in \mathfrak{M}(\mathbf{r})$  of minimal  $(0, k-1)$ -weighted  $q$ -degree such that  $\text{qdeg}(N(x)) \leq \text{qdeg}(D(x)) + k - 1$  and  $N(x)$  is symbolically divisible on the right by  $D(x)$ . This minimality requirement leads us to construct a minimal Gröbner basis for  $\mathfrak{M}(\mathbf{r})$ . Note that this is a generalization of the interpolation-based decoding method from [10]. The difference is that our method can also decode beyond the unique decoding radius.

In contrast to our previous paper [7] the algorithm below is iterative in the sense that it adds an extra pair of interpolation points  $(g_i, r_i)$  at the  $i$ -th step of the algorithm. More specifically, in the  $i$ -th step a minimal Gröbner basis is constructed for the interpolation module corresponding to  $(g_1, \dots, g_i), (r_1, \dots, r_i)$ .

**Theorem 13.** *For  $i = 1, \dots, n$  denote by  $\mathfrak{M}_i$  the interpolation module for  $(g_1, \dots, g_i)$  and  $(r_1, \dots, r_i)$ . Let*

$$\begin{bmatrix} P(x) & -K(x) \\ N(x) & -D(x) \end{bmatrix}$$

be a basis for  $\mathfrak{M}_{i-1}$  and

$$\Delta_i := N(g_i) - D(r_i) \quad , \quad \Gamma_i := P(g_i) - K(r_i).$$

If  $\Gamma_i \neq 0$ , then the row vectors of

$$\begin{bmatrix} x^q - \Gamma_i^{q-1}x & 0 \\ \Delta_i x & -\Gamma_i x \end{bmatrix} \circ \begin{bmatrix} P(x) & -K(x) \\ N(x) & -D(x) \end{bmatrix}$$

form a basis of  $\mathfrak{M}_i$ . If  $\Delta_i \neq 0$ , then the row vectors of

$$\begin{bmatrix} \Delta_i x & -\Gamma_i x \\ 0 & x^q - \Delta_i^{q-1}x \end{bmatrix} \circ \begin{bmatrix} P(x) & -K(x) \\ N(x) & -D(x) \end{bmatrix}$$

form a basis of  $\mathfrak{M}_i$ .

*Proof:* We first consider the first case and show that both row vectors are in  $\mathfrak{M}_i$ . From the assumptions it follows that  $P(g_j) = K(r_j)$  and that  $N(g_j) = D(r_j)$  for  $1 \leq j < i$ . Moreover, the two entries of the first row are given by

$$(x^q - \Gamma_i^{q-1}x) \circ P(x) = P(x)^q - \Gamma_i^{q-1}P(x),$$

$$(x^q - \Gamma_i^{q-1}x) \circ K(x) = K(x)^q - \Gamma_i^{q-1}K(x),$$

thus  $P(g_j)^q - \Gamma_i^{q-1}P(g_j) - K(r_j)^q + \Gamma_i^{q-1}K(r_j) = 0$  for  $1 \leq j \leq i$ . For the second row we get

$$\Delta_i P(g_j) - \Gamma_i N(g_j) - \Delta_i K(r_j) + \Gamma_i D(r_j) =$$

$$\Delta_i (P(g_j) - K(r_j)) - \Gamma_i (N(g_j) - D(r_j)) = \Delta_i \Gamma_i - \Gamma_i \Delta_i = 0$$

for  $1 \leq j \leq i$ . Thus, the two row vectors are elements of  $\mathfrak{M}_i$ .

It remains to show that the two row vectors span the whole interpolation module (and not just a submodule of it). For this, we note that there exist  $\bar{a}(x), \bar{b}(x) \in \mathcal{L}_q(x, q^m)$  such that  $\bar{a}(x) \circ [P(x) - K(x)] + \bar{b}(x) \circ [N(x) - D(x)] = [\Pi_{i-1}(x) \ 0]$ . Let  $\bar{b}(x) = -(x^q - \Pi_{i-1}(g_i)^{q-1}x) \circ \bar{b}(\frac{1}{\Gamma_i}x) \in \mathcal{L}_q(x, q^m)$  and let  $a(x) \in \mathcal{L}_q(x, q^m)$  such that  $a(x) \circ (x^q - \Gamma_i^{q-1}x) = (x^q - \Pi_{i-1}(g_i)^{q-1}x) \circ (\bar{b}(\frac{1}{\Gamma_i}x) + \bar{a}(x))$ . Note that  $a(x)$  is well-defined by Lemma 9 since  $\Gamma_i$  is a root of the right side of the previous equation. Denote the first and second row of the new basis by  $b_1$  and  $b_2$ , respectively. Then  $a(x) \circ b_1 + b(x) \circ b_2 = [\Pi_i(x) \ 0]$ , i.e.  $[\Pi_i(x) \ 0]$  is in the module spanned by the new basis. Analogously we can construct  $a(x), b(x) \in \mathcal{L}_q(x, q^m)$  such that  $a(x) \circ b_1 + b(x) \circ b_2 = [\Lambda_i(x) \ -x]$ . Hence, we have shown that the new basis spans the whole interpolation module.

For the second case note that

$$\begin{aligned} & \text{rs} \left( \begin{bmatrix} \Delta_i x & -\Gamma_i x \\ 0 & x^q - \Delta_i^{q-1} x \end{bmatrix} \circ \begin{bmatrix} P(x) & -K(x) \\ N(x) & -D(x) \end{bmatrix} \right) \\ &= \text{rs} \left( \begin{bmatrix} x^q - \Delta_i^{q-1} x & 0 \\ \Gamma_i x & -\Delta_i x \end{bmatrix} \circ \begin{bmatrix} N(x) & -D(x) \\ P(x) & -K(x) \end{bmatrix} \right), \end{aligned}$$

which corresponds to the first case after exchanging  $P(x)$  with  $N(x)$  and  $K(x)$  with  $D(x)$  (and vice versa).  $\blacksquare$

**Remark 14.** In the notation of Proposition 3, applying the previous theorem to  $P(x) = \Pi_{i-1}(x), K(x) = 0, N(x) = \Lambda_{i-1}(x)$  and  $D(x) = -x$ , leads to a computation that is identical up to a constant to the one in Proposition 3 in which the  $q$ -annihilator polynomial and the  $q$ -Lagrange polynomial are iteratively constructed.

#### IV. THE ALGORITHM

Using Theorem 13 as our main ingredient, we now set out to design an iterative algorithm that computes a minimal Gröbner basis for  $\mathfrak{M}_i$  at each step  $i$ . We note that the calculation of the matrices  $B_i$  in our algorithm coincides with the calculations in the interpolations algorithms of [5], [25]. The complete decoding algorithm, stated in Algorithm 1, first computes a minimal Gröbner basis for  $\mathfrak{M}_n$  and then uses a parametrization to find all closest codewords to the received word.

**Theorem 15.** *Algorithm 1 yields a list of all message polynomials such that the corresponding codeword is closest to the received word.*

*Proof:* Denote by  $M_1$  the matrix we multiply by on the left in the first IF statement and by  $M_2$  the one in the ELSE statement of the algorithm. We know from Theorem 13 that at each step  $B_i$  is a basis for the interpolation module  $\mathfrak{M}_i$  (swap the roles of  $\Gamma_i$  and  $\Delta_i$  where needed). We now show that it is a minimal Gröbner basis with respect to the  $(0, k-1)$ -weighted TOP monomial order via induction on  $i$ . Assume that at step  $i$  the first row has leading position 1 and the second row has leading position 2, i.e.  $\text{qdeg}(P_i(x)) > \text{qdeg}(K_i(x)) + k - 1$  and  $\text{qdeg}(N_i(x)) \leq \text{qdeg}(D_i(x)) + k - 1$ . Furthermore assume that  $\text{qdeg}(P_i(x)) \geq \text{qdeg}(N_i(x))$ . If  $\text{qdeg}(P_i(x)) \leq \text{qdeg}(D_i(x)) + k - 1$  we multiply on the left by  $M_1$ . Hence,

$$\text{qdeg}(P_{i+1}(x)) = \text{qdeg}(P_i(x)) + 1,$$

$$\text{qdeg}(K_{i+1}(x)) = \text{qdeg}(K_i(x)) + 1,$$

---

**Algorithm 1** Iterative minimal list decoding of Gabidulin codes.

---

**Require:** Linearly independent  $g_1, \dots, g_n \in \mathbb{F}_{q^m}$ , received  $r_1, \dots, r_n \in \mathbb{F}_{q^m}$ .

Initialize **list** := [],  $j := 0$ ,  $B_0 := \begin{bmatrix} x & 0 \\ 0 & x \end{bmatrix}$ .

We denote  $B_i := \begin{bmatrix} P_i(x) & -K_i(x) \\ N_i(x) & -D_i(x) \end{bmatrix}$ .

**for**  $i$  from 1 to  $n$  **do**

$\Delta_i := N_{i-1}(g_i) - D_{i-1}(r_i)$  ,  $\Gamma_i := P_{i-1}(g_i) - K_{i-1}(r_i)$ .

**if**  $[\text{qdeg}(P_{i-1}(x)) \leq \text{qdeg}(D_{i-1}(x)) + k - 1$  **and**  $\Gamma_i \neq 0]$  **or**  $\Delta_i = 0$  **then**

$$B_i := \begin{bmatrix} x^q - \Gamma_i^{q-1} x & 0 \\ \Delta_i x & -\Gamma_i x \end{bmatrix} \circ B_{i-1}$$

**else**

$$B_i := \begin{bmatrix} \Delta_i x & -\Gamma_i x \\ 0 & x^q - \Delta_i^{q-1} x \end{bmatrix} \circ B_{i-1}$$

**end if**

**end for**

Set  $b_1(x) :=$  first row of  $B_n$ ,  $b_2(x) :=$  second row of  $B_n$ ,  $\ell_1 := \text{qdeg}(b_1)$ ,  $\ell_2 := \text{qdeg}(b_2)$ .

**while** **list** = [] **do**

**for all**  $a(x) \in \mathcal{L}_q(x, q^m)$ ,  $\text{qdeg}(a(x)) \leq \ell_2 - \ell_1 + j$  **do**

**for all** monic  $c(x) \in \mathcal{L}_q(x, q^m)$ ,  $\text{qdeg}(b(x)) = j$  **do**

$f(x) := a(x) \circ b_1(x) + c(x) \circ b_2(x)$

**if**  $f^{(1)}(x)$  is symb. (right) divisible by  $f^{(2)}(x)$  **then**

add the respective symb. quotient to **list**

**end if**

**end for**

**end for**

$j := j + 1$

**end while**

**return list**

---

where the latter is less than  $\text{qdeg}(P_i(x)) - k + 2 = \text{qdeg}(P_{i+1}(x)) - k + 1$  by the assumption. Thus, the leading position of the first row of  $B_{i+1}$  is 1. Moreover,

$$\begin{aligned} \text{qdeg}(N_{i+1}(x)) &\leq \max\{\text{qdeg}(P_i(x)), \text{qdeg}(N_i(x))\} \\ &= \text{qdeg}(P_i(x)) \leq \text{qdeg}(D_i(x)) + k - 1 \end{aligned}$$

and, since the assumptions imply that  $\text{qdeg}(K_i(x)) < \text{qdeg}(D_i(x))$ ,

$$\text{qdeg}(D_{i+1}(x)) = \max\{\text{qdeg}(K_i(x)), \text{qdeg}(D_i(x))\} = \text{qdeg}(D_i(x)).$$

Thus the leading position of the second row is 2. Moreover,  $\text{qdeg}(P_{i+1}(x)) \geq \text{qdeg}(N_{i+1}(x))$ . Since the assumptions are true for  $B_0$  the statement follows via induction.

Analogously one can prove that multiplication with  $M_2$  yields a basis of  $\mathfrak{M}_i$  with different leading positions in the two rows. Thus, after  $n$  steps,  $B_n$  is a minimal Gröbner basis for the interpolation module  $\mathfrak{M}(r)$ . Consequently,  $B_n$  has the so-called Predictable Leading Monomial Property, see [6] and [1]. As a result of this property, the parametrization used for  $a(x)$  and  $c(x)$  in the second part of the algorithm will then yield all closest codewords. For the sake of brevity we omit the details.  $\blacksquare$

**Remark 16.** It can be verified that, due to the linear independence of  $g_1, \dots, g_k$ , the first  $k$  steps of the algorithm coincide up to a constant with the computation in Proposition 3. In other words, up to a constant, at step  $k$  the algorithm has computed

the  $q$ -annihilator polynomial and the  $q$ -Lagrange polynomial corresponding to the data so far.

**Example 17.** Consider the Gabidulin code in  $\mathbb{F}_{2^3} \cong \mathbb{F}_2[\alpha]$  (with  $\alpha^3 = \alpha + 1$ ) with generator matrix

$$G = \begin{pmatrix} 1 & \alpha & \alpha^2 \\ 1 & \alpha^2 & \alpha^4 \end{pmatrix}$$

(i.e.  $g_1 = 1, g_2 = \alpha, g_3 = \alpha^2$  and  $k = 2$ ) and the received word  $\mathbf{r} = (\alpha^3 \ 0 \ \alpha)$ . We iteratively compute

$$B_1 = \begin{bmatrix} x^2 + x & 0 \\ (\alpha + 1)x & x \end{bmatrix},$$

$$B_2 = \begin{bmatrix} x^4 + (\alpha^2 + \alpha + 1)x^2 + (\alpha^2 + \alpha)x & 0 \\ (\alpha^2 + \alpha)x^2 + (\alpha^2 + \alpha + 1)x & (\alpha^2 + \alpha)x \end{bmatrix},$$

$$B_3 = \begin{bmatrix} \alpha^2 x^4 + \alpha^5 x & x \\ \alpha x^4 + \alpha^4 x^2 + x & \alpha x^2 + \alpha^6 x \end{bmatrix}.$$

$B_3$  is a minimal  $(0, 1)$ -weighted Gröbner basis of the interpolation module. We get  $\ell_1 = 2$  and  $\ell_2 = 2$ , i.e. we want to use all  $a(x) \in \mathcal{L}_2(x, 2^3)$  with 2-degree less than or equal to 0 and all monic  $c(x) \in \mathcal{L}_2(x, 2^3)$  with 2-degree equal to 0. Thus,  $a(x) = a_0 x$  for  $a_0 \in \mathbb{F}_{2^3}$  and  $c(x) = x$ . We get divisibility for  $a_0 \in \mathbb{F}_{2^3} \setminus \{\alpha^6\}$ . The corresponding message polynomials and codewords are

$$m_1(x) = x^2 + \alpha x \quad , \quad c_1 = (\alpha^3 \ 1 \ \alpha^3),$$

$$m_2(x) = \alpha^5 x^2 + \alpha^2 x \quad , \quad c_1 = (\alpha^3 \ \alpha \ \alpha),$$

$$m_3(x) = \alpha^3 x^2 + \alpha^4 x \quad , \quad c_1 = (\alpha^2 + 1 \ 0 \ \alpha^2),$$

$$m_4(x) = \alpha^4 x^2 \quad , \quad c_3 = (\alpha^2 + \alpha \ \alpha^2 + 1 \ \alpha),$$

$$m_5(x) = \alpha^6 x^2 + \alpha^6 x \quad , \quad c_1 = (0 \ \alpha^3 \ 1),$$

$$m_6(x) = \alpha^2 x^2 + \alpha^3 x \quad , \quad c_2 = (\alpha^5 \ 0 \ \alpha),$$

$$m_7(x) = \alpha x^2 + x \quad , \quad c_2 = (\alpha^3 \ 1 \ \alpha^3).$$

All these codewords are rank distance 1 away from  $\mathbf{r}$  (note that some of them are Hamming distance 1, 2 or even 3 away from  $\mathbf{r}$ ).

## V. CONCLUSION

In this paper we used a parametrization approach to the decoding of Gabidulin codes with respect to the rank metric. We presented a iterative algorithm with simple update steps that is similar to the ones found in the literature. Our main result is that we use this algorithm to compute a list of message polynomials that correspond to *all* codewords that are closest to a given received word. In our algorithm we construct, via a simple update matrix, a minimal Gröbner basis at each step. This setup allows for straightforward conclusions on minimality and parametrization due to the Predictable Leading Monomial Property, as in [1] and [6].

## REFERENCES

- [1] M. Ali and M. Kuijper. A parametric approach to list decoding of Reed-Solomon codes using interpolation. *IEEE Trans. Inform. Theory*, 57(10):6718–6728, 2011.
- [2] P. Delsarte. Bilinear forms over a finite field, with applications to coding theory. *Journal of Combinatorial Theory, Series A*, 25(3):226–241, 1978.
- [3] E. M. Gabidulin. Theory of codes with maximum rank distance. *Problemy Peredachi Informatsii*, 21(1):3–16, 1985.
- [4] E. M. Gabidulin. A fast matrix decoding algorithm for rank-error-correcting codes. In *Algebraic coding (Paris, 1991)*, volume 573 of *Lecture Notes in Comput. Sci.*, pages 126–133. Springer, Berlin, 1992.
- [5] R. Kötter and F. R. Kschischang. Coding for errors and erasures in random network coding. *IEEE Transactions on Information Theory*, 54(8):3579–3591, 2008.
- [6] M. Kuijper and K. Schindelar. Minimal Gröbner bases and the predictable leading monomial property. *Linear Algebra and its Applications*, 434(1):104–116, 2011.
- [7] M. Kuijper and A.-L. Trautmann. List decoding Gabidulin codes via interpolation and the Euclidean algorithm. In *arXiv:1404.5716 [cs.IT]*, 2014.
- [8] R. Lidl and H. Niederreiter. *Finite Fields*. Cambridge University Press, Cambridge, London. Second edition.
- [9] P. Loidreau. Decoding rank errors beyond the error correcting capability. In *International Workshop on Algebraic and Combinatorial Coding Theory (ACCT)*, pages 186–190, Sept. 2006.
- [10] P. Loidreau. A Welch-Berlekamp like algorithm for decoding Gabidulin codes. In *Coding and cryptography*, volume 3969 of *Lecture Notes in Comput. Sci.*, pages 36–45. Springer, Berlin, 2006.
- [11] P. Lusina, E. Gabidulin, and M. Bossert. Maximum rank distance codes as space-time codes. *Information Theory, IEEE Transactions on*, 49(10):2757–2760, Oct 2003.
- [12] H. Mahdaviyar and A. Vardy. List-decoding of subspace codes and rank-metric codes up to singleton bound. In *Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on*, pages 1488–1492, 2012.
- [13] Oystein Ore. On a Special Class of Polynomials. *Transactions of the American Mathematical Society*, 35:559–584, 1933.
- [14] G. Richter and S. Plass. Fast decoding of rank-codes with rank errors and column erasures. In *Information Theory, 2004. ISIT 2004. Proceedings. International Symposium on*, pages 398–398, 2004.
- [15] R. M. Roth. Maximum-rank array codes and their application to crisscross error correction. *IEEE Transactions on Information Theory*, 37(2):328–336, mar 1991.
- [16] 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, June 2010.
- [17] V. Sidorenko, L. Jiang, and M. Bossert. Skew-feedback shift-register synthesis and decoding interleaved Gabidulin codes. *IEEE Trans. Inform. Theory*, 57(2):621–632, 2011.
- [18] N. Silberstein, A. S. Rawat, and S. Vishwanath. Adversarial error resilience in distributed storage using MRD codes and MDS array codes. *arXiv:1202.0800v1 [cs.IT]*, 2012.
- [19] D. Silva and F. R. Kschischang. On metrics for error correction in network coding. *IEEE Transactions on Information Theory*, 55(12):5479–5490, dec. 2009.
- [20] D. Silva, F. R. Kschischang, and R. Kötter. A rank-metric approach to error control in random network coding. *IEEE Transactions on Information Theory*, 54(9):3951–3967, 2008.
- [21] D. Silva and F.R. Kschischang. Fast encoding and decoding of gabidulin codes. In *Information Theory, 2009. ISIT 2009. IEEE International Symposium on*, pages 2858–2862, June 2009.
- [22] A.-L. Trautmann, N. Silberstein, and J. Rosenthal. List decoding of lifted Gabidulin codes via the Plücker embedding. In *Preproceedings of the International Workshop on Coding and Cryptography (WCC) 2013*, pages 539–549, Bergen, Norway, 2013.
- [23] A. Wachter-Zeh. *Decoding of Block and Convolutional Codes in Rank Metric*. PhD thesis, Ulm University, Germany, 2013.
- [24] A. Wachter-Zeh and A. Zeh. Interpolation-based decoding of interleaved Gabidulin codes. In *Preproceedings of the International Workshop on Coding and Cryptography (WCC) 2013*, pages 527–537, Bergen, Norway, 2013.
- [25] H. Xie, Z. Yan, and B.W. Suter. General linearized polynomial interpolation and its applications. In *Network Coding (NetCod), 2011 International Symposium on*, pages 1–4, July 2011.