Algebraic Soft Decoding of Hermitian Codes with Re-encoding Transform

Jiwei Liang[†], Li Chen[‡§],
[†] School of System Science and Engineering, Sun Yat-sen University, Guangzhou, China
[‡] School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou, China
[§] Guangdong Provincial Key Laboratory of Information Security Technology, Guangzhou, China
Email: liangjw59@mail2.sysu.edu.cn, chenli55@mail.sysu.edu.cn

2025 IEEE International Symposium on Information Theory (ISIT) | 979-8-3315-4399-0/25/$31.00 ©2025 IEEE | DOI: 10.1109/ISIT63088.2025.11195257

***Abstract*—This paper proposes a new interpolation-based algebraic soft decoding (ASD) for Hermitian codes. The interpolation is realized through basis reduction (BR) that is facilitated by the re-encoding transform (ReT). The ReT is formulated by defining Lagrange interpolation polynomials over the Hermitian function fields and choosing proper re-encoding points. It transforms the interpolation points, which can lead to a reduced BR interpolation complexity. With a designed decoding output list size (OLS), the interpolation module basis can be formulated. The ReT results in polynomials of the module basis having a common factor. It can be extracted from the polynomials, resulting in a simpler basis reduction. An enhanced ReT is further proposed. It enables the basis polynomials having a common factor with a greater degree, yielding a more significant complexity reduction. Numerical results show that both of the two proposed ReT can facilitate the ASD for the advanced decoding of Hermitian codes.**

***Index Terms*—Algebraic-geometric codes, algebraic soft decoding, basis reduction, Hermitian codes, re-encoding transform**

I. Introduction

Algebraic-geometric (AG) codes are linear block codes constructed based on algebraic curves [1]. AG codes compromise a large family, including the widely used Reed-Solomon (RS) codes, elliptic codes, Hermitian codes, etc. Among them, RS codes can be considered as a special class of AG codes since they are constructed from a straight line. But the length of an RS code cannot exceed the size of finite field in which it is defined, limiting its error-correction capability. Compared with RS codes, general AG codes have larger codeword lengths, leading to their stronger error-correction capabilities.

Similar to RS codes, AG codes can be decoded by the syndrome-based decoding algorithms [2, 3]. They cannot correct errors beyond half of the code's minimum Hamming distance. The Guruswami-Sudan (GS) algorithm can correct errors beyond this limit by formulating the decoding as a curve-fitting problem [4]. The GS decoding consists of interpolation and root-finding, where the former dominates the complexity. The interpolation can be realized by Kötter's interpolation [5] which constructs the interpolation polynomial in an iterative manner. It can also be realized by the basis reduction (BR) approach through treating the decoding as computing a desired Gröbner basis that contains the interpolation polynomial. The Gröbner basis can be obtained by first constructing the interpolation module basis, and then reducing it. The BR interpolation-based GS decoding was first proposed for RS codes [6] [7]. It can be generalized to decode other AG codes, such as Hermitian codes [8] and elliptic codes [9] [10]. Although GS decoding can yield a better error-correction capability, it still exhibits a high complexity of $O(l^4 n(n-k))$[1] in decoding general AG codes [11], where l is the designed decoding output list size (OLS), n and k are the length and dimension of the code, respectively. Addressing this problem, re-encoding transform (ReT) was proposed to facilitate the decoding of RS codes [12]. Recently, Wan *et al.* generalized it into the decoding of general AG codes [11], reducing the complexity to $O(l^4(n-k)^2)$.

The error-correction capability of GS decoding can be further improved by utilizing soft information. By formulating test-vectors and exploiting their similarities, the low-complexity Chase (LCC) decoding was applied to decode RS codes [13] and elliptic codes [14]. Recently, Liang *et al.* proposed the LCC decoding of Hermitian codes, which is facilitated by the ReT-assisted BR interpolation and improved root-finding [15]. The algebraic soft decoding (ASD) is another soft decoding approach for enhancing the decoding performance. By transforming soft information into the interpolation multiplicities, the ASD of RS codes was proposed by Kötter *et al.* [16]. The ASD of elliptic codes was proposed by Wan *et al.* [17], where the ReT was employed. Chen *et al.* [18] and Lee *et al.* [19] proposed the ASD of Hermitian codes through Kötter's interpolation and the BR interpolation, respectively. However, they both exhibit an interpolation complexity of $O(l^5 n^{7/3})$. Moreover, the ReT has not yet been studied for the ASD of Hermitian codes.

This paper proposes the ASD of Hermitian codes, for which the interpolation is realized through the ReT-assisted BR interpolation. By defining Lagrange interpolation polynomials and choosing proper re-encoding points, the ReT transforms the interpolation points into having zero z-coordinates and makes the module basis polynomials having a common factor, leading to a simpler BR interpolation. An enhanced ReT is further proposed to enable the module basis polynomials having a greater common factor, which leads to a more significant complexity reduction. Numerical results show that both of the two proposed ReT facilitated ASD exhibit a lower complexity than that without applying the ReT for Hermitian codes.

[1] In this paper, the "big-O" notations ignore the logarithmic factors.

II. Background Knowledge

This section presents the background knowledge of Hermitian codes, the GS decoding and the ASD.

A. Hermitian Codes

Let $\mathbb{F}_q = \{\sigma_0, \sigma_1, \ldots, \sigma_{q-1}\}$ denote the finite field of size q. In this paper, q is a square as required by the definition of Hermitian codes. Let $\mathbb{F}_q[X, Y]$ further denote the bivariate polynomial ring defined over $\mathbb{F}_q$. An affine Hermitian curve defined over $\mathbb{F}_q$ can be written as [20]

$$H_w = Y^w + Y - X^{w+1}, \tag{1}$$

where $w = \sqrt{q}$ and the curve has a genus $g = \frac{w(w-1)}{2}$. There are w^3 affine points $P_j = (x_j, y_j)$ that satisfy $H_w(x_j, y_j) = 0$, and a point at infinity P_∞. They form the set of $\mathbb{F}_q$-rational points at H_w. Let $\mathcal{P} = \{P_j = (x_j, y_j) \mid H_w(x_j, y_j) = 0\}$ denote the set of affine points and $|\mathcal{P}| = w^3$. The coordinate ring of H_w is

$$\mathcal{R} = \mathbb{F}_q[X, Y] / < Y^w + Y - X^{w+1} > . \tag{2}$$

Let x and y denote the residue classes of X and Y, respectively. The pole basis $\mathcal{L}_w$ comprises a set of bivariate monomials $\phi_a(x, y) = x^{i_x} y^{i_y}$. For a nonzero polynomial $S \in \mathcal{R}$, its order at a rational point P is denoted as $\nu_P(S)$. Let $\nu_{P_\infty}(\phi_a^{-1})$ denote the pole order at P_∞ and $\nu_{P_\infty}(\phi_a^{-1}) = \nu_{P_\infty}((x^{i_x} y^{i_y})^{-1}) = w \cdot i_x + (w+1) \cdot i_y$, $0 \le i_x \le w$ and $i_y \ge 0$. Monomials of the pole basis exhibit an increasing pole order as $\mathcal{L}_w = \{\phi_a | \nu_{P_\infty}(\phi_a^{-1}) < \nu_{P_\infty}(\phi_{a+1}^{-1})\}$. Let $\mu = k + g - 1$ and $\mathcal{L}(\mu P_\infty)$ is the Riemann-Roch space defined by μ and P_∞. For an (n, k) Hermitian code, given a message polynomial

$$f(x, y) = f_0\phi_0 + f_1\phi_1 + \cdots + f_{k-1}\phi_{k-1} \in \mathcal{L}(\mu P_\infty), \tag{3}$$

where $f_0, f_1, \ldots, f_{k-1} \in \mathbb{F}_q$ are the message symbols, the codeword $\underline{c} \in \mathbb{F}_q^n$ is generated by

$$\underline{c} = (c_0, c_1, \ldots, c_{n-1}) = (f(P_0), f(P_1), \ldots, f(P_{n-1})), \tag{4}$$

where $\{P_0, P_1, \ldots, P_{n-1}\} \subseteq \mathcal{P}$. Let $[a : b] = \{a, a+1, \ldots, b\}$, where $a, b \in \mathbb{Z}$ and $a \le b$. The index set of the codeword is $[0 : n-1]$. The Riemann-Roch theorem [21] defines the relationship between μ and $\nu_{P_\infty}(\phi_{k-1}^{-1})$ as

$$\nu_{P_\infty}(\phi_{k-1}^{-1}) \le \mu. \tag{5}$$

B. The GS Decoding

For GS decoding of an (n, k) Hermitian code, the following definition is needed.

Definition 1: Let $\mathcal{R}[z]$ denote the polynomial ring defined over $\mathcal{R}$. Monomials $\phi_a z^b \in \mathcal{R}[z]$ are ordered according to their $(1, w_z)$-weighted degrees as

$$\deg_{1,w_z} \phi_a z^b = \nu_{P_\infty}(\phi_a^{-1}) + w_z b, \tag{6}$$

where $w_z = \nu_{P_\infty}(\phi_{k-1}^{-1})$. The $(1, w_z)$-reverse lexicographic (revlex) order can be established as follows. Given two monomials $\phi_{a_1} z^{b_1}$ and $\phi_{a_2} z^{b_2}$, $\mathrm{ord}(\phi_{a_1} z^{b_1}) < \mathrm{ord}(\phi_{a_2} z^{b_2})$, if $\deg_{1,w_z} \phi_{a_1} z^{b_1} < \deg_{1,w_z} \phi_{a_2} z^{b_2}$, or $\deg_{1,w_z} \phi_{a_1} z^{b_1} = \deg_{1,w_z} \phi_{a_2} z^{b_2}$ and $b_1 < b_2$. Given a polynomial $Q(x, y, z) = \sum_{a,b \in \mathbb{N}} Q_{ab} \phi_a(x, y) z^b$, the $(1, w_z)$-weighted degree of Q is $\deg_{1,w_z} Q = \max\{\deg_{1,w_z} \phi_a z^b \mid Q_{ab} \ne 0\}$ and its leading order is $\mathrm{lod}(Q) = \max\{\mathrm{ord}(\phi_a z^b) \mid Q_{ab} \ne 0\}$.

In decoding an (n, k) Hermitian code, polynomials are organized under the $(1, w_z)$-revlex order. Given two polynomials Q_1 and Q_2, we claim $Q_1 < Q_2$, if $\mathrm{lod}(Q_1) < \mathrm{lod}(Q_2)$.

***Theorem 1** [8]:* Given the polynomial $\mathcal{Q}$ which has a zero of multiplicity at least m over the n interpolation points, if $m(n - |\{j \mid f(P_j) \ne w_j, \forall j \in [0 : n-1]\}|) > \deg_{1,w_z} \mathcal{Q}$, $\mathcal{Q}(x, y, f) = 0$, or equivalently $(z - f) | \mathcal{Q}$.

Interpolation constructs the polynomial $\mathcal{Q}$. The z-roots of $\mathcal{Q}$ are the estimated message polynomials. In the following, the ASD will be further introduced.

C. The ASD and Its BR Interpolation

1) *Reliability Transform*: Assume that a Hermitian codeword $\underline{c}$ is transmitted and $\underline{r} = (r_0, r_1, \ldots, r_{n-1})$ is the channel output. The reliability matrix $\mathbf{\Pi} \in \mathbb{R}^{q \times n}$ with entries $\pi_{ij} = \Pr(r_j \mid c_j = \sigma_i)$ can be obtained. Parametrized by l, $\mathbf{\Pi}$ will be transformed into a multiplicity matrix $\mathbf{M}$ of the same size, where its entries m_{ij} is nonnegative integer and is considered to be associated with the interpolation points (P_j, σ_i). Let i_j denote the index of σ_i that satisfies $\sigma_i = c_j$. The codeword score of $\mathbf{M}$ is defined as

$$s_{\mathbf{M}}(\underline{c}) = \sum_{j=0}^{n-1} m_{i_j j}. \tag{7}$$

2) *Interpolation and Root-Finding*: Let $\mathrm{mult}_{(P_j, \sigma_i)}(Q)$ denote the multiplicity of polynomial Q at point (P_j, σ_i). Given $\mathbf{M}$, the interpolation module $\mathcal{I}_{\mathbf{M},l}$ can be defined as

$$\mathcal{I}_{\mathbf{M},l} = \{Q \in \mathcal{R}[z] \mid \mathrm{mult}_{(P_j, \sigma_i)}(Q) \ge m_{ij} \text{ and } \deg_z Q \le l \text{ for } 0 \le i \le q-1, 0 \le j \le n-1\}.$$

With $\mathbf{M}$, interpolation constructs the minimum polynomial $\mathcal{Q}$ in $\mathcal{I}_{\mathbf{M},l}$ w.r.t. the $(1, w_z)$-revlex order. The following theorem reveals the sufficient condition of a successful ASD .

***Theorem 2** [19]:* Given an (n, k) Hermitian code and the interpolation polynomial $\mathcal{Q} \in \mathcal{I}_{\mathbf{M},l}$, if $s_{\mathbf{M}}(\underline{c}) > \deg_{1,w_z} \mathcal{Q}$, $\mathcal{Q}(x, y, f) = 0$, or equivalently $(z - f) | \mathcal{Q}$.

The interpolation polynomial $\mathcal{Q}$ can be determined by BR interpolation. It first constructs a basis of the interpolation module $\mathcal{I}_{\mathbf{M},l}$, which will then be reduced into a Gröbner basis [8]. Its minimum candidate is $\mathcal{Q}$. For the construction of the module basis, a series of multiplicity matrices need to be generated based on $\mathbf{M}$. Let $\eta = \max_j\{\sum_{i=0}^{q-1} m_{ij}\}$. These intermediate multiplicity matrices are denoted as $\mathbf{M}^{(u)}$, where $u = 0, 1, \ldots, \eta$. Let $\mathbf{M}^{(0)} = \mathbf{M}$. The entries of $\mathbf{M}^{(1)}$ to $\mathbf{M}^{(\eta)}$ are determined by

$$m_{ij}^{(u+1)} = \begin{cases} m_{ij}^{(u)} - 1, & \text{if } i = i_j^{(u)} \text{ and } m_{ij}^{(u)} \ne 0, \\ m_{ij}^{(u)}, & \text{if } i \ne i_j^{(u)} \text{ or } m_{ij}^{(u)} = 0, \end{cases} \tag{8}$$

where $i_j^{(u)} = \arg\max_i\{m_{ij}^{(u)}\}$.

Definition 2: Given an index set $\mathcal{J} \subseteq [0 : n-1]$, let $\mathbb{A}(\mathcal{J}) = \{x_j \mid j \in \mathcal{J}\}$. For $\sigma \in \mathbb{A}(\mathcal{J})$, let $\mathbb{B}_\sigma(\mathcal{J}) = \{y_j \mid (\sigma, y_j) \in \mathcal{P},\ j \in \mathcal{J}\}$, $\mathbb{C}(j) = \{j' \mid x_{j'} = x_j\}$ and $\mathbb{S}(\mathcal{J}) = \{j \mid |\mathbb{B}_{x_j}(\mathcal{J})| = w\}$. The affine points defined by $\mathcal{J}$ form a maximum semi-grid if $|\mathbb{B}_{x_j}(\mathcal{J})| = w, \forall x_j \in \mathbb{A}(\mathcal{J})$.

Given an affine point index set $\mathcal{J} \subseteq [0 : n-1]$, the Lagrange interpolation polynomial defined by $\mathcal{J}$ is written as

$$L_{\mathcal{J},j}(x,y) = \prod_{\alpha \in \mathbb{A}(\mathcal{J}) \setminus \{x_j\}} \frac{x-\alpha}{x_j-\alpha} \prod_{\beta \in \mathbb{B}_{x_j}(\mathcal{J}) \setminus \{y_j\}} \frac{y-\beta}{y_j-\beta}. \quad (9)$$

Note that $L_{\mathcal{J},j}(P_j) = 1$ and $L_{\mathcal{J},j}(P_{j'}) = 0$, if $j \neq j'$, where $j, j' \in \mathcal{J}$. Let $\omega_j^{(u)} = \sigma_{i_j^{(u)}}$ and $\underline{\omega}^{(u)} = (\omega_0^{(u)}, \omega_1^{(u)}, \ldots, \omega_{n-1}^{(u)})$. We define the following polynomial

$$K_{\underline{\omega}^{(u)}}(x,y) = \sum_{j \in [0:n-1]} \omega_j^{(u)} L_{[0:n-1],j}(x,y). \quad (10)$$

Note that $K_{\underline{\omega}^{(u)}}(P_j) = \omega_j^{(u)}$. Let $\mathfrak{m}_j^{(u)} = \max_i\{m_{ij}^{(u)}\}$. Let $\mathcal{E}_{\mathbf{M}^{(u)}}$ be an ideal in R and $\mathcal{E}_{\mathbf{M}^{(u)}} = \{S \in \mathcal{R} \mid \nu_{P_j}(S) \geq \mathfrak{m}_j^{(u)}\}$. The elements of $\mathcal{I}_{\mathbf{M},l}$ can be obtained using the elements of $\mathcal{E}_{\mathbf{M}^{(u)}}$ and the polynomial $K_{\underline{\omega}^{(u)}}$ [19]. In order to compute $\mathcal{E}_{\mathbf{M}^{(u)}}$, indices of the w^3 affine points need to be reassigned by grouping them into w^2 classes of different x-coordinates. Thus, P_j is reassigned to $P_{a,b}$ with $a = \lfloor j/w \rfloor$ and $b = j \bmod w$. If the point P_j is reassigned to $P_{a,b}$, $v_{a,b}^{(u)} = \mathfrak{m}_j^{(u)}$. It is also assumed that for each $0 \leq a \leq w^2-1$, index b has been arranged such that $v_{a,b}^{(u)}$ are written in a decreasing order as

$$v_{a,0}^{(u)} \geq v_{a,1}^{(u)} \geq \ldots \geq v_{a,w-1}^{(u)}. \quad (11)$$

Given $B_{b,c}^{(u)} \in \mathbb{F}_q[x]$ and it satisfies $\nu_{P_{a,b}}(y - B_{b,c}^{(u)}) \geq v_{a,b}^{(u)} - v_{a,c}^{(u)}$, we further define the following polynomial

$$T_{u,c}(x,y) = \prod_{a=0}^{q-1} (x-\sigma_a)^{v_{a,c}^{(u)}} \prod_{0 \leq b \leq c-1} (y - B_{b,c}^{(u)}), \quad (12)$$

where $0 \leq u \leq \eta$ and $0 \leq c \leq w-1$, as the basis polynomial of $\mathcal{E}_{\mathbf{M}^{(u)}}$. The following theorem finally defines the module basis construction for $\mathcal{I}_{\mathbf{M},l}$.

Theorem 3 [19]: $\mathcal{I}_{\mathbf{M},l}$ can be generated by

$$\mathcal{M} = \{M_{u,c} \mid M_{u,c} = T_{u,c} \prod_{r=0}^{u-1} (z - K_{\underline{\omega}^{(r)}})\}. \quad (13)$$

The Mulders-Storjohann (MS) algorithm [22] can be applied to reduce $\mathcal{M}$ into a Gröbner basis, in which the minimum polynomial is chosen as the interpolation polynomial $\mathcal{Q}$. Afterwards, the estimated message polynomial will be decoded by finding the z-roots of $\mathcal{Q}$. This can be realized by the recursive coefficient search (RCS) algorithm [23, 24]. If multiple z-roots are found, the one whose corresponding codeword has the minimum Euclidean distance to $\underline{\omega}$ is chosen as the decoding output.

III. The ReT-Based Algebraic Soft Decoding

This section further introduces the ReT facilitated ASD, for which the interpolation is realized through the BR approach. We begin with the interpolation points transform.

A. Interpolation Points Transform

The ReT realizes its interpolation complexity reduction through transforming the interpolation points. The z-coordinates of some interpolation points will be transformed into zero, resulting in the module basis polynomials share a common divisor. These interpolation points are called the re-encoding points. The common divisor can be removed during the basis reduction, leading to a reduced complexity.

Let Γ denote the index set of re-encoding points. The re-encoding points are denoted by $\boldsymbol{P}_\Gamma = \{(P_j, \omega_j^{(0)}) \mid j \in \Gamma\}$. The re-encoding polynomial is further defined as

$$K_\Gamma(x,y) = \sum_{j \in \Gamma} \omega_j^{(0)} L_{\Gamma,j}(x,y). \quad (14)$$

Lemma 4 [15]: If $|\Gamma| \leq w\lfloor (k-g)/w \rfloor$ and $\mathbb{S}(\Gamma) = \Gamma$, $K_\Gamma \in \mathcal{L}(\mu P_\infty)$.

Based on (14), the re-encoding codeword is generated by

$$\begin{aligned} \underline{h} &= (K_\Gamma(P_0), K_\Gamma(P_1), \ldots, K_\Gamma(P_{n-1})) \\ &= (h_0, h_1, \ldots, h_{n-1}). \end{aligned} \quad (15)$$

Note that $h_j = \omega_j^{(0)}$, $\forall j \in \Gamma$. Consequently, the multiplicity matrix $\mathbf{M}$ can be transformed into $\overline{\mathbf{M}}$. The interpolation points (P_j, σ_i) are transformed into $(P_j, \sigma_{\bar{i}})$ as

$$(P_j, \sigma_i) \to (P_j, \sigma_{\bar{i}}) : \sigma_{\bar{i}} = \sigma_i - h_j. \quad (16)$$

Let $\overline{\mathbf{M}}^{(u)}$ denote the intermediate multiplicity matrices w.r.t. $\overline{\mathbf{M}}$. Their entries, denoted by $\overline{m}_{ij}^{(u)}$, can also be generated through the process defined by (8).

Let us further define the following polynomials

$$G(x) = \prod_{\sigma_a \in \mathbb{A}(\Gamma)} (x-\sigma_a)^{v_{a,w-1}^{(0)}} \quad (17)$$

and

$$G_\Gamma(x) = \prod_{\sigma_a \in \mathbb{A}(\Gamma)} (x-\sigma_a). \quad (18)$$

In order to show G is a common divisor of the basis polynomials, we define

$$\begin{aligned} \Gamma_u = \{&j \mid \omega_{w\lfloor j/w \rfloor + b}^{(u)} = h_{w\lfloor j/w \rfloor + b}, j \in \Gamma, \\ &\forall b \in [0 : w-1]\}. \end{aligned} \quad (19)$$

Based on Γ_u, we further define

$$G_{\Gamma_u}(x) = \prod_{\sigma_a \in \mathbb{A}(\Gamma_u)} (x-\sigma_a) \quad (20)$$

and

$$G_{\Gamma \setminus \Gamma_u}(x) = \prod_{\sigma_a \in \mathbb{A}(\Gamma \setminus \Gamma_u)} (x-\sigma_a). \quad (21)$$

Polynomial G_Γ can be factorized into $G_\Gamma(x) = G_{\Gamma_u}(x) \cdot G_{\Gamma\setminus\Gamma_u}(x)$. Note that G satisfies

$$G = \prod_{u=0}^{\eta} G_{\Gamma_u}. \tag{22}$$

Let $\overline{\mathfrak{m}}_j^{(u)} = \max_i \{\overline{m}_{ij}^{(u)}\}$ and $\mathcal{E}_{\overline{\mathbf{M}}^{(u)}} = \{S \in \mathcal{R} \mid \nu_{P_j}(S) \geq \overline{\mathfrak{m}}_j^{(u)}\}$. The following Lemma reveals that G is a common divisor of the basis polynomials of $\mathcal{I}_{\overline{\mathbf{M}},l}$.

Lemma 5: If $Q(x,y,z) \in \mathcal{I}_{\overline{\mathbf{M}},l}$, $G|Q(x,y,zG_\Gamma)$.

Proof: Let $\bar{i}_j^{(u)} = \arg\max_i \{\overline{m}_{ij}^{(u)}\}$. Let $z_j^{(u)} = \sigma_{\bar{i}_j^{(u)}}$ and $\underline{z}^{(u)} = (z_0^{(u)}, z_1^{(u)}, \ldots, z_{n-1}^{(u)})$. Since $Q(x,y,z) \in \mathcal{I}_{\overline{\mathbf{M}},l}$, Q can be generated by

$$\overline{\mathcal{M}} = \{\overline{M}_{u,c} \mid \overline{M}_{u,c} = \overline{T}_{u,c} \cdot \prod_{r=0}^{u-1}(z - K_{\underline{z}^{(r)}})\}, \tag{23}$$

where $\overline{T}_{u,c}$ is the basis polynomial of $\mathcal{E}_{\overline{\mathbf{M}}^{(u)}}$, and $K_{\underline{z}^{(r)}}$ is

$$K_{\underline{z}^{(r)}}(x,y) = \sum_{j\in[0:n-1]} z_j^{(r)} L_{[0:n-1],j}, \tag{24}$$

$0 \leq u \leq \eta$ and $0 \leq c \leq w-1$. For each $\overline{M}_{u,c}(x,y,zG_\Gamma)$, it can be expressed as

$$\overline{M}_{u,c}(x,y,zG_\Gamma) = \overline{T}_{u,c} \cdot \prod_{r=0}^{u-1}(zG_\Gamma - K_{\underline{z}^{(r)}}). \tag{25}$$

Since $\prod_{\sigma_a\in\mathbb{A}([0:n-1])}(x-\sigma_a)^{v_{a,w-1}^{(u)}} | \overline{T}_{u,c}$, $\prod_{r=u}^{\eta} G_{\Gamma_r} | \overline{T}_{u,c}$.

Polynomial $K_{\underline{z}^{(r)}}$ can be written as

$$\begin{aligned} K_{\underline{z}^{(r)}} &= \sum_{j\in\Gamma_r} 0 \cdot L_{[0:n-1],j} + \sum_{j\in[0:n-1]\setminus\Gamma_r} z_j^{(r)} L_{[0:n-1],j} \\ &= G_{\Gamma_r} \cdot \sum_{j\in[0:n-1]\setminus\Gamma_r} \frac{z_j^{(r)}}{G_{\Gamma_r}(x_j)} \prod_{\alpha\in\mathbb{A}([0:n-1]\setminus\Gamma_r)\setminus\{x_j\}} \frac{x-\alpha}{x_j-\alpha} \\ &\quad \prod_{\beta\in\mathbb{B}_{x_j}([0:n-1]\setminus\Gamma_r)\setminus\{y_j\}} \frac{y-\beta}{y_j-\beta}. \end{aligned} \tag{26}$$

Therefore, $\prod_{r=0}^{u-1} G_{\Gamma_r} | \prod_{r=0}^{u-1}(zG_\Gamma - K_{\underline{z}^{(r)}})$. Since $G = \prod_{r=0}^{u-1} G_{\Gamma_r} \cdot \prod_{r=u}^{\eta} G_{\Gamma_r}$, $G|Q(x,y,zG_\Gamma)$. □

It can be seen with more re-encoding points, a greater common factor will be obtained. In order to maximize the complexity reduction brought by the ReT, we have

$$|\Gamma| = w\lfloor (k-g)/w \rfloor. \tag{27}$$

Moreover, points of $\boldsymbol{P}_\Gamma$ should correspond to larger multiplicities. By sorting $v_{a,w-1}^{(0)}$ in a descending order, a refreshed index sequence of x-coordinates $a_0, a_1, \ldots, a_{q-1}$ can be obtained. It indicates $v_{a_0,w-1}^{(0)} \geq v_{a_1,w-1}^{(0)} \geq \ldots \geq v_{a_{q-1},w-1}^{(0)}$. The index set of $\boldsymbol{P}_\Gamma$ should correspond to the first $|\Gamma|/w$ of these ordered x-coordinates. Therefore,

$$\begin{aligned} \Gamma = \{&j \mid j = a_\iota + b, 0 \leq \iota \leq \lfloor (k-g)/w \rfloor - 1, \\ &0 \leq b \leq w-1\}. \end{aligned} \tag{28}$$

B. The Modified BR Interpolation

As described in Lemma 5, polynomial G becomes a common divisor of the module basis polynomials. Accordingly, the common divisor can be extracted from the module basis polynomials, resulting a module isomorphism. Let Φ denote the isomorphic module w.r.t. $\mathcal{I}_{\overline{\mathbf{M}},l}$. The module isomorphism between $\mathcal{I}_{\overline{\mathbf{M}},l}$ and Φ is

$$\begin{aligned} &\mathcal{I}_{\overline{\mathbf{M}},l} \to \Phi \\ &\overline{Q}(x,y,z) = G\widetilde{Q}(x,y,\frac{z}{G_\Gamma}) \to \widetilde{Q}(x,y,z). \end{aligned} \tag{29}$$

The isomorphic module Φ can be generated as an $\mathbb{F}_q[x]$-module by

$$\widetilde{\mathcal{M}} = \{\widetilde{M}_{u,c} \mid \widetilde{M}_{u,c} = \widetilde{T}_{u,c} \cdot \prod_{r=0}^{u-1}(zG_{\Gamma\setminus\Gamma_r} - \widetilde{K}_{\underline{z}_r})\}, \tag{30}$$

where

$$\widetilde{T}_{u,c}(x,y) = \frac{\overline{T}_{u,c}(x,y)}{\prod_{r=u}^{\eta} G_{\Gamma_r}(x)} \tag{31}$$

and

$$\begin{aligned} \widetilde{K}_{\underline{z}^{(r)}}(x,y) = &\sum_{j\in[0:n-1]\setminus\Gamma_r} \frac{z_j^{(r)}}{G_{\Gamma_r}(x_j)} \prod_{\alpha\in\mathbb{A}([0:n-1]\setminus\Gamma_r)\setminus\{x_j\}} \frac{x-\alpha}{x_j-\alpha} \\ &\cdot \prod_{\beta\in\mathbb{B}_{x_j}([0:n-1]\setminus\Gamma_r)\setminus\{y_j\}} \frac{y-\beta}{y_j-\beta}. \end{aligned} \tag{32}$$

Based on (30), we can first construct the isomorphism of module basis, and then reduce it into a Gröbner basis, from which $\widetilde{\mathcal{Q}}$ can be retrieved. It will then be restored into $\overline{\mathcal{Q}}$ by

$$\overline{\mathcal{Q}}(x,y,z) = G \cdot \widetilde{\mathcal{Q}}(x,y,\frac{z}{G_\Gamma(x)}). \tag{33}$$

Note that $\widetilde{\mathcal{Q}}$ is the minimum polynomial in Φ w.r.t. $(1, w_z - |\Gamma|)$-revlex order. Afterwards, the RCS algorithm will be applied to find the z-roots of $\overline{\mathcal{Q}}$. Let $\overline{f}$ denote the z-roots of $\overline{\mathcal{Q}}$. The estimated message $\hat{f}$ will be obtained by

$$\hat{f}(x,y) = \overline{f}(x,y) + K_\Gamma(x,y). \tag{34}$$

C. Complexity Reduction

This subsection analyzes the complexity reduction brought by the ReT. We first consider the basis construction complexity. Without the ReT, the complexity of computing $T_{u,c}$ and $K_{\underline{\omega}^{(u)}}$ are $O(l^3n^2w)$ and $O(ln^2)$, respectively. With the ReT, the complexity of computing $\widetilde{T}_{u,c}$ and $\widetilde{K}_{\underline{z}^{(r)}}$ are $O(l^3n(n-|\Gamma|)w)$ and $O(l(n-|\Gamma|)^2)$, respectively. Hence, with the ReT, the basis construction complexity is characterized as $O(l^3n(n-|\Gamma|)w)$. The ReT attributes to a reduction factor of $1-(n-|\Gamma|)/n$.

We further consider the BR complexity that dominates the interpolation. With the MS algorithm, the BR complexity is mainly determined by the x-degree of the maximum polynomial in the module basis [25]. Since the common factor is extracted, x-degrees of the polynomials are reduced. Hence, the BR complexity are characterized as $O(l^5 n^{7/3})$ and $O(l^5(n-|\Gamma|)^2 n^{1/3})$ in the cases without and with the ReT, respectively. Therefore, the ReT can also reduce the BR complexity by a factor of $1-(n-|\Gamma|)^2/n^2$. Hence, it is important to maximize the number of re-encoding points.

IV. The Enhanced Re-encoding Transform

As mentioned above, if more re-encoding points are chosen, the ReT can reduce the interpolation complexity with a greater common factor. However, the sufficient condition in Lemma 4 limits the maximum number of $\boldsymbol{P}_\Gamma$ to $w\lfloor (k-g)/w \rfloor$. It should be noted that even $|\Gamma| > w\lfloor (k-g)/w \rfloor$, it is still possible that $K_\Gamma \in \mathcal{L}(\mu P_\infty)$. In this section, we propose an iterative approach to formulate a larger index set of re-encoding points. For this enhanced ReT variant, their index set and the re-encoding codeword are denoted by $\boldsymbol{P}_{\Gamma'}$, Γ' and $\underline{h}'$, respectively.

Let us initialize Γ' as $\Gamma' = \mathbb{S}([0:n-1])$. The re-encoding polynomial $K_{\Gamma'}$ is computed by

$$K_{\Gamma'}(x,y) = \sum_{j\in\Gamma'} \omega_j^{(0)} L_{\Gamma',j}(x,y). \tag{35}$$

If $\deg_{1,w_z} K_{\Gamma'} \le \mu$, i.e. $K_{\Gamma'}$ fall into the desired Riemann-Roch space, $\underline{h}'$ will be generated by

$$\begin{aligned} \underline{h}' &= (K_{\Gamma'}(P_0), K_{\Gamma'}(P_1), \ldots, K_{\Gamma'}(P_{n-1})) \\ &= (h'_0, h'_1, \ldots, h'_{n-1}). \end{aligned} \tag{36}$$

It will be applied for transforming the interpolation points as in (16). Afterwards, Γ' will be applied for the module basis construction. If $\deg_{1,w_z} K_{\Gamma'} > \mu$, Γ' should be updated. Let $\pi'_j = \max_i\{\pi_{ij}\}$ and $j' = \arg\min_j\{\pi'_j\}$, where $j \in \Gamma'$. The index set Γ' will be updated by

$$\Gamma' = \Gamma' \backslash \mathbb{C}(j'), \tag{37}$$

and the Lagrange interpolation polynomial will be updated by

$$L_{\Gamma',j} = L_{\Gamma',j} \cdot \frac{x_j - x_{j'}}{x - x_{j'}}. \tag{38}$$

The polynomial $K_{\Gamma'}$ will be computed again and checked if $\deg_{1,w_z} K_{\Gamma'} \le \mu$. This iterative formulation of Γ' terminates if any of the following conditions occurs:

1) $\deg_{1,w_z} K_{\Gamma'} \le \mu$;
2) $|\Gamma'| \le w\lfloor (k-g)/w \rfloor$;
3) $\sum_{j\in\Gamma} v^{(0)}_{\lfloor j/w \rfloor, w-1} \ge \sum_{j\in\Gamma'} v^{(0)}_{\lfloor j/w \rfloor, w-1}$.

If 1) occurs, it indicates that a valid Γ' has been generated. Hence, the interpolation points transform and the BR interpolation will be proceeded. The conditions 2) and 3) both indicate that the common divisor generated by Γ' has a lower x-degree than the one generated by Γ. Under these conditions, Γ will be applied for the ReT and the subsequent BR interpolation. The following Section V shows the numerical effect brought by the ReT and the enhanced ReT for the BR interpolation.

V. Numerical Results

This section shows the complexity of the proposed ASD algorithms. The ASD with the ReT facilitated BR interpolation and that with the enhanced ReT facilitated BR interpolation are marked as ASD (ReT-BR) and ASD (E-ReT-BR), respectively. Complexity of the prototype ASD [19] (marked as ASD (BR)) is used as the benchmark. The complexity was measured as the average number of finite field multiplications. They were obtained over the additive white Gaussian noise (AWGN) channel using binary phase shift keying (BPSK) modulation.

TABLE I
Complexity insights of ASD algorithms of the $(64, 47)$ Hermitian code with $l = 4$ and SNR = 8 dB

Algorithm	ReT	Basis Construction	Basis Reduction	Root -finding	Total
ASD (BR)	–	6.14×10^5	5.51×10^5	7.61×10^3	1.17×10^6
ASD (ReT-BR)	3.90×10^3	1.35×10^5	2.57×10^5	6.65×10^3	4.02×10^5
ASD (E-ReT-BR)	1.11×10^4	4.89×10^4	9.14×10^4	1.15×10^3	1.53×10^5

TABLE II
ASD complexity at different SNRs in decoding the $(64, 47)$ Hermitian code

SNR (dB)	ASD (BR)		ASD (ReT-BR)		ASD (E-ReT-BR)	
	$l=2$	$l=4$	$l=2$	$l=4$	$l=2$	$l=4$
4	4.65×10^5	5.19×10^6	$\mathbf{2.62\times10^5}$	$\mathbf{2.96\times10^6}$	2.94×10^5	3.00×10^6
6	3.82×10^5	3.08×10^6	1.97×10^5	1.28×10^6	$\mathbf{1.91\times10^5}$	$\mathbf{1.15\times10^6}$
8	2.00×10^5	1.17×10^6	1.07×10^5	4.02×10^5	$\mathbf{5.74\times10^4}$	$\mathbf{1.53\times10^5}$
10	1.17×10^5	4.99×10^5	6.52×10^4	1.34×10^5	$\mathbf{3.66\times10^4}$	$\mathbf{4.54\times10^4}$

Table I shows the complexity insights for the three ASD algorithms in decoding the $(64, 47)$ Hermitian code with the decoding OLS $l = 4$ and a signal-to-noise ratio (SNR) of 8 dB. Although there is an additional cost for the ReT, both of the two ReT facilitated ASD algorithms exhibit a lower complexity than their prototype ASD. By yielding a greater common factor for the basis polynomials, the enhanced ReT results in a more significant complexity reduction. The ASD (E-ReT-BR) exhibits the lowest interpolation complexity. Table II shows the decoding complexity at different SNRs. The table emphasizes the best performing ASD in grey shapes for each set of l and SNR. For the ASD (E-ReT-BR), in low SNR, it is difficult to generate an improved set of re-encoding points, i.e. Γ'. But there is an additional cost in checking if $K_{\Gamma'} \in \mathcal{L}(\mu P_\infty)$, resulting in a slightly higher complexity than the ASD (ReT-BR). However, as the SNR increases, it is more likely to generate an improved set. The ASD (E-ReT-BR) hence yields the lowest complexity.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62471503; and in part by the Natural Science Foundation of Guangdong Province under Grant 2024A1515010213.

REFERENCES

[1] V. Goppa, "Codes associated with divisors," *Problemy Peredachi Informatsii*, vol. 13, no. 1, pp. 33–39, 1977.

[2] J. Massey, "Shift register synthesis and BCH decoding," *IEEE Trans. Inform. Theory*, vol. 15, no. 1, pp. 122–127, Jan. 1969.

[3] S. Sakata, J. Justesen, Y. Madelung, H. Jensen, and T. Høholdt, "Fast decoding of algebraic-geometric codes up to the designed minimum distance," *IEEE Trans. Inform. Theory*, vol. 41, no. 6, pp. 1672–1677, Nov. 1995.

[4] V. Guruswami and M. Sudan, "Improved decoding of Reed-Solomon and algebraic-geometric codes," *IEEE Trans. Inform. Theory*, vol. 45, no. 6, pp. 1757–1767, Sept. 1999.

[5] R. Kötter, *On algebraic decoding of algebraic-geometric and cyclic codes*. Ph. D Thesis, Univ. Linköping, Linköping, Sweden, 1996.

[6] M. Alekhnovich, "Linear diophantine equations over polynomials and soft decoding of Reed-Solomon codes," *IEEE Trans. Inform. Theory*, vol. 51, no. 7, pp. 2257–2265, Jun. 2005.

[7] K. Lee and M. O'Sullivan, "List decoding of Reed-Solomon codes from a Gröbner basis perspective," *J. Symb. Comput.*, vol. 43, no. 9, pp. 645-658, Sept. 2008.

[8] K. Lee and M. O'Sullivan, "List decoding of Hermitian codes using Gröbner bases," *J. Symb. Comput.*, vol. 44, no. 12, pp. 1662-1675, Nov. 2008.

[9] P. Beelen and K. Brander, "Efficient list decoding of a class of algebraic-geometry codes," *Advances in Mathematics of Communications*, vol. 4, no. 4, pp. 485-518, Nov. 2010.

[10] Y. Wan, L. Chen, and F. Zhang, "Guruswami-Sudan decoding of elliptic codes through module basis reduction," *IEEE Trans. Inform. Theory*, vol. 67, no. 11, pp. 7197–7209, Nov. 2021.

[11] Y. Wan, J. Xing, Y. Huang, T. Wu, B. Bai and G. Zhang, "The re-encoding transform in algebraic list decoding of algebraic geometric codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Taipei, Taiwan, Jun. 2023, pp. 19-24.

[12] R. Kötter, J. Ma, and A. Vardy, "The re-encoding transformation in algebraic list-decoding of Reed-Solomon codes," *IEEE Trans. Inform. Theory*, vol. 57, no. 2, pp. 633–647, Feb. 2011.

[13] J. Bellorado and A. Kavcic, "Low-complexity soft-decoding algorithms for Reed-Solomon codes - part I: an algebraic soft-in hard-out Chase decoder," *IEEE Trans. Inform. Theory*, vol. 56, no. 3, pp. 945–959, Mar. 2010.

[14] Y. Wan, L. Chen, and F. Zhang, "Algebraic Chase decoding of elliptic codes through computing the Gröbner basis," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Espoo, Finland, Jun. 2022, pp. 180-185.

[15] J. Liang, J. Zhao, and L. Chen, "Low-complexity Chase decoding of Hermitian codes with improved interpolation and root-finding," *IEEE Trans. Commun.*, Accepted, 2024.

[16] R. Kötter and A. Vardy, "Algebraic soft-decision decoding of Reed-Solomon codes," *IEEE Trans. Inform. Theory*, vol. 49, no. 11, pp. 2809–2825, Nov. 2003.

[17] Y. Wan, L. Chen and F. Zhang, "Algebraic Soft Decoding of Elliptic Codes," *IEEE Trans. Commun.*, vol. 70, no. 3, pp. 1522-1534, Mar. 2022.

[18] L. Chen, R. Carrasco, and M. Johnson, "Soft-decision list decoding of Hermitian codes," *IEEE Trans. Commun.*, vol. 57, no. 8, pp. 2169-2176, Aug. 2009.

[19] K. Lee and M. O'Sullivan., "Algebraic soft-decision decoding of Hermitian codes," *IEEE Trans. Inform. Theory*, vol. 56, no. 6, pp. 2587-2600, May 2010.

[20] I. Blake, C Heegard, and T. Høholdt, "Algebraic-geometry codes," *IEEE Trans. Inform. Theory*, vol. 44, no. 6, pp. 2596–2618, Oct. 1998.

[21] H. Stichtenoth, *Algebraic Function Fields and Codes*, 2nd ed. New York, NY, USA: Springer-Verlag, 2009.

[22] T. Mulders and A. Storjohann, "On lattice reduction for polynomial matrices," *J. Symb. Comput.*, vol. 35, no. 4, pp. 377–401, Apr. 2003.

[23] R. Roth and G. Ruckenstein, "Efficient decoding of Reed-Solomon codes beyond half the minimum distance," *IEEE Trans. Inform. Theory*, vol. 46, no. 1, pp. 246-257, Jan. 2000.

[24] X. Wu and P. Siegel, "Efficient root-finding algorithm with application to list decoding of algebraic-geometric codes," *IEEE Trans. Inform. Theory*, vol. 47, no. 6, pp. 2579-2587, Sept. 2001.

[25] J. Nielsen and P. Beelen, "Sub-quadratic decoding of one-point Hermitian codes," *IEEE Trans. Inform. Theory*, vol. 61, no. 6, pp. 3225-3240, Jun. 2015.