Progressive Algebraic Soft Decoding of Hermitian Codes

Jiwei Liang, Li Chen, *Senior Member, IEEE*

***Abstract*—Algebraic codes remain an important type of error-correction codes for some data communication and storage systems in which soft information is limited and decoding efficiency is critical. In particular, algebraic-geometric (AG) codes have the potential to replace the widely used Reed-Solomon (RS) codes, due to their greater codeword length and hence error-correction capability. However, their more sophisticated structure leads to complicated decoding computation, which hinders their applications. This paper proposes the progressive algebraic soft decoding (PASD) for one of the most celebrated AG codes - Hermitian codes. The decoding is formulated into finding a desired Gröbner basis that contains the minimum interpolation polynomial. Consequently, size of the interpolation submodule basis can be enlarged progressively, adapting the decoding computation to reliability of the received information. To further reduce complexity, the re-encoding transform (ReT) is introduced to facilitate the decoding. By exploiting the linear code property, ReT results in a common factor in the submodule basis polynomials, which can be extracted to simplify the basis reduction. Our analysis shows that the proposed PASD can reduce the average decoding complexity over its non-progressive prototype, i.e., the algebraic soft decoding (ASD). The complexity reduction effect of ReT is also analyzed. Finally, our simulation results show the progressive decoding preserves the error-correction performance of the ASD.**

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

I. Introduction

ALGEBRAIC codes remain important in data communication and storage systems where soft information is either unavailable, severely limited, or coarsely quantized. E.g., in magnetic recording channels with hard-decision outputs or in some low-resolution receivers, modern codes that rely on probabilistic decoding can hardly demonstrate their error-correction capability. Alternatively, algebraic properties, in particular length n and minimum Hamming distance d of a code, are critical for ensuring decoding performance. Hence, algebraic decoding and their soft-decision variants that only exploit soft information in defining the algebraic decoding computation remain important to be developed. Algebraic geometric (AG) codes form a broad class of algebraic codes [1]. This family encompasses several well-known subclasses, including Reed–Solomon (RS) codes, elliptic codes, and Hermitian codes. RS codes can be viewed as the special AG codes that are constructed based on a straight line. However, length of an RS code cannot exceed size of the underlying finite field, limiting their error-correction capability. In contrast, more general AG codes can achieve a greater codeword lengths and consequently offer stronger error-correction capabilities. AG codes also provide a viable route toward constructing binary codes with favorable distance properties [2, 3].

Similar to RS codes, AG codes can be decoded using the syndrome-based decoding algorithms, where error-correction is realized by determining the error locations and magnitudes, respectively. The celebrated Berlekamp-Massey (BM) algorithm [4] and the Sakata algorithm [5] are the well known syndrome-based decoding algorithms for RS and Hermitian codes, respectively. They exhibit a complexity of $O(n^2)$ and $O(n^{7/3})$ [1], respectively. However, they cannot correct errors beyond half of the code's minimum Hamming distance [2], i.e., $\lfloor \frac{d-1}{2} \rfloor$. By formulating the decoding as a curve-fitting problem [6], the Guruswami-Sudan (GS) algorithm can correct errors beyond this limit. The GS decoding consists of interpolation and root-finding, where the former dominates the decoding complexity. One approach to implement the interpolation is through Kötter's interpolation [7] which constructs the interpolation polynomial in an iterative manner. Alternatively, the basis reduction (BR) approach can be employed. It views the decoding as computing a desired Gröbner basis that contains the interpolation polynomial. The Gröbner basis can be obtained by reducing an interpolation module basis. The BR interpolation-based GS decoding of RS codes was first proposed in [8]. It was later extended to decode the more general AG codes, such as Hermitian codes [9] and elliptic codes [10, 11]. A fast realization for the decoding of Hermitian codes was proposed in [12], yielding a complexity that is subquadratic in n. The computation method is further generalized and improved for general AG codes [13, 14]. Although GS decoding has a better error-correction capability, its interpolation remains computationally expensive. It also dominates the overall GS decoding complexity. Transforming the received word into one with more zero symbols can simplify the interpolation, which is also called re-encoding transform (ReT) technique. It was first introduced for decoding RS codes [15],

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.

J. Liang is with the School of System Science and Engineering, Sun Yat-sen University, Guangzhou, 510006, China (email: liangjw59@mail2.sysu.edu.cn).

L. Chen is with the School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou, 510006, China and also with Guangdong Provincial Key Laboratory of Information Security Technology, Guangzhou, 510006, China (email: chenli55@mail.sysu.edu.cn).

[1] In this paper, the decoding complexity is defined as the number of finite field multiplications in decoding a codeword. In the "big-O" notations, the logarithmic factors are dropped.

[2] For AG codes, it is often called the designed distance which lower bounds the minimum Hamming distance.

and recently generalized to decode general AG codes [16]. It helps reduce the interpolation complexity from $O(l^4 n(n-k))$ to $O(l^4(n-k)^2)$, where k and l are dimension of the code and the designed maximum decoding output list size (OLS), respectively.

Error-correction capability of GS decoding can be improved by utilizing soft information. By formulating test-vectors and exploiting their similarity, the low-complexity Chase (LCC) decoding was applied to decode RS codes [17] and elliptic codes [18, 19], respectively. For Hermitian codes, LCC decoding was first introduced in [20] and later improved using the ReT-facilitated BR interpolation and the fast root-finding in [21]. The algebraic soft decoding (ASD) is another soft-decision GS decoding approach. By transforming soft information into interpolation multiplicity, it achieves a more flexible and proper interpolation multiplicity assignment, enhancing the decoding performance. It was first proposed for RS codes in [22]. Later, the ASD of Hermitian codes using Kötter's interpolation and the BR interpolation were proposed in [23] and [24], respectively. To the other type of AG codes, the BR interpolation-based ASD of elliptic codes was proposed in [25]. Although ASD yields substantial decoding performance improvements, its decoding complexity still remains high. The ReT can be introduced to reduce the ASD complexity. It was adopted to facilitate the ASD of elliptic codes [25], and recently applied by the authors for the ASD of Hermitian codes [26]. The progressive ASD (PASD) was introduced from another perspective, aiming to adapt the decoding computation to the reliability of received information [27, 28]. This is realized by progressively enlarging the OLS l until the intended message (or codeword) is decoded. In the paradigm of BR interpolation, it can be realized through progressively enlarging size of the interpolation submodule basis [29]. However, despite seemingly feasible, PASD has not been studied for decoding AG codes. Designing this more flexible decoding for general AG codes requires a more complex process. Note that earlier research of [29] showed progressive BR interpolation advances from its Kötter counterpart [28], as it does not need to memorize the intermediate decoding operation outcomes, and also being less computationally expensive. Major contributions of this paper are summarized as follows:

1) The PASD for Hermitian codes is proposed. The decoding process starts from an initial interpolation submodule basis. It can be progressively enlarged until the intended message (or codeword) is decoded or the maximum decoding iterations is reached, adapting the decoding computation to quality of the received information.
2) The ReT is introduced to facilitate PASD. By extracting the common factor that ReT introduces in the submodule basis polynomials, the progressive interpolation is further simplified, yielding additional complexity reduction.
3) A comprehensive complexity analysis for the proposed PASD is presented, both theoretically and numerically. It shows that the proposed PASD effectively adapts the decoding complexity to reliability of the received information. Furthermore, the complexity reduction achieved by the ReT is also verified. The error-correction performance of the PASD is provided, demonstrating that the PASD preserves the error-correction performance of its non-progressive prototype.

Note that a preliminary version of the PASD for Hermitian codes was introduced in our earlier conference paper [30]. Due to space constraints, that work omitted many algorithmic details and experimental results, and the integration of the ReT with PASD was not presented. This paper substantially extends [30] by providing the complete algorithmic derivation, comprehensive complexity analysis, and full simulation results, as well as the ReT-facilitated PASD that achieves further complexity reduction.

II. Background Knowledge

This section presents background knowledge of the proposed work, including the encoding 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 $\mathbb{F}_q[X, Y]$ denote the bivariate polynomial ring defined over $\mathbb{F}_q$. An affine Hermitian curve defined over $\mathbb{F}_q$ is given by [31]

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

where $w = \sqrt{q}$ and the curve has a genus $g = w(w-1)/2$. 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_∞. Note that the w^3 affine points can be grouped into w^2 classes with distinct x-coordinates. They form the set of $\mathbb{F}_q$-rational points at H_w. Let $\mathrm{card}(\cdot)$ denote the cardinality of a set. Let $\mathcal{P} = \{P_j = (x_j, y_j) : H_w(x_j, y_j) = 0\}$ denote the set of affine points, and $\mathrm{card}(\mathcal{P}) = w^3$. 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 $\phi_a(x, y) = x^{i_x} y^{i_y}$, where $i_x, i_y \in \mathbb{N}$. 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 $\nu_{P_\infty}(\phi_a^{-1}) = \nu_{P_\infty}((x^{i_x} y^{i_y})^{-1}) = w \cdot i_x + (w+1) \cdot i_y$, where $0 \le i_x \le w$ and $i_y \ge 0$. Monomials of the pole basis exhibit an increasing pole order as $\mathcal{L}_w = \{\phi_a : \nu_{P_\infty}(\phi_a^{-1}) < \nu_{P_\infty}(\phi_{a+1}^{-1})\}$. For each P_j, there also exists a zero basis comprising polynomials with an increasing zero order (or multiplicity) at the point. The zero basis polynomials are [32]

$$\psi_{P_j, v}(x, y) = (x - x_j)^{\lambda} [(y - y_j) - x_j^w (x - x_j)]^{\varkappa}, \tag{3}$$

where $(\lambda, \varkappa) \in \mathbb{N}$ and $v = \lambda + (w+1)\varkappa$. For any non-negative integer a, we use the notation $[a]$ to denote the set $\{0, 1, \ldots, a-1\}$.

Definition I *(Encoding of Hermitian Codes):* For an (n, k) Hermitian code of length n and dimension k, given a message

$\underline{f} = (f_0, f_1, \ldots, f_{k-1}) \in \mathbb{F}_q^k$, its corresponding message polynomial is

$$f(x,y) = f_0\phi_0 + f_1\phi_1 + \cdots + f_{k-1}\phi_{k-1} \in \mathcal{L}(\mu P_\infty), \quad (4)$$

where $f_0, f_1, \ldots, f_{k-1}$ are the message symbols, $\mu = k + g - 1$ and $\mathcal{L}(\mu P_\infty)$ is the Riemann-Roch space defined by μ and P_∞. Codeword $\underline{c} \in \mathbb{F}_q^n$ is generated by

$$\begin{aligned} \underline{c} &= (c_0, c_1, \ldots, c_{n-1}) \\ &= (f(P_0), f(P_1), \ldots, f(P_{n-1})), \end{aligned} \quad (5)$$

where $\{P_0, P_1, \ldots, P_{n-1}\} \subseteq \mathcal{P}$. Note that index set of $\underline{c}$ is $[n]$.

For an (n,k) Hermitian code, Riemann-Roch theorem [33] provides the relationship between μ and $v_{P_\infty}(\phi_{k-1}^{-1})$ as

$$v_{P_\infty}(\phi_{k-1}^{-1}) \leq \mu. \quad (6)$$

Example 1: Consider the Hermitian curve $H_2 = Y^2 + Y - X^3$ defined over $\mathbb{F}_4 = \{0, 1, \alpha, \alpha^2\}$, where α is the primitive element of this field. There are eight affine points on H_2: $P_0 = (0,0)$, $P_1 = (0,1)$, $P_2 = (1,\alpha)$, $P_3 = (1,\alpha^2)$, $P_4 = (\alpha,\alpha)$, $P_5 = (\alpha,\alpha^2)$, $P_6 = (\alpha^2,\alpha)$, $P_7 = (\alpha^2,\alpha^2)$. Let $\mu = 5$. The Riemann-Roch space $\mathcal{L}(5P_\infty)$ is spanned by the basis $\{1, x, y, x^2, xy\}$. Given the message vector $\underline{f} = (\alpha, 1, 0, \alpha^2, 1)$, its corresponding message polynomial is

$$f(x,y) = \alpha + x + \alpha^2 x^2 + xy.$$

The codeword $\underline{c}$ is generated by

$$\begin{aligned} \underline{c} &= (c_0, c_1, c_2, c_3, c_4, c_5, c_6, c_7) \\ &= (f(P_0), f(P_1), f(P_2), f(P_3), f(P_4), f(P_5), f(P_6), f(P_7)) \\ &= (\alpha, \alpha, \alpha, \alpha^2, 1, \alpha^2, 1, \alpha). \end{aligned}$$

B. The GS Decoding

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

Definition II *(Polynomial Ordering):* 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

$$\deg_{1,w_z} \phi_a z^b = v_{P_\infty}(\phi_a^{-1}) + w_z b, \quad (7)$$

where $w_z = v_{P_\infty}(\phi_{k-1}^{-1})$. 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 $\deg_{1,w_z} Q = \max\{\deg_{1,w_z} \phi_a z^b :$
$Q_{ab} \neq 0\}$ and its leading order is $\mathrm{lod}(Q) = \max\{\mathrm{ord}(\phi_a z^b) : Q_{ab} \neq 0\}$.

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 $\mathrm{lod}(Q_1) < \mathrm{lod}(Q_2)$.

The interpolation constraint for a polynomial Q in $\mathcal{R}[z]$ is explained as the follows. Let $\underline{\omega} = (\omega_0, \omega_1, \ldots, \omega_{n-1}) \in \mathbb{F}_q^n$ denote the received word. The set of n interpolation points is defined as

$$\mathbf{P} = \{(P_0, \omega_0), (P_1, \omega_1), \ldots, (P_{n-1}, \omega_{n-1})\}. \quad (8)$$

Given an interpolation point (P_j, ω_j), a polynomial Q can be written as

$$Q = \sum_{\alpha,\beta \in \mathbb{N}} Q_{\alpha,\beta}^{(P_j,\omega_j)} \psi_{P_j,\alpha} (z - \omega_j)^\beta, \quad (9)$$

where $Q_{\alpha,\beta}^{(P_j,\omega_j)} \in \mathbb{F}_q$. If $Q_{\alpha,\beta}^{(P_j,\omega_j)} = 0$ for all $\alpha + \beta < m$, Q has a zero of multiplicity at least m at the point.

Theorem 1 *[9]:* Given a polynomial $\mathfrak{f}$ in the form of (4) and polynomial $\mathcal{Q}$ which has a zero of multiplicity at least m over the n interpolation points, if $m(n - \mathrm{card}(\{j : \mathfrak{f}(P_j) \neq \omega_j,$ for all $j \in [n]\}) > \deg_{1,w_z} \mathcal{Q}$, $\mathcal{Q}(x, y, \mathfrak{f}) = 0$, or equivalently $(z - \mathfrak{f}) | \mathcal{Q}$.

When decoding Hermitian codes, the intended interpolation polynomial $\mathcal{Q}$ which satisfies the interpolation constraints and being the minimum under the $(1, w_z)$-revlex order needs to be determined. The z-roots of $\mathcal{Q}$ are the estimated message polynomials.

C. The ASD and Its BR Interpolation

This subsection introduces the ASD and its BR interpolation. Unlike GS decoding that assigns the same multiplicity to n points, ASD defines interpolation points and their respective multiplicities based on their received soft information. We begin with the reliability transform.

1) *Reliability Transform*: Assume that a 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 $m_{ij} \in \mathbb{Z}_0^+$ is the multiplicity for the interpolation points (P_j, σ_i) [22, 23]. Let i_j denote the row index such that $\sigma_{i_j} = \omega_j$, where ω_j is the hard-decision symbol for the j-th position. Codeword score of $\mathbf{M}$ is defined as

$$s_{\mathbf{M}}(\underline{c}) = \sum_{j=0}^{n-1} m_{i_j j}. \quad (10)$$

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 $\mathbf{M}$, an interpolation ideal $\mathcal{I}_{\mathbf{M}}$ can be defined as

$$\begin{aligned} \mathcal{I}_{\mathbf{M}} = \{Q \in \mathcal{R}[z] : &\mathrm{mult}_{(P_j,\sigma_i)}(Q) \geq m_{ij}, \\ &\text{for all } 0 \leq i \leq q-1 \text{ and } 0 \leq j \leq n-1\}. \end{aligned} \quad (11)$$

Let $\mathcal{R}[z]_l = \{Q \in \mathcal{R}[z] : \deg_z Q \leq l\}$ denote a free module over $\mathbb{F}_q$ of rank $l+1$. The interpolation module $\mathcal{I}_{\mathbf{M},l}$ is defined as [3]

$$\mathcal{I}_{\mathbf{M},l} = \mathcal{I}_{\mathbf{M}} \cap \mathcal{R}[z]_l. \quad (12)$$

[3]The interpolation module $\mathcal{I}_{\mathbf{M},l}$ is a submodule of the free module $\mathcal{R}[z]_l$. In the sequel, the term submodule (of this interpolation module) refers specifically to those submodules of $\mathcal{I}_{\mathbf{M},l}$ that correspond to different intermediate multiplicity matrices $\mathbf{M}^{(u)}$.

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

Theorem 2 *(Sufficient Condition of a Successful ASD Event) [23, 24]:* Given a polynomial $\mathfrak{f}$ in the form of (4) and the interpolation polynomial $\mathcal{Q} \in \mathcal{I}_{\mathbf{M},l}$, if $s_{\mathbf{M}}(\underline{c}) > \deg_{1,w_z} \mathcal{Q}$, $\mathcal{Q}(x, y, \mathfrak{f}) = 0$, or equivalently $(z - \mathfrak{f}) | \mathcal{Q}$.

Polynomial $\mathcal{Q}$ can be determined by BR interpolation. It first derives a series of intermediate multiplicity matrices from $\mathbf{M}$. Let $\eta = \max_j\{\sum_{i=0}^{q-1} m_{ij}\}$. These intermediate multiplicity matrices are denoted as $\mathbf{M}^{(u)}$, where $u = 0, 1, \ldots, \eta$. The first matrix is set as $\mathbf{M}^{(0)} = \mathbf{M}$. Let $i_j^{(u)} = \arg\max_i\{m_{ij}^{(u)}\}$. Matrices $\mathbf{M}^{(1)}, \mathbf{M}^{(2)}, \ldots, \mathbf{M}^{(\eta)}$ are determined sequentially, with their entries $m_{ij}^{(1)}, m_{ij}^{(2)}, \ldots, m_{ij}^{(\eta)}$ updated by

$$m_{ij}^{(u+1)} = \begin{cases} m_{ij}^{(u)} - 1, & \text{if } i = i_j^{(u)} \text{ and } m_{ij}^{(u)} \neq 0, \\ m_{ij}^{(u)}, & \text{if } i \neq i_j^{(u)} \text{ or } m_{ij}^{(u)} = 0. \end{cases} \tag{13}$$

The following definition is further needed for constructing the module basis of $\mathcal{I}_{\mathbf{M},l}$.

Definition III *(Maximum Semi-Grid):* Given an index set $\mathcal{J} \subseteq [n]$, let $\mathbb{A}(\mathcal{J}) = \{x_j : j \in \mathcal{J}\}$. For $\sigma \in \mathbb{A}(\mathcal{J})$, let $\mathbb{B}_\sigma(\mathcal{J}) = \{y_j : (\sigma, y_j) \in \mathcal{P},\ j \in \mathcal{J}\}$. The affine points defined by $\mathcal{J}$ form a maximum semi-grid if $\mathrm{card}(\mathbb{B}_{x_j}(\mathcal{J})) = w$, for all $x_j \in \mathbb{A}(\mathcal{J})$.

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

$$L_{\mathcal{J},j}(x, y) = \prod_{\alpha \in \mathbb{A}(\mathcal{J}) \setminus \{x_j\}} \frac{x - \alpha}{x_j - \alpha} \prod_{\beta \in \mathbb{B}_{x_j}(\mathcal{J}) \setminus \{y_j\}} \frac{y - \beta}{y_j - \beta}. \tag{14}$$

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 $\underline{\omega}^{(u)} = (\omega_0^{(u)}, \omega_1^{(u)}, \ldots, \omega_{n-1}^{(u)})$. The following polynomial can be defined

$$K_{\underline{\omega}^{(u)}}(x, y) = \sum_{j \in [n]} \omega_j^{(u)} L_{[n],j}(x, y). \tag{15}$$

Note that $K_{\underline{\omega}^{(u)}}(P_j) = \omega_j^{(u)}$. Let $\mathfrak{m}_j^{(u)} = \max_i\{m_{ij}^{(u)}\}$, and $\mathcal{E}_{\mathbf{M}^{(u)}}$ be an ideal in $\mathcal{R}$, i.e. $\mathcal{E}_{\mathbf{M}^{(u)}} = \{S \in \mathcal{R} : \nu_{P_j}(S) \geq \mathfrak{m}_j^{(u)}, \text{ for all } j\}$. The interpolation module $\mathcal{I}_{\mathbf{M},l}$ can be computed using $\mathcal{E}_{\mathbf{M}^{(u)}}$ and polynomial $K_{\underline{\omega}^{(u)}}$ as detailed in *Theorem 3* below. To compute $\mathcal{E}_{\mathbf{M}^{(u)}}$, P_j is rearranged to $P_{a,b}$ with $a = \lfloor j/w \rfloor$ and $b = j \mod w$. Subsequently, let $v_{a,b}^{(u)} = \mathfrak{m}_j^{(u)}$. For each a, index b is arranged such that $v_{a,b}^{(u)}$ are assigned in decreasing order as

$$v_{a,0}^{(u)} \geq v_{a,1}^{(u)} \geq \ldots \geq v_{a,w-1}^{(u)}. \tag{16}$$

Given $B_{b,c}^{(u)} \in \mathbb{F}_q[x]$ satisfying $\nu_{P_{a,b}}(y - B_{b,c}^{(u)}) \geq v_{a,b}^{(u)} - v_{a,c}^{(u)}$, we further define the following polynomial as the generator polynomial of $\mathcal{E}_{\mathbf{M}^{(u)}}$

$$T_{u,c}(x, y) = \prod_{a=0}^{q-1} (x - \sigma_a)^{v_{a,c}^{(u)}} \prod_{0 \leq b \leq c-1} (y - B_{b,c}^{(u)}), \tag{17}$$

where $0 \leq u \leq \eta$ and $0 \leq c \leq w - 1$. The basis of $\mathcal{I}_{\mathbf{M},l}$ can be computed based on the following theorem.

Theorem 3 *[24]:* $\mathcal{I}_{\mathbf{M},l}$ can be generated by

$$\mathcal{M} = \{M_{u,c} : M_{u,c} = T_{u,c} \prod_{r=0}^{u-1} (z - K_{\underline{\omega}^{(r)}})\}. \tag{18}$$

Note that the interpolation constraints with respect to $\mathbf{M}^{(u)}$ and $\mathbf{M} \backslash \mathbf{M}^{(u)}$ are satisfied by $T_{u,c}$ and $\prod_{r=0}^{u-1}(z - K_{\underline{\omega}^{(r)}})$, respectively.

We further introduce some properties of multiplicity that are the algebraic foundation for the progressive construction of interpolation submodule. Let $\mathcal{A}_{\mathbb{F}_q}^a$ denote the a-dimensional affine space over $\mathbb{F}_q$. Given a smooth surface $\mathcal{V} = H_w \times \mathcal{A}_{\mathbb{F}}^1$ and a nonzero polynomial $S \in \mathcal{R}$, the following properties hold for multiplicity on $\mathcal{V}$ [24]:

(1) For an interpolation point (P_j, σ_i), $\nu_P(S) = \mathrm{mult}_{(P_j,\sigma_i)}(S)$;

(2) $\mathrm{mult}_{(P_j,\sigma_i)}(z - S) = 1$ if $S(P_j,) = \sigma_i$, and 0 otherwise;

(3) For $S_a, S_b \in \mathcal{R}[z]$, $\mathrm{mult}_{(P_j,\sigma_i)}(S_a S_b) = \mathrm{mult}_{(P_j,\sigma_i)}(S_a) + \mathrm{mult}_{(P_j,\sigma_i)}(S_b)$.

These properties, which delineate how multiplicity behaves under linear and multiplicative operations, are the foundation for progressively constructing the interpolation submodule. The following lemma exploits them to establish how the interpolation submodule of $\mathbf{M}^{(u-1)}$ can be constructed progressively from that of $\mathbf{M}^{(u)}$.

Lemma 4 *[24]:* Given two intermediate multiplicity matrices $\mathbf{M}^{(u-1)}$ and $\mathbf{M}^{(u)}$, their corresponding interpolation submodule $\mathcal{I}_{\mathbf{M}^{(u-1)},l}$ and $\mathcal{I}_{\mathbf{M}^{(u)},l}$ satisfy

$$\mathcal{I}_{\mathbf{M}^{(u-1)},l} = \mathcal{I}_{\mathbf{M}^{(u)},l}(z - K_{\underline{\omega}^{(u-1)}}) + \mathcal{E}_{\mathbf{M}^{(u-1)}}. \tag{19}$$

It formulates the key principle of the proposed PASD. It shows that the submodule basis for a more constrained interpolation problem can be derived incrementally from the solution of a less constrained one. Explicit progressive construction of the submodule basis will be presented in Section III.

After the basis construction, the Mulders-Storjohann (MS) algorithm [34] can be applied to reduce $\mathcal{M}$ into a Gröbner basis, from which the minimum polynomial is chosen as the interpolation polynomial [4]. Afterwards, the estimated message polynomial f_{est} will be decoded by finding z-roots of $\mathcal{Q}$. This can be realized by the recursive coefficient search (RCS) algorithm [35–37]. If multiple z-roots are found, the one whose corresponding codeword yields the minimum Hamming distance to $\underline{\omega}$ will be chosen as the decoding output.

Example 2 (Continued from Example 1): Supposed that the codeword in the *Example 1* in transmitted over a noisy channel. Given $l = 4$ and the multiplicity matrix

$$\mathbf{M} = \begin{bmatrix} 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 4 & 0 & 3 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 & 0 & 4 \\ 0 & 1 & 0 & 4 & 0 & 3 & 0 & 0 \end{bmatrix},$$

[4] Although there are other algorithms that have better asymptotic complexity, the MS algorithm is more efficient for handling codes of practical length.

The intermediate multiplicity matrices can be generated. Let $\mathbf{M}^{(0)} = \mathbf{M}$ and $\mathbf{M}^{(4)} = \mathbf{0}$ (zero matrix). Based on (13), matrices $\mathbf{M}^{(1)}$ to $\mathbf{M}^{(3)}$ can be generated as

$$\mathbf{M}^{(1)} = \begin{bmatrix} 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 3 & 0 & 2 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 3 \\ 0 & 1 & 0 & 3 & 0 & 3 & 0 & 0 \end{bmatrix},$$

$$\mathbf{M}^{(2)} = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 2 \\ 0 & 1 & 0 & 2 & 0 & 2 & 0 & 0 \end{bmatrix},$$

$$\mathbf{M}^{(3)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix},$$

respectively, vectors $\underline{\omega}^{(0)}$ to $\underline{\omega}^{(3)}$ can also be constituted by

$$\begin{aligned} \underline{\omega}^{(0)} &= (0, 1, \alpha, \alpha^2, 1, \alpha^2, 1, \alpha), \\ \underline{\omega}^{(1)} &= (0, 1, \alpha, \alpha^2, 1, \alpha^2, 1, \alpha), \\ \underline{\omega}^{(2)} &= (0, \alpha^2, \alpha, \alpha^2, 1, \alpha^2, 1, \alpha), \\ \underline{\omega}^{(3)} &= (0, \times, \times, \alpha^2, 1, \alpha^2, \times, \alpha^2). \end{aligned}$$

The module basis $\mathcal{M}$ is constructed. Its full expression is provided in Appendix. The subsequent basis reduction and root-finding are performed. This decoding procedure requires 6085 finite field additions and 3803 multiplications.

III. Progressive BR Interpolation

This section presents the progressive BR interpolation, which enables ASD to find the interpolation polynomial $\mathcal{Q}$ by progressively constructing the interpolation submodule.

A. Progressive Submodule Basis Construction

To realize the progressive submodule basis construction, the following definition is needed.

Definition IV *(Delta Function):* For integers a and b, the function $\delta_{a,b}$ is defined as:

$$\delta_{a,b} = \begin{cases} 1, & \text{if } a = b, \\ 0, & \text{otherwise.} \end{cases} \tag{20}$$

Let $\mathbf{M}$ and $\mathbf{M}'$ denote two multiplicity matrices where their entries m_{ij} and m'_{ij} satisfy

$$m'_{ij} = \begin{cases} m_{ij} - 1, & \text{if } i = i_j \text{ and } m_{ij} \neq 0, \\ m_{ij}, & \text{if } i \neq i_j \text{ or } m_{ij} = 0, \end{cases} \tag{21}$$

where $i_j = \arg\max_i \{m_{ij}\}$. Recall that $\mathcal{E}_{\mathbf{M}}$ is an ideal in $\mathcal{R}$ and $\mathcal{E}_{\mathbf{M}} = \{S \in \mathcal{R} : \nu_{P_j}(S) \geq \mathfrak{m}_j\}$. Let $\mathcal{I}_{\mathbf{M}',l}$ denote the interpolation module with respect to $\mathbf{M}'$. Note that $\mathcal{I}_{\mathbf{M}',l}$ is a submodule of $\mathcal{I}_{\mathbf{M},l}$. The following lemma shows how to construct the basis of $\mathcal{I}_{\mathbf{M},l}$ using the module basis of $\mathcal{I}_{\mathbf{M}',l}$ and $\mathcal{E}_{\mathbf{M}}$.

Lemma 5*:* The interpolation module $\mathcal{I}_{\mathbf{M},l}$ can be generated by

$$\mathcal{M} = \left\{ M_{u,c} = \delta_{u,0} T_{u,c} + (1 - \delta_{u,0}) M'_{u-1,c} \cdot (z - K_{\underline{\omega}^{(0)}}) \right\}, \tag{22}$$

where $M'_{u,c}$ is the module basis polynomial of $\mathcal{I}_{\mathbf{M}',l}$.

Proof: Since $M_{u,c} = T_{u,c} \prod_{r=0}^{u-1} (z - K_{\underline{\omega}^{(r)}})$, if $u = 0$, $M_{0,c} = T_{0,c}$.

If $u \neq 0$, based on (13), the intermediate multiplicity matrices of $\mathbf{M}'$ and $\mathbf{M}$ satisfy

$$\mathbf{M}'^{(u-1)} = \mathbf{M}^{(u)}. \tag{23}$$

Let $i'^{(u)}_j = \arg\max_i \{m'^{(u)}_{ij}\}$ and $\mathfrak{m}'^{(u)}_j = \max_i \{m'^{(u)}_{ij}\}$. Further let $\mathcal{E}_{\mathbf{M}'}$ denote an ideal in $\mathcal{R}$ and $\mathcal{E}_{\mathbf{M}'^{(u)}} = \{S \in \mathcal{R} : \nu_{P_j}(S) \geq \mathfrak{m}'^{(u)}_j\}$. Based on (23), $\mathcal{E}_{\mathbf{M}'^{(u-1)}} = \mathcal{E}_{\mathbf{M}^{(u)}}$ and their generators satisfy

$$T'_{u-1,c} = T_{u,c}. \tag{24}$$

Let $\omega'^{(u)}_j = \sigma_{i'^{(u)}_j}$ and $\underline{\omega}'^{(u)} = (\omega'^{(u)}_0, \omega'^{(u)}_1, \ldots, \omega'^{(u)}_{n-1})$. We define the following polynomial

$$K_{\underline{\omega}'^{(u)}}(x, y) = \sum_{j \in [n]} \omega'^{(u)}_j L_{[n],j}(x, y). \tag{25}$$

Since $\underline{\omega}'^{(u-1)} = \underline{\omega}^{(u)}$,

$$K_{\underline{\omega}'^{(u-1)}} = K_{\underline{\omega}^{(u)}}. \tag{26}$$

Therefore,

$$\begin{aligned} M'_{u-1,c} &= T'_{u-1,c} \prod_{r=0}^{u-2} (z - K_{\underline{\omega}'^{(r)}}) \\ &= T_{u,c} \prod_{r=1}^{u-1} (z - K_{\underline{\omega}^{(r)}}). \end{aligned} \tag{27}$$

Based on (27), if $u \neq 0$,

$$\begin{aligned} M_{u,c} &= T_{u,c} \prod_{r=0}^{u-1} (z - K_{\underline{\omega}^{(r)}}) \\ &= M'_{u-1,c} \cdot (z - K_{\underline{\omega}^{(0)}}). \end{aligned} \tag{28}$$

□

Lemma 5 can also be generalized for constructing the submodule basis of $\mathcal{M}^{(u-1)}$, as presented in the following corollary.

Corollary 6*:* For $1 \leq u \leq \eta - 1$, $0 \leq u' \leq \eta - u + 1$ and $0 \leq c \leq w - 1$, the submodule basis $\mathcal{M}^{(u-1)}$ with respect to the intermediate multiplicity matrix $\mathbf{M}^{(u-1)}$ can be generated by

$$\begin{aligned} \mathcal{M}^{(u-1)} = \{ & M^{(u-1)}_{u',c} = \delta_{u',0} T_{u-1,c} \\ & + (1 - \delta_{u',0}) M^{(u)}_{u'-1,c} \cdot (z - K_{\underline{\omega}^{(u-1)}}) \}. \end{aligned} \tag{29}$$

Lemma 5 and *Corollary 6* imply that the submodule bases can be constructed progressively, leading to the proposed progressive interpolation. As described in Section II, in the ASD, $\mathbf{M}$ is first formulated using the designed maximum OLS l. This matrix is then used to generate a series of intermediate

multiplicity matrices $\mathbf{M}^{(u)}$, where $u = 0, 1, \ldots, \eta$. Since $\mathbf{M}^{(\eta)}$ is an all-zero matrix, the submodule basis with respect to $\mathbf{M}^{(\eta-1)}$, denoted by $\mathcal{M}^{(\eta-1)}$, is first computed [5]. The MS algorithm is applied to reduce it into a Gröbner basis, from which the interpolation polynomial can be obtained. The estimated message polynomial is computed by the root-finding procedure. If it is decoded, the decoding terminates and outputs the result [6] If the decoding fails, $\mathcal{M}^{(\eta-1)}$ is applied to construct $\mathcal{M}^{(\eta-2)}$. The subsequent basis reduction and root-finding will be performed. This progressive iteration continues until the decoding with respect to $\mathbf{M}^{(u)}$ succeeds, or the decoding with respect to $\mathbf{M}^{(0)}$ fails, i.e. the maximum number of progressive iterations is reached. As described above, the interpolation constraints with respect to the multiplicity matrix increase progressively. With reliable received information, the decoding can often terminate earlier, i.e. decoding succeeds with satisfying fewer interpolation constraints, thereby adapting the decoding complexity to the channel conditions. Similar to *Theorem* 2, a sufficient condition for successful PASD is given in the following corollary.

Corollary 7: Given a polynomial $\mathfrak{f}$ in the form of (4) and the interpolation polynomial $\mathcal{Q} \in \mathcal{I}_{\mathbf{M}^{(u)},l}$, if $s_{\mathbf{M}^{(u)}}(\underline{c}) > \deg_{1,w_z} \mathcal{Q}$, $\mathcal{Q}(x, y, \mathfrak{f}) = 0$, or equivalently $(z - \mathfrak{f}) | \mathcal{Q}$.

Example 3: Given $l = 4$, and the same multiplicity matrix in *Example 2*, the progressive decoding can start with $\mathbf{M}^{(3)}$. The submodule basis with respect to $\mathbf{M}^{(3)}$ is first constructed.

$$\begin{aligned}
M_{0,0}^{(3)} &= x + x^4, \\
M_{0,1}^{(3)} &= \alpha y + \alpha^2 x + xy + x^3, \\
M_{1,0}^{(3)} &= z + x + xy + x^2 y + \alpha^2 x^3 y, \\
M_{1,1}^{(3)} &= yz + x^2 y + \alpha^2 x^3 y + x^4 + x^5 + \alpha^2 x^6.
\end{aligned}$$

After basis reduction and root-finding, the decoding fails to find the intended message polynomial. Therefore, $\mathcal{M}^{(2)}$ will be constructed based on $\mathcal{M}^{(3)}$:

$$\begin{aligned}
M_{0,0}^{(2)} &= x^2 + x^8, \\
M_{0,1}^{(2)} &= \alpha xy + \alpha^2 x^2 + x^2 y + x^4 + \alpha x^4 y + \alpha^2 x^5 \\
&\quad + x^5 y + x^7, \\
M_{1,0}^{(2)} &= M_{0,0}^{(3)} \cdot (z - K_{\underline{\omega}^{(2)}}), \\
M_{1,1}^{(2)} &= M_{0,1}^{(3)} \cdot (z - K_{\underline{\omega}^{(2)}}), \\
M_{2,0}^{(2)} &= M_{1,0}^{(3)} \cdot (z - K_{\underline{\omega}^{(2)}}), \\
M_{2,1}^{(2)} &= M_{1,1}^{(3)} \cdot (z - K_{\underline{\omega}^{(2)}}).
\end{aligned}$$

Note that the fully expanded forms of $M_{1,0}^{(2)}$ through $M_{2,1}^{(2)}$ are provided in Appendix. It terminates after the decoding with respect to $\mathcal{M}^{(1)}$ and successfully outputs the estimated message polynomial. In this example, the progressive decoding does not need to proceed to the decoding with respect to $\mathcal{M}^{(0)}$. It requires 2474 finite field additions and 1905 multiplications, fewer than those required in *Example 2*.

[5]The PASD is presented as starting from $\mathcal{M}^{(\eta-1)}$ to maintain generality across all multiplicity assignment methods. For some methods (e.g., Algorithm A of [22]), decoding may not succeed immediately at this stage due to weak constraints. Similarly, the maximum number of progressive iterations is determined by the specific multiplicity assignment method and may exceed l.

[6]The estimated message polynomial will be encoded into the estimated codeword. It can be assessed by the maximum likelihood (ML) criterion [38, 39] or the acceptance criterion [40].

B. Progressive Construction Using Reduced Submodule Basis

In the aforementioned progressive decoding, since $\mathcal{M}^{(u-1)}$ is generated from $\mathcal{M}^{(u)}$, $\mathcal{M}^{(u)}$ should be retained in memory, resulting in supplementary storage requirements. Let $\Theta^{(u)} = \{\theta_{u',c}^{(u)}\}$ denote the reduced Gröbner basis derived from $\mathcal{M}^{(u)}$, where $0 \le u \le \eta - 1$, $0 \le u' \le \eta - u + 1$ and $0 \le c \le w - 1$. The following lemma reveals that $\mathcal{M}^{(u-1)}$ can also be generated based on $\Theta^{(u)}$.

Lemma 8: The module basis $\mathcal{M}^{(u-1)}$ with respect to $\mathbf{M}^{(u-1)}$ can be generated by

$$\begin{aligned}
\mathcal{M}^{(u-1)} = \{ & M_{u',c}^{(u-1)} = \delta_{u',0} T_{u-1,c} \\
& + (1 - \delta_{u',0}) \theta_{u'-1,c}^{(u)} \cdot (z - K_{\underline{\omega}^{(u-1)}}) \},
\end{aligned} \tag{30}$$

where $1 \le u \le \eta - 1$, $0 \le u' \le \eta - u + 1$ and $0 \le c \le w - 1$.

Proof: Since all basis polynomials $\theta_{u'-1,c}^{(u)}$ in $\Theta^{(u)}$ satisfy $\mathrm{mult}_{(P_j,\sigma_i)}(\theta_{u'-1,c}^{(u)}) \ge m_{ij}^{(u)}$,

$$\mathrm{mult}_{(P_j,\sigma_i)}(z - K_{\underline{\omega}^{(u-1)}}) \ge m_{ij}^{(u-1)} - m_{ij}^{(u)} \tag{31}$$

and

$$\mathrm{mult}_{(P_j,\sigma_i)}(\theta_{u'-1,c}^{(u)}(z - K_{\underline{\omega}^{(u-1)}})) \ge m_{ij}^{(u-1)}. \tag{32}$$

Therefore, $\mathcal{M}^{(u-1)}$ belongs to the module defined by $\mathbf{M}^{(u-1)}$.

We continue to prove that $\mathcal{M}^{(u-1)}$ is the submodule basis with respect to $\mathbf{M}^{(u-1)}$. For any polynomial Q in the module defined by $\mathbf{M}^{(u-1)}$,

$$Q = \sum_{s=0}^{\eta-u+1} Q_{[s]} z^s, \tag{33}$$

where $Q_{[s]} \in \mathcal{E}_{\mathbf{M}^{(s)}}$. There exist $\zeta_0^{(s)}, \zeta_1^{(s)}, \ldots, \zeta_{w-1}^{(s)} \in \mathbb{F}_q[x]$, such that

$$Q_{[s]} = \sum_{c=0}^{w-1} \zeta_c^{(s)} T_{s,c}. \tag{34}$$

It enables

$$Q^{(\eta-u)} = Q - \sum_{c=0}^{w-1} \zeta_c^{(\eta-u+1)} M_{\eta-u+1,c}^{(u-1)}, \tag{35}$$

where $\deg_z Q^{(\eta-u)} \le \eta - u$. Again, there exist $\zeta_0^{(\eta-u)}, \zeta_1^{(\eta-u)}, \ldots, \zeta_{w-1}^{(\eta-u)} \in \mathbb{F}_q[x]$ such that

$$Q^{(\eta-u-1)} = Q^{(\eta-u)} - \sum_{c=0}^{w-1} \zeta_c^{(\eta-u)} M_{\eta-u,c}^{(u-1)}, \tag{36}$$

where $\deg_z Q^{(\eta-u-1)} \le \eta - u - 1$. This deduction continues, leading to the existence of $\zeta_0^{(1)}, \zeta_1^{(1)}, \ldots, \zeta_{w-1}^{(1)} \in \mathbb{F}_q[x]$ such that

$$Q^{(0)} = Q^{(1)} - \sum_{c=0}^{w-1} \zeta_c^{(1)} M_{1,c}^{(u-1)}. \tag{37}$$

Therefore, polynomial $Q^{(0)}$ can be written as $Q^{(0)} = \sum_{c=0}^{w-1} \zeta_c^{(0)} M_{0,c}^{(u-1)}$, where $\zeta_0^{(0)}, \zeta_1^{(0)}, \ldots, \zeta_{w-1}^{(0)} \in \mathbb{F}_q[x]$. Any polynomial Q in the module defined by $\mathbf{M}^{(u-1)}$ can be expressed as an $\mathbb{F}_q[x]$-linear combination of $M_{u',c}^{(u-1)}$. □

For the MS basis reduction algorithm, its complexity is determined by weighted degree of the maximum polynomial in the module basis [12]. Since $\Theta^{(u)}$ is reduced from $\mathcal{M}^{(u)}$, its weighted degree is not greater than that of $\mathcal{M}^{(u)}$. Therefore, for the basis reduction of $\mathcal{M}^{(u-1)}$, performing it as in (30) has lower complexity than performing it as in (29).

The proposed PASD of Hermitian codes is summarized in Algorithm 1.

Algorithm 1 The PASD of Hermitian codes

Input: $\mathbf{\Pi}$;
Output: f_{est};
1: Transform $\mathbf{\Pi}$ to $\mathbf{M}$;
2: Generate $\mathbf{M}^{(0)}, \mathbf{M}^{(1)}, \ldots, \mathbf{M}^{(\eta)}$ as in (13) and set $\kappa = 1$;
3: Construct the submodule basis with respect to $\mathbf{M}^{(\eta-1)}$ as in (18);
4: Perform the basis reduction using the basis reduction algorithm, obtaining $\mathcal{Q}$;
5: Perform the root-finding algorithm to compute f_{est};
6: **While** decoding fails or $\kappa < \eta$
7: Let $\kappa = \kappa + 1$;
8: Construct the module basis with respect to $\mathbf{M}^{(\eta-\kappa)}$ as in (30);
9: Perform the basis reduction using the basis reduction algorithm, obtaining $\mathcal{Q}$;
10: Perform the root-finding algorithm to compute f_{est}.
11: **End While**

Example 4: Given the same Hermitian code and the same $\mathbf{M}$ as in *Example 2*, the module basis with respect to $\mathcal{M}^{(3)}$ is constructed as illustrated by *Example 3*. The decoding fails to find the estimated message polynomial. We continue to construct $\mathcal{M}^{(2)}$ as in (30).

$$\begin{aligned}
M_{0,0}^{(2)} &= x^2 + x^8,\\
M_{0,1}^{(2)} &= \alpha xy + \alpha^2 x^2 + x^2 y + x^4 + \alpha x^4 y + \alpha^2 x^5\\
&\quad + x^5 y + x^7,\\
M_{1,0}^{(2)} &= \theta_{0,0}^{(3)} \cdot (z - K_{\underline{\omega}^{(2)}}),\\
M_{1,1}^{(2)} &= \theta_{0,1}^{(3)} \cdot (z - K_{\underline{\omega}^{(2)}}),\\
M_{2,0}^{(2)} &= \theta_{1,0}^{(3)} \cdot (z - K_{\underline{\omega}^{(2)}}),\\
M_{2,1}^{(2)} &= \theta_{1,1}^{(3)} \cdot (z - K_{\underline{\omega}^{(2)}}),
\end{aligned}$$

where $\theta_{u',c}^{(3)}$ denotes the reduced basis obtained from $\mathcal{M}^{(3)}$ in *Example 3*. The fully expanded forms are provided in Appendix. It can be seen that the weighted degrees of the maximum basis polynomial in *Examples 3* and *4* are 21 and 17, respectively. The decoding is terminated after performing the BR interpolation with respect to $\mathcal{M}^{(1)}$, which is the same as in *Example 3*. The whole decoding procedure requires 2116 finite field additions and 1393 multiplications. Compared with *Example 3*, decoding complexity is reduced.

IV. ReT-Based Progressive BR Interpolation

This section introduces the ReT-based progressive BR interpolation. It exploits the linear property of Hermitian codes to transform a subset of interpolation points using a regenerated codeword. This results in a common factor in the module basis polynomials, which can be extracted for simplifying the progressive BR interpolation. First, we revisit the ReT for Hermitian codes.

A. Re-encoding Transform

For the ReT, a certain number of interpolation points will be chosen to reconstruct a codeword. Let Γ denote the index set of the re-encoding points, which can be further denoted by $\boldsymbol{P}_\Gamma = \{(P_j, \omega_j^{(0)}) : j \in \Gamma\}$. The re-encoding polynomial is further defined as

$$K_\Gamma(x, y) = \sum_{j \in \Gamma} \omega_j^{(0)} L_{\Gamma,j}(x, y). \tag{38}$$

Let $\mathbb{S}(\mathcal{J}) = \{j : \text{card}(\mathbb{B}_{x_j}(\mathcal{J})) = w\}$. The following lemma implies the sufficient condition for generating a valid re-encoding codeword through evaluating K_Γ over the points of $\mathcal{P}$.

Lemma 9 *[21]:* If $\text{card}(\Gamma) \le w\lfloor (k-g)/w \rfloor$ and $\mathbb{S}(\Gamma) = \Gamma$, $K_\Gamma \in \mathcal{L}(\mu P_\infty)$.

Based on (38), the re-encoding codeword is generated by

$$\begin{aligned}
\underline{h} &= (K_\Gamma(P_0), K_\Gamma(P_1), \ldots, K_\Gamma(P_{n-1}))\\
&= (h_0, h_1, \ldots, h_{n-1}).
\end{aligned} \tag{39}$$

Note that $h_j = \omega_j^{(0)}$, for all $j \in \Gamma$. Using $\underline{h}$, the interpolation points (P_j, σ_i) are transformed as

$$(P_j, \sigma_i) \mapsto (P_j, \sigma_{\widehat{i}}) : \sigma_{\widehat{i}} = \sigma_i - h_j. \tag{40}$$

Accordingly, the multiplicity matrix $\mathbf{M}$ can be transformed into $\widehat{\mathbf{M}}$, where its entries are denoted by $\widehat{m}_{ij}$. This transformation maps each entry according to the rule induced by (40), such that

$$\widehat{m}_{\widehat{ij}} = m_{ij}. \tag{41}$$

Let $\widehat{\mathbf{M}}^{(u)}$ denote the intermediate multiplicity matrices with respect to $\widehat{\mathbf{M}}$. Their entries, denoted by $\widehat{m}_{ij}^{(u)}$, can also be generated through the process defined by (13).

In order to maximize the complexity reduction brought by the ReT, the number of re-encoding points is set as

$$|\Gamma| = w\lfloor \frac{k-g}{w} \rfloor. \tag{42}$$

Moreover, points in $\boldsymbol{P}_\Gamma$ should be associated with larger multiplicities in the columns they belong to. 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 $v_{a_0,w-1}^{(0)} \ge v_{a_1,w-1}^{(0)} \ge \ldots \ge v_{a_{q-1},w-1}^{(0)}$. The index set of $\boldsymbol{P}_\Gamma$ should be defined by the first $\text{card}(\Gamma)/w$ of the ordered x-coordinates. Therefore,

$$\Gamma = \{j : j = a_\iota + b, 0 \le \iota \le \lfloor \frac{k-g}{w} \rfloor - 1, \quad 0 \le b \le w-1\}. \tag{43}$$

Let us define

$$G(x) = \prod_{\sigma_a \in \mathbb{A}(\Gamma)} (x - \sigma_a)^{v^{(0)}_{a,w-1}} \tag{44}$$

and

$$G_\Gamma(x) = \prod_{\sigma_a \in \mathbb{A}(\Gamma)} (x - \sigma_a). \tag{45}$$

Furthermore, let

$$\Gamma_u = \{j : \ \omega^{(u)}_{w\lfloor j/w \rfloor + b} = h_{w\lfloor j/w \rfloor + b}, j \in \Gamma, \text{ for all } b \in [w]\}. \tag{46}$$

Subsequently, let

$$G_{\Gamma_u}(x) = \prod_{\sigma_a \in \mathbb{A}(\Gamma_u)} (x - \sigma_a) \tag{47}$$

and

$$G_{\Gamma \setminus \Gamma_u}(x) = \prod_{\sigma_a \in \mathbb{A}(\Gamma \setminus \Gamma_u)} (x - \sigma_a). \tag{48}$$

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

$$G = \prod_{u=0}^{\eta} G_{\Gamma_u}. \tag{49}$$

The following lemma reveals that G is a common factor of the basis polynomials of $\mathcal{I}_{\widehat{\mathbf{M}},l}$.

Lemma 10*[26]:* If $Q(x,y,z) \in \mathcal{I}_{\widehat{\mathbf{M}},l}$, $G | Q(x,y,zG_\Gamma)$.

Let Φ denote the isomorphic module with respect to $\mathcal{I}_{\widehat{\mathbf{M}},l}$. The module isomorphism between $\mathcal{I}_{\widehat{\mathbf{M}},l}$ and Φ is

$$\begin{aligned} &\mathcal{I}_{\widehat{\mathbf{M}},l} \mapsto \Phi \\ &\widehat{Q}(x,y,z) = G\widetilde{Q}(x,y,\frac{z}{G_\Gamma}) \mapsto \widetilde{Q}(x,y,z). \end{aligned} \tag{50}$$

Let $\widehat{\mathfrak{m}}^{(u)}_j = \max_i \{\widehat{m}^{(u)}_{ij}\}$ and $\mathcal{E}_{\widehat{\mathbf{M}}^{(u)}} = \{S \in \mathcal{R} : \nu_{P_j}(S) \geq \widehat{\mathfrak{m}}^{(u)}_j\}$. Let $\widehat{i}^{(u)}_j = \arg\max_i \{\widehat{m}^{(u)}_{ij}\}$, $z^{(u)}_j = \sigma_{\widehat{i}^{(u)}_j}$ and $\underline{z}^{(u)} = (z^{(u)}_0, z^{(u)}_1, \ldots, z^{(u)}_{\eta-1})$. The isomorphic module Φ can be generated as an $\mathbb{F}_q[x]$-module by

$$\widetilde{\mathcal{M}} = \{\widetilde{M}_{u,c} \mid \widetilde{M}_{u,c} = \widetilde{T}_{u,c} \cdot \prod_{r=0}^{u-1} (zG_{\Gamma \setminus \Gamma_r} - \widetilde{K}_{\underline{z}^{(r)}})\}, \tag{51}$$

where

$$\widetilde{T}_{u,c}(x,y) = \frac{\widehat{T}_{u,c}(x,y)}{\prod_{r=u}^{\eta} G_{\Gamma_r}(x)} \tag{52}$$

and

$$\begin{aligned} \widetilde{K}_{\underline{z}^{(r)}}(x,y) = &\sum_{j \in [n] \setminus \Gamma_r} \frac{z^{(r)}_j}{G_{\Gamma_r}(x_j)} \prod_{\alpha \in \mathbb{A}([n] \setminus \Gamma_r) \setminus \{x_j\}} \frac{x - \alpha}{x_j - \alpha} \\ &\cdot \prod_{\beta \in \mathbb{B}_{x_j}([n] \setminus \Gamma_r) \setminus \{y_j\}} \frac{y - \beta}{y_j - \beta}. \end{aligned} \tag{53}$$

Note that with the ReT, the polynomials is organized under the $(1, w_z - \mathrm{card}(\Gamma))$-revlex order. As mentioned above, the BR interpolation is performed on the isomorphic module. Extracting the common factor $G(x)$ of the basis polynomials of $\mathcal{I}_{\widehat{\mathbf{M}},l}$ leads to a simpler basis reduction in finding the interpolation polynomial $\mathcal{Q}$.

B. ReT-Based Progressive Submodule Basis Construction

The following subsection shows how to construct the submodule basis progressively under the ReT paradigm. Similar to the progressive submodule basis construction described in Section III, the ReT-based submodule basis construction can also be performed in a progressive manner. Let us define the following polynomial

$$G^{(u)} = \prod_{u'=u}^{\eta} G_{\Gamma_{u'}}. \tag{54}$$

Let $\mathcal{I}_{\widehat{\mathbf{M}}^{(u)},l}$ denote the interpolation submodule with respect to $\widehat{\mathbf{M}}^{(u)}$. Similar to *Lemma 10*, the following lemma reveals that $G^{(u)}$ is a common factor of the module basis polynomials of $\mathcal{I}_{\widehat{\mathbf{M}}^{(u)},l}$.

Lemma 11*:* If $Q(x,y,z) \in \mathcal{I}_{\widehat{\mathbf{M}}^{(u-1)},l}$, $G^{(u-1)} | Q(x,y,zG_\Gamma)$.

Proof: Let $\widehat{\Theta}^{(u)} = \{\widehat{\theta}^{(u)}_{u',c}\}$ denote the Gröbner basis derived from $\widehat{\mathcal{M}}^{(u)}$, where $1 \leq u \leq \eta - 1$, $0 \leq u' \leq \eta - u$ and $0 \leq c \leq w - 1$. Based on (30), $Q(x,y,zG_\Gamma)$ can be generated by

$$\begin{aligned} \widehat{\mathcal{M}}^{(u-1)} = \{&\widehat{M}^{(u-1)}_{u',c} = \delta_{u',0}\widehat{T}_{u-1,c} \\ &+ (1 - \delta_{u',0})\widehat{\theta}^{(u)}_{u'-1,c} \cdot (zG_\Gamma - K_{\underline{z}^{(u-1)}})\}, \end{aligned} \tag{55}$$

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

$$K_{\underline{z}^{(r)}}(x,y) = \sum_{j \in [n]} z^{(r)}_j L_{[n],j}, \tag{56}$$

Since $G \mid \prod_{a=0}^{q-1}(x - \sigma_a)^{v^{(0)}_{a,c}}$, $G^{(u-1)} \mid \prod_{a=0}^{q-1}(x - \sigma_a)^{v^{(u-1)}_{a,c}}$. Therefore, $G^{(u-1)} \mid \widehat{T}_{u-1,c}$, and subsequently $G_{\Gamma_{u-1}} \mid (zG_\Gamma - K_{\underline{z}^{(u-1)}})$. It remains to further prove $G^{(u)} \mid \widehat{\theta}^{(u)}_{u',c}$. For $u = \eta - 1$, this holds since $G^{(\eta-1)}$ is the common factor of polynomials in $\mathcal{I}_{\widehat{\mathbf{M}}^{(\eta-1)},l}$. Based on (30), this also holds for $u = 0, 1, ..., \eta - 2$ by induction. Therefore, $G^{(u)} \mid \widehat{\theta}^{(u)}_{u',c}$. □

Based on *Lemma 11*, the isomorphic module basis can be generated in a progressive manner as

$$\begin{aligned} \widetilde{\mathcal{M}}^{(u-1)} = \{&\widetilde{M}^{(u-1)}_{u',c} = \delta_{u',0}\widetilde{T}_{u-1,c} \\ &+ (1 - \delta_{u',0})\widetilde{\theta}^{(u)}_{u'-1,c} \cdot (zG_\Gamma - \widetilde{K}_{\underline{z}^{(u-1)}})\}, \end{aligned} \tag{57}$$

where

$$\widetilde{T}_{u-1,c}(x,y) = \frac{\widehat{T}_{u-1,c}(x,y)}{\prod_{u'=u-1}^{\eta} G_{\Gamma_{u'}}(x)} \tag{58}$$

and $\widetilde{\theta}^{(u)}_{u',c}$ is the isomorphic submodule basis polynomial obtained by the basis reduction in the $(\eta - u)$-th progressive iteration. Let $\widetilde{\mathcal{Q}}$ denote the minimum polynomial in Φ. It can be restored into $\widehat{\mathcal{Q}}$ by

$$\widehat{\mathcal{Q}}(x,y,z) = G^{(u)}(x) \cdot \widetilde{\mathcal{Q}}(x,y,\frac{z}{G_\Gamma(x)}). \tag{59}$$

Note that over the progressive submodule basis construction process, the common factor of the basis polynomials grows its degree as

$$G^{(u-1)} = G^{(u)} \cdot G_{\Gamma_{u-1}}. \tag{60}$$

Afterwards, the RCS algorithm will be applied to find the z-roots of $\widehat{\mathcal{Q}}$. The z-roots of $\widehat{\mathcal{Q}}$, denoted by $\widehat{f}$, are restored into f_{est} by

$$f_{\text{est}}(x,y) = \widehat{f}(x,y) + K_{\Gamma}(x,y). \tag{61}$$

Example 5: Given the same Hermitian code and the same multiplicity matrix $\mathbf{M}$ as in *Example 2*, since $w\lfloor\frac{k-g}{w}\rfloor = 4$, $\Gamma = \{4,5,6,7\}$. The re-encoding points are $(P_4,1)$, (P_5,α^2), $(P_6,1)$ and (P_7,α). Based on (38), the re-encoding polynomial is

$$K_{\Gamma}(x,y) = 1 + \alpha x + xy.$$

The re-encoding codeword is

$$\underline{h} = (1,1,1,0,1,\alpha^2,1,\alpha).$$

Based on this, $\mathbf{M}$ can be transformed into $\widehat{\mathbf{M}}$ as

$$\widehat{\mathbf{M}} = \begin{bmatrix} 0 & 2 & 0 & 0 & 4 & 3 & 4 & 4 \\ 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 4 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Let $\widehat{\mathbf{M}}^{(0)} = \widehat{\mathbf{M}}$ and $\widehat{\mathbf{M}}^{(4)} = \mathbf{0}$. Matrices $\widehat{\mathbf{M}}^{(1)}$ to $\widehat{\mathbf{M}}^{(3)}$ can be generated as

$$\widehat{\mathbf{M}}^{(1)} = \begin{bmatrix} 0 & 1 & 0 & 0 & 3 & 3 & 2 & 3 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 3 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\widehat{\mathbf{M}}^{(2)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 2 & 2 & 1 & 2 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\widehat{\mathbf{M}}^{(3)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

The progressive decoding can start with $\widehat{\mathbf{M}}^{(3)}$. Its corresponding submodule basis isomorphism is constituted by

$$\begin{aligned} \widetilde{M}^{(3)}_{0,0} &= \alpha^2 x + \alpha x^2 + x^3, \\ \widetilde{M}^{(3)}_{0,1} &= \alpha y + \alpha^2 x + xy + x^3, \\ \widetilde{M}^{(3)}_{1,0} &= \alpha^2 + \alpha^2 z + \alpha^2 y + x + xz + \alpha x^2 y, \\ \widetilde{M}^{(3)}_{1,1} &= \alpha^2 yz + xy + xyz + \alpha x^2 y + \alpha^2 x^3 + \alpha x^5. \end{aligned}$$

Their common factor is

$$G^{(3)} = \alpha + x.$$

After basis reduction and root-finding, the decoding fails to find the estimated message polynomial. We continue to construct $\widetilde{\mathcal{M}}^{(2)}$ as in (57), which is constituted by

$$\begin{aligned} \widetilde{M}^{(2)}_{0,0} &= \alpha^2 x^2 + x^3 + \alpha^2 x^4 + x^5, \\ \widetilde{M}^{(2)}_{0,1} &= xy + \alpha x^2 + x^2 y + \alpha^2 x^3 + x^4, \\ \widetilde{M}^{(2)}_{1,0} &= \widetilde{\theta}^{(3)}_{0,0} \cdot (zG_{\Gamma} - \widetilde{K}_{\underline{z}^{(2)}}), \\ \widetilde{M}^{(2)}_{1,1} &= \widetilde{\theta}^{(3)}_{0,1} \cdot (zG_{\Gamma} - \widetilde{K}_{\underline{z}^{(2)}}), \\ \widetilde{M}^{(2)}_{2,0} &= \widetilde{\theta}^{(3)}_{1,0} \cdot (zG_{\Gamma} - \widetilde{K}_{\underline{z}^{(2)}}), \\ \widetilde{M}^{(2)}_{2,1} &= \widetilde{\theta}^{(3)}_{1,1} \cdot (zG_{\Gamma} - \widetilde{K}_{\underline{z}^{(2)}}), \end{aligned}$$

where $\widetilde{\theta}^{(3)}_{u',c}$ denotes the reduced basis obtained from $\widetilde{\mathcal{M}}^{(3)}$. Again, the fully expanded forms of the polynomials are listed in Appendix. Their common factor becomes

$$\begin{aligned} G^{(2)} &= G^{(3)} G_{\Gamma_2} \\ &= 1 + x + x^2. \end{aligned}$$

The weighted degrees of the maximum basis polynomial in *Examples 4* and *5* are 17 and 11, respectively. This implies that the decoding complexity is further reduced. As a result, the decoding procedure requires 1797 additions and 1249 multiplications.

V. Decoding Complexity

This section analyzes the decoding complexity. In verifying our analysis, it is measure as the average finite field multiplications in a decoding event.

A. Theoretical Analysis

Since the PASD can terminate during its progressive iterations, we characterize its complexity at a certain iteration. Let ξ denote the number of progressive iterations. At the ξ-th progressive iteration, complexity of the progressive BR interpolation can be characterized by the following two lemmas, which addresses the cases without and with ReT, respectively.

Lemma 12: Without ReT, complexity of the basis construction and reduction are $O(\xi^6 w^2 n^2)$ and $O(\xi^5 w^2 n^3)$, respectively.

Proof: According to (30), the basis construction mainly consists of three steps: computing $T_{u-1,c}$, computing $K_{\underline{\omega}^{(u-1)}}$ and the multiplication to obtain $M^{(u-1)}_{u',c}$. First, we consider computing $T_{u-1,c}$. Based on (17), $\deg_x \prod_{a=0}^{q-1}(x-\sigma_a)^{v^{(u-1)}_{a,c}} \le \xi w^2$, $\deg_x \prod_{0\le b\le c-1}(y - B^{(u)}_{b,c}) \le \xi n$ and $\deg_y \prod_{0\le b\le c-1}(y - B^{(u)}_{b,c}) \le w-1$. Therefore, $T_{u-1,c}$ can be determined with a complexity of $O(\xi^2 n^2)$. For $K_{\underline{\omega}^{(u-1)}}$, the computation follows from (15). It involves multiplying each received symbol $\omega^{(u-1)}_j$ with the corresponding polynomial $L_{[n],j}$ and accumulating the resulting products. As $L_{[n],j}$ can be pre-computed, computing $K_{\underline{\omega}^{(u-1)}}$ exhibits a complexity of $O(n^2)$. Moreover, since $\theta^{(u)}_{u',c}$ is reduced from $M^{(u)}_{u',c}$, $\deg_x \theta^{(u)}_{u',c} \le \xi(n/w-1)$, $\deg_y \theta^{(u)}_{u',c} \le \xi(w-1)$ and

$\deg_z \theta_{u',c}^{(u)} \leq \xi - 1$. Hence, obtaining $M_{u',c}^{(u-1)}$ exhibits a complexity of $O(\xi^5 w n^2)$. Because there are $(\xi-1)w$ basis polynomials should be computed, the complexity of basis construction is $O(\xi^6 w^2 n^2)$.

We further consider the basis reduction complexity. With the MS algorithm, the basis reduction complexity is mainly determined by the weighted degree of the maximum polynomial in the module basis [12], denoted by γ. The basis reduction complexity can be bounded by $O(\xi^3 w^3 \gamma^2)$. Since $\gamma = O(\xi w n)$, which is defined by the term $\prod_{0\leq b\leq c-1}(y - B_{b,c}^{(u)})$, the basis reduction complexity is $O(\xi^5 w^2 n^3)$. □

Lemma 13*:* With the ReT, the complexity of basis construction and reduction are $O(\xi^6 w^2 n(n - \mathrm{card}(\Gamma)))$ and $O(\xi^5 w^2 n^3)$, respectively.

Proof: According to (57), the basis construction mainly consists of three steps: computing $\widetilde{T}_{u-1,c}$, computing $\widetilde{K}_{\underline{z}^{(u-1)}}$ and the multiplication to obtain $\widetilde{M}_{u',c}^{(u-1)}$. We first consider computing $\widetilde{T}_{u-1,c}$. Based on (52), $\deg_x \prod_{a=0}^{q-1}(x-\sigma_a)^{v_{a,c}^{(u-1)}} / G^{(u-1)} \leq \xi w^2 - \sum_{r=u-1}^{\eta} \mathrm{card}(\Gamma_r)/w$, $\deg_x \prod_{0\leq b\leq c-1}(y - B_{b,c}^{(u)}) \leq \xi w^3$ and $\deg_y \prod_{0\leq b\leq c-1}(y - B_{b,c}^{(u)}) \leq w-1$. Hence, $\widetilde{T}_{u',c}$ can be computed with a complexity of $O(\xi^2 n(n - \mathrm{card}(\Gamma)))$. Based on (53), $\deg_x \widetilde{K}_{\underline{z}^{(u-1)}} \leq w^2 - 1 - \mathrm{card}(\Gamma_{u-1})/w$ and $\deg_y \widetilde{K}_{\underline{z}^{(u-1)}} \leq w-1$. Computing $\widetilde{K}_{\underline{z}^{(u-1)}}$ yields a complexity of $O((n-\mathrm{card}(\Gamma))n)$. Since $\widetilde{\theta}_{u',c}^{(u)}$ is reduced from $\widetilde{M}_{u',c}^{(u)}$, $\deg_x \widetilde{\theta}_{u',c}^{(u)} \leq \xi((n - \mathrm{card}(\Gamma))/w - 1)$, $\deg_y \widetilde{\theta}_{u',c}^{(u)} \leq \xi(w-1)$ and $\deg_z \widetilde{\theta}_{u',c}^{(u)} \leq \xi - 1$. Therefore, obtaining $M_{u',c}^{(u-1)}$ exhibits a complexity of $O(\xi^5 w n(n - \mathrm{card}(\Gamma)))$. Since basis construction computes $(\xi-1)w$ polynomials, its complexity is $O(\xi^6 w^2 n(n - \mathrm{card}(\Gamma)))$. Since the ReT does not affect the term $\prod_{0\leq b\leq c-1}(y - B_{b,c}^{(u)})$, $\gamma = O(\xi w n)$. Therefore, the basis reduction complexity is $O(\xi^5 w^2 n^3)$. □

The prototype ASD constructs the interpolation module with respect to $\mathbf{M}$. Therefore, complexity of ASD is obtained by replacing the factor ξ in the above complexity formulation by η. Since the basis reduction complexity dominates the overall complexity and $\xi \leq \eta$, conducting the BR interpolation in a progressive manner reduces the complexity by a factor of $1 - (\xi/\eta)^5$. Note that at the ξ-th progressive iteration, complexity of root-finding is $O(\xi k n)$. Therefore, despite the PASD may perform multiple basis reductions and root-findings, this extra cost can be easily compensated by performing fewer progressive iterations. It is important to note that the theoretical complexity, which characterizes the worst-case scenario, remains unchanged by the ReT. However, practical simulations demonstrate that ReT yields a significant reduction in average complexity. This will be verified numerically as the follows.

B. Numerical Results

In the following, we present the numerical results of decoding complexity. They are obtained over the additive white Gaussian noise (AWGN) channel using binary phase shift keying (BPSK) modulation. The proposed PASD and that facilitated by the ReT are marked as PASD (BR) and PASD (ReT-BR), respectively. The ML criterion of [38, 39] is applied to assess the estimated codeword and further terminate the decoding when an ML codeword is found. The ReT-facilitated ASD of [26] (marked as ASD (ReT-BR)) and the prototype ASD of [24] (marked as ASD (BR)) are used as the comparison benchmarks.

Tables I and II show the ASD complexity at different signal-to-noise ratio (SNR) in decoding the (64, 49) and the (512, 409) Hermitian codes, respectively. They are defined in $\mathbb{F}_{16}$ and $\mathbb{F}_{64}$, respectively. In the two tables, the best performing algorithm is highlighted in grey for each set of designed maximum decoding OLS l and SNR. It can be seen that when the SNR is low, complexity of the ASD and PASD is similar, with PASD occasionally exhibiting higher complexity. This can be attributed to the extra computational cost introduced by multiple basis reductions and root-findings during the progressive iterations. As the channel conditions improve, all these four algorithms yield a reducing complexity. Among them, the progressive decoding algorithms yield a more significant complexity reduction than the non-progressive ones. This behavior aligns with the theoretical analysis. The average number of progressive iterations decreases as the SNR increases, which directly lowers the effective module size and hence the complexity, which is consistent with *Lemmas 12* and *13*. Note that since the multiplicity assignment follows Algorithm A of [22], the PASD rarely succeeds at the first progressive iteration. At high SNR, decoding typically succeeds at the second iteration. The two PASD algorithms converge to their respective complexity, regardless of the designed OLS l. Consequently, the complexity advantage of the proposed PASD over the ASD becomes more substantial if the ASD performs with a large OLS l. It is also observed that the PASD (ReT-BR) exhibits lower complexity than the PASD (BR), demonstrating the merit of ReT. The measured overall complexity exhibits a proportional saving due to the ReT, consistent with the theoretical analysis.

VI. Decoding Performance

This section shows decoding performance of the proposed algorithms. They are measured as frame error rate (FER) over the AWGN channel using BPSK modulation. For comparison, we also evaluate GS decoding with $m = l = 5$. Fig. 1 shows the performance of ASD and PASD of the (64, 49) Hermitian code, along with the GS decoding result. Note that the ReT does not affect the ASD or PASD performance [26]. It can be seen that the PASD preserves the decoding performance of the prototype ASD. Moreover, attributed to the fact that the PASD makes multiple attempts to find the intended message polynomial, it yields a slight performance gain. This is because the intermediate multiplicity matrices may lead to better multiplicity assignment than that of $\mathbf{M}$. In particular, this can be theoretically explained by cases where *Corollary 7* is satisfied while *Theorem 2* is not, which allows the PASD to succeed but the prototype ASD would fail.

TABLE I
ASD COMPLEXITY AT DIFFERENT SNR IN DECODING THE (64, 49) HERMITIAN CODE

SNR (dB)	ASD (BR)		ASD (ReT-BR)		PASD (BR)		PASD (ReT-BR)	
	$l=3$	$l=4$	$l=3$	$l=4$	$l=3$	$l=4$	$l=3$	$l=4$
4	1.81×10^{6}	5.44×10^{6}	1.12×10^{6}	2.63×10^{6}	2.14×10^{6}	5.45×10^{6}	$\mathbf{1.00\times10^{6}}$	$\mathbf{2.16\times10^{6}}$
5	1.32×10^{6}	3.58×10^{6}	6.89×10^{5}	1.57×10^{6}	6.68×10^{5}	1.52×10^{6}	$\mathbf{4.15\times10^{5}}$	$\mathbf{1.01\times10^{6}}$
6	1.25×10^{6}	3.12×10^{6}	4.76×10^{5}	1.18×10^{6}	1.59×10^{5}	3.54×10^{5}	$\mathbf{1.13\times10^{5}}$	$\mathbf{2.33\times10^{5}}$
7	8.16×10^{5}	1.85×10^{6}	4.49×10^{5}	7.82×10^{5}	1.10×10^{5}	1.25×10^{5}	$\mathbf{6.97\times10^{4}}$	$\mathbf{8.36\times10^{4}}$
8	5.25×10^{5}	1.03×10^{6}	2.09×10^{5}	4.03×10^{5}	9.28×10^{4}	9.59×10^{4}	$\mathbf{6.30\times10^{4}}$	$\mathbf{6.49\times10^{4}}$

TABLE II
ASD COMPLEXITY AT DIFFERENT SNR IN DECODING THE (512, 409) HERMITIAN CODE

SNR (dB)	ASD (BR)		ASD (ReT-BR)		PASD (BR)		PASD (ReT-BR)	
	$l=2$	$l=3$	$l=2$	$l=3$	$l=2$	$l=3$	$l=2$	$l=3$
4	2.01×10^{8}	6.32×10^{8}	$\mathbf{1.43\times10^{8}}$	4.48×10^{8}	1.98×10^{8}	5.50×10^{8}	1.77×10^{8}	$\mathbf{4.02\times10^{8}}$
5	9.80×10^{7}	3.21×10^{8}	$\mathbf{5.78\times10^{7}}$	$\mathbf{1.77\times10^{8}}$	1.23×10^{8}	4.49×10^{8}	9.04×10^{7}	2.61×10^{8}
6	9.85×10^{7}	3.12×10^{8}	5.25×10^{7}	1.58×10^{8}	2.75×10^{7}	6.04×10^{7}	$\mathbf{2.28\times10^{7}}$	$\mathbf{5.25\times10^{7}}$
7	9.28×10^{7}	2.92×10^{8}	4.74×10^{7}	1.32×10^{8}	2.66×10^{7}	2.69×10^{7}	$\mathbf{2.07\times10^{7}}$	$\mathbf{2.17\times10^{7}}$
8	8.73×10^{7}	2.68×10^{8}	4.44×10^{7}	1.11×10^{8}	2.54×10^{7}	2.62×10^{7}	$\mathbf{2.01\times10^{7}}$	$\mathbf{2.04\times10^{7}}$

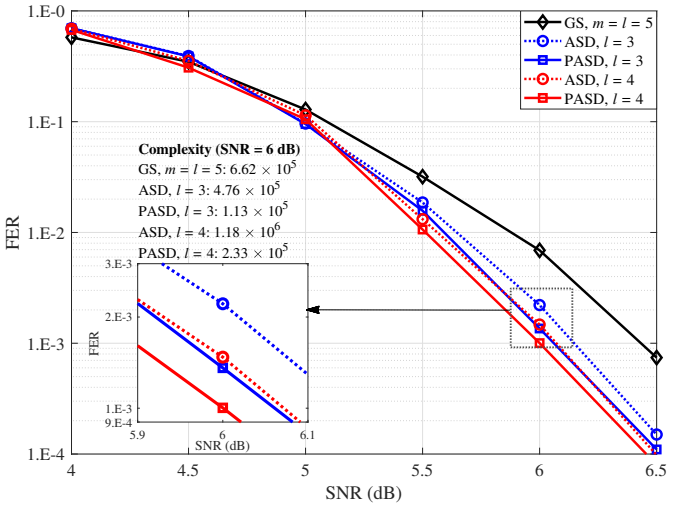

Fig. 1. Decoding performance comparison of ASD and PASD in decoding the (64, 49) Hermitian code.

It can also be observed that GS decoding exhibits a higher FER than ASD and PASD over the considered SNR range. Taking the SNR of 6 dB as an example, where the GS decoding ($m = l = 5$) has a decoding complexity that is similar to that of ASD and PASD, its FER is notably worse. This indicates that ASD and PASD not only yields a complexity advantage but also provide substantially enhanced error-correction capability than the GS decoding.

VII. CONCLUSION

This paper has proposed the PASD of Hermitian codes. The decoding has been formulated as computing the Gröbner basis that contains the interpolation polynomial, which allows the size of the interpolation submodule basis to be enlarged progressively. To further reduce complexity, the ReT has been introduced to facilitate the PASD, which simplifies the interpolation by extracting a common factor in the submodule basis polynomials. Comprehensive complexity analysis has shown that the proposed PASD adapts the decoding complexity to channel conditions, and the complexity reduction brought by the ReT has been verified. Moreover, simulation results have demonstrated that the proposed PASD preserves the error-correction performance of the prototype ASD. This work has provided an efficient and adaptive decoding solution for Hermitian codes, with potential applicability to other AG codes.

APPENDIX
NOTATION

The primary notation is summarized below, grouped by category.

Unless otherwise specified, the following conventions are adopted throughout the paper:

- Uppercase bold letters denote matrices.
- Uppercase italic letters generally refer to sets, spaces, or polynomials.
- Lowercase letters denote scalar variables.

- Lowercase underlined letters denote vectors.

Finite Fields

- q: Cardinality of the finite field.
- $\mathbb{F}_q$: The finite field with q elements, whose elements are denoted by $\sigma_0, \sigma_1, \ldots, \sigma_{q-1}$.
- $\mathbb{F}_q[X, Y]$: The ring of bivariate polynomials over $\mathbb{F}_q$.

Hermitian Curve

- w: The positive integer satisfying $w^2 = q$.
- H_w: The affine Hermitian curve over $\mathbb{F}_q$, defined by (1).
- g: The genus of H_w, given by $g = w(w-1)/2$.
- $P_j = (x_j, y_j)$: An affine point of H_w; there are w^3 such points.
- P_∞: The unique point at infinity of H_w.
- $\mathcal{P}$: The set of affine points of H_w.
- $\mathcal{R}$: The coordinate ring of H_w, defined as in (2).
- x, y: The residue classes of X and Y in $\mathcal{R}$, respectively.
- $\mathcal{L}_w$: The pole basis of $\mathcal{R}$ with respect to P_∞.
- $\phi_a(x, y)$: An element of $\mathcal{L}_w$, of the form $x^{i_x} y^{i_y}$ for integers $0 \le i_x \le w$ and $i_y \ge 0$.
- $\nu_P(S)$: The order (valuation) of a nonzero polynomial $S \in \mathcal{R}$ at a rational point P.
- $\psi_{P,v}(x, y)$: The zero basis element at a rational point P, parameterized by $v = \lambda + (w+1)\varkappa$ for nonnegative integers λ, $\varkappa$, and defined as in (3).

Encoding

- n: Length of the code. For Hermitian codes, $n = w^3$.
- k: Dimension of the code.
- $\mathcal{L}(\mu P_\infty)$: The Riemann-Roch space used to define Hermitian codes, where $\mu = k + g - 1$.
- $\underline{f} = (f_0, f_1, \ldots, f_{k-1})$: A message vector in $\mathbb{F}_q^k$ to be encoded.
- $f(x, y)$: The message polynomial corresponding to $\underline{f}$, expressed as in (4).
- $\underline{c} = (c_0, c_1, \ldots, c_{n-1})$: The codeword obtained by evaluating $f(x, y)$ as in (5).

GS Decoding

- $\mathcal{R}[z]$: The polynomial ring in variable z over the coordinate ring $\mathcal{R}$.
- $\phi_a z^b$: A monomial in $\mathcal{R}[z]$, where $\phi_a \in \mathcal{L}_w$ and $b \ge 0$.
- $\deg_{1,w_z}(\cdot)$: The $(1, w_z)$-weighted degree defined as in (7).
- w_z: The weight parameter for the variable z.
- m: Multiplicity.

ASD

- l: Designed maximum decoding OLS.
- $\underline{r} = (r_0, r_1, \ldots, r_{n-1})$: The channel output.
- $\mathbf{\Pi}$: The $q \times n$ reliability matrix with entries $\pi_{ij} = \Pr(r_j \mid c_j = \sigma_i)$.
- $\mathbf{M}$: The $q \times n$ multiplicity matrix with nonnegative integer entries m_{ij}.
- i_j: The row index satisfying $\sigma_{i_j} = \omega_j$, where ω_j is the hard-decision symbol at position j.
- $s_{\mathbf{M}}(\underline{c})$: The codeword score of $\underline{c}$ with respect to $\mathbf{M}$ defined as in (10).
- $\mathrm{mult}_{(P_j, \sigma_i)}(Q)$: The multiplicity of a polynomial Q at the interpolation point (P_j, σ_i).
- $\mathcal{I}_{\mathbf{M}}$: The ideal determined by the multiplicity matrix $\mathbf{M}$, defined as in (11).
- $\mathcal{R}[z]_l$: The set of polynomials in $\mathcal{R}[z]$ with z-degree at most l.
- $\mathcal{I}_{\mathbf{M},l}$: The interpolation module defined as in (12).

BR Interpolation

- η: The maximum column sum of the multiplicity matrix $\mathbf{M}$, defined as $\eta = \max_j \sum_{i=0}^{q-1} m_{ij}$.
- $\mathbf{M}^{(u)}$: The u-th intermediate multiplicity matrix with entries $m_{ij}^{(u)}$.
- $i_j^{(u)}$: The row index that attains the maximum entry in column j of $\mathbf{M}^{(u)}$, i.e., $i_j^{(u)} = \arg\max_i \{m_{ij}^{(u)}\}$.
- $\mathbb{A}(\mathcal{J})$: The set of distinct x-coordinates from the points $\{P_j : j \in \mathcal{J}\}$.
- $\mathbb{B}_\sigma(\mathcal{J})$: For a given x-coordinate σ, the set of corresponding y-coordinates from points in $\{P_j : j \in \mathcal{J}\}$.
- $L_{\mathcal{J},j}(x, y)$: The Lagrange interpolation polynomial defined as in (14).
- $K_{\underline{\omega}^{(u)}}(x, y)$: A polynomial defined as in (15).
- $\mathfrak{m}_j^{(u)}$: The maximum entry in column j of $\mathbf{M}^{(u)}$, i.e., $\mathfrak{m}_j^{(u)} = \max_i \{m_{ij}^{(u)}\}$.
- $\mathcal{E}_{\mathbf{M}^{(u)}}$: The ideal in $\mathcal{R}$ defined by $\mathfrak{m}_j^{(u)}$, i.e., $\{S \in \mathcal{R} : \nu_{P_j}(S) \ge \mathfrak{m}_j^{(u)},\ j \in [n]\}$.
- $v_{a,b}^{(u)}$: Reassigned multiplicity, such that $v_{a,b}^{(u)} = \mathfrak{m}_j^{(u)}$ for the corresponding index j with $P_j \mapsto P_{a,b}$.
- $B_{b,c}^{(u)}$: A polynomial in $\mathbb{F}_q[x]$ satisfying $\nu_{P_{a,b}}(y - B_{b,c}^{(u)}) \ge v_{a,b}^{(u)} - v_{a,c}^{(u)}$ for relevant indices a, b, c.
- $T_{u,c}(x, y)$: A generator polynomial for the ideal $\mathcal{E}_{\mathbf{M}^{(u)}}$, defined as in (17).
- $\mathcal{M} = \{M_{u,c}\}$: The module basis as defined in (18).
- $\mathcal{I}_{\mathbf{M}^{(u)},l}$: The interpolation submodule determined by $\mathbf{M}^{(u)}$.

PASD

- $\delta_{a,b}$: The Kronecker delta function, defined as in (20).
- $\mathbf{M}'$: An auxiliary multiplicity matrix defined by (21), used to illustrate the progressive interpolation process.
- $\mathcal{M}^{(u)} = \{M_{u',c}^{(u)}\}$: The submodule basis as defined in (29).
- $\Theta^{(u)} = \{\theta_{u',c}^{(u)}\}$: The reduced Gröbner basis derived from $\mathcal{M}^{(u)}$.
- $\zeta_c^{(s)}$: A polynomial over $\mathbb{F}_q$ introduced in the proof of *Lemma 8*.

ReT-Facilitated PASD

- Γ: The index set selecting the points used for re-encoding.
- $\boldsymbol{P}_\Gamma$: The set of re-encoding points defined as $\boldsymbol{P}_\Gamma = \{(P_j, \omega_j^{(0)}) : j \in \Gamma\}$.
- $K_\Gamma(x, y)$: The re-encoding polynomial defined as in (38).
- $\mathbb{S}(\mathcal{J})$: The set of indices $j \in \mathcal{J}$ for which $\mathrm{card}(\mathbb{B}_{x_j}(\mathcal{J})) = w$.
- $\underline{h}$: The re-encoded codeword defined as in (39).
- $\widehat{\mathbf{M}}$: The $q \times n$ multiplicity matrix (from $\mathbf{M}$ via (40), (41)), with entries $\widehat{m}_{ij} \in \mathbb{Z}_0^+$.

- $\widehat{\mathbf{M}}^{(u)}$: The u-th intermediate multiplicity matrix with respect to $\widehat{\mathbf{M}}$, whose entries are $\widehat{m}_{ij}^{(u)}$.
- Γ_u: A subset of the re-encoding index set Γ defined as in (46).
- $G(x), G_{\Gamma}(x), G_{\Gamma_u}(x), G_{\Gamma\setminus\Gamma_u}(x)$: The common factor polynomials defined as in (44), (45), (47) and (48), respectively.
- $\mathcal{I}_{\widehat{\mathbf{M}},l}$: The interpolation module with respect to $\widehat{\mathbf{M}}$.
- Φ: The isomorphic module with respect to $\mathcal{I}_{\widehat{\mathbf{M}},l}$.
- $\widehat{\mathfrak{m}}_j^{(u)}$: The maximum entry in column j of $\widehat{\mathbf{M}}^{(u)}$, i.e., $\widehat{\mathfrak{m}}_j^{(u)} = \max_i\{\widehat{m}_{ij}^{(u)}\}$.
- $\mathcal{E}_{\widehat{\mathbf{M}}^{(u)}}$: The ideal in $\mathcal{R}$ defined by $\widehat{\mathfrak{m}}_j^{(u)}$, i.e., $\{S \in \mathcal{R} : \nu_{P_j}(S) \geq \widehat{\mathfrak{m}}_j^{(u)}, \forall j\}$.
- $\widehat{i}_j^{(u)}$: The row index of the maximum entry in column j of $\widehat{\mathbf{M}}^{(u)}$, i.e., $\widehat{i}_j^{(u)} = \arg\max_i\{\widehat{m}_{ij}^{(u)}\}$.
- $z_j^{(u)}$: The finite field element corresponding to $\widehat{i}_j^{(u)}$, defined as $z_j^{(u)} = \sigma_{\widehat{i}_j^{(u)}}$.
- $\underline{z}^{(u)}$: The vector $(z_0^{(u)}, z_1^{(u)}, \ldots, z_{n-1}^{(u)})$.
- $\widetilde{T}_{u,c}(x,y)$: A polynomial defined as in (52).
- $\widetilde{K}_{\underline{z}^{(r)}}$: A polynomial defined as in (53).
- $\widetilde{\mathcal{M}} = \{\widetilde{M}_{u,c}\}$: The module basis isomorphism defined as in (51).
- $\mathcal{I}_{\widehat{\mathbf{M}}^{(u)},l}$: The interpolation submodule with respect to $\widehat{\mathbf{M}}^{(u)}$.
- $K_{\underline{z}^{(u)}}(x,y)$: A polynomial defined as in (56).
- $\widehat{T}_{u,c}$: A basis polynomial for the ideal $\mathcal{E}_{\widehat{\mathbf{M}}^{(u)}}$.
- $\widehat{\mathcal{M}}^{(u)} = \{\widehat{M}_{u',c}^{(u)}\}$: The isomorphic submodule basis as defined as in (55).
- $\widetilde{\theta}_{u',c}^{(u)}$: The isomorphic submodule basis polynomial obtained by the basis reduction in the $(\eta-u)$-th progressive iteration.
- $\widehat{\mathcal{Q}}$ and $\widetilde{\mathcal{Q}}$: The minimum polynomial in $\mathcal{I}_{\widehat{\mathbf{M}}^{(u)},l}$ and Φ, respectively.

Others

- $[a]$: For a nonnegative integer a, the set $\{0,1,\ldots,a-1\}$.
- $f_{\text{est}}(x,y)$: The estimated message polynomial.
- γ: The weighted degree of the maximum polynomial in the module basis.

EXAMPLE DETAILS

A. Expanded Module Basis for Example 2

The module basis $\mathcal{M}$ for *Example 2* is given by the following polynomials:

$$\begin{aligned}
M_{0,0} &= x^4 + x^{16},\\
M_{0,1} &= \alpha x^2 y + x^3 y + x^5 + \alpha x^5 y + \alpha^2 x^6\\
&\quad + x^6 y + x^8 + \alpha x^8 y + \alpha^2 x^9 + x^9 y\\
&\quad + x^{11} + \alpha x^{11} y + \alpha^2 x^{12} + x^{12} y + x^{14}\\
&\quad + \alpha^2 x^{15},\\
M_{1,0} &= x^3 z + x^3 y + x^4 + x^4 y + \alpha^2 x^5\\
&\quad + \alpha x^6 + x^6 z + x^7 + x^7 y + \alpha^2 x^8\\
&\quad + \alpha x^9 + x^9 z + x^{10} + x^{10} y + \alpha^2 x^{11}\\
&\quad + \alpha x^{12} + x^{12} z + x^{13} + x^{13} y + \alpha^2 x^{14}\\
&\quad + \alpha x^{15} + x^{15} y,\\
M_{1,1} &= \alpha x y + \alpha x y z + x^2 y + x^2 y z + x^3 y\\
&\quad + \alpha x^4 + x^4 z + \alpha^2 x^4 y + \alpha x^5 + \alpha^2 x^5 z\\
&\quad + x^5 y + x^6 + \alpha^2 x^6 y + \alpha x^7 + \alpha^2 x^7 y\\
&\quad + \alpha x^7 y z + \alpha x^8 y + x^8 y z + x^9 y + \alpha x^{10}\\
&\quad + x^{10} z + \alpha^2 x^{10} y + \alpha x^{11} + \alpha^2 x^{11} z + x^{11} y\\
&\quad + x^{12} + \alpha^2 x^{12} y + \alpha x^{13} + x^{13} y + \alpha^2 x^{14} y,\\
M_{2,0} &= x^2 z^2 + x^2 y + x^4 + x^4 y + x^5\\
&\quad + \alpha x^6 + x^7 + \alpha^2 x^8 + x^8 z^2 + x^{10}\\
&\quad + x^{10} y + \alpha x^{12} + x^{13} + \alpha^2 x^{14} + x^{14} y\\
&\quad + x^{17},\\
M_{2,1} &= \alpha x y + \alpha x y z^2 + \alpha^2 x^2 z^2 + \alpha x^2 y + x^2 y z^2\\
&\quad + x^4 + x^4 z^2 + \alpha x^4 y + \alpha x^4 y z^2 + \alpha x^5\\
&\quad + \alpha^2 x^5 z^2 + x^5 y z^2 + \alpha x^6 + x^6 y + \alpha x^7\\
&\quad + x^7 z^2 + \alpha x^7 y + \alpha x^8 + \alpha^2 x^8 y + \alpha^2 x^9\\
&\quad + x^9 y + \alpha x^{10} + x^{10} y + \alpha x^{11} + x^{12}\\
&\quad + \alpha^2 x^{13} y + \alpha x^{14} + x^{14} y + x^{16},\\
M_{3,0} &= x z^3 + \alpha^2 x y + x y z + \alpha^2 x y z^2 + x^2 z^2\\
&\quad + x^2 y z^2 + x^3 z + \alpha^2 x^3 z^2 + \alpha^2 x^3 y + x^3 y z\\
&\quad + \alpha x^4 + x^4 z + \alpha x^4 z^2 + x^4 z^3 + x^4 y z\\
&\quad + \alpha^2 x^5 + \alpha x^5 z + x^5 z^2 + \alpha^2 x^5 y + x^5 y z^2\\
&\quad + \alpha^2 x^6 z^2 + x^6 y z + \alpha x^7 + \alpha x^7 z + \alpha x^7 z^2\\
&\quad + x^7 y z + \alpha^2 x^7 y z^2 + \alpha x^8 z + x^8 y + \alpha^2 x^9\\
&\quad + x^9 z + \alpha^2 x^9 y + \alpha x^{10} + \alpha x^{10} z + \alpha^2 x^{10} y\\
&\quad + x^{10} y z + \alpha^2 x^{11} + \alpha^2 x^{11} y + \alpha^2 x^{13} + x^{13} z\\
&\quad + \alpha^2 x^{13} y + x^{14} y + \alpha^2 x^{15} + x^{16} + \alpha^2 x^{16} y,\\
M_{3,1} &= y + \alpha y z + y z^2 + \alpha y z^3 + \alpha^2 x z^3\\
&\quad + x y + \alpha x y z + x y z^2 + x y z^3 + \alpha^2 x^2 z^2\\
&\quad + x^2 y + \alpha x^2 y z^2 + x^3 + x^3 z + \alpha^2 x^3 z^2\\
&\quad + x^3 z^3 + \alpha x^3 y z^2 + \alpha x^4 + \alpha x^4 z + x^4 z^2\\
&\quad + \alpha^2 x^4 y + \alpha x^4 y z + \alpha x^4 y z^2 + \alpha^2 x^5 + \alpha x^5 z\\
&\quad + \alpha x^5 z^2 + x^5 y z + \alpha^2 x^6 z + \alpha^2 x^6 z^2 + \alpha^2 x^6 y\\
&\quad + \alpha x^6 y z + \alpha^2 x^6 y z^2 + x^7 + \alpha^2 x^7 z^2 + x^7 y z\\
&\quad + x^8 z + \alpha x^8 y + x^9 z + \alpha^2 x^9 y + \alpha^2 x^9 y z\\
&\quad + \alpha x^{10} + \alpha x^{10} z + \alpha x^{10} y + x^{10} y z + x^{11}\\
&\quad + \alpha^2 x^{11} y + x^{12} z + x^{13} + x^{13} y + \alpha x^{14}\\
&\quad + \alpha^2 x^{15} y + \alpha^2 x^{16},
\end{aligned}$$

$$
\begin{aligned}
M_{4,0} = {} & z^4 + \alpha^2 yz + yz^2 + \alpha^2 yz^3 + \alpha^2 x^2 z^3 \\
& + \alpha^2 x^2 y + \alpha x^2 yz + \alpha^2 x^2 yz^2 + x^2 yz^3 + \alpha^2 x^3 z \\
& + \alpha x^3 z^2 + \alpha x^3 z^3 + \alpha x^3 y + x^3 yz + x^3 yz^2 \\
& + x^4 + \alpha^2 x^4 z + \alpha^2 x^4 z^2 + x^4 y + \alpha x^4 yz \\
& + x^4 yz^2 + \alpha^2 x^5 z + \alpha^2 x^5 z^2 + \alpha^2 x^5 yz^2 + x^6 \\
& + \alpha^2 x^6 z + \alpha^2 x^6 y + \alpha^2 x^6 yz + \alpha x^6 yz^2 + x^7 z \\
& + \alpha^2 x^7 y + x^7 yz + x^8 + \alpha^2 x^8 z^2 + \alpha x^8 y \\
& + x^8 yz + x^9 + \alpha x^9 z + \alpha^2 x^9 z^2 + x^9 yz \\
& + \alpha x^{10} y + \alpha^2 x^{11} + \alpha x^{11} z + x^{11} yz + \alpha^2 x^{12} \\
& + \alpha x^{12} z + \alpha^2 x^{12} y + x^{13} y + x^{14} + x^{15} \\
& + x^{15} y + \alpha^2 x^{17} + \alpha x^{18}, \\
M_{4,1} = {} & \alpha^2 yz + yz^2 + \alpha^2 yz^3 + yz^4 + \alpha^2 x^2 y \\
& + \alpha x^2 yz + \alpha^2 x^2 yz^2 + \alpha x^2 yz^3 + \alpha^2 x^3 z + x^3 z^2 \\
& + \alpha^2 x^3 z^3 + \alpha x^3 y + \alpha x^3 yz + \alpha^2 x^3 yz^2 \\
& + \alpha x^3 yz^3 + x^4 yz + \alpha x^4 yz^2 + \alpha^2 x^5 + \alpha x^5 z \\
& + \alpha^2 x^5 z^2 + x^5 z^3 + \alpha^2 x^5 yz + \alpha x^6 + x^6 z \\
& + x^6 z^2 + \alpha x^6 y + \alpha x^6 yz^2 + x^7 + \alpha x^7 z \\
& + x^7 z^2 + \alpha^2 x^7 y + \alpha^2 x^8 z^2 + \alpha^2 x^8 y + x^8 yz \\
& + \alpha^2 x^8 yz^2 + \alpha^2 x^9 + \alpha^2 x^9 z + \alpha x^9 z^2 + x^9 y \\
& + \alpha^2 x^9 yz + \alpha^2 x^9 yz^2 + \alpha^2 x^{10} + x^{10} z + \alpha x^{10} y \\
& + \alpha x^{11} + x^{11} z + \alpha^2 x^{11} y + \alpha^2 x^{11} yz + x^{12} z \\
& + \alpha x^{12} yz + \alpha x^{13} + x^{13} y + x^{14} z + x^{14} y \\
& + \alpha^2 x^{15} + x^{16} + \alpha^2 x^{17} y + x^{18} + \alpha x^{18} y.
\end{aligned}
$$

B. Expanded Module Basis for Example 3

The following polynomials are the fully expanded forms of the basis elements $M^{(2)}_{u',c}$.

$$
\begin{aligned}
M^{(2)}_{1,0} = {} & xz + \alpha^2 xy + x^2 + x^2 y + \alpha^2 x^3 \\
& + \alpha x^4 + x^4 z + x^5 + x^5 y + \alpha^2 x^6 \\
& + \alpha x^7 + \alpha^2 x^7 y, \\
M^{(2)}_{1,1} = {} & y + \alpha yz + \alpha^2 xz + xy + xyz \\
& + \alpha^2 x^2 + \alpha x^2 y + \alpha^2 x^3 + x^3 z + \alpha x^3 y \\
& + x^4 + \alpha x^4 y + \alpha x^5 + \alpha^2 x^6 + \alpha^2 x^6 y \\
& + \alpha^2 x^7, \\
M^{(2)}_{2,0} = {} & z^2 + \alpha^2 yz + x^2 + \alpha^2 x^2 z + \alpha x^2 y \\
& + x^2 yz + \alpha^2 x^3 + \alpha x^3 z + x^3 y + x^4 \\
& + x^4 y + \alpha x^5 + \alpha^2 x^5 y + \alpha^2 x^6 + \alpha^2 x^6 y \\
& + \alpha^2 x^8 + \alpha x^9, \\
M^{(2)}_{2,1} = {} & \alpha^2 yz + yz^2 + \alpha^2 x^2 y + \alpha x^2 yz + \alpha^2 x^3 z \\
& + \alpha x^3 y + \alpha x^3 yz + \alpha x^5 + x^5 z + x^5 y \\
& + x^6 + x^7 + \alpha^2 x^8 + \alpha^2 x^8 y + \alpha^2 x^9 \\
& + \alpha x^9 y.
\end{aligned}
$$

C. Expanded Module Basis for Example 4

The following polynomials are the fully expanded forms of the basis elements $M^{(2)}_{u',c}$.

$$
\begin{aligned}
M^{(2)}_{1,0} = {} & \alpha xy + \alpha^2 x^2 + x^2 y + x^4 + \alpha x^4 y \\
& + \alpha^2 x^5 + x^5 y + x^7, \\
M^{(2)}_{1,1} = {} & xz + \alpha xy + \alpha xyz + x^2 + \alpha^2 x^2 z \\
& + x^2 yz + \alpha x^3 y + x^4 + \alpha x^4 y + \alpha^2 x^5 y \\
& + x^6 + x^7 + \alpha^2 x^8, \\
M^{(2)}_{2,0} = {} & z^2 + \alpha y + xz + \alpha xy + \alpha xyz \\
& + x^3 + \alpha x^3 z + \alpha x^3 y + \alpha^2 x^3 yz + \alpha^2 x^4 \\
& + \alpha^2 x^4 y + \alpha x^5 + \alpha x^5 y + \alpha x^6 + \alpha x^7, \\
M^{(2)}_{2,1} = {} & \alpha^2 yz + yz^2 + \alpha x^2 z + x^2 y + \alpha^2 x^2 yz \\
& + \alpha x^3 + \alpha x^3 z + x^3 y + x^3 yz + \alpha x^4 y \\
& + x^5 + x^6 + \alpha^2 x^6 z + \alpha^2 x^6 y + \alpha x^7 \\
& + \alpha^2 x^7 y,
\end{aligned}
$$

D. Expanded Module Basis for Example 5

The following polynomials are the fully expanded forms of the basis elements $\widetilde{M}^{(2)}_{u',c}$.

$$
\begin{aligned}
\widetilde{M}^{(2)}_{1,0} = {} & \alpha^2 x + \alpha^2 xz + xyz + x^2 + \alpha^2 x^2 y \\
& + \alpha^2 x^4 + \alpha^2 x^5, \\
\widetilde{M}^{(2)}_{1,1} = {} & \alpha y + yz + \alpha x + \alpha xz + \alpha x^2 \\
& + x^2 z + \alpha x^2 y + x^3 + \alpha^2 x^3 y + \alpha^2 x^4, \\
\widetilde{M}^{(2)}_{2,0} = {} & \alpha^2 + \alpha^2 z^2 + y + \alpha x + \alpha^2 xz \\
& + xz^2 + \alpha xy + xyz + x^2 + \alpha x^2 z \\
& + \alpha x^2 y + \alpha^2 x^2 yz + x^3 + x^4, \\
\widetilde{M}^{(2)}_{2,1} = {} & yz + \alpha^2 yz^2 + \alpha xy + xyz^2 + \alpha^2 x^2 \\
& + \alpha^2 x^2 z + \alpha^2 x^2 y + x^2 yz + \alpha x^3 + x^3 z \\
& + \alpha x^4 + x^4 z + \alpha x^4 y + \alpha^2 x^5 + \alpha^2 x^5 z.
\end{aligned}
$$

REFERENCES

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

[2] B. Shen, "A Justesen construction of binary concatenated codes that asymptotically meet the Zyablov bound for low rate," *IEEE Trans. Inform. Theory*, vol. 39, no. 1, pp. 239–242, Jan. 1993.

[3] K. W. Shum, I. Aleshnikov, P. V. Kumar, H. Stichtenoth, and V. Deolalikar, "A low-complexity algorithm for the construction of algebraic-geometric codes better than the Gilbert-Varshamov bound," *IEEE Trans. Inform. Theory*, vol. 47, no. 6, pp. 2225–2241, Sept. 2001.

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

[5] 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.

[6] 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.
[7] R. Kötter, *On Algebraic Decoding of Algebraic-geometric and Cyclic Codes*. Ph. D Thesis, Univ. Linköping, Linköping, Sweden, 1996.
[8] 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.
[9] 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.
[10] P. Beelen and K. Brander, "Efficient list decoding of a class of algebraic-geometry codes," *Adv. Math. Commun.*, vol. 4, no. 4, pp. 485-518, Nov. 2010.
[11] 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.
[12] 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.
[13] P. Beelen, J. Rosenkilde, and G. Solomatov, "Fast decoding of AG codes," *IEEE Trans. Inform. Theory*, vol. 68, no. 11, pp. 7215-7232, Nov. 2022.
[14] P. Beelen and V. Neiger, "Faster list decoding of AG codes," *IEEE Trans. Inform. Theory*, vol. 71, no. 5, pp. 3397-3408, May 2025.
[15] 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.
[16] 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. Inform. Theory (ISIT)*, Taipei, Taiwan, Jun. 2023, pp. 19-24.
[17] 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.
[18] Y. Wan, J. Liang, L. Chen, and F. Zhang, "Low-complexity Chase decoding of elliptic codes," *IEEE Trans. Commun.*, vol. 73, no. 10, pp. 8574-8586, Oct. 2025.
[19] J. Zhao and L. Chen, "Algebraic Chase decoding of elliptic codes with generalized re-encoding transform and improved root-finding", *IEEE Trans. Inform. Theory*, vol. 71, no. 7, pp. 5089-5108, Jul. 2025.
[20] S. Wu, L. Chen, and M. Johnson, "Interpolation-based low-complexity Chase decoding algorithms for Hermitian codes," *IEEE Trans. Commun.*, vol. 66, no. 4, pp. 1376–1385, Apr. 2018.
[21] J. Liang, J. Zhao, and L. Chen, "Low-complexity Chase decoding of Hermitian codes with improved interpolation and root-finding," *IEEE Trans. Commun.*, vol. 73, no. 8, pp. 5509–5522, Aug. 2025.
[22] 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.
[23] 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.
[24] 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.
[25] 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.
[26] J. Liang, and L. Chen, "Low-complexity algebraic soft decoding of Hermitian codes with re-encoding transform," *IEEE Trans. Inform. Theory*, Early Access, 2026.
[27] S. Tang, L. Chen and X. Ma, "Progressive list-enlarged algebraic soft decoding of Reed-Solomon codes", *IEEE Commun. Lett.*, vol. 16, no. 6, pp. 901-904, Jun. 2012.
[28] L. Chen, S. Tang and X. Ma, "Progressive algebraic soft-decision decoding of Reed-Solomon codes", *IEEE Trans. Commun.*, vol. 61, no. 2, pp. 433-442, Feb. 2013.
[29] Xing, L. Chen, M. Bossert, "Progressive algebraic soft-decision decoding of Reed-Solomon codes using module minimization", *IEEE Trans. Commun.*, vol. 67, no. 11, pp. 7379-7391, Nov. 2019.
[30] J. Liang, and L. Chen, "Progressive algebraic soft decoding of Hermitian codes," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, Guangzhou, China, Jun. 2026.
[31] I. Blake, C Heegard, and T. Høholdt, "Algebraic-geometry codes," *IEEE Trans. Inform. Theory*, vol. 44, no. 6, pp. 2596–2618, Oct. 1998.
[32] R. Nielsen, *List Decoding of Linear Block Codes*. Ph. D Thesis, Tech. Univ. Denmark, Kongens Lyngby, Denmark, 2001.
[33] H. Stichtenoth, *Algebraic Function Fields and Codes*, 2nd ed. New York, NY, USA: Springer-Verlag, 2009.
[34] T. Mulders and A. Storjohann, "On lattice reduction for polynomial matrices," *J. Symb. Comput.*, vol. 35, no. 4, pp. 377–401, Apr. 2003.
[35] 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.
[36] 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.
[37] L. Chen, R. Carrasco, M. Johnston and E. Chester, "Efficient factorisation algorithm for list decoding algebraic-geometric and Reed-Solomon codes," in *Proc. IEEE International Conference of Communications (ICC)*, Glasgow, UK, 2007, pp. 851-856.
[38] T. Kaneko, T. Nishijima, H. Inazumi, and S. Hirasawa, "An efficient maximum-likelihood-decoding algorithm for linear block codes with algebraic decoder," *IEEE Trans. Inform. Theory*, vol. 40, no. 2, pp. 320–327, Mar. 1994.
[39] X. Ma, and S. Tang, "Correction to "an efficient maximum-likelihood-decoding algorithm for linear block codes with algebraic decoder" [Mar 94 320-327]," *IEEE Trans. Inform. Theory*, vol. 58, no. 6, pp. 4073, Jun. 2012.
[40] M. Safieh, D. N. Bailon and J. Freudenberger, "An acceptance criterion for hybrid algebraic and soft-input decoding," in *Proc. Int. Conf. Inform. Technol. (IT)*, Zabljak, Montenegro, 2020, pp. 1-5.

Jiwei Liang (Student Member, IEEE) received the B.E. degree in electronic engineering from Southeast University, Nanjing, China, in 2013, and the M.S. degree in communication engineering from Guilin University of Electronic Technology, Guilin, China, in 2017. He is currently pursuing the Ph.D. degree in information and communication engineering in Sun Yat-sen University, Guangzhou, China. In 2013, he was with Avonaco Communication Systems Company Ltd., Suzhou, China, as a DSP engineer, where he was involved in the development of multimedia communication systems. From 2017 to 2021, he was with Allwinner Technology Company Ltd., Zhuhai, China, as a firmware engineer, where he was involved in the development of WLAN/bluetooth combo chip. His research interests include channel coding and data communication.

Li Chen (Senior Member, IEEE) received the B.Sc. degree in applied physics from Jinan University, China, in 2003, and the M.Sc. degree in communications and signal processing and the Ph.D. degree in communications engineering from Newcastle University, U.K., in 2004 and 2008, respectively. From 2007 to 2010, he was a Research Associate with Newcastle University.

In 2010, he returned China, as a Lecturer with the School of Information Science and Technology, Sun Yat-sen University, Guangzhou. From 2011 to 2012, he was a Visiting Researcher with the Institute of Network Coding, The Chinese University of Hong Kong, where he was an Associate Professor and a Professor from 2011 and 2016. Since 2013, he has been the Associate Head of the Department of Electronic and Communication Engineering (ECE). From July 2015 to October 2015, he was a Visitor with the Institute of Communications Engineering, Ulm University, Germany. From October 2015 to June 2016, he was a Visiting Associate Professor with the Department of Electrical Engineering, University of Notre Dame, USA. From 2017 to 2020, he was the Deputy Dean of the School of Electronics and Communication Engineering. His research interests include information theory, error-correction codes, and data communications. He is a Senior Member of Chinese Institute of Electronics (CIE). He is a member of the IEEE Information Theory Society Board of Governors and its External Nomination Committee and the Chair of its Conference Committee. He is also the Chair of the IEEE Information Theory Society Guangzhou Chapter. He has been organizing several international conferences and workshops, including the 2018 IEEE Information Theory Workshop (ITW) at Guangzhou and the 2022 IEEE East Asian School of Information Theory (EASIT) at Shenzhen, for which he is the General Co-Chair. He is also the TPC Co-Chair of the 2022 IEEE/CIC International Conference on Communications in China (ICCC) at Foshan. He was an Associate Editor of IEEE TRANSACTIONS ON COMMUNICATIONS and is currently an Associate Editor of IEEE TRANSACTIONS ON INFORMATION THEORY.