Progressive Module Minimization for Re-encoding Transformed Soft Decoding of RS Codes

Jiongyue Xing †, Li Chen ‡, Martin Bossert §
† School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou, China
‡ School of Electronics and Communication Engineering, Sun Yat-sen University, Guangzhou, China
§ Institute of Communications Engineering, Ulm University, Ulm, Germany
Email: xingjyue@mail2.sysu.edu.cn, chenli55@mail.sysu.edu.cn, martin.bossert@uni-ulm.de

***Abstract*—The interpolation based algebraic decoding for Reed-Solomon (RS) codes can correct errors beyond half of the code's minimum Hamming distance through constructing a minimum polynomial $Q(x, y)$ and finding its y-roots. The progressive algebraic soft decoding (PASD) constructs $Q(x, y)$ with a progressively enlarged y-degree and terminates once the message is decoded, adapting the decoding capability and computation to the channel. This paper proposes the re-encoding transformed PASD algorithm, in which $Q(x, y)$ is progressively constructed by the low-complexity module minimization (MM) technique. Re-encoding transform (ReT) results in a common divisor for polynomials of the image of the submodule basis. It can be removed, leading to a simpler image expansion and reduction. Consequently, $Q(x, y)$ is constructed through the isomorphic image of the progressively enlarged submodule basis. Our complexity analysis characterizes the complexity reduction brought by the transform and shows high rate codes benefit a greater complexity reduction.**

***Index Terms*—Module minimization, progressive algebraic soft decoding, Reed-Solomon codes, re-encoding transform**

I. Introduction

Reed-Solomon (RS) codes are widely applied in digital communication and storage devices. The interpolation based Guruswami-Sudan algebraic decoding algorithm [1] breaks the classical error-correction limit that is half of the code's minimum Hamming distance. Utilizing soft information, Koetter and Vardy [2] further proposed the algebraic soft decoding (ASD) algorithm. However, the algebraic decoding algorithms remain complex due to the construction of the interpolated polynomial $Q(x, y)$. To facilitate the decoding, the progressive ASD (PASD) algorithm [3] constructs $Q(x, y)$ with a progressively enlarged y-degree. The decoding terminates once the message polynomial $f(x)$ is a y-root of Q, i.e., $Q(x, f(x)) = 0$, thus adapting the decoding capability and computation to the channel.

$Q(x, y)$ can be constructed using the concept of Gröbner basis of module [4]. A module basis contains a set of bivariate polynomials that satisfy the prescribed interpolation condition with a maximum y-degree. It can be further reduced into the Gröbner basis whose minimum candidate is $Q(x, y)$. This interpolation technique is called module minimization (MM). Earlier research [5] showed it requires less finite field arithmetic operations than the conventional Koetter's interpolation [6]. The basis reduction can be further facilitated by a number of advanced techniques [7]–[9]. The recently introduced progressive MM interpolation constructs $Q(x, y)$ through the image of the progressively enlarged submodule basis, namely the PASD-MM algorithm [10]. More than reducing the complexity, it removes the memory requirement of the original PASD algorithm [3].

Re-encoding can further reduce the interpolation complexity by transforming the interpolation points [11]. As for the MM interpolation, this will result in a common divisor (polynomial in x) for module generators. As a result, the basis reduction can be simplified by trimming down the x-degree of the generators [5]. This paper proposes the progressive MM interpolation based on the re-encoding transform (ReT), yielding a further complexity reduction over the PASD-MM algorithm [10]. The greatest common divisor (GCD) of the image polynomials will be derived. Isomorphic image can be created by removing the GCD. This attributes to a simpler image expansion and reduction. Our analysis shows that the ReT yields a complexity reduction factor of $\frac{k}{n}$ (k-dimension, n-length of an RS code), revealing the complexity advantage of high rate codes. Since the transform requires computation, we also study the tradeoff between this extra cost and the complexity reduction it brings. Our study shows when $\frac{k}{n} > 0.5$, an overall complexity reduction can be ensured.

II. Prerequisite Knowledge

Let $\mathbb{F}_q = \{\sigma_0, \sigma_1, \ldots, \sigma_{q-1}\}$ denote a finite field of size q. $\mathbb{F}_q[x]$ and $\mathbb{F}_q[x, y]$ are the univariate and bivariate polynomial rings defined over $\mathbb{F}_q$, respectively. Given an (n, k) RS code, message polynomial $f(x) \in \mathbb{F}_q[x]$ can be written as

$$f(x) = f_0 + f_1 x + \cdots + f_{k-1} x^{k-1}, \tag{1}$$

where $f_0, f_1, \ldots, f_{k-1}$ are message symbols. Codeword $\underline{c} = (c_0, c_1, \ldots, c_{n-1}) \in \mathbb{F}_q^n$ is generated by

$$\underline{c} = (f(\alpha_0), f(\alpha_1), \ldots, f(\alpha_{n-1})), \tag{2}$$

where $\alpha_0, \alpha_1, \ldots, \alpha_{n-1}$ are the n distinct nonzero elements of $\mathbb{F}_q$. They are called code locators.

Assume codeword $\underline{c}$ is transmitted through a memoryless channel and $\underline{\mathsf{r}} = (\mathsf{r}_0, \mathsf{r}_1, \ldots, \mathsf{r}_{n-1}) \in \mathbb{R}^n$ is the received symbol vector. A reliability matrix $\mathbf{\Pi}_{q \times n}$ can be obtained. Its entry $\pi_{ij} = \Pr[c_j = \sigma_i \mid \mathsf{r}_j]$ is the symbol wise *a posteriori* probability [1]. It will be transformed into a multiplicity matrix

[1]It is assumed that $\Pr[c_j = \sigma_i] = \frac{1}{q}, \forall (i, j)$.

$\mathbf{M}_{q\times n}$, where entry m_{ij} is the interpolation multiplicity for point (α_j, σ_i). Interpolation constructs a minimum polynomial $Q(x,y)$ that interpolates all the points with their multiplicity. Given $Q(x,y) = \sum_{a,b} Q_{ab}x^a y^b \in \mathbb{F}_q[x,y]$, its monomials $x^a y^b$ can be organized under the (μ,ν)-revlex order [2]. Let $x^{a'}y^{b'}$ denote the leading monomial of Q with $Q_{a'b'} \neq 0$, the (μ,ν)-weighted degree of Q is $\deg_{\mu,\nu} Q = \deg_{\mu,\nu} x^{a'}y^{b'}$. Given two distinct polynomials Q_1 and Q_2 with leading monomials $x^{a'_1}y^{b'_1}$ and $x^{a'_2}y^{b'_2}$, respectively, $Q_1 < Q_2$ if $x^{a'_1}y^{b'_1} < x^{a'_2}y^{b'_2}$. For an (n,k) RS code, interpolation constructs a minimum polynomial Q w.r.t. the $(1,k-1)$-revlex order. Let $l = \deg_y Q$ be the decoding parameter. Root-finding decodes $f(x)$ through $Q(x,f(x)) = 0$ [12].

III. The PASD-MM

A. MM Interpolation

In the $\mathbf{\Pi} \to \mathbf{M}$ transform [2], let

$$\mathsf{m}_j = \sum_{i=0}^{q-1} m_{ij} \tag{3}$$

and $\mathsf{m} = \max\{\mathsf{m}_j, \forall j\}$. This process terminates when $\mathsf{m} = l$.

Definition I. Given $\mathbf{M}$, module $\mathcal{M}_l$ is the space of all polynomials in $\mathbb{F}_q[x,y]$ that interpolate points (α_j, σ_i) with a multiplicity of m_{ij} and with a maximum y-degree of l.

Let L_j denote a list that enumerates interpolation points (α_j, σ_i) from column j of $\mathbf{M}$ as

$$L_j = [\underbrace{(\alpha_j,\sigma_i),\ldots,(\alpha_j,\sigma_i)}_{m_{ij}}, \forall i \text{ and } m_{ij} \neq 0], \tag{4}$$

where $|L_j| = \mathsf{m}_j$. Its balanced list L'_j is created by moving one of the most frequent elements of L_j to the back of L'_j, and repeating this process m_j times until L_j becomes empty.

Remark 1. If more than one point have the same frequency in L_j, the one that corresponds to a larger π_{ij} is prioritized to be moved to L'_j.

L'_j can be denoted as

$$L'_j = [(\alpha_j, y_j^{(0)}), (\alpha_j, y_j^{(1)}), \ldots, (\alpha_j, y_j^{(\mathsf{m}_j-1)})], \tag{5}$$

where $y_j^{(0)}, y_j^{(1)}, \ldots, y_j^{(\mathsf{m}_j-1)} \in \mathbb{F}_q$. Given L'_j, let

$$\mathsf{m}_j(t) = \max\{\text{multi.}((\alpha_j, y_j^{(\varepsilon)})) \mid \varepsilon = t, t+1, \ldots, \mathsf{m}_j - 1\}. \tag{6}$$

Note that $\mathsf{m}_j(0) = \max\{m_{ij}, \forall i\}$ and $\mathsf{m}_j(t) = 0$ for $t \geq \mathsf{m}_j$.

To construct a basis for $\mathcal{M}_l$, let

$$F_\varepsilon(x) = \sum_{j=0}^{n-1} y_j^{(\varepsilon)} \prod_{j'=0, j'\neq j}^{n-1} \frac{x-\alpha_{j'}}{\alpha_j - \alpha_{j'}}, \tag{7}$$

where $\varepsilon = 0, 1, \ldots, l-1$. Hence, $F_\varepsilon(\alpha_j) = y_j^{(\varepsilon)}, \forall j$. $\mathcal{M}_l$ can be generated as an $\mathbb{F}_q[x]$-module by the following polynomials

$$P_t(x,y) = \prod_{j=0}^{n-1}(x-\alpha_j)^{\mathsf{m}_j(t)} \prod_{\varepsilon=0}^{t-1}(y - F_\varepsilon(x)), \tag{8}$$

where $t = 0, 1, \ldots, l$. They are called module generators, constructing a basis of $\mathcal{M}_l$ (denoted as $\mathcal{B}_l$). Since $P_t(x,y) = \sum_{\tau\leq t} P_t^{(\tau)}(x) y^\tau$ where $P_t^{(\tau)}(x) \in \mathbb{F}_q[x]$, $\mathcal{B}_l$ can be presented as an $(l+1)\times(l+1)$ matrix over $\mathbb{F}_q[x]$, where its entry of row-t column-τ (denoted as $\mathcal{B}_l|_t^{(\tau)}$) is $P_t^{(\tau)}(x)$. $\mathcal{B}_l$ will be reduced into the Gröbner basis $\mathcal{B}'_l$ that is in weak Popov form [3] [13]. The minimum candidate of $\mathcal{B}'_l$ is the desired $Q(x,y)$.

B. Progressive MM Interpolation

The progressive MM interpolation constructs $Q(x,y)$ with a progressively enlarged y-degree. Let v denote the progressive iteration index and $1 \leq v \leq l$. $Q_v(x,y)$ with $\deg_y Q_v = v$ is constructed at iteration v.

Definition II. Given a module $\mathcal{M}_l$, its submodule $\mathfrak{M}_v$ is the subspace spanned by $P_0(x,y), \ldots, P_v(x,y)$.

Polynomials $P_0(x,y), \ldots, P_v(x,y)$ construct a basis of $\mathfrak{M}_v$ (denoted as $\mathfrak{B}_v$). Since when $t \leq v$, $\deg_y P_t(x,y) \leq v$, $\mathfrak{B}_v$ can be presented as a $(v+1)\times(v+1)$ matrix over $\mathbb{F}_q[x]$.

For a balanced list L'_j, let us define

$$\delta_j(t) = \mathsf{m}_j(t) - \mathsf{m}_j(t+1), \tag{9}$$

where $t = 0, 1, \ldots, l$. Since $\mathsf{m}_j \leq l$, $\mathsf{m}_j(l+1) = \mathsf{m}_j(l) = 0$. Consequently, $\delta_j(l-1) = \mathsf{m}_j(l-1)$ and $\delta_j(l) = 0$. Let

$$G_t(x) = \prod_{j=0}^{n-1}(x-\alpha_j)^{\mathsf{m}_j(t)} \tag{10}$$

and

$$R_t(x) = \prod_{j=0}^{n-1}(x-\alpha_j)^{\delta_j(t)}. \tag{11}$$

Based on (9), we have

$$G_t(x) = G_{t+1}(x)R_t(x). \tag{12}$$

Since $\mathsf{m}_j(l) = \delta_j(l) = 0, \forall j$, $G_l(x) = R_l(x) = 1$.

Let $\Theta_t^\tau = \{\theta \subset \{0,1,\ldots,t-1\} \mid |\theta| = \tau\}$. Note that $\Theta_t^0 = \{\emptyset\}$, $|\Theta_t^\tau| = \binom{t}{\tau}$ and $|\Theta_t^t| = 1$. Therefore, generators (8) can be rewritten as

$$P_t(x,y) = G_t(x)W_t(x,y), \tag{13}$$

where

$$W_t(x,y) = \prod_{\varepsilon=0}^{t-1}(y - F_\varepsilon(x)) = \sum_{\tau=0}^{t} w_t^{(\tau)}(x) y^\tau \tag{14}$$

and

$$w_t^{(\tau)}(x) = \sum_{\theta\in\Theta_t^{t-\tau}} \prod_{\varepsilon\in\theta}(-F_\varepsilon(x)). \tag{15}$$

Note that $w_t^{(t)}(x) = 1$ and $W_0(x,y) = 1$.

Theorem 2 *[10].* Basis $\mathfrak{B}_v$ can be constructed through

$$\mathfrak{B}_v = G_v(x) \cdot \Xi_v, \tag{16}$$

where

$$\Xi_v = \begin{bmatrix} & R_{v-1}(x)\cdot\Xi_{v-1} & & \mathbf{0}_{v\times 1} \\ w_v^{(0)}(x) & \cdots & w_v^{(v-1)}(x) & w_v^{(v)}(x) \end{bmatrix} \tag{17}$$

[2]The (μ,ν)-weighted degree of $x^a y^b$ is $\deg_{\mu,\nu} x^a y^b = \mu a + \nu b$. Given $x^{a_1}y^{b_1}$ and $x^{a_2}y^{b_2}$, $x^{a_1}y^{b_1} < x^{a_2}y^{b_2}$, if $\deg_{\mu,\nu} x^{a_1}y^{b_1} < \deg_{\mu,\nu} x^{a_2}y^{b_2}$, or $\deg_{\mu,\nu} x^{a_1}y^{b_1} = \deg_{\mu,\nu} x^{a_2}y^{b_2}$ and $b_1 < b_2$.

[3]In an $(l+1)\times(l+1)$ matrix over $\mathbb{F}_q[x]$, each row represents a polynomial in $\mathbb{F}_q[x,y]$. The weak Popov form implies the y-degree of the leading monomial of each row is different.

is the image of $\mathfrak{B}_v$ and $\mathbf{0}_{v\times 1}$ is an all zero vector.

When $v = l$, $G_l(x) = 1$ and $\mathcal{B}_l = \mathfrak{B}_l = \Xi_l$. Therefore, the desired basis $\mathcal{B}_l$ can be constructed through the image of the progressively enlarged submodule basis. Theorem 2 shows $G_v(x)$ is the GCD of polynomials in $\mathfrak{B}_v$. One can perform the basis reduction on Ξ_v instead of $\mathfrak{B}_v$. That says by reducing Ξ_v into weak Popov form Ξ'_v, the desired reduced submodule basis $\mathfrak{B}'_v$ can be constructed by $\mathfrak{B}'_v = G_v(x)\cdot\Xi'_v$. $Q_v(x,y)$ is the minimum candidate of $\mathfrak{B}'_v$. Theorem 2 also shows that Ξ_v can be recursively constructed from Ξ_{v-1}. Therefore, $Q_v(x,y)$ can be determined from the progressively enlarged image Ξ_v. We further define generators of Ξ_v as

$$\mathcal{P}_{v,t}(x,y) = R_{v-1}(x)\mathcal{P}'_{v-1,t}(x,y), \text{ if } 0 \le t \le v-1, \quad (18)$$

$$\mathcal{P}_{v,v}(x,y) = W_v(x,y), \quad (19)$$

where $\mathcal{P}'_{v-1,t}(x,y)$ are the polynomials of Ξ'_{v-1} and $\mathcal{P}'_{0,0}(x,y) = 1$. Reducing Ξ_v into Ξ'_v, we can determine $Q_v(x,y)$. Once $Q_v(x,f(x)) = 0$, the decoding terminates.

IV. The ReT Based PASD-MM

A. ReT

Let $\tilde{\pi}_j = \max\{\pi_{ij}, \forall i\}$. Entries $\tilde{\pi}_0, \tilde{\pi}_1, \ldots, \tilde{\pi}_{n-1}$ can be sorted to obtain an index sequence $j_0, j_1, \ldots, j_{n-1}$ such that $\tilde{\pi}_{j_0} \ge \tilde{\pi}_{j_1} \ge \cdots \ge \tilde{\pi}_{j_{n-1}}$. Let $\Upsilon = \{j_0, j_1, \ldots, j_{k-1}\}$ and $\bar{\Upsilon} = \{j_k, j_{k+1}, \ldots, j_{n-1}\}$. The above sorting ensures points $(\alpha_j, y_j^{(0)}), \forall j \in \Upsilon$, correspond to the k largest multiplicities in $\mathbf{M}$. They are chosen to construct the re-encoding polynomial

$$H(x) = \sum_{j\in\Upsilon} y_j^{(0)} \prod_{j'\in\Upsilon, j'\ne j} \frac{x-\alpha_{j'}}{\alpha_j - \alpha_{j'}}, \quad (20)$$

where $H(\alpha_j) = y_j^{(0)}, \forall j \in \Upsilon$. In the n balanced lists L'_j, all points are transformed by $(\alpha_j, u_j^{(\varepsilon)}) = (\alpha_j, y_j^{(\varepsilon)} - H(\alpha_j))$. Consequently, all lists L'_j are transformed into

$$\tilde{L}'_j = [(\alpha_j, u_j^{(\varepsilon)}) \mid \varepsilon = 0, 1, \ldots, \mathsf{m}_j - 1]. \quad (21)$$

For $j \in \Upsilon$, if $y_j^{(\varepsilon)} = y_j^{(0)}$, then $u_j^{(\varepsilon)} = 0$. Further let $\Lambda_\varepsilon = \{j \mid u_j^{(\varepsilon)} = 0, j \in \Upsilon\}$ and $\bar{\Lambda}_\varepsilon = \Upsilon \backslash \Lambda_\varepsilon$. Note that $\Lambda_0 = \Upsilon$.

Lemma 3. Given $\tilde{L}'_j$ and $\varepsilon \ge 1$, if $j \in \Lambda_\varepsilon$, $\delta_j(\varepsilon-1) = 1$. Otherwise, $\delta_j(\varepsilon-1) = 0$. Note that $\delta_j(\varepsilon)$ is defined as in (9).

Proof: This is ensured by Remark 1. Point $(\alpha_j, 0)$ would appear earlier in $\tilde{L}'_j$ if it has the same multiplicity as another point. Therefore, when $\varepsilon \ge 1$, if $j \in \Lambda_\varepsilon$, multi.$((\alpha_j, u_j^{(\varepsilon-1)}))$ − multi.$((\alpha_j, u_j^{(\varepsilon)})) = 1$ and $\mathsf{m}_j(\varepsilon-1) - \mathsf{m}_j(\varepsilon) = 1$. Otherwise, multi.$((\alpha_j, u_j^{(\varepsilon-1)}))$ = multi.$((\alpha_j, u_j^{(\varepsilon)}))$ and $\mathsf{m}_j(\varepsilon-1) = \mathsf{m}_j(\varepsilon)$. ∎

B. ReT Based Progressive MM Interpolation

Based on the above transform, $F_\varepsilon(x)$ of (7) is redefined as

$$F_\varepsilon(x) = \sum_{j=0}^{n-1} u_j^{(\varepsilon)} \prod_{j'=0, j'\ne j}^{n-1} \frac{x-\alpha_{j'}}{\alpha_j - \alpha_{j'}}. \quad (22)$$

Let us define

$$\psi(x) = \prod_{j\in\Upsilon}(x-\alpha_j). \quad (23)$$

Based on (19),

$$\begin{aligned}\mathcal{P}_{v,v}(x, y\psi(x)) &= \prod_{\varepsilon=0}^{v-1}(y\psi(x) - F_\varepsilon(x)) \\ &= \prod_{\varepsilon=0}^{v-1}\prod_{j\in\bar{\Lambda}_\varepsilon}(x-\alpha_j)\cdot\tilde{W}_v(x,y), \end{aligned} \quad (24)$$

where

$$\tilde{W}_v(x,y) = \prod_{\varepsilon=0}^{v-1}(y\prod_{j\in\bar{\Lambda}_\varepsilon}(x-\alpha_j) - T_\varepsilon(x)) \quad (25)$$

and

$$T_\varepsilon(x) = \sum_{j\in\bar{\Upsilon}\cup\bar{\Lambda}_\varepsilon} \frac{u_j^{(\varepsilon)}}{\prod_{j'=0, j'\ne j}^{n-1}(\alpha_j - \alpha_{j'})} \prod_{j'\in\bar{\Upsilon}\cup\bar{\Lambda}_\varepsilon, j'\ne j}(x-\alpha_{j'}). \quad (26)$$

Note that $\tilde{W}_0(x,y) = 1$. When $\varepsilon = 0$, $\Lambda_0 = \Upsilon$ and $\prod_{j\in\Lambda_0}(x-\alpha_j) = \prod_{j\in\Upsilon}(x-\alpha_j)$. Further based on Lemma 3,

$$\begin{aligned}\prod_{\varepsilon=0}^{v-1}\prod_{j\in\Lambda_\varepsilon}(x-\alpha_j) &= \prod_{j\in\Upsilon}(x-\alpha_j)\cdot\prod_{\varepsilon=1}^{v-1}\prod_{j\in\Upsilon}(x-\alpha_j)^{\delta_j(\varepsilon-1)} \\ &= \prod_{j\in\Upsilon}(x-\alpha_j)^{1+\sum_{\varepsilon=0}^{v-2}\delta_j(\varepsilon)}. \end{aligned} \quad (27)$$

Therefore,

$$\mathcal{P}_{v,v}(x, y\psi(x)) = \prod_{j\in\Upsilon}(x-\alpha_j)^{1+\sum_{\varepsilon=0}^{v-2}\delta_j(\varepsilon)}\cdot\tilde{W}_v(x,y). \quad (28)$$

It is now sufficient to derive the GCD for polynomials $\mathcal{P}_{v,t}(x, y\psi(x))$, where $t = 0, 1, \ldots, v$.

Theorem 4. The GCD of polynomials $\mathcal{P}_{v,t}(x, y\psi(x))$ is

$$U_v(x) = \prod_{j\in\Upsilon}(x-\alpha_j)^{\sum_{\varepsilon=0}^{v-1}\delta_j(\varepsilon)}. \quad (29)$$

Proof: Based on (18) and (28), when $v = 1$,

$$\mathcal{P}_{1,0}(x, y\psi(x)) = R_0(x) = \prod_{j\in\Upsilon}(x-\alpha_j)^{\delta_j(0)}\prod_{j\in\bar{\Upsilon}}(x-\alpha_j)^{\delta_j(0)},$$

$$\mathcal{P}_{1,1}(x, y\psi(x)) = \prod_{j\in\Upsilon}(x-\alpha_j)\cdot\tilde{W}_1(x,y).$$

Since $\delta_j(0) = 1$ or 0, $U_1(x) = \prod_{j\in\Upsilon}(x-\alpha_j)^{\delta_j(0)}$ is the GCD of $\mathcal{P}_{1,0}(x, y\psi(x))$ and $\mathcal{P}_{1,1}(x, y\psi(x))$, as well as $\mathcal{P}'_{1,0}(x, y\psi(x))$ and $\mathcal{P}'_{1,1}(x, y\psi(x))$. When $v = 2$,

$$\mathcal{P}_{2,t}(x, y\psi(x)) = R_1(x)\mathcal{P}'_{1,t}(x, y\psi(x)), \text{ if } t = 0, 1,$$

$$\mathcal{P}_{2,2}(x, y\psi(x)) = \prod_{j\in\Upsilon}(x-\alpha_j)^{1+\delta_j(0)}\cdot\tilde{W}_2(x,y).$$

Since $R_1(x) = \prod_{j\in\Upsilon}(x-\alpha_j)^{\delta_j(1)}\prod_{j\in\bar{\Upsilon}}(x-\alpha_j)^{\delta_j(1)}$, $U_2(x) = \prod_{j\in\Upsilon}(x-\alpha_j)^{\delta_j(0)+\delta_j(1)}$ is the GCD of $\mathcal{P}_{2,t}(x, y\psi(x))$, as well as $\mathcal{P}'_{2,t}(x, y\psi(x))$, where $t = 0, 1, 2$. The above deduction shows that (29) is the GCD of $\mathcal{P}_{v,0}(x, y\psi(x)), \ldots, \mathcal{P}_{v,v}(x, y\psi(x))$. ∎

Therefore, the following bijective mapping can be performed for polynomials $\mathcal{P}_{v,t}(x,y)$,

$$\begin{aligned}\varphi_v : \quad & \Xi_v \to \Phi_v \\ & \mathcal{P}_{v,t}(x,y) \mapsto U_v(x)^{-1}\mathcal{P}_{v,t}(x, y\psi(x)), \end{aligned} \quad (30)$$

where $t = 0, 1, \ldots, v$ and φ_v is an isomorphism between Ξ_v and Φ_v, i.e., $\Phi_v = \varphi_v(\Xi_v)$. Hence, entires of Φ_v have lower degree than those of Ξ_v, leading to a simpler image reduction.

C. The Proposed Algorithm

The proposed ReT based PASD-MM algorithm determines $Q_v(x, y)$ through the progressively enlarged isomorphic image Φ_v. At the beginning, Φ_1 is generated by

$$\tilde{\mathcal{P}}_{1,0}(x, y) = \prod_{j \in \Upsilon} (x - \alpha_j)^{\delta_j(0)}, \tag{31}$$

$$\tilde{\mathcal{P}}_{1,1}(x, y) = \prod_{j \in \Upsilon} (x - \alpha_j)^{1-\delta_j(0)} \cdot \tilde{W}_1(x, y). \tag{32}$$

With ReT, monomials are organized under the $(1, -1)$-revlex order. Φ_1 will then be reduced into weak Popov form Φ'_1. Choose the minimum candidate of Φ'_1 as $\tilde{Q}_1(x, y)$. $Q_1(x, y)$ will be further restored by $Q_1(x, y) = U_1(x)\tilde{Q}_1(x, \frac{y}{\psi(x)})$. If $Q_1(x, f'(x)) = 0$ and the estimated codeword $\underline{\hat{c}} = (\hat{c}_0, \hat{c}_1, \ldots, \hat{c}_{n-1})$, where $\hat{c}_j = f'(\alpha_j) + H(\alpha_j), \forall j$, satisfies the maximum-likelihood (ML) criterion [14] [4], the decoding terminates and the decoded message is $\hat{f}(x) = f'(x) + H(x)$. Otherwise, the decoding continues by expanding Φ'_1 to Φ_2 in pursuit of $Q_2(x, y)$.

In general, at iteration $v-1$ ($v \geq 2$), if the message cannot be decoded from Φ'_{v-1}, Φ'_{v-1} will be expanded to Φ_v by

$$\tilde{\mathcal{P}}_{v,t}(x, y) = \tilde{R}_{v-1}(x)\tilde{\mathcal{P}}'_{v-1,t}(x, y) \text{ if } 0 \leq t \leq v-1, \tag{33}$$

$$\tilde{\mathcal{P}}_{v,v}(x, y) = \prod_{j \in \Upsilon} (x - \alpha_j)^{1-\delta_j(v-1)} \cdot \tilde{W}_v(x, y), \tag{34}$$

where

$$\tilde{R}_{v-1}(x) = \prod_{j \in \Upsilon} (x - \alpha_j)^{\delta_j(v-1)} \tag{35}$$

and $\tilde{\mathcal{P}}'_{v-1,t}(x, y)$ are polynomials of Φ'_{v-1}. Φ_v will be reduced into weak Popov form Φ'_v. Choose its minimum candidate as $\tilde{Q}_v(x, y)$. $Q_v(x, y)$ will be further restored by

$$Q_v(x, y) = U_v(x)\tilde{Q}_v\Big(x, \frac{y}{\psi(x)}\Big). \tag{36}$$

If $Q_v(x, f'(x)) = 0$ and $\hat{f}(x) = f'(x) + H(x)$ can yield a codeword that satisfies the ML criterion, the decoding terminates and outputs $\hat{f}(x)$. Otherwise, the decoding continues by updating $v = v+1$. If $v > l$, the designed maximum y-degree of $Q(x, y)$ is exceeded. The decoding terminates and fails.

V. Complexity Insight

The proposed algorithm consists of re-encoding transform, progressive MM interpolation and root-finding, where the complexity of root-finding is marginal in comparison with the other two. In particular, the progressive MM interpolation consists of image expansion and its reduction. For the original PASD-MM algorithm [10], at progressive iteration v, its image expansion and reduction exhibit a complexity of $n^2v^3 + n^2 + \frac{1}{2}n^2v^2$ and $(n-k+1)nv^3(v+1)$, respectively, where the complexity is measured as the number of finite field arithmetic operations. Since re-encoding transform incurs extra computation, it is important to study the tradeoff between the complexity of the transform and its complexity reduction effect for the MM interpolation.

For re-encoding transform, construting $H(x)$ and transforming the coordinates require $(n-k)(4n-3k)$ and $(n-k)k$ operations, respectively. For progressive MM interpolation, complexity of the image expansion at iteration v (computing (33) and (34)) is

$$\mathcal{C}^{(1)}(v) = (n-k)^2v^3 + 2(n-k)^2 + \frac{1}{2}(n-k)^2v^2 \tag{37a}$$

$$\approx (n-k)^2v^3. \tag{37b}$$

Reducing Ξ_v into weak Popov form requires at most $(n-k+1)v^2$ row operations [13]. Since $\deg \Phi_v|_t^{(\tau)} \leq (n-k+1)v$, complexity of the image reduction is

$$\mathcal{C}^{(2)}(v) = (n-k+1)^2v^3(v+1) \tag{38a}$$

$$\approx (n-k)^2v^4. \tag{38b}$$

Restoring $\tilde{Q}_v$ into Q_v (computing (36)) requires

$$\mathcal{C}^{(3)}(v) = \frac{v^2k^2}{2} + k(n-k)v^3 + \frac{knv^3}{2} \tag{39a}$$

$$\approx knv^3 \tag{39b}$$

operations. Therefore, if the decoding terminates at iteration v, the MM interpolation complexity will be $\sum_{v'=1}^{v}(\mathcal{C}^{(1)}(v') + \mathcal{C}^{(2)}(v') + \mathcal{C}^{(3)}(v'))$.

TABLE I
MM Interpolation Complexity Comparison

	Image Exp.	Image Red.	Poly. Restoration
PASD-MM	n^2v^3	$(n-k)nv^4$	–
ReT PASD-MM	$(n-k)^2v^3$	$(n-k)^2v^4$	knv^3

Table I compares the MM interpolation complexity (at progressive iteration v) between the PASD-MM and the ReT PASD-MM algorithms. For the ReT PASD-MM algorithm, $(n-k)^2v^3 + knv^3 \approx n^2v^3$. The complexity reduction in image expansion compensates the extra computation of polynomial restoration. For both algorithms, when v ($\deg_y Q_v$) is sufficiently large, image reduction dominates the interpolation complexity. As a result, re-encoding transform yields a complexity reduction factor of $\frac{k}{n}$. That says high rate codes will benefit a greater complexity reduction from the transform.

However, if the progressive decoding terminates at an early stage, e.g., $v = 1$, the transform computation may not be compensated by the MM complexity reduction. The above analysis shows that code rate plays an important role at this tradeoff. Let us take $v = 1$ for more insights. Now complexity $\mathcal{C}^{(1)}(1)$, $\mathcal{C}^{(2)}(1)$ and $\mathcal{C}^{(3)}(1)$ will be characterized by (37a), (38a) and (39a), respectively. Further considering the transform complexity, the total computational cost is $\frac{1}{2}(n-k)(19n-13k) + \frac{1}{2}(k^2+kn)$. For the PASD-MM algorithm, when $v = 1$, its MM complexity is $\frac{5}{2}n^2 + 2(n-k+1)n$ [10]. Enforing the ReT PASD-MM complexity smaller than the PASD-MM, we have $14\big(\frac{k}{n}\big)^2 - 27\big(\frac{k}{n}\big) + 10 < 0$. It requires

[4] Cyclic redundant check can also be used for validation.

TABLE II
AVERAGE COMPLEXITY IN DECODING THE (63, 55) RS CODE ($l = 4$)

SNR (dB)	4.0	4.5	5.0	5.5	6.0	6.5	7.0
PASD	4.31×10^6	2.74×10^6	1.30×10^6	4.96×10^5	2.46×10^5	1.87×10^5	1.84×10^5
PASD-MM	1.07×10^6	7.04×10^5	3.40×10^5	1.28×10^5	6.01×10^4	4.28×10^4	4.06×10^4
ReT PASD-MM	8.09×10^5	5.22×10^5	2.47×10^5	9.05×10^4	3.86×10^4	2.49×10^4	2.28×10^4

$\frac{k}{n} > 0.5$. That says when terminating at the first iteration, re-encoding tranform will yield a lower complexity than the non-ReT counterpart for codes with a rate greater than 0.5.

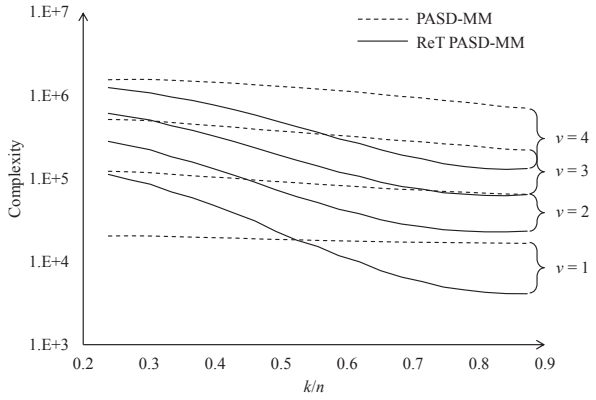

Fig. 1. Progressive interpolation complexity for RS codes defined over $\mathbb{F}_{64}$.

Fig. 1 shows our simulation results of progressive interpolation complexity in decoding RS codes defined over $\mathbb{F}_{64}$. Note that each curve shows the interpolation complexity when the progressive decoding terminates at iteration v for either the PASD-MM algorithm or the ReT PASD-MM algorithm. They show that re-encoding transform yields a lower complexity as the code rate increases, resulting in a more significant complexity reduction over the PASD-MM algorithm. In particular, when $v = 1$, the ReT PASD-MM algorithm becomes less complex when $\frac{k}{n} > 0.5$, validating the above analysis. If the progressive decoding terminates at a later stage, e.g., $v = 4$, image reduction dominates the complexity. Re-encoding transform results in a lower complexity for codes of all rate. Note that if the channel condition is sufficiently good, most of the progressive decoding will terminate with $v = 1$. Rate of 0.5 is also a watershed for the asymptotic complexity performance of the two algorithms.

Table II further shows how the progressive decoding complexity varies w.r.t. channel condition. They are obtained over the additive white Gaussian noise (AWGN) channel using BPSK. The PASD algorithm [3] applies Koetter's interpolation. As the signal-to-noise ratio (SNR) increases, more decoding events terminate at an earlier stage, resulting in a smaller complexity. By comparing the PASD and the PASD-MM algorithms, the MM interpolation can reduce the complexity by an order of magnitude. The re-encoding transform further reduces the PASD-MM complexity. Echoing Fig. 1, with this code rate, the ReT PASD-MM algorithm is always simpler than the PASD-MM algorithm. It should be pointed out that the proposed algorithm achieves the same performance as the PASD-MM algorithm.

VI. CONCLUSION

This paper has introduced the re-encoding transformed ASD for RS codes, where its interpolation is realized by the progressive MM technique. Re-encoding transform results in a common divisor for polynomials of the image of submodule basis. Removing it leads to a simpler progressive interpolation. Complexity analysis has revealed the proposed ReT PASD-MM algorithm yields a complexity reduction factor of $\frac{k}{n}$ over its non-ReT counterpart. Its complexity advantage can be ensured if the code rate is greater than 0.5.

ACKNOWLEDGEMENT

This work is sponsored by National Natural Science Foundation of China (NSFC) with project ID 61671486 and International Program for Ph.D. Candidates, Sun Yat-sen University.

REFERENCES

[1] 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.
[2] R. Koetter and A. Vardy, "Algebraic soft-decision decoding of Reed-Solomon codes," *IEEE Trans. Inform. Theory*, vol. 49, no. 11, pp. 2809–2825, Nov. 2003.
[3] 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.
[4] K. Lee and M. O'Sullivan, "An interpolation algorithm using Gröbner bases for soft-decision decoding of Reed-Solomon codes," in *Proc. the IEEE Int. Symp. Inform. Theory (ISIT)*, Seattle, USA, Jul. 2006.
[5] L. Chen and M. Bossert, "Algebraic Chase decoding of Reed-Solomon codes using module minimisation," in *Proc. the Int. Symp. Inform. Theory App. (ISITA)*, Monterey, USA, Oct. 2016.
[6] R. Koetter, "On algebraic decoding of algebraic-geometric and cyclic codes," Ph.D. dissertation, Univ. Linköping, Linköping, Sweden, 1996.
[7] M. Alekhnovich, "Linear Diophantine equations over polynomials and soft decoding of Reed-Solomon codes," *IEEE Trans. Inform. Theory*, vol. 51, no. 7, pp. 2257–2265, Jul. 2005.
[8] J. Ma and A. Vardy, "A complexity reducing transformation for the Lee-O'Sullivan interpolation algorithm," in *Proc. the IEEE Int. Symp. Inform. Theory (ISIT)*, Nice, France, Jun. 2007.
[9] C. Jeannerod, V. Neiger, É. Schost, and G. Villard, "Computing minimal interpolation bases," *J. Symb. Comput.*, vol. 83, pp. 272–314, Nov. 2017.
[10] J. Xing, L. Chen, and M. Bossert, "Progressive algebraic soft decoding of Reed-Solomon codes using module minimization," in *Proc. the IEEE Int. Symp. Inform. Theory (ISIT)*, Vail, USA, Jun. 2018.
[11] R. Koetter and A. Vardy, "A complexity reducing transformation in algebraic list decoding of Reed-Solomon codes," in *Proc. the IEEE Inform. Theory Workshop (ITW)*, Paris, France, Apr. 2003.
[12] 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.
[13] T. Mulders and A. Storjohann, "On lattice reduction for polynomial matrices," *J. Symb. Comput.*, vol. 35, no. 4, pp. 377–401, Apr. 2003.
[14] 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.