Efficient Interpolation and Factorization in Algebraic Soft-Decision Decoding of Reed-Solomon Codes

Ralf Koetter
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
1308 W. Main Street, Urbana, IL 61801, U.S.A.
koetter@csl.uiuc.edu

Jun Ma
Department of Electrical and Computer Engineering
University of California San Diego
9500 Gilman Drive, La Jolla, CA 92093–0407, U.S.A.
jma@qualcomm.com

Alexander Vardy
Department of Electrical and Computer Engineering
University of California San Diego
9500 Gilman Drive, La Jolla, CA 92093–0407, U.S.A.
vardy@kilimanjaro.ucsd.edu

Arshad Ahmed
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
1308 W. Main Street, Urbana, IL 61801, U.S.A.
ahmed4@uiuc.edu

Abstract

Algebraic soft-decision decoding of Reed-Solomon codes delivers promising coding gains over conventional hard-decision decoding. The main computational steps in algebraic soft-decoding (as well as Sudan-type list-decoding) are bivariate interpolation and factorization. We discuss a new computational technique, based upon re-encoding and coordinate transformation, that significantly reduces the complexity these steps. The re-encoding procedure and the coordinate transformation convert the original interpolation problem into another *reduced interpolation problem*, which is orders of magnitude smaller than the original one. We present a rigorous proof that the two interpolation problems are equivalent. We also present an efficient factorization procedure that applies directly to the reduced interpolation problem.

I. Introduction

Reed-Solomon (RS) codes are the most widely used error-correcting codes in digital communications and data storage. Traditional *hard-decision decoders* correct up to $\frac{n-k}{2}$ symbol errors for an (n,k) Reed-Solomon code. Recently, several breakthroughs have been achieved in improving the error-correction capability of a Reed-Solomon decoder. Sudan [8] and Guruswami-Sudan [3] discovered a *list-decoding* algorithm, which can correct up to $n - \sqrt{nk}$ symbol errors. This list-decoding method was later extended by Koetter and Vardy [4] to an algebraic *soft-decision decoding* algorithm, which significantly outperforms hard-decision list-decoding. Both list-decoding and algebraic soft-decision decoding use interpolation and factorization of bivariate polynomials, which is much more computationally complex than hard-decision decoding. Fast interpolation and factorization algorithms have been studied in [1,6,7,9]. Here, we discuss a new technique [2], based on re-encoding and coordinate transformation, which reduces the complexity of the interpolation process by a factor of at least $\frac{n}{n-k}$. The following example sheds some light on the magnitude of savings in computational complexity.

Example 1: Consider an RS code of length $n = 458$ and dimension $k = 410$ over $\mathbb{F}_{1024}$. As shown in [4], the computational complexity of the interpolation process is a function of the *cost*, the total number of linear constraints on the coefficients the interpolation polynomial. The interpolation is usually performed using an iterative (Groebner-basis) algorithm, with the number of iterations equal to the total cost. The cost of a single interpolation point of multiplicity m is $\frac{1}{2}m(m+1)$.

multiplicity	16	15	14	14	13	12	11	10	9	8	7	6	5	4	3	2
# of points	374	29	7	10	10	4	11	6	4	5	3	6	8	5	9	15
total cost	50864	3480	735	1050	910	312	726	330	180	180	84	126	120	50	54	45

The table above shows a typical list of interpolation points. The total cost of interpolation through all the points on the list is 59246. However, using the technique of re-encoding and coordinate transformation, we do not need to interpolate through points in the first three columns of the table, thereby eliminating a cost of 55079. Thus the reduced interpolation procedure has a total cost of only $59246 - 55079 = 4167$. This reduces the number of required iterations by a factor of about 14.2. Another significant reduction in complexity is due to the fact that the resulting interpolation polynomial $Q(X, Y)$ is of much smaller degree. The original interpolation problem leads to a polynomial of X-degree about 6500 and Y-degree 17. The reduced interpolation problem results in a polynomial of maximum X-degree 270 and Y-degree 17. Thus not only is the number of operations reduced, but the number of operations involved in each iteration is also significantly reduced. □

In this paper, we present a rigorous proof that the optimal solution of the reduced interpolation problem is also the optimal solution of the original interpolation problem. In Section II, we briefly review the interpolation process based on re-encoding and coordinate transformation, and discuss the relationship between this process and the conventional interpolation process. A proof that the two problems are equivalent is then presented in Section III. In Section IV, we show that the new interpolation technique also leads to a simplified factorization algorithm.

II. THE RE-ENCODING METHOD AND THE REDUCED INTERPOLATION PROBLEM

We assume an (n, k) Reed-Solomon code $\mathbb{C}$ over $\mathbb{F}_q$. We also assume a $q \times n$ *multiplicity matrix* $M = [m_{a,b}]$, where $0 \leqslant a \leqslant n-1$ and $0 \leqslant b \leqslant q-1$. See [4] for an explanation of how this matrix is determined. The number $m_{a,b} \geqslant 0$ represents the multiplicity of the interpolation point $(x_a, y_{a,b}) \in \mathbb{F}_q^2$. We let $\mathbf{P}$ denote the set of all interpolation points.

Original Interpolation Problem: *The original interpolation problem consists of computing the (non-trivial) bivariate polynomial $A(X, Y)$ of minimal $(1, k-1)$-weighted degree that passes through all the points in the set $\mathbf{P}$ with the corresponding multiplicities.*

The original interpolation problem can be solved by an iterative algorithm using Groebner bases. The number of iterations required is $N = \frac{1}{2}\sum_{(x_a, y_{a,b}) \in \mathbf{P}} m_{a,b}(m_{a,b} + 1)$. We now introduce the re-encoding procedure. It will be shown later that this procedure significantly reduces the number of iterations in the interpolation process.

Re-encoding Procedure:

- Determine the maximum-likelihood hard-decision vector $\underline{u} = (u_0, u_1, \ldots, u_{n-1})$ with $u_a = y_{a,\tilde{b}(a)}$, where $\tilde{b}(a) = \text{argmax}_b\{m_{a,b}\}$ for all $a = 0, 1, \ldots, n-1$. Let $(m_0, m_1, \ldots, m_{n-1})$ denote the corresponding multiplicities $m_{a,\tilde{b}(a)}$.
- Determine a polynomial $f(X)$ with $\deg f(X) < k$ and the corresponding codeword $\underline{c} = (c_0, c_1, \ldots, c_{n-1}) \in \mathbb{C}$, with $c_a = f(x_a)$, that agree with $\underline{u}$ in the k positions $\{a_i\}_{0 \leqslant i \leqslant k-1} \subset \{0, 1, \ldots, n-1\}$ with the largest multiplicities; i.e., there does not exist m_{a_j}, where $a_j \in \{0, 1, \ldots, n-1\}$, such that $m_{a_j} > m_{a_i}$.
- Modify all the interpolation points in the following way: $y_{a,b} := y_{a,b} - c_a$.

Since Reed-Solomon codes are MDS, we can choose any k positions in a codeword as the "systematic" positions. Thus the second step of the above re-encoding procedure is always valid for Reed-Solomon codes.

After the re-encoding procedure, we can classify the interpolation points into three sets: $\mathbf{R}$, $\mathbf{I}$ and $\mathbf{E}$. We use $(x_i, y_{i,j})$, $(x_{\hat{\imath}}, y_{\hat{\imath},\hat{\jmath}})$, and $(x_{\tilde{\imath}}, y_{\tilde{\imath},\tilde{\jmath}})$ to represent points in these three sets, respectively. Let us use $\mathbf{S}$ to denote the index set of the "systematic" positions. We also define three index sets as follows:

$$
\begin{aligned}
\mathbf{I_R} &\stackrel{\text{def}}{=} \{(i,j) : i \in \mathbf{S} \text{ and } (x_i, y_{i,j}) \in \mathbf{R}\} \\
\mathbf{I_I} &\stackrel{\text{def}}{=} \{(\hat{\imath},\hat{\jmath}) : \hat{\imath} \notin \mathbf{S} \text{ and } (x_{\hat{\imath}}, y_{\hat{\imath},\hat{\jmath}}) \in \mathbf{I}\} \\
\mathbf{I_E} &\stackrel{\text{def}}{=} \{(\tilde{\imath},\tilde{\jmath}) : \tilde{\imath} \in \mathbf{S} \text{ and } (x_{\tilde{\imath}}, y_{\tilde{\imath},\tilde{\jmath}}) \in \mathbf{E}\}
\end{aligned}
$$

Here, $\mathbf{R}$ represents the set of points through which we have re-encoded, $\mathbf{I}$ represents the set of interpolation points $(x_a, y_{a,b})$ such that $a \notin \mathbf{S}$, and $\mathbf{E}$ represents the set of interpolation points $(x_a, y_{a,b})$ such that $a \in \mathbf{S}$. The re-encoding procedure ensures the following properties for the interpolation points:

$$y_{i,j} = 0 \text{ for all } (i,j) \in \mathbf{I_R} \tag{1}$$

$$m_{i,j} \geqslant m_{\hat{i},\hat{j}} \text{ for all } (i,j) \in \mathbf{I_R} \text{ and } (\hat{i},\hat{j}) \in \mathbf{I_I} \tag{2}$$

$$m_{\tilde{i},j} \geqslant m_{\tilde{i},\tilde{j}} \text{ for all} (\tilde{i},j) \in \mathbf{I_R} \text{ and } \tilde{i},\tilde{j} \in \mathbf{I_E} \tag{3}$$

The key property is expressed in (1) above: after re-encoding most interpolation points have zero Y-coordinate. Using the definitions of $\mathbf{R}$, $\mathbf{I}$, and $\mathbf{E}$, we can now state the "shifted" interpolation problem as follows.

Shifted Interpolation Problem: *The shifted interpolation problem consists of computing the (nontrivial) bivariate polynomial $Q(X,Y)$ of minimal $(1,k-1)$-weighted degree that passes through all the points in three sets $\mathbf{R}$, $\mathbf{I}$, and $\mathbf{E}$ with the corresponding multiplicities.*

It is easy to see that if $Q(X,Y)$ is a solution to the shifted interpolation problem then $A(X,Y) = Q(X,Y+f(X))$ is a solution to the original interpolation problem. The savings in the number of iterations come from the following observation.

LEMMA 1. *Let $Q(X,Y)$ be a bivariate polynomial, with Y-degree equal to a positive integer l, that passes through all the points in the three sets $\mathbf{R}$, $\mathbf{I}$, and $\mathbf{E}$, as defined above, with given multiplicities. Then $Q(X,Y)$ can be expressed as*

$$Q(X,Y) \;=\; \sum_{t=0}^{l} Q_t(X) \prod_{(i,j)\in\mathbf{I_R}} (X-x_i)^{\max(m_{i,j}-t,0)} Y^t$$

For a proof of this lemma, we refer the reader to [5]. Let us now define the notation $[i]_+ \stackrel{\text{def}}{=} \max(i,0)$, and introduce the polynomial $T_t(X) \stackrel{\text{def}}{=} \prod_{(i,j)\in\mathbf{I_R}} (X-x_i)^{[t-m_{i,j}]_+}$. Based on Lemma 1 and the definitions above, we can re-write $Q(X,Y)$ as follows:

$$Q(X,Y) \;\stackrel{\text{def}}{=}\; \sum_{t=0}^{l} Q_t(X) \prod_{(i,j)\in\mathbf{I_R}} (X-x_i)^{m_{i,j}-t} T_t(X) Y^t \tag{4}$$

Thus the interpolation process can be carried out only on $\mathbf{I}$ and $\mathbf{E}$, as long as we properly initialize the Groebner basis polynomials. From (4), we can see that $\prod_{(i,j)\in\mathbf{I_R}} (X-x_i)^{m_{i,j}}$ is carried as an "overhead" throughout the interpolation process. This prompts us to implement the interpolation using a coordinate system transformation, in order to reduce the computation and the memory requirements. With this coordinate system transformation, we have two interpolation point sets, defined as follows:

$$\hat{\mathbf{I}} \;\stackrel{\text{def}}{=}\; \left\{ (x_{\hat{i}}, z_{\hat{i},\hat{j}}) : z_{\hat{i},\hat{j}} = \frac{y_{\hat{i},\hat{j}}}{V(x_{\hat{i}})} \text{ and } (x_{\hat{i}}, y_{\hat{i},\hat{j}}) \in \mathbf{I} \right\} \quad \text{and} \quad \hat{\mathbf{E}} \;\stackrel{\text{def}}{=}\; \left\{ (x_{\tilde{i}}, z_{\tilde{i},\tilde{j}}) : z_{\tilde{i},\tilde{j}} = \frac{y_{\tilde{i},\tilde{j}}}{V^{(1)}(x_{\tilde{i}})} \text{ and } (x_{\tilde{i}}, y_{\tilde{i},\tilde{j}}) \in \mathbf{E} \right\}$$

where $V(X) \stackrel{\text{def}}{=} \prod_{i\in\mathbf{R}} (X-x_i)$ and $V^{(1)}(X)$ is the first order derivative of $V(X)$. Clearly $V^{(1)}(x_{\tilde{i}}) = \prod_{i\in\mathbf{S}, i\neq\tilde{i}} (x_{\tilde{i}} - x_i)$.

Reduced Interpolation Problem: *The reduced interpolation problem consists of finding a set of polynomials $\{Q_t(X)\}_{0\leqslant t\leqslant l}$ satisfying the following three conditions:*

- CONDITION 1. *For each of the interpolation points $(x_{\hat{i}}, z_{\hat{i},\hat{j}}) \in \hat{\mathbf{I}}$, the polynomial $\Phi(X,Z)$ passes through $(x_{\hat{i}}, z_{\hat{i},\hat{j}})$ with multiplicity $m_{\hat{i},\hat{j}}$, where $\Phi(X,Z)$ is defined as follows:*

$$\Phi(X,Z) \;\stackrel{\text{def}}{=}\; \sum_{t=0}^{l} Q_t(X) T_t(X) Z^t. \tag{5}$$

- CONDITION 2. *For each $(x_{\tilde{i}}, z_{\tilde{i},\tilde{j}}) \in \hat{\mathbf{E}}$, the polynomial $\Theta^{\tilde{i}}(X,Z)$ passes through $(x_{\tilde{i}}, z_{\tilde{i},\tilde{j}})$ with multiplicity $m_{\tilde{i},\tilde{j}}$, where $\Theta^{\tilde{i}}(X,Z)$ is defined as follows:*

$$\Theta^{\tilde{i}}(X,Z) \;\stackrel{\text{def}}{=}\; \sum_{t=0}^{l} Q_t(X)(X-x_{\tilde{i}})^{m_{\tilde{i},j}-t} T_t(X) Z^t. \tag{6}$$

- CONDITION 3. *The polynomial $\Phi(X,Z)$ has minimal $(1,-1)$-weighted degree subject to Condition 1.*

III. Proof of the equivalence of the reduced and the original interpolation problems

As we have shown so far, the shifted interpolation problem can be converted to the reduced interpolation problem, which significantly benefits decoder implementation for high-rate Reed-Solomon codes. However, it remains to be shown that the reduced interpolation problem is equivalent to the shifted interpolation problem, i.e., an optimal solution for the latter is also an optimal solution for the former. In the following, we will present a few theorems that prove this statement.

THEOREM 2. *The polynomial $Q(X,Y)$ defined in (4) passes through all the points $(x_{\hat{i}}, y_{\hat{i},\hat{j}}) \in \mathbf{I}$ with multiplicities $m_{\hat{i},\hat{j}}$ if and only if $\{Q_t(X)\}_{0\leqslant t\leqslant l}$ satisfies Condition 1.*

A sketch of of the proof is given in the following. Readers may refer to [5] for more details. We first prove the "if" direction of the theorem. To this end, let us expand $\Phi(X,Z)$ into a power series form

$$\Phi(X,Z) \;=\; \sum_{t=0}^{l} Q_t(X)T_t(X)Z^t \;=\sum_{t=0}^{l} Z^t \sum_{s\geqslant 0} q_{s,t}X^s \tag{7}$$

where

$$Q_t(X)T_t(X) \;=\; \sum_{s\geqslant 0} q_{s,t}X^s \qquad\qquad \text{for } t = 0,1,\ldots,l \tag{8}$$

We then rewrite $\Phi(X,Z)$ as follows:

$$\Phi(X,Z) \;=\; \sum_{t=0}^{l} \left(Z - z_{\hat{i},\hat{j}}\right)^t \sum_{s\geqslant 0} \tilde{q}_{s,t}\,(X - x_{\hat{i}})^s \tag{9}$$

where

$$\tilde{q}_{s,t} \;=\; \sum_{t'\geqslant t}\sum_{s'\geqslant s} \binom{t'}{t}\binom{s'}{s} z_{\hat{i},\hat{j}}^{t'-t} x_{\hat{i}}^{s'-s} q_{s',t'} \tag{10}$$

We see from (9) that $\Phi(X + x_{\hat{i}}, Z + z_{\hat{i},\hat{j}}) = \sum_{t=0}^{l} Z^t \sum_{s\geqslant 0} \tilde{q}_{s,t}X^s$. Since $\Phi(X,Z)$ passes through the point $(x_{\hat{i}}, z_{\hat{i},\hat{j}})$ with multiplicity $m_{\hat{i},\hat{j}}$, we have $\tilde{q}_{s,t} = 0$ for $s + t < m_{\hat{i},\hat{j}}$. Let us now define an auxiliary polynomial

$$W(X) \;\stackrel{\text{def}}{=}\; \prod_{(i,j)\in\mathbf{I_R}} (X - x_i)^{m_{i,j}} \tag{11}$$

Plugging $W(X)$ into the definition of $Q(X,Y)$ in (4), we can rewrite $Q(X,Y)$ in the following form:

$$Q(X,Y) \;=\; W(X)\sum_{t=0}^{l}\left(\frac{Y}{V(X)}\right)^t \sum_{s\geqslant 0} q_{s,t}X^s \tag{12}$$

where $V(X)$ is as defined in Section II. Applying the same "shift" trick as in (9), we can further rewrite $Q(X,Y)$ as

$$Q(X,Y) \;=\; W(X)\sum_{t=0}^{l} \frac{\left(Y - z_{\hat{i},\hat{j}}V(X)\right)^t}{V(X)^t} \sum_{s\geqslant 0} \tilde{q}_{s,t}\,(X - x_{\hat{i}})^s\,, \tag{13}$$

where $\tilde{q}_{s,t}$ are given by (10). It can be now shown that $Q(X + x_{\hat{i}}, Y + y_{\hat{i},\hat{j}})$ does not have any monomials of degree less than $m_{\hat{i},\hat{j}}$ (for more details, see [5]). This completes the proof of the "if" direction of Theorem 2.

We now consider the "only if" direction. We need to show that if $\Phi(X,Z)$ *does not* pass through $(x_{\hat{i}}, z_{\hat{i},\hat{j}}) \in \hat{\mathbf{I}}$ with multiplicity $m_{\hat{i},\hat{j}}$, then $Q(X,Y)$ does not pass through $(x_{\hat{i}}, y_{\hat{i},\hat{j}}) \in \mathbf{I}$ with multiplicity $m_{\hat{i},\hat{j}}$. Based on (9), we have

$$\Phi(X + x_{\hat{i}}, Z + z_{\hat{i},\hat{j}}) \;=\; \sum_{t=0}^{l} Z^t \sum_{s\geqslant 0} \tilde{q}_{s,t}X^s$$

If $\Phi(X,Z)$ does not pass through the point $(x_{\hat{i}}, z_{\hat{i},\hat{j}}) \in \hat{\mathbf{I}}$ then, by definition, there exists some $\tilde{q}_{\sigma,\tau}$ such that $\tilde{q}_{\sigma,\tau} \neq 0$ and $\sigma + \tau < m_{\hat{i},\hat{j}}$. Let us now define the following two polynomials

$$\Phi_1(X + x_{\hat{i}}, Z + z_{\hat{i},\hat{j}}) \;\stackrel{\text{def}}{=}\; \sum_{t=0}^{m_{\hat{i},\hat{j}}-1} Z^t \sum_{s\geqslant 0} \tilde{q}_{s,t}X^s \qquad\qquad \Phi_2(X + x_{\hat{i}}, Z + z_{\hat{i},\hat{j}}) \;\stackrel{\text{def}}{=}\; \sum_{t=m_{\hat{i},\hat{j}}}^{l} Z^t \sum_{s\geqslant 0} \tilde{q}_{s,t}X^s \tag{14}$$

LEMMA 3. *The polynomial $\Phi_2(X,Z)$ always passes through the point $(x_{\hat{i}}, z_{\hat{i},\hat{j}})$ with multiplicity $m_{\hat{i},\hat{j}}$, regardless of the values of the coefficients $\tilde{q}_{s,t}$ in (14).*

Proof. Follows by observing that the outer sum in the definition of $\Phi_2(X + x_{\hat{i}}, Z + z_{\hat{i},\hat{j}})$ starts from $t = m_{\hat{i},\hat{j}}$. ∎

Using Lemma 3 and our assumption that $\Phi(X,Z)$ does *not* pass through the point $(x_{\hat{i}}, z_{\hat{i},\hat{j}})$ with multiplicity $m_{\hat{i},\hat{j}}$, we can conclude that there exists at least one $\tilde{q}_{s,t}$ in (14) such that $\tilde{q}_{s,t} \neq 0$ and $s + t < m_{\hat{i},\hat{j}}$. We define the order "$\prec$" on the coefficients $\tilde{q}_{s,t}$ as follows: $\tilde{q}_{s_1,t_1} \prec \tilde{q}_{s_2,t_2}$ if either $s_1 + t_1 < s_2 + t_2$, or $s_1 + t_1 = s_2 + t_2$ and $s_1 < s_2$. Let us now order all the coefficients in (14) such that $\tilde{q}_{s,t} = 0$ and $s + t < m_{\hat{i},\hat{j}}$ according to $\prec$, and select

$$\tilde{q}_{\sigma,\tau} \stackrel{\text{def}}{=} \min_{\prec}\{\tilde{q}_{s,t} : s + t < m_{\hat{i},\hat{j}} \text{ and } \tilde{q}_{s,t} \neq 0\} \tag{15}$$

Without going into detail, it can be shown that the coefficient of the monomial $X^\sigma Y^\tau$ in $Q(X + x_{\hat{i}}, Y + y_{\hat{i},\hat{j}})$, where (σ,τ) are given by (15), cannot be zero. Hence, $Q(X,Y)$ does not pass through the point $(x_{\hat{i}}, y_{\hat{i},\hat{j}})$ with multiplicity $m_{\hat{i},\hat{j}}$. This completes the proof of Theorem 2.

THEOREM 4. *The polynomial $Q(X,Y)$ defined in (4) passes through all the points $(x_{\tilde{i}}, y_{\tilde{i},\tilde{j}}) \in \mathbf{E}$ with multiplicities $m_{\tilde{i},\tilde{j}}$ if and only if $\{Q_t(X)\}_{0\leqslant t\leqslant l}$ satisfies Condition 2.*

The proof of this theorem is very similar to that of Theorem 2, and is therefore omitted. We can now combine Theorem 2 and Theorem 4 into the following statement.

THEOREM 5. *The polynomial $Q(X,Y)$ defined in (4) passes through all the points in the set $\mathbf{I}$ and all the points in the set $\mathbf{E}$ with the required multiplicities, if and only if $\{Q_t(X)\}_{0\leqslant t\leqslant l}$ satisfies both Condition 1 and Condition 2.*

Next, we establish a simple relationship between the $(1, k-1)$-weighted degree of the polynomial $Q(X,Y)$ defined in (4) and the $(1,-1)$-weighted degree of the polynomial $\Phi(X,Z)$ defined in (5). We have

$$\deg_{1,k-1} Q(X,Y) \;=\; \deg W(X) + \deg_{1,-1} \Phi(X,Z) \tag{16}$$

where $W(X)$ is defined by (11). This follows from the fact that $Q(X,Y) = W(X)\Phi(X,Z)|_{Z=\frac{Y}{V(X)}}$ and $\deg V(X) = k$. We are now in a position to prove our main result in this section.

THEOREM 6. *The set of polynomials $\{Q_t(X)\}_{0\leqslant t\leqslant l}$ which satisfies Conditions 1, 2 and 3 (generated by reduced interpolation on $\hat{\mathbf{I}}$ and $\hat{\mathbf{E}}$) is optimal in the sense that the corresponding polynomial $Q(X,Y)$ passes through all the points in $\mathbf{R}$, $\mathbf{I}$, and $\mathbf{E}$ with the required multiplicities and has the minimum possible $(1, k-1)$-weighted degree.*

Proof. From Theorem 5, we know that $Q(X,Y)$, which relates to $\Phi(X,Z)$ by the set $\{Q_t(X)\}_{0\leqslant t\leqslant l}$, passes through all the points in $\mathbf{I}$ and $\mathbf{E}$ with the required multiplicities. It is easy to see that any polynomial that can be written in the form (4) passes through all the points in $\mathbf{R}$ with the required multiplicities. Hence, what remains to be proved is that $Q(X,Y)$ has the minimum possible $(1,k-1)$-weighted degree. Let us assume to the contrary that there exists a $P(X,Y)$ with

$$\deg_{1,k-1} P(X,Y) \;<\; \deg_{1,k-1} Q(X,Y) \tag{17}$$

that passes through all the points in $\mathbf{R}$, $\mathbf{I}$, and $\mathbf{E}$ with the required multiplicities. By Lemma 1, (4), and (11), the polynomial $P(X,Y)$ can be expressed as

$$P(X,Y) \;=\; \sum_{t=0}^{l} P_t(X) \prod_{(i,j)\in\mathbf{I_R}} (X - x_i)^{m_{i,j}-t} T_t(X) Y^t \;=\; W(X) \sum_{t=0}^{l} P_t(X) T_t(X) \frac{Y^t}{V(X)^t}$$

Note that in the above equation, the upper limit of the sum does not have to be the same l as in the expression of $Q(X,Y)$; however, this does not affect the proof of the theorem. From the "only if" direction of Theorem 5, we know that the set of polynomials $\{P_t(X)\}_{0\leqslant t\leqslant l}$ satisfies Conditions1 and 2. Let us define $\Psi(X,Z) = \sum_{t=0}^{l} P_t(X)T_t(X)Z^t$. By (16), we have that $\deg_{1,-1} \Psi(X,Z) = \deg_{1,k-1} P(X,Y) - \deg W(X)$. Combining this with (17) yields

$$\deg_{1,-1} \Psi(X,Z) \;<\; \deg_{1,-1} \Phi(X,Z)$$

However, this contradicts the fact that $\Phi(X,Z)$ satisfies Condition 3. Hence, the polynomial $Q(X,Y)$, determined from the set $\{Q_t(X)\}_{0\leqslant t\leqslant l}$ generated by the re-encoding approach, has the least possible $(1,k-1)$-weighted degree. ∎

IV. Reduced factorization

As proved in [4], if the "score condition" is satisfied, then $Q(X,Y)$ has a factor $Y - f(X)$, where $f(X)$ evaluates to a!codeword $\underline{c} \in \mathbb{C}$. In this case, the polynomial $Q(X,Y)$ can be written as

$$Q(X,Y) \;=\; \big(Y - f(X)\big) D(X,Y) \tag{18}$$

Now let us write $Q(X,Y)$ in the same way as the first equality in (12), and divide both sides of (18) by $W(X)$. Thus

$$\sum_{t=0}^{l} Q_t(X) T_t(X) \left(\frac{Y}{V(X)}\right)^t \;=\; \frac{Y - f(X)}{V(X)} \, \frac{D(X,Y)V(X)}{W(X)} \tag{19}$$

By a change of variable $Z = \frac{Y}{V(X)}$, we can rewrite (19) as follows

$$\sum_{t=0}^{l} Q_t(X) T_t(X) Z^t \;=\; \left(Z - \frac{f(X)}{V(X)}\right) \frac{D(X, ZV(X))V(X)}{W(X)} \tag{20}$$

We see that the left-hand side of the above equation is precisely the polynomial $\Phi(X,Z)$ as defined in (5). This observation leads to the following result.

LEMMA 7. *If the polynomial $Q(X,Y)$ has a factor $Y - f(X)$, then the polynomial $\Phi(X,Z)$ has a factor $Z - \frac{\Omega(X)}{\Lambda(X)}$, where $\Lambda(X)$ is the error-locator polynomial for the re-encoded codeword.*

Proof. From (20), we see that $\Phi(X,Z)$ has a factor $Z - (\Omega(X)/\Lambda(X))$. Let us assume that ν correctable errors occurred among the k re-encoded positions of the Reed-Solomom codeword and denote the indices of these positions by

$$\mathbf{I_F} \;\stackrel{\text{def}}{=}\; \{i : \text{position } i \text{ is in error and } i \in \mathbf{S}\}$$

Clearly, $f(X)$ evaluates to zero at all those re-encoded positions that are not in error. Thus $f(X)$ can be written as

$$f(X) \;=\; \Omega(X) \prod_{i \in \mathbf{I_R} \setminus \mathbf{I_F}} (X - x_i) \tag{21}$$

We can also write $V(X)$ as follows

$$V(X) \;=\; \prod_{i \in \mathbf{I_F}} (X - x_i) \prod_{i \in \mathbf{I_R} \setminus \mathbf{I_F}} (X - x_i) \tag{22}$$

From (21) and (22), we have

$$\frac{f(X)}{V(X)} \;=\; \frac{\Omega(X)}{\prod_{i \in \mathbf{I_R} \setminus \mathbf{I_F}} (X - x_i)}$$

We can see that the denominator of the right-hand side of this above equation can be identified as the error-locator polynomial $\Lambda(X)$, whose roots are precisely the X values of the error locations. ∎

It follows that the polynomials $\Omega(X)$ and $\Lambda(X)$ can be obtained by the Berlekamp-Massey algorithm. The error magnitudes at the error locations can be computed by evaluating $f(X)$ at the X values of these locations, since the re-encoding implies that $f(x_i) + e_i = 0$ for $i \in \mathbf{I_F}$. Finally, we can apply re-encoding again to recover the transmitted codeword.

References

[1] G.-L. Feng and X. Giraud, "Fast algorithm in Sudan decoding procedure for Reed-Solomon codes," preprint, August 2002.

[2] W.J. Gross, F.R. Kschsichang, R. Koetter, and P.G. Gulak, "A VLSI architecture for interpolation in soft-decision list decoding of Reed-Solomon codes," preprint, August 2002.

[3] V. Guruswami and M. Sudan, "Improved decoding of Reed-Solomon and algebraic-geometric codes," *IEEE Trans. Inform. Theory*, vol. 45, pp. 1755–1764, September 1999.

[4] R. Koetter and A. Vardy, "Algebraic soft-decision decoding of Reed-Solomon codes," *IEEE Trans. Inform. Theory*, submitted, September 2001.

[5] R. Koetter, J. Ma, A. Vardy, and A. Ahmed, "On the Equivalency of reduced interpolation and original interpolation in soft-decision decoding of Reed-Solomon codes," manuscript in preparation, October 2002.

[6] R.R. Nielsen and T. Høholdt, "Decoding Reed-Solomon codes beyond half the minimum distance," preprint, December 1998.

[7] R.M. Roth and G. Ruckenstein, "Efficient decoding of Reed-Solomon codes beyond half the minimum distance", *IEEE Trans. Inform. Theory*, vol. 46, pp. 246–258, January 2000.

[8] M. Sudan, "Decoding of Reed-Solomon codes beyond the error correction bound," *Journal of Complexity*, vol. 12, pp. 180–193, 1997.

[9] X-W. Wu and P.H. Siegel, "Efficient root-finding algorithm with application to list decoding of algebraic-geometric codes," *IEEE Trans. Inform. Theory*, vol. 47, pp. 2579–2587, September 2001.