

A Complexity Reducing Transformation in Algebraic List Decoding of Reed-Solomon Codes

Ralf Koetter

Coordinated Science Laboratory
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
1308 West Main Street, Urbana, IL 61801

Alexander Vardy

Department of Electrical and Computer Engineering
Department of Computer Science and Engineering
University of California San Diego
9500 Gilman Drive, La Jolla, CA 92093

Abstract — The main computational steps in algebraic soft-decoding, as well as Sudan-type list-decoding, of Reed-Solomon codes are interpolation and factorization. A series of transformations is given for the interpolation problem that arises in these decoding algorithms. These transformations reduce the space and time complexity to a small fraction of the complexity of the original interpolation problem. A factorization procedure that applies directly to the reduced interpolation problem is also presented.

I. INTRODUCTION

Reed-Solomon (RS) codes are the most widely used error-correcting codes in digital communications and data storage. Recently, major breakthroughs have been achieved in improving the error-correction capability of Reed-Solomon codes. Sudan [13] and Guruswami-Sudan [4] have discovered a *list-decoding* algorithm, which can correct up to $n - \sqrt{nk}$ symbol errors. Their methods were later extended by Koetter and Vardy [6] to an algebraic *soft-decision decoding* algorithm for Reed-Solomon codes. Both list-decoding and algebraic soft-decision decoding use interpolation and factorization of bivariate polynomials, which is much more computationally intensive than hard-decision decoding. Fast interpolation and factorization algorithms have been studied in [2, 8, 11] among other papers. While polynomial-time, these algorithms fall short of making the required computation feasible in practical applications, involving long high-rate Reed-Solomon codes.

Here, we present a series of transformations that drastically reduce the space and time complexity of the interpolation process, by a factor of at least $n^2/(n-k)^2$. The main goal of this paper is to give a streamlined formulation of this transformation process and of the corresponding factorization procedure.

II. BACKGROUND AND PRELIMINARY RESULTS

Let \mathbb{F}_q be the finite field with q elements. The ring of polynomials over \mathbb{F}_q is denoted $\mathbb{F}_q[X]$. Reed-Solomon codes are obtained by evaluating certain subspaces of $\mathbb{F}_q[X]$ in a set of points $\mathcal{D} = \{x_1^*, x_2^*, \dots, x_n^*\} \subseteq \mathbb{F}_q$. Specifically, the RS code $\mathbb{C}_q(n, k)$ of length n and dimension k is defined as follows:

$$\mathbb{C}_q(n, k) \stackrel{\text{def}}{=} \{(f(x_1^*), \dots, f(x_n^*)) : x_1^*, \dots, x_n^* \in \mathcal{D}, f(X) \in \mathbb{F}_q[X], \deg f(X) < k\} \quad (1)$$

The point set \mathcal{D} is usually taken as \mathbb{F}_q or as \mathbb{F}_q^* , where \mathbb{F}_q^* is the set of all the nonzero elements of \mathbb{F}_q . Unless stated otherwise, we shall henceforth assume that $\mathcal{D} = \mathbb{F}_q^*$, so that $n = q-1$.

Let $K_{\alpha, \beta}$ denote the ring of rational functions in $\mathbb{F}_q(X, Y)$ without poles at the point $(\alpha, \beta) \in \mathbb{F}_q \times \mathbb{F}_q$. A rational function $\mathcal{A}(X, Y) \in K_{\alpha, \beta}$ has a power-series expansion in basis functions of type $(X - \alpha)^i (Y - \beta)^j$. Thus

$$\mathcal{A}(X, Y) = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} a_{i,j} (X - \alpha)^i (Y - \beta)^j \quad (2)$$

The function $\mathcal{A}(X, Y)$ is said to *pass through the point* (α, β) with *multiplicity* m if $a_{i,j} = 0$ for all $i + j < m$ in (2). Define the *multiplicity function* $\mu_{\alpha, \beta} : K_{\alpha, \beta} \rightarrow \mathbb{N}$ as follows:

$$\mu_{\alpha, \beta}(\mathcal{A}(X, Y)) \stackrel{\text{def}}{=} \max\{m \in \mathbb{N} : a_{i,j} = 0 \ \forall i + j < m\}$$

where $a_{i,j}$ are the coefficients in (2) and \mathbb{N} is the set of natural numbers. The following observation is obvious: for any two functions $\mathcal{A}(X, Y)$ and $\mathcal{B}(X, Y)$ in $K_{\alpha, \beta}$, we have

$$\mu_{\alpha, \beta}(\mathcal{A}\mathcal{B}) = \mu_{\alpha, \beta}(\mathcal{A}) + \mu_{\alpha, \beta}(\mathcal{B}) \quad (3)$$

Fix a polynomial $g(X)$ of degree k over \mathbb{F}_q . Let \mathcal{Z} be the set of $\alpha \in \mathbb{F}_q$ such that $g(\alpha) = 0$. Consider the mapping $\varphi_g : (\mathbb{F}_q - \mathcal{Z}) \times \mathbb{F}_q \rightarrow \mathbb{F}_q \times \mathbb{F}_q$ defined by $\varphi_g(x, y) = (x, y/g(x))$. The inverse mapping is given by $\varphi_g^{-1}(x, z) = (x, zg(x))$. It is easy to see that φ_g and φ_g^{-1} are birational isomorphisms. We present the following results without proof. Similar statements in a more general context of birational isomorphisms defined on an open subset of plane curves can be found in [12].

Lemma 1. Let $\mathcal{A}(X, Y)$ be a rational function, and let \mathcal{X} be the set of solutions to $\mathcal{A}(X, Y) = 0$ in $(\mathbb{F}_q - \mathcal{Z}) \times \mathbb{F}_q$. Then there exists a rational function $\mathcal{B}(X, Y)$ such that $\varphi_g(\mathcal{X})$ is the set of solutions to $\mathcal{B}(X, Y) = 0$ in $\mathbb{F}_q \times \mathbb{F}_q$.

We say that $\mathcal{B}(X, Y)$ is the image of $\mathcal{A}(X, Y)$ under φ_g , and write $\mathcal{B} = \varphi_g(\mathcal{A})$, thereby extending φ_g to rational functions.

Lemma 2. For all points $(\alpha, \beta) \in (\mathbb{F}_q - \mathcal{Z}) \times \mathbb{F}_q$, and for all $\mathcal{A}(X, Y) \in K_{\alpha, \beta}$, we have $\mu_{\alpha, \beta}(\mathcal{A}) = \mu_{\varphi_g(\alpha, \beta)}(\varphi_g(\mathcal{A}))$.

As in [4, 6, 8, 11], we define the weighted degree of a polynomial as follows. Let $\mathcal{A}(X, Y) = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} a_{i,j} X^i Y^j$ be a bivariate polynomial over \mathbb{F}_q and let w_X, w_Y be real numbers. Then the (w_X, w_Y) -weighted degree of $\mathcal{A}(X, Y)$, denoted $\deg_{w_X, w_Y} \mathcal{A}(X, Y)$, is defined as the maximum over all real numbers $i w_X + j w_Y$ such that $a_{i,j} \neq 0$. For reasons that will soon become clear, we do not restrict the definition of weighted degree to the usual case [4, 6, 8] where w_X, w_Y are nonnegative integers. Thus the weighted degree of a polynomial $\mathcal{A}(X, Y)$ can assume negative values.

III. THE INTERPOLATION PROBLEM

Our interest in the foregoing definitions and results is motivated by the fact that, as a consequence of Bezout's theorem [12], two polynomials $\mathcal{A}(X, Y)$ and $\mathcal{B}(X, Y)$ cannot both pass through an arbitrary large number of points without having a common factor. In particular, the polynomial $Y - f(X)$, with $\deg f(X) < k$, passes through the n points $(x_1^*, c_1), (x_2^*, c_2), \dots, (x_n^*, c_n)$, where $c_i = f(x_i^*)$ may be thought of as the n transmitted symbols. Then Bezout's theorem implies that any nonzero polynomial $Q(X, Y)$ such that

$$\sum_{i=1}^n \mu_{x_i^*, c_i}(Q) > \deg_{1, k-1} Q(X, Y)$$

is divisible by $Y - f(X)$. This leads to the interpolation-based decoding algorithms of [4, 6, 8, 13]. The central idea of all these decoding algorithms is to construct a polynomial $Q(X, Y)$ that passes through a prescribed set of points $\mathcal{P} = \{(x_1, y_1), (x_2, y_2), \dots, (x_s, y_s)\}$, where $x_1, \dots, x_s \in \mathcal{D}$ and $y_1, y_2, \dots, y_s \in \mathbb{F}_q$, with prescribed multiplicities $m_{x_1, y_1}, m_{x_2, y_2}, \dots, m_{x_s, y_s} \in \mathbb{N}$. If these points and multiplicities agree "sufficiently well" with the n points (x_i^*, c_i) that define the transmitted codeword, then the divisibility of $Q(X, Y)$ by $Y - f(X)$ is guaranteed [6]. How should the two sets \mathcal{P} and $M = \{m_{x_1, y_1}, m_{x_2, y_2}, \dots, m_{x_s, y_s}\}$ be determined from the channel output? Various answers to this question can be found in [4, 6, 7, 9, 10]. In all cases, a key part of the decoding algorithm consists of solving the following interpolation problem.

Original interpolation problem: *Given a set of points $\mathcal{P} = \{(x_1, y_1), (x_2, y_2), \dots, (x_s, y_s)\}$ and a set of multiplicities $M = \{m_{x_1, y_1}, m_{x_2, y_2}, \dots, m_{x_s, y_s}\}$, find a nonzero polynomial $Q(X, Y)$ of minimal $(1, k-1)$ -weighted degree, such that $\mu_{x_i, y_i}(Q) \geq m_{x_i, y_i}$ for $i = 1, \dots, s$.*

We shall refer to this interpolation problem as $\mathbf{IP}_{1, k-1}(\mathcal{P}, M)$, and say that $Q(X, Y)$ is a *solution to $\mathbf{IP}_{1, k-1}(\mathcal{P}, M)$* . Observe that x_i and x_j in the set $\mathcal{P} = \{(x_1, y_1), (x_2, y_2), \dots, (x_s, y_s)\}$ do not have to be distinct, all we require is that they belong to \mathcal{D} . In fact, in soft-decision decoding, we often interpolate through different points having the same X -coordinate [5, 6].

By definition, requiring that a polynomial $Q(X, Y)$ passes through a point with multiplicity m imposes $\frac{1}{2}m(m+1)$ linear constraints on the vector space of polynomials in two variables (cf. (2)). Hence, solving $\mathbf{IP}_{1, k-1}(\mathcal{P}, M)$ is tantamount to solving a system of $N(M) = \frac{1}{2} \sum_{i=1}^s m_{x_i, y_i}(m_{x_i, y_i} + 1)$ linear (although not necessarily linearly independent) equations. As shown in [4, 6], there are

$$\nu_{1, k-1}(\delta) = \left\lceil \frac{\delta+1}{k-1} \right\rceil \left(\delta - \frac{k-1}{2} \left\lfloor \frac{\delta}{k-1} \right\rfloor + 1 \right)$$

monomials $X^i Y^j$ with $i + (k-1)j \leq \delta$. Hence, choosing δ to be large enough will guarantee a solution to $\mathbf{IP}_{1, k-1}(\mathcal{P}, M)$. Let δ^* be the smallest integer such that $\nu_{1, k-1}(\delta^*) > N(M)$. Then $\deg_{1, k-1} Q(X, Y) \leq \delta^*$, and the Y -degree of $Q(X, Y)$ can be estimated as $r = \lfloor \delta^*/(k-1) \rfloor$.

In principle, $\mathbf{IP}_{1, k-1}(\mathcal{P}, M)$ is a simple linear problem that can be solved in a number of ways. Fast algorithms for this purpose can be found in [1–3, 8]. These algorithms compute $Q(X, Y)$ in time $O(rN^2)$, where $N = N(M)$ is the

total number of linear equations. While this is substantially faster than straightforward Gaussian elimination, the problem is that the number of equations N is often too large to make an $O(rN^2)$ computation feasible in practice. The following example sheds some light on the magnitude of this problem.

Example 1. Let $\mathbb{C}_q(n, k)$ be a Reed-Solomon code of length $n = 255$ and dimension $k = 239$ over \mathbb{F}_{256} . A typical interpolation problem arising in the algebraic soft-decision decoding [6] of $\mathbb{C}_q(n, k)$ might involve the following multiplicities:

multiplicity	10	9	8	7	6	5	4	3	2	1
# of points	236	6	5	2	4	3	2	3	5	4

for a total of $N = 13,807$ linear equations. The corresponding value of δ^* is 2446, and the required Y -degree of $Q(X, Y)$ is $r = 10$. Computing $Q(X, Y)$ with the fast algorithms of [1–3, 8] thus takes about $1.9 \cdot 10^9$ finite-field operations. \square

This example illustrates a major problem with interpolation-based decoding. While, for a fixed maximal multiplicity, the complexity of decoding is bounded by a polynomial in the length of the code, the actual complexity of computing $Q(X, Y)$ is prohibitively large in practice. In the next section, we will drastically reduce this complexity.

IV. A COMPLEXITY REDUCING TRANSFORMATION

Rather than seeking an efficient way to solve $\mathbf{IP}_{1, k-1}(\mathcal{P}, M)$, we will modify the interpolation problem itself, by means of a shift and a coordinate transformation. Our approach is similar to the re-encoding idea of the Berlekamp-Welch [14].

Given the set $\mathcal{P} = \{(x_1, y_1), (x_2, y_2), \dots, (x_s, y_s)\}$, we will identify some k points $(x_{i_1}, y_{i_1}), (x_{i_2}, y_{i_2}), \dots, (x_{i_k}, y_{i_k})$ in \mathcal{P} such that $x_{i_1}, x_{i_2}, \dots, x_{i_k} \in \mathcal{D}$ are all distinct. Define $\mathcal{R} = \{(x_{i_1}, y_{i_1}), (x_{i_2}, y_{i_2}), \dots, (x_{i_k}, y_{i_k})\}$. Observe that if \mathcal{P} contains at most $n - e < k$ points with distinct X -coordinates, then the resulting polynomial $Q(X, Y)$ will have at least $q^{e-(n-k)}$ factors of type $Y - f(X)$. This situation corresponds to $e > n - k$ erasures, in which case the transmitted codeword cannot be uniquely determined. This shows that, unless the interpolation problem $\mathbf{IP}_{1, k-1}(\mathcal{P}, M)$ is ill-conditioned by too many erasures, a set \mathcal{R} with the required property always exists. In fact, there will usually be exponentially many ways to choose \mathcal{R} from \mathcal{P} . As far as the theory developed in this paper, the choice of \mathcal{R} is arbitrary. In practice, the set \mathcal{R} will be chosen to consist of the points with the highest possible multiplicities (cf. Example 2). To simplify notation in what follows, we assume without loss of generality that \mathcal{R} consists of the first k points of \mathcal{P} , that is $\mathcal{R} = \{(x_1, y_1), (x_2, y_2), \dots, (x_k, y_k)\}$.

The set $\mathcal{R} = \{(x_1, y_1), (x_2, y_2), \dots, (x_k, y_k)\}$ determines the *re-encoding polynomial* $h(X)$ of degree $< k$, defined by

$$h(x_i) = y_i \quad \text{for all } (x_i, y_i) \in \mathcal{R} \quad (4)$$

Note that the codeword \underline{c}' obtained by evaluating $h(X)$ at $x_1^*, x_2^*, \dots, x_n^*$ agrees with the "given" values y_1, y_2, \dots, y_k at the k positions corresponding to x_1, x_2, \dots, x_k . Thus computing $h(X)$ is equivalent to re-encoding through k given values at some k positions. If these k positions are consecutive and $\mathbb{C}_q(n, k)$ is cyclic, this can be achieved through division by

the generator polynomial for $\mathbb{C}_q(n, k)$. Otherwise, such re-encoding is tantamount to correcting $n-k$ erasures in $\mathbb{C}_q(n, k)$. Various efficient algorithms for this purpose are known.

Given the set $\mathcal{P} = \{(x_1, y_1), (x_2, y_2), \dots, (x_s, y_s)\}$ and the re-encoding polynomial $h(X)$, we define

$$\mathcal{P}' \stackrel{\text{def}}{=} \{(x_1, y_1 - h(x_1)), \dots, (x_k, y_s - h(x_s))\} \quad (5)$$

Notice that, by the definition of $h(X)$ in (4), the first k points in \mathcal{P}' are of the form $(x_1, 0), (x_2, 0), \dots, (x_k, 0)$.

Theorem 3. *Let $\mathcal{Q}'(X, Y)$ be a solution to $\mathbf{IP}_{1, k-1}(\mathcal{P}', M)$. Then $\mathcal{Q}'(X, Y - h(X))$ is a solution to $\mathbf{IP}_{1, k-1}(\mathcal{P}, M)$.*

Sketch of proof. Consider an arbitrary point $(\alpha, \beta) \in \mathcal{P}$. By (2), (5), and the definition of $\mathbf{IP}_{1, k-1}(\mathcal{P}', M)$, the polynomial $\mathcal{Q}'(X, Y)$ can be written as

$$\mathcal{Q}'(X, Y) = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} q'_{i,j} (X - \alpha)^i (Y - (\beta - h(\alpha)))^j$$

where $q'_{i,j} = 0$ for all $i + j < m_{\alpha, \beta}$. Since $h(X) - h(\alpha)$ vanishes at α , the function $h_{\alpha}(X) = (h(X) - h(\alpha))/(X - \alpha)$ is a polynomial. Let $\mathcal{Q}(X, Y) = \mathcal{Q}'(X, Y - h(X))$. Then

$$\begin{aligned} \mathcal{Q}(X, Y) &= \sum_{i,j=0}^{\infty} q'_{i,j} (X - \alpha)^i (Y - h(X) - (\beta - h(\alpha)))^j \\ &= \sum_{i,j=0}^{\infty} q'_{i,j} (X - \alpha)^i ((Y - \beta) - (h(X) - h(\alpha)))^j \\ &= \sum_{i,j=0}^{\infty} q'_{i,j} (X - \alpha)^i ((Y - \beta) - (X - \alpha)h_{\alpha}(X))^j \end{aligned}$$

Since $q'_{i,j} = 0$ for all $i + j < m_{\alpha, \beta}$, each *nonzero* term above passes through the point (α, β) with multiplicity at least $i + j \geq m_{\alpha, \beta}$. Therefore, for each $(\alpha, \beta) \in \mathcal{P}$, the polynomial $\mathcal{Q}(X, Y) = \mathcal{Q}'(X, Y - h(X))$ passes through the point (α, β) with multiplicity at least $m_{\alpha, \beta}$. It is easy to see that $\deg_{1, k-1} \mathcal{Q}(X, Y) = \deg_{1, k-1} \mathcal{Q}'(X, Y)$. Hence, the minimality of $\mathcal{Q}'(X, Y)$ implies the minimality of $\mathcal{Q}(X, Y)$. ■

Let the symbol $[\cdot]^+$ be defined as $[i]^+ = \max\{i, 0\}$. We can now proceed with the complexity reducing transformations.

Lemma 4. *The polynomial $\mathcal{A}(X, Y) = \sum_{j=0}^{\infty} a_j(X)Y^j$ passes through a point $(\alpha, 0)$ with multiplicity m if and only if the univariate polynomials $a_j(X)$ are divisible by $(X - \alpha)^{[m-j]^+}$.*

Proof. Expand $a_j(X)$ in the basis functions $(X - \alpha)^j$, that is write $a_j(X)$ as $a_j(X) = \sum_{i=0}^{\infty} a_{i,j}(X - \alpha)^i$. Then the expansion (2) of $\mathcal{A}(X, Y)$ at the point $(\alpha, 0)$ is given by

$$\mathcal{A}(X, Y) = \sum_{i,j=0}^{\infty} a_{i,j}(X - \alpha)^i (Y - 0)^j = \sum_{i,j=0}^{\infty} a_{i,j}(X - \alpha)^i Y^j$$

Clearly, the polynomial $a_j(X)$ is divisible by $(X - \alpha)^{[m-j]^+}$ if and only if $a_{i,j} = 0$ for all $i < [m-j]^+$. This is just a reformulation of the definition of multiplicity. ■

Corollary 5. *The polynomial $\mathcal{A}(X, Y) = \sum_{j=0}^{\infty} a_j(X)Y^j$ passes through the k points $(x_1, 0), (x_2, 0), \dots, (x_k, 0)$ with multiplicities $m_{x_1, y_1}, m_{x_2, y_2}, \dots, m_{x_k, y_k}$ if and only if all the polynomials $a_j(X)$ are divisible by $\prod_{i=1}^k (X - x_i)^{[m_{x_i, y_i} - j]^+}$.*

Now, let the auxiliary polynomials $g(X)$, $\Phi(X)$, and the “tail” polynomials $T_j(X)$ be defined as follows:

$$g(X) \stackrel{\text{def}}{=} \prod_{i=1}^k (X - x_i) \quad (6)$$

$$\Phi(X) \stackrel{\text{def}}{=} \prod_{i=1}^k (X - x_i)^{m_{x_i, y_i}} \quad (7)$$

$$T_j(X) \stackrel{\text{def}}{=} \prod_{i=1}^k (X - x_i)^{[j - m_{x_i, y_i}]^+} \quad \text{for } j = 0, 1, \dots, r$$

where $r = \lfloor \delta^*/(k-1) \rfloor$ is the Y -degree of $\mathcal{Q}(X, Y)$, as defined in Section III. From Corollary 5, we know that the solution $\mathcal{Q}'(X, Y)$ to the interpolation problem $\mathbf{IP}_{1, k-1}(\mathcal{P}', M)$ must have the form

$$\begin{aligned} \mathcal{Q}'(X, Y) &= \sum_{j=0}^r \left(b_j(X) \prod_{i=1}^k (X - x_i)^{[m_{x_i, y_i} - j]^+} \right) Y^j \\ &= \Phi(X) \sum_{j=0}^r b_j(X) T_j(X) \left(\frac{Y}{g(X)} \right)^j \end{aligned} \quad (8)$$

for some polynomials $b_j(X)$. Computing $\mathcal{Q}'(X, Y)$ and thereby solving both $\mathbf{IP}_{1, k-1}(\mathcal{P}', M)$ and $\mathbf{IP}_{1, k-1}(\mathcal{P}, M)$ (in view of Theorem 3) thus reduces to finding $b_j(X)$. The following two propositions show that computing $b_j(X)$ is tantamount to solving a much smaller interpolation problem!

Proposition 6. *Let $(\alpha, \beta) \in \mathcal{P}'$ be such that $g(\alpha) \neq 0$. Then $\mathcal{Q}'(X, Y)$ passes through (α, β) with multiplicity m if and only if the polynomial $\mathcal{Q}''(X, Z) = \sum_{j=0}^r b_j(X) T_j(X) Z^j$ passes through the point $(\alpha, \beta/g(\alpha))$ with multiplicity m .*

Sketch of proof. We require that $\mu_{\alpha, \beta}(\mathcal{Q}'(X, Y)) = m$. Note that in view of (6) and (7), we have $\mu_{\alpha, \beta}(\Phi(X)) = 0$ whenever $g(\alpha) \neq 0$. Hence it follows from (8) and (3) that $\mu_{\alpha, \beta}(\mathcal{Q}''(X, Y/g(X)))$ must be equal to m . We now consider the birational mapping $\varphi_g(x, y) = (x, y/g(x))$ with inverse $\varphi_g^{-1}(x, z) = (x, zg(x))$. By assumption $g(\alpha) \neq 0$, so the mapping $\varphi_g(x, y)$ is well-defined at $(x, y) = (\alpha, \beta)$. The proposition now follows immediately from Lemma 2. ■

Proposition 6 works for those points $(\alpha, \beta) \in \mathcal{P}'$ for which $g(\alpha) \neq 0$. The next proposition achieves the same goal for the case $g(\alpha) = 0$. We present this proposition without proof.

Proposition 7. *Let $(\alpha, \beta) \in \mathcal{P}'$ be such that $g(\alpha) = 0$ while $\beta \neq 0$ (if $\beta = 0$, then (α, β) is among the first k points of \mathcal{P}'). Let*

$$\mathcal{Q}''_{\alpha}(X, Z) \stackrel{\text{def}}{=} \sum_{j=0}^r b_j(X) (X - \alpha)^{m_{\alpha, \beta} - j} T_j(X) Z^j$$

Then the polynomial $\mathcal{Q}'(X, Y)$ passes through the point (α, β) with multiplicity m if and only if $\mathcal{Q}''_{\alpha}(X, Z)$ passes through the point $(\alpha, \beta/g'(\alpha))$ with multiplicity m , where $g'(X)$ denotes the formal derivative of $g(X)$ in (6).

Propositions 6 and 7 are the cornerstone of our complexity reducing transformation. We can now summarize all of the above in terms of the *reduced interpolation problem* below.

V. THE FACTORIZATION PROCEDURE

Reduced interpolation problem: Suppose we are given a set of points $\mathcal{P}' = \{(x_1, y_1), (x_2, y_2), \dots, (x_s, y_s)\}$, such that $y_1 = y_2 = \dots = y_k = 0$ and x_1, x_2, \dots, x_k are all distinct. We are furthermore given a set of associated multiplicities $M = \{m_{x_1, y_1}, m_{x_2, y_2}, \dots, m_{x_s, y_s}\}$. Let the polynomials $h(X)$, $g(X)$, and $T_j(X)$ be defined follows:

- $h(X)$ is the least degree polynomial such that $h(x_i) = y_i$ for $i = 1, 2, \dots, k$.
- $T_j(X) = \prod_{i=1}^k (X - x_i)^{[j - m_{x_i, y_i}]^+}$
- $g(X) = \prod_{i=1}^k (X - x_i)$

Then the reduced interpolation problem consists of finding a nonzero polynomial $Q''(X, Z) = \sum_{j=0}^r b_j(X) T_j(X) Z^j$ of minimal $(1, -1)$ -weighted degree, such that for all $(x_{k+1}, y_{k+1}), (x_{k+2}, y_{k+2}), \dots, (x_s, y_s)$, we have

◊ if $g(x_i) \neq 0$, then

$$\mu_{x_i, \frac{y_i}{g(x_i)}}(Q''(X, Z)) \geq m_{x_i, y_i}$$

◊ if $g(x_i) = 0$, then

$$\mu_{x_i, \frac{y_i}{g'(x_i)}}\left((X - x_i)^{m_{x_i, y_i}} Q''\left(X, \frac{Z}{X - x_i}\right)\right) \geq m_{x_i, y_i}$$

where $g'(X)$ is the formal derivative of $g(X)$.

We shall refer to the reduced interpolation problem above as **RIP**_{1,-1}(\mathcal{P}' , M). The next theorem summarizes our results and establishes the connection between **RIP**_{1,-1}(\mathcal{P}' , M) and the original interpolation problem **IP**_{1,k-1}(\mathcal{P} , M).

Theorem 8. Let $Q''(X, Z)$ be a solution to **RIP**_{1,-1}(\mathcal{P}' , M). Then a solution to **IP**_{1,k-1}(\mathcal{P} , M) is given by

$$Q(X, Y) = \Phi(X) Q''\left(X, \frac{Y - h(X)}{g(X)}\right) \quad (9)$$

Proof. Follows from Theorem 3 and Propositions 6, 7. ■

Observe that the efficient interpolation algorithms of [1–3, 8] can be easily adapted to solve **RIP**_{1,-1}(\mathcal{P}' , M). While **RIP**_{1,-1}(\mathcal{P}' , M) appears to be more convoluted than the original problem **IP**_{1,k-1}(\mathcal{P} , M), its complexity is often orders of magnitude lower. This is due to the fact that we *do not even need to consider* the first k points of \mathcal{P}' in computing $Q''(X, Z)$. These k interpolation points (which are chosen to have the largest multiplicities) are effectively pre-solved!

Example 2. Consider again the situation of Example 1. Judiciously choosing the re-encoding set \mathcal{R} , we can eliminate the 236 interpolation points of multiplicity 10, plus another 3 points of multiplicity 9. In other words, rather than solving the 13,807 linear equations, we have reduced the problem to efficiently solving only 692 equations, which is a feasible task. This reduction in complexity by a factor of 400 is augmented by a corresponding reduction in memory requirements, due to the fact that the polynomials $b_j(X)$ have very small degree. □

The reductions in complexity obtained in Section IV would be less significant if we were forced to actually compute the original polynomial $Q(X, Y)$ (using (9), say) in order to find a factor of type $Y - f(X)$. Thus, rather than factoring $Q(X, Y)$, we will directly factor the much smaller polynomial $Q''(X, Z)$.

Suppose that $Q'(X, Y)$ contains a factor $Y - e(X)$. Note that due to re-encoding, $e(X)$ evaluates to zero in exactly those positions x_i where y_i was the transmitted symbol, for all $i = 1, 2, \dots, k$. Thus, with the substitution $Z = Y/g(X)$, a factor of type $Y - e(X)$ in $Q'(X, Y)$ translates into a factor of type $g(X)Z - e(X)$ in $Q''(X, Z)$. Using, say, the factorization procedure of [11] will reveal the power-series expansion of the rational function $e(X)/g(X)$. Upon canceling common terms, we can therefore determine a power-series expansion for the function $\omega(X)/\sigma(X)$, where $\sigma(X)$ is a classical error-locator polynomial, locating errors in positions x_i for $i = 1, 2, \dots, k$. The corresponding error values are found by observing that the function Z equals $\omega(X)/\sigma(X)$ on the curve at hand. This implies, under the birational morphism $\varphi_g^{-1}(x, z) = (x, zg(x))$, that $Y/g(X) = \omega(X)/\sigma(X)$. Thus the error values are given by the L'Hôpital rule as

$$y_i = \left. \frac{g'(X)\omega(X)}{\sigma'(X)} \right|_{x_i}$$

whenever $\sigma(x_i) = 0$. Note that $\omega(X)$, $\sigma(X)$ can be found from the power-series expansion of $\omega(X)/\sigma(X)$ by a Padé approximation procedure, such as the Berlekamp-Massey algorithm.

REFERENCES

- [1] A. Ahmed, R. Koetter, and N. Shanbhag, "VLSI architectures for soft-decision decoding of Reed-Solomon codes," *IEEE Trans. VLSI Systems*, submitted for publication, February 2003.
- [2] G.-L. Feng and X. Giraud, "Fast algorithm in Sudan decoding procedure for Reed-Solomon codes," preprint, August 2002.
- [3] W.J. Gross, F.R. Kschischang, R. Koetter, and P.G. Gulak, "Towards a VLSI architecture for interpolation-based soft-decision Reed-Solomon decoders," *J. VLSI Signal Processing*, submitted, January 2003.
- [4] 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.
- [5] R. Koetter, J. Ma, A. Vardy, and A. Ahmed, "Efficient interpolation and factorization in algebraic soft-decision decoding of Reed-Solomon codes," *IEEE Int. Symp. Inform. Theory*, Yokohama, Japan, July 2003.
- [6] R. Koetter and A. Vardy, "Algebraic soft-decision decoding of Reed-Solomon codes," *IEEE Trans. Inform. Theory*, submitted, August 2001.
- [7] R.R. Nielsen, "Decoding concatenated codes with Sudan's algorithm," *IEEE Trans. Inform. Theory*, submitted for publication, May 2000.
- [8] R.R. Nielsen and T. Høholdt, "Decoding Reed-Solomon codes beyond half the minimum distance," in *CODING THEORY, CRYPTOGRAPHY, AND RELATED AREAS*, (Buchmann et al., Eds.), pp. 221–236, 1999.
- [9] F. Parvaresh and A. Vardy, "Multiplicity assignments for algebraic soft-decoding of Reed-Solomon codes," *IEEE Int. Symp. Inform. Theory*, Yokohama, Japan, July 2003, submitted for presentation.
- [10] L. Pecquet, "Décodage en liste des codes géométriques," Ph.D. Thesis, Université Pierre et Marie Curie, Paris, France, 2002.
- [11] 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, 2000.
- [12] I.R. Shafarevich, *Basic Algebraic Geometry*. Springer-Verlag, 1995.
- [13] M. Sudan, "Decoding of Reed-Solomon codes beyond the error correction bound," *J. Complexity*, vol. 12, pp. 180–193, 1997.
- [14] L.R. Welch and E.R. Berlekamp, *Error correction for algebraic block codes*, U.S. Patent No. 4,633,470, issued December 30, 1986.