Divide-and-Conquer Interpolation for List Decoding of Reed-Solomon Codes

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

Peter Trifonov

Distributed Computing and Networking Department
St. Petersburg State Polytechnic University
Polytechnicheskaya 29, St. Petersburg 195251, Russia
petert@dcn.nord.nw.ru

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

Abstract

Algebraic soft-decision decoding of Reed-Solomon codes delivers promising coding gains over conventional hard-decision decoding. The most computationally demanding step in algebraic soft-decoding (as well as Sudan-type list-decoding) is bivariate polynomial interpolation. We discuss a divide-and-conquer algorithm that could potentially reduce the complexity of the interpolation step.

I. Introduction

Recently Sudan [14] and Guruswami-Sudan [5] introduced a powerful algorithm for list decoding of Reed-Solomon and some other codes. This list-decoding method was later extended by Koetter and Vardy [9] to an algebraic *soft-decision decoding* algorithm, which significantly outperforms hard-decision list-decoding. Both list-decoding and algebraic soft-decision rely on interpolation of bivariate polynomials, which is much more computationally complex than hard-decision decoding. The high complexity of the interpolation results from the large-degree polynomials manipulated in the iterative interpolation procedure, thus efficient implementations of the interpolation algorithm are desired.

Section II gives a brief review of the interpolation algorithm based on [3, 10]. In section III, we present a matrix interpretation and an algebraic geometry interpretation of the interpolation algorithm. The later leads to a divide-and-conquer algorithm.

II. Background

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 $\mathcal{D} = \{x_1^*, x_2^*, \ldots, x_n^*\} \subseteq \mathbb{F}_q$. Specifically, the RS code $\mathbb{C}_q(n,k)$ of length n and dimension k is defined as follows:

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

The interpolation step solves the following problem: Given a set of points $\mathcal{P} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_s, y_s)\}$ and a set of multiplicities $M = \{m_1, m_2, \ldots, m_s\}$, find a nonzero polynomial $\mathcal{Q}(X,Y)$ of minimal $(1, k-1)$-weighted degree, such that $\mathcal{Q}(X,Y)$ passes through points in $\mathcal{P}$ with prescribed multiplicities. Fast algorithm to solve the interpolation problem can be found in [3, 11]. And in this paper, we refer the algorithm as IIA (iterative interpolation algorithm).

The Iterative Interpolation Algorithm

- **Initialization:**
  $\mathcal{Q}_v(X,Y) = \sum_{t=0}^{l} q_{v,t}(X)Y^t$, for $0 \leqslant v \leqslant r$.
- **Iteration:**
  Input: $\{(x_i, y_i, m_{x_i,y_i}) : (x_i, y_i) \in \mathcal{P}\}$
  - For each triple (x_i, y_i, m_{x_i,y_i}),
    $O_v = \deg_{1,k-1} \mathcal{Q}_v(X,Y)$, for $0 \leqslant v \leqslant r$.
    for $a = 0$ to $m_{x_i,y_i} - 1$
      for $b = 0$ to $m_{x_i,y_i} - 1 - a$
        Discrepancy Computation:
        for $v = 0$ to r
          $d_v^{(a,b)} = \mathbf{coef}\big(\mathcal{Q}_v(X + x_i, Y + y_i), X^a Y^b\big)$
        end
        Polynomial Update:
        if there exist $\eta = \operatorname{argmin}_{\substack{0 \leqslant v \leqslant r \\ d_v^{(a,b)} \neq 0}} \{O_v\}$
          for $v = 0$ to r
            if $v \neq \eta$ and $d_v^{(a,b)} \neq 0$
              $\mathcal{Q}_v(X,Y) := \mathcal{Q}_v(X,Y) + \frac{d_v^{(a,b)}}{d_\eta^{(a,b)}} \mathcal{Q}_\eta(X,Y)$
            end
          end
          $\mathcal{Q}_\eta(X,Y) := \mathcal{Q}_\eta(X,Y)(X - x_i)$, and $O_\eta := O_\eta + 1$
        end
      end
    end

- **Result:** $\mathcal{Q}(X,Y) = \{\mathcal{Q}_\eta(X,Y)\}$, where $\eta = \operatorname{argmin}_{0 \leqslant v \leqslant r}\{O_v\}$. So that

$$\mathcal{Q}(X,Y) = \sum_{t=0}^{r} q_t(X)Y^t$$

The complexity of the algorithm can be estimated as $O(rC^2)$, where r is the largest Y-degree of the polynomials involved in the iterative interpolation algroithm and $C \stackrel{\text{def}}{=} \sum_{i=1}^{s} \binom{m_i+1}{2}$ is the total interpolation cost for a prescribed multiplicity set M.

III. The Divide-and-Conquer Approach

The complexity of the IIA is proportional to the number of polynomials being updated in the IIA and to 2nd power of the total cost of interpolation points. Thus one may expect to reduce the complexity if the original interpolation problem is split into a number of smaller ones. The following subsections describe an approach which is based on splitting the original set of interpolation points into a number of subsets, independent computation of their interpolation polynomials and their unionization.

A. Matrix interpretation of the interpolation algorithm

Lemma 1. *Let $Q_j(X,Y)$, $j = 0, ..., r$ be a set of polynomials, produced by the IIA after processing the set $\mathcal{P}$ of interpolation points with corresponding multiplicities M. Then all polynomials $\mathcal{Q}(X,Y) : \operatorname{wdeg}_{(0,1)} \mathcal{Q}(X,Y) \leqslant$*

r pass through points in $\mathcal{P}$ with corresponding multiplicities M can be represented as

$$\mathcal{Q}(X,Y) = \sum_{j=0}^{r} p_j(X)\mathcal{Q}_j(X,Y)$$

Proof. Let $\mathcal{Q}(X,Y) : \mathrm{wdeg}_{(0,1)}\,\mathcal{Q}(X,Y) \leqslant r$ be a polynomial having points from $\mathcal{P}$ as roots of corresponding multiplicities from M. The following procedure is very similar both to the multivariate polynomial division [2] and matrix polynomial division algorithms [4, 8].

1) Order $\mathcal{Q}(X,Y)$ terms accordingly to $\deg_{1,k-1}$ and let $R(X,Y) = 0, p_j(X) = 0$.
2) Let LT $\mathcal{Q}(X,Y) = \alpha x^a y^b$ and LT $\mathcal{Q}_b(X,Y) = x^c y^b$. (Note that the IIA guarantees that LT $\mathcal{Q}_b(X,Y) = X^c Y^b$ throughout the whole iterative procedure.) If $a \geqslant c$, then $p_b(X) := p_b(X) + \alpha x^{a-c}$, $\mathcal{Q}(X,Y) := \mathcal{Q}(X,Y) - \alpha x^{a-c}\mathcal{Q}_b(X,Y)$. Otherwise, $R(X,Y) := R(X,Y) + \text{LT}\,\mathcal{Q}(X,Y)$, $\mathcal{Q}(X,Y) := \mathcal{Q}(X,Y) - \text{LT}\,\mathcal{Q}(X,Y)$.
3) Repeat step 2 until $\mathcal{Q}(X,Y) = 0$.

Clearly, this procedure would lead to $\mathcal{Q}(X,Y) = \sum_j p_j(X)\mathcal{Q}_j(X,Y) + R(X,Y)$. Since all $\mathcal{Q}_j(X,Y)$ have points from $\mathcal{P}$ as roots of corresponding multiplicities M, these points will be roots of the same multiplicity of $R(X,Y)$. Thus we have obtained a polynomial with degree $j = \mathrm{wdeg}_{(0,1)}\,R(X,Y) \leqslant r$ in y, having roots of multiplicities in M at all points of $\mathcal{P}$ such that $\deg_{1,k-1} R(X,Y) < \deg_{1,k-1}\mathcal{Q}_j(X,Y)$, which contradicts to the property of $\mathcal{Q}_j(X,Y)$ minimality (see [10] for proof). Thus $R(X,Y) = 0$. This lemma proves that the $\mathcal{Q}_j(X,Y)$ polynomials form a basis of a polynomial module and any polynomial $\mathcal{Q}(X,Y)$ as described in Lemma 1 may be represented as algorithm is

$$\mathcal{Q}(X,Y) = \mathcal{Y}\mathcal{Q}(X)P(X) = \begin{pmatrix}1 & y & \dots & y^r\end{pmatrix}\begin{pmatrix}q_{00}(X) & q_{01}(X) & \dots & q_{0r}(X)\\ q_{10}(X) & q_{11}(X) & \dots & q_{1r}(X)\\ \dots & & & \\ q_{r0}(X) & q_{r1}(X) & \dots & q_{rr}(X)\end{pmatrix}\begin{pmatrix}p_0(X)\\ p_1(X)\\ \dots\\ p_r(X)\end{pmatrix} \tag{2}$$

Then each step of the IIA may be represented as multiplication of matrix polynopmial $\mathcal{Q}(X)$ obtained during previous steps by matrix

$$\begin{pmatrix}1 & 0 & \dots & 0 & \dots & 0\\ 0 & 1 & \dots & 0 & \dots & 0\\ \dots & & & & & \\ -\frac{\Delta_0}{\Delta_{j0}} & -\frac{\Delta_1}{\Delta_{j0}} & \dots & (x-x_i) & \dots & -\frac{\Delta_r}{\Delta_{j0}}\\ \dots & & & & & \\ 0 & 0 & \dots & 0 & \dots & 1\end{pmatrix}. \tag{3}$$

Note that the determinant of this matrix equals to $\delta(x - x_i), \delta \in GF(q)\backslash\{0\}$, which implies that for each extension of the original field $GF(q)$ it is singular only for $x = x_i$. Thus IIA may be considered as a process of sequential construction of a matrix polynomial $\mathcal{Q}(X)$ whose columns form a basis of a polynomial module.

It can be easily proved that application of the IIA with different order of interpolation points leads to equivalent results. Let us assume that $\{\mathcal{Q}_j^{(1)}(X,Y)\}$ and $\{\mathcal{Q}_j^{(2)}(X,Y)\}$ are interpolation polynomials obtained for the sets $\mathcal{P}_1, \mathcal{P}_2 : \mathcal{P}_1 \bigcap \mathcal{P}_2 = \emptyset$ of interpolation points with associated multiplicities $M1$ and $M2$, respectively. Then application of IIA to the set of points $\mathcal{P} = \mathcal{P}_1 \bigcup \mathcal{P}_2$, with multiplicity set $M = M1 \bigcup M2$ would lead to polynomials $\{\mathcal{Q}_j(X,Y)\}$ such that

$$\mathcal{Q}(X) = \mathcal{Q}^{(1)}(X)P_1(X) = \mathcal{Q}^{(2)}(X)P_2(X), \tag{4}$$

where $\mathcal{Q}(X), \mathcal{Q}^{(s)}(X)$ are matrix polynomials obtained from $\mathcal{Q}_j(X,Y)$ and $\mathcal{Q}_j^{(i)}(X,Y), i = 1,2$ respectively, while $P_i(X)$ are equal to products of matrices (3) and some unimodular matrices (i.e. matrices with

determinants in $GF(q)\backslash\{0\}$) of linear transformations required to obtain an equation. Thus the problem of "merging" two sets of interpolation polynomials may be considered as finding a *least common right multiple* of the respective matrix polynomials. Unfortunately, even the most recent algorithms for performing this task (e.g. [1]) appear to be too computationally expensive.

B. Algebraic-geometric interpretation of the interpolation algorithm

As it can be seen from (4), the divide-and-conquer interpolation method may be considered as intersection of two modules. However, these modules posess certain additional properties which may be used to simplify computations:

1) It is possible to introduce the multiplication operation for two module elements. However, its result belongs to a module of higher dimension.
2) The module may be considered as a subset of bivariate polynomial ideal.

Thus one can perform intersection operation in an ideal which is a superset of a module, and then convert its result back into the module. Actually we have the following lemma.

Lemma 2. *Let $\mathcal{Q}_j(X,Y)$, $j=0,...,r$ be a set of polynomials, produced by the IIA after processing the set $\mathcal{P}$ of interpolation points with corresponding multiplicities M. If $\mathrm{LT}\,\mathcal{Q}_r(X,Y) = Y^r$, then $\langle \mathcal{Q}_0(X,Y), ..., \mathcal{Q}_r(X,Y)\rangle$ is a basis (actually a Groebner basis) for ideal of polynomials $\mathcal{Q}(X,Y)$ that pass through points in $\mathcal{P}$ with corresponding multiplicities M.*

Proof. Similar to the proof of Lemma 1, thus omitted here.

It is possible to generalize the definition of affine variety to accommodate the case of roots with high multiplicity. Then the IIA may be considered as a process of adding interpolation points to an affine variety defined by the ideal of interpolation polynomials. Thus the operation of module intersection may be replaced with intersection of two ideals. But again, the ideal intersection algorithm [2] appears to be too complex. However, in some cases computation of ideal intersection may be replaced with computation of their product. Let $\mathcal{I} \subseteq \mathcal{R}$ be a polynomial ideal. Then the quotient ring $\mathcal{R}/\mathcal{I}$ is isomorphic to the $\mathbb{F}_q$-vector space spanned by the set

$$\left\{ X^a Y^b \ : \ X^a Y^b \notin \langle \mathrm{LT}(\mathcal{I})\rangle \right\}$$

where $\langle \mathrm{LT}(\mathcal{I})\rangle$ denotes the ideal generated by the leading terms of the elements of $\mathcal{I}$. This is Proposition 4 of [2, p. 229]. We let $\dim_{\mathbb{F}_q} \mathcal{R}/\mathcal{I}$ denote the dimension of this vector space. The *footprint* $\Delta(\mathcal{I})$ of $\mathcal{I}$, also called the *deltaset* of $\mathcal{I}$, can be defined as the set of all monomials in $\mathcal{R}$ that are *not* the leading monomials of elements of $\mathcal{I}$. It is known [7] that $\dim_{\mathbb{F}_q} \mathcal{R}/\mathcal{I} = |\Delta(\mathcal{I})|$, provided $\Delta(\mathcal{I})$ is finite. However, we will not need this result.

Theorem 1. *Let $\mathcal{P} = \{(x_1,y_1),(x_2,y_2),\ldots,(x_s,y_s)\}$ be a set of s distinct points in $\mathbb{F}_q^2$ and let $M = (m_1,m_2,\ldots,m_s)$ be a sequence of positive integers, called the multiplicities of the points in $\mathcal{P}$. Consider the ideal*

$$\mathcal{I} \stackrel{\mathrm{def}}{=} \left\{ \mathcal{Q}(X,Y) \in \mathcal{R} \ : \ \mathsf{coef}\left\{ \mathcal{Q}(X+x_i, Y+y_i), X^a Y^b \right\} = 0 \ \text{ for } a+b < m_i \right\} \tag{5}$$

Following [9], *define the cost of M as $\mathcal{C}(M) = \frac{1}{2}\sum_{i=1}^{s} m_i(m_i+1)$. Then $|\Delta(\mathcal{I})| = \mathcal{C}(M)$. That is, the dimension of $\mathcal{R}/\mathcal{I}$ as a vector space over $\mathbb{F}_q$ is equal to the cost of M.*

Proof. Let $\mathcal{C}(M) = n$. We will construct a bijection between the quotient ring $\mathcal{R}/\mathcal{I}$ and an n-dimensional vector space over $\mathbb{F}_q$. To this end, let us introduce a map $\Phi : \mathcal{R} \to \mathbb{F}_q^n$, defined as follows

$$\Phi(\mathcal{Q}) \stackrel{\mathrm{def}}{=} \Big(c_{0,0;1}(\mathcal{Q}), c_{1,0;1}(\mathcal{Q}), \ldots, c_{0,m_1;1}(\mathcal{Q}), \ \ldots, c_{0,0;s}(\mathcal{Q}), c_{1,0;s}(\mathcal{Q}), \ldots, c_{0,m_s;s}(\mathcal{Q}) \Big)$$

where $c_{a,b;i}(Q) \stackrel{\mathrm{def}}{=} \mathsf{coef}\{Q(X+x_i, Y+y_i), X^a Y^b\}$ for all nonnegative integers a, b such that $a+b < m_i$ and for all $i = 1,2,\ldots,s$. Thus $\Phi(Q)$ is precisely the set of coefficients in (5), arranged in a fixed

order. Specifically, $c_{a',b';j}(Q)$ precedes $c_{a,b;i}(Q)$ in the n-dimensional vector $\Phi(Q)$ iff $j < i$ or $j = i$ and $(a', b') \prec (a, b)$, where $\prec$ is the graded lex order on $\mathbb{N}^2$ (actually, any graded order on $\mathbb{N}^2$ would suffice for our purposes). Now let us define the following polynomials:

$$G_{a,b;i}(X,Y) \stackrel{\text{def}}{=} (X-x_i)^a (Y-y_i)^b \prod_{x_r \neq x_i} (X-x_r)^{m_i} (Y-y_r)^{m_i} \prod_{\substack{x_r = x_i \\ y_r \neq y_i}} (Y-y_r)^{m_i} \tag{6}$$

for all nonnegative integers a, b such that $a + b < m_i$ and for all $i = 1, 2, \ldots, s$. By definition, these polynomials satisfy $c_{a,b;i}(G_{a,b;i}) \neq 0$. Moreover, $c_{a',b';j}(G_{a,b;i}) = 0$ if $j \neq i$ or if $j = i$ and either $a > a'$ or $b > b'$ — in particular, if $(a', b') \prec (a, b)$. Let us arrange the n polynomials $G_{a,b;i}$ in (6) in the same order as the n coefficients $c_{a,b;i}(Q)$ in the vector $\Phi(Q)$, and consider the $n \times n$ matrix A having $\Phi(G_{a,b;i})$ as its rows. It follows from the properties of the polynomials $G_{a,b;i}$ that the matrix A is upper triangular. Hence its rows constitute a basis for $\mathbb{F}_q^n$. This implies that for each vector $v \in \mathbb{F}_q^n$, we can construct a polynomial $Q(X,Y) \in \mathcal{R}$ such that $\Phi(Q) = v$. Indeed, if v is expressed as a linear combination of the rows of A, then $Q(X,Y)$ is just the corresponding linear combination of the polynomials $G_{a,b;i}$ in (6). This shows that the mapping Φ is surjective.

Now consider the mapping $\Psi : \mathcal{R}/\mathcal{I} \to \mathbb{F}_q^n$ defined by $\Psi([Q]) = \Phi(Q)$, where $[Q]$ is the equivalence class of Q in $\mathcal{R}/\mathcal{I}$. It follows from [2, Chapter 5] that the mapping Ψ is well-defined. Moreover, since Φ is surjective, then so is Ψ. The mapping Ψ, on the other hand, is also injective. Indeed, if $\Psi([Q]) = \Phi(Q) = \mathbf{0}$ in $\mathbb{F}_q^n$, then $Q \in \mathcal{I}$ by (5) and the definition of $\Phi(Q)$. Hence $\Psi([Q]) = \mathbf{0}$ if and only if $[Q] = [0] = \mathcal{I}$. Together with the linearity of Ψ, this shows that Ψ is an injection, as claimed. We thus conclude that Ψ is a bijection from $\mathcal{R}/\mathcal{I}$ to $\mathbb{F}_q^n$. Hence $\dim_{\mathbb{F}_q} \mathcal{R}/\mathcal{I} = n$. ∎

Remark. The mapping Ψ constructed in the proof above is, in fact, a vector space isomorphism between $\mathcal{R}/\mathcal{I}$ and $\mathbb{F}_q^n$. Thus $\mathcal{R}/\mathcal{I} \simeq \mathbb{F}_q^n$, for $n = \mathcal{C}(M)$.

Corollary 2. *The affine variety $\mathbf{V}(\mathcal{I})$ defined by the ideal $\mathcal{I}$ in* (5) *is equal to $\mathcal{P}$.*

Proof. Clearly $\mathcal{P} \subseteq \mathbf{V}(\mathcal{I})$ by definition. Assume to the contrary that there exists a point P such that $P \in \mathbf{V}(\mathcal{I})$ but $P \notin \mathcal{P}$. Let $\mathcal{P}' = \mathcal{P} \cup P$, let $M' = (m_1, m_2, \ldots, m_s, 1)$, and let $\mathcal{I}'$ be the ideal defined as in (5) in terms of $\mathcal{P}'$ and M'. Then the fact that $P \in \mathbf{V}(\mathcal{I})$ implies that $\mathcal{I}' = \mathcal{I}$. However, this contradicts Theorem 1, since $\mathcal{C}(M') \neq \mathcal{C}(M)$. ∎

Theorem 3. *Let $\mathcal{I}(\mathcal{P}, M)$ denote the ideal defined in* (5) *with respect to the point set $\mathcal{P}$ and multiplicity vector M. Suppose $\mathcal{P}_1, M_1$ and $\mathcal{P}_2, M_2$ are such that $\mathcal{P}_1 \cap \mathcal{P}_2 = \varnothing$. Then*

$$\mathcal{I}(\mathcal{P}_1, M_1) \cap \mathcal{I}(\mathcal{P}_2, M_2) \;=\; \mathcal{I}(\mathcal{P}_1, M_1) \cdot \mathcal{I}(\mathcal{P}_2, M_2) \tag{7}$$

Proof. Let $\mathcal{I}_1 = \mathcal{I}(\mathcal{P}_1, M_1)$ and $\mathcal{I}_2 = \mathcal{I}(\mathcal{P}_2, M_2)$. Then $\mathbf{V}(\mathcal{I}_1) = \mathcal{P}_1$ and $\mathbf{V}(\mathcal{I}_2) = \mathcal{P}_2$ by Corollary 2. It follows that

$$\mathbf{V}(\mathcal{I}_1 + \mathcal{I}_2) = \mathbf{V}(\mathcal{I}_1) \cap \mathbf{V}(\mathcal{I}_2) = \mathcal{P}_1 \cap \mathcal{P}_2 = \varnothing$$

Hence, by the weak Nullstellensatz [2, p. 168], we have $\mathcal{I}_1 + \mathcal{I}_2 = \bar{F}[X,Y]$, where $\bar{F}$ is the algebraic closure of $\mathbb{F}_q$. Thus $\mathcal{I}_1$ and $\mathcal{I}_2$ are co-prime, which implies (7) by the Chineese remainder theorem for polynomials. ∎

Accordingly to the Chinese remainder theorem for ideals, the intersection of two coprime ideals equals to their product. Thus one can perform interpolation as follows: apply IIA to the disjoint sets V_1 and V_2 of interpolation points obtaining basis polynomials $Q_1 = \{Q_j^{(1)}(X,Y) > j = 0..r\}$ and $Q_2 = \{Q_j^{(2)}(X,Y), j = 0..r\}$. compute their pairwise product (i.e. basis of the ideal product) and apply an elimination algorithm to reduce ρ^2 polynomials to $2\rho - 1$ ones forming a basis of the module. Note, that this procedure increases the number of basis polynomials. Of course, it may be applied recursively.

The Divide-and-Conquer Interpolation Algorithm

$SplitInterpolation(s, \mathcal{P}, M)$
$\mathcal{P}_1 := \{(x_i, y_i)\}, M_1 := \{m_i\}, i = 1, ..., \lfloor s/2 \rfloor;$
$\mathcal{P}_2 := \{(x_i, y_i)\}, M_2 := \{m_i\}, i = \lfloor s/2 \rfloor + 1, ..., s;$
$\mathcal{Q}_1 := SplitInterpolation(\lfloor s/2 \rfloor, \mathcal{P}_1, M_1);$
$\mathcal{Q}_2 := SplitInterpolation(s - \lfloor s/2 \rfloor, \mathcal{P}_2, M_2);$
$\mathcal{Q}' := \{\mathcal{Q}_{j_1}^{(1)}(X, Y)\mathcal{Q}_{j_2}^{(2)}(X, Y), \mathcal{Q}_{j_i}^{(i)} \in \mathcal{Q}_i\};$
$\mathcal{Q} := Eliminate(Q');$
Return $\mathcal{Q}$

One can see that the complexity of this algorithm is $C(n, r, \rho) = 2C(n/2, r, \rho/2) + C_m + C_e$, where C_m is the cost of polynomial multiplication and C_e is the cost of the elimination step. For the multiplication step there is a problem of efficient multiplication of large polynomials which may appear after processing of many interpolation points. Thus one has either to use FFT-based algorithm operating over some sextension of the original field, or apply a combination of Winograd linear convolution algorithm with some other methods (e.g., again FFT-based algorithm). For the elimination step, the task is to find a Groebner basis of the polynomial module. One can use either the Buchberger algorithm or perform elimination in spectral domain [6].

IV. Conclusions

The main difference between our approach and the one suggested in [3] is that interpolation subproblems are solved independently and only afterwards their solutions are "merged." This allows one to solve these subproblems in parallel. Moreover, each of the subproblems has a much smaller dimension than the original problem. However, further analysis is required to find an efficient way for eliminating the redundant entries in module bases obtained during the merging step.

References

[1] B. Beckermann and G. Labahn. Fraction-free computation of matrix rational interpolants and matrix GCDs. *SIAM Journal on Matrix Analysis and Applications*, 22(1):114–144, 2001.

[2] D. Cox, G. Little, and D. O'Shea. *Ideals, varieties and algorithms*. Springer-Verlag, 1992.

[3] G.-L. Feng and X. Giraud. Fast algorithm in sudan decoding procedure for reed-solomon codes. *preprint*, August 2002.

[4] F.R. Gantmakher. *Matrices theory*. Moscow: Nauka, In Russian, 4 edition, 1988.

[5] V. Guruswami and M. Sudan. Improved decoding of Reed-Solomon and algebraic-geometric codes. *IEEE Transactions on Information Theory*, 45(6):1757–1767, September 1999.

[6] D. Henrion and M. Sebek. Reliable numerical methods for polynomial matrix triangularization. *IEEE Transactions on Automatic Control*, 44(3), March 1999.

[7] T. HØHOLDT, J.H. VAN LINT, and R. PELLIKAAN, Algebraic geometry codes, in V.S. Pless and W.C. Huffman, Editors, *Handbook of Coding Theory*, Elsevier 1998.

[8] T. Kailath. *Linear systems*. Prentice Hall, 1985.

[9] R. KOETTER and A. VARDY, Algebraic soft-decision decoding of Reed-Solomon co- des, *IEEE Trans. Inform. Theory*, vol. 49, pp. 2009–2025, November 2003.

[10] R. Refslund Nielsen and T. Hoholdt. Decoding Reed-Solomon codes beyond half the minimum distance. In *Proceedings of the International Conference on Coding Theory and Cryptography, Mexico 1998*. Springer-Verlag, 1998.

[11] R.R. Nielsen. *List decoding of linear block codes*. PhD thesis, Technical University of Denmark, 2001.

[12] H. O'Keeffe and P. Fitzpatrick. Groebner basis solution of constrained interpolation problems. *Linear Algebra and Applications*, 351-352:533–551, 2002.

[13] R. Roth and G. Ruckenstein. Efficient decoding of Reed-Solomon codes beyond half the minimum distance. *IEEE Transactions on Information Theory*, 46(1):246–257, 2000.

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