On Fractional Decoding of Reed-Solomon Codes

Welington Santos

***Abstract*—We define a virtual projection of a Reed-Solomon code $RS(q^l, n, k)$ to an $RS(q, n, k)$ Reed-Solomon code. A new probabilistic decoding algorithm that can be used to perform fractional decoding beyond the α- decoding radius is considered. An upper bound for the failure probability of the new algorithm is given, and the performance is illustrated by examples.**

***Index Terms*—Fractional decoding, Virtual projection, Interleaved Reed-Solomon codes.**

I. Introduction

An Interleaved Reed-Solomon code [4], [6], [10] is obtained by stacking m codewords of different m $RS(q, n, k_j)$ codes of the same length n. A codeword of an Interleaved Reed-Solomon code is an $m \times n$ matrix over the field $\mathbb{F}_q$. Interleaved Reed-Solomon codes make sense in scenarios where the error affects all m RS codewords at the same positions. In [7], Schmidt et al. presented a scheme that virtually extends a low-rate RS code to an Interleaved Reed-Solomon code and a probabilistic decoding algorithm that can correct errors beyond the unique decoding radius of the RS-code.

Recently, Tamo at al. [3], considered error correction by maximum distance separable (MDS) codes based on part of the received codeword to define a fractional decoding problem, and the α-decoding radius of an (n, k, l) array code over a finite field $\mathbb{F}_q$. The fractional decoding problem is motivated by the fact that in distributed systems [2], usually there is a limitation on the disk operation as well as on the amount of information transmitted for the purpose of decoding.

In this contribution, we consider a Reed-Solomon code $RS(q^l, n, k)$ with evaluation set $\mathcal{L} \subseteq \mathbb{F}_q$ and define a virtual projection to an $RS(q, n, k)$ Reed-Solomon code. We also present a probabilistic approach to the problem of fractional decoding. For $\alpha = m/l$ and an $RS(q^l, n, k)$ code of rate $R \leqslant \frac{\alpha}{m(1-\alpha)+1}$ our method corrects more errors than guaranteed by the α-decoding radius with failure probability given approximately by $mq^{-(m+1)(\tau_{P_\alpha}-t)-1}$, where $t < \tau_{P_\alpha}$ is the number of errors that we would like to correct.

This work is structured as follows. In Sect. 2, we recall collaborative decoding of Interleaved Reed-Solomon codes [6] and fractional decoding [3]. In Sect. 3, we define a virtual projection to an $RS(q, n, k)$-code, and we show how the virtual projection can be used to perform fractional decoding beyond the α-decoding radius.

II. Preliminaries

A. Reed-Solomon and Interleaved Reed-Solomon Codes

Definition 1. *Let $\mathcal{L} = \{\gamma_1, \ldots, \gamma_n\}$ where $\gamma_1, \ldots, \gamma_n$ are distinct nonzero elements of the finite field $\mathbb{F}_q$. For a given univariante polynomial $f(x) \in \mathbb{F}_q[x]$ denote*

$$f(\mathcal{L}) = (f(\gamma_1), \ldots, f(\gamma_n)).$$

W. Santos is with the Programa de Pós-Graduação em Matemática, Universidade Federal do Paraná, Caixa Postal 19081, 81531-990, Curitiba-PR Brazil. His study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior- Brasil (CAPES)-Finance Code 001. Email: wsantos.math@gmail.com

A Reed-Solomon code $RS(q, n, k)$ over a field $\mathbb{F}_q$ with $n < q$ is given by

$$RS(q, n, k) = \{c = f(\mathcal{L}) : \ f(x) \in \mathbb{F}_q[x]_k\}, \tag{1}$$

where $\mathbb{F}_q[x]_k$ denotes the set of all univariate polynomials of degree less than k. The set $\mathcal{L}$ is called the evaluation set of $\mathcal{C}$.

An Interleaved Reed-Solomon code of order m is given by m underlying RS codes, which are arranged in a matrix form.

Definition 2. *Let the set $\mathcal{K} = \{k_0, k_2, \ldots, k_{m-1}\}$, consist of m integers, where all $k_j < n$. An Interleaved Reed-Solomon code $IRS(q, n, \mathcal{K}, m)$ of order m is given by*

$$IRS(q, n, \mathcal{K}, m) = \left\{ C = \begin{pmatrix} f_0(\mathcal{L}) \\ f_1(\mathcal{L}) \\ \vdots \\ f_{m-1}(\mathcal{L}) \end{pmatrix} : \ f_j(x) \in \mathbb{F}_q[x]_{k_j} \right\}, \tag{2}$$

The codewords $f_j(\mathcal{L}) \in RS(q, n, k_i)$ are called elementary codewords of the $IRS(q, n, \mathcal{K}, m)$-code. If the dimensions k_j are equal for all $j = 0, \ldots, m-1$ the IRS code is called homogneous. Otherwise, the IRS code is called heterogeneous.

In considering IRS codes we are interested in column errors. This is equivalent to transmission of the IRS code over a q^m-ary channel.

Let $C \in IRS(q, n, \mathcal{K}, m)$ and $R = C + E$, where $E = (E_1, \ldots, E_n)$ and $w(E) := |\{i : E_i \neq 0\}| = t$, denote the received word. The m elementary codewords of an IRS code are affected by m elementary error words $e^{(0)}, e^{(1)}, \ldots, e^{(m-1)}$ of weight $wt(e^{(j)}) = t_j \leqslant t$. Let $\mathcal{E}^{(j)}$ denote the set of error positions for the $j-th$ elementary received word. Since we are considering column erros, the union of the m sets of error positions $\mathcal{E} = \mathcal{E}^{(0)} \cup \mathcal{E}^{(1)} \cup \ldots \cup \mathcal{E}^{(m-1)} \subseteq \{1, \ldots, n\}$ has cardinality $|\mathcal{E}| = t$.

B. Collaborative Decoding of Interleaved Reed-Solomon Codes

In [6], Schmidt et al. introduced the concept of collaborative decoding for Interleaved Reed-Solomon codes. This decoder is based on the fact that the errors occur in the same positions of each elementary codeword of the Interleaved Reed-Solomon code.

In the first step of collaborative decoding, m syndrome polynomials $S^{(0)}(x), S^{(1)}(x), \ldots, S^{(m-1)}(x)$ of degree smaller than $n - k_j$ are calculated. The syndrome polynomial is

$$S^{(j)}(x) = \sum_{i=1}^{n-k_j} S_i^{(j)} x^{i-1} \tag{3}$$

with coefficients:

$$S_i^{(j)} = r^{(j)}(\gamma_i^{k_j}) = \sum_{h=1}^{n} r_h^{(j)} \gamma_i^{k_j(h-1)} \tag{4}$$

for all $i = 1, \ldots, n - k_j$ and $j = 0, \ldots, m-1$.

As in the classical case, these syndromes are used to form a linear system of equations $S\Lambda = V$,

$$\begin{pmatrix} S^{(0)} \\ S^{(1)} \\ \vdots \\ S^{(m-1)} \end{pmatrix} \cdot \begin{pmatrix} \Lambda_1 \\ \Lambda_2 \\ \vdots \\ \Lambda_t \end{pmatrix} = \begin{pmatrix} V^{(0)} \\ V^{(1)} \\ \vdots \\ V^{(m-1)} \end{pmatrix}, \tag{5}$$

where each sub-matrix $S^{(j)}$ is a $(n - k_j - t) \times t$ matrix and each $V^{(j)}$ is a column vector of length $n - k_j - t$:

$$S^{(j)} = \begin{pmatrix} S^{(j)}_{t+1} & S^{(j)}_{t} & \cdots & S^{(j)}_{2} \\ S^{(j)}_{t+2} & S^{(j)}_{t+1} & \cdots & S^{(j)}_{3} \\ \vdots & \vdots & \ddots & \vdots \\ S^{(j)}_{n-k_j} & S^{(j)}_{n-k_j-1} & \cdots & S^{(j)}_{n-k_j-t+1} \end{pmatrix},$$

$$V^{(j)} = \begin{pmatrix} -S^{(j)}_{1} \\ -S^{(j)}_{2} \\ \vdots \\ -S^{(j)}_{n-k_j-t} \end{pmatrix} \tag{6}$$

The system of equations (5) has $\sum_{j=0}^{m-1}(n-k_j-t)$ equations and t unknowns. In order to guarantee unambiguous decoding, the number of linearly independent equations has to be greater than or equal to the number of unknowns. Under the assumption that all equations in (5) are linearly independent we obtain the following restriction on t:

$$\sum_{j=0}^{m-1}(n - k_j - t) \geqslant t \tag{7}$$

Which can be rewritten as

$$t \leqslant \frac{m}{m+1}\left(n - \frac{1}{m}\sum_{i=0}^{m-1} k_j\right) := \tau_{IRS} \tag{8}$$

However, there is a certain probability that some of the equations (5) are linearly dependent. In this case, there is no unique solution of the system of equations and we declare a *decoding failure*.

The collaborative decoder presented by Schimidt et al. [6], can corrects t erros, $t \leqslant \tau_{IRS}$ with a failure probability of

$$\left(\frac{q^m - \frac{1}{q}}{q^m - 1}\right)^t \frac{q^{-(m+1)(\tau_{IRS}-t)}}{q-1}. \tag{9}$$

C. Fractional decoding

Tamo et al. [3], introduced the concept of fractional decoding where error correction by maximum distance separable codes based on part of the received codeword is considered. The idea is that the decoder downloads an α proportion of each of the codeword's coordinates. Below we will describe the α-decoding problem.

Fractional decoding is defined in the following

Definition 3. *Let $\mathcal{C}$ be an (n, k, l) array code over field $\mathbb{F}_q$. We say that $\mathcal{C}$ corrects up to t errors by downloading αnl symbols of $\mathbb{F}_q$ if there exist functions*

$$f_i : \mathbb{F}_q^l \longrightarrow \mathbb{F}_q^{\alpha_i l}, i = 1, \ldots, n \text{ and } g : \mathbb{F}_q^{\left(\sum_{i=1}^n \alpha_i\right)} \longrightarrow \mathbb{F}_q^{nl} \tag{10}$$

such that $\sum_{i=1}^n \alpha_i \leqslant n\alpha$ and for any codeword $C \in \mathcal{C}$ and any error $E \in \left(\mathbb{F}_q^l\right)^n, w(E) \leqslant t$

$$g(f_1(C_1 + E_1), \ldots, f_n(C_n + E_n)) = (C_1, \ldots, C_n). \tag{11}$$

For $\alpha \geqslant k/n$, we define the α-decoding radius of $\mathcal{C}$ as the maximum number of errors that $\mathcal{C}$ can correct by downloading αnl symbols of $\mathbb{F}_q$, and denote it as $r_\alpha(\mathcal{C})$.

Define the α-decoding radius $r_\alpha(n, k)$ as follows:

$$r_\alpha(n, k) = \max\{r_\alpha(\mathcal{C}) : \mathcal{C} \text{ is an (n,k)-code}\}. \tag{12}$$

Given an (n, k)-linear code we should take $\alpha \geqslant \frac{k}{n}$ because the codeword encodes k data symbols, and even without errors to recover the data the decoder needs at least as many imput symbols. If $\alpha = 1$, we return to the standard problem, so the goal of fractional decoding is study error correction for α in the range $\frac{k}{n} \leqslant \alpha < 1$.

It was also shown in [3] that the α-decoding radius of a (n, k)-linear code is

$$\tau_\alpha = \left\lfloor \frac{n - k/\alpha}{2} \right\rfloor \tag{13}$$

and that an $RS(q^l, n, k, \mathcal{L})$ with $\mathcal{L} \subseteq \mathbb{F}_q$ achieves the optimal α-decoding radius (13).

III. Fractional Decoding and collaborative decoding

A. Virtual Projection to an Interleaved Reed-Solomon Code

Schmidt et al. [7], [8], suggested to extend a low-rate $RS(n, k)$ code to an IRS code to perform syndrome decoding of the $RS(n, k)$ code beyond half the minimum distance, of course, with some failure probability. Zeh et al. [9], defined the mixed virtual extension of a homogeneous interleaved Reed-Solomon code to an heterogeneous interleaved Reed-Solomon code with objective of decoding beyond its joined error-correcting capability [4].

In this subsection, we will introduce the concept of virtual projection of a Reed-Solomon code $RS(q^l, n, k) \subseteq \mathbb{F}_{q^l}^n$ with evaluation set $\mathcal{L} = \{\gamma_1, \ldots, \gamma_n\} \subseteq \mathbb{F}_q$ to a heterogeneous Reed-Solomon code $IRS(q, n, \mathcal{K}, m)$. Our purpose is to use the virtual projection to perform fractional decoding beyond the α-decoding radius.

Definition 4. *Let $A_0, A_1, \ldots, A_{m-1} \subseteq \mathbb{F}_q$ be m pairwise disjoint sets of the field $\mathbb{F}_q$. For $j = 0, 1, \ldots, m-1$, define the annihilator polynomials of the set A_j to be*

$$p_j(x) = \prod_{\omega \in A_j}(x - \omega). \tag{14}$$

Note that, $\deg p_j(x) = |A_j| \ \forall j = 0, \ldots, m-1$.

Definition 5. *Let $F = \mathbb{F}_{q^l}$ be a finite field extension of $B = \mathbb{F}_q$ of degree l. The field trace is defined*

$$tr_{F/B}(\beta) = \beta + \beta^q + \beta^{q^2} + \ldots + \beta^{q^{l-1}}.$$

Let $\zeta_0, \zeta_1, \ldots, \zeta_{l-1}$ be a basis of F over B, and let $\nu_0, \nu_1, \ldots, \nu_{l-1}$ be the dual basis (i.e., $tr_{F/B}(\zeta_i \nu_j) = \delta_{i,j}$ for all i, j). Then

$$\beta = \sum_{i=0}^{l-1} tr_{F/B}(\zeta_i \beta)\nu_i.$$

In other words, any element β in F can be calculated from its l projections $\left\{tr_{F/B}(\zeta_i\beta)\right\}_{i=0}^{l-1}$ on B.

Definition 6. *Given a polynomial* $h(x) = a_{k-1}x^{k-1} + a_{k-2}x^{k-2} + \ldots + a_0 \in \mathbb{F}_{q^l}[x]$ *and* m *pairwise disjoint subsets* $A_0, \ldots, A_{m-1} \subseteq \mathbb{F}_q$. *Define*

$$T_j(h)(x) = h_{l-m+j}(x)(p_j(x))^{(l-m)(j+1)} + \sum_{u=0}^{l-m-1} h_u(x)(p_j(x))^{u(j+1)} \quad (15)$$

for all $j = 0, \ldots, m-1$ *and the polynomial* $h_i(x) \in \mathbb{F}_q[x]$ *is given by*

$$h_i(x) = tr(\zeta_i a_{k-1})x^{k-1} + tr(\zeta_i a_{k-2})x^{k-2} + \ldots + tr(\zeta_i a_0). \quad (16)$$

Lemma 7. *Let* $\mathcal{C} = RS(q^l, n, k)$ *be a Reed-Solomon code and* $h(\mathcal{L}) \in \mathcal{C}$ *where* $\mathcal{L} \subset \mathbb{F}_q$ *is the evaluation set of* $\mathcal{C}$. *Then each codeword* $T_j(h)(\mathcal{L})$ *is a codeword of the Reed-Solomon code*

$$\mathcal{C}_j = RS\left(q, n, k + |A_j|(l-m)(j+1)\right). \quad (17)$$

Proof. Fist note that

$$\deg T_j(h)(x) \leqslant \max\left\{\deg h_{l-m+j}(x)(p_j(x))^{(l-m)(j+1)}, \deg \sum_{u=0}^{l-m-1} h_u(x)(p_j(x))^{u(j+1)}\right\}$$

and we can check that

$$\begin{aligned}\deg h_{l-m+j}(x)(p_j(x))^{(l-m)(j+1)} &= \deg h_{l-m+j}(x) \\ &\quad + |A_j|(l-m)(j+1) \\ &< k + |A_j|(l-m)(j+1).\end{aligned}$$

and

$$\deg \sum_{u=0}^{l-m-1} h_u(x)(p_j(x))^{u(j+1)} \quad < \quad k + |A_j|(l-m)(j+1).$$

So, $\deg T_j(h)(x) < k + |A_j|(l-m)(j+1)$ for all $j = 0, 1, \ldots m-1$. Now we must check that $T_j(h)(\mathcal{L}) \in \mathbb{F}_q^n$. By definition, $T_j(h)(\mathcal{L}) = (T_j(h)(\gamma_1), \ldots, T_j(h)(\gamma_n))$, so we just need to prove that $T_j(h)(\gamma_i) \in \mathbb{F}_q$ for all $i = 1, \ldots, n$. For all $j = 0, \ldots, m-1$. we have

$$T_j(h)(\gamma_i) = h_{m-l+j}(\gamma_i)(p_j(\gamma_i))^{(l-m)(j+1)} + \sum_{u=0}^{l-m-1} h_u(\gamma_i)(p_j(\gamma_i))^{u(j+1)}$$

as $h_u(x), p_j(x) \in \mathbb{F}_q[x]$ and $\gamma_i \in \mathbb{F}_q$ it is clear that $T_j(h)(\gamma_i) \in \mathbb{F}_q$ for all $i = 1, \ldots, n$ and $j = 0, \ldots, m-1$. □

Definition 8. *Let* $\mathcal{C} = RS(q^l, n, k)$ *be a Reed-Solomon code with evaluation set* $\mathcal{L} = \{\gamma_1, \ldots, \gamma_n\} \subseteq \mathbb{F}_q$ *and let* $A_0, \ldots, A_{m-1}$ *any pairwise disjoint subsets of* $\mathbb{F}_q$ *such that* $\sum_{j=0}^{m-1} |A_j| \geqslant k$. *The Virtual Projection* $\mathcal{C}_{P_{m/l}}(q, n, \mathcal{K})$ *is given by*

$$\mathcal{C}_{P_{m/l}} = \left\{ \begin{pmatrix} c_{(0)} \\ c_{(1)} \\ \vdots \\ c_{(m-1)} \end{pmatrix} = \begin{pmatrix} T_0(h)(\mathcal{L}) \\ T_1(h)(\mathcal{L}) \\ \vdots \\ T_{m-1}(h)(\mathcal{L}) \end{pmatrix} \right\}, \quad (18)$$

where $T_j(h)(x)$ *is given by (15) and* $\mathcal{K} = \{k_0, \ldots, k_{m-1}\}$ *with* $k_j = k + |A_j|(l-m)(j+1)$ *for all* $j = 0, \ldots, m-1$.

Assume that a codeword $c(\mathcal{L}) \in \mathcal{C}$ is transmitted over a noisy channel, which adds t erros in such a way, that the word $y(\mathcal{L}) = c(\mathcal{L}) + e(\mathcal{L})$ is observed at the channel output. Using the observed word $y(\mathcal{L})$, we calculate the m polynomials $T_j(y)(x)$, $j = 0, \ldots, m-1$, and create the matrix

$$Y = \begin{pmatrix} T_0(y)(\gamma_1) & \ldots & T_0(y)(\gamma_n) \\ T_1(y)(\gamma_1) & \ldots & T_1(y)(\gamma_n) \\ \vdots & \ddots & \vdots \\ T_{m-1}(y)(\gamma_1) & \ldots & T_{m-1}(y)(\gamma_n) \end{pmatrix} \quad (19)$$

The matrix Y can be considered as received word of the virtual projection $\mathcal{C}_{P_{m/l}}(q, n, \mathcal{K})$ of $\mathcal{C} = RS(q^l, n, k)$.

Theorem 9. *Let* $c(\mathcal{L}) \in RS(q^l, n, k)$ *be a codeword of a Reed-Solomon code* $\mathcal{C}$ *transmitted over a noisy channel. Assume that the word* $y(\mathcal{L}) = c(\mathcal{L}) + e(\mathcal{L})$ *is received, if* $e = (e_1, \ldots, e_n)$ *has* t *nonzero coefficients* $e_{i_1}, \ldots, e_{i_t}$ *then the matrix* Y *is a corrupted codeword of the* $\mathcal{C}_{P_{m/l}}(q, n, \mathcal{K})$ *code with at most* t *erroneous columns at the positions* $i_1, \ldots, i_t$.

Proof. If $e = 0$, then $y = c \in \mathcal{C}$, and by Lemma (7) we know that Y is a codeword of the virtually projection $\mathcal{C}_{P_{m/l}}(q, n, \mathcal{K})$. Note that

$$T_j(y)(\gamma_i) = T_j(c+e)(\gamma_i) = T_j(c)(\gamma_i) + T_j(e)(\gamma_i).$$

Clearly, if $e_i = 0$, that is, if $i \notin \{i_1, \ldots, i_t\}$, then $T_j(e)(\gamma_i) = 0$ for all $j = 0, \ldots, m-1$. If $i \in \{i_1, \ldots, i_t\}$, then $T_j(e)(\gamma_i)$ may be non-zero, so Y has at most t erroneous columns. □

Unlike the virtual extension to an IRS code [8], where it is possible to ensure that given a word $y = c + e$ the virtual extension of y is a word with exactly t erroneous columns, in the virtual projection we can not assure it.

In addition, in the virtual extension approach given a codeword $c \in RS(q, n, k)$ and its virtual extension $C \in IRS$ when we recover the word $C \in IRS$, we immediately recover the codeword $c \in RS(q, k)$ (the first row of the codeword C). In virtual projection it is not so immediately that given a codeword $c \in RS(q^l, n, k)$ and its virtual projection $C \in \mathcal{C}_{P_{m/l}}$ we can recover the codeword $c \in RS(q^l, n, k)$ just by recovering the codeword $C \in \mathcal{C}_{P_{m/l}}$, but the following ensures it.

Lemma 10. *Given polynomials* $\{T_j(h)(x)\}_{j=0}^{m-1}$ *as in (15). Suppose that* $\sum_{j=0}^{m-1} |A_j| \geqslant \deg h(x)$ *then we can recover the polynomials* $\{h_j(x)\}$ *and consequently we can recover* $h(x)$.

Proof. $T_j(h)(\omega) = h_0(\omega)$ for all $\omega \in A_j$; of course, we can rewrite (15) as

$$\begin{aligned}T_j(h)(x) &= h_{l-m+j}(x)(p_j(x))^{(l-m)(j+1)} + \sum_{u=0}^{l-m-1} h_u(x)(p_j(x))^{u(j+1)} \\ &= h_{l-m+j}(x)(p_j(x))^{(l-m)(j+1)} \\ &\quad + h_0(x)(p_j(x))^{0(j+1)} + \sum_{u=1}^{l-m-1} h_u(x)(p_j(x))^{u(j+1)}.\end{aligned}$$

So, $T_j(h)(\omega) = h_0(\omega)$ for all $\omega \in A_j$. Then, we know the evaluations of $h_0(\omega)$ at all the points $\cup_{j=0}^{m-1} A_j$ and by assumption, $\sum_{j=0}^{m-1} |A_j| \geqslant \deg h(x) \geqslant \deg h_0(x)$, so we can recover $h_0(x)$. Now from $h_0(x)$ and $\{T_j(h)(x)\}_{j=0}^{m-1}$, we can calculate the polynomials

$$T_j^{(1)}(h)(x) = \frac{T_j(h)(x) - h_0(x)}{p_j(x)^{j+1}} = h_{l-m+j}(x)(p_j(x))^{(l-m-1)(j+1)} + h_1(x) + \sum_{u=2}^{l-m-1} h_u(x)(p_j(x))^{(u-1)(j+1)}.$$

So, $T_j^{(1)}(h)(\omega) = h_1(\omega)$ for all $\omega \in A_j$, and again, we know the evaluation of $h_1(x)$ in $\cup_{j=0}^{m-1} A_j$. So, we can recover $h_1(x)$. From $h_0(x), h_1(x)$ and $\{T_j(h)(x)\}_{j=0}^{m-1}$ we can calculate the polynomials

$$T_j^{(2)} = \frac{T_j^{(1)}(h)(x) - h_1(x)}{p_j(x)^{j+1}}.$$

Since $T_1^{(2)}(h)(\omega) = h_2(\omega)$ for all $\omega \in A_j$, by the previous argument we can recover $h_2(x)$. Generally, the polynomials $\{h_{l-m+j}(x)\}_{j=0}^{m-1}$ can be recovered from

$$h_{l-m+j}(x) = \frac{T_j(h)(x) - \sum_{u=0}^{l-m-1} h_u(x)(p_j(x))^{u(j+1)}}{(p_j(x))^{(l-m)(j+1)}}.$$

□

By Lemma 10, we conclude that given an $RS(q^l, n, k)$-code with evaluation set $\mathcal{L} \subseteq \mathbb{F}_q$ and its virtual projection $\mathcal{C}_{P_{m/l}}$ it is possible to recover a codeword $c \in \mathcal{C}$ using the code $\mathcal{C}_{P_{m/l}}$ whenever the received word $y = c + e$ has no more than t errors with $t < \tau_{P_{m/l}}$, where $\tau_{P_{m/l}}$ denotes the decoding radius of $\mathcal{C}_{P_{m/l}}$. Hence, we have the following algorithm.

Algorithm 1: Virtual Projection IRS Decoder

Input: Received word $y(\mathcal{L}) = c(\mathcal{L}) + e(\mathcal{L})$, $\alpha = m/l$
For: $j = 0$ to $m-1$ **do**
Create the matrix Y from $T_j(y)(\mathcal{L})$ and calculate the syndromes $S^{(0)}, \ldots, S^{(m-1)}$.
Compute t and $\Lambda(x)$ by Algorithm 1 in [6] .
if $t < \tau_{P_\alpha}$ *and* $\Lambda(x)$ *is t-valid* **then**
  for each j from 0 to $m-1$ **do**
  evaluate errors, and calculate $T_j(e)(\mathcal{L})$
  calculate $T_j(\hat{c})(\mathcal{L}) = T_j(y)(\mathcal{L}) - T_j(e)(\mathcal{L})$
  Use Lemma 10 to compute $c(\mathcal{L})$
else
  decoding failure
output: $c(\mathcal{L}) \in \mathcal{C}$ or decoding failure

Theorem 11. *Let $\mathcal{C} = RS(q^l, n, k)$ be a Reed-Solomon code then its virtual projection code $\mathcal{C}_{P_{m/l}}(q, n, \mathcal{K})$ given by Definition 8 has maximum decoding radius $\tau_{P_{m/l}}$ given by*

$$\tau_{P_{m/l}} = \frac{m}{m+1}\left(n - k - \frac{(l-m)}{m}\sum_{j=0}^{m-1}|A_j|(j+1)\right). \quad (20)$$

Proof. The decoding radius of the code $\mathcal{C}_{P_{m/l}}(q, n, \mathcal{K})$ is the error-correcting radius of the heterogeneous $IRS(q, n, \mathcal{K}, m)$ code with $\mathcal{K} = \{k_0, \ldots, k_{m-1}\}$ and dimensions k_j given by $k_j = k + |A_j|(l-m)(j+1)$ for all $j = 0, \ldots, m-1$. The correcting radius is given by (8)

$$\tau_{P_{m/l}} = \frac{m}{m+1}\left(n - \frac{1}{m}\sum_{j=0}^{m-1} k_i\right) = \frac{m}{m+1}\left(n - k - \frac{l-m}{m}\sum_{j=0}^{m-1}|A_j|(j+1)\right).$$

□

Corollary 12. *Let $\mathcal{C} = RS(q^l, n, k)$ be a Reed-Solomon code and $\mathcal{C}_{P_{m/l}}(q, n, \mathcal{K})$ its virtual projection as in (18), then:*

i) *If $l = m$ then $\tau_{P_{m/l}} = \frac{l}{l+1}(n-k)$;*
ii) *If $l = m = 1$ then $\tau_{P_{m/l}} = \frac{n-k}{2} = \tau$;*
iii) *If $|A_j| = b$ for all $j = 0, \ldots, m-1$ then*

$$\tau_{P_{m/l}} = \frac{m}{m+1}\left(n - k - b\frac{(l-m)}{m}\binom{m+1}{2}\right).$$

Proof. Straight forward calculation from (20). □

Note that if, $l = m$ then $\tau_{P_{m/l}}$ is the decoding radius of a *homogeneous* Interleaved Reed-Solomon code [6], [8]. For $l = m = 1$ the result $\tau_{P_{m/l}}$ is the decoding radius of the $RS(q, n, k)$ Reed-Solomon code over $\mathbb{F}_q$.

B. Fractional decoding beyond the α-decoding radius

Let $\mathcal{C} = RS(q^l, n, k)$ be a Reed-Solomon code with evaluation set $\mathcal{L} = \{\gamma_1, \ldots, \gamma_n\} \subseteq \mathbb{F}_q$. Let $\alpha = m/l$, where m and l are positive integers and $m|k$. We will show that is possible to perform fractional decoding beyond the α-decoding radius.

Let $c = (c_1, \ldots, c_n) = (h(\gamma_1), \ldots, h(\gamma_n)) \in \mathcal{C}$, where $h(x) \in \mathbb{F}_{q^l}[x]_k$. Let also $A_0, \ldots, A_{m-1} \subseteq \mathbb{F}_q$ be m pairwise disjoint subsets of $\mathbb{F}_q$, each of size k/m. The m symbols we download from the i-th coordinate are

$$d_i^j = tr_{\mathbb{F}_{q^l}/\mathbb{F}_q}(\zeta_{l-m+j}c_i)(p_j(\gamma_i))^{(l-m)(j+1)} + \sum_{u=0}^{l-m-1} tr_{\mathbb{F}_{q^l}/\mathbb{F}_q}(\zeta_u c_i)(p_j(\gamma_i))^{u(j+1)}. \quad (21)$$

Substituting c_i by $h(\gamma_i)$ for all $i = 1, \ldots, n$, we see that $(d_1^j, \ldots, d_n^j) = (T_j(h)(\gamma_1), \ldots, T_j(h)(\gamma_n))$ is the j-th row of the virtual projection code $\mathcal{C}_{P_\alpha}$ of $\mathcal{C}$. Now by the fact that $|A_j| = k/m$ for all j and by the Corollary 12 we know that τ_{P_α} is given by

$$\tau_{P_\alpha} = \frac{1}{m+1}\left(mn + k\binom{m}{2} - \frac{k}{\alpha}\binom{m+1}{2}\right). \quad (22)$$

As $\sum_{j=0}^{m-1}|A_j| = k$, using the Algorithm 1 it is possible to recover the codeword $c \in \mathcal{C}$ with failure probability given by Theorem 14 if c has no more than $t \leqslant \tau_{P_\alpha}$ erros.

Note that if $m = 1$ then $\alpha = 1/l$ and

$$\tau_{P_\alpha} = \frac{1}{2}\left(n + k\binom{1}{2} - lk\binom{2}{2}\right) = \frac{1}{2}\left(n - \frac{k}{\alpha}\right) = \tau_\alpha.$$

For $m \geqslant 2$, we would like to improve the fractional decoding radius of $\mathcal{C}$, it means that we are interested in the case $\tau_{P_\alpha} \geqslant \tau_\alpha$

$$\tau_{P_\alpha} = \frac{1}{m+1}\left(mn + k\binom{m}{2} - \frac{k}{\alpha}\binom{m+1}{2}\right) \geqslant \frac{n - k/\alpha}{2}. \quad (23)$$

and it is possible to check that (23) is true if and only if

$$R = \frac{k}{n} \leqslant \frac{\alpha}{m(1-\alpha)+1} = \frac{m}{m(l-m)+l}. \tag{24}$$

This can be summarized in the following theorem.

Theorem 13. *Let $\mathcal{C} = RS(q^l, n, k)$ be a Reed-Solomon Code with evaluation set $\mathcal{L} = \{\gamma_1, \ldots, \gamma_n\} \subseteq \mathbb{F}_q$ and $\alpha = m/l$. If $m \geqslant 2$ and the rate of $\mathcal{C}$ is restricted as in (24) then the maximum α-decoding radius of $\mathcal{C}$ using Algorithm 1 is*

$$\tau_{P_\alpha} = \frac{1}{m+1}\left(mn + k\binom{m}{2} - \frac{k}{\alpha}\binom{m+1}{2}\right). \tag{25}$$

Moreover, in this case $\tau_{P_\alpha} \geqslant \tau_\alpha$.

IV. Failure Probability of the Algorithm I

The failure probability can be calculated in the same way that [6] and [9].

Note that the values of $T_{j_1}(e)(\gamma_i)$ and $T_{j_2}(e)(\gamma_i)$ do not depend of each other for all $j_1, j_2 \in \{0, \ldots, m-1\}$ and we can assume that if Y in (19) is corrupted by t errors, that is, $Y = C+E$ where E has t non-zero columns, then each non-zero column is an independent random vector uniformly distributed over $\mathbb{F}_q^m \setminus \{0\}$. Hence, we can apply Lemma 6 and Theorem 6 of [6] to upper bounded the failure probability of Algorithm 1.

Theorem 14. *Let $\mathcal{C} = RS(q^l, n, k)$ be a Reed-Solomon Code with evaluation set $\mathcal{L} = \{\gamma_1, \ldots, \gamma_n\} \subseteq \mathbb{F}_q$ and $\alpha = m/l$. If $m \geqslant 2$ and the rate of $\mathcal{C}$ is restricted as in (24). The probability for a decoding failure using the Algorithm 1 is upper bounded by*

$$P_{f_\alpha}(t) \leqslant \left(\frac{q^m - \frac{1}{q}}{q^m - 1}\right)^t \frac{q^{-(m+1)(\tau_{P_\alpha}-t)}}{q-1}.$$

Example 15. *Let $\mathcal{C} = RS(2^5, 31, 4)$ be a Reed-Solomon code with evaluation set $\mathcal{L} \in \mathbb{F}_q$ in this case the decoding radius of $\mathcal{C}$ is $\tau = 13$ and $R \simeq 0.1290$. By definition $\alpha = \frac{m}{5}$ and $\frac{4}{31} \leqslant \frac{m}{5} < 1$ thus $m \in \{1, 2, 3, 4\}$. Let $\alpha_i = \frac{i}{5}$ for $i = 2, 3, 4$ then for each α_i we have*

a) $\tau_{\alpha_1} = \tau_{P_{\alpha_1}} = 5$.
b) $\tau_{\alpha_2} = 10 < 12 = \tau_{P_{\alpha_2}}$.
c) $\tau_{\alpha_3} = 12 < 16 = \tau_{P_{\alpha_3}}$.
d) $\tau_{\alpha_4} = 13 < 19 = \tau_{P_{\alpha_4}}$.

The failure probability of c) is given in Table I.

TABLE I
FAILURE PROBABILITY $P_{f_{\alpha_3}}(t)$ FOR THE REED-SOLOMON CODE $RS(2^5, 31, 4)$.

t	12	13	14	15
$P_{f_{\alpha_3}}(t)$	2×10^{-6}	7×10^{-5}	2×10^{-3}	8×10^{-2}

Example 16. *Let $\mathcal{C} = RS(2^5, 31, 6)$ be a Reed-Solomon code with evaluation set $\mathcal{L} \in \mathbb{F}_q$ in this case the decoding radius of $\mathcal{C}$ is $\tau = \lfloor \frac{n-k}{2} \rfloor = 12$ and $R = \frac{k}{n} \simeq 0.1935$. By definition $\alpha = \frac{m}{5}$ and $\frac{6}{31} \leqslant \frac{m}{5} < 1$ thus $m \in \{1, 2, 3, 4\}$. If we denoted $\alpha_i = \frac{i}{5}$ for $i = 2, 3, 4$ then for each α_i we have*

a) $\tau_{\alpha_2} = 8 > \tau_{P_{\alpha_2}} = 7$. *This is due to the fact that $R \simeq 0.1935$ and $\frac{\alpha_2}{2(1-\alpha_2)+1} \simeq 0.1818$ that is (24) is not true in this case.*
b) $\tau_{\alpha_3} = 10 < 12 = \tau_{P_{\alpha_3}}$.
c) $\tau_{\alpha_4} = 11 < 16 = \tau_{P_{\alpha_4}}$. *Note that $\tau_{P_{\alpha_4}}$ is even greater than the decoding radius of $\mathcal{C}$. So, without accessing the entire codeword it is possible to recover more than $\lfloor \frac{n-k}{2} \rfloor$ errors with failure probability given in the Table II.*

TABLE II
FAILURE PROBABILITY $P_{f_{\alpha_4}}(t)$ FOR THE REED-SOLOMON CODE $RS(2^5, 31, 6)$.

t	11	12	13	14	15	16
$P_{f_{\alpha_4}}(t)$	10^{-9}	4×10^{-8}	10^{-6}	4×10^{-5}	10^{-3}	5×10^{-2}

Acknowledgment

I would like to thank professor Alexander Barg for his help during my internship at University of Maryland and his comments on this manuscript.

References

[1] E.R. Berlekamp, *Algebraic Coding Theory*. McGraw-Hill, New York 1968.
[2] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, *Network Coding for Distributed Storage Systems*, IEEE Trans. Inform. Theory, vol. 56, no. 9, pp. 4539-4551, 2010.
[3] I. Tamo, M. Ye, and A. Barg, *Fractional Decoding: Error Correction from Partial Information*, in Proc. 2017 IEEE International Symposium on Information Theory (ISIT), pp. 998-1002, 2017.
[4] D. Bleichenbacher, A. Kiayias, and M. Yung, *Decoding of Interleaved Reed Solomon Codes over a Noisy Data*, in Springer Lecture Notes in Computer Science (LNCS), vol. 2719, pp. 97-108, 2003.
[5] V. Y. Krachkovsky and Y. X. Lee, *Decoding for Interleaved Reed Solomon Schemes*, IEEE Trans. Magn., vol. 33, pp. 2740-2743, 1997.
[6] G. Schmidt, V. R. Sidorenko, and M. Bossert, *Collaborative Decoding of Interleaved Reed-Solomon Codes and Concatenated Code Designs*, IEEE Trans. Inform. Theory, vol. 55, no. 7, pp. 2991-3012, 2009.
[7] G. Schmidt, V. R. Sidorenko, and M. Bossert, *Syndrome Decoding of Reed-Solomon Codes Beyond Half the Minimum Distance Based on Shift-Register Synthesis*, IEEE Trans. Inform. Theory, vol. 56, no. 10, pp. 5245-5252, 2010.
[8] G. Schmidt, V. Sidorenko and M. Bossert, *Enhancing the Correcting Radius of Interleaved Reed-Solomon Decoding Using Syndrome Extension Techniques*, in Proc. 2007 IEEE International Symposium on Information Theory (ISIT), pp. 1341-1345, 2007.
[9] A. W. Zeh, A. Zeh and M. Bossert, *Decoding Interleaved Reed-Solomon Codes Beyond their Joint Error-Correcting Capability*, Designs, Codes and Cryptography vol. 71, pp. 261-281, 2014.
[10] V.Y. Krachkovsky, *Reed Solomon Codes for Correcting Phased Errors Bursts*, IEEE Trans. Inform. Theory vol. 49, pp. 2975-2984, 2003.
[11] W. Li, V. Sidorenko, and J. S. R. Nielsen, *On Decoding Interleaved Chinese Remainder Codes*, in Proc. 2013 IEEE International Symposium on Information Theory (ISIT), pp. 1052-1056, 2013.