Decoding of Interleaved Reed-Solomon Codes via Simultaneous Partial Inverses

Jiun-Hung Yu and Hans-Andrea Loeliger
Department of Information Technology and Electrical Engineering
ETH Zurich, Switzerland
Email: {yu, loeliger}@isi.ee.ethz.ch

***Abstract*—The partial-inverse approach is further developed to decoding interleaved Reed-Solomon codes and subfield-evaluation codes beyond half the minimum distance. The resulting decoding algorithm is new, and its decoding capability is shown to be state-of-the-art.**

I. Introduction

The partial-inverse approach to decoding Reed-Solomon codes and generalizations thereof was introduced in [1]. In this paper, we continue the extension of this approach to interleaved Reed-Solomon codes that was begun in [2].

Let $F = F_q$ be a finite field with q elements. We will consider codes where codewords are $L \times n$ arrays over F such that each row is a codeword in the same (n, k) Reed-Solomon code over F. We will only consider the correction of column errors, and we will not distinguish between columns with a single error and columns with many errors.

The Reed-Solomon code for each row will be defined as follows. Let $\beta_0, \ldots, \beta_{n-1}$ be n different elements of F. The code is then defined as the set

$$\big\{\big(a(\beta_0), \ldots, a(\beta_{n-1})\big) : a(x) \in F[x] \text{ with } \deg a(x) < k\big\}. \tag{1}$$

Note that such codes include both shortened and singly-extended Reed-Solomon codes.

It is well known [3]–[5] that such interleaved Reed-Solomon codes can equivalently be viewed as shortened Reed-Solomon codes over F_{q^L} simply by replacing $F[x] = F_q[x]$ in (1) by $F_{q^L}[x]$ while the evaluation points $\beta_0, \ldots, \beta_{n-1}$ remain in F_q. Note that symbol errors in F_{q^L} correspond to column errors in the array code.

Decoding such subfield-evaluation codes beyond the Guruswami-Sudan decoding radius [6] was pioneered in [3], [4], [7], [8], and decoding of interleaved Reed-Solomon codes has been much advanced in [5], [9]–[12]. Note that some of these papers use list-decoding algorithms [4], [6], [8] while others use unique-decoding algorithms that return at most one codeword [3], [5], [7], [9]–[12]. The best unique-decoding algorithms can now correct t errors (column errors or F_{q^L}-symbol errors) with t up to the bound

$$t \leq \frac{L}{L+1}\big(n-k\big) \tag{2}$$

with high probability if q is large [5], [7].

In [2], we began to study such codes from a partial-inverse perspective; we outlined a corresponding new unique-decoding algorithm and presented a new bound for the guaranteed error correction (repeated below as Theorem 2) patterned after a similar bound (for a different decoding algorithm) in [12].

In the present paper, we develop this approach further. In particular, we state and we analyze the new decoding algorithm more explicitly, and we complement the mentioned bound for guaranteed error correction by the following bound on the probability of decoding failure for random errors.

Theorem 1 (Probability of Decoding Failure). Assume $L > 1$. If the t nonzero columns of the error pattern are uniformly distributed over $F_q^L \setminus \{0\}$, then the probability P_f that the proposed decoding algorithm fails is bounded by

$$P_f < \frac{q^{-L(n-k)+(L+1)t}}{q-1} \tag{3}$$

□

The proof is given in Section VII.

Note that (3) implies that errors can be corrected (with high probability, if q is large) up to the bound (2).

The bound (3) almost agrees with, but is strictly better than, the bound of [5], and it beats the bound of [7]. The small, but positive, advantage over the bound of [5] appears to depend essentially on the partial-inverse approach. Moreover, the proof of Theorem 1 is shorter than the proof of the bound in [5].

We will see that the proposed decoding algorithm is very efficient even if L is large. In addition (and in contrast to the prior literature [5], [12]), the set $\{\beta_0, \ldots, \beta_{n-1}\}$ of evaluation points will be allowed to contain 0.

The paper is structured as follows. The partial-inverse algorithm from [2] is briefly recalled in Section II. The required basics about the codes are summarized in Section III. The new decoding algorithm is stated in Section IV. Guaranteed error correction is addressed in Sections V and VI, and the correction of random errors is addressed in Sections VI and VII.

II. The Simultaneous Partial-Inverse Problem and the SPI Algorithm

We begin by recalling the following material from [2].

Simultaneous Partial-Inverse (SPI) Problem: For $i = 1, 2, \ldots, L$, let $b^{(i)}(x)$ and $m^{(i)}(x)$ be nonzero polynomials over some field with $\deg b^{(i)}(x) < \deg m^{(i)}(x)$. The problem is to find a nonzero polynomial $\Lambda(x)$ of the smallest degree such that

$$\deg\Big(b^{(i)}(x)\Lambda(x) \bmod m^{(i)}(x)\Big) < \tau^{(i)} \tag{4}$$

for given $\tau^{(i)} \in \mathbb{Z}$ with $1 \leq \tau^{(i)} \leq \deg m^{(i)}(x)$. □

Proposition 1 (Uniqueness and Degree Bound). The solution $\Lambda(x)$ of the SPI Problem is unique (up to a scale factor) and satisfies

$$\deg \Lambda(x) \leq \sum_{i=1}^{L} \Big(\deg m^{(i)}(x) - \tau^{(i)}\Big). \tag{5}$$

□

Proposition 1 has no counterpart in the multi-sequence shift-register synthesis setting of [5]. The uniqueness of the solution will be used in the proof of Theorem 1.

As shown in [2], the SPI problem can be solved by the following algorithm, which we will later modify into a decoding algorithm.

Simultaneous Partial-Inverse (SPI) Algorithm

Input: $m^{(i)}(x)$, $b^{(i)}(x), \tau^{(i)}$ for $i = 1, \ldots, L$.
Output: $\Lambda(x)$ as in the problem statement.

```
1     for $i = 1, \ldots, L$ begin
2         $\Lambda^{(i)}(x) := 0$
3         $d^{(i)} := \deg m^{(i)}(x)$
4         $\kappa^{(i)} :=$ leading coefficient of $m^{(i)}(x)$
5     end
6     $\Lambda(x) := 1$
7     $\delta := \max_{i\in\{1,\ldots,L\}} \big(\deg m^{(i)}(x) - \tau^{(i)}\big)$
8     $i := 1$
9     loop begin
10        repeat
11            if $i > 1$ begin $i := i - 1$ end
12            else begin
13                if $\delta \leq 0$ return $\Lambda(x)$
14                $i := L$
15                $\delta := \delta - 1$
16            end
17            $d := \delta + \tau^{(i)}$
18            $\kappa :=$ coefficient of $x^d$ in
                      $b^{(i)}(x)\Lambda(x) \bmod m^{(i)}(x)$
19        until $\kappa \neq 0$
20        if $d < d^{(i)}$ begin
21            swap $(\Lambda(x), \Lambda^{(i)}(x))$
22            swap $(d, d^{(i)})$
23            swap $(\kappa, \kappa^{(i)})$
24            $\delta := d - \tau^{(i)}$
25        end
26        $\Lambda(x) := \kappa^{(i)}\Lambda(x) - \kappa x^{d-d^{(i)}}\Lambda^{(i)}(x)$
27    end
```

In the special case where $m^{(i)}(x) = x^{\deg m^{(i)}(x)}$, line 18 amounts to

```
41        $\kappa := b_d^{(i)}\Lambda_0 + b_{d-1}^{(i)}\Lambda_1 + \ldots + b_{d-\nu}^{(i)}\Lambda_\nu$
```

where $\nu \triangleq \deg \Lambda(x)$ and where $b_\mu^{(i)} \triangleq 0$ for $\mu < 0$. In another special case where $m^{(i)}(x) = x^{\deg m^{(i)}(x)} - 1$ for all i, line 18 becomes

```
51        $\kappa := b_d^{(i)}\Lambda_0 + b_{[d-1]}^{(i)}\Lambda_1 + \ldots + b_{[d-\nu]}^{(i)}\Lambda_\nu$
```

with $b_{[\mu]}^{(i)} \triangleq b_{\mu \bmod n}^{(i)}$. In both cases, the computation of line 18 only requires $O(n)$ operations, where $n \triangleq \max \deg m^{(i)}(x)$, and the algorithm has the complexity $O(Ln^2)$, cf. [2].

III. About the Codes

We will need the following (more or less standard) concepts.

A. Error Support and Error Locator Polynomial

Recall from Section I that we have an array code over F where each row is a codeword from a Reed-Solomon code as in (1). Let $Y = C + E \in F^{L\times n}$ be the received word where $C \in F^{L\times n}$ is the transmitted (array-) codeword and $E \in F^{L\times n}$ is the error pattern. The columns of codewords and error patterns will be indexed beginning with zero as in $E = (e_0, \ldots, e_{n-1})$. Let $U_E \subset \{0, \ldots, n-1\}$ be the index set of the nonzero columns of E, i.e.,

$$U_E \triangleq \big\{\ell \in \{0, \ldots, n-1\} : e_\ell \neq 0\big\}. \tag{6}$$

The error-locator polynomial is then defined as

$$\Lambda_E(x) \triangleq \prod_{j\in U_E} (x - \beta_j). \tag{7}$$

Note that

$$|U_E| = \deg \Lambda_E(x) = \text{number of column errors}. \tag{8}$$

B. Evaluation Isomorphism

Let $m(x) \triangleq \prod_{\ell=0}^{n-1}(x-\beta_\ell)$. Let ψ be the evaluation mapping

$$\psi : F[x]/m(x) \to F^n : a(x) \mapsto \big(a(\beta_0), \ldots, a(\beta_{n-1})\big), \tag{9}$$

which is a ring isomorphism. The row code (1) can then be described as

$$\{c \in F^n : \deg \psi^{-1}(c) < k\}. \tag{10}$$

In the special case where $\alpha \in F$ is a primitive n-th root of unity and $\beta_\ell = \alpha^\ell$, $\ell = 0, \ldots, n-1$, the mapping (9) is a discrete Fourier transform [15] and both ψ and ψ^{-1} may be computed by fast Fourier transform algorithms. In general, ψ^{-1} may be computed by Lagrange interpolation or according to the Chinese remainder theorem, cf., e.g., [13], [14].

C. Notation for Individual Rows

Let $y^{(i)}$ denote the i-th row of the matrix Y, let $c^{(i)}$ be the i-th row of C, and let $e^{(i)}$ be the i-th row of E. We then have $y^{(i)} = c^{(i)} + e^{(i)}$, $i = 1, \ldots, L$, and therefore

$$Y^{(i)}(x) = a^{(i)}(x) + E^{(i)}(x) \tag{11}$$

where $Y^{(i)}(x) \triangleq \psi^{-1}(y^{(i)})$, $a^{(i)}(x) \triangleq \psi^{-1}(c^{(i)})$, and $E^{(i)}(x) \triangleq \psi^{-1}(e^{(i)})$. Note that $\deg E^{(i)}(x) < \deg m(x) = n$ and $\deg a^{(i)}(x) < k$.

D. Error Locator Equation

With all this notation, we have

$$\deg\bigl(Y^{(i)}(x)\Lambda_E(x) \bmod m(x)\bigr) < k + \deg \Lambda_E(x) \qquad (12)$$

for $i = 1, \ldots, L$, since $E^{(i)}(x)\Lambda_E(x) \bmod m(x) = 0$.

IV. The New Decoding Algorithm

A. Locating the Errors

Eq. (12) makes it plausible that the error locator polynomial $\Lambda_E(x)$ may be found by the following algorithm.

Outline of Error-Locating Algorithm [2]:

1) Run the SPI algorithm of Section II with $b^{(i)}(x) = Y^{(i)}(x)$, $m^{(i)}(x) = m(x)$, and $\tau^{(i)} = n - 1$ for $i \in \{1, \ldots, L\}$.
2) If the returned polynomial $\Lambda(x)$ satisfies the condition
$$\deg\bigl(Y^{(i)}(x)\Lambda(x) \bmod m(x)\bigr) < k + \deg \Lambda(x) \qquad (13)$$
for every $i \in \{1, \ldots, L\}$, then stop.
3) Otherwise, decrease all $\tau^{(i)}$ by 1 and continue the **SPI** algorithm.
4) Go to 2). □

The test (13) actually requires no extra computations. Indeed, this error location method can be implemented by modifying the SPI algorithm of Section II as follows (which was not spelled out in [2]).

SPI Error Locating Algorithm:

Input: $Y^{(i)}(x)$, $i = 1, \ldots, L$.
Output: nonzero $\Lambda(x) \in F[x]$, a candidate for the error locator $\Lambda_E(x)$ (up to a scale factor).

The algorithm is the same as the **SPI** algorithm of Section II (with $b^{(i)}(x) = Y^{(i)}(x)$, $m^{(i)} = m(x)$, and $\tau^{(i)} = n - 1$), except that line 13 is replaced by following lines:

```
61    if $\delta \leq 0$ begin
62        if $d \leq \deg \Lambda(x) + k$ return $\Lambda(x)$
63        else begin
64            $\delta := \delta + 1$
65            for $j = 1, \ldots, L$ begin $\tau^{(j)} := \tau^{(j)} - 1$ end
66        end
67    end
```

Note that line 62 suffices to check (13) for all i. Note also that $\tau^{(1)} = \ldots = \tau^{(L)}$ throughout the algorithm.

A sufficient condition for the algorithm to return $\Lambda(x) = \gamma\Lambda_E(x)$ (for some nonzero $\gamma \in F$) is given by Theorem 3 in Section V.

B. Decoding

Putting things together, we have the following decoding algorithm.

1) Compute $Y^{(i)}(x) = \psi^{-1}(y^{(i)})$ for all i.
2) Run the SPI error locating algorithm to obtain a candidate $\Lambda(x)$ for the error locator polynomial.
3) Complete decoding in any standard way [15], or by means of
$$a^{(i)}(x) = \frac{Y^{(i)}(x)\Lambda(x) \bmod m(x)}{\Lambda(x)} \qquad (14)$$
as proposed in [13], or by
$$a^{(i)}(x) = Y^{(i)}(x) \bmod \tilde{m}(x) \qquad (15)$$
with $\tilde{m}(x) \triangleq m(x)/\Lambda(x)$ as proposed in [2].
4) If the user data is in the codeword C (the "time domain"), rather than in the polynomials $a^{(i)}(x)$ (the "frequency domain"), the additional computation of $\psi\bigl(a^{(i)}(x)\bigr)$, $i = 1, \ldots, L$, is required.

If the division in (14) does not work out, or if $\Lambda(x)$ does not divide $m(x)$, or if the resulting polynomials $a^{(i)}(x)$ do not satisfy $\deg a^{(i)}(x) < k$, then a decoding failure should be declared.

V. Guaranteed Error Correction

Recall the error support set U_E (6) and let r_E be the rank of the submatrix formed by the nonzero columns of E.

The justification of the SPI error locating algorithm hinges on the following theorem from [2].

Theorem 2. If
$$2|U_E| \leq n - k + r_E - 1, \qquad (16)$$
then the error locator polynomial (7) satisfies
$$\deg\Bigl(Y^{(i)}(x)\Lambda_E(x) \bmod m(x)\Bigr) < \frac{n + k + r_E - 1}{2} \qquad (17)$$
for all $i \in \{1, \ldots, L\}$. Conversely, for any Y and any $E \in F^{L \times n}$ (of rank r_E) and $t \in \mathbb{R}$ with
$$|U_E| \leq t \leq \frac{n - k + r_E - 1}{2} \qquad (18)$$
if some nonzero $\Lambda(x) \in F[x]$ with $\deg \Lambda(x) \leq t$ satisfies
$$\deg\Bigl(Y^{(i)}(x)\Lambda(x) \bmod m(x)\Bigr) < n - t + r_E - 1 \qquad (19)$$
for all $i \in \{1, \ldots, L\}$, then $\Lambda(x)$ is a multiple of $\Lambda_E(x)$. □

The direct part (17) is an immediate consequence of (12), but the converse part is not trivial.

We now state the consequences of Theorem 2 more carefully than outlined in [2].

Corollary 1. Assume that (16) holds. If some nonzero $\Lambda(x) \in F[x]$ with $\deg \Lambda(x) \leq |U_E|$ satisfies
$$\deg\Bigl(Y^{(i)}(x)\Lambda(x) \bmod m(x)\Bigr) < k + |U_E| \qquad (20)$$
for all $i \in \{1, \ldots, L\}$, then $\Lambda(x) = \gamma\Lambda_E(x)$ for some nonzero $\gamma \in F$. □

Proof: Eq. (16) can be rewritten as
$$k + |U_E| \leq n - |U_E| + r_E - 1. \qquad (21)$$

A polynomial $\Lambda(x)$ satisfying (20) thus satisfies (19) with $t = |U_E|$, and thus $\Lambda(x)$ is a multiple of $\Lambda_E(x)$. But $\deg \Lambda(x) \leq |U_E| = \deg \Lambda_E(x)$, and $\Lambda(x) = \gamma\Lambda_E(x)$ follows. □

Lemma 1 (SPI Error Location). If $\Lambda_E(x)$ is a solution of the SPI problem

$$\deg\Big(Y^{(i)}(x)\Lambda(x) \bmod m(x)\Big) < k + |U_E| \tag{22}$$

for all $i \in \{1, \ldots, L\}$, then the SPI error locating algorithm stops with $\tau^{(i)} \geq k + |U_E|$ and returns $\Lambda(x) = \gamma\Lambda_E(x)$ for some nonzero $\gamma \in F$. □

Proof: If $\Lambda_E(x)$ is a solution of the SPI problem (22), then, because of (12), the SPI error locating algorithm stops with $\tau^{(i)} \geq k + |U_E|$ and $\deg \Lambda(x) \leq \deg \Lambda_E(x)$. Because of (13), $\Lambda(x)$ satisfies (22), which is a contradiction unless $\deg \Lambda(x) = \deg \Lambda_E(x)$. □

Theorem 3 (Guaranteed Error Correction). If

$$2|U_E| < n - k + r_E, \tag{23}$$

then the SPI error locating algorithm of Section IV-A stops with $\tau^{(i)} \geq k + |U_E|$ and returns $\Lambda(x) = \gamma\Lambda_E(x)$ for some nonzero $\gamma \in F$. □

Proof: If (23) holds, then by Corollary 1, $\Lambda_E(x)$ is a solution of the SPI problem (22) for $i \in \{1, \ldots, L\}$, and the theorem follows from Lemma 1. □

The error correction capability guaranteed by Theorem 3 agrees with the guarantee in [12], which improves on Theorem 2 of [5] by a margin of $r_E/2$. Note that the rank r_E is not used in the algorithm and need not be computed.

VI. The Full-Rank Case

It is instructive to consider the special case where $r_E = |U_E|$, which is very likely if $|U_E| \leq L$, cf. Proposition 3 below. In this case, (23) reduces to

$$|U_E| < n - k, \tag{24}$$

and we have the following improvement on Theorem 3.

Proposition 2 (Full-Rank-Error Location). If

$$r_E = |U_E| < n - k, \tag{25}$$

the SPI error locating algorithm of Section IV-A stops with $\tau^{(j)} = n - 1$ (i.e., when (13) is checked for the first time) and returns $\Lambda(x) = \gamma\Lambda_E(x)$ for some nonzero $\gamma \in F$. □

Proof: Assume that (25) holds and consider the very first test of (13), where $\tau^{(i)} = n - 1$, $i = 1, \ldots, L$. We then have

$$\deg\Big(Y^{(i)}(x)\Lambda(x) \bmod m(x)\Big) < n - 1 \tag{26}$$

with $\deg \Lambda(x) \leq \deg \Lambda_E(x) = |U_E|$. Then (19) is satisfied with $t = |U_E|$, and it follows that $\Lambda(x) = \gamma\Lambda_E(x)$. Thus $\Lambda(x)$ passes the test (13) and the algorithm stops. □

Proposition 3 (Full-Rank Probability). If $|U_E| \leq L$ and if the $|U_E|$ nonzero columns of the $L \times n$ matrix E are uniformly and independently distributed over $F^L \setminus \{0\}$, then

$$\Pr\big(r_E \neq |U_E|\big) < \frac{q^{-L+|U_E|}}{q-1} \tag{27}$$

□

The proof is omitted.

Propositions 2 and 3 show, in particular, that the SPI error locating algorithm can be very efficient even if L is large.

For $|U_E| \leq L$, both (27) and (3) apply. In general, the bound (27) is much weaker than (3), but the bounds agree in the special case where $|U_E| = n - k - 1$; in this special case, these bounds agree also with the bounds in [9], [11], [12] (where different decoding algorithms are used).

VII. Proof of Theorem 1

We now consider random errors without the constraint $|U_E| \leq L$. Our main result here is Theorem 1, which was stated in Section I and will now be proved.

Let U be an arbitrary, but fixed, subset of $\{0, \ldots, n-1\}$, and assume that the $|U|$ nonzero columns of the $L \times n$ matrix E are uniformly and independently distributed over $F^L \setminus \{0\}$. We will prove the following: the probability of the event that the SPI error location algorithm of Section IV-A fails is bounded by

$$P_f < \frac{q^{-L(n-k)+(L+1)|U|}}{q-1} \tag{28}$$

for $L > 1$. Note that this bound depends only on $|U|$ (and not otherwise on U) and thus implies Theorem 1.

Recall the polynomial $E^{(i)}(x)$ from (11). The proof starts with the following fact, which is an immediate consequence of Lemma 1.

Proposition 4. Assume that error location fails, i.e., the SPI error locating algorithm returns $\tilde{\Lambda}(x) \neq \gamma\Lambda_E(x)$. Then there exists some nonzero polynomial $\Lambda(x)$ such that $\deg \Lambda(x) < |U|$ and

$$\deg\big(E^{(i)}(x)\Lambda(x) \bmod m(x)\big) < k + |U| \tag{29}$$

for all $i = 1, \ldots, L$. □

Let S_U be the set of all the possible error matrices E with the given support set U. Let $S_f \subset S_U$ be the set of all $E \in S_U$ that admit some $\Lambda(x) \in F[x]$ with $0 \leq \deg \Lambda(x) < |U|$ that satisfies (29) for all $i \in \{1, \ldots, L\}$. Then the probability P_f of failing to correct $E \in S_U$ is bounded by

$$P_f \leq \frac{|S_f|}{|S_U|} = \frac{|S_f|}{(q^L - 1)^{|U|}} \tag{30}$$

It thus remains to bound $|S_f|$.

For $t = 0, \ldots, |U| - 1$, let $\mathcal{L}_t$ be the set of monic polynomials $\Lambda(x) \in F[x]$ with $\deg \Lambda(x) < |U|$ and with exactly t zeros in the set $\mathcal{B}_U \triangleq \{\beta_\ell : \ell \in U\}$.

Lemma 2. For any fixed $\Lambda(x) \in \mathcal{L}_t$, the number of error patterns $E \in S_U$ that satisfy (29) is upper bounded by $q^{L(2|U|-(n-k)-t)}$. □

The proof will be given below. We then have

$$|S_f| \leq \sum_{t=0}^{|U|-1} |\mathcal{L}_t| \, q^{L(2|U|-(n-k)-t)}. \tag{31}$$

Lemma 3.

$$|\mathcal{L}_t| = \binom{|U|}{t} (q-1)^{|U|-t-1}. \tag{32}$$

□

The proof is given below. Thus (31) becomes

$$\begin{aligned}
|S_f| &\leq \sum_{t=0}^{|U|-1} \binom{|U|}{t} (q-1)^{|U|-t-1} q^{L(2|U|-(n-k)-t)} && (33)\\
&= w \sum_{t=0}^{|U|-1} \binom{|U|}{t} (q-1)^{-t} q^{-Lt} && (34)\\
&< w \sum_{t=0}^{|U|} \binom{|U|}{t} \big((q-1)^{-1} q^{-L}\big)^t && (35)\\
&= w\big(1 + (q-1)^{-1} q^{-L}\big)^{|U|} && (36)\\
&= \frac{q^{L(|U|-(n-k))}}{q-1} \big((q-1)q^L + 1\big)^{|U|} && (37)
\end{aligned}$$

with $w \triangleq (q-1)^{|U|-1} q^{L(2|U|-(n-k))}$ in (34)–(36). From (30), we then have

$$\begin{aligned}
P_f &< \frac{q^{L(|U|-(n-k))}}{q-1} \left(\frac{q^{L+1} - q^L + 1}{q^L - 1}\right)^{|U|} && (38)\\
&= \frac{q^{-L(n-k-|U|)+|U|}}{q-1} \left(\frac{q^L - (q^{L-1} - q^{-1})}{q^L - 1}\right)^{|U|} && (39)
\end{aligned}$$

and (28) follows if $L > 1$.

For the proof of Lemma 2, we will use the following elementary fact.

Proposition 5. The number of nonzero polynomials over F of degree at most ν and with $\mu \leq \nu$ prescribed zeros in F (and allowing additional zeros in F) is $|F|^{\nu-\mu+1} - 1$. □

Proof of Lemma 2: Consider the polynomial $E^{(i)}(x) = \psi^{-1}(e^{(i)})$ where $e^{(i)}$ is a row of E, and let $\tilde{E}^{(i)}(x) \triangleq E^{(i)}(x)\Lambda(x) \bmod m(x)$. From (9), we have

$$\tilde{E}^{(i)}(\beta_\ell) = e_{i,\ell} \Lambda(\beta_\ell) \tag{40}$$

where $e_{i,\ell}$ denotes the element in row i and column ℓ of E. From (29), we have $\deg \tilde{E}^{(i)}(x) < k + |U|$. But (40) implies that $\tilde{E}^{(i)}(x)$ has at least $n - |U| + t$ zeros in prescribed positions: $e_{i,\ell} = 0$ for $\ell \notin U$ and $\Lambda(x)$ has t zeros in $\mathcal{B}_U = \{\beta_\ell : \ell \in U\}$. By Proposition 5, the number of such polynomials $\tilde{E}^{(i)}$ is bounded by $q^{2|U|-(n-k)-t}$, and putting all rows together yields the lemma. □

Proof of Lemma 3: Consider nonzero polynomials $\Lambda(x) \in F[x]$ with $\deg \Lambda(x) < |U|$ and with t prescribed zeros in $\mathcal{B}_U$ ($= \{\beta_\ell : \ell \in U\}$) and no other zeros in $\mathcal{B}_U$. The number of such polynomials $\Lambda(x)$ is $(q-1)^{|U|-t}$, as is obvious from the ring isomorphism

$$F[x]/m_U(x) \to F^{|U|} : \Lambda(x) \mapsto \big(\Lambda(\beta'_1), \ldots, \Lambda(\beta'_{|U|})\big) \tag{41}$$

with $m_U(x) \triangleq \prod_{\ell \in U}(x - \beta_\ell)$ and $\{\beta'_1, \ldots, \beta'_{|U|}\} \triangleq \mathcal{B}_U$. Lemma 3 then follows from noting that it counts only monic polynomials. □

VIII. Conclusion

We have continued to develop the partial-inverse approach to decoding subfield-evaluation codes and interleaved Reed-Solomon codes. The proposed new decoding algorithm is as efficient as the best algorithms in the prior literature, and it permits to prove both the best bound for guaranteed error correction (using the rank of the error matrix) and the best bound on error probability for random errors; this combination has not previously been reported in the literature.

References

[1] J.-H. Yu and H.-A. Loeliger, “Reverse Berlekamp-Massey decoding,” *IEEE Int. Symp. on Information Theory,* Istanbul, Turkey, July 7–12, 2013.

[2] J.-H. Yu and H.-A. Loeliger, “An algorithm for simultaneous partial inverses,” *Proc. 52th Annual Allerton Conference on Communication, Control, and Computing,* Monticello, Illinois, USA, Oct. 1–3, 2014. Available from http://people.ee.ethz.ch/~loeliger/

[3] A. Brown, L. Minder, and M. A. Shokrollahi, “Probabilistic decoding of interleaved RS-codes on the Q-ary symmetric channel,” *IEEE Int. Symp. on Information Theory,* Chicago, USA, June 27–July 2, 2004.

[4] F. Parvaresh and A. Vardy, “On the performance of multivariate interpolation decoding of Reed-Solomon codes,” *IEEE Int. Symp. on Information Theory,* Seattle, USA, July 9–14, 2006.

[5] G. Schmidt, V. R. Sidorenko, and M.Bossert, “Collaborative decoding of interleaved Reed-Solomon codes and concatenated codes designs,” *IEEE Trans. Information Theory*, vol. 55, pp. 2991–3012, July 2009.

[6] V. Guruswami and M. Sudan, “Improved decoding of Reed-Solomon codes and algebraic-deometric codes,” *IEEE Trans. Information Theory*, vol. 45, pp. 1755–1764, Sept. 1999.

[7] D. Bleichenbacher, A. Kiayias, and M. Yung, “Decoding of interleaved Reed-Solomon codes over noisy data,” *Lect. notes Computer Sci.*, vol. 2719, pp. 97–108, 2003.

[8] F. Parvaresh and A. Vardy, “Multivariate interpolation decoding beyond the Guruswami-Sudan radius,” *Proc. 42th Annual Allerton Conference on Communication, Control, and Computing,* Urbana, Illinois, USA, October, 2004.

[9] J. J. Metzner and E. J. Kapturowski, “A general decoding technique applicable to replicated file disagreement location and concatenated code decoding,” *IEEE Trans. Information Theory*, vol. 36, pp. 911–917, July 1990.

[10] C. Haslach and A. J. H. Vinck, “A deoding algorithm with restrictions for array codes,” *IEEE Trans. Information Theory*, vol. 45, pp. 2339–2344, Nov. 1999 (and correction in the same publication, vol. 47, p. 470, Jan. 2001).

[11] H. Kurzweil, M. Seidl, and J. B. Huber, “Reduced-complexity collaborative decoding of interleaved Reed-Solomon and Gabidulin codes,” *IEEE Int. Symp. on Information Theory,* Saint Petersburg, Russia, July 31–Aug. 5, 2011.

[12] R. M. Roth and P. O. Vontobel, “Coding for combined block-symbol error correction,” *IEEE Trans. Information Theory*, vol. 60, pp. 2697–2713, May 2014.

[13] J.-H. Yu and H.-A. Loeliger, “On irreducible polynomial remainder codes,” *IEEE Int. Symp. on Information Theory,* Saint Petersburg, Russia, July 31–Aug. 5, 2011.

[14] J.-H. Yu and H.-A. Loeliger, “On polynomial remainder codes,” http://arxiv.org/abs/1201.1812.

[15] R. E. Blahut, *Algebraic Codes for Data Transmission.* Cambridge University Press, Cambridge, UK, 2003.