On the Joint Error-and-Erasure Decoding for Irreducible Polynomial Remainder Codes

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

Abstract—A general class of polynomial remainder codes is considered. Such codes are very flexible in rate and length and include Reed-Solomon codes as a special case.

As an extension of previous work, two joint error-and-erasure decoding approaches are proposed. In particular, both the decoding approaches by means of a fixed transform are treated in a way compatible with the error-only decoding. In the end, a collection of gcd-based decoding algorithm is obtained, some of which appear to be new even when specialized to Reed-Solomon codes.

I. INTRODUCTION

Polynomial remainder codes, constructed by means of the Chinese remainder theorem, were proposed by Stone [1], who also pointed out that these codes include Reed-Solomon codes [2] as a special case. Variations of Stone’s construction were studied in [3]–[5], but no efficient decoding algorithm for random error was presented in these papers. There is also a connection between Goppa [6] codes and polynomial remainder codes, as noted in [9].

In 1988, Shiozaki [7] proposed an efficient error-only decoding algorithm for Stone’s codes constructed by irreducible moduli [9]. However, the algorithm is restricted to codes with a fixed symbol size, i.e., fixed-degree moduli. This restriction was overcome by the decoding algorithms in [8], [9], which explicitly work for codes with variable symbol sizes, i.e., variable-degree moduli. Note that by adding moduli of different degrees, a Reed-Solomon code can be easily lengthened by adding some higher-degree symbols without increasing the size of the underlying field [8], [9]. Note also that Shiozaki’s algorithm, when applied to Reed-Solomon codes, is the same as Gao’s [10] algorithm as pointed out in [9], [11], [12].

In presence of both error and erasures, the error-only decoding algorithm, as observed by Shiozaki [7], can be applied to shortened polynomial remainder codes interpolated by ignoring the erased symbols. Such an interpolation, however, involves a lot of re-computation of the interpolating basis and thus greatly increases the decoding complexity. When applied to Reed-Solomon codes, the same problem exists, as pointed out in [12], but for Reed-Solomon codes, the problem can be bypassed by the decoding algorithm of [12].

In this paper, we consider the extension of the error-only decoding algorithms of [8], [9] to joint error-and-erasure decoding of irreducible polynomial remainder codes. Two fixed-transform approaches are proposed for decoding such codes. When applied to Reed-Solomon codes, the first approach is essentially identical to the one in [12], but the second approach appears to be new. For each approach, the decoding algorithm consists of two steps: in the first step, a polynomial which factorizes the error locator polynomial is computed by means of a gcd algorithm; in the second step, the message is recovered, for which we also propose two different methods.

The paper is organized as follows. In Section II, we recall the Chinese remainder theorem and the definition of irreducible polynomial remainder codes. In Section III we address the problem of joint error-and-erasure decoding and propose two fixed-transform decoding approaches. In Sections IV and V we derive gcd-based decoding algorithms for the respective approaches. A collection of these algorithms is summarized in Section VI. Section VII concludes the paper.

II. IRREDUCIBLE POLYNOMIAL REMAINDER CODES

In this section, we quickly recall the Chinese remainder theorem, the definition of irreducible polynomial remainder codes, and some basic properties of such codes as in [8], [9].

Let $R = F[x]$ be the ring of polynomials over some field $F$. For any monic polynomial $m(x) \in F[x]$, let $R_m$ denote the ring of polynomials over $F$ of degree less than $\deg m(x)$ with addition and multiplication modulo $m(x)$.

A. CRT Theorem and Polynomial Remainder Codes

Theorem 1 (Chinese Remainder Theorem). For some integer $n > 1$, let $m_0(x), m_1(x), \ldots, m_{n-1}(x) \in R$ be relatively prime polynomials, and let $M_n(x) \triangleq \prod_{i=0}^{n-1} m_i(x)$. The mapping

$$
\psi : R_{M_n} \rightarrow R_{m_0} \times \ldots \times R_{m_n} : a(x) \mapsto (\psi_0(a), \ldots, \psi_{n-1}(a))
$$

with $\psi_i(a) \triangleq a(x) \mod m_i(x)$ is a ring isomorphism. The inverse mapping is

$$
\psi^{-1} : (c_0, \ldots, c_{n-1}) \mapsto \sum_{i=0}^{n-1} c_i(x) \beta_i(x) \mod M_n(x)
$$

with coefficients

$$
\beta_i(x) = \frac{M_n(x)}{m_i(x)} \cdot \left( \frac{M_n(x)}{m_i(x)} \right)^{-1} \mod m_i(x)
$$

where $(b(x))^{-1} \mod m_i(x)$ denotes the inverse of $b(x)$ in $R_{m_i}$. □
Definition 1. An irreducible polynomial remainder code over $F[x]$ is a set of the form

$$C \equiv \{(c_0, \ldots, c_{n-1}) \in \psi(a) \text{ for some } a(x) \in R_{M_k}\}$$

(4)

where $n$ and $k$ are integers satisfying $1 \leq k \leq n$, where $m_0(x), m_1(x), \ldots, m_{n-1}(x) \in F[x]$ are different monic irreducible polynomials, and where $M_k(x) \equiv \prod_{i=0}^{k-1} m_i(x)$. □

B. Distance and Error Correction

Let $C$ be a code as in Definition 1. Let $y = c + e$ denote a corrupted codeword that the receiver gets to see, where $c \in C$ is the transmitted codeword corresponding to some $a(x) \in R_{M_k}$. Where $e$ is an error pattern.

For any $a(x) \in R_{M_k}$, the degree weight of $\psi(a) = (\psi_0(a), \ldots, \psi_{n-1}(a))$ is

$$w_D(\psi(a)) \equiv \sum_{i; \psi_i(a) \neq 0} \deg m_i(x).$$

(5)

For any $a(x), b(x) \in R_{M_k}$, the degree-weighted distance between $\psi(a)$ and $\psi(b)$ is

$$d_D(\psi(a), \psi(b)) \equiv w_D(\psi(a) - \psi(b)).$$

(6)

Let

$$N \equiv \deg M_n(x) = \sum_{i=0}^{n-1} \deg m_i(x)$$

(7)

and

$$K \equiv \deg M_k(x) = \sum_{i=0}^{k-1} \deg m_i(x).$$

(8)

Then, the degree weight of any nonzero codeword $\psi(a) (a(x) \in R_{M_k}, a(x) \neq 0)$ satisfies

$$w_D(\psi(a)) > N - K$$

(9)

and the minimum degree-weighted distance of $C$ satisfies

$$d_{\min}(C) > N - K.$$  

(10)

If $C$ also satisfies the Ordered-Degree Condition

$$\deg m_0(x) \leq \deg m_1(x) \leq \ldots \leq \deg m_{n-1}(x),$$

(11)

then the Hamming weight of any nonzero codeword $\psi(a)$ ($a(x) \in R_{M_k}, a(x) \neq 0$) satisfies $w_H(\psi(a)) \geq n - k + 1$ and the minimum Hamming distance of $C$ is $d_{\min}(C) = n - k + 1$.

An error-only decoding algorithm, which is guaranteed to correct all the error patterns of $w_D(e) < d_{\min}(C)/2$ and also the error patterns of $w_H(e) < d_{\min}(C)/2$ if the code satisfies (11), was proposed in [8, 9] to deal with the error $e = (e_0, e_1, \ldots, e_{n-1})$ with unknown error positions (i.e. $e_i \neq 0, 0 \leq i < n - 1$, are unknown). Moreover, an efficient interpolation formula was also proposed in [3, 9] to recover $a(x)$ from $y = c + e$ when the positions $i$ of $e_i$ are all known.

In the following, we consider the problem where only some (rather than all) positions $i$ of $e_i$ are known before decoding.

III. ERROR-AND-ERASURE DECODING

In this section, we present three possible approaches to joint error-and-erasure decoding of the code $C$ as in Definition 1.

Let $y = c + e$ denote a corrupted codeword, where $c \equiv (c_0, c_1, \ldots, c_{n-1}) \in C$ and where $e = (e_0, e_1, \ldots, e_{n-1})$ is an error pattern. Let $S_c \subset \{0, 1, \ldots, n - 1\}$ denote the set of positions $i$ of $e_i \neq 0$. Let $S_e \subset S_c$ denote the set of known positions $i$ of $e_i = 0$ and let $S_e \equiv S_c \setminus S_e$ denote the set of the unknown positions $i$ of $e_i \neq 0$.

A. A Modified-Transform Approach

A first approach, as observed by Shiozaki [7], is to reduce the joint error-and-erasure decoding of such codes to the error-only decoding of the shortened codes. Specifically, let $\mathcal{S} \equiv \{0, 1, \ldots, n - 1\} \setminus S_e$, let $M_S(x) \equiv \prod_{i \in \mathcal{S}} m_i(x)$, and let $\mathcal{R} \equiv \langle \prod_{i \in \mathcal{S}} R_{m_i} \rangle$ denote the direct product of the rings $R_{m_i}$ with $i \in \mathcal{S}$. Moreover, let $\tilde{c} \equiv (c_i)$ with $i \in \mathcal{S}$, i.e., $\tilde{c}$ is the shortened codeword of $c$. It then follows from Theorem 1 that the mapping $\phi: R_{M_S} \rightarrow \mathcal{R}$ is a ring isomorphism. The inverse mapping is

$$\phi^{-1}: \tilde{c} \mapsto \sum_{i \in \mathcal{S}} c_i(x) \beta_i(x) \bmod M_S(x)$$

(12)

with interpolating basis

$$\beta_i(x) = M_S(x) m_i(x) \bmod m_i(x)^{-1}$$

(13)

We can then use the error-only decoding algorithms as in [8, 9] to decode $\tilde{c}$. This approach requires, however, a lot of re-computation of (13) and thus greatly increases the decoding complexity, as the case for Reed-Solomon codes [12].

In the following two subsections, we propose two other approaches which avoid the re-computation of (13) and use the fixed transform $\psi^{-1}$ and the fixed $\beta_i(x)$ in (2) and (3).

B. A Fixed-Transform Approach I

Recall that $y = c + e$. Let $Y(x) = a(x) + E(x)$ denote the pre-image $\psi^{-1}(y)$ of $y$ with $\psi^{-1}$ as in (2), where $a(x) = \psi^{-1}(e)$ of deg $a(x) < K$ and where $E(x) = \sum_{\ell=0}^{N-1} E_\ell x^\ell$ denotes the pre-image $\psi^{-1}(e)$ of $e$.

Let

$$\Lambda_\tau(x) \equiv \prod_{i \in \mathcal{S}_e} m_i(x)$$

(14)

be the unique monic error locator polynomial of the smallest degree $\deg \Lambda_\tau(x) = \deg \lambda_\tau(x)$. With

$$\Lambda_\rho(x) \equiv \prod_{i \in \mathcal{S}_e} m_i(x)$$

(15)

and

$$\Lambda_\sigma(x) \equiv \prod_{i \in \mathcal{S}_e} m_i(x),$$

(16)

[14] can then be written as $\Lambda_\tau(x) = \Lambda_\rho(x) \Lambda_\sigma(x)$, and the key equation in Theorem 6 of [8] can be written as

$$A(x) M_n(x) = \Lambda_\rho(x) \Lambda_\tau(x) E(x).$$

(17)
Now let
\[ \hat{E}(x) = \Lambda_\rho(x)E(x). \] (18)
and
\[ \hat{Y}(x) = \Lambda_\rho(x)Y(x) = \Lambda_\rho(x)a(x) + \hat{E}(x). \] (19)

**Theorem 2.** The polynomial (16) satisfies
\[ A(x)M_n(x) = \Lambda_\tau(x)\hat{E}(x) \] (20)
for some polynomial \( A(x) \in F[x] \) of degree smaller than \( \deg \Lambda_e(x) = \Lambda_\rho(x) + \Lambda_\tau(x) \). Conversely, if some polynomial \( G(x) \in F[x] \) satisfies
\[ A(x)M_n(x) = G(x)\hat{E}(x) \] (21)
for some \( A(x) \in F[x] \), then \( G(x) \) is a multiple of \( \Lambda_\tau(x) \). \( \square \)

**Theorem 3 (Fixed-Transform Interpolation).** If \( G(x) \) is a multiple of \( \Lambda_\tau(x) \) with
\[ \deg G(x) \leq N - K - \deg \Lambda_\rho(x), \] (22)
then
\[ a(x) = \frac{\hat{Y}(x)G(x) \mod M_n(x)}{\Lambda_\rho(x)G(x)} \] (23)
\[ r(x) := M_n(x) \]
\[ \tilde{r}(x) := \hat{E}(x) \]
\[ s(x) := 1 \]
\[ t(x) := 0 \]
\[ \tilde{s}(x) := 0 \]
\[ \tilde{t}(x) := 1 \]

**Extended GCD Algorithm I**

Input: \( M_n(x) \) and \( E(x) \).
Output: polynomials \( s(x) \) and \( t(x) \in F[x] \), cf. Theorem 6

**Theorem 6 (GCD Output).** When the algorithm terminates, we have both
\[ s(x) \cdot M_n(x) + t(x) \cdot \hat{E}(x) = 0 \] (30)
and
\[ t(x) = \tilde{\gamma}\Lambda_\tau(x) \] (31)
for some scalar \( \tilde{\gamma} \in F \). \( \square \)

IV. Solving (21) by the Extended GCD Algorithm

It is known that an extended gcd algorithm can be used to solve a key equation and compute an error locator polynomial \([8], [9]\), which is also one of the standard ways of decoding Reed-Solomon codes \([14]\). We now adapt this approach to solve the modified key equation (21).

A. An Extended GCD Algorithm

In this subsection, we assume that \( \hat{E}(x) \neq 0 \) is fully known; in the next subsection, we state the modifications that are required when \( \hat{E}(x) \) is only partially known. We prefer the following gcd algorithm \([8], [9]\).

**Extended GCD Algorithm I**

Input: \( M_n(x) \) and \( \hat{E}(x) \).
Output: polynomials \( s(x) \) and \( t(x) \in F[x] \), cf. Theorem 6.
B. Modifications for Partially Known \( E(x) \)

Recall that \( Y(x) = a(x) + E(x) \). Since \( \deg a(x) < K \) the receiver knows the coefficients \( E_K, E_{K+1}, \ldots, E_{N-1} \) of \( E(x) \). It follows from (18) and (19) that the upper \( N - K \) coefficients of \( E(x) \), obtained from \( Y(x) \), are also known, which can then be used to compute \( t(x) = \gamma \Lambda_r(x) \) as follows.

Partial GCD Algorithm I

Input: \( M_n(x) \) and \( Y(x) \).
Output: \( r(x), s(x), t(x) \), cf. Theorem 7 below.

The algorithm is the same as the Extended GCD Algorithm I of Section IV-A except for the following changes:

- Line 2: \( \tilde{r}(x) := \tilde{Y}(x) \)
- Line 17

\[
\text{if } \deg r(x) < \deg t(x) + \deg \Lambda_\rho(x) + K \text{ begin} \quad (32)
\]

or alternatively

\[
\text{if } \deg r(x) < (N + K + \deg \Lambda_\rho(x))/2 \text{ begin} \quad (33)
\]

Theorem 7. If \( \Lambda_r(x) \) satisfies

\[
\deg \Lambda_r(x) \leq (N - K - \deg \Lambda_\rho)/2, \quad (34)
\]

then the Partial GCD Algorithm I (with either (32) or (33)) returns the same polynomials \( s(x) \) and \( t(x) \) (after the same number of iterations) as the Extended GCD Algorithm I of Section IV-A. Moreover, the returned \( r(x) \) is such that

\[
r(x) = t(x) \Lambda_\rho(x) a(x). \quad (35)
\]

Note that \( a(x) \) can be recovered directly from (35).

V. SOLVING (20) BY THE EXTENDED GCD ALGORITHM

A. An Extended GCD Algorithm

The following Extended GCD Algorithm II, which is fully described for clarity and ease of reference, is the same as the Extended GCD Algorithm I in Section IV-A except having different input polynomials. We first assume that \( E(x) \neq 0 \) is fully known, and then in the next subsection, we state the required modifications when \( E(x) \) is partially known.

Extended GCD Algorithm II

Input: \( \tilde{M}_n(x) \) and \( E(x) \).
Output: polynomials \( s(x) \) and \( t(x) \in F[x] \).

1. \( r(x) := \tilde{M}_n(x) \)
2. \( \tilde{r}(x) := E(x) \)
3. \( s(x) := 1 \)
4. \( t(x) := 0 \)
5. \( \tilde{s}(x) := 0 \)
6. \( \tilde{t}(x) := 1 \)
7. \text{loop begin}
8. \( \quad i := \deg r(x) \)
9. \( \quad j := \deg \tilde{r}(x) \)
10. \text{while } i \geq j \text{ begin}

For this algorithm, the loop invariant

\[
r(x) = s(x) \cdot \tilde{M}_n(x) + t(x) \cdot E(x), \quad (36)
\]

holds between lines 9 and 10 and between lines 16 and 17.

Theorem 8 (GCD Output). When the algorithm terminates, we have both

\[
s(x) \cdot \tilde{M}_n(x) + t(x) \cdot E(x) = 0. \quad (37)
\]

and

\[
t(x) = \tilde{\gamma} \Lambda_r(x) \quad (38)
\]

for some scalar \( \tilde{\gamma} \in F \).

B. Modifications for Partially Known \( E(x) \)

Recall that the known coefficients \( E_K, E_{K+1}, \ldots, E_{N-1} \) of \( E(x) \) can be obtained from \( Y(x) \).

Partial GCD Algorithm II

Input: \( \tilde{M}_n(x) \) and \( Y(x) \).
Output: \( r(x), s(x), t(x) \), cf. Theorem 9 below.

The algorithm is the same as the Extended GCD Algorithm II of Section IV-A except for the following changes:

- Line 2: \( \tilde{r}(x) := Y(x) \)
- Line 17: \( \text{if } \deg r(x) < \deg t(x) + K \text{ begin} \quad (39) \)

or alternatively

\[
\text{if } \deg r(x) < (N + K - \deg \Lambda_\rho(x))/2 \text{ begin} \quad (40)
\]

Theorem 9. If the condition (34) is satisfied, then the Partial GCD Algorithm II (with either (39) or (40)) returns the same polynomials \( s(x) \) and \( t(x) \) (after the same number of iterations) as the Extended GCD Algorithm II of Section IV-A. Moreover, the returned \( r(x) \) is such that

\[
r(x) = t(x) \Lambda_\rho(x), \quad (41)
\]

Note that \( a(x) \) can be recovered directly from (41).
VI. SUMMARY OF DECODING

Let us summarize the proposed decoding algorithm and add some details. The receiver sees \( y = c + e \) where \( c \in C \) is the transmitted codeword and \( e \) is an error pattern. We thus have \( Y(x) = a(x) + E(x) \) where \( Y(x) \), \( a(x) \), and \( E(x) \) are the images of \( y \), \( c \), and \( e \) under the fixed transform \( \psi^{-1} \) and \( \deg a(x) < K \).

A. Decoding using Fixed-Transform Approach I

By Fixed-Transform Approach I, we first compute \( \hat{Y}(x) \) from \( (19) \), and then run the Partial GCD Algorithm I. If \( (34) \) is satisfied, then the algorithm yields \( s(x) \), \( t(x) \) and \( r(x) \) that satisfy \( (30), (31) \) and \( (35) \). We can then recover \( a(x) \) from \( (19) \), and then run the Partial GCD Algorithm I. If \( (34) \) is satisfied, then the algorithm yields \( a(x) \) from \( (24) \), and then run the Partial GCD Algorithm II.

B. Decoding using Fixed-Transform Approach II

When applied to Reed-Solomon codes, Approach I with the recovery of \( \hat{Y}(x) \) is identical to the algorithm proposed in \( [12] \), but recovering \( a(x) \) by \( (42) \) is new.

VII. CONCLUSION

We have extended previous work on the error-only decoding of irreducible polynomial remainder codes to the joint error-and-erasure decoding of such codes, for which we have proposed two fixed-transform approaches. As we have shown, for each approach, the joint error-and-erasure decoding is carried out by an efficient gcd algorithm, and is fully compatible in implementation with the error-only decoding. Of particular interest is the second approach, which appears to be new even when specialized to Reed-Solomon codes.

VIII. ACKNOWLEDGEMENT

The author is deeply grateful to Prof. H.-A. Loeliger for his encouragement and great support of this work.

REFERENCES

[1] J. J. Stone, “Multiple-burst error correction with the Chinese Remainder Theorem,” J. SIAM, vol. 11, pp. 74–81, Mar. 1963.
[2] I. S. Reed and G. Solomon, “Polynomial codes over certain finite fields,” J. SIAM, vol. 8, pp. 300–304, Oct. 1962.
[3] D. C. Bossen and S. S. Yau, “Redundant residue polynomial codes,” Information and Control, vol. 13, pp. 597–618, 1968.
[4] D. Mandelbaum, “A method of coding for multiple errors,” IEEE Trans. Information Theory, vol. 14, pp. 518–621, May 1968.
[5] D. Mandelbaum, “On efficient burst correcting residue polynomial codes,” Information and Control, vol. 16, pp. 319–330, 1970.
[6] V. D. Goppa, “A new class of linear error-correction codes,” Probl. Peredach. Inform., vol. 6, pp. 24–30, Sept. 1970.
[7] A. Shiozaki, “Decoding of redundant residue polynomial codes using Euclid’s algorithm,” IEEE Trans. Information Theory, vol. 34, pp. 1351–1354, Sep. 1988.
[8] 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.
[9] J.-H. Yu and H.-A. Loeliger, “On polynomial remainder codes,” submitted to IEEE Trans. Information Theory. (available on http://arxiv.org/abs/1201.1382)
[10] S. Gao, “A new algorithm for decoding Reed-Solomon codes,” in Communications, Information and Network Security, V. Bhargava, H. V. Poor, V. Tarokh, and S. Yoon, Eds. Norwell, MA: Kluwer, 2003, vol. 712, pp. 55–68.
[11] S. V. Fedorenko, “Correction to ‘A simple algorithm for decoding Reed-Solomon codes and its relation to the Welch-Berlekamp algorithm’,” IEEE Trans. Information Theory, vol. IT-52, pp. 1278, Mar. 2006.
[12] T.-C. Lin, P.-D. Chen, and T.-K. Truong “Simplified procedure for decoding nonsystematic Reed-Solomon codes over GF(2^m) using Euclid’s algorithm and the fast Fourier transform,” IEEE Trans. Communications, vol. 57, pp. 1588–1592, Jun. 2009.
[13] Y. Sugiyama, M. Kasahara, S. Hisasawa, and T. Namekawa, “A method for solving key equation for decoding Goppa codes,” Information and Control, vol. 27, pp. 87–99, 1975.
[14] R. M. Roth, Introduction to Coding Theory. New York: Cambridge University Press, 2006.