On Irreducible Polynomial Remainder Codes

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*—A general class of polynomial remainder codes is considered. These codes are very flexible in rate and length and include Reed-Solomon codes as a special case. In general, the code symbols of such codes are polynomials of different degree, which leads to two different notions of weights and of distances.**

The notion of an error locator polynomial is generalized to such codes. A key equation is proposed, from which the error locator polynomial can be computed by means of a gcd algorithm. From the error locator polynomial, the transmitted message can be recovered in two different ways, which may 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 [2], who also pointed out that these codes include Reed-Solomon codes [1] as a special case. Variations of Stone's construction were studied in [3]–[5]. In [2] and [3], the focus is on codes with a fixed symbol size, i.e., the moduli are relatively prime polynomials of the same degree. Mandelbaum proposed a generalized encoding rule [4] and pointed out that using moduli of different degrees can be advantageous for burst error correction [5]. Although the codes in [2]–[5] can, in principle, correct many random errors, no efficient decoding algorithm for random errors was proposed in these papers. In 1988, Shiozaki [6] proposed an efficient decoding algorithm for Stone's codes [2] using Euclid's algorithm, and he also adapted this algorithm to decode Reed-Solomon codes. However, the algorithm of [6] is restricted to codes with a fixed symbol size, i.e., fixed-degree moduli.

There is also a body of work on Chinese remainder codes over integers, cf. [7], [8]. However, the results of the present paper are not directly related to that work.

In this paper, we revisit polynomial remainder codes and propose a practical decoding algorithm. In contrast to most prior work, we explicitly allow moduli of different degrees (i.e., variable symbol sizes) within a codeword. In consequence, we obtain two different notions of distance—Hamming distance and degree-weighted distance—and the corresponding minimum-distance decoding rules. By admitting moduli of different degrees, we can, e.g., lengthen a Reed-Solomon code by adding some higher-degree symbols without increasing the size of the underlying field.

The proposed decoding algorithm consists of two steps: in the first step, an error locator polynomial is computed by means of a gcd algorithm; in the second step, the message is recovered, for which we propose two different methods. When applied to Reed-Solomon codes, the first step is standard but the second step may be new.

The paper is organized as follows. In Section II, we recall the Chinese Remainder Theorem and define irreducible polynomial remainder codes. In Section III, we introduce two types of minimum distance decoders as well as basic error and erasure correction bounds. In Section IV, we introduce error locator polynomials and we present a key equation as well as two additional theorems. In Section V, we describe a modified Euclidean algorithm for solving the key equation. The resulting practical decoding algorithm is summarized in Section VI. A extension of this algorithm is outlined in Section VII. Section VIII concludes the paper.

The theorems and decoding algorithms of this paper are stated without proofs; for the proofs, we refer to [9].

II. Chinese Remainder Theorem and Polynomial Remainder Codes

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 $m(x)$.

We will need the Chinese Remainder Theorem [2] in the following form.

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

$$\begin{aligned} \psi : R_{M_n} &\to R_{m_0} \times \ldots \times R_{m_n} : \\ a(x) &\mapsto \psi(a) \triangleq \big(\psi_0(a), \ldots, \psi_{n-1}(a)\big) \end{aligned} \tag{1}$$

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

$$\psi^{-1} : (c_0, \ldots, c_{n-1}) \mapsto \sum_{i=0}^{n-1} c_i(x)\beta_i(x) \bmod M_n(x) \tag{2}$$

with coefficients

$$\beta_i(x) = \frac{M_n(x)}{m_i(x)} \cdot \left(\frac{M_n(x)}{m_i(x)}\right)^{-1}_{\bmod m_i(x)} \tag{3}$$

where $\big(b(x)\big)^{-1}_{\bmod m_i(x)}$ denotes the inverse of $b(x)$ in R_{m_i}. □

We will henceforth assume that $m_0(x), \ldots, m_{n-1}(x)$ are different monic irreducible polynomials in $R = F[x]$.

Definition 1. For different monic irreducible polynomials $m_0(x), \ldots, m_{n-1}(x)$ and some fixed integer k, $1 \le k \le n$, an *irreducible polynomial remainder code* is the image of ψ as in (1) of polynomials $a(x)$ of degree less than $\deg M_k(x)$ with $M_k(x) \triangleq \prod_{i=0}^{k-1} m_i(x)$, i.e.,

$$C = \big\{(c_0, \ldots, c_{n-1}) = \psi(a) \text{ for some } a(x) \in R_{M_k}\big\}. \quad (4)$$

□

Note that such codes are linear (i.e., vector spaces) over F.

The components $c_i = \psi_i(a)$ in (1) and (4) will be called *symbols*. Note that each symbol is from a different ring R_{m_i}; these rings need not have the same number of elements.

Let $N \triangleq \deg M_n(x) = \sum_{i=0}^{n-1} \deg m_i(x)$ and $K \triangleq \deg M_k(x) = \sum_{i=0}^{k-1} \deg m_i(x)$. The number of codewords of a code C as in (4) is $|F|^K$. By the *rate* of the code, we mean the quantity

$$\frac{1}{N} \log_{|F|} |C| = \frac{K}{N}. \quad (5)$$

In the special case where all the moduli $m_0(x), \ldots, m_{n-1}(x)$ have the same degree, we have $K/N = k/n$.

In the special case where all moduli $m_0(x), \ldots, m_{n-1}(x)$ are (different) monic polynomials of degree one, all symbols are in F and the code is a Reed-Solomon code. By adding some moduli of degree 2, we can lengthen a Reed-Solomon code without increasing the size of the underlying field.

We will usually assume that the moduli $m_i(x)$ in Definition 1 satisfy the *Ordered-Degree Condition*

$$\deg m_0(x) \le \deg m_1(x) \le \ldots \le \deg m_{n-1}(x). \quad (6)$$

III. Distances and Error Correction

For any $a(x) \in R_{M_n}$, the Hamming weight of $\psi(a)$ (i.e., the number of nonzero symbols $\psi_i(a)$, $0 \le i \le n-1$) will be denoted by $\mathrm{w_H}(\psi(a))$. For any $a(x), b(x) \in R_{M_n}$, the Hamming distance between $\psi(a)$ and $\psi(b)$ will be denoted by $\mathrm{d_H}(\psi(a), \psi(b)) \triangleq \mathrm{w_H}(\psi(a) - \psi(b))$. The minimum Hamming distance of a code C will be denoted by $\mathrm{d_{minH}}(C)$.

Theorem 2. Let C be a code as in Definition 1 satisfying (6). Then the Hamming weight of any nonzero codeword $\psi(a)$ ($a(x) \in R_{M_k}$, $a(x) \ne 0$) satisfies

$$\mathrm{w_H}(\psi(a)) \ge n - k + 1 \quad (7)$$

and

$$\mathrm{d_{minH}}(C) \ge n - k + 1. \quad (8)$$

□

Definition 2. For any $a(x) \in R_{M_n}$, the *degree weight* of $\psi(a) = \big(\psi_0(a), \ldots, \psi_{n-1}(a)\big)$ is

$$\mathrm{w_D}(\psi(a)) \triangleq \sum_{i:\psi_i(a)\ne 0} \deg m_i(x). \quad (9)$$

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

$$\mathrm{d_D}(\psi(a), \psi(b)) \triangleq \mathrm{w_D}(\psi(a) - \psi(b)). \quad (10)$$

□

Moreover, the *minimum degree-weighted distance* of an irreducible polynomial remainder code C is

$$\mathrm{d_{minD}}(C) \triangleq \min_{c,c' \in C: c \ne c'} \mathrm{d_D}(c, c'). \quad (11)$$

We then have the following analog of Theorem 2:

Theorem 3. Let C be a code as in Definition 1. Then the degree weight of any nonzero codeword $\psi(a)$ ($a(x) \in R_{M_k}$, $a(x) \ne 0$) satisfies

$$\mathrm{w_D}(\psi(a)) \ge N - K + 1 \quad (12)$$

and

$$\mathrm{d_{minD}}(C) \ge N - K + 1. \quad (13)$$

□

In the special case where the moduli $m_0(x), \ldots, m_{n-1}(x)$ all have the same degree, the two triples $(N, K, \mathrm{d_{minD}})$ and $(n, k, \mathrm{d_{minH}})$ coincide up to a scale factor.

Let C be a code as in Definition 1 that satisfies (6). The receiver sees $y = c + e$, where $c \in C$ is the transmitted codeword and e is an error pattern. A *minimum Hamming distance decoder* is a decoder that produces

$$\hat{c} = \operatorname*{argmin}_{c \in C} \mathrm{d_H}(c, y). \quad (14)$$

A *minimum degree-weighted distance decoder* is a decoder that produces

$$\hat{c} = \operatorname*{argmin}_{c \in C} \mathrm{d_D}(c, y). \quad (15)$$

Theorem 4 (Basic Error Correction Bounds). If

$$\mathrm{w_H}(e) \le \mathrm{t_H} \triangleq \left\lfloor \frac{n-k}{2} \right\rfloor, \quad (16)$$

then the rule (14) produces $\hat{c} = c$. If

$$\mathrm{w_D}(e) \le \mathrm{t_D} \triangleq \left\lfloor \frac{N-K}{2} \right\rfloor, \quad (17)$$

then the rule (15) produces $\hat{c} = c$. □

In general, the decoding rules (14) and (15) produce different estimates $\hat{c}$ [9].

For erasures-only decoding, we have

Theorem 5 (Erasures Correction Bound). Let C be a code as in Definition 1. For $e = (e_0, \ldots, e_{n-1})$, assume that the indices i where $e_i \ne 0$ are known. If

$$\mathrm{w_D}(e) \le N - K, \quad (18)$$

then the message polynomial $a(x) \in R_{M_k}$ can be reconstructed from $y = \psi(a) + e$. □

IV. Error Locator Polynomial and Erasures-Only Decoding

Decoding Reed-Solomon codes can be reduced to solving a key equation that involves an error locator polynomial [11]. We now propose such an approach for polynomial remainder codes.

Let C be a code as in Definition 1 satisfying (6). The receiver sees $y = c + e$, where $c \in C$ is the transmitted codeword and e is an error pattern. Let $Y(x) = a(x) + E(x)$ denote the pre-image $\psi^{-1}(y)$ of y, where $a(x) = \psi^{-1}(c)$ is the transmitted message polynomial and where $E(x)$ denotes the pre-image $\psi^{-1}(e)$ of the error e.

Definition 3. $\Lambda(x) \in F[x]$ is an *error locator polynomial* if

$$\Lambda(x) \bmod m_\ell(x) = 0 \text{ if and only if } e_\ell \neq 0 \tag{19}$$

for $0 \leq \ell \leq n-1$. □

Clearly, the polynomial

$$\Lambda_e(x) \triangleq \prod_{\ell: e_\ell \neq 0} m_\ell(x) \tag{20}$$

of $\deg \Lambda_e(x) = \mathrm{w_D}(e)$ is the unique monic error locator polynomial of the smallest degree.

Recall that $M_n(x) \triangleq \prod_{i=0}^{n-1} m_i(x)$.

Theorem 6 (Key Equation). The error locator polynomial (20) satisfies

$$A(x)M_n(x) = \Lambda_e(x)E(x) \tag{21}$$

for some polynomial $A(x) \in F[x]$ of degree smaller than $\deg \Lambda_e(x)$. Conversely, if some polynomial $G(x) \in F[x]$ satisfies

$$A(x)M_n(x) = G(x)E(x) \tag{22}$$

for some $A(x) \in F[x]$, then $G(x)$ is a multiple of $\Lambda_e(x)$. □

Theorem 7 (Error Locator-based Interpolation). If $G(x)$ is a multiple of $\Lambda_e(x)$ with

$$\deg G(x) \leq N - K, \tag{23}$$

then

$$Y(x)G(x) \bmod M_n(x) = a(x)G(x) \tag{24}$$

□

Note that (24) amounts to a closed formula for computing $a(x)$ from $Y(x)$ and $G(x)$ by dividing the left-hand side of (24) by $G(x)$. In contrast to most other statements in this paper, Theorem 7 appears to be new even when specialized to Reed-Solomon codes (where we usually have $M_n(x) = x^n - 1$).

Let $N_{\text{zero}}(G)$ denote the number of indices $i \in \{0, \ldots, n-1\}$ such that $G(x) \bmod m_i(x) = 0$. Note that $N_{\text{zero}}(\Lambda_e) = \mathrm{w_H}(e)$.

Recall the definition of $\mathrm{t_H}$ from (16).

Theorem 8 (Error Locator Test). Let $y = \psi(a) + e$ as above. For some polynomial $G(x)$ and

$$Z(x) \triangleq Y(x)G(x) \bmod M_n(x), \tag{25}$$

assume that the following conditions are satisfied:

1) $\mathrm{w_H}(e) \leq \mathrm{t_H}$
2) $N_{\text{zero}}(G) \leq \mathrm{t_H}$ and $\deg G(x) \leq \sum_{\ell=n-\mathrm{t_H}}^{n-1} \deg m_\ell(x)$.
3) $G(x)$ divides $Z(x)$
4) $\deg Z(x) - \deg G(x) < K$.

Then $G(x)$ is a multiple of $\Lambda_e(x)$ and $Z(x) = a(x)G(x)$. □

Note that the conditions in the theorem are satisfied for $G(x) = \Lambda_e(x)$.

V. Computing the Error Locator Polynomial by an Extended GCD Algorithm

Let $\gcd(a, b)$ denote the greatest common divisor (gcd) of $a, b \in R = F[x]$, not both zero.

For Reed-Solomon codes, the use of an extended gcd algorithm to compute an error locator polynomial is standard [10], [11]. We now adapt this approach to solve our key equation (22). We prefer the following gcd algorithm (but Euclid's algorithm could also be adapted to our purpose).

A. An Extended GCD Algorithm

In this subsection, we assume that $E(x)$ is fully known; in the next subsection, we state the modifications that are required when $E(x)$ is only partially known.

Extended GCD Algorithm
Input: $M_n(x)$ and $E(x)$ with $\deg M_n(x) > \deg E(x)$.
Output: polynomials $\tilde{r}(x), s(x), t(x) \in F[x]$ where $\tilde{r}(x) = \gamma \gcd(M_n(x), E(x))$ for some $\gamma \in F$ and where $s(x)$ and $t(x)$ satisfy $s(x) \cdot M_n(x) + t(x) \cdot E(x) = 0$.

$r(x) := M_n(x)$
$\tilde{r}(x) := E(x)$
$s(x) := 1$
$t(x) := 0$
$\tilde{s}(x) := 0$
$\tilde{t}(x) := 1$
loop begin
  if $\deg r(x) < \deg \tilde{r}(x)$ **begin**
    $(r(x), \tilde{r}(x)) := (\tilde{r}(x), r(x))$
    $(s(x), \tilde{s}(x)) := (\tilde{s}(x), s(x))$
    $(t(x), \tilde{t}(x)) := (\tilde{t}(x), t(x))$
  end
  $i := \deg r(x)$
  $j := \deg \tilde{r}(x)$
  while $i \geq j$ **begin**
    $q(x) := \frac{r_i}{\tilde{r}_j}\, x^{i-j}$
    $r(x) := r(x) - q(x) \cdot \tilde{r}(x)$
    $s(x) := s(x) - q(x) \cdot \tilde{s}(x)$
    $t(x) := t(x) - q(x) \cdot \tilde{t}(x)$
    $i := \deg r(x)$
  end
  if $r(x) = 0$ **begin**
    return $\tilde{r}(x)$, $s(x)$, $t(x)$
  end
end

□

In this algorithm, $r_i \in F$ denotes the coefficient of x^i in $r(x)$ and $\tilde{r}_j \in F$ denotes the coefficient of x^j in $\tilde{r}(x)$. For polynomials over $F = GF(2)$, the scalar division in line 16 disappears.

The standard loop invariant [11] holds also for this gcd algorithm:

Theorem 9 (GCD Loop Invariant). The condition

$$r(x) = s(x) \cdot M_n(x) + t(x) \cdot E(x) \tag{26}$$

holds throughout the algorithm (as stated above) and the condition

$$\deg M_n(x) = \deg \tilde{r}(x) + \deg t(x) \tag{27}$$

holds between lines 21 and 22. □

The algorithm terminates when $r(x) = 0$ and returns $\tilde{r}(x), s(x)$, and $t(x)$. Since $M_n(x)$ consists of monic irreducible polynomials $m_0(x), \ldots, m_{n-1}(x)$, we then have

$$\begin{aligned} \tilde{r}(x) &= \gamma \ \gcd(M_n(x), E(x)) && (28)\\ &= \gamma \prod_{\ell: e_\ell = 0} m_\ell(x) && (29)\\ &= \gamma \ \frac{M_n(x)}{\Lambda_e(x)} && (30) \end{aligned}$$

(for some nonzero $\gamma \in F$) with $\deg \tilde{r}(x) = \deg M_n(x) - \deg \Lambda_e(x)$. It then follows from (27) that

$$\deg t(x) = \deg \Lambda_e(x). \tag{31}$$

With $r(x) = 0$, (26) becomes

$$s(x) \cdot M_n(x) + t(x) \cdot E(x) = 0. \tag{32}$$

We then conclude from the second part of Theorem 6 that $t(x)$ is a multiple of $\Lambda_e(x)$. Finally, we conclude from (31) that $t(x) = \tilde{\gamma}\Lambda_e(x)$ for some scalar $\tilde{\gamma} \in F$.

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

Recall that $Y(x) = a(x) + E(x)$ is the pre-image $\psi^{-1}(y)$ of the received message y where $E(x) = \sum_{\ell=0}^{N-1} E_\ell \, x^\ell$ is the pre-image of the error pattern e. Since $\deg a(x) < K$, the receiver knows the coefficients $E_K, E_{K+1}, \ldots, E_{N-1}$ of $E(x)$, but not $E_0, \ldots, E_{K-1}$. With the following modifications, the extended gcd algorithm as described above can still be used to compute the error locator polynomials $\Lambda_e(x)$.

Let

$$E_U(x) \triangleq \sum_{\ell=0}^{N-K-1} E_{K+\ell} \, x^\ell \tag{33}$$

be the known upper part of $E(x)$ and let

$$M_U(x) \triangleq \sum_{\ell=0}^{N-K} (M_n)_{K+\ell} \, x^\ell \tag{34}$$

be the corresponding upper part of $M_n(x) = \sum_{\ell=0}^{N} (M_n)_\ell \, x^\ell$.

Modified Extended GCD Algorithm
Input: $M_U(x)$ and $E_U(x)$ with $\deg M_U(x) > \deg E_U(x)$.
Output: $s(x)$ and $t(x)$, cf. Theorem 10 below.

The algorithm is the same as the extended gcd algorithm of Section V-A except for the following changes:

- Line 1: $r(x) := M_U(x)$.
- Line 2: $\tilde{r}(x) := E_U(x)$.
- Line 22: **if** $\deg r(x) < \deg t(x)$ **begin** □

Theorem 10. If $\mathrm{w_D}(e)$ (= $\deg \Lambda_e(x)$) satisfies

$$\deg \Lambda_e(x) \le (N - K)/2, \tag{35}$$

then the modified gcd algorithm of this section returns the same polynomials $s(x)$ and $t(x)$ (after the same number of iterations) as the gcd algorithm of Section V-A. □

We thus obtain $\Lambda_e(x) = t(x)/\tilde{\gamma}$ for some scalar $\tilde{\gamma} \in F$ as in Section V-A.

The computation of the polynomials $s(x)$ and $\tilde{s}(x)$ may actually be unnecessary (see Section VI). In consequence, lines 3, 5, 10, and 18 of the gcd algorithm may be deleted.

VI. Summary of Decoding Algorithm

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 ψ^{-1} and where $\deg a(x) < K$. The first step of our decoding algorithm is to compute $Y(x) = \psi^{-1}(y)$. If $\deg Y(x) < K$, we conclude $E(x) = 0$ and $a(x) = Y(x)$.

For erasures-only decoding (i.e., if the positions of the errors are known), we can directly compute the error locator polynomial $\Lambda_e(x)$ (20) and compute $a(x)$ from (24) with $G(x) = \Lambda_e(x)$. The only condition for this to work is $\deg \Lambda_e(x) \le N - K$.

Otherwise (i.e., for decoding errors in unknown positions), we form

$$E_U(x) = \sum_{\ell=0}^{N-K-1} Y_{K+\ell} \, x^\ell. \tag{36}$$

We then run the modified gcd algorithm of Section V-B, which yields the error locator polynomial $\Lambda_e(x)$ provided that $\mathrm{w_D}(e) \le (N - K)/2$. (If the polynomial $t(x)$ returned by the gcd algorithm has degree larger than $(N - K)/2$, we declare a decoding failure.)

From $\Lambda_e(x)$, we can compute $a(x)$ from (24) with $G(x) = \Lambda_e(x)$. Alternatively, we can compute $E(x)$ from (32) and obtain $a(x) = Y(x) - E(x)$. In the special case of Reed-Solomon codes, both methods do not seem to be readily available in the literature and are perhaps new.

The described algorithm is guaranteed to correct all errors e with $\mathrm{w_D}(e) \le \mathrm{t_D}$ (17). If the code satisfies the Ordered-Degree Condition (6) as well as the additional condition

$$\deg m_k(x) = \cdots = \deg m_{n-1}(x), \tag{37}$$

then the algorithm is guaranteed to correct also all errors e with $\mathrm{w_H}(e) \le \mathrm{t_H}$ (16).

VII. An Extension

Assume that the code satisfies the Ordered-Degree Condition (6) but not the additional condition (37). In this case, we can still correct all errors e with $\mathrm{w_H}(e) \leq \mathrm{t_H}$ by the following procedure, which, however, is practical only in special cases.

Decoder with List of Special Error Positions
First, run the gcd decoder of the previous section. If it succeeds, stop. Otherwise, let S_Λ be a precomputed list of candidate error locator polynomials $G(x)$ with $N_{\text{zero}}(G) \leq \mathrm{t_H}$ and $\deg G(x) > (N-K)/2$. Check if any $G(x) \in S_\Lambda$ satisfies all conditions of Theorem 8. If such a polynomial $G(x)$ exists, we conclude that it is a multiple of the error locator polynomial and we compute $a(x)$ from (24). □

Such a decoder corrects all error patterns e with either $\mathrm{w_D}(e) \leq \mathrm{t_D}$ or $\mathrm{w_H}(e) \leq \mathrm{t_H}$.

VIII. Conclusion

We have revisited polynomial remainder codes explicitly allowing moduli of different degrees, i.e., variable symbol sizes within a codeword. In consequence, we have two different notions of distance—Hamming distance and degree-weighted distance—and the corresponding minimum-distance decoding rules. We have adapted gcd-based decoding for such codes, which is guaranteed to correct all error patterns of degree-weight less than half the minimum degree-weighted distance. (We also give an extension that allows to correct up to half the minimum Hamming distance, but this extension may not be practical.)

As second step of the decoding algorithm (or as main step in erasures-only decoding), we have proposed two different methods to recover the message from the error locator polynomial. These methods are nonstandard (and perhaps new) even when specialized to Reed-Solomon codes.

References

[1] I. S. Reed and G. Solomon, "Polynominal codes over certain finite fields," *J. SIAM*, vol. 8, pp. 300–304, Oct. 1962.

[2] J. J. Stone, "Multiple-burst error correction with the Chinese Remainder Theorem," *J. SIAM*, vol. 11, pp. 74–81, Mar. 1963.

[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] A. Shiozaki, "Decoding of redundant residue polynomial codes using Euclid's algorithm," *IEEE Trans. Information Theory*, vol. 34, pp. 1351–1354, Sep. 1988.

[7] O. Goldreich, D. Ron, and M. Sudan, "Chinese remaindering with errors," *IEEE Trans. Information Theory*, vol. 46, pp. 1330–1338, July 2000.

[8] V. Guruswami, A. Sahai, and M. Sudan, "Soft-decision decoding of Chinese remainder codes," *Proc. 41st IEEE Symp. Foundations Computer Science*, Redondo Beach, CA, 2000, pp. 159–168.

[9] J.-H. Yu and H.-A. Loeliger, "On polynomial remainder codes," *to be submitted to IEEE Trans. Information Theory*.

[10] Y. Sugiyama, M. Kasahara, S. Hirasawa, and T. Namekawa, "A method for solving key equation for decoding Goppa codes," *Information and Control*, vol. 27, pp. 87–99, 1975.

[11] R. M. Roth, *Introduction to Coding Theory.* New York: Cambridge University Press, 2006.