A Future Simplification of Procedure for Decoding Nonsystematic Reed-Solomon Codes Using the Berlekamp-Massey Algorithm

Tsung-Ching Lin, Trieu-Kien Truong, Hsin-Chiu Chang, and Hung-Peng Lee

Abstract—It is well-known that the Euclidean algorithm can be used to find the systematic errata-locator polynomial and the errata-evaluator polynomial simultaneously in Berlekamp’s key equation that is needed to decode a Reed-Solomon (RS) code. In this paper, a simplified decoding algorithm to correct both errors and erasures is used in conjunction with the Euclidean algorithm for efficiently decoding nonsystematic RS codes. In fact, this decoding algorithm is an appropriate modification to the algorithm developed by Shiozaki and Gao. Based on the ideas presented above, a fast algorithm described from Blahut’s classic book is derivated and proved in this paper to correct erasures as well as errors by replacing the Euclidean algorithm by the Berlekamp-Massey (BM) algorithm. These facts lead to significantly reduce the decoding complexity of the proposed RS decoder. In addition, computer simulations show that this simple and fast decoding technique reduces the decoding time when compared with existing efficient algorithms including the new Euclidean-algorithm-based decoding approach proposed in this paper.

Index Terms—Berlekamp-Massey algorithm, discrete Fourier transform, Euclidean algorithm, Reed-Solomon codes. 

I. INTRODUCTION

REED-SOLOMON codes for correcting both errors and erasures are used extensively in space communication links[1], compact-disc (CD) audio system[2], HDTV[3], digital versatile discs (DVD)[4], and the IEEE 802.16 standard[5]. By the use of Chinese remainder theorem together with the Euclidean algorithm, an efficient algorithm proposed earlier by Shiozaki[6] can be developed to correct errors and erasures of nonsystematic RS codes. One of the interesting aspects of this method is that the Chien search can be replaced by only a polynomial division once the error-erase-locator polynomial, called the errata-location polynomial are determined. However, Shiozaki’s algorithm has the disadvantage that uses Lagrange’s interpolation formula[7]. The complexity of the reconstruction of the corrupted message polynomial from the received symbols is too high due to use many arithmetic operations for computing the interpolating polynomial. So it entails highly computational complexity. This, in turn, severely lowers the decoding speed. More general, in 1990, the second author among others in [8] first proposed an improvement of Shiozaki’s algorithm[6] to decode RS over the ground fields denoted as $GF(F_n)$, where $F_n = 2^n + 1$ for $1 \leq n \leq 4$ are Fermat numbers, by the use of the fast Fermat number transform (FNT). This fast FNT eliminates polynomial multiplications needed in Lagrange’s interpolation formula and reduces substantially the number of multiplications and additions needed to reconstruct the corrupted message polynomial to $n \log_2 n[9],[10]$. Such a modified decoding scheme is much faster than that of Shiozaki’s algorithm. Gao[11] described later an algorithm for decoding RS codes over extended fields, i.e., $GF(2^m)$. Actually, the ideas presented in his algorithm are analogous to Shiozaki’s decoding algorithm. It was shown in detail [12] that Gao’s algorithm can also be derived from the Welch-Berlekamp algorithm [13],[14] and the Euclidean algorithm [15]–[17].

More recently, Lin et al. in [18] show that Gao’s algorithm is extended to correct erasures as well as errors by replacing the initial conditions of the Euclidean algorithm by the erasure-locator polynomial and errata interpolating polynomial. This modified decoding procedure similar to the Shiozaki-Truong-Cheung-Reed (STCR) decoding scheme [8] substantially reduced the complexity of interpolating polynomial computation, called the Lin-Chen-Truong (LCT) algorithm [18]. This transform decoding scheme utilizes an efficient Fourier transform over $GF(2^m)$ [19]–[21] to compute the corrupted information polynomial in a manner analogous to the syndrome computation in the conventional decoding scheme. In addition, computer simulations show in [18] that it reduces the decoding time when compared with the previous known algorithms. It is clear, however, that the Euclidean-algorithm-based decoding approach is easier to comprehend than the BM algorithm but is less efficient in practice. In addition, the disadvantage of this decoding algorithm is that a polynomial division needed to recapture the message polynomial is not quite practical and efficient for RS decoders.

It is well-known that a simplified procedure developed by Truong et al’s decoding algorithm given in [16] was proved to correct erasures as well as errors of systematic RS codes by replacing the initial condition of the Euclidean algorithm by the erasure-locator polynomial and the Forney syndrome polynomial. A simple decoding algorithm proposed in this paper is based on the fact that the codeword used in Euclid’s algorithm is a nonsystematic RS code so that Truong et al’s
algorithm can be modified to solve the key equation for the errata-locator polynomial. Then it uses the recursive extension to compute the remaining unknown syndromes. Finally, the message symbols are thus obtained by only subtracting all known syndromes from the coefficients of the corrupted information polynomial. In other words, a polynomial division used to evaluate the messaging polynomial in the LCT algorithm can be replaced by a recursive extension and a simple addition. The speed of the new Euclidean-algorithm-based decoding approach is shown to be slightly faster than that of the LCT algorithm. Actually, a further reduction in the number of arithmetic operations of the algorithm mentioned above can be achieved by using the BM algorithm [22]–[26] instead of Euclid’s algorithm. It can also be utilized to find the errata-locator polynomial from Berlekamp’s key equation provided that the message vector has the same format as the one given previously. In fact, the decoder depicted in the seconding block algorithm in Fig 9.2 of Reference 17 is of the general derivation from the frequency-domain point of view. The advantage of the proposed decoding algorithm is that the separate computation of the Forney syndrome polynomial and the errata-evaluation polynomial usually needed in the RS decoder using Euclid’s algorithm is completely avoided. Simulation results show that this fast decoding method may be adapted to the Chase algorithm [27], a soft-decision decoding, for RS codes.

The structure of this paper is as follows: In Section II, a simple decoder using the Euclid’s algorithm is provided for correcting both errors and erasures of nonsystematic RS codes. In Section III, an efficient and simple RS decoder using the BM algorithm is developed in detail. It is based on the ideas of the decoding algorithm using Euclid’s algorithm given in Section II. Simulation results are presented in Section IV. Finally, conclusions are given in the last section of the paper.

II. Decoding Algorithm

A. Review Stage

A simply algorithm for decoding systematic RS codes suggested by the authors in [16] uses Euclid’s algorithm to correct errors as well as erasures. In this section, the above-mentioned algorithm is modified for correcting both errors and erasures of nonsystematic RS codes.

Let $C$ be a $(n,k)$ RS code over $GF(2^m)$ with minimum distance $d$, where $n=2^m-1$ is the block length, $k$ is the number of $m$-bit message symbols and $d-1=n-k$ is the number of parity symbols. The maximum number of errors in an RS code which can be corrected is $t = \lfloor (d-1)/2 \rfloor$, where $\lfloor x \rfloor$ denotes the largest integer less than or equal to $x$.

The message polynomial of the RS code is now defined as

$$M(x) = \sum_{i=0}^{k-1} m_i x^i.$$  

Written as a vector, the codeword is

$$c = (c_0, c_1, \ldots, c_{n-1}),$$

where the component $c_j$ of $c$ is computed by $c_j = M(\alpha^{-j})$ for $0 \leq j \leq n-1$ and $\alpha^{-1}$ is an inverse primitive element in $GF(2^m)$.

The decoding as given in (2) originally discovered by Reed and Solomon in [28] is, what is now called, a nonsystematic code. Evidently, the codeword is given by $c$ in (2) so that the message symbols do not appear as codeword symbols. To illustrate this, the first symbol and the last consecutive $k-1$ symbols are assigned to be information symbols. The remaining $(n-k)$ symbols are assumed to be zero. That is,

$$m = (m_0, m_1, \ldots, m_{d-1}, m_d, \ldots, m_{n-1}),$$

where $m$ denotes the message vector such that $m_1 = m_2 = \ldots = m_{d-1} = 0$. In fact, the code vector can be computed next by the inverse Fourier transform (FT) of the message vector over $GF(2^m)$ as follows:

$$c_j = \frac{1}{n} \sum_{i=0}^{n-1} m_i \cdot \alpha^{-ij} = \sum_{i=0}^{n-1} m_i \cdot \alpha^{-ij} = M(\alpha^{-j})$$

where $n$ is restricted to be an odd number.

In general, the forward FT of the received vector is an usual method to obtain the syndromes of the code for the conventional RS decoder. Due to this reason, the inverse rather forward FT of the message vector needs to be used for generating the code vector given in (4). Suppose a RS codeword is transmitted through a noisy channel. Let $e = (e_0, e_1, \ldots, e_{n-1})$ and $\mu = (\mu_0, \mu_1, \ldots, \mu_{n-1})$ be, respectively, two error and erasure vectors. Next, the errata vector is defined as $\tilde{\mu} = \mu + e$. Error represents the value and the position of the received vector is unknown and erasure is a type of error with the known position information. Errata means either the error or erasure. Thus, the received vector has the form $r = c + \tilde{\mu} = (r_0, r_1, \ldots, r_{n-1})$. Finally, let there be $v$ error symbols and $s$ erasure symbols occurred in the received vector $r$ of $n$ symbols such that $2v + s \leq d-1$, where $v \leq t$. In the present of errors as well as erasures, the authors in [6, 8] described the key equation using the well-known Chinese remainder theorem. The Euclidean algorithm is then utilized to solve it for correcting both errors and erasures. In this section, a different way is suggested to derive a key equation not similar to the result of (11) in [18].

For a received vector $r = (r_0, r_1, \ldots, r_{n-1})$ over $GF(2^m)$, the forward FT of $r$ over $GF(2^m)$ is

$$a_i = \sum_{j=0}^{n-1} r_j \cdot \alpha^{ij} \quad \text{for } 0 \leq i \leq n-1,$$

where $a_i \in GF(2^m)$. Actually, these transform values are the coefficients of the interpolating polynomial defined in the LCT algorithm, see Eq. (1) of [18]; that is, $T(x) = \sum_{i=0}^{n-1} a_i x^i$. The substitution of $r = c + \tilde{\mu}$ into (5) yields

$$a_i = \sum_{j=0}^{n-1} r_j \cdot \alpha^{ij} = \sum_{j=0}^{n-1} (c_j + \tilde{\mu}_j) \cdot \alpha^{ij} = m_i + s_i \quad \text{for } 0 \leq i \leq n-1,$$
where \( m_i \) are the components of the message vector and \( S_i \) for \( 0 \leq i \leq n - 1 \) are referred to as the syndromes of the code given by the following set of equations:

\[
S_i = \sum_{j=0}^{n-1} \mu_j \cdot \alpha^{i-j} = \sum_{i=1}^{v+s} Y_j \cdot X_i^j \quad \text{for} \quad 0 \leq i \leq n - 1. \tag{7}
\]

In (7), \( X_j \) is either the \( i \)-th erasure or error location, and \( Y_j \) is either the \( i \)-th erasure or error magnitude. Since \( m_i = 0 \) for \( 1 \leq i \leq d - 1 \), \( S_i = a_i \) for \( 1 \leq i \leq d - 1 \) are the known components of \( S \), called the primary known syndromes which are \( d - 1 \) of the \( n \) components of the spectrum of the errata pattern. It is not difficult to show [22] that

\[
S(x) = \sum_{k=1}^{d-1} S_k \cdot x^{k-1} = \sum_{j=1}^{v+s} Y_j x_j \cdot \frac{x^{v+s}}{1 - X_j x} - \sum_{j=1}^{v+s} Y_j x_j^{d-1} x^{d-1} - \frac{x^{v+s}}{1 - X_j x}. \tag{8}
\]

Now define four different polynomials as follows: The erasure-locator polynomial:

\[
\Lambda(x) = \prod_{j=1}^{s} (1 - Z_j x) = \sum_{j=0}^{s} \Lambda_j x^j, \tag{9a}
\]

where \( \Lambda_0 = 1 \) and \( Z_j \) is the locator of the erasure position in the erasure vector \( \mu \).

The error-locator polynomial:

\[
\lambda(x) = \prod_{j=1}^{v} (1 - X_j x) = \sum_{j=0}^{v} \lambda_j x^j, \tag{9b}
\]

where \( \lambda_0 = 1 \) and \( X_j \) is the locator of the error position in the error vector \( e \).

The errata-locator polynomial:

\[
\tau(x) = \Lambda(x) \cdot \lambda(x) = \prod_{j=1}^{v+s} (1 - X_j x) = \sum_{j=0}^{v+s} \tau_j x^j, \tag{9c}
\]

where \( \tau_0 = 1 \).

The errata-evaluation polynomial:

\[
A(x) = \sum_{j=1}^{v+s} (Y_j X_j \cdot \prod_{i=1, i \neq j}^{v+s} (1 - X_i x)), \tag{9d}
\]

where \( Y_j \) and \( X_j \) are defined in (7). In terms of the polynomial defined above, (8) can be shown to become the congruence relation given by

\[
S(x) \equiv \frac{A(x)}{\lambda(x) \Lambda(x)} \mod x^{d-1}. \tag{10}
\]

Define a generalization of the Forney syndrome polynomial as

\[
F(x) \equiv S(x) \Lambda(x) \mod x^{d-1} = f_0 + f_1 x + f_2 x^2 + \cdots + f_{d-2} x^{d-2} \tag{11}
\]

where \( F(x) \in F = \{ \sum_{i=0}^{d-2} f_i x^i \mid f_i \in GF(2^m) \} \), the set of formal power series.

By (11), the key equation in (10) for \( \lambda(x) \) and \( A(x) \) is

\[
A(x) \equiv F(x) \lambda(x) \mod x^{d-1}, \tag{12a}
\]

where

\[
\deg \{ \lambda(x) \} \leq \left\lfloor (d-1-s/2) \right\rfloor \quad \text{and} \quad \deg \{ A(x) \} \leq \left\lfloor (d+s-3)/2 \right\rfloor \tag{12b}
\]

The following evident theorem, see Theorem 2 in [16], was developed to find an unique pair of the polynomials \( A(x) \) and \( \tau(x) \) simultaneously from the known \( F(x) \) defined previously and the key equation (12a).

**Theorem 1:** Let \( F(x) \) in (11) be the Forney syndrome polynomial of a \( v \)-error and \( s \)-erasure correcting RS codes under the condition \( s + v \leq d - 1 \), where \( d - 1 \) is either an error or an odd integer. Consider the two polynomials \( x^{d-1} \) and \( F(x) \). The Euclidean algorithm for polynomials on \( GF(2^m) \) can be used to develop two finite sequences \( R_s(x) \) and \( \tau_s(x) \) from the following two recursive formulas:

\[
\tau_s(x) = (-Q_{s-1}(x)) \tau_{s-1}(x) + \tau_{s-2}(x) \tag{13a}
\]

and

\[
R_s(x) = R_{s-2}(x) - Q_{s-1}(x) R_{s-1}(x) \tag{13b}
\]

for \( s = 1, 2, \ldots, \), where the initial conditions are \( \tau_0(x) = \Lambda(x) \), \( \tau_1(x) = 0 \), \( R_0(x) = F(x) \). Hence, \( Q_{s-1}(x) \) is obtained as the principal part of \( R_{s-2}(x)/R_{s-1}(x) \). The recursion in (13a) and (13b) for \( R_s(x) \) and \( \tau_s(x) \) terminates when \( \deg \{ R_s(x) \} \leq \left\lfloor (i + s - 3)/2 \right\rfloor \) for the first time for some value \( s = s' \). Finally, one yields

\[
A(x) = R_{s'}(x)/\Delta \tag{14a}
\]

and

\[
\tau(x) = \tau_{s'}(x)/\Delta, \tag{14b}
\]

where \( \Delta = s'(0) \) is a field element in \( GF(2^m) \) which is chosen so that \( \tau_0 = 1 \).

Once the coefficients of the errata-locator polynomial, i.e., \( \tau_i \) for \( 0 \leq i \leq v + s \) are determined, the principal problem is to find the remaining \( k \) unknown syndromes, namely \( S_i \) for \( d \leq i \leq n \) from the following well-known recursive formulas:

\[
S_i = \tau_1 S_{i-1} + \tau_2 S_{i-2} + \cdots + \tau_v S_{i-(v+s)} \tag{15}
\]

where \( S_i \) for \( 1 \leq i \leq d - 1 \) are known and \( S_0 = S_n \).

After determining the remaining components of \( S \), one has enough syndromes to apply (6) from which the message pattern is recovered. The overall decoding of nonsystematic RS codes for correcting errors and erasures using Theorem 1 and the Euclidean algorithm is summarized as follows:

**Algorithm 1**

1. Using the fast Fourier transform over \( GF(2^m) \), compute the transform over \( GF(2^m) \) of the received codeword from (5). This yields \( a_i \) for \( 0 \leq i \leq n - 1 \) which contains \( d - 1 \) consecutive primary known syndromes in them. That is, \( S_i = a_i \) for \( 1 \leq i \leq d - 1 \). Next calculate the erasure-locator polynomial \( \Lambda(x) \) from (9a) and define \( \deg \{ \Lambda(x) \} = s \).

2. Compute the Forney syndrome polynomial \( F(x) \) from (11).

3. The extended Euclidean algorithm applied next to the known polynomials \( x^{d-1} \) and \( F(x) \) is utilized to solve (12) for the polynomials \( A(x) \) and \( \tau(x) \). The initial values of the Euclidean algorithm are \( \tau_0(x) = \Lambda(x) \), \( \tau_{1}(x) = 0 \), \( R_{-1}(x) = x^{d-1} \), and \( R_0(x) = F(x) \). For \( s = d - 1 \), set \( \tau(x) = \Lambda(x) \) and \( A(x) = F(x) \).
(4) After determining the errata-locator polynomial $\tau(x)$ for $0 \leq i \leq s+v$, calculate the remaining unknown syndromes $S_i$ from the recursive formula as given in (15). Thus, with the aid of primary known syndromes, the set of all known syndromes denoted as $S = \{S_0, S_1, S_2, \ldots, S_{d-1}\}$ is obtained. Finally, the corrected message pattern is found by subtracting the syndrome pattern from the unknown $a_i$’s for $0 \leq i \leq n-1$; that is, $m_i = a_i - S_i$ for $0 \leq i \leq n-1$.

The advantage of the above-mentioned algorithm is that once the classic errata-locator polynomial is known, the Chien search or the inverse finite field transform needed in the conventional time-domain decoder for systematical RS codes is completely avoided.

To illustrate the decoding procedure mentioned above for correcting errors and erasures, an elementary example of an RS code over $GF(2^3)$ is now presented.

**Example 1:** Consider the $(7,4)$ RS code over $GF(2^3)$ with minimum distance $d = 4$. In this code, $s$ erasures and $v$ errors under the condition $2v + s \leq 3$ can be corrected. Assume the message vector is $m = (m_0, m_1, m_2, m_3, m_4, m_5, m_6) = (\alpha^4, 0, 0, 0, \alpha, \alpha^2, \alpha^5)$, where $m_1, m_2,$ and $m_3$ are set to be zero and $\alpha$ is a root of a 3rd degree primitive irreducible polynomial $p(x) = x^3 + x + 1$ over $GF(2)$. From (4), we conclude that the codeword vector is $c = (c_0, c_1, c_2, c_3, c_4, c_5, c_6)$, namely $\alpha^4, \alpha^3, \alpha^3, \alpha^4, \alpha, \alpha^2, \alpha^5$.

Furthermore, assume the erasure vector and the erasure error vector are $\mu = (0, 0, 0, 0, 0, \alpha^3, 0)$ and $e = (0, \alpha, 0, 0, 0, 0, 0)$, respectively. Thus, the errata vector is $\epsilon = \mu + e = (0, 0, 0, 0, \alpha, \alpha^2, \alpha^2)$. Hence, the received vector has the form $r = c + \epsilon = (r_0, r_1, r_2, r_3, r_4, r_5, r_6) = (\alpha^5, 1, \alpha^3, \alpha^6, \alpha^2, \alpha^6, \alpha^4)$.

To decode this code, applying (6), the coefficients of $T(x)$, i.e., $a_i$ for $0 \leq i \leq n-1$ can be computed by the fast transform over $GF(2^3)$, so that $a = (\alpha^5, \alpha^4, \alpha^4, 0, 1, 0, 1)$ is obtained. Hence, the primary known syndromes are $S_i = a_i$ for $0 \leq i \leq d-1$; that is, $S_0 = a_0 = \alpha^4$, $S_1 = a_2 = \alpha^4$, and $S_3 = a_3 = 0$.

With $A(x) = (1 + \alpha x^2)$ and $S(x) = \alpha^4 x^4 + \alpha x + \alpha^4$, the extended Euclidean algorithm applied next to polynomial $x^d - 1$ and $F(x)$ is used to solve Eq. (12) for the polynomials $A(x)$ and $\tau(x)$. This is accomplished by the recursive formulas (13a) and (13b) illustrated in Table 1, where initially $R_{-1}(x) = x^d - 1 = x^3$ and $R_0(x) = F(x) = \alpha^2 x^2 + \alpha x + \alpha^4$. From Table 1, one observes that $\deg\{R_{-1}(x)\} = \deg\{R_1(x)\} = 1 \leq \lfloor (d + s - 3)/2 \rfloor = 1$. Thus, the computation terminates at this point for $s' = 1$, thereby yielding $R_1(x) = \alpha^3 x^2 + \alpha$ and $\tau(x) = \alpha^3 x^2 + \alpha^3 x + \alpha^4$. According to (14a) and (14b), one has $\tau(x) = \tau(x)/\Delta = \alpha^2 x^2 + \alpha x + 1$ and $A(x) = R_1(x)/\Delta = \alpha^6 x^3 + \alpha^4$, where $\Delta = \tau(x)/\alpha^4$. Now, calculate the remaining four unknown syndromes from (15); that is, $S_1 = \alpha^6 S_{i-1} + \alpha^6 S_{i-2}$ for all $i$, where $S_1$, $S_2$, and $S_3$ are the primary known syndromes. The desired results are $S_4 = \alpha^6, S_5 = \alpha^2, S_6 = \alpha^4,$ and $S_7 = S_1 = 1$. Finally, computing the message vector from (6) yields $m = (\alpha^4, 0, 0, 0, \alpha, \alpha^2, \alpha^5)$.

The principle benefit of Euclid's approach to the decoding of RS codes is the case with which it can be understood and applied but has somewhat less efficient in practice [17, 7.7]. Moreover, the performance of this decoding algorithm can be further improved by the use of the fast Euclidean algorithm developed in [29]-[31].

### III. Decoding Algorithm Using the BM Algorithm

As seen in [22, [23], [32], the development of the BM algorithm is somewhat tedious, but it is an efficient algorithm for determining the errata-location polynomial. Towards this end, using the BM algorithm, a frequency-domain decoder without derivation (see, for example, the third scheme of Fig. 9.2 in [17]) was suggested to correct all errors and erasures within the error-correcting capacity of non-systematic RS codes. In this section, the method mentioned above is generated and derived mathematically. In other words, using a procedure similar to that used to derive the key equation in (12), one can also derive Berlekamp's key equation for solving the errata-locator polynomial, provided that the message vector is the same as given in Algorithm 1, see (3). Again, the code vector $c$ can be computed via Eq. (4). For a given the received vector, the coefficients of $T(x)$, namely $a_i$ for $0 \leq i \leq n-1$, are computed from (6). Since $m_i = 0$ for $0 \leq i \leq d-1$, from (6), one obtains the primary known syndromes, $S_i = a_i$ for $0 \leq i \leq d-1$.

The first step in deriving the key equation in the BM-algorithm-based decoding approach is to chose $f_k$ for $0 \leq k \leq d - 1 - s$ of Eq. (11) such that these syndromes, called the Fornay syndromes, can be expressed in terms of the coefficients of $A(x)$ and the syndromes given by

$$f_k = \sum_{j=0}^{s} A_j S_{k+s-j} \quad \text{for } 1 \leq k \leq d-1-s \quad (16)$$

which are all known. By substituting $S_i$ as in (7) for $1 \leq i \leq d-1$ into (16), it can be shown that

$$f_k = \sum_{i=1}^{v} D_i X_i^k \quad \text{for } 1 \leq k \leq d-1-s \quad (17)$$

Here the quantities $D_i = Y_i \cdot \sum_{j=0}^{s} A_j X_i^{s-j}$. Now, let $F'(x)$ be the modified Fornay syndrome polynomial defined by

$$F'(x) = \sum_{k=1}^{d-s-1} f_k x^k \quad (18)$$

where $F'(x) \in F' = \{\sum_{i=1}^{d-s} f_j x^j | f_i \in GF(2^m)\}$ and $F \subset F$.

Obviously, the terms of $F'(x)$ are the second consecutive $d - 1 - s$ terms of $F(x)$ given in (11). A substitution of (17) into (18) yields the following Berlekamp’s key equation:

$$(1 + F'(x)) \cdot \lambda(x) \equiv B(x) \mod x^{d-s}, \quad (19)$$

where $B(x) = \lambda(x) + F(x)$ and

$$P(x) = \sum_{j=1}^{v} (D_j X_j x \cdot \prod_{l=1,l\neq j}^{v} (1-X_l x)) \quad \text{with } \deg\{P(x)\} = v.$$
TABLE I
EXAMPLE OF THE EUCLIDEAN ALGORITHM USED TO FIND $\tau(X)$

<table>
<thead>
<tr>
<th>$s$</th>
<th>$R_{x-2}(x) = Q_{x-1}(x)R_{x-1}(x) + R_x(x)$</th>
<th>$Q_{x-1}(x)$</th>
<th>$R_x(x)$</th>
<th>$\tau_x$</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>$x^3 = (α^5x + α^4)(α^2x^2 + αx + α^4) + α^3x + α$</td>
<td>$α^2x^2 + αx + α^4$</td>
<td>$α^3x + α$</td>
<td>$α^5x + 1$</td>
</tr>
<tr>
<td>0</td>
<td>$x^3 = (α^5x + α^4)(α^2x^2 + αx + α^4)$</td>
<td>$α^2x^2 + αx + α^4$</td>
<td>$α^3x + α$</td>
<td>$α^5x + 1$</td>
</tr>
</tbody>
</table>

TABLE II
EXAMPLE OF THE MODIFIED BM ALGORITHM USED TO FIND $\tau(X)$

<table>
<thead>
<tr>
<th>$k$</th>
<th>$\delta(k) = \sum_{j=0}^{d-1} \mu_j^{(k-1)} \cdot S_{k+j-s-j}$</th>
<th>$\mu^{(k)}(x) = \gamma^{(k-1)}(x)/\delta(k)$</th>
<th>$\lambda^{(k)}(x)$</th>
<th>$\gamma^{(k)}$</th>
<th>$l^{(k)}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$\mu^{(0)}(x) = \Lambda(x) = 1 + α^5x$</td>
<td>$\lambda^{(0)}(x) = \Lambda(x) = 1 + α^5x$</td>
<td>$\gamma^{(0)} = 1$</td>
<td>$l^{(0)} = 0$</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>$\delta(1) = \sum_{j=0}^{d-1} \mu_j^{(0)} \cdot S_{k+j-s-j}$</td>
<td>$\mu^{(1)}(x) = \gamma^{(0)}(x) / \delta(1)$</td>
<td>$\lambda^{(1)}(x) = \mu^{(0)}(x)$</td>
<td>$\gamma^{(1)} = \delta(1) = α$</td>
<td>$l^{(1)} = 1 - l^{(0)} = 1$</td>
</tr>
<tr>
<td>2</td>
<td>$\delta(2) = \sum_{j=0}^{d-1} \mu_j^{(1)} \cdot S_{k+j-s-j}$</td>
<td>$\mu^{(2)}(x) = \gamma^{(1)}(x) / \delta(2)$</td>
<td>$\lambda^{(2)}(x) = x \cdot \lambda^{(1)}(x)$</td>
<td>$\gamma^{(2)} = \gamma^{(1)} = 1$</td>
<td>$l^{(2)} = 1$</td>
</tr>
</tbody>
</table>

syndromes. This BM algorithm is described in the following three steps:

1. Initially define $\mu^{(0)}(x) = 1$, $\lambda^{(0)}(x) = 1$, $l^{(0)} = 0$, $k = 0$, and $\gamma^{(k)} = 0$ if $k \leq 0$.

2. Set $k = k + 1$. If $T_k$ is unknown, stop. Otherwise, first compute

$$\delta^{(k)} = \sum_{j=0}^{l^{(k-1)}} \mu_j^{(k-1)} \cdot T_{k-j},$$

where $l^{(k-1)} \leq d - 1$.

Then compute

$$\mu^{(k)}(x) = \gamma^{(k-1)} \mu^{(k-1)}(x) - \delta^{(k)} \lambda^{(k-1)}(x) \cdot x,$$

$$\lambda^{(k)}(x) = \begin{cases} \mu^{(k)}(x), & \text{if } \delta^{(k)} = 0 \text{ or } 2l^{(k-1)} > k - 1 \\ \lambda^{(k-1)}(x), & \text{if } \delta^{(k)} \neq 0 \text{ and } 2l^{(k-1)} \leq k - 1 
\end{cases}$$

$$l^{(k)} = \begin{cases} l^{(k-1)}, & \text{if } \delta(k) = 0 \text{ or } 2l^{(k-1)} > k - 1 \\ k - l^{(k-1)}, & \text{if } \delta(k) \neq 0 \text{ and } 2l^{(k-1)} \leq k - 1 
\end{cases}$$

$$\gamma^{(k)} = \begin{cases} \gamma^{(k-1)}, & \text{if } \delta(k) = 0 \text{ or } 2l^{(k-1)} > k - 1 \\ \delta^{(k)}, & \text{if } \delta(k) \neq 0 \text{ and } 2l^{(k-1)} \leq k - 1 
\end{cases}$$

3. Return to step 2.

Based on the ideas of Blahut [17], if the Forney syndromes are replaced by the syndromes in (20) for $\delta^{(k)}$, and if the polynomials $\mu(x)$ and $\lambda(x)$ are initialized with the erasure-locator polynomial instead of one, then the errata-locator polynomial can be obtained directly by an inverse-free BM algorithm. So the modified BM algorithm initialized with the classic erasure-locator polynomial for generating the classic errata-locator polynomial is revised to have three steps as follows:

1. Initially, define $\mu^{(0)}(x) = \lambda^{(0)}(x) = \Lambda(x)$, $l^{(0)} = 0$, and $\gamma^{(k)} = 1$ if $k \leq 0$.

2. Set $k = k + 1$. If $k \geq d - 1 - s$, stop. Otherwise, compute

$$\delta^{(k)} = \sum_{j=0}^{d-1} \mu_j^{(k-1)} \cdot S_{k+j-s-j},$$

where $\mu_j^{(k-1)}$ is the coefficient of $\mu^{(k-1)}(x)$.

Then use (21)-(24) to compute the classic errata-locator polynomial in the inverse-free BM algorithm given by

$$\tau(x) = \lambda(x) \cdot \Lambda(x) = \frac{\mu^{(d-1-s)}(x)}{\Delta} = \sum_{j=0}^{v+s} \tau_j x^j,$$

where $\Delta = \mu^{d-1-s}(0)$ is a field element in $GF(2^m)$.

(3) Return to step 2.

Once the coefficients of the errata-locator polynomial, i.e., $\tau_i$ for $0 \leq i \leq v + s$ are determined, using a procedure similar to that used to find the remaining $k$ unknown syndromes, namely $S_i$ for $d \leq i \leq n$ in algorithm 1, one has enough syndromes to apply (6) from which the message pattern is recovered. Let us recapitulate the overall decoding of nonsystematic RS codes for both errors and erasures using
the inverse-free BM algorithm initialized with the erasure-locator polynomial and the syndromes. The new procedure is composed of the following three steps:

Algorithm 2

(1) Using step 1 in Algorithm 1 to obtain the primary known syndromes $S_i = a_i$ for $1 \leq i \leq d - 1$ and the erasure-locator polynomial $A(x)$.

(2) Use the inverse-free BM algorithm to determine the errata-locator polynomial from the erasure-locator polynomial $A(x)$ and the known $S_i$'s for $1 \leq i \leq d - 1$.

(3) With the coefficients of $\tau(x)$, i.e., $\tau_i$ for $0 \leq i \leq s + v$ known, use step 4 in Algorithm 1 to obtain the remaining unknown syndromes $S_i$ thereby yielding the corrected message pattern; that is, $m_i = a_i - S_i$ for $0 \leq i \leq n - 1$.

The advantage of Algorithm 2 over Algorithm 1 is that there is no need for extra computation of the Forney syndrome polynomial and the errata-evaluation polynomial $A(x)$ needed in Algorithm 1, provided that both of algorithms have the same information vector. Consequently, this decoding algorithm is simpler in computational complexity than that of Algorithm 1.

The decoding procedure using the modified BM algorithm is illustrated by the following example:

Example 2. Consider the $(7, 4)$ RS code over $GF(2^3)$ with minimum distance $d = 4$. In this code, $s$ erasures and $v$ errors under the condition $2v + s \leq 3$ can be corrected. In this case, the message vector and the codeword are the same as that given in Example 1; that is, $m = (\alpha^4, 0, 0, 0, \alpha, \alpha^2, \alpha^5)$ and $c = (\alpha^5, \alpha^3, \alpha^3, \alpha^0, \alpha^2, \alpha^4, \alpha^4)$. Furthermore, the erasure, error, and errata vectors are assumed to be identical to Example 1. Thus, the received vector has the form $r = c + \hat{m} = (\alpha^5, 1, \alpha^3, \alpha^6, \alpha^6, \alpha^6, \alpha^4)$ and the coefficients of $T(x)$, i.e., $a_i$ for $0 \leq i \leq n - 1$ are $a = (\alpha^5, \alpha^4, \alpha^4, 0, 1, 0, 1)$. Hence, the primary known syndromes are $S_1 = a_1 = \alpha^4$, $S_2 = a_2 = \alpha^4$, and $S_3 = a_3 = 0$, which are the same as given in Example 1.

The inverse-free BM algorithm is applied next to $S_k$ for $1 \leq k \leq 3$. In this way, the classic eratta-locator polynomial $\tau(x)$ is determined by a use of the inverse-free BM algorithm initialized with $\mu(0)(x) = \Delta(x) = (1 + \alpha^5 x)$. This is accomplished by the recursive formula as in (21) and is illustrated in Table II. From this table, one observes that the computation terminates at $k = d - 1 - s = 2$ and $\tau(x) = \mu(2)(x) / \Delta = 1 + \alpha^6 x + \alpha^6 x^2$ which is the same as
the one given in Example 1 is thus obtained, where $\Delta = \alpha$.

After knowing the errata-locator polynomial, using the same procedure that used for Example 1, one obtains the message vector $m = (\alpha^4, 0, 0, 0, \alpha, \alpha^2, \alpha^3)$.

IV. Simulation Results

Compared Algorithm 1 to Algorithm 2 for decoding errors and erasures of nonsystematic RS codes, both of which written in C++ language implemented on Intel Core 2 Quad Q6600 2.4GHz processor on windows XP operating system has been verified. Several hundred random codes with errors and erasures were created. The computational times listed in Table III were averaged over 100 computations. An examination of the decoding time in this table indicates that the proposed algorithm results in a substantial reduction of decoding complexity in terms of CPU time for correcting both errors and erasures of RS codes under the condition $s + 2v \leq d - 1$. However, the most time-consuming usage in this new algorithm is to calculate the finite field transform or the syndromes because it is calculated by the means of the well-known Horner’s rule [2]. In order to further speed up the proposed RS decoder, the fast algorithm developed in [19], instead of using Horner’s rule, is applied to evaluate the syndrome of the received word. Simulation results show that Algorithm 1 is far inferior to Algorithm 2 in terms of CPU time. The computational time of both Algorithm 1 and Algorithm 2 together with the fast syndrome evaluation method for nonsystematic RS decoders are given in Table IV. One observes from this table that the speed up rates for the $(7, 4)$, $(15, 9)$, $(31, 21)$, $(63, 47)$, $(127, 111)$, and $(255, 223)$ are approximately 4.06, 6.89, 5.34, 8.27, 9.7, and 12.32, respectively. Furthermore, the percentage of speed up rates attributable to steps 1, 2, 3 and 4 needed in Algorithm 1 and steps 1, 2, 3 needed in Algorithm 2 can also be derived from Table IV. In addition, the computational time of the LCT algorithm together with the Horner’s rule and the fast syndrome evaluation method is listed in Table V. Upon the inspect of Tables III, IV and V, Algorithm 2 is the most fast decoding technique when compared with existing efficient algorithms including Algorithm 1.

V. Conclusion

In this paper, the LCT decoding algorithm similar to the Shiozaki-Gao algorithm is modified to correct all patterns of errors as well as erasures for nonsystematic RS codes. It utilizes the Euclidean algorithm to find the errata-locator polynomial. As illustrated in Algorithm 1, a polynomial division used to evaluate the message polynomial in the LCT algorithm can be replaced by a recursive extension and a simple addition. This fact leads to reduce the decoding time in comparison with the existing LCT algorithm. It is shown that the decoding complexity of Algorithm 1 can be further reduced by using the BM algorithm. By this means, the BM algorithm instead of Euclid’s algorithm can also be utilized to solve the key equation for the errata-locator polynomial. The advantage of this proposed algorithm is that both the Forney syndrome polynomial and the errata-evaluation polynomial usually needed in Algorithm 1 are completely avoided. With this improved technique, simulation results show that this decoding algorithm significantly reduces in terms of CPU time from Algorithm 1.

ACKNOWLEDGMENT

The authors would like to thank the editor and reviewers for their useful comments and suggestions that improved the presentation of this paper.

REFERENCES

Tsung-Ching Lin (M’08) was born in Taiwan on April 6, 1970. He received the B.S. degree in mechanical engineering from Chung Yuan Christian Univ., Chung Li, Taiwan, in 1993, the M.S. degree in mechanical engineering from Nation Chiao Tung Univ., Hsinchu, Taiwan, in 1995, and the Ph.D. degree in electrical engineering from National Taiwan Univ. Taipei, Taiwan, in 2005. Currently, he is an Associate Professor of Information Engineering Department, I-Shou University, Kaohsiung, Taiwan. His research interests include image compression, digital information, data mining, control system, and analysis of high-tech industries.

Hsin-Chiu Chang was born in Taiwan in 1954. He received the B.S. degree in Applied Mathematics from National Chung Hsing University, Taichung, Taiwan, in 1976, received his M.S. degrees from National Sun Yat-sen University, Kaohsiung City, Taiwan, in 2005, and the Ph.D. degree in Information Engineering from the I-Shou University, Kaohsiung, Taiwan, in 2010. He is currently a post doctor researcher in the Department of Information Engineering, I-Shou University, Kaohsiung, Taiwan. His research interests include error control coding, statistics, probability, and information theory.

Hung-Peng Lee was born in Taiwan in 1958. He received the B.S. degree in Electronic Engineering from Feng Chia University, Taichung, Taiwan, in 1987, the M.S. degree in Electrical Engineering from the University of Memphis, Memphis, Tennessee, in 1989, and the Ph.D. degree in Information Engineering from the I-Shou University, Kaohsiung, Taiwan, in 2010. He is currently a lecturer in the Department of Computer Science and Information Engineering, Fortune Institute of Technology, Kaohsiung, Taiwan. His areas of research include error control coding, digital circuit design, and embedded system.

Trieu-Kien Truong (M’82-SM’83-F’99) was born in Vietnam on December 4, 1944. He received the B.S. degree from National Cheng Kung University, Tainan, Taiwan, in 1967, the M.S. degree from Washington University, St. Louis, MO, in 1971, and the Ph.D. degree from the University of Southern California, Los Angeles, in 1976, all in electrical engineering.

From 1975 to 1992, he was a Senior Member of Technical Staff (E6) with the Communication System Research Section, Jet Propulsion Laboratory, Pasadena, CA. Currently, he is a Chair Professor of College of Electrical and Information Engineering, I-Shou University, Kaohsiung, Taiwan. His research interests include error-correcting codes, VLSI architecture design, communication systems, signal processing, and image compression. Dr. Truong served as an Editor in the Asia area for the Journal of Visual Communication and Image Representation and as an Editor in the area of Coding Theory and Techniques for the IEEE TRANSACTIONS ON COMMUNICATIONS.