TDA Progress Report 42-99

November 15, 1989

Fast Transform Decoding of Nonsystematic Reed-Solomon Codes

T. K. Truong and K.-M. Cheung
Communications Systems Research Section

I. S. Reed
University of Southern California, Department of Electrical Engineering

A. Shiozaki
Osaka Electro-Communication University, Osaka, Japan

This article considers a Reed-Solomon (RS) code to be a special case of a redundant residue polynomial (RRP) code, and presents a fast transform decoding algorithm to correct both errors and erasures. This decoding scheme is an improvement of the decoding algorithm for the RRP code suggested by Shiozaki and Nishida [1], and can be realized readily on VLSI chips.

I. Introduction

Classes of redundant residue polynomial (RRP) codes were introduced first in [3,4]. These codes are constructed by use of the Chinese remainder theorem for polynomials over a finite field $GF(q)$. The codeword symbols of the RRP codes are expressed as polynomials over this field. The RRP codes can correct t error symbols with the aid of $2t$ redundant symbols.

Reed-Solomon (RS) codes constitute a subclass of RRP codes and are used in many sectors of today's industry. Some examples are the (255,223) 16-error-correcting RS code (NASA code) used in deep-space communications, the (31,15) 8-error-correcting RS code (JTIDS code) used in military communications, and the Cross Interleaving RS code (CIRC code) used in the compact-disc industry.

As Shiozaki [5] points out, by using the Chinese remainder theorem together with the Euclidean algorithm, an RRP code can be decoded without solving the error-locator polynomial and the error-evaluator polynomial. The decoder developed in [5] is a general frequency-domain implementation type depicted in the second block diagram in Fig. 9.2 of [2]. The advantage of the decoder in [5] over the decoder in [2] is that both the recursive extension and the inverse transform can be replaced by a single polynomial division. However, the method proposed by Shiozaki has the disadvantage that the reconstruction of the corrupted information polynomial $F'(x)$ from the received symbols involves n polynomial multiplications in $GF(q)$, followed by the operation modulo $M(x)$, where n is the codeword length and $M(x)$ is a product of n polynomials. These operations severely lower the decoding speed.

This article considers RS codes to be a special case of the RRP codes and proposes to decode RS codes by the use of both the Fermat number transform [6,7] and the Euclidean algorithm. The Fermat number transform (FNT) eliminates polynomial multiplications and reduces the number of multiplications needed to reconstruct $F'(x)$ to $n \log_2 n$. The fast transform decoding scheme proposed in this article is faster than the decoding algorithm in [5].

II. Some Preliminaries on Finite Fields and the Fast Fermat Number Transform

Given that $GF(q)$ is a finite field, let $GF(q)[x]$ be the ring of polynomials over $GF(q)$.

Definition 1. The two polynomials $m_1(x)$ and $m_2(x)$ over $GF(q)$ are said to be relatively prime if and only if the greatest common multiple of $m_1(x)$ and $m_2(x)$ is a constant in $GF(q)$.

Definition 2. The two polynomials $m_1(x)$ and $m_2(x)$ over $GF(q)$ are said to be congruent modulo $m(x)$, i.e., $m_1(x) \equiv m_2(x) \bmod m(x)$ if and only if $m(x)$ divides $m_1(x) - m_2(x)$.

The Chinese remainder theorem is presented here for convenience; proof can be found in [11]. Let $M(x) = \prod_{i=1}^{r} m_i(x)$ be a product of pairwise relatively prime polynomials. Let $A_1(x), A_2(x), \ldots, A_r(x)$ be any r polynomials such that $\deg[A_i(x)] \leq \deg[m_i(x)], i = 1, 2, \ldots, r$. Finally, let $t_i(x)$ satisfy

$$\frac{M(x)}{m_1 x} t_i(x) \equiv 1 \bmod m_i(x) \quad \text{for } (i = 1, 2, \ldots, r)$$

There then exists one and only one polynomial $f(x)$ of $GF(q)[x]$ of degree satisfying $\deg[f(x)] \leq \deg[M(x)]$, which uniquely solves the system of congruences:

$$f(x) \equiv A_i(x) \bmod m_i(x)$$

The polynomial $f(x)$ is given by

$$f(x) \equiv \sum_{i=1}^{r} \frac{M(x)}{m_i(x)} t_i(x) A_i(x) \bmod M(x) \qquad (1)$$

Let $GF(q)$ be a finite field, let n be a number that divides $q-1$, and let γ be a primitive nth root of unity. Define $(a_i)_{i=0}^{n-1}$ to be a sequence of n elements from $GF(q)$. A discrete Fourier transform of this sequence of length n is defined by

$$A_k \equiv \sum_{i=o}^{n-1} a_i \gamma^{ki} \bmod q \quad \text{for } (k = 0, 1, \ldots, n-1) \qquad \text{(2a)}$$

The inverse discrete Fourier transform of A_k is defined by

$$a_i \equiv n^{-1} \left[\sum_{k=0}^{n-1} A_k \gamma^{-ik} \right] \bmod q \quad \text{for } (i = 0, 1, \ldots, n-1) \qquad \text{(2b)}$$

A direct computation of the transform in Eq. (2a) or its inverse transform in Eq. (2b) requires $n(n-1)$ multiplications.

When q is a Fermat prime, the Fermat number transform (FNT) over $GF(q)$ can be used. A Fermat prime F_m is defined by

$$F_m = 2^{2^m} + 1 \quad \text{for } (m = 1, 2, 3, 4)$$

$$F_m = 2^{2^m} + 1 \quad \text{for } (m = 1, 2, 3, 4)$$

It is shown in [6,7] that integer 3 is a primitive $l = 2^{2^m}$th root of unity in $GF(F_m)$. Next, let n divide 2^{2^m}. Finally, suppose γ is a primitive nth root of unity in $GF(F_m)$ where

$$\gamma = 3^{l/n}$$

The purpose of an FNT of length n is to compute efficiently the transform sequence $(A_k)_{k=0}^{n-1}$ using Eq. (2a). On the other hand, the inverse Fermat number transform (IFNT) of length n reconstructs the sequence $(a_i)_{i=0}^{n-1}$ from the sequence $(A_k)_{k=0}^{n-1}$ via Eq. (2b). Since the order of γ is a power of 2 in $GF(F_m)$, the length of the sequence to be transformed is a power of 2. As a consequence, the very efficient FNT can then be used to yield a fast transform [6] which is analogous to the fast Fourier transform (FFT). In this case, the number of multiplications involved in evaluating such a transform sequence of length n is $n \log_2 n$ [8]. A new type of Fermat number multiplier is developed in [9]. More details about the FNT can be found in [6].

III. Nonsystematic RS Codes

First, a set of RRP codes is defined. As shown next, these codes are constructed using the Chinese remainder theorem for polynomials over a finite field $GF(q)$. Let $m_0(x), m_1(x), \ldots$, and $m_{n-1}(x)$ be n relatively prime polynomials, and

$$M(x) = \prod_{i=0}^{n-1} m_i(x)$$

Assume that the degree of each $m_i(x)$ is d, and that kd information symbols $\underline{u} = (u_0, u_1, \ldots, u_{kd-1})$ are represented by the information polynomial as

$$F(x) = u_0 + u_1 x + \cdots + u_{kd-1} x^{kd-1}$$

where $u_i \epsilon GF(q)$ and $k < n$. Then an RRP code is the residue representation of $F(x)$, that is,

$$\underline{\nu} = \left(A_0(x), A_1(x), \ldots, A_{n-1}(x)\right)$$

where $A_i(x) \equiv F(x) \bmod m_i(x)$ and $\deg[A_i(x)] < d$. By the Chinese remainder theorem, $F(x)$ can be recaptured from $A_i(x)$. The vector corresponding to the polynomial $A_i(x)$ is named the ith symbol. A code vector $\underline{\nu}$ can correct error symbols less than or equal to t symbols if $n - k \geq 2t$ [3,4].

The following shows that RS codes are a subclass of RRP codes. In order to facilitate the fast encoding and decoding procedures, which make use of the fast FNT methods as described in Section II, the codeword length n is required to be a power of 2.

Let $m_0(x), m_1(x), \ldots, m_{n-1}(x)$ be n relatively prime polynomials given by

$$m_i(x) = x - \gamma^i \quad \text{for } (i = 0, 1, \ldots, n-1)$$

Also let the k information symbols

$$(u_0, u_1, \ldots, u_{k-1}), u_i \epsilon GF(q)$$

be denoted by the information polynomial

$$F(x) = u_0 + u_1 x + \cdots + u_{k-1} x^{k-1}$$

Then the equations

$$F(1) \equiv F(x) \bmod m_0(x)$$

$$F(\gamma) \equiv F(x) \bmod m_1(x), \ldots,$$

and

$$F(\gamma^{n-1}) \equiv F(x) \bmod m_{n-1}(x)$$

lead to a code vector $\underline{\nu}$ represented by

$$\underline{\nu} = (A_0, A_1, \ldots, A_{n-1}) = \left(F(1), F(\gamma), \ldots, F(\gamma^{n-1})\right)$$

The code vector $\underline{\nu}$ is a nonsystematic RS codeword. It is not difficult to see that $\underline{\nu} = (A_0, A_1, \ldots, A_{n-1})$ is just the FNT of the sequence $(u_0, u_1, \ldots, u_{k-1}, 0, \ldots, 0)$ and the k information symbols $(u_0, u_1, \ldots, u_{k-1})$, i.e., $F(x)$ can be recaptured by an IFNT on the code vector $\underline{\nu} = (A_0, A_1, \ldots, A_{n-1})$.

On the other hand, since an RS code is a special case of an RRP code, the information polynomial $F(x)$ can be recaptured also from $\underline{\nu} = (A_0, A_1, \ldots, A_{n-1})$ by the use of the Chinese remainder theorem. Let $t_i(x)$ denote a polynomial that satisfies

$$\frac{M(x)}{m_i(x)} t_i(x) \equiv 1 \bmod m_i(x) \quad \text{for } (i = 0, 1, \ldots, n-1) \quad (3)$$

where

$$M(x) = \prod_{i=0}^{n-1} M_i(x) = \prod_{i=0}^{n-1} (x - \gamma^i)$$

Then the information polynomial $F(x)$ can be reconstructed as

$$F(x) \equiv \left[\sum_{i=0}^{n-1} \frac{M(x)}{m_{i(x)}} t_i(x) A_i\right] \quad (\text{mod } M(x))$$

IV. Decoding RS Codes

As Shiozaki et al. [1,5] point out, by using the Chinese remainder theorem together with the Euclidean algorithm, the RRP codes, which include the RS codes, can be decoded without solving the error-locator polynomial and the error-evaluator polynomial. However, that method has the disadvantage that the reconstruction of the corrupted information polynomial $F'(x)$ from the received symbols involves n polynomial multiplications in $GF(q)$ followed by the operation modulo $M(x)$. These operations can significantly lower the decoding speed. A modified decoding scheme, which makes use of the fast transform technique to bypass the tedious polynomial multiplications and modulo $M(x)$ operation, is given in the Appendix.

A. Decoding for Correcting Errors

The overall decoding of nonsystematic RS codes for correcting errors using the Euclidean algorithm is summarized in the following (see the Appendix for details):

1. Compute the IFNT of the received code word $\underline{v}' = (A_0', A_1', \ldots, A_{n-1}')$ from Eq. (A-1) in the Appendix to obtain

$$F'(x) = u_0' + u_1'x + \cdots + u_{n-1}'x^{n-1}$$

in Eq. (A-3). Next, calculate the degree of $F'(x)$. If $\deg[F'(x)] < k$, where k is the number of information symbols, then the information polynomial $F(x) = F'(x)$; otherwise, go to step 2.

2. To determine the error-locator polynomial $D(x)$ in Eq. (A-5) and $F'(x)D(x)$, apply the Euclidean algorithm to $M(x)$ defined in Eq. (3) and $F'(x)$. The initial values of the Euclidean algorithm are $p_1(x) = 0, p_0(x) = 1, r_{-1}(x) = M(x)$, and $r_0(x) = F'(x)$. The iterative procedure of the Euclidean algorithm terminates when $\deg[r_i(x)] < n - \lfloor (d-1)/2 \rfloor$ where $\lfloor x \rfloor$ denotes the greatest integer less than or equal to x.

3. Compute $F(x)$ from Eq. (A-14).

A flowchart of a decoding algorithm to correct errors only is depicted in Fig. 1. An example of this decoding scheme is given in Example 1.

B. Decoding to Correct Errors and Erasures

Shiozaki [5] suggests a decoding scheme to correct both errors and erasures. This algorithm ignores the erasure locations and uses the Chinese remainder theorem and the Euclidean algorithm to decode the shortened RS codeword. However, the shortened codeword loses the FFT structure; thus, a fast transform decoding scheme cannot be used. In this section, an improved decoding scheme is suggested which uses the fast-transform techniques discussed in the previous sections to decode RS codewords with both errors and erasures.

Suppose an RS codeword is transmitted through a noisy channel. Let there be s erasure symbols and t error symbols in the codeword such that $2t + s \leq n - k$. Next, assume that the symbols at positions $k_1, k_2, \ldots, k_s$ are erasure symbols and that the symbols at positions $\ell_1, \ell_2, \ldots, \ell_t$ are error symbols. Finally, define

$$D_1(x) = \prod_{i=1}^{s} (x - \gamma^{k_i}) \qquad \text{(known)} \qquad (4)$$

and

$$D_2(x) = \prod_{i=1}^{t} (x - \gamma^{\ell_i}) \qquad \text{(unknown)}$$

and

$$D(x) = D_1(x)D_2(x)$$

By an extension of the derivation of the key Eq. (A-9) given in the Appendix, the following key equation for both errors and erasures can be obtained:

$$-M(x)B(x) + F'(x)D_1(x)D_2(x) = F(x)D_1(x)D_2(x) \qquad (5)$$

where $B(x)$ is as defined in Eq. (A-5) in the Appendix, $\deg[D_2(x)] \leq \lfloor (d-1-s)/2 \rfloor$, and $\deg[F(x)D_1(x)D_2(x)] \leq n - \lfloor (d-1-s)/2 \rfloor - 1$.

The Euclidean algorithm is an iterative procedure which can be used to find in Eq. (5) the greatest common divisor (GCD) of $M(x)$ and $F'(x)D_1(x)$ [10]. An important intermediate relationship among the polynomials of the Euclidean algorithm is given in the equation

$$F'(x)D_1(x)p_i(x) + M(x)s_i(x) = r_i(x) \qquad (6a)$$

and

$$\deg[p_i(x)] + \deg[r_i(x)] < \deg[M(x)] \quad \text{for } -1 \leq i \leq m \qquad (6b)$$

where i is the iterative index and $r_m(x)$ is the GCD of $F'(x)D_1(x)$ and $M(x)$. The algorithm involves four sequences of polynomials: $s_i(x)$, $p_i(x)$, $r_i(x)$, and $q_i(x)$. The initial conditions are set in accordance with the following rules:

1. For $\deg[F'(x)D_1(x)] \leq \deg[M(x)]$, set $s_{-1}(x) = 1$, $s_0(x) = 0$, $p_{-1}(x) = 0$, $p_0(x) = 1$, $r_{-1}(x) = M(x)$, and $r_0(x) = F'(x)D_1(x)$.

2. For $\deg[F'(x)D_1(x)] > \deg[M(x)]$, set $s_{-1}(x) = 0$, $s_0(x) = 1$, $p_1(x) = 1$, $p_0(x) = 0$, $r_{-1}(x) = F'(x)D_1(x)$, and $r_0(x) = M(x)$.

Since $2t + s \leq n - k$,

$$\deg[D_2(x)] + \deg[F(x)D_1(x)D_2(x)] = 2t + s + k - 1 < n = \deg[M(x)] \tag{7}$$

Therefore, let $2t + s \leq n - k$, $u = \lfloor (d-1-s)/2 \rfloor$, and $v = n - \lfloor (d-1-s)/2 \rfloor - 1$. By the proof of the theorem in the Appendix and Eqs. (5), (6a), (6b), and (7), there exists a unique index j in the Euclidean algorithm such that $D_2(x) = \lambda(x)p_j(x)$ and $F(x)D_1(x)D_2(x) = \lambda(x)r_j(x)$, where $\lambda(x)$ is some polynomial, $\deg[p_j(x)] \leq \lfloor (d-1-s)/2 \rfloor$, and $\deg[r_j(x)] < n - \lfloor (d-1-s)/2 \rfloor$. Thus, $F(x)$ can be reconstructed as follows:

$$F(x) = \frac{r_i(x)}{P_i(x)D_1(x)} \tag{8}$$

The overall decoding of nonsystematic RS codes for correcting errors and erasures using the Euclidean algorithm is summarized in the following steps:

1. Use step 1. from the description of decoding for correcting errors.

2. Compute the erasure-locator polynomial $D_1(x)$ from Eq. (4). Next, compare $\deg[F'(x)D_1(x)]$ with $\deg[M(x)]$. If $\deg[F'(x)D_1(x)] \leq \deg[M(x)]$, set $p_{-1}(x) = 0$, $p_0(x) = 1$, $r_{-1}(x) = M(x)$, and $r_0(x) = F'(x)D_1(x)$; otherwise, set $p_{-1}(x) = 1$, $p_0(x) = 0$, $r_{-1}(x) = F'(x)D_1(x)$, and $r_0(x) = M(x)$.

3. To determine the error-locator polynomial $D_2(x)$ and $F'(x)D(x)$, apply the Euclidean algorithm to $M(x)$ and $F'(x)D_1(x)$. The initial values of the Euclidean algorithm are defined in step 2; the iterative procedure of the algorithm terminates when $\deg[r_i(x)] < n - \lfloor (d-1-s)/2 \rfloor$.

4. Compute $F(x)$ from Eq. (8).

A flowchart of the decoding scheme for correcting both errors and erasures is given in Fig. 2. A depiction of this decoding scheme is presented in Example 2.

This simpler, faster transform-decoding scheme using the FNT for RS codes is particularly suitable for pipeline VLSI implementation. The transform-decoding scheme utilizes an efficient FNT to compute the corrupted information polynomial $F'(x)$ in a manner analogous to syndrome computation in the conventional decoding schemes. However, this new algorithm does not require the extra steps needed to solve the error-locator and error-evaluator polynomials.

C. Examples of the Two Decoding Methods

Example 1. Consider the Fermat prime $F_3 = 17$, and let $k = 4$. This is an (8,4) RS code over $GF(17)$, which is capable of correcting two errors or less. It is shown in [6] that $\gamma = 2$ is a primitive 8th root of unity. Also, for this case,

$$M(x) = \prod_{i=0}^{i=7}(x - \gamma^i) = x^8 - 1$$

Let the four information symbols be $\underline{u} = (2,3,1,4)$. Then $F(x) = 2 + 3x + x^2 + 4x^3$. An FNT on the sequence $(2,3,1,4,0,0,0,0)$ yields the codeword $\underline{\nu} = (10,10,14,13,13,2,5,0)$. Next, let the third and seventh symbols be erroneous. Thus, $\underline{e} = (0,0,5,0,0,0,15,0)$ and $\underline{\nu}' = (10,10,2,13,13,2,3,0)$, where $\underline{\nu}$, $\underline{e}$, and $\underline{\nu}'$ are as defined in the Appendix. After taking the IFNT on $\underline{\nu}'$, one obtains $F'(x) = 13 + 8x + 7x^2 + 16x^3 + 11x^4 + 5x^5 + 6x^6 + 12x^7$. The Euclidean algorithm stops after the second iteration to yield $r_2(x) = 10x^5 + 11x^4 + 9x^3 + 16x^2 + 16x + 5$ and $p_2(x) = 11x^2 + 11$. Then $F(x)$ is recaptured as

$$F(x) = \frac{r_2(x)}{p_2(x)} = 2 + 3x + x^2 + 4x^3$$

That is, $\underline{u} = (2,3,1,4)$.

Example 2. Consider the same codeword $\underline{\nu} = (10,10,14,13,13,2,5,0)$ given in Example 1. Let the first symbol be an error symbol, and the third and seventh symbols be erasure symbols. Thus, $\underline{e} = (1,0,5,0,0,0,15,0)$, and $\underline{\nu}' = (11,10,2,13,13,2,3,0)$. After the IFNT is taken of ν', one obtains $F'(x) = 11 + 6x + 5x^2 + 14x^3 + 9x^4 + 3x^5 + 4x^6 + 10x^7$. Since the erasure symbols are at the third and seventh positions, $D_1(x) = (x - 2^2)(x - 2^6) = x^2 + 1$. Thus, $F'(x)D_1(x) = 10x^9 + 4x^8 + 13x^7 + 13x^6 + 14x^4 + 3x^3 + 16x^2 + 6x + 11$. The Euclidean algorithm stops after the second iteration to yield $r_2(x) = x^6 + 12x^5 + 10x^4 + 16x^3 + 4x + 8$ and $p_2(x) = 13x + 4$. Then $F(x)$ is recaptured as

$$F(x) = \frac{r_2(x)}{p_2(x)D_1(x)} = 2 + 3x + x^2 + 4x^3$$

That is, $\underline{u} = (2,3,1,4)$.

V. Conclusions

In this article, a fast transform decoding scheme is introduced which is particularly suitable for VLSI implementation. This scheme first utilizes the highly efficient Fermat number transform to calculate the corrupted information polynomial $F'(x)$. It then uses the Euclidean algorithm to evaluate the information polynomial $F(x)$ directly, without going through the intermediate steps of solving the error-locator and error-evaluator polynomials. Thus, this fast-transform decoding scheme is faster and simpler than the decoding scheme in [1].

Acknowledgments

The authors wish to thank Dr. Les Deutsch, Dr. Robert J. McEliece, Dr. Ed Satorius, and Dr. Laif Swanson at the Jet Propulsion Laboratory for their mathematical suggestions pertaining to the research which led to this article.

References

[1] A. Shiozaki and F. Nishida, "A New Decoding Method of Redundant Residue Polynomial Codes," *Bull. Univ. of Osaka Prefecture*, series A, vol. 24, no. 1, pp. 101–112, 1975.

[2] R. Blahut, *Theory and Practice of Error Control Codes*, Reading, Massachusetts: Addison-Wesley, 1984.

[3] I. S. Reed and G. Solomon, "Polynomial Codes Over Certain Finite Fields," *SIAM*, vol. 8, no. 2, pp. 300–304, 1960.

[4] J. J. Stone, "Multiple-burst Error Correction with the Chinese Remainder Theorem," *SIAM*, vol. 11, no. 1, pp. 74–81, 1963.

[5] A. Shiozaki, "Decoding of Redundant Residue Polynomial Codes Using Euclid's Algorithm," *IEEE Trans. Information Theory*, vol. 34, no. 5, pp. 1351–1354, September 1988.

[6] R. C. Agarwal and C.S. Burrows, "Fast Convolution Using Fermat Number Transforms with Applications to Digital Filtering," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. ASSP-22, no. 2, pp. 87–97, April 1974.

[7] I. S. Reed, T. K. Truong, and L. R. Welch, "The Fast Decoding of Reed-Solomon Codes Using Fermat Transforms," *IEEE Trans. Information Theory*, vol. IT-24, no. 4, pp. 497–499, July 1978.

[8] A. Oppenheim and R. Schafer, *Digital Signal Processing*, New York: Prentice-Hall, 1975.

[9] H. C. Shyu, T. K. Truong, I. S. Reed, I. S. Hsu, and J. J. Chang, "A New VLSI Complex Integer Multiplier Which Uses a Quadratic-Polynomial Residue System with Fermat Number," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. ASSP-35, no. 7, pp. 1076–1079, July 1987.

[10] R. McEliece, *The Theory of Information and Coding*, vol. 3 of *The Encyclopedia of Mathematics and Its Applications*, Reading, Massachusetts: Addison-Wesley, 1977.

[11] E. R. Berlekamp, *Algebraic Coding Theory*, New York: McGraw-Hill, 1968.

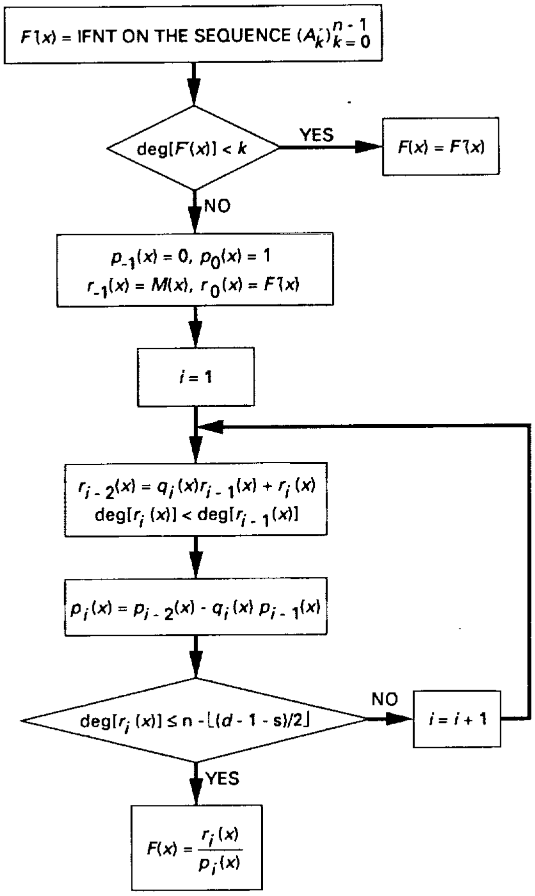
F'(x) = IFNT ON THE SEQUENCE $(A_k')_{k=0}^{n-1}$
deg[F'(x)] < k
YES
F(x) = F'(x)
NO
$p_{-1}(x) = 0$, $p_0(x) = 1$
$r_{-1}(x) = M(x)$, $r_0(x) = F'(x)$
i = 1
$r_{i-2}(x) = q_i(x)r_{i-1}(x) + r_i(x)$
$\deg[r_i(x)] < \deg[r_{i-1}(x)]$
$p_i(x) = p_{i-2}(x) - q_i(x)\,p_{i-1}(x)$
$\deg[r_i(x)] \le n - \lfloor(d-1-s)/2\rfloor$
NO
i = i + 1
YES
$F(x) = \frac{r_i(x)}{p_i(x)}$

Fig. 1. Flowchart of decoding procedure to correct errors only.

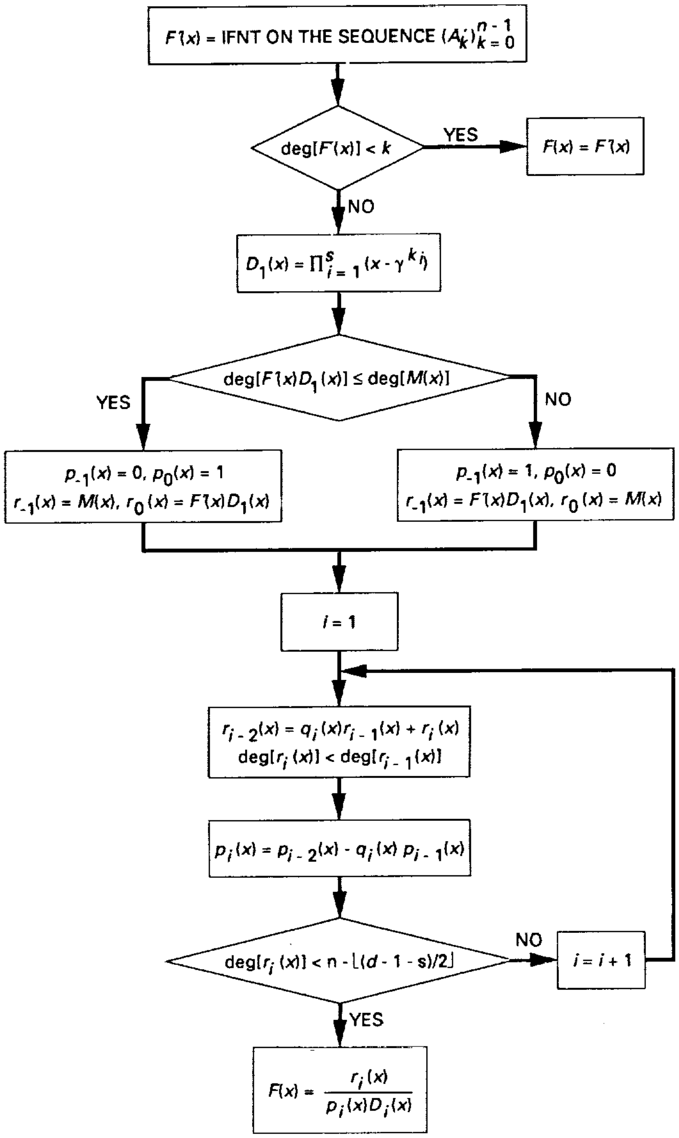
F'(x) = IFNT ON THE SEQUENCE $(A_k')_{k=0}^{n-1}$
deg[F'(x)] < k
YES
F(x) = F'(x)
NO
$D_1(x) = \Pi_{i=1}^{s}(x - \gamma^{k_i})$
$\deg[F'(x)D_1(x)] \le \deg[M(x)]$
YES
NO
$p_{-1}(x) = 0$, $p_0(x) = 1$
$r_{-1}(x) = M(x)$, $r_0(x) = F'(x)D_1(x)$
$p_{-1}(x) = 1$, $p_0(x) = 0$
$r_{-1}(x) = F'(x)D_1(x)$, $r_0(x) = M(x)$
i = 1
$r_{i-2}(x) = q_i(x)r_{i-1}(x) + r_i(x)$
$\deg[r_i(x)] < \deg[r_{i-1}(x)]$
$p_i(x) = p_{i-2}(x) - q_i(x)\,p_{i-1}(x)$
$\deg[r_i(x)] < n - \lfloor(d-1-s)/2\rfloor$
NO
i = i + 1
YES
$F(x) = \frac{r_i(x)}{p_i(x)D_i(x)}$

Fig. 2. Flowchart of decoding procedure to correct errors and erasures.

Appendix
Decoding RS Codes Using the Euclidean Algorithm

Suppose the codeword $\underline{v} = (A_0, A_1, \ldots, A_{n-1})$ is transmitted through a noisy channel. Assume that the symbols at positions $\ell_1, \ell_2, \ldots,$ and ℓ_t are in error. The received codeword $\underline{v}'$ is thus represented by

$$\underline{v}' = \underline{v} + \underline{e} = (A_0', A_1', \ldots, A_{n-1}' \qquad \text{(A-1)}$$

where $\underline{e}$ is the error vector defined by

$$\underline{e} = (0, \ldots, e_{\ell_1}, 0, \ldots, e_{\ell_t}, \ldots, 0)V \qquad \text{(A-2)}$$

Let $(u_0', u_1', \ldots, u_{n-1}')$ and $(w_0, w_1, \ldots, w_{n-1})$ be the inverse transforms of $\underline{v}'$ and $\underline{e}$ respectively. Also let $F'(x) = u_0' + u_1'x + \cdots + u_{n-1}'x^{n-1}$ be defined as the corrupted information polynomial, and $E(x) = w_0 + w_1 x + \cdots + w_{n-1}x^{n-1}$ be defined as the error polynomial.

It is not difficult to see that the residue representations of $F'(x)$ and $E(x)$, modulo $m_i(x)$ are Eqs. (A-1) and (A-2) respectively. That is, $\underline{v}'$ and $\underline{e}$ can be written, respectively, as

$$\underline{v}' = (F'(1), F'(\gamma), \ldots, F'(\gamma^{n-1}))$$

and

$$\underline{e} = (E(1), E(\gamma), \ldots, E(\gamma^{n-1}))$$

From Section III, an RS codeword $\underline{v}$ is generated by an information polynomial $F(x)$ via the following:

$$\underline{v} = (F(1), F(\gamma), \ldots, F(\gamma^{n-1}))$$

Since $\underline{v}' = \underline{v} + \underline{e}$, one obtains $F'(\gamma^i) = F(\gamma^i) + E(\gamma^i)$ for $0 \le i \le n-1$. Thus, there are at least n values of x for which $F'(x)$ and $F(x) + E(x)$ are equal. It is obvious that $\deg[F(x)] < k, \deg[F'(x)] < n$, and $\deg[E(x)] < n$. Hence, by the fundamental theorem of algebra,

$$F'(x) = F(x) + E(x) \qquad \text{(A-3)}$$

Since RS codes are a special case of RRP codes, it is shown in [5] that $F'(x)$ and $E(x)$ can also be reconstructed using the Chinese remainder theorem as follows:

$$F'(x) \equiv \left[\sum_{i=0}^{n-1} \frac{M(x)}{m_P(x)} t_i(x) A_i'\right] \bmod M(x)$$

and

$$E(x) \equiv \left[\sum_{i=1}^{t} \frac{M(x)}{m_{\ell_i}(x)} t_{\ell_i}(x) e_{\ell_i}\right] \bmod M(x) \qquad \text{(A-4)}$$

Let

$$B(x) \equiv \left[\sum_{i=1}^{t} \frac{D(x)}{m_{\ell_i}(x)} t_{\ell_i}\right] \bmod D(x) \qquad \text{(A-5)}$$

where

$$D(x) = \prod_{i=1}^{t} m_{\ell_i}(x)$$

is called the error-locator polynomial. The key equation of the decoding algorithm is derived from these relationships. First, let

$$A(x) = \frac{M(x)}{D(x)} \qquad \text{(A-6)}$$

and

$$B'(x) = \sum_{i=1}^{t} \frac{D(x)}{m_{\ell_i}(x)} t_{\ell_i}(x) e_{\ell_i}$$

Then, by Eq. (A-4), one has

$$E(x) \equiv \left[\frac{M(x)}{D(x)} \sum_{i=1}^{t} \frac{D(x)}{m_{\ell_i}} t_{\ell_i}(x) e_{\ell_i}\right] \bmod M(x)$$

$$[A(x)B'(x)] \bmod A(x)D(x) \qquad \text{(A-7)}$$

Equation (A-7) can now be re-expressed as:

$$E(x) = A(x)\,[\lambda(x)D(x) + B'(x)]$$

$$= A(x)\,[B'(x) \bmod D(x)] \qquad \text{(A-8)}$$

where $\lambda(x)$ is some polynomial over the finite field. Using Eq. (A-5), a substitution of $A(x)$ and $B'(x)$ in Eq. (A-6) into Eq. (A-8) yields

$$E(x) = \frac{M(x)}{D(x)}B(x) \tag{A-9}$$

The proof of Eq. (A-9) is similar to the proof given in [5]. A similar result of Eq. (A-9) is given by Blahut in theorem 9.1.1 of [2] using a spectral technique. The decoder in Fig. 9.2 of [2] applies the Euclidean algorithm to $M(x)$ and the $2t$ high-order coefficients of $E(x)$ to determine the error-locator polynomial $D(x)$. Then a recursive extension is used to compute the rest of the coefficients of $E(x)$ from the known $D(x)$. Finally, the inverse transform over $GF(2^n)$ of E_j is taken to recover the error pattern.

The next paragraph describes how the decoder developed in this article applies the Euclidean algorithm to the polynomials $M(x)$ and $F'(x)$ rather than to the usual $M(x)$ and the syndrome polynomial $S(x)$, i.e., the $2t$ high-order coefficients of $E(x)$. In other words, to determine polynomials $D(x)$ and $F(x)D(x)$, this new decoder applies the Euclidean algorithm to $M(x)$ and $F'(x)$. Thus, $F(x)$ can be reconstructed from $F(x) = F(x)D(x)/D(x)$. The advantage of this new decoder over the decoder developed in Fig. 9.3 of [2] is that both the recursive extension and the inverse transform can be replaced by a single polynomial division.

By combining Eqs. (A-3) and (A-9), the key equation is obtained as follows:

$$-M(x)B(x) + F'(x)D(x) = F(x)D(x) \tag{A-10}$$

where $B(x)$ is defined as in Eq. (A-5).

The Euclidean algorithm is an iterative procedure to find the greatest common divisor (GCD) of $M(x)$ and $F'(x)$ [10]. An important intermediate relationship among the polynomials of the Euclidean algorithm is given in the following:

$$-M(x)s_i(x) + F'(x)p_i(x) = r_i(x) \tag{A-11}$$

and

$$\deg[P_i(a)] + \deg[r_i(x)] < \deg[M(x)]$$

and

$$\text{for } -1 \leq i \leq m \tag{A-12}$$

where i is the iterative index, and $r_m(x)$ is the GCD of $F'(x)$ and $M(x)$. The algorithm involves four sequences of polynomials: $s_i(x)$, $p_i(x)$, $r_i(x)$, and $q_i(x)$. The initial conditions are: $s_{-1}(x) = 1$, $s_0(x) = 0$, $p_{-1}(x) = 0$, $p_0(x) = 1$, $r_{-1}(x) = M(x)$, and $r_0(x) = F'(x)$; $q_{-1}(x)$ and $q_0(x)$ are not defined.

The following lemma and theorem [10] show that the Euclidean algorithm can be applied to the key Eq. (A-10) to solve for the information polynomial $F(x)$.

Lemma. Given two non-negative integers μ and ν with $\nu \geq \deg[r_m(x)]$ satisfying $\mu + \nu = \deg[M(x)] - 1$, there exists a unique index j, $0 \leq j \leq m$, such that

$$\deg[p_j(x)] \leq \mu$$

and

$$\deg[r_j(x)] \leq \nu$$

For the proof, see [10].

Using the above lemma, the following theorem can be proved [10]:

Theorem. Suppose $p(x)$, $s(x)$, and $r(x)$ are nonzero polynomials satisfying

$$-M(x)s(x) + F'(x)p(x) = r(x)$$

and

$$\deg[p(x)] + \deg[r(x)] < \deg[M(x)]$$

There then exists a unique index j, $0 \leq j \leq m$, and a polynomial $\lambda(x)$ such that

$$p(x) = \lambda(x)p_j(x)$$

and

$$r(x) = \lambda(x)r_j(x)$$

Now let $n - k = 2T$, where T is the maximum number of errors in an RS code which can be corrected. If the number of errors t in a received RS codeword is less than or equal to T, then $\deg[D(x)] \leq T$ and $\deg[F(x)D(x)] \leq k + T - 1 = n - T - 1$. Thus,

$$\deg[D(x)] + \deg[F(x)D(x)] < \deg[M(x)] = n \tag{A-13}$$

Thus, let $n-k \geq 2t$, $u = T$, and $v = \deg[M(x)] - 1 - \mu = n - T - 1$. By the proof of the above theorem and Eqs. (A-10), (A-11), (A-12), and (A-13), there exists a unique index j in the Euclidean algorithm such that $D(x) = \lambda(x)p_j(x)$ and $F(x)D(x) = \lambda(x)r_j(x)$, where $\deg[p_j(x)] \leq T$ and $\deg[r_j(x)] \leq n - T - 1$. Thus, $F(x)$ can be reconstructed as:

$$F(x) = \frac{r_j(x)}{p_j(x)} \tag{A-14}$$