CHAPTER 7

# Nonbinary BCH Codes, Reed–Solomon Codes, and Decoding Algorithms

This chapter presents nonbinary BCH codes with symbols from $GF(q)$ and their decoding algorithms. The most important and popular subclass of nonbinary BCH codes is the class of Reed–Solomon codes. Even though Reed–Solomon codes form a subclass of BCH codes, they were constructed independently using a totally different approach by Reed and Solomon in 1960 [1], the same year as BCH codes were discovered. The relationship between Reed–Solomon codes and BCH codes was proved by Gorenstein and Zierler in 1961 [2]. The minimum distance of a Reed–Solomon code is equal to the number of its parity-check symbols plus one. Reed–Solomon codes are very effective in correcting random symbol errors and random burst errors, and they are widely used for error control in communication and data storage systems, ranging from deep-space telecommunications to compact discs. Concatenation of these codes as outer codes with simple binary codes as inner codes provides high data reliability with reduced decoding complexity.

Decoding of a nonbinary BCH code or a Reed–Solomon code requires determination of both the locations and the values of the symbol errors. The first error-correction procedure for nonbinary BCH and Reed–Solomon codes was found by Gorenstein and Zierler [2], and it was later improved by Chien [3] and Forney [4]. But Berlekamp's iterative decoding algorithm [5] presented in the previous chapter was the first efficient decoding algorithm for both binary and nonbinary BCH codes. In 1975 Sugiyama, Kasahara, Hirasawa, and Namekawa showed that the Euclidean algorithm for finding the greatest common divisor of two polynomials can be used for decoding BCH and Reed–Solomon codes [6]. This Euclidean decoding algorithm is simple in concept and easy to implement. BCH and Reed–Solomon codes can also be decoded in the frequency domain. The first such frequency-domain decoding algorithm was introduced by Gore [7], and it was later much improved by Blahut [8]. All the preceding decoding algorithms can be modified for correcting both symbol errors and erasures.

Reed–Solomon codes have been proved to be good error-detecting codes [9], and their weight distribution has been completely determined [10]–[12].

Good treatment of nonbinary BCH and Reed–Solomon codes and their decoding algorithms can be found in [5] and [13–20].

## 7.1 $q$-ARY LINEAR BLOCK CODES

Consider a Galois field $GF(q)$ with $q$ elements. It is possible to construct codes with symbols from $GF(q)$. These codes are called $q$-ary codes. The concepts and properties developed for the binary codes in the previous chapters apply to $q$-ary codes with few modifications.

Consider the vector space of all the $q^n$ $n$-tuples over $GF(q)$:

$$(v_0, v_1, \cdots, v_{n-1})$$

with $v_i \in GF(q)$ for $0 \leq i < n$. The vector addition is defined as follows:

$$(u_0, u_1, \cdots, u_{n-1}) + (v_0, v_1, \cdots, v_{n-1}) \triangleq (u_0 + v_0, u_1 + v_1, \cdots, u_{n-1} + v_{n-1}),$$

where the addition $u_i + v_i$ is carried out in $GF(q)$. The multiplication of a scalar in $GF(q)$ and an $n$-tuple $(v_0, v_1, \cdots, v_{n-1})$ over $GF(q)$ is given as follows:

$$a \cdot (v_0, v_1, \cdots, v_{n-1}) \triangleq (a \cdot v_0, a \cdot v_1, \cdots, a \cdot v_{n-1})$$

where the product $a \cdot v_i$ is carried out in $GF(q)$. The inner product of two $n$-tuples, $(u_0, u_1, \cdots, u_{n-1})$ and $(v_0, v_1, \cdots, v_{n-1})$, is defined as follows:

$$(u_0, u_1, \cdots, u_{n-1}) \cdot (v_0, v_1, \cdots, v_{n-1}) \triangleq \sum_{i=0}^{n-1} u_i \cdot v_i$$

where addition and multiplication are carried out in $GF(q)$.

DEFINITION 7.1 An $(n, k)$ linear block code with symbols from $GF(q)$ is simply a $k$-dimensional subspace of the vector space of all the $n$-tuples over $GF(q)$.

A $q$-ary linear block code has all the structures and properties developed for binary linear block codes. A $q$-ary linear block code is specified either by a generator matrix or by a parity-check matrix over $GF(q)$. Encoding and decoding of $q$-ary linear codes are the same as for binary linear codes, except that operations and computations are performed over $GF(q)$.

A $q$-ary $(n, k)$ cyclic code is generated by a polynomial of degree $n - k$ over $GF(q)$,

$$\mathbf{g}(X) = g_0 + g_1 X + \cdots + g_{n-k-1} X^{n-k-1} + X^{n-k}$$

where $g_0 \neq 0$ and $g_i \in GF(q)$. The generator polynomial $\mathbf{g}(X)$ is a factor of $X^n - 1$. A polynomial $\mathbf{v}(X)$ of degree $n - 1$ or less over $GF(q)$ is a code polynomial if and only if $\mathbf{v}(X)$ is divisible by the generator polynomial $\mathbf{g}(X)$.

In this chapter we present two important classes of cyclic codes over $GF(q)$ whose constructions are based on an extension field of $GF(q)$. Construction of an extension field of $GF(q)$ is similar to the construction of an extension field of $GF(2)$.

A polynomial $f(X)$ with coefficients from $GF(q)$ is called *monic* if the coefficient of the highest power of $X$ is 1. A polynomial $p(X)$ of degree $m$ over $GF(q)$ is said to be *irreducible* if it is not divisible by any polynomial over $GF(q)$ of degree less than $m$ but greater than zero. An irreducible polynomial $p(X)$ of degree $m$ over $GF(q)$ is called *primitive* if the smallest positive integer $n$ for which $p(X)$ divides $X^n - 1$ is $n = q^m - 1$.

For any positive integer $m$, a Galois field $GF(q^m)$ with $q^m$ elements can be constructed from the ground field $GF(q)$. The construction is exactly the same as the construction of $GF(2^m)$ from $GF(2)$. The construction of $GF(q^m)$ is based on a monic primitive polynomial $p(X)$ of degree $m$ over $GF(q)$. Let $\alpha$ be a root of $p(X)$. Then,

$$0,\ 1,\ \alpha,\ \alpha^2,\ \cdots,\ \alpha^{q^m-2}$$

form all the elements of $GF(q^m)$, and $\alpha^{q^m-1} = 1$. The element $\alpha$ is called a *primitive element*. Every element $\beta$ in $GF(q^m)$ can be expressed as a polynomial in $\alpha$,

$$\beta = a_0 + a_1\alpha + a_2\alpha^2 + \cdots + a_{m-1}\alpha^{m-1}$$

where $a_i \in GF(q)$ for $0 \leq i < m$. Then, $(a_0, a_1, \cdots, a_{m-1})$ is a vector representation of $\beta$. Therefore, every element in $GF(q^m)$ has three forms: power (0 is represented by $\alpha^\infty$), polynomial, and vector forms.

The elements in $GF(q^m)$ form all the roots of $X^{q^m} - X$. Let $\beta$ be an element in $GF(q^m)$. The minimal polynomial of $\beta$ is the monic polynomial $\boldsymbol{\phi}(X)$ of the smallest degree over $GF(q)$ that has $\beta$ as a root; that is, $\boldsymbol{\phi}(\beta) = 0$. Just as in the binary case, $\boldsymbol{\phi}(X)$ is irreducible. Let $e$ be the smallest nonnegative integer for which $\beta^{q^e} = \beta$. The integer $e$ is called the *exponent* of $\beta$ and $e \leq m$. The elements $\beta, \beta^q, \beta^{q^2}, \cdots, \beta^{q^{e-1}}$ are conjugates. Then,

$$\boldsymbol{\phi}(X) = \prod_{i=0}^{e-1}(X - \beta^{q^i}),$$

and $\boldsymbol{\phi}(X)$ divides $X^{q^m} - X$.

## 7.2 PRIMITIVE BCH CODES OVER *GF*(*q*)

The binary BCH codes defined in Section 6.1 can be generalized to nonbinary codes in a straightforward manner. Let $\alpha$ be a primitive element in $GF(q^m)$. The generator polynomial $\mathbf{g}(X)$ of a $t$-error-correcting primitive $q$-ary BCH code is the polynomial of the smallest degree over $GF(q)$ that has $\alpha, \alpha^2, \cdots, \alpha^{2t}$ as roots. For $1 \leq i \leq 2t$, let $\boldsymbol{\phi}_i(X)$ be the minimal polynomial of $\alpha^i$. Then,

$$\mathbf{g}(X) = \text{LCM}\{\boldsymbol{\phi}_1(X), \boldsymbol{\phi}_2(X), \cdots, \boldsymbol{\phi}_{2t}(X)\}. \tag{7.1}$$

Because each $\boldsymbol{\phi}_i(X)$ divides $X^{q^m-1} - 1$, $\mathbf{g}(X)$ divides $X^{q^m-1} - 1$. Since $\alpha$ is a primitive element in $GF(q^m)$, $\boldsymbol{\phi}_1(X)$ is a primitive polynomial of degree $m$. Hence, the smallest positive integer $n$ for which $\boldsymbol{\phi}_1(X)$ divides $X^n - 1$ is $n = q^m - 1$. This result implies that $q^m - 1$ is the smallest positive integer for which $\mathbf{g}(X)$ divides $X^{q^m-1} - 1$. Because the degree of each $\boldsymbol{\phi}_i(X)$ is $m$ or less, the degree of $\mathbf{g}(X)$ is $2mt$ or less. Similar to the way we proved the BCH bound for the minimum distance of a binary BCH code, we can prove that the minimum distance of the $q$-ary BCH code generated by $\mathbf{g}(X)$ of (7.1) is lower bounded by $2t + 1$.

Summarizing the foregoing results, we see that the $q$-ary BCH code generated by the polynomial $\mathbf{g}(X)$ of (7.1) is a cyclic code with the following parameters:

Block length: $n = q^m - 1$,

Number of parity-check symbols: $n - k \leq 2mt$,

Minimum distance: $d_{min} \geq 2t + 1$.

This code is capable of correcting $t$ or fewer random symbol errors over a span of $q^m - 1$ symbol positions. For $q = 2$, we obtain the binary primitive BCH codes. Similar to the binary case, the matrix over $GF(q^m)$,

$$\mathbb{H} = \begin{bmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^{n-1} \\ 1 & \alpha^2 & (\alpha^2)^2 & \cdots & (\alpha^2)^{n-1} \\ 1 & \alpha^3 & (\alpha^3)^2 & \cdots & (\alpha^3)^{n-1} \\ \vdots & & & & \vdots \\ 1 & \alpha^{2t} & (\alpha^{2t})^2 & \cdots & (\alpha^{2t})^{n-1} \end{bmatrix},$$

is a parity-check matrix of the $t$-error-correcting primitive $q$-ary BCH code generated by the polynomial $g(X)$ of (7.1).

## 7.3 REED–SOLOMON CODES

The special subclass of $q$-ary BCH codes for which $m = 1$ is the most important subclass of $q$-ary BCH codes. The codes of this subclass are called the Reed–Solomon (RS) codes in honor of their discoverers, Irving S. Reed and Gustave Solomon [1]. RS codes have been widely used for error control in both digital communication and storage systems.

Let $\alpha$ be a primitive element in $GF(q)$. The generator polynomial $g(X)$ of a $t$-error-correcting RS code with symbols from $GF(q)$ has $\alpha, \alpha^2, \cdots, \alpha^{2t}$ as all its roots. Because $\alpha^i$ is an element of $GF(q)$, its minimal polynomial $\phi_i(X)$ is simply $X - \alpha^i$. Then, it follows from (7.1) that

$$\begin{aligned} g(X) &= (X - \alpha)(X - \alpha^2)\cdots(X - \alpha^{2t}) \\ &= g_0 + g_1X + g_2X^2 + \cdots + g_{2t-1}X^{2t-1} + X^{2t} \end{aligned} \tag{7.2}$$

with $g_i \in GF(q)$ for $0 \le i < 2t$. Since $\alpha, \alpha^2, \cdots, \alpha^{2t}$ are roots of $X^{q-1} - 1$, $g(X)$ divides $X^{q-1} - 1$. Therefore, $g(X)$ generates a $q$-ary cyclic code of length $n = q - 1$ with exactly $2t$ parity-check symbols. It follows from the BCH bound that the minimum distance of the code is at least $2t + 1$; however, the generator polynomial $g(X)$ is a code polynomial and has $2t + 1$ terms. None of the coefficients in $g(X)$ can be zero, otherwise the resulting codeword would have weight less than $2t + 1$, which would contradict the BCH bound on the minimum distance. Therefore, $g(X)$ corresponds to a codeword of weight exactly $2t + 1$. This implies that the minimum distance of the RS code generated by the polynomial $g(X)$ of (7.2) is exactly $2t + 1$, and the code is capable of correcting $t$ or fewer symbol errors. In summary, a $t$-error-correcting RS code with symbols from $GF(q)$ has the following parameters:

Block length: $n = q - 1$,

Number of parity-check symbols: $n - k = 2t$,

Dimension: $k = q - 1 - 2t$,

Minimum distance: $d_{min} = 2t + 1$.

Thus, we see that RS codes have two important features: (1) the length of the code is one less than the size of the code alphabet, and (2) the minimum distance is one greater than the number of parity-check symbols. Codes with minimum distance one greater than the number of parity-check symbols are called *maximum distance separable* (MDS) codes. RS codes form the most important class of MDS codes.

### EXAMPLE 7.1

Let $\alpha$ be a primitive element in $GF(2^6)$ constructed based on the primitive polynomial $p(X) = 1 + X + X^6$ (see Table 6.2). Consider the triple-error-correcting RS code with symbols from $GF(2^6)$. The generator polynomial $\mathbf{g}(X)$ of this code has $\alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^5$, and $\alpha^6$ as all its roots; hence,

$$\begin{aligned}\mathbf{g}(X) &= (X+\alpha)(X+\alpha^2)(X+\alpha^3)(X+\alpha^4)(X+\alpha^5)(X+\alpha^6)\\ &= \alpha^{21} + \alpha^{10}X + \alpha^{55}X^2 + \alpha^{43}X^3 + \alpha^{48}X^4 + \alpha^{59}X^5 + X^6.\end{aligned}$$

The code is a (63, 57) triple-error-correcting RS code over $GF(2^6)$.

Encoding of a RS code is similar to that of the binary case. Let

$$\mathbf{a}(X) = a_0 + a_1X + a_2X^2 + \cdots + a_{k-1}X^{k-1}$$

be the message to be encoded, where $k = n - 2t$. In systematic form, the $2t$ parity-check symbols are the coefficients of the remainder $\mathbf{b}(X) = b_0 + b_1X + \cdots + b_{2t-1}X^{2t-1}$ resulting from dividing the message polynomial $X^{2t}\mathbf{a}(X)$ by the generator polynomial. In hardware implementation, this is accomplished by using a division circuit as shown in Figure 7.1. As soon as the message $\mathbf{a}(X)$ has entered the channel and the circuit, the parity-check symbols appear in the register.

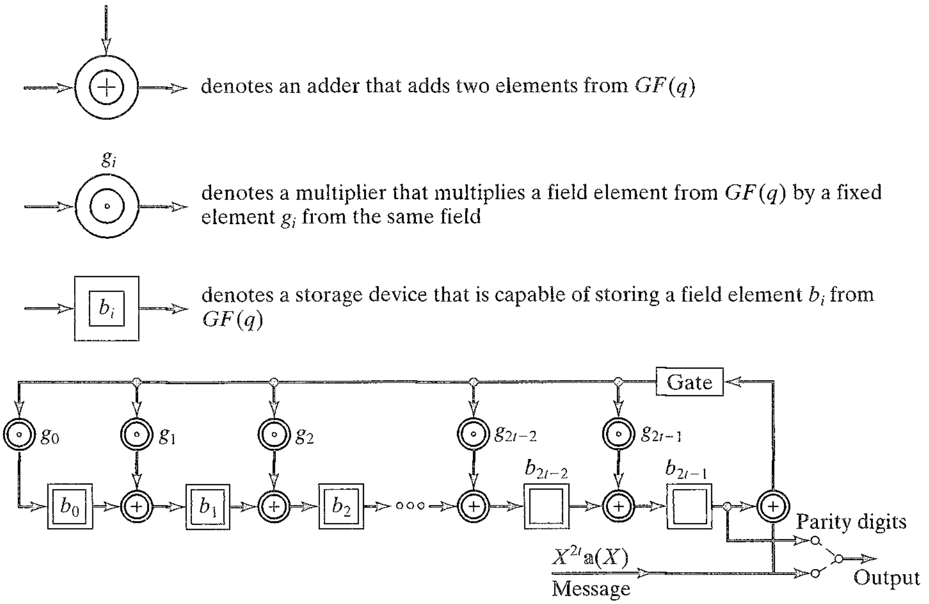
The weight distribution of Reed–Solomon codes has been completely determined [10]–[12]. For a $t$-error-correcting RS code of length $q-1$ with symbols from $GF(q)$, the number of codewords of weight $i$ is given by

$$A_i = \binom{q-1}{i} q^{-2t}\{(q-1)^i + \sum_{j=0}^{2t}(-1)^{i+j}\binom{i}{j}(q^{2t} - q^i)\}, \tag{7.3}$$

for $2t + 1 \le i \le q - 1$.

Suppose a $q$-ary RS code is used for error detection on a discrete memoryless channel with $q$ inputs and $q$ outputs. Let $(1-\varepsilon)$ be the probability that a transmitted symbol is received correctly and $\varepsilon/(q-1)$ be the probability that a transmitted symbol is changed into each of the $q-1$ other symbols. Using the weight distribution given by (7.3), it can be shown that the probability of undetected error for a RS code is [9]

$$\begin{aligned}P_u(E) = q^{-2t}\Bigg\{1 &+ \sum_{i=0}^{2t-1}\binom{q-1}{i}(q^{2t} - q^i)\left(\frac{\varepsilon}{q-1}\right)^i\\ &\times \left(1 - \frac{q\varepsilon}{q-1}\right)^{q-1-i} - q^{2t}(1-\varepsilon)^{q-1}\Bigg\}.\end{aligned} \tag{7.4}$$



denotes an adder that adds two elements from $GF(q)$

denotes a multiplier that multiplies a field element from $GF(q)$ by a fixed element $g_i$ from the same field

denotes a storage device that is capable of storing a field element $b_i$ from $GF(q)$

FIGURE 7.1: Encoding circuit for a $q$-ary RS code with generator polynomial $\mathbf{g}(X) = g_0 + g_1X + g_2X^2 + \cdots + g_{2t-1}X^{2t-1} + X^{2t}$.

It also has been shown in [9] that $P_u(E) < q^{-2t}$ and decreases monotonically as $\varepsilon$ decreases from $(q-1)/q$ to 0. Hence, RS codes are good for error detection.

Let $\lambda$ be a nonnegative integer less than $t$. Suppose a $t$-error-correcting $q$-ary RS code is used to correct all error patterns with $\lambda$ or fewer symbol errors. Let $P_u(E, \lambda)$ denote the probability of undetected error after correction. It has also been proved in [9] that

$$P_u(E,\lambda) = \sum_{h=0}^{\lambda} \binom{q-1}{h} \Bigg[ q^{-2t}(q-1)^h - \varepsilon^h(1-\varepsilon)^{q-1-h} + \sum_{l=0}^{\min\{2t-1, q-1-h\}} \binom{q-1-h}{l} \left(\frac{\varepsilon}{q-1}\right)^l \left(1 - \frac{q\varepsilon}{q-1}\right)^{q-1-h-l} R_{h,l}(\varepsilon) \Bigg], \tag{7.5}$$

where

$$R_{h,l}(\varepsilon) = \sum_{j=0}^{\min\{2t-1-l, h\}} (-1)^{h-j} \binom{h}{j} \left(1 - q^{-2t+l+j}\right) \left(1 - \frac{\varepsilon}{q-1}\right)^j \left(1 - \frac{q\varepsilon}{q-1}\right)^{h-j} \tag{7.6}$$

for $0 \le l < 2t$. It is easy to check that $P_u(E)$ given by (7.4) can be obtained from $P_u(E, \lambda)$ of (7.5) by setting $\lambda = 0$. In [9] it is shown that the probability $P_u(E, \lambda)$ of undetected error after decoding decreases monotonically from

$$(q^{-2t} - q^{-n}) \sum_{h=0}^{\lambda} \binom{q-1}{h} (q-1)^h \tag{7.7}$$

as $\varepsilon$ decreases from $(q-1)/q$ to 0. Therefore, we have the following upper bound on $P_u(E, \lambda)$ from (7.7):

$$P_u(E, \lambda) < q^{-2t} \sum_{h=0}^{\lambda} \binom{q-1}{h} (q-1)^h \tag{7.8}$$

for $0 \leq \varepsilon \leq (q-1)/q$.

The preceding results on error probabilities indicate that RS codes are effective for pure error detection or simultaneous error correction and detection.

Consider the set of codewords in a $(q-1, q-1-2t)$ $q$-ary RS code whose $l$ leading information symbols are identical to zero, with $0 \leq l < q-1-2t$. Deleting these $l$ zero-information symbols from each codeword in the set, we obtain a shortened $(q-1-l, q-1-2t-l)$ RS code. The minimum distance of this shortened RS code is still $2t+1$. Hence, the shortened code is also a MDS code. Encoding and decoding of a shortened RS code are the same as those for the original RS code of natural length (see Section 5.10).

Two information symbols can be added to a RS code of length $q-1$ without reducing its minimum distance. The extended RS code has length $q+1$ and the same number of parity-check symbols as the original code. For a $t$-error-correcting RS code of length $q-1$, the parity-check matrix takes the form

$$H = \begin{bmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^{q-2} \\ 1 & \alpha^2 & (\alpha^2)^2 & \cdots & (\alpha^2)^{q-2} \\ \vdots & & & & \\ 1 & \alpha^{2t} & (\alpha^{2t})^2 & \cdots & (\alpha^{2t})^{q-2} \end{bmatrix}.$$

Then, the parity-check matrix of the extended RS code is

$$H_1 = \left[\begin{array}{cc|c} 0 & 1 & \\ 0 & 0 & \\ \vdots & \vdots & H \\ 0 & 0 & \\ 1 & 0 & \end{array}\right]. \tag{7.9}$$

This code is called a *doubly extended RS code* and is also a MDS code. The preceding result was first obtained by Kasami, Lin, and Peterson [11, 21] and later independently by Wolf [22]. If we delete the first column of $H_1$, we obtain a singly extended RS code of length $q$, which is again a MDS code.

In all practical applications of RS codes for error control, $q$ is set to $2^m$, and code symbols are from the Galois field $GF(2^m)$.

## 7.4 DECODING OF NONBINARY BCH AND RS CODES: THE BERLEKAMP ALGORITHM

In this and the next three sections, we present various algorithms for decoding nonbinary BCH and RS codes.

Let

$$\mathbb{v}(X) = v_0 + v_1X + \cdots + v_{n-1}X^{n-1}$$

be the transmitted code polynomial, and let

$$\mathbb{r}(X) = r_0 + r_1X + \cdots + r_{n-1}X^{n-1}$$

be the corresponding received polynomial. Then, the error pattern added by the channel is

$$\begin{aligned}\mathbb{e}(X) &= \mathbb{r}(X) - \mathbb{v}(X)\\ &= e_0 + e_1X + \cdots + e_{n-1}X^{n-1},\end{aligned}$$

where $e_i = r_i - v_i$ is a symbol in $GF(q)$. Suppose the error pattern $\mathbb{e}(X)$ contains $\nu$ errors (nonzero components) at locations $X^{j_1}, X^{j_2}, \cdots, X^{j_\nu}$ with $0 \leq j_1 < j_2 < \cdots < j_\nu \leq n-1$. Then,

$$\mathbb{e}(X) = e_{j_1}X^{j_1} + e_{j_2}X^{j_2} + \cdots + e_{j_\nu}X^{j_\nu} \tag{7.10}$$

where $e_{j_1}, e_{j_2}, \cdots, e_{j_\nu}$ are error values. Hence, to determine $\mathbb{e}(X)$, we need to know the error locations $X^{j_i}$'s and the error values $e_{j_i}$'s (i.e., we need to know the $\nu$ pairs $(X^{j_i}, e_{j_i})$'s).

As with binary BCH codes, the syndrome is a $2t$-tuple over $GF(q^m)$:

$$(S_1, S_2, \cdots, S_{2t})$$

with $S_i = \mathbb{r}(\alpha^i)$ for $1 \leq i \leq 2t$. Because $\mathbb{r}(X) = \mathbb{v}(X) + \mathbb{e}(X)$, we have

$$S_i = \mathbb{v}(\alpha^i) + \mathbb{e}(\alpha^i) = \mathbb{e}(\alpha^i). \tag{7.11}$$

From (7.10) and (7.11), we obtain the following set of equations that relate the error locations and error values to the syndrome of the received polynomial $\mathbb{r}(X)$:

$$\begin{aligned}S_1 &= e_{j_1}\alpha^{j_1} + e_{j_2}\alpha^{j_2} + \cdots + e_{j_\nu}\alpha^{j_\nu}\\ S_2 &= e_{j_1}\alpha^{2j_1} + e_{j_2}\alpha^{2j_2} + \cdots + e_{j_\nu}\alpha^{2j_\nu}\\ &\vdots\\ S_{2t} &= e_{j_1}\alpha^{2tj_1} + e_{j_2}\alpha^{2tj_2} + \cdots + e_{j_\nu}\alpha^{2tj_\nu}.\end{aligned} \tag{7.12}$$

For $1 \leq i \leq \nu$, let

$$\beta_i \triangleq \alpha^{j_i} \qquad \text{and} \qquad \delta_i \triangleq e_{j_i}$$

which are simply the error-location numbers and error values. With the preceding definitions of $\beta_i$ and $\delta_i$, we can simplify the equations of (7.12) as follows:

$$\begin{aligned} S_1 &= \delta_1\beta_1 + \delta_2\beta_2 + \cdots + \delta_\nu\beta_\nu \\ S_2 &= \delta_1\beta_1^2 + \delta_2\beta_2^2 + \cdots + \delta_\nu\beta_\nu^2 \\ &\vdots \\ S_{2t} &= \delta_1\beta_1^{2t} + \delta_2\beta_2^{2t} + \cdots + \delta_\nu\beta_\nu^{2t}. \end{aligned} \tag{7.13}$$

For decoding a $q$-ary BCH code or a RS code, the same three steps used for decoding a binary BCH code are required; in addition, a fourth step involving computation of error values is required. Therefore, the decoding consists of the following four steps:

1. Compute the syndrome $(S_1, S_2, \cdots, S_{2t})$.
2. Determine the error-location polynomial $\sigma(X)$.
3. Determine the error-value evaluator.
4. Evaluate error-location numbers and error values and perform error correction.

As with binary BCH codes, the error-location polynomial $\sigma(X)$ is defined as

$$\begin{aligned} \sigma(X) &= (1 - \beta_1X)(1 - \beta_2X)\cdots(1 - \beta_\nu X) \\ &= \sigma_0 + \sigma_1X + \sigma_2X^2 + \cdots + \sigma_\nu X^\nu, \end{aligned} \tag{7.14}$$

where $\sigma_0 = 1$. The error-location numbers are the reciprocals of the roots of $\sigma(X)$. From (7.13) and (7.14), it is possible to obtain the following set of equations that relates the coefficients $\sigma_i$'s of $\sigma(X)$ and the syndrome components $S_i$'s:

$$\begin{aligned} S_{\nu+1} + \sigma_1S_\nu + \sigma_2S_{\nu-1} + \cdots + \sigma_\nu S_1 &= 0 \\ S_{\nu+2} + \sigma_1S_{\nu+1} + \sigma_2S_\nu + \cdots + \sigma_\nu S_2 &= 0 \\ &\vdots \\ S_{2t} + \sigma_1S_{2t-1} + \sigma_2S_{2t-2} + \cdots + \sigma_\nu S_{2t-\nu} &= 0. \end{aligned} \tag{7.15}$$

(These equalities will be derived later.) These equations are known as the *generalized Newton's identities*. Our objective is to find the minimum-degree polynomial $\sigma(X)$ whose coefficients satisfy these generalized Newton's identities. Once we have found $\sigma(X)$, we can determine the error locations and error values.

We can compute the error-location polynomial $\sigma(X)$ iteratively in $2t$ steps with Berlekamp's algorithm presented in Section 6.2. At the $\mu$th step, we determine a polynomial of minimum degree

$$\sigma^{(\mu)}(X) = \sigma_0^{(\mu)} + \sigma_1^{(\mu)}X + \cdots + \sigma_{l_\mu}^{(\mu)}X^{l_\mu}$$

whose coefficients satisfy the following $\mu - l_\mu$ identities:

$$\begin{aligned} S_{l_\mu+1} + \sigma_1^{(\mu)} S_{l_\mu} + \cdots + \sigma_{l_\mu}^{(\mu)} S_1 &= 0 \\ S_{l_\mu+2} + \sigma_1^{(\mu)} S_{l_\mu+1} + \cdots + \sigma_{l_\mu}^{(\mu)} S_2 &= 0 \\ &\vdots \\ S_\mu + \sigma_1^{(\mu)} S_{\mu-1} + \cdots + \sigma_{l_\mu}^{(\mu)} S_{\mu-l_\mu} &= 0. \end{aligned} \tag{7.16}$$

The next step is to find a new polynomial of minimum degree

$$\sigma^{(\mu+1)}(X) = \sigma_0^{(\mu+1)} + \sigma_1^{(\mu+1)} X + \cdots + \sigma_{l_{\mu+1}}^{(\mu+1)} X^{l_{\mu+1}}$$

whose coefficients satisfy the following $(\mu + 1) - l_{\mu+1}$ identities:

$$\begin{aligned} S_{l_{\mu+1}+1} + \sigma_1^{(\mu+1)} S_{l_{\mu+1}} + \cdots + \sigma_{l_{\mu+1}}^{(\mu+1)} S_1 &= 0 \\ S_{l_{\mu+1}+2} + \sigma_1^{(\mu+1)} S_{l_{\mu+1}+1} + \cdots + \sigma_{l_{\mu+1}}^{(\mu+1)} S_2 &= 0 \\ &\vdots \\ S_{\mu+1} + \sigma_1^{(\mu+1)} S_\mu + \cdots + \sigma_{l_{\mu+1}}^{(\mu+1)} S_{\mu+1-l_{\mu+1}} &= 0. \end{aligned} \tag{7.17}$$

We continue the foregoing process until $2t$ steps have been completed. At the $2t$th step, we have

$$\sigma(X) = \sigma^{(2t)}(X),$$

which is the true error-location polynomial provided that the number of errors in $\mathbf{e}(X)$ does not exceed the error-correcting capability $t$. In this case, the coefficients of $\sigma(X)$ satisfy the set of generalized Newton's identities given by (7.15).

Suppose we have just completed the $\mu$th step and found the solution $\sigma^{(\mu)}(X)$. To find $\sigma^{(\mu+1)}(X)$, we first check whether the coefficients of $\sigma^{(\mu)}(X)$ satisfy the next generalized Newton's identity; that is,

$$S_{\mu+1} + \sigma_1^{(\mu)} S_\mu + \cdots + \sigma_{l_\mu}^{(\mu)} S_{\mu+1-l_\mu} \overset{?}{=} 0 \tag{7.18}$$

If yes, $\sigma^{(\mu+1)}(X) = \sigma^{(\mu)}(X)$ is the minimum-degree polynomial whose coefficients satisfy the generalized Newton's identities of (7.17). If not, we add a correction term to $\sigma^{(\mu)}(X)$ so that its coefficients satisfy the set of generalized Newton's identities of (7.17). To test the equality of (7.18), we compute the discrepancy,

$$d_\mu = S_{\mu+1} + \sigma_1^{(\mu)} S_\mu + \cdots + \sigma_{l_\mu}^{(\mu)} S_{\mu+1-l_\mu}. \tag{7.19}$$

If $d_\mu = 0$, we set

$$\sigma^{(\mu+1)}(X) = \sigma^{(\mu)}(X).$$

If $d_\mu \neq 0$, we need to adjust $\sigma^{(\mu)}(X)$ to satisfy the equalities of (7.17). We make the correction as follows: we go back to the steps prior to the $\mu$th step and determine a

TABLE 7.1: Berlekamp's iterative procedure for finding the error-location polynomial of a $q$-ary BCH code.

| $\mu$ | $\sigma^{(\mu)}(X)$ | $d_\mu$ | $l_\mu$ | $\mu - l_\mu$ |
|---|---|---|---|---|
| $-1$ | 1 | 1 | 0 | $-1$ |
| 0 | 1 | $S_1$ | 0 | 0 |
| 1 | $1 - S_1X$ | | | |
| 2 | | | | |
| 3 | | | | |
| $\vdots$ | | | | |
| $2t$ | | | | |

polynomial $\sigma^{(\rho)}(X)$ such that $d_\rho \neq 0$ and $\rho - l_\rho$ has the largest value, where $l_\rho$ is the degree of $\sigma^{(\rho)}(X)$. Then,

$$\sigma^{(\mu+1)}(X) = \sigma^{(\mu)}(X) - d_\mu d_\rho^{-1} X^{(\mu-\rho)} \sigma^{(\rho)}(X), \tag{7.20}$$

and $\sigma^{(\mu+1)}(X)$ is the solution at the $(\mu + 1)$th step of the iteration process.

As with binary BCH codes, to find $\sigma(X)$, we fill out Table 7.1 (reproduction of Table 6.5).

We can determine the roots of $\sigma(X)$ in $GF(q^m)$ by substituting the elements of $GF(q^m)$ into $\sigma(X)$ cyclically. If $\sigma(\alpha^i) = 0$, then $\alpha^i$ is a root of $\sigma(X)$, and

$$\alpha^{-i} = \alpha^{q^m - 1 - i}$$

is an error-location number. We can do this systematically with Chien's search.

## EXAMPLE 7.2

Consider a triple-error-correcting RS code with symbols from $GF(2^4)$. The generator polynomial of this code is

$$\begin{aligned}\mathbf{g}(X) &= (X + \alpha)(X + \alpha^2)(X + \alpha^3)(X + \alpha^4)(X + \alpha^5)(X + \alpha^6) \\ &= \alpha^6 + \alpha^9 X + \alpha^6 X^2 + \alpha^4 X^3 + \alpha^{14} X^4 + \alpha^{10} X^5 + X^6.\end{aligned}$$

Let the all-zero vector be the transmitted codeword, and let $\mathbf{r} = (0\,0\,0\,\alpha^7\,0\,0\,\alpha^3\,0\,0\,0\,0\,0\,\alpha^4\,0\,0)$ be the received vector. Thus, $\mathbf{r}(X) = \alpha^7 X^3 + \alpha^3 X^6 + \alpha^4 X^{12}$.

**Step 1.** We compute the syndrome components as follows (using Table 2.8):

$$\begin{aligned}S_1 &= \mathbf{r}(\alpha) = \alpha^{10} + \alpha^9 + \alpha = \alpha^{12}, \\ S_2 &= \mathbf{r}(\alpha^2) = \alpha^{13} + 1 + \alpha^{13} = 1, \\ S_3 &= \mathbf{r}(\alpha^3) = \alpha + \alpha^6 + \alpha^{10} = \alpha^{14}, \\ S_4 &= \mathbf{r}(\alpha^4) = \alpha^4 + \alpha^{12} + \alpha^7 = \alpha^{10}, \\ S_5 &= \mathbf{r}(\alpha^5) = \alpha^7 + \alpha^3 + \alpha^4 = 0, \\ S_6 &= \mathbf{r}(\alpha^6) = \alpha^{10} + \alpha^9 + \alpha = \alpha^{12}.\end{aligned}$$

TABLE 7.2: Steps for finding the error-location polynomial of the (15,9) RS code over $GF(2^4)$.

| $\mu$ | $\sigma^{(\mu)}(X)$ | $d_\mu$ | $l_\mu$ | $\mu - l_\mu$ |
|---|---|---|---|---|
| $-1$ | $1$ | $1$ | $0$ | $-1$ |
| $0$ | $1$ | $\alpha^{12}$ | $0$ | $0$ |
| $1$ | $1+\alpha^{12}X$ | $\alpha^7$ | $1$ | $0$(take $\rho = -1$) |
| $2$ | $1+\alpha^3 X$ | $1$ | $1$ | $1$(take $\rho = 0$) |
| $3$ | $1+\alpha^3 X+\alpha^3 X^2$ | $\alpha^7$ | $2$ | $1$(take $\rho = 0$) |
| $4$ | $1+\alpha^4 X+\alpha^{12} X^2$ | $\alpha^{10}$ | $2$ | $2$(take $\rho = 2$) |
| $5$ | $1+\alpha^7 X+\alpha^4 X^2+\alpha^6 X^3$ | $0$ | $3$ | $2$(take $\rho = 3$) |
| $6$ | $1+\alpha^7 X+\alpha^4 X^2+\alpha^6 X^3$ | — | — | — |

**Step 2.** To find the error-location polynomial $\sigma(X)$, we fill out Table 7.1 and obtain Table 7.2. Thus, $\sigma(X) = 1 + \alpha^7 X + \alpha^4 X^2 + \alpha^6 X^3$.

**Step 3.** By substituting $1, \alpha, \alpha^2, \cdots, \alpha^{14}$ into $\sigma(X)$, we find that $\alpha^3, \alpha^9$, and $\alpha^{12}$ are roots of $\sigma(X)$. The reciprocals of these roots are $\alpha^{12}, \alpha^6$, and $\alpha^3$, which are the error-location numbers of the error pattern $\mathbb{e}(X)$. Thus, errors occur at positions $X^3$, $X^6$, and $X^{12}$.

Next, we need to determine the error values, by finding the error-value evaluator. We define the syndrome polynomial $\mathbb{S}(X)$ as follows:

$$\begin{aligned}\mathbb{S}(X) &\triangleq S_1 + S_2 X + \cdots + S_{2t} X^{2t-1} + S_{2t+1} X^{2t} + \cdots \\ &= \sum_{j=1}^{\infty} S_j X^{j-1}.\end{aligned} \tag{7.21}$$

Note that only the coefficients of the first $2t$ terms are known. For $1 \le j < \infty$, we also define

$$S_j = \sum_{l=1}^{\nu} \delta_l \beta_l^j. \tag{7.22}$$

The first $2t$ such $S_j$'s are simply the $2t$ equalities of (7.13). Combining (7.21) and (7.22), we can put $S(X)$ in the following form:

$$\begin{aligned}\mathbb{S}(X) &= \sum_{j=1}^{\infty} X^{j-1} \sum_{l=1}^{\nu} \delta_l \beta_l^j \\ &= \sum_{l=1}^{\nu} \delta_l \beta_l \sum_{j=1}^{\infty} (\beta_l X)^{j-1}.\end{aligned} \tag{7.23}$$

Note that

$$\frac{1}{(1-\beta_l X)} = \sum_{j=1}^{\infty} (\beta_l X)^{j-1}. \tag{7.24}$$

It follows from (7.23) and (7.24) that

$$\mathbb{S}(X) = \sum_{l=1}^{\nu} \frac{\delta_l \beta_l}{1 - \beta_l X}. \tag{7.25}$$

Consider the product $\sigma(X)\mathbb{S}(X)$,

$$\begin{aligned}\sigma(X)\mathbb{S}(X) &= (1 + \sigma_1 X + \cdots + \sigma_\nu X^\nu) \cdot (S_1 + S_2 X + S_3 X^2 + \cdots) \\ &= S_1 + (S_2 + \sigma_1 S_1)X + (S_3 + \sigma_1 S_2 + \sigma_2 S_1)X^2 + \cdots + \\ &\quad (S_{2t} + \sigma_1 S_{2t-1} + \cdots + \sigma_\nu S_{2t-\nu})X^{2t-1} + \cdots\end{aligned} \tag{7.26}$$

Using (7.25), we can also put $\sigma(X)S(X)$ in the following form:

$$\begin{aligned}\sigma(X)\mathbb{S}(X) &= \left\{\prod_{i=1}^{\nu}(1 - \beta_i X)\right\} \cdot \left\{\sum_{l=1}^{\nu} \frac{\delta_l \beta_l}{1 - \beta_l X}\right\} \\ &= \sum_{l=1}^{\nu} \frac{\delta_l \beta_l}{1 - \beta_l X} \cdot \prod_{i=1}^{\nu}(1 - \beta_i X) \\ &= \sum_{l=1}^{\nu} \delta_l \beta_l \prod_{i=1, i \neq l}^{\nu} (1 - \beta_i X).\end{aligned} \tag{7.27}$$

We define

$$\mathbb{Z}_0(X) \triangleq \sum_{l=1}^{\nu} \delta_l \beta_l \prod_{i=1, i \neq l}^{\nu} (1 - \beta_i X). \tag{7.28}$$

Note that $\mathbb{Z}_0(X)$ is a polynomial of degree $\nu - 1$. From (7.26), (7.27), and (7.28), we see that $\mathbb{Z}_0(X)$ must be equal to the first $\nu$ terms from $X^0$ to $X^{\nu-1}$ in $\sigma(X)\mathbb{S}(X)$ of (7.26); that is,

$$\begin{aligned}\mathbb{Z}_0(X) &= S_1 + (S_2 + \sigma_1 S_1)X + (S_3 + \sigma_1 S_2 + \sigma_2 S_1)X^2 \\ &\quad + \cdots + (S_\nu + \sigma_1 S_{\nu-1} + \cdots + \sigma_{\nu-1} S_1)X^{\nu-1}.\end{aligned} \tag{7.29}$$

Because the degree of $\mathbb{Z}_0(X)$ is $\nu - 1$, the coefficients of powers from $X^\nu$ to $X^{2t-1}$ in the expansion of $\sigma(X)\mathbb{S}(X)$ ((7.26)) must be zero. Setting these coefficients to zero, we have exactly the same set of equations as (7.15).

Next, we show that the error values can be determined from $\mathbb{Z}_0(X)$ and $\sigma(X)$. Substituting $\beta_k^{-1}$ in $\mathbb{Z}_0(X)$ (given by (7.28)), we have

$$\begin{aligned}\mathbb{Z}_0(\beta_k^{-1}) &= \sum_{l=1}^{\nu} \delta_l \beta_l \prod_{i=1, i \neq l}^{\nu} (1 - \beta_i \beta_k^{-1}) \\ &= \delta_k \beta_k \prod_{i=1, i \neq k}^{\nu} (1 - \beta_i \beta_k^{-1}).\end{aligned} \tag{7.30}$$

We take the derivative of $\sigma(X)$ in (7.14),

$$\begin{aligned}\sigma'(X) &= \frac{d}{dX}\prod_{i=1}^{\nu}(1-\beta_i X) \\ &= -\sum_{l=1}^{\nu}\beta_l \prod_{i=1, i\neq l}^{\nu}(1-\beta_i X).\end{aligned} \tag{7.31}$$

Then,

$$\sigma'(\beta_k^{-1}) = -\beta_k \prod_{i=1, i\neq k}^{\nu}(1-\beta_i\beta_k^{-1}). \tag{7.32}$$

From (7.30) and (7.32), we find that the error value $\delta_k$ at location $\beta_k$ is given by

$$\delta_k = \frac{-\mathbb{Z}_0(\beta_k^{-1})}{\sigma'(\beta_k^{-1})}. \tag{7.33}$$

This expression was derived by Forney [4]. The polynomial $\mathbb{Z}_0(X)$ is called the *error-value evaluator*.

A slightly different error-value evaluator is defined as

$$\begin{aligned}\mathbb{Z}(X) &\triangleq \sigma(X) + X\mathbb{Z}_0(X) \\ &= 1 + (S_1+\sigma_1)X + (S_2+\sigma_1 S_1+\sigma_2)X^2 \\ &\quad + \cdots + (S_\nu + \sigma_1 S_{\nu-1} + \cdots + \sigma_\nu)X^\nu.\end{aligned} \tag{7.34}$$

Then,

$$\delta_k = \frac{-\mathbb{Z}(\beta_k^{-1})}{\prod_{i=1,i\neq k}^{\nu}(1-\beta_i\beta_k^{-1})}. \tag{7.35}$$

The expression for evaluating $\delta_k$ was derived by Berlekamp [5].

## EXAMPLE 7.3

Consider the triple-error-correcting RS code of length 15 given in Example 7.2, where we assumed that the all-zero codeword was transmitted, and $\mathbf{r} = (0\,0\,0\,\alpha^7\,0\,0\,\alpha^3\,0\,0\,0\,0\,0\,\alpha^4\,0\,0)$ was received. Using the Berlekamp algorithm, we find that

$$\sigma(X) = 1 + \alpha^7 X + \alpha^4 X^2 + \alpha^6 X^3,$$

and error-location numbers are $\alpha^{12}$, $\alpha^6$, and $\alpha^3$. The errors occur at $X^3$, $X^6$, and $X^{12}$. Now, we are ready to evaluate the error values at these positions. From (7.29) we find that

$$\begin{aligned}\mathbb{Z}_0(X) &= S_1 + (S_2+\sigma_1 S_1)X + (S_3+\sigma_1 S_2+\sigma_2 S_1)X^2 \\ &= \alpha^{12} + (1+\alpha^7\alpha^{12})X + (\alpha^{14}+\alpha^7+\alpha^4\alpha^{12})X^2 \\ &= \alpha^{12} + (1+\alpha^4)X + (\alpha^{14}+\alpha^7+\alpha)X^2 \\ &= \alpha^{12} + \alpha X.\end{aligned}$$

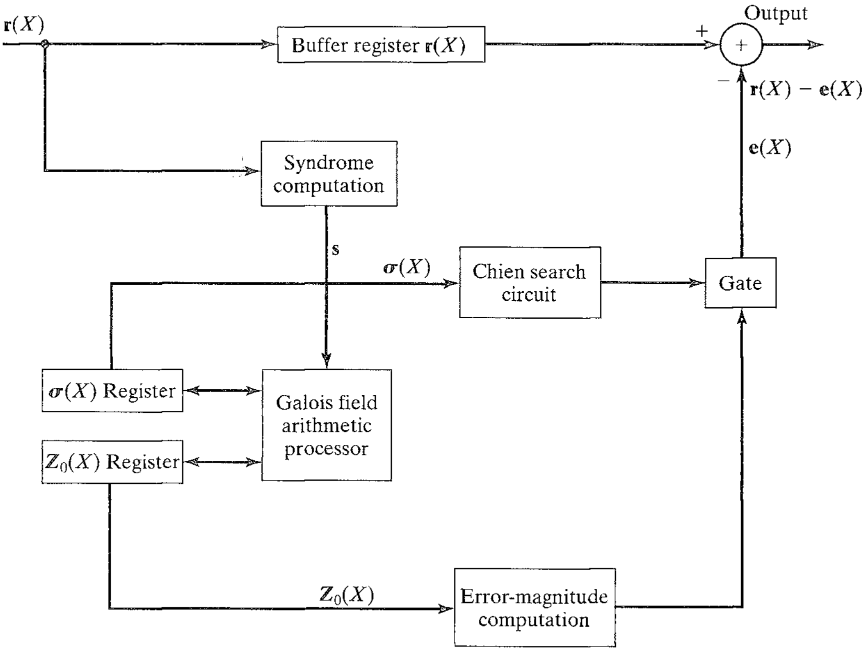


FIGURE 7.2: A general organization of a $q$-ary BCH decoder.

(The computations are carried out in $GF(2^4)$ given by Table 2.8.) Using (7.33), we obtain the error values at locations $X^3$, $X^6$, and $X^{12}$:

$$e_3 = \frac{-\mathbf{Z}_0(\alpha^{-3})}{\sigma'(\alpha^{-3})} = \frac{\alpha^{12} + \alpha\alpha^{-3}}{\alpha^3(1+\alpha^6\alpha^{-3})(1+\alpha^{12}\alpha^{-3})} = \frac{\alpha}{\alpha^9} = \alpha^7,$$

$$e_6 = \frac{-\mathbf{Z}_0(\alpha^{-6})}{\sigma'(\alpha^{-6})} = \frac{\alpha^{12} + \alpha\alpha^{-6}}{\alpha^6(1+\alpha^3\alpha^{-6})(1+\alpha^{12}\alpha^{-6})} = \frac{\alpha^3}{1} = \alpha^3,$$

$$e_{12} = \frac{-\mathbf{Z}_0(\alpha^{-12})}{\sigma'(\alpha^{-12})} = \frac{\alpha^{12} + \alpha\alpha^{-12}}{\alpha^{12}(1+\alpha^3\alpha^{-12})(1+\alpha^6\alpha^{-12})} = \frac{\alpha^6}{\alpha^2} = \alpha^4.$$

Thus, the error pattern is

$$\mathbf{e}(X) = \alpha^7X^3 + \alpha^3X^6 + \alpha^4X^{12},$$

which is exactly the difference between the received vector and the transmitted vector. The decoding is completed by taking $\mathbf{r}(X) - \mathbf{e}(X)$.

---

A general organization of a BCH decoder is shown in Figure 7.2.

## 7.5 DECODING WITH THE EUCLIDEAN ALGORITHM

In the expansion of $\sigma(X)\mathbf{S}(X)$ ((7.26)), only the coefficients of the first $2t$ terms ($X^0$ to $X^{2t-1}$) are known. Let

$$[\sigma(X)\mathbf{S}(X)]_{2t}$$

denote the first $2t$ terms of $\sigma(X)\mathbb{S}(X)$. Then,

$$\sigma(X)\mathbb{S}(X) - [\sigma(X)\mathbb{S}(X)]_{2t}$$

is divisible by $X^{2t}$. This simply says that if $\sigma(X)\mathbb{S}(X)$ is divided by $X^{2t}$, the remainder is $[\sigma(X)\mathbb{S}(X)]_{2t}$. Mathematically, this statement is expressed as follows:

$$\sigma(X)\mathbb{S}(X) \equiv [\sigma(X)\mathbb{S}(X)]_{2t} \bmod X^{2t}. \tag{7.36}$$

In fact,

$$\mathbb{Z}_0(X) = [\sigma(X)\mathbb{S}(X)]_{2t}. \tag{7.37}$$

Therefore, we have

$$\sigma(X)\mathbb{S}(X) \equiv \mathbb{Z}_0(X) \bmod X^{2t} \tag{7.38}$$

which is called the *key equation* in decoding of BCH codes [5].

Any method of solving the key equation to find $\sigma(X)$ and $\mathbb{Z}_0(X)$ is a decoding method for $q$-ary BCH codes. If the number of errors $\nu$ during the transmission of a code polynomial $\mathbb{v}(X)$ is less than or equal to $t$, (i.e., $\nu \leq t$), then the key equation has a unique pair of solutions, $(\sigma(X), \mathbb{Z}_0(X))$, with

$$\deg \mathbb{Z}_0(X) < \deg \sigma(X) \leq t, \tag{7.39}$$

where $\deg f(X)$ denotes the degree of polynomial $f(X)$. We have already presented Berlekamp's algorithm for solving the key equation, which is a very effective method for practical implementation and has been widely used.

There is another method for solving the key equation that is much easier to understand. This method is based on the Euclidean algorithm for finding the greatest common divisor (GCD) of two polynomials.

Consider two polynomials, $a(X)$ and $b(X)$, over $GF(q)$. Assume that

$$\deg a(X) \geq \deg b(X).$$

Let GCD$[a(X), b(X)]$ denote the greatest common divisor of $a(X)$ and $b(X)$. Then, we can find GCD$[a(X), b(X)]$ by iteratively applying the division algorithm as follows:

$$\begin{aligned}
a(X) &= q_1(X)b(X) + r_1(X)\\
b(X) &= q_2(X)r_1(X) + r_2(X)\\
r_1(X) &= q_3(X)r_2(X) + r_3(X)\\
&\vdots\\
r_{i-2}(X) &= q_i(X)r_{i-1}(X) + r_i(X)\\
&\vdots\\
r_{n-2}(X) &= q_n(X)r_{n-1}(X) + r_n(X)\\
r_{n-1}(X) &= q_{n+1}(X)r_n(X),
\end{aligned} \tag{7.40}$$

where $q_i(X)$ and $r_i(X)$ are the quotient and the remainder, respectively, at the $i$th step of the iterative division. The iteration stops when the remainder is identical to zero. Then, the last nonzero remainder $r_n(X)$ is the GCD of $a(X)$ and $b(X)$ (may be different by a constant scalar $c$); that is,

$$r_n(X) = c\text{GCD}[a(X), b(X)],$$

where $c \in GF(q)$. Note that for $1 \leq i \leq n$,

$$\deg r_{i-1}(X) > \deg r_i(X).$$

From (7.40), it is possible to show that

$$\text{GCD}[a(X), b(X)] = f(X)a(X) + g(X)b(X), \tag{7.41}$$

where $f(X)$ and $g(X)$ are polynomials over $GF(q)$ [Euclid's algorithm].

In fact, we can express the remainder at each division step as follows:

$$\begin{aligned} r_1(X) &= f_1(X)a(X) + g_1(X)b(X) \\ r_2(X) &= f_2(X)a(X) + g_2(X)b(X) \\ &\vdots \\ r_i(X) &= f_i(X)a(X) + g_i(X)b(X) \\ &\vdots \\ r_n(X) &= f_n(X)a(X) + g_n(X)b(X). \end{aligned} \tag{7.42}$$

From (7.41) and (7.42), we have

$$\begin{aligned} f(X) &= c^{-1}f_n(X) \\ g(X) &= c^{-1}g_n(X). \end{aligned} \tag{7.43}$$

From (7.40) and (7.42), we obtain the following recursive equations for finding $r_i(X)$, $f_i(X)$, and $g_i(X)$:

$$\begin{aligned} r_i(X) &= r_{i-2}(X) - q_i(X)r_{i-1}(X) \\ f_i(X) &= f_{i-2}(X) - q_i(X)f_{i-1}(X) \\ g_i(X) &= g_{i-2}(X) - q_i(X)g_{i-1}(X) \end{aligned} \tag{7.44}$$

for $1 \leq i \leq n$. The initial conditions for the recursion are

$$\begin{aligned} r_{-1}(X) &= a(X), \\ r_0(X) &= b(X), \\ f_{-1}(X) &= g_0(X) = 1, \\ f_0(X) &= g_{-1}(X) = 0. \end{aligned} \tag{7.45}$$

TABLE 7.3: Steps for finding the GCD of $X^3 + 1$ $X^2 + 1$ given in Example 7.4.

| $i$ | $r_i(X)$ | $q_i(X)$ | $f_i(X)$ | $g_i(X)$ |
|---|---|---|---|---|
| $-1$ | $X^3+1$ | — | $1$ | $0$ |
| $0$ | $X^2+1$ | — | $0$ | $1$ |
| $1$ | $X+1$ | $X$ | $1$ | $X$ |
| $2$ | $0$ | $X+1$ | $X+1$ | $X^2+X+1$ |

An important property of Euclid's algorithm is

$$\deg a(X) = \deg g_i(X) + \deg r_{i-1}(X). \tag{7.46}$$

We see that as $i$ increases, the degree of $r_{i-1}(X)$ decreases, and the degree of $g_i(X)$ increases. This result will be used for solving the key equation.

EXAMPLE 7.4

Let $a(X) = X^3 + 1$ and $b(X) = X^2 + 1$ be two polynomials over $GF(2)$. Euclid's algorithm for finding the GCD$[X^3 + 1, X^2 + 1]$ is shown in Table 7.3. We see that last nonzero remainder is

$$r_1(X) = X + 1,$$

which is the GCD of $X^3 + 1$ and $X^2 + 1$.

### 7.5.1 Solving the Key Equation [6]

We can express the key equation in the following form:

$$\sigma(X)\mathbb{S}(X) = \mathbb{Q}(X)X^{2t} + \mathbb{Z}_0(X). \tag{7.47}$$

Rearranging (7.47), we have

$$\mathbb{Z}_0(X) = -\mathbb{Q}(X)X^{2t} + \sigma(X)\mathbb{S}(X). \tag{7.48}$$

Setting $a(X) = X^{2t}$ and $b(X) = \mathbb{S}(X)$, we see that (7.48) is exactly in the form given by (7.41). This suggests that $\sigma(X)$ and $\mathbb{Z}_0(X)$ can be found by the Euclidean iterative division algorithm for the two polynomials:

$$\begin{aligned} a(X) &= X^{2t}, \\ b(X) &= \mathbb{S}(X), \end{aligned} \tag{7.49}$$

where

$$\mathbb{S}(X) = S_1 + S_2X + S_3X^2 + \cdots + S_{2t}X^{2t-1}.$$

Let

$$\begin{aligned} \mathbb{Z}_0^{(i)}(X) &= r_i(X) \\ \sigma^{(i)}(X) &= g_i(X) \\ \gamma^{(i)}(X) &= f_i(X). \end{aligned} \tag{7.50}$$

Then, it follows from (7.49) and (7.50) that we can put (7.42), (7.44), and (7.45) in the following forms:

$$\mathbb{Z}_0^{(i)}(X) = \gamma^{(i)}(X)X^{2t} + \sigma^{(i)}(X)\mathbb{S}(X), \tag{7.51}$$

and

$$\begin{aligned} \mathbb{Z}_0^{(i)}(X) &= \mathbb{Z}_0^{(i-2)}(X) - q_i(X)\mathbb{Z}_0^{(i-1)}(X), \\ \sigma^{(i)}(X) &= \sigma^{(i-2)}(X) - q_i(X)\sigma^{(i-1)}(X), \\ \gamma^{(i)}(X) &= \gamma^{(i-2)}(X) - q_i(X)\gamma^{(i-1)}(X), \end{aligned} \tag{7.52}$$

with

$$\begin{aligned} \mathbb{Z}_0^{(-1)}(X) &= X^{2t}, \\ \mathbb{Z}_0^{(0)}(X) &= \mathbb{S}(X), \\ \gamma^{(-1)}(X) &= \sigma^{(0)}(X) = 1, \\ \gamma^{(0)}(X) &= \sigma^{(-1)}(X) = 0. \end{aligned}$$

To find $\sigma(X)$ and $\mathbb{Z}_0(X)$, we carry out the iteration process given by (7.52) as follows: at the $i$th step,

1. We divide $\mathbb{Z}_0^{(i-2)}(X)$ by $\mathbb{Z}_0^{(i-1)}(X)$ to obtain the quotient $q_i(X)$ and the remainder $\mathbb{Z}_0^{(i)}(X)$.
2. We find $\sigma^{(i)}(X)$ from

$$\sigma^{(i)}(X) = \sigma^{(i-2)}(X) - q_i(X)\sigma^{(i-1)}(X).$$

Iteration stops when we reach a step $\rho$ for which

$$\deg \mathbb{Z}_0^{(\rho)}(X) < \deg \sigma^{(\rho)}(X) \le t. \tag{7.53}$$

Then,

$$\begin{aligned} \mathbb{Z}_0(X) &= \mathbb{Z}_0^{(\rho)}(X), \\ \sigma(X) &= \sigma^{(\rho)}(X). \end{aligned}$$

If the number of errors is $t$ or less, there always exists a step $\rho$ for which the condition given by (7.53) holds. It is easy to see that

$$\rho \le 2t.$$

Note that $\sigma(X) = \sigma^{(\rho)}(X)$ found by the foregoing algorithm may be different from $\sigma(X)$ defined by (7.14) by a constant scalar in $GF(q^m)$; however, it gives the same roots.

The iteration process for finding $\sigma(X)$ and $\mathbb{Z}_0(X)$ can be carried out by setting up and filling Table 7.4.

TABLE 7.4: Euclidean's iterative procedure for finding the error-location polynomial and error-value evaluator.

| $i$ | $\mathbb{Z}_0^{(i)}(X)$ | $q_i(X)$ | $\sigma_i(X)$ |
|---|---|---|---|
| $-1$ | $X^{2t}$ | — | 0 |
| 0 | $\mathbb{S}(X)$ | — | 1 |
| 1 | | | |
| 2 | | | |
| ⋮ | | | |
| $i$ | | | |
| ⋮ | | | |

EXAMPLE 7.5

Consider the triple-error-correcting RS code of length $n = 15$ over $GF(2^4)$ whose generator polynomial has $\alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^5$, and $\alpha^6$ as roots; that is,

$$\mathbb{g}(X) = (X + \alpha)(X + \alpha^2)((X + \alpha^3)(X + \alpha^4)(X + \alpha^5)(X + \alpha^6).$$

Suppose the received polynomial is

$$\mathbb{r}(X) = \alpha^7 X^3 + \alpha^{11} X^{10}.$$

The syndrome components are

$$\begin{aligned}
S_1 &= \mathbb{r}(\alpha) = \alpha^{10} + \alpha^{21} = \alpha^7, \\
S_2 &= \mathbb{r}(\alpha^2) = \alpha^{13} + \alpha^{31} = \alpha^{12}, \\
S_3 &= \mathbb{r}(\alpha^3) = \alpha^{16} + \alpha^{41} = \alpha^6, \\
S_4 &= \mathbb{r}(\alpha^4) = \alpha^{19} + \alpha^{51} = \alpha^{12}, \\
S_5 &= \mathbb{r}(\alpha^5) = \alpha^7 + \alpha = \alpha^{14}, \\
S_6 &= \mathbb{r}(\alpha^6) = \alpha^{10} + \alpha^{11} = \alpha^{14}.
\end{aligned}$$

Hence, the syndrome polynomial is

$$\mathbb{S}(X) = \alpha^7 + \alpha^{12} X + \alpha^6 X^2 + \alpha^{12} X^3 + \alpha^{14} X^4 + \alpha^{14} X^5.$$

Using the Euclidean algorithm, we find

$$\begin{aligned}
\sigma(X) &= \alpha^{11} + \alpha^8 X + \alpha^9 X^2 \\
&= \alpha^{11}(1 + \alpha^{12} X + \alpha^{13} X^2),
\end{aligned}$$

and

$$\mathbb{Z}_0(X) = \alpha^3 + \alpha^2 X,$$

as shown in the Table 7.5.

TABLE 7.5: Steps for finding the error-location polynomial and error-value evaluator of the RS code given in Example 7.5.

| $i$ | $\mathbb{Z}_0^{(i)}(X)$ | $q_i(X)$ | $\sigma_i(X)$ |
|---|---|---|---|
| $-1$ | $X^6$ | — | $0$ |
| $0$ | $\mathbb{S}(X)$ | — | $1$ |
| $1$ | $\alpha^8+\alpha^3X+\alpha^5X^2+\alpha^5X^3+\alpha^6X^4$ | $\alpha+\alpha X$ | $\alpha+\alpha X$ |
| $2$ | $\alpha^3+\alpha^2X$ | $\alpha^{11}+\alpha^8X$ | $\alpha^{11}+\alpha^8X+\alpha^9X^2$ |

From $\sigma(X)$, we find that the roots are $\alpha^5$ and $\alpha^{12}$. Hence, the error location numbers are $\alpha^{10}$ and $\alpha^3$. The error values at these locations are

$$e_3=\frac{-\mathbb{Z}_0(\alpha^{-3})}{\sigma'(\alpha^{-3})}=\frac{\alpha^3+\alpha^2\alpha^{-3}}{\alpha^{11}\cdot\alpha^3(1+\alpha^{10}\alpha^{-3})}=\frac{1}{\alpha^8}=\alpha^7,$$

$$e_{10}=\frac{-\mathbb{Z}_0(\alpha^{-10})}{\sigma'(\alpha^{-10})}=\frac{\alpha^3+\alpha^2\cdot\alpha^{-10}}{\alpha^{11}\cdot\alpha^{10}(1+\alpha^3\alpha^{-10})}=\frac{\alpha^4}{\alpha^8}=\alpha^{11}.$$

Therefore, the error polynomial is $\mathbb{e}(X)=\alpha^7X^3+\alpha^{11}X^{10}$, and the decoded codeword $\mathbb{v}(X)=\mathbb{r}(X)-\mathbb{e}(X)$ is the all-zero codeword.

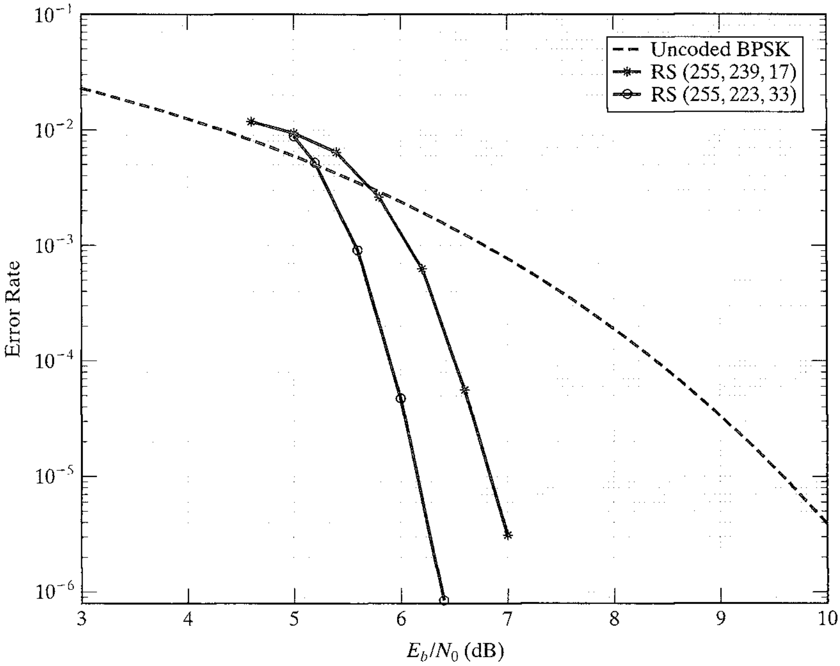


FIGURE 7.3: Error performances for (255, 223, 33) and (255, 239, 17) RS codes.

The two most commonly used RS codes for error control in data communication and storage systems are the (255, 223, 33) and the (255, 239, 17) RS codes over $GF(2^8)$. The (255, 223, 33) RS code is also a NASA standard code for space and satellite communications. Both codes are decoded with either the Berlekamp algorithm or the Euclidean algorithm. Their error performances with BPSK transmission over the AWGN channel are shown in Figure 7.3.

## 7.6 FREQUENCY-DOMAIN DECODING

So far, BCH and RS codes have been decoded in the time domain; however, these codes also can be decoded in the frequency domain. In this section, we first give a spectral description of these codes and then present a frequency-domain decoding algorithm for them [8, 17].

Consider the Galois field $GF(q)$ with characteristic $p$ (see Section 2.2). Let 1 be the unit element of $GF(q)$. Then, $p$ is the smallest positive integer such that the sum

$$\sum_{i=1}^{p} 1 = \underbrace{1 + 1 + \ldots + 1}_{p} = 0.$$

For any nonnegative integer $n$, the sum

$$\sum_{i=1}^{n} 1 = \underbrace{1 + 1 + \ldots + 1}_{n} = \lambda,$$

where $\lambda$ is the remainder resulting from dividing $n$ by $p$. Mathematically, we write

$$\lambda = n(\text{modulo} \;\; p). \tag{7.54}$$

Note that $\lambda$ is an element in $GF(q)$.

Let $\mathbf{v}(X) = v_0 + v_1X + \ldots + v_{n-1}X^{n-1}$ be a polynomial over $GF(q)$, where $n$ divides $q^m - 1$, and $n \neq 1$. Let $\alpha$ be an element of order $n$ in $GF(q^m)$. Then, $\alpha^n = 1$ and is a root of $X^n - 1$. The Galois field Fourier transform of $\mathbf{v}(X)$ is defined as the polynomial

$$\mathbb{V}(X) = V_0 + V_1X + \ldots + V_{n-1}X^{n-1} \tag{7.55}$$

over $GF(q^m)$, where for $0 \leq j < n$,

$$V_j = \mathbf{v}(\alpha^j) = \sum_{i=0}^{n-1} v_i\alpha^{ij}. \tag{7.56}$$

The coefficient $V_j$ is called the $j$th *spectral component* of $\mathbb{V}(X)$.

Given $\mathbb{V}(X)$, the polynomial $\mathbf{v}(X)$ can be determined by taking the inverse Fourier transform of $\mathbb{V}(X)$, as shown in Theorem 7.1.

**Theorem 7.1** Let $\mathbb{V}(X) = V_0 + V_1X + \ldots + V_{n-1}X^{n-1}$ be the Galois field Fourier transform of $\mathbf{v}(X) = v_0 + v_1X + \ldots + v_{n-1}X^{n-1}$. Then,

$$\begin{aligned} v_i &= \frac{1}{n(\text{modulo } p)}\mathbb{V}(\alpha^{-i}) \\ &= \frac{1}{n(\text{modulo } p)}\sum_{j=0}^{n-1} V_j\alpha^{-ij}. \end{aligned} \tag{7.57}$$

***Proof.*** Factor $X^n - 1$ as

$$X^n - 1 = (X-1)(X^{n-1} + X^{n-2} + \ldots + X + 1).$$

Because $\alpha$ is a root of $X^n - 1$, and $\alpha \neq 1$, $\alpha$ must be a root of $X^{n-1} + X^{n-2} + \ldots + X + 1$; that is,

$$1 + \alpha + \ldots + \alpha^{n-2} + \alpha^{n-1} = 0.$$

For any integer $r$ that is not a multiple of $n$ (i.e., $r \neq 0(\text{modulo } p)$), $(\alpha^r)^n = 1$, and hence $\alpha^r$ is also a root of $X^n - 1$. Since $\alpha^r \neq 1$, $\alpha^r$ must be a root of $X^{n-1} + X^{n-2} + \ldots + X + 1$, and hence

$$\sum_{j=0}^{n-1} \alpha^{rj} = 0. \tag{7.58}$$

Now, consider $\mathbb{V}(\alpha^{-i})$. It follows from (7.55) and (7.56) that

$$\begin{aligned} \mathbb{V}(\alpha^{-i}) &= \sum_{j=0}^{n-1}\alpha^{-ij}\sum_{l=0}^{n-1} v_l\alpha^{lj} \\ &= \sum_{l=0}^{n-1} v_l \sum_{j=0}^{n-1}\alpha^{(l-i)j}. \end{aligned} \tag{7.59}$$

It follows from (7.58) that for $l \neq i$, the second sum on the right side of (7.59) is equal to 0. For $l = i$, the second sum becomes $1 + 1 + \ldots + 1 = n(\text{modulo } p)$. Consequently, (7.59) becomes

$$\mathbb{V}(\alpha^{-i}) = v_i \cdot n(\text{modulo } p). \tag{7.60}$$

Note that $p$ divides $q$ and does not divide $q^m - 1$. Because $n$ is a factor of $q^m - 1$, then $n$ and $p$ are relatively prime. Therefore, $n(\text{modulo } p) \neq 0$. It then follows from (7.60) that

$$v_i = \frac{1}{n(\text{modulo } p)}\mathbb{V}(\alpha^{-i}).$$

This completes the proof. **Q.E.D.**

The polynomials $\mathbf{v}(X)$ and $\mathbb{V}(X)$ form a *transform pair*. $\mathbb{V}(X)$ is the *spectrum polynomial* (or simply spectrum) of $\mathbf{v}(X)$. From (7.56) and (7.57), we readily see that the transform pair have the following properties:

1. The $j$th spectral component $V_j$ is zero if and only if $\alpha^j$ is a root of $\mathbf{v}(X)$.
2. The $i$th component of $\mathbf{v}(X)$ is zero if only if $\alpha^{-i}$ is a root of $\mathbb{V}(X)$.

Now, we are ready to characterize BCH and RS codes in the frequency domain. Consider a primitive $q$-ary $t$-error-correcting BCH code of length $n = q^m - 1$ whose generator polynomial $\mathbf{g}(X)$ has $\alpha, \alpha^2, \ldots, \alpha^{2t}$ as roots. Recall that a polynomial $\mathbf{v}(X)$ of degree $n-1$ or less over $GF(q)$ is a code polynomial if and only if $\mathbf{v}(X)$ is divisible by $\mathbf{g}(X)$. This is equivalent to saying that $\mathbf{v}(X)$ is a code polynomial if and only if $\mathbf{v}(X)$ has $\alpha, \alpha^2, \ldots, \alpha^{2t}$ as roots. Let $\mathbf{v}(X) = v_0 + v_1X + \ldots + v_{n-1}X^{n-1}$ be a code polynomial in a primitive $q$-ary $t$-error-correcting BCH code of length $n = q^m - 1$, and let $\mathbb{V}(X) = V_0 + V_1X + \ldots + V_{n-1}X^{n-1}$ be its Fourier transform. It follows from the first property of the transform pair $(\mathbf{v}(X), \mathbb{V}(X))$ that the $2t$ consecutive spectral components of $\mathbb{V}(X)$ from position $X$ to position $X^{2t}$ are zero; that is, $V_1 = V_2 = \ldots = V_{2t} = 0$. Consequently, a primitive $q$-ary $t$-error-correcting BCH code of length $n = q^m - 1$ is the set of polynomials of degree $n-1$ or less over $GF(q)$ whose Fourier transforms have $2t$ consecutive zero spectral components from position $X$ to position $X^{2t}$. This description is a frequency-domain characterization of a BCH code.

For a $q$-ary RS code, both $\mathbf{v}(X)$ and its Fourier transform $\mathbb{V}(X)$ are polynomials over $GF(q)$. In the frequency domain, a $t$-error-correcting RS code with symbols from $GF(q)$ consists of all the polynomials

$$\mathbb{V}(X) = V_0 + V_1X + \ldots + V_{n-1}X^{n-1}$$

of degree of $n-1$ or less for which

$$V_1 = V_2 = \ldots = V_{2t} = 0.$$

## EXAMPLE 7.6

Again, we consider the triple-error-correcting RS code of length 15 over $GF(2^4)$ with generator polynomial

$$\begin{aligned}\mathbf{g}(X) &= (X+\alpha)(X+\alpha^2)(X+\alpha^3)(X+\alpha^4)(X+\alpha^5)(X+\alpha^6)\\ &= \alpha^6 + \alpha^9X + \alpha^6X^2 + \alpha^4X^3 + \alpha^{14}X^4 + \alpha^{10}X^5 + X^6.\end{aligned}$$

Substituting $X$ with $\alpha^i$ for $0 \leq i < 15$ in $\mathbf{g}(X)$, we obtain the Fourier transform of $\mathbf{g}(X)$:

$$\mathbb{G}(X) = \alpha^5 + \alpha^{11}X^7 + \alpha X^8 + \alpha^{10}X^9 + \alpha^3X^{10} + \alpha^7X^{11} + \alpha^9X^{12} + \alpha^7X^{13} + \alpha^4X^{14}.$$

(Table 2.8 for $GF(2^4)$ is used for computations.) Because $\mathbf{g}(X)$ has $\alpha$ to $\alpha^6$ as roots, $\mathbb{G}(X)$ has zero spectral components from $X^1$ to $X^6$; that is, $G_1 = G_2 = \ldots = G_6 = 0$. Now, consider the code polynomial

$$\mathbf{v}(X) = (X^8 + X + 1)\mathbf{g}(X), \tag{7.61}$$

which has $\alpha$ to $\alpha^6$ and $\alpha^{10}$ as roots. The Fourier transform of $\mathbf{v}(X)$ is

$$\mathbf{V}(X) = \alpha^5 + \alpha^{13}X^7 + \alpha^{11}X^8 + \alpha^{12}X^9 + \alpha^8X^{11} + \alpha^{10}X^{12} + X^{13} + \alpha^8X^{14}.$$

We see that $\mathbf{V}(X)$ has seven zero spectral components at locations $X^1$ to $X^6$ and $X^{10}$.

Before we discuss decoding of BCH and RS codes in the frequency domain, we present an important property of Galois field Fourier transforms. Let

$$\mathbf{a}(X) = a_0 + a_1X + \ldots + a_{n-1}X^{n-1},$$
$$\mathbf{b}(X) = b_0 + b_1X + \ldots + b_{n-1}X^{n-1}$$

be two polynomials over $GF(q)$. We define the following product of $\mathbf{a}(X)$ and $\mathbf{b}(X)$:

$$\begin{aligned}\mathbf{c}(X) &\triangleq \mathbf{a}(X)\cdot\mathbf{b}(X)\\ &= c_0 + c_1X + \ldots + c_{n-1}X^{n-1},\end{aligned}$$

where $c_i = a_i \cdot b_i$ for $0 \leq i < n$. Let

$$\mathbf{A}(X) = A_0 + A_1X + \ldots + A_{n-1}X^{n-1}$$

and

$$\mathbf{B}(X) = B_0 + B_1X + \ldots + B_{n-1}X^{n-1}$$

be the Fourier transforms of $\mathbf{a}(X)$ and $\mathbf{b}(X)$, respectively. Then, the Fourier transform of the product polynomial $\mathbf{c}(X) = \mathbf{a}(X)\cdot\mathbf{b}(X)$ is the convolution of $\mathbf{A}(X)$ and $\mathbf{B}(X)$ given in Theorem 7.2.

**Theorem 7.2** The Fourier transform of $\mathbf{c}(X) = \mathbf{a}(X)\cdot\mathbf{b}(X)$ is given by

$$\mathbf{C}(X) = C_0 + C_1X + \ldots + C_{n-1}X^{n-1},$$

where for $0 \leq j < n$,

$$C_j = \frac{1}{n(\text{modulo } p)}\sum_{k=0}^{n-1} A_kB_{j-k}. \tag{7.62}$$

***Proof.*** Taking the Fourier transform of $\mathbf{c}(X)$, we have

$$C_j = \sum_{i=0}^{n-1} c_i\alpha^{ij} = \sum_{i=0}^{n-1} a_ib_i\alpha^{ij}. \tag{7.63}$$

Expressing $a_i$ in terms of the inverse transform of $\mathbf{A}(X)$, we have

$$a_i = \frac{1}{n(\text{modulo } p)}\sum_{k=0}^{n-1} A_k\alpha^{-ik}. \tag{7.64}$$

Combining (7.63) and (7.64), we have

$$C_j = \frac{1}{n(\text{modulo } p)} \sum_{k=0}^{n-1} A_k \sum_{i=0}^{n-1} b_i \alpha^{i(j-k)}; \tag{7.65}$$

however,

$$\sum_{i=0}^{n-1} b_i \alpha^{i(j-k)} = B_{j-k}. \tag{7.66}$$

From (7.65) and (7.66), we have

$$C_j = \frac{1}{n(\text{modulo } p)} \sum_{k=0}^{n-1} A_k B_{j-k}. \tag{7.67}$$

This completes the proof. Q.E.D.

Now, we consider decoding of BCH and RS codes in the frequency domain. Let $\mathbb{r}(X) = r_0 + r_1 X + \ldots + r_{n-1} X^{n-1}$ be the received polynomial, where $n = q^m - 1$. Then, $\mathbb{r}(X) = \mathbb{v}(X) + \mathbb{e}(X)$, where $\mathbb{v}(X)$ and $\mathbb{e}(X)$ are the transmitted code polynomial and the error polynomial, respectively. The Fourier transform of $\mathbb{r}(X)$ is

$$\mathbb{R}(X) = R_0 + R_1 X + \ldots + R_{n-1} X^{n-1},$$

where

$$R_j = \mathbb{r}(\alpha^j) = \sum_{i=0}^{n-1} r_i \alpha^{ij}. \tag{7.68}$$

Let $\mathbb{V}(X) = V_0 + V_1 X + \ldots + V_{n-1} X^{n-1}$, and $\mathbb{E}(X) = E_0 + E_1 X + \ldots + E_{n-1} X^{n-1}$ be the Fourier transforms of $\mathbb{v}(X)$ and $\mathbb{e}(X)$, respectively. Then,

$$\mathbb{R}(X) = \mathbb{V}(X) + \mathbb{E}(X),$$

with

$$R_j = V_j + E_j \tag{7.69}$$

for $0 \leq j < n$. Because $\mathbb{v}(X)$ is a code polynomial that has $\alpha, \alpha^2, \ldots, \alpha^{2t}$ as roots, then

$$V_j = 0 \tag{7.70}$$

for $1 \leq j \leq 2t$. From (7.69) and (7.70), we find that for $1 \leq j \leq 2t$,

$$R_j = E_j. \tag{7.71}$$

Let $\mathbb{S} = (S_1, S_2, \ldots, S_{2t})$ be the syndrome of $\mathbb{r}(X)$. Then, for $1 \leq j \leq 2t$,

$$S_j = \mathbb{r}(\alpha^j). \tag{7.72}$$

It follows from (7.68), (7.71), and (7.72) that

$$R_j = E_j = S_j = \mathbf{r}(\alpha^j) \tag{7.73}$$

for $1 \leq j \leq 2t$. This result says that the $2t$ spectral components $R_1, R_2, \ldots, R_{2t}$ of $\mathbf{R}(X)$ are the $2t$ syndrome components and are equal to the $2t$ spectral components $E_1, E_2, \ldots, E_{2t}$ of the Fourier transform $\mathbf{E}(X)$ of the error polynomial $\mathbf{e}(X)$. If we can determine the spectral components $E_0, E_{2t+1}, \ldots, E_{n-1}$, then $\mathbf{E}(X)$ is determined, and the inverse transform of $\mathbf{E}(X)$ gives the error polynomial $\mathbf{e}(X)$. Decoding is accomplished by subtracting $\mathbf{e}(X)$ from $\mathbf{r}(X)$.

Suppose there are $\nu \leq t$ errors, and

$$\mathbf{e}(X) = e_{j_1}X^{j_1} + e_{j_2}X^{j_2} + \ldots + e_{j_\nu}X^{j_\nu}. \tag{7.74}$$

The error-location numbers are then $\alpha^{j_1}, \alpha^{j_2}, \ldots, \alpha^{j_\nu}$. The error-location polynomial is

$$\begin{aligned}\sigma(X) &= (1 - \alpha^{j_1}X)(1 - \alpha^{j_2}X)\ldots(1 - \alpha^{j_\nu}X).\\ &= \sigma_0 + \sigma_1 X + \ldots + \sigma_\nu X^\nu,\end{aligned}$$

which has $\alpha^{-j_1}, \alpha^{-j_2}, \ldots, \alpha^{-j_\nu}$ as roots. Hence, for $1 \leq i \leq \nu$,

$$\sigma(\alpha^{-j_i}) = 0. \tag{7.75}$$

Note that $\sigma(X)$ is a polynomial over $GF(q^m)$. We may regard $\sigma(X)$ as the Fourier transform of a polynomial

$$\lambda(X) = \lambda_0 + \lambda_1 X + \ldots + \lambda_{n-1}X^{n-1}$$

over $GF(q)$, where

$$\lambda_j = \frac{1}{n(\text{modulo } p)}\sigma(\alpha^{-j}) \tag{7.76}$$

for $0 \leq j < n$. It follows from (7.75) and (7.76) that for $1 \leq i \leq \nu$,

$$\lambda_{j_i} = 0. \tag{7.77}$$

Consider the product $\lambda(X) \cdot \mathbf{e}(X) = \sum_{j=0}^{n-1} \lambda_j \cdot e_j X$ as defined earlier in this section. From (7.74) and(7.77), we readily see that

$$\lambda(X) \cdot \mathbf{e}(X) = 0; \tag{7.78}$$

that is, $\lambda_j \cdot e_j = 0$ for $0 \leq j < n - 1$. Taking the Fourier transform of $\lambda(X) \cdot \mathbf{e}(X)$ and using (7.62; Theorem 7.2) and (7.78), we have

$$\sum_{k=0}^{n-1} \sigma_k E_{j-k} = 0 \tag{7.79}$$

for $0 \leq j < n$. Since the degree of $\sigma(X)$ is $\nu$, $\sigma_k = 0$ for $k > \nu$. Then, (7.79) becomes

$$\sigma_0 E_j + \sigma_1 E_{j-1} + \ldots + \sigma_\nu E_{j-\nu} = 0 \tag{7.80}$$

for $0 \le j < n-1$. Because $\sigma_0 = 1$, the preceding equation can be put in the following form:

$$E_j = -(\sigma_1 E_{j-1} + \sigma_2 E_{j-2} + \ldots + \sigma_\nu E_{j-\nu}) \tag{7.81}$$

for $0 \le j < n$. Since $E_1, E_2, \ldots, E_{2t}$ are already known (see (7.73)), it follows from (7.81) that we obtain the following recursive equation for computing $E_{2t+1}$ to $E_{n-1}$:

$$E_{l+t} = -(\sigma_1 E_{l+t-1} + \sigma_2 E_{l+t-2} + \ldots + \sigma_\nu E_{l+t-\nu}) \tag{7.82}$$

for $t+1 \le l \le n-1-t$. Setting $j = \nu$ in (7.81), we obtain

$$E_\nu = -(\sigma_1 E_{\nu-1} + \sigma_2 E_{\nu-2} + \ldots + \sigma_\nu E_0).$$

From this equation, we find that

$$E_0 = -\frac{1}{\sigma_\nu}(E_\nu + \sigma_1 E_{\nu-1} + \ldots + \sigma_{\nu-1} E_1). \tag{7.83}$$

From (7.82) and (7.83), we can determine the entire $\mathbb{E}(X)$. Taking the inverse transform of $\mathbb{V}(X) = \mathbb{R}(X) - \mathbb{E}(X)$, we obtain the decoded code polynomial $\mathbb{v}(X)$, which completes the decoding.

The error-location polynomial can be computed by using the Berlekamp iterative algorithm. The decoding consists of the following steps:

1. Take the Fourier transform $\mathbb{R}(X)$ of $\mathbb{r}(X)$.
2. Find $\sigma(X)$.
3. Compute $\mathbb{E}(X)$.
4. Take the inverse transform $\mathbb{v}(X)$ of $\mathbb{V}(X) = \mathbb{R}(X) - \mathbb{E}(X)$.

A transform decoder is depicted in Figure 7.4.

### EXAMPLE 7.7

Again, consider the (15, 9) RS code over $GF(2^4)$ given in Example 7.2. Suppose a code polynomial $\mathbb{v}(X)$ is transmitted, and $\mathbb{r}(X) = \alpha^7 X^3 + \alpha^3 X^6 + \alpha^4 X^{12}$ is received. The Fourier transform of $\mathbb{r}(X)$ is

$$\begin{aligned}\mathbb{R}(X) = {} & \alpha^{12} X + X^2 + \alpha^{14} X^3 + \alpha^{10} X^4 + \alpha^{12} X^6 \\ & + X^7 + \alpha^{14} X^8 + \alpha^{10} X^9 + \alpha^{12} X^{11} \\ & + X^{12} + \alpha^{14} X^{13} + \alpha^{10} X^{14}.\end{aligned}$$

The coefficients of powers $X$ to $X^6$ give the syndrome components; that is, $S_1 = \alpha^{12}$, $S_2 = 1$, $S_3 = \alpha^{14}$, $S_4 = \alpha^{10}$, $S_5 = 0$, and $S_6 = \alpha^{12}$. They are also the spectral components $E_1, E_2, E_3, E_4, E_5$, and $E_6$ of the error spectral polynomial $\mathbb{E}(X)$. Using the Berlekamp algorithm based on the syndrome $(S_1, S_2, S_3, S_4, S_5, S_6)$, we find the error-location polynomial

$$\sigma(X) = 1 + \alpha^7 X + \alpha^4 X^2 + \alpha^6 X^3$$

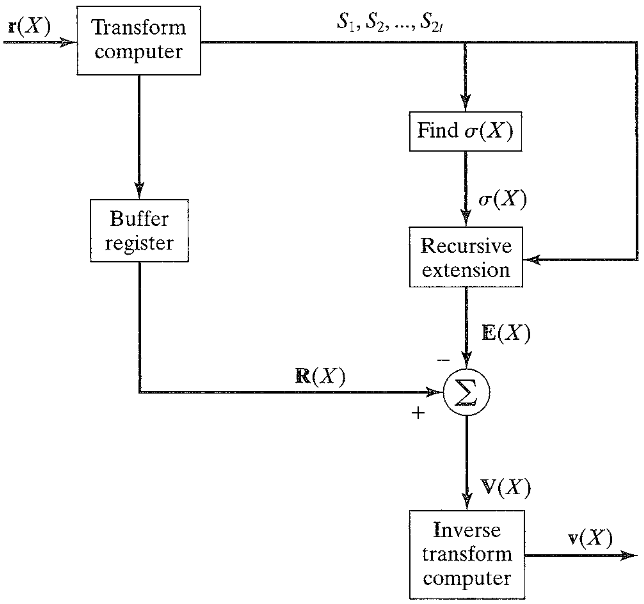


FIGURE 7.4: A transform decoder for a $q$-ary BCH or RS code.

(see Example 7.2). From (7.82), we obtain the following recursion equation for computing spectral components $E_7$ to $E_{14}$ of $\mathbf{E}(X)$:

$$\begin{aligned} E_{l+3} &= \sigma_1 E_{l+2} + \sigma_2 E_{l+1} + \sigma_3 E_l \\ &= \alpha^7 E_{l+2} + \alpha^4 E_{l+1} + \alpha^6 E_l \end{aligned}$$

for $4 \leq l \leq 11$. We compute the spectral component $E_0$ from

$$\begin{aligned} E_0 &= \frac{1}{\sigma_3}(E_3 + \sigma_1 E_2 + \sigma_2 E_1) \\ &= \alpha^{-6}(E_3 + \alpha^7 E_2 + \alpha^4 E_1) \\ &= \alpha^{-6}(\alpha^{14} + \alpha^7 + \alpha^{16}) \\ &= 0. \end{aligned}$$

The resultant error spectral polynomial is

$$\begin{aligned} \mathbf{E}(X) = \alpha^{12}X + X^2 + \alpha^{14}X^3 + \alpha^{10}X^4 + \alpha^{12}X^6 \\ +X^7 + \alpha^{14}X^8 + \alpha^{10}X^9 + \alpha^{12}X^{11} \\ +X^{12} + \alpha^{14}X^{13} + \alpha^{10}X^{14}. \end{aligned}$$

We find that $\mathbf{E}(X) = \mathbf{R}(X)$, and $\mathbf{V}(X) = 0$. Therefore, the decoded codeword is the all-zero codeword. The inverse transform of $\mathbf{E}(X)$ is $\mathbf{e}(X) = \alpha^7 X^3 + \alpha^3 X^6 + \alpha^4 X^{12}$. This is exactly the same result as given in Example 7.2.

## 7.7 CORRECTION OF ERRORS AND ERASURES

A $q$-ary $t$-error-correcting BCH (or RS) code can be used to correct all combinations of $\nu$ symbol errors and $e$ symbol erasures provided that the inequality

$$\nu + e/2 \leq t \tag{7.84}$$

holds. Each of the decoding algorithms presented in the last three sections can be modified to do the job.

Suppose the received polynomial $\mathbb{r}(X)$ contains $\nu$ symbol errors at positions $X^{i_1}, X^{i_2}, \cdots, X^{i_\nu}$, and $e$ symbol erasures at positions $X^{j_1}, X^{j_2}, \cdots, X^{j_e}$. Because the erased positions are known, decoding is to find the locations and values of the errors and the values of the erased symbols. The erasure-location numbers corresponding to the erased positions $X^{j_1}, X^{j_2}, \cdots, X^{j_e}$ are $\alpha^{j_1}, \alpha^{j_2}, \cdots, \alpha^{j_e}$. We form the erasure-location polynomial:

$$\beta(X) \triangleq \prod_{l=1}^{e}(1 - \alpha^{j_l}X). \tag{7.85}$$

Now, we fill the $e$ erased positions in $\mathbb{r}(X)$ with zeros (or arbitrary symbols from $GF(q)$). This substitution of $e$ zeros into the erased positions in $\mathbb{r}(X)$ can introduce up to $e$ additional errors. Let $\mathbb{r}^*(X)$ denote the modified received polynomial. Let

$$\sigma(X) \triangleq \prod_{k=1}^{\nu}(1 - \alpha^{i_k}X) \tag{7.86}$$

be the error-location polynomial for the errors in $\mathbb{r}(X)$ at positions $X^{i_1}, X^{i_2}, \cdots, X^{i_\nu}$. Then, the error-location polynomial for the modified received polynomial $\mathbb{r}^*(X)$ is

$$\gamma(X) = \sigma(X)\beta(X), \tag{7.87}$$

for which $\beta(X)$ is known. Now, decoding is to find $\sigma(X)$ and the error-value evaluator $\mathbb{Z}_0(X)$ for $\mathbb{r}^*(X)$.

We compute the syndrome polynomial

$$\mathbb{S}(X) = S_1 + S_2X + \cdots + S_{2t}X^{2t-1}$$

from the modified received polynomial $\mathbb{r}^*(X)$. Then, the key equation becomes

$$\sigma(X)\beta(X)\mathbb{S}(X) \equiv \mathbb{Z}_0(X) \text{mod } X^{2t}. \tag{7.88}$$

The decoding problem is to find the solution $(\sigma(X), \mathbb{Z}_0(X))$ of this equation such that $\sigma(X)$ has minimum degree $\nu$, and deg $\mathbb{Z}_0(X) < \nu + e$. Since $\beta(X)$ and $\mathbb{S}(X)$ are known, we can combine them. Let

$$\begin{aligned}\mathbb{T}(X) &\triangleq [\beta(X)\mathbb{S}(X)]_{2t} \\ &= T_1 + T_2X + \cdots + T_{2t}X^{2t-1}\end{aligned} \tag{7.89}$$

denote the polynomial that consists of the $2t$ terms of $\boldsymbol{\beta}(X)\mathbb{S}(X)$ from $X^0$ to $X^{2t-1}$. Then, we can write the key equation of (7.88) as

$$\sigma(X)\mathbb{T}(X) \equiv \mathbb{Z}_0(X) \bmod X^{2t}. \tag{7.90}$$

This key equation may be solved by using either Euclid's or Berlekamp's algorithm.

The Euclidean algorithm for error/erasure decoding consists of the following:

1. Compute the erasure-location polynomial $\boldsymbol{\beta}(X)$ using the erasure information from the received polynomial $\mathbb{r}(X)$.
2. Form the modified received polynomial $\mathbb{r}^*(X)$ by replacing the erased symbols with zeros. Compute the syndrome polynomial $\mathbb{S}(X)$ from $\mathbb{r}^*(X)$.
3. Compute the modified syndrome polynomial $\mathbb{T}(X) = [\boldsymbol{\beta}(X)\ \mathbb{S}(X)]_{2t}$.
4. Set the following initial conditions:

$$\mathbb{Z}_0^{(-1)}(X) = X^{2t}, \quad \mathbb{Z}_0^{(0)}(X) = \mathbb{T}(X),$$
$$\sigma^{(-1)}(X) = 0, \quad \text{and} \quad \sigma^{(0)}(X) = 1.$$

5. Execute the Euclidean algorithm iteratively as described in Section 7.5 until a step $\rho$ is reached for which

$$\deg \mathbb{Z}_0^{(\rho)}(X) < \begin{cases} t + e/2, & \text{for even } e, \\ t + (e-1)/2, & \text{for odd } e. \end{cases} \tag{7.91}$$

Then, set $\sigma(X) = \sigma^{(\rho)}(X)$, and $\mathbb{Z}_0(X) = \mathbb{Z}_0^{(\rho)}(X)$.

6. Find the roots of $\sigma(X)$ and determine the error locations in $\mathbb{r}(X)$.
7. Determine the values of errors and erasures from $\mathbb{Z}_0(X)$ and $\boldsymbol{\gamma}(X) = \sigma(X)\boldsymbol{\beta}(X)$. The error values are given by

$$e_{i_k} = \frac{-\mathbb{Z}_0(\alpha^{-i_k})}{\boldsymbol{\gamma}'(\alpha^{-i_k})} \tag{7.92}$$

for $1 \le k \le \nu$, and the values of the erased symbols are given by

$$f_{j_l} = \frac{-\mathbb{Z}_0(\alpha^{-j_l})}{\boldsymbol{\gamma}'(\alpha^{-j_l})} \tag{7.93}$$

for $1 \le l \le e$, where $\boldsymbol{\gamma}'(X)$ is the derivative of the overall error/erasure-location polynomial $\boldsymbol{\gamma}(X) = \sigma(X)\boldsymbol{\beta}(X)$; that is,

$$\begin{aligned} \boldsymbol{\gamma}'(X) &= \frac{d}{dX}\boldsymbol{\gamma}(X) \\ &= -a \sum_{l=1}^{\nu+e} \alpha^{j_l} \prod_{i=1, i \ne l}^{\nu+e} (1 - \alpha^{j_i} X). \end{aligned} \tag{7.94}$$

($a$ is the constant $\ne 1$ that may appear in $\sigma(X)$ when the Euclidean algorithm is used).

## EXAMPLE 7.8

Again consider the triple-error-correcting RS code of length 15 over $GF(2^4)$ generated by $\mathbf{g}(X) = (X+\alpha)(X+\alpha^2)(X+\alpha^3)(X+\alpha^4)(X+\alpha^5)(X+\alpha^6)$. This code is capable of correcting all combinations of two or fewer errors and two or fewer erasures. Suppose the all-zero codeword is transmitted, and the received vector is

$$\mathbf{r} = (0\,0\,0 * 0\,0 * 0\,0\alpha\,0\,0\alpha^4\,0\,0),$$

where $*$ denotes an erasure. The received polynomial is

$$\mathbf{r}(X) = (*)X^3 + (*)X^6 + \alpha X^9 + \alpha^4 X^{12}.$$

Because the erased positions are $X^3$ and $X^6$, the erasure-location polynomial is

$$\begin{aligned}\boldsymbol{\beta}(X) &= (1+\alpha^3 X)(1+\alpha^6 X)\\ &= 1 + \alpha^2 X + \alpha^9 X^2.\end{aligned}$$

Replacing the erased symbols with zeros, we obtain the following modified received polynomial:

$$\mathbf{r}^*(X) = \alpha X^9 + \alpha^4 X^{12}.$$

The syndrome components computed from $\mathbf{r}^*(X)$ are

$$\begin{aligned}S_1 &= \mathbf{r}^*(\alpha) = \alpha^8, & S_4 &= \mathbf{r}^*(\alpha^4) = 0,\\ S_2 &= \mathbf{r}^*(\alpha^2) = \alpha^{11}, & S_5 &= \mathbf{r}^*(\alpha^5) = 1,\\ S_3 &= \mathbf{r}^*(\alpha^3) = \alpha^9, & S_6 &= \mathbf{r}^*(\alpha^6) = \alpha^8.\end{aligned}$$

The syndrome polynomial is then

$$\mathbb{S}(X) = \alpha^8 + \alpha^{11} X + \alpha^9 X^2 + X^4 + \alpha^8 X^5,$$

and the modified syndrome polynomial is

$$\begin{aligned}\mathbb{T}(X) &= [\boldsymbol{\beta}(X)\mathbb{S}(X)]_{2t}\\ &= \alpha^8 + \alpha^{14} X + \alpha^4 X^2 + \alpha^3 X^3 + \alpha^{14} X^4 + X^5.\end{aligned}$$

Using the Euclidean decoding algorithm, we set the initial conditions as follows:

$$\begin{aligned}&\mathbb{Z}_0^{(-1)}(X) = X^6, \quad \mathbb{Z}_0^{(0)}(X) = \mathbb{T}(X),\\ &\sigma^{(-1)}(X) = 0, \quad \text{and} \quad \sigma^{(0)}(X) = 1.\end{aligned}$$

Since $t = 3$ and $e = 2$, the algorithm terminates when $\deg \mathbb{Z}_0(X) < 4$. Executing the algorithm, we obtain Table 7.6. The error-location polynomial is

$$\begin{aligned}\sigma(X) &= \alpha(1 + \alpha^8 X + \alpha^6 X^2)\\ &= \alpha(1 + \alpha^9 X)(1 + \alpha^{12} X).\end{aligned}$$

The two roots of $\sigma(X)$ are $\alpha^{-9}$ and $\alpha^{-12}$. The reciprocals of these two roots give the error locations, $\alpha^9$ and $\alpha^{12}$. The error-value evaluator is

$$\mathbb{Z}_0(X) = \alpha^9 + \alpha^8 X + \alpha X^2 + \alpha X^3.$$

The overall error/erasure-location polynomial is

$$\begin{aligned}\boldsymbol{\gamma}(X) &= \boldsymbol{\sigma}(X)\boldsymbol{\beta}(X)\\ &= \alpha(1+\alpha^3 X)(1+\alpha^6 X)(1+\alpha^9 X)(1+\alpha^{12} X),\end{aligned}$$

and its derivative is

$$\begin{aligned}\gamma'(X) = \alpha^4&(1+\alpha^6 X)(1+\alpha^9 X)(1+\alpha^{12} X)\\ &+\alpha^7(1+\alpha^3 X)(1+\alpha^9 X)(1+\alpha^{12} X)\\ &+\alpha^{10}(1+\alpha^3 X)(1+\alpha^6 X)(1+\alpha^{12} X)\\ &+\alpha^{13}(1+\alpha^3 X)(1+\alpha^6 X)(1+\alpha^9 X).\end{aligned}$$

It follows from (7.92) and (7.93) that the error values at positions $X^9$ and $X^{12}$ are

$$e_9 = \frac{-\mathbb{Z}_0(\alpha^{-9})}{\gamma'(\alpha^{-9})} = \frac{\alpha^{13}}{\alpha^{12}} = \alpha,$$

$$e_{12} = \frac{-\mathbb{Z}_0(\alpha^{-12})}{\gamma'(\alpha^{-12})} = \frac{\alpha^3}{\alpha^{14}} = \alpha^{-11} = \alpha^4,$$

and the values of the erased symbols at positions $X^3$ and $X^6$ are

$$f_3 = \frac{-\mathbb{Z}_0(\alpha^{-3})}{\gamma'(\alpha^{-3})} = \frac{0}{\alpha^8} = 0,$$

$$f_6 = \frac{-\mathbb{Z}_0(\alpha^{-6})}{\gamma'(\alpha^{-6})} = \frac{0}{1} = 0.$$

Then, the estimated error polynomial is

$$\mathbf{e}(X) = \alpha X^9 + \alpha^4 X^{12}.$$

Subtracting $\mathbf{e}(X)$ from $\mathbf{r}^*(X)$, we obtain the decoded code polynomial $\mathbf{v}(X) = \mathbf{0}$, which is the transmitted code polynomial.

---

## EXAMPLE 7.9

Consider the (63, 55, 9) RS code generated by

$$\begin{aligned}\mathbf{g}(X) &= (X+\alpha)(X+\alpha^2)(X+\alpha^3)(X+\alpha^4)(X+\alpha^5)(X+\alpha^6)(X+\alpha^7)(X+\alpha^8)\\ &= X^8 + \alpha^{43}X^7 + \alpha^{59}X^6 + \alpha^{31}X^5 + \alpha^{10}X^4 + \alpha^{40}X^3 + \alpha^{14}X^2 + \alpha^7 X + \alpha^{36}.\end{aligned}$$

TABLE 7.6: Steps for finding the error-location polynomial and error-value evaluator of the RS code given in Example 7.6.

| $i$ | $\mathbb{Z}_0^{(i)}(X)$ | $\mathbb{q}_i(X)$ | $\sigma(X)$ |
|---|---|---|---|
| $-1$ | $X^6$ | — | $0$ |
| $0$ | $\mathbb{T}(X)$ | — | $1$ |
| $1$ | $\alpha^7+\alpha^3X+X^2+\alpha^{10}X^3+\alpha^8X^4$ | $\alpha^{14}+X$ | $\alpha^{14}+X$ |
|  | $\alpha^9+\alpha^8X+\alpha X^2+\alpha X^3$ | $\alpha^5+\alpha^7X$ | $\alpha+\alpha^9X+\alpha^7X^2$ |

This code is capable of correcting all combinations of three or fewer errors and two or fewer erasures. Suppose the all-zero codeword is transmitted, and the received vector is

$$\mathbb{r}=(000000\alpha^{15}0000000000000\alpha^{37}0000000*000$$
$$00\alpha^4000000000000000000*000000000).$$

The received polynomial is

$$\mathbb{r}=\alpha^{15}X^6+\alpha^{37}X^{20}+(*)X^{28}+\alpha^4X^{34}+(*)X^{53}.$$

Because the erased positions are $X^{28}$ and $X^{53}$, the erasure-location polynomial is

$$\begin{aligned}\boldsymbol{\beta}(X)&=(1+\alpha^{28}X)(1+\alpha^{53}X)\\&=1+\alpha^{39}X+\alpha^{18}X^2.\end{aligned}$$

Replacing the erased symbols with zeros, we obtain the following modified received polynomial:

$$\mathbb{r}^*(X)=\alpha^{15}X^6+\alpha^{37}X^{20}+\alpha^4X^{34}.$$

The syndrome components computed from $\mathbb{r}^*(X)$ are

$$\begin{aligned}S_1&=\mathbb{r}^*(\alpha)=\alpha^{19}, & S_5&=\mathbb{r}^*(\alpha^5)=\alpha^{43},\\S_2&=\mathbb{r}^*(\alpha^2)=\alpha, & S_6&=\mathbb{r}^*(\alpha^6)=\alpha^4,\\S_3&=\mathbb{r}^*(\alpha^3)=1, & S_7&=\mathbb{r}^*(\alpha^7)=\alpha^{58},\\S_4&=\mathbb{r}^*(\alpha^4)=\alpha^{22}, & S_8&=\mathbb{r}^*(\alpha^8)=\alpha^{28}.\end{aligned}$$

The syndrome polynomial is then

$$\mathbb{S}(X)=\alpha^{19}+\alpha X+X^2+\alpha^{22}X^3+\alpha^{43}X^4+\alpha^4X^5+\alpha^{58}X^6+\alpha^{28}X^7,$$

and the modified syndrome polynomial is

$$\begin{aligned}\mathbb{T}(X)&=[\boldsymbol{\beta}(X)\mathbb{S}(X)]_{2t}\\&=\alpha^{19}+\alpha^{59}X+\alpha X^2+\alpha^{41}X^3+\alpha^{32}X^4+\alpha^{62}X^5+\alpha^{60}X^6+\alpha^{48}X^7.\end{aligned}$$

Using the Euclidean decoding algorithm, we set the initial condition as follows:

$$\mathbb{Z}_0^{(-1)}(X)=X^8,\quad \mathbb{Z}_0^{(0)}(X)=\mathbb{T}(X),$$
$$\sigma^{(-1)}(X)=0,\quad \text{and}\quad \sigma^{(0)}(X)=1.$$

TABLE 7.7: Steps finding the error-location polynomial and error-value evaluator of the (63,55,9) RS code over $GF(2^6)$ given in Example 7.9.

| $i$ | $\mathbb{Z}_0^{(i)}(X)$ | $\mathbb{q}_i(X)$ | $\sigma(X)$ |
|---|---|---|---|
| $-1$ | $X^8$ | — | $0$ |
| $0$ | $\mathbb{T}(X)$ | — | $1$ |
| $1$ | $\alpha^{46}+\alpha^{48}X+\alpha^{58}X^2+\alpha^{30}X^3$<br>$+\alpha^{25}X^4+\alpha^5X^5+\alpha^{12}X^6$ | $\alpha^{27}+\alpha^{15}X$ | $\alpha^{27}+\alpha^{15}X$ |
| $2$ | $\alpha^{57}+\alpha^{31}X+\alpha^{56}X^2+\alpha^{44}X^3$<br>$+\alpha^{17}X^4+\alpha^{19}X^5$ | $\alpha^{22}+\alpha^{36}X$ | $\alpha^{38}+\alpha^{44}X+\alpha^{51}X^2$ |
| $3$ | $\alpha^3+\alpha^{53}X+\alpha^{30}X^2$<br>$+\alpha^{24}X^3+\alpha^{13}X^4$ | $\alpha^{48}+\alpha^{56}X$ | $\alpha^{47}+\alpha^{22}X+\alpha^{42}X^2+\alpha^{44}X^3$ |

Since $t = 4$ and $e = 2$, the algorithm terminates when $\deg \mathbb{Z}_0(X) < 5$. Executing the algorithm, we obtain Table 7.7. The error-location polynomial is

$$\begin{aligned}\sigma(X) &= \alpha^{47}(1+\alpha^{38}X+\alpha^{58}X^2+\alpha^{60}X^3)\\ &= \alpha^{47}(1+\alpha^6X)(1+\alpha^{20}X)(1+\alpha^{34}X).\end{aligned}$$

The three roots of $\sigma(X)$ are $\alpha^{-6}$, $\alpha^{-20}$, and $\alpha^{-34}$. The reciprocals of these three roots give the error locations, $\alpha^6$, $\alpha^{20}$, and $\alpha^{34}$. The error-value evaluator is

$$\mathbb{Z}_0(X) = \alpha^3+\alpha^{53}X+\alpha^{30}X^2+\alpha^{24}X^3+\alpha^{13}X^4.$$

The overall error/erasure-location polynomial is

$$\begin{aligned}\gamma(X) &= \sigma(X)\boldsymbol{\beta}(X)\\ &= \alpha^{47}(1+\alpha^6X)(1+\alpha^{20}X)(1+\alpha^{28}X)(1+\alpha^{34}X)(1+\alpha^{53}X)\\ &= \alpha^{47}+\alpha^{28}X+\alpha^{18}X^2+\alpha^{48}X^3+\alpha^{12}X^4+\alpha^{62}X^5,\end{aligned}$$

and its derivative is

$$\gamma'(X) = \alpha^{28}+\alpha^{48}X^2+\alpha^{62}X^4.$$

The error values at positions $X^6$, $X^{20}$, and $X^{34}$ are

$$\begin{aligned}e_6 &= \frac{-\mathbb{Z}_0(\alpha^{-6})}{\gamma'(\alpha^{-6})} = \frac{\alpha^{39}}{\alpha^{24}} = \alpha^{15},\\ e_{20} &= \frac{-\mathbb{Z}_0(\alpha^{-20})}{\gamma'(\alpha^{-20})} = \frac{\alpha^{6}}{\alpha^{32}} = \alpha^{37},\\ e_{34} &= \frac{-\mathbb{Z}_0(\alpha^{-34})}{\gamma'(\alpha^{-34})} = \frac{\alpha^{61}}{\alpha^{57}} = \alpha^{4},\end{aligned}$$

and the values of the erased symbols at positions $X^{28}$ and $X^{53}$ are

$$f_{28} = \frac{-\mathbb{Z}_0(\alpha^{-28})}{\gamma'(\alpha^{-28})} = \frac{0}{\alpha^{29}} = 0,$$

$$f_{53} = \frac{-\mathbb{Z}_0(\alpha^{-53})}{\gamma'(\alpha^{-53})} = \frac{0}{\alpha^{13}} = 0.$$

Then, the estimated error polynomial is

$$\mathbb{e}(X) = \alpha^{15}X^6 + \alpha^{37}X^{20} + \alpha^4 X^{34}.$$

Subtracting $\mathbb{e}(X)$ from $\mathbb{r}^*(X)$, we obtain the decoded code polynomial $\mathbb{v}(X) = \mathbb{0}$, which is the transmitted code polynomial.

## PROBLEMS

7.1 Consider the triple-error-correcting RS code given in Example 7.2. Find the code polynomial for the message

$$\mathbb{a}(X) = 1 + \alpha^5 X + \alpha X^4 + \alpha^7 X^8.$$

7.2 Using the Galois field $GF(2^5)$ given in Appendix A, find the generator polynomials of the double-error-correcting and triple-error-correcting RS codes of length 31.

7.3 Using the Galois field $GF(2^6)$ given in Table 6.2, find the generator polynomials of double-error-correcting and triple-error-correcting RS codes of length 63.

7.4 Consider the triple-error-correcting RS code of length 15 given in Example 7.2. Decode the received polynomial

$$\mathbb{r}(X) = \alpha^4 X^3 + \alpha^9 X^8 + \alpha^3 X^{13}$$

using the Berlekamp algorithm.

7.5 Continue Problem 7.4. Decode the received polynomial with the Euclidean algorithm.

7.6 Consider the triple-error-correcting RS code of length 31 constructed in Problem 7.2. Decode the received polynomial

$$\mathbb{r}(X) = \alpha^2 + \alpha^{21} X^{12} + \alpha^7 X^{20}$$

using the Euclidean algorithm.

7.7 Continue Problem 7.6. Decode the received polynomial in the frequency domain using transform decoding.

7.8 For the same RS code of Problem 7.6, decode the following received polynomial with two erasures:

$$\mathbb{r}(X) = (*)X^3 + \alpha^5 X^7 + (*)X^{18} + \alpha^3 X^{21}$$

with the Euclidean algorithm.

7.9 Prove that the dual code of a RS code is also a RS code.

7.10 Prove that the $(2^m - 1, k)$ RS code with minimum distance $d$ contains the primitive binary BCH code of length $2^m - 1$ with designed distance $d$ as a subcode. This subcode is called a *subfield subcode.*

**7.11** Let $\alpha$ be a primitive element in $GF(2^m)$. Consider the $(2^m - 1, k)$ RS code of length of $2^m - 1$ and minimum distance $d$ generated by

$$\mathbf{g}(X) = (X - \alpha)(X - \alpha^2)...(X - \alpha^{d-1}).$$

Prove that extending each codeword $v = (v_0, v_1, \cdots, v_{2^m-2})$ by adding an overall parity-check symbol

$$v_\infty = -\sum_{i=0}^{2^m-2} v_i$$

produces a $(2^m, k)$ code with a minimum distance of $d + 1$.

**7.12** Consider a $t$-symbol error-correcting RS code over $GF(2^m)$ with the following parity-check matrix:

$$\mathbf{H} = \begin{bmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^{n-1} \\ 1 & \alpha^2 & (\alpha^2)^2 & \cdots & (\alpha^2)^{n-1} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & \alpha^{2t} & (\alpha^{2t})^2 & \cdots & (\alpha^{2t})^{n-1} \end{bmatrix},$$

where $n = 2^m - 1$, and $\alpha$ is a primitive element in $GF(2^m)$. Consider the extended Reed–Solomon code with the following parity-check matrix:

$$\mathbf{H}_1 = \left[\begin{array}{cc|c} 0 & 1 & \\ 0 & 0 & \\ \vdots & \vdots & \mathbf{H} \\ 0 & 0 & \\ 1 & 0 & \end{array}\right]$$

Prove that the extended code also has a minimum distance of $2t + 1$.

**7.13** Let $\mathbf{a}(X) = a_0 + a_1X + \cdots + a_{k-1}X^{k-1}$ be a polynomial of degree $k - 1$ or less over $GF(2^m)$. There are $(2^m)^k$ such polynomials. Let $\alpha$ be a primitive element in $GF(2^m)$. For each polynomial $\mathbf{a}(X)$, form the following polynomial of degree $2^m - 2$ or less over $GF(2^m)$:

$$\mathbf{v}(X) = \mathbf{a}(1) + \mathbf{a}(\alpha)X + \mathbf{a}(\alpha^2)X^2 + \cdots + \mathbf{a}(\alpha^{2^m-2})X^{2^m-2}.$$

Prove that the set $\{\mathbf{v}(X)\}$ forms the $(2^m - 1, k)$ RS code over $GF(2^m)$. (*Hint:* Show that $\mathbf{v}(X)$ has $\alpha, \alpha^2, \cdots, \alpha^{2^m-k-1}$ as roots). This original definition of a RS code is given by Reed and Solomon [1].

## BIBLIOGRAPHY

1. I. S. Reed and G. Solomon, "Polynomial Codes over Certain Fields," *J. Soc. Ind. Appl. Math.*, 8: 300–304, June 1960.

2. D. Gorenstein and N. Zierler, "A Class of Cyclic Linear Error-Correcting Codes in $p^m$ Symbols," *J. Soc. Ind. Appl. Math.*, 9: 107–214, June 1961.

3. R. T. Chien, "Cyclic Decoding Procedure for the Bose–Chaudhuri–Hocquenghem Codes," *IEEE Trans. Inf. Theory*, IT-10: 357–63, October 1964.

4. G. D. Forney, "On Decoding BCH Codes," *IEEE Trans. Inf. Theory*, IT-11: 549–57, October 1965.

5. E. R. Berlekamp, *Algebraic Coding Theory*, McGraw-Hill, New York, 1968.

6. Y. Sugiyama, M. Kasahara, S. Hirasawa, and T. Namekawa, "A Method for Solving Key Equation for Decoding Goppa Codes," *Inf. Control*, 27: 87–99, January 1975.

7. W. C. Gore, "Transmitting Binary Symbols with Reed–Solomon Codes," *Proc. Conf. Infor. Sci. and Syst.*, Princeton, N.J., 495–97, 1973.

8. R. E. Blahut, "Transform Techniques for Error-Control Codes," *IBM J. Res. Dev.*, 23(3), 299–315 May 1979.

9. T. Kasami and S. Lin, "On the Probability of Undetected Error for the Maximum Distance Separable Codes," *IEEE Trans. Commun.*, COM-32: 998–1006, September 1984.

10. E. F. Assmus, Jr., H. F. Mattson, Jr., and R. J. Turyn, "Cyclic Codes," *Scientific Report No. AFCRL-65-332*, Air Force Cambridge Research Labs, Bedford, Mass., April 1965.

11. T. Kasami, S. Lin, and W. W. Peterson, "Some Results on Weight Distributions of BCH Codes," *IEEE Trans. Inf. Theory*, IT-12(2): 274, April 1966.

12. G. D. Forney, Jr., *Concatenated Codes*, MIT Press, Cambridge, 1966.

13. J. L. Massey, "Shift-Register Synthesis and BCH Decoding," *IEEE Trans. Inf. Theory*, IT-15: 122–27, January 1969.

14. W. W. Peterson and E. J. Weldon, Jr., *Error-Correcting Codes*, 2d ed., MIT Press, Cambridge, 1970.

15. F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*, North Holland, Amsterdam, 1977.

16. G. C. Clark, Jr., and J. B. Cain, *Error-Correcting Coding for Digital Communications*, Plenum Press, New York, 1981.

17. R. E. Blahut, *Theory and Practice of Error Control Codes*, Addison-Wesley, Reading, Mass., 1983.

18. A. M. Michelson and A. H. Levesque, *Error-Control Techniques for Digital Communication*, John Wiley, New York, 1985.

19. S. B. Wicker, *Error Control Systems for Digital Communication and Storage*, Prentice Hall, Englewood Cliffs, N.J., 1995.

20. S. B. Wicker and V. K. Bhargava, *Reed–Solomon Codes and Their Applications*, IEEE Press, New York, 1994.

21. T. Kasami, S. Lin, and W. W. Peterson, "Some Results on Cyclic Codes Which Are Invariant under the Affine Group," *Scientific Report AFCRL-66-622*, Air Force Cambridge Research Labs, Bedford, Mass., 1966.

22. J. K. Wolf, "Adding Two Information Symbols to Certain Nonbinary BCH Codes and Some Applications," *Bell Syst. Tech. J.*, 48: 2405–24, 1969.

23. M. Morii and M. Kasahara, "Generalized Key-Equation of Remainder Decoding Algorithm for Reed–Solomon Codes," *IEEE Trans. Inf. Theory*, IT-38 (6): 1801–7, November 1992.