

## 10. Error-Evaluator

- The syndrome components  $s_i$  can be represented by

$$S_i = r(\alpha^i) = \sum_{j=0}^{n-1} r_j \alpha^{ij}$$

After the syndrome components are evaluated, the error pattern

$e_j$  for  $j = 0, 1, 2, \dots, n-1$  can be determined.

- Suppose that  $\nu$  errors,  $0 \leq \nu \leq t$ , occur in the unknown locations  $j_1, j_2, \dots, j_\nu$ .

$$\text{Then } e(x) = e_{j_1} x^{j_1} + e_{j_2} x^{j_2} + \dots + e_{j_\nu} x^{j_\nu}$$

The error locations, the error values and the number of errors are unknown.

- Define the error values to be

$$Y_l = e_{j_l} \quad \text{for } l = 1, 2, \dots, \nu$$

And the error locators to be

$$X_l = \alpha^{j_l} \quad \text{for } l = 1, 2, \dots, \nu$$

- The  $2t = d - 1$  syndrome components can then be

$$X_j = Y_1 X_1^j + Y_2 X_2^j + \dots + Y_v X_v^j$$

The right-hand side of the above equation is called “power-sum symmetric functions”.

- This set of equations must have at least one solution for the unknown  $X_i$ 's and  $Y_i$ 's because of the way in which the syndromes are defined in eq. (6-14).

It can be shown that the solution is unique.

- Methods to determine error values:

(1) Straightforward solution of the equations

$$S_j = Y_1 X_1^j + Y_2 X_2^j + \dots + Y_v X_v^j \quad j = 1, 2, \dots, 2t$$

(2) Use of Forney's algorithm (will be derived latter)

This method requires the evaluation of  $\Lambda(x)$ .

First, define the syndrome polynomial

$$S(x) = \sum_{i=1}^{2t} S_i x^i$$

Let the error-evaluator polynomial  $\Omega(x)$  be formed in terms of the known polynomials  $S(x)$  and  $\Lambda(x)$ , which is called, the key equation:

$$\Omega(x) = [1 + S(x)]\Lambda(x) \bmod x^{2t+1}$$

$$\text{where } \Lambda(x) = \prod_{l=1}^{\nu} (1 - xX_l), \quad X_l = \alpha^l$$

$$\text{Then } \Omega(X_l^{-1}) = Y_l \prod_{i \neq l} (1 - X_i X_l^{-1}) \quad \text{for } l = 1, 2, \dots, \nu$$

Thus the error values are given explicitly by the formula

$$Y_l = -X_l \frac{\Omega(X_l^{-1})}{\Lambda'(x_l^{-1})} \quad \text{for } l = 1, 2, \dots, \nu$$

where  $\Lambda'(x)$  denotes the formal first derivative of  $\Lambda(x)$  with respect to  $x$ .

The above equation defines the Forney algorithm

(G. D. Forney, 1965)

### (3) Transformed version of the Forney technique

From the relationship

$$\Lambda(x) = \prod_{l=1}^{\nu} (1 - xX_l), \quad X_l = \alpha^l$$

$$\text{we have } \Lambda(X_l^{-1}) = \prod_{j \neq l} (1 - X_j X_l^{-1})$$

and the from eq. (6-21) then

$$Y_l = \frac{-\Omega(X_l^{-1})}{\prod_{i \neq l} (1 - X_i X_l^{-1})} \quad \text{for } l = 1, 2, \dots, \nu$$

Combining eq. (6-25) and eq. (6-20) yield the explicit formula:

$$Y_l = \frac{\sum_{i=1}^{\nu} S_{\nu-i} \sigma_{l,i}}{X_l \prod_{i \neq l} (X_l - X_i)} \quad \text{for } l = 1, 2, \dots, \nu$$

where  $\sigma_{l,i}$  is defined as the  $(\nu-1-i)$ -th coefficient of the polynomial

$$\bar{\Lambda}(x) = \prod_{\substack{i=1 \\ i \neq l}}^{\nu} (x - X_i) = \sum_{i=0}^{\nu-1} \sigma_{l,i} x^{\nu-1-i}$$

### Example 6.8 (p. 261)

Consider the (15, 9) RS code over  $\text{GF}(2^4)$

The generator of the field  $\text{GF}(2^4)$ , primitive polynomial,

$$p_4(x) = 1 + x + x^4$$

codeword generator

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

Assuming that two error locations are already determined.

$$X_1 = \alpha^2, \quad X_2 = \alpha^8$$

with (example 6.7)

$$r(x) = x^8 + \alpha^{11}x^7 + \alpha^8x^5 + \alpha^{10}x^4 + \alpha^4x^3 + \alpha^3x^2 + \alpha^8x + \alpha^{12}$$

$$S_1 = r(\alpha) = 1$$

$$S_2 = r(\alpha^2) = 1$$

$$S_3 = r(\alpha^3) = \alpha^5$$

$$S_4 = r(\alpha^4) = 1$$

$$S_5 = r(\alpha^5) = 0$$

$$S_6 = r(\alpha^6) = \alpha^{10}$$

$$\Lambda(x) = (1 + \alpha^2x)(1 + \alpha^8x) = 1 + \alpha^{10}x$$

From the 1<sup>st</sup> method

$$Y_1\alpha^2 + Y_2\alpha^8 = 1$$

$$Y_1\alpha^4 + Y_2\alpha = 1$$

we then obtain  $Y_1 = 1, \quad Y_2 = 1$

Use of Forney's formula

$$\Omega(x) = [1 + S(x)]\Lambda(x) \bmod x^{2t+1} = 1 + \alpha^{10}x^2$$

$$\Lambda(x) = (1 - x\alpha^2)(1 - x\alpha^8)$$

$$\Lambda'(x) = -\alpha^2(1 - \alpha^8x) - \alpha^8(1 - \alpha^2x)$$

$$Y_1 = \frac{\Omega\left(\frac{1}{x_1}\right)}{\left(1 + \frac{x_2}{x_1}\right)} = \frac{1 + \frac{\alpha^{10}}{\alpha^4}}{1 + \alpha^6} = 1, \quad Y_2 = \frac{\Omega\left(\frac{1}{x_2}\right)}{\left(1 + \frac{x_1}{x_2}\right)} = 1$$

## 11. Error-and –Erasure Decoding of RS Codes

### Berlekamp-Forney key equation

- The Euclidean algorithm is used to directly solve the Berlekamp-Forney key equation for the errata-locator polynomial and errata-evaluator polynomial at the same time with a common algorithm.
- Suppose  $s$  errors and  $\nu$  erasures occur in the received word  $\bar{r}$ , and that  $2\nu + s \leq d - 1 = 2t$ .

Next, let  $\alpha$  be a primitive element in  $\text{GF}(2^m)$ , the  $\gamma = \alpha^l$  is also a primitive element in  $\text{GF}(2^m)$ , where  $(l, n) = 1$ ,  $n = 2^m - 1$ .

- If  $\gamma$  is a root of the generator polynomial of the code, it is shown (by Berlekamp) that the generator polynomial  $g(x)$  is symmetric if and only if

$$g(x) = \sum_{j=0}^{d-1} g_j x^j = \prod_{j=b}^{b+(d-2)} (x - \gamma^j)$$

where  $g_0 = g_{d-1} = 1$  and  $b$  satisfies the equality

$$2b + d - 2 = 2^m - 1$$

- The syndrome of the code are given by

$$\begin{aligned}
 S_{(b-1)+w} &= r(\gamma^{b-1+w}) \\
 &= \sum_{i=0}^{n-1} u_i \gamma^{i(b-1+w)} \\
 &= \sum_{j=1}^{v+s} Y_j X_j^{(b-1)+w}
 \end{aligned}$$

For  $1 \leq w \leq d-1$ , where  $u_i$  for  $0 \leq i \leq n-1$  are the coefficients of the errata polynomial  $u(x)$ ,  $X_j$  is either the  $j$ -th erasure of error location, and  $Y_j$  is either the  $j$ -th erasure or error magnitude.

It can be shown that [Reed & Solomon, 1960]

$$\begin{aligned}
 S(x) &= \sum_{w=1}^{d-1} S_{(b-1)+w} x^{w-1} \\
 &= \sum_{j=1}^{v+s} \frac{Y_j X_j^b}{(1 - X_j x)} - \sum_{j=1}^{v+s} \frac{Y_j X_j^{b+d-1} x^{d-1}}{(1 - X_j x)}
 \end{aligned}$$

- Now define the following four polynomials:

(1) erasure

$$\tau(x) = \prod_{j=1}^s (1 - X_j x)$$

(2) error locator

$$\lambda(x) = \prod_{j=1}^v (1 - X_j x)$$

**(3) errata locator**

$$\Lambda(x) = \tau(x)\lambda(x) = \prod_{j=1}^{\nu+s} (1 - X_j x)$$

**(4) errata evaluator**

$$A(x) = \sum_{j=1}^{\nu+s} Y_j X_j^b \left( \prod_{i \neq j} (1 - X_i x) \right)$$

**In terms of the polynomial defined above, eq. (6-30) becomes**

$$S(x) = \frac{A(x)}{\Lambda(x)} + \frac{x^{d-1} \sum_{j=1}^{\nu+s} Y_j X_j^{b+d-1} \left( \prod_{i \neq j} (1 - X_i x) \right)}{\Lambda(x)}$$

**or**

$$S(x)\Lambda(x) = A(x) + x^{d-1} \sum_{j=1}^{\nu+s} Y_j X_j^{b+d-1} \left( \prod_{i \neq j} (1 - X_i x) \right)$$

**From eq. (6-36) one obtains finally the congruence relation,**

$$S(x)\Lambda(x) = A(x) \pmod{x^{d-1}}$$

**for key equation**

- **Equation (6-37) can be solved for  $S(x)$  to yield:**

$$S(x) \equiv \frac{A(x)}{\lambda(x)\tau(x)} \pmod{x^{d-1}}$$

- **Now define, what is called, the Forney syndrome polynomial  $T(x)$**

**as follows:**

$$T(x) \equiv S(x)\tau(x) \pmod{x^{d-1}}$$

- **By eq. (6-38) & eq. (6-39), one obtains, what is called, the Berlekamp-forney key equation for  $\lambda(x)$ , and  $A(x)$  is given by**

$$A(x) \equiv T(x)\lambda(x) \pmod{x^{d-1}}$$

**Where  $\deg\{\lambda(x)\} \leq \lfloor (d-1-s)/2 \rfloor$  since the maximum number of errors in an RS code that can be corrected is  $\lfloor (d-1-s)/2 \rfloor$ .**

**Here also,  $\deg\{A(x)\} \leq \lfloor (d+v-3)/2 \rfloor$ .**

- **The Forney syndrome polynomial  $T(x)$  can be computed from eq. (6-39) since both  $S(x)$  and  $\tau(x)$  are known.**

## 12. Euclid's Algorithm:

- Euclid's algorithm is a fast method for finding the greatest common divisor (GCD) of a collection of elements in an Euclidean domain.
- Theorem (Theorem 6.2, page 273)

For a  $(n, k)$  RS code, with  $v$ -error and  $s$ -erasure in the received word. Forney syndrome polynomial  $T(x)$  is expressed by

$$T(x) \equiv S(x)\tau(x) \pmod{x^{d-1}}$$

Consider the two polynomials  $x^{d-1}$  and  $T(x)$  in eq. (6-39). The Euclidean algorithm for polynomials over  $\text{GF}(2^m)$  can be used to develop two finite sequences  $R_i(x)$  and  $\Lambda_i(x)$  from the following two recursive formulas:

$$\Lambda_i(x) = (-Q_{i-1}(x))\Lambda_{i-1}(x) + \Lambda_{i-2}(x)$$

and  $R_i(x) = R_{i-2}(x) - Q_{i-1}(x)R_{i-1}(x)$

For  $i = 1, 2, \dots, 2t$ , where the initial conditions are

$$\begin{aligned} \Lambda_0(x) &= \tau(x) & R_0(x) &= T(x) \\ \Lambda_{-1}(x) &= 0 & R_{-1}(x) &= x^{d-1} \end{aligned}$$

Here  $Q_{i-1}(x)$  is obtained as the principal part of  $\frac{R_{i-2}(x)}{R_{i-1}(x)}$ .

The recursion in eq. (6-40a) and eq. (6-40b) for  $R_i(x)$  and  $\Lambda_i(x)$  terminates when  $\deg\{R_i(x)\} \leq \lfloor (d + \nu - 3)/2 \rfloor$  for the first time for some value  $i = i'$

Let  $A(x) = R_{i'}(x) / \Delta$

and  $\Lambda(x) = \Lambda_{i'}(x) / \Delta$

Also in eq. (6.41)  $\Delta = \Lambda_{i'}(0)$  is a field element in  $\text{GF}(2^m)$  which is chosen so that  $\Lambda_0 = 1$ .

The pair of polynomials  $A(x)$  and  $\Lambda(x)$  in eq. (6.41) constitute the unique solution of  $A(x) \equiv T(x)\Lambda(x) \pmod{x^{d-1}}$ , where both of the inequalities,  $\deg\{\Lambda(x)\} \leq \lfloor (d + \nu - 1)/2 \rfloor$  and  $\deg\{A(x)\} \leq \lfloor (d + \nu - 3)/2 \rfloor$  are satisfied.

- **Errata-value computation**

The locations of the errata can be obtained by finding the roots of  $\Lambda(x)$  using the Chien search procedure.

By eq. (6.34), it is shown readily that the errata values are

$$Y_j = \frac{A(X_j^{-1})}{(X_j^{b-1} \Lambda'(X_j^{-1}))} \quad \text{for } j = 1, 2, \dots, \nu + s$$

where  $\Lambda'(X_j^{-1})$  is the derivative with respect to  $x$  of  $\Lambda(x)$ , evaluated at  $x = X_j^{-1}$ .

- **The overall decoding process of RS codes for correcting errors and erasures, which used the Euclidean algorithm, is summarized in the following steps:**

**(1) Compute the syndrome  $S_1, S_2, \dots, S_{2t}$  with all of the erasure positions in the received word being replaced by 0s from eq. (6. 29), (6. 30) and next calculate  $\lambda(x)$  from eq. (6. 31) and let  $\deg\{\lambda(x)\} = \nu$ .**

**(2) Compute the Forney syndrome polynomial from  $T(x)$  by the use of eq. (6. 37)**

**(3) To determine  $\Lambda(x)$  and  $A(x)$ , where  $0 \leq \nu \leq d - 1$ , apply the Euclidean algorithm to  $x^{d-1}$  and  $T(x)$  as given by eq. (6. 39)**

**The initial values are**

$$\begin{aligned} \Lambda_0(x) &= \tau(x) & R_0(x) &= T(x) \\ \Lambda_{-1}(x) &= 0 & R_{-1}(x) &= x^{d-1} \end{aligned}$$

**For  $\nu = d - 1$ , set  $\Lambda(x) = \tau(x)$  and  $A(x) = T(x)$**

**(4) Compute the errata value from eq. (6. 42)**

**Example 6.10 (page 277)**

Consider the (15, 9) RS code,  $\alpha^{15} = 1$

The minimum distance of the code is  $d = 7$

In this code,  $s$  erasures and  $\nu$  errors under the condition  $s + 2\nu \leq d - 1$  can be corrected.

Assuming that the received word is

$$\bar{r} = (\alpha^{10}, \alpha^{12}, \alpha^8, \alpha^5, \alpha, \alpha^{14}, \alpha^{13}, \alpha^9, \alpha^9, 1, \alpha, \alpha^{12}, \alpha^6, \alpha^{12}, \alpha^8)$$

Also assume that the erasure vector is

$$\bar{e}^* = (0, 0, 0, 0, 0, 0, 0, \alpha^2, 0, 0, 0, 0, 0, 0, 0)$$

And the error vector is

$$\bar{e} = (0, 0, 0, 0, \alpha^{11}, 0, 0, 0, 0, 0, 0, \alpha^7, 0, 0, 0)$$

The errata vector is

$$\bar{u} = (0, 0, 0, 0, \alpha^{11}, 0, 0, \alpha^2, 0, 0, 0, \alpha^7, 0, 0, 0)$$

The syndrome  $S_i$  satisfy

$$S_i = \sum_{j=0}^{14} r_j \alpha^{ij} = \alpha^7 (\alpha^3)^i + \alpha^2 (\alpha^7)^i + \alpha^{11} (\alpha^{10})^i \quad \text{for } 1 \leq i \leq 6$$

This yields

$$\begin{array}{lll} S_1 = 1 & S_3 = \alpha^{14} & S_5 = \alpha \\ S_2 = \alpha^{13} & S_4 = \alpha^{11} & S_6 = 0 \end{array}$$

Thus  $S(x) = 1 + \alpha^{13}x + \alpha^{14}x^2 + \alpha^{11}x^3 + \alpha^{14}$

**The erasure locator polynomial is**

$$\tau(x) = 1 + \alpha^7 x$$

**By eq. (6. 39),  $T(x) \equiv S(x)\tau(x) \pmod{x^{d-1}}$ ,**

**one obtains**

$$\begin{aligned} T(x) &\equiv (1 + \alpha^7 x)(1 + \alpha^{13} x + \alpha^{14} x^2 + \alpha^{11} x^3 + \alpha x^4) \pmod{x^6} \\ &= \alpha^8 x^5 + \alpha^9 x^4 + \alpha x^3 + \alpha^{12} x^2 + \alpha^5 x + 1 \end{aligned}$$

**Now applying the Euclidean algorithm to  $x^{d-1}$  and  $T(x)$ .**

**Initial conditions:**

$$R_{-1}(x) = x^{d-1} = x^6$$

$$R_0(x) = T(x) = \alpha^8 x^5 + \alpha^9 x^4 + \alpha x^3 + \alpha^{12} x^2 + \alpha^5 x + 1$$

$$\Lambda_0(x) = \tau(x) = 1 + \alpha^7 x$$

$$\Lambda_{-1}(x) = 0$$

$$d = 7 \quad \nu = 2$$

**Iterations:**

$$\Lambda_i(x) = (-Q_{i-1}(x))\Lambda_{i-1}(x) + \Lambda_{i-2}(x)$$

$$R_i(x) = R_{i-2}(x) - Q_{i-1}(x)R_{i-1}(x)$$

$$Q_{i-1}(x) = \left[ \begin{array}{c} R_{i-2}(x) \\ R_{i-1}(x) \end{array} \right]$$