Chapter 6 Reed-Solomon Codes

- 6.1 Finite Field Algebra
- 6.2 Reed-Solomon Codes
- 6.3 Syndrome Based Decoding
- 6.4 Curve-Fitting Based Decoding

§ 6.1 Finite Field Algebra

- Nonbinary codes: message and codeword symbols are represented in a finite field of size q, and q>2.
- Advantage of presenting a code in a nonbinary image.

A binary codeword sequence in {0,1}

b_0 $\boxed{b_1}$ $\boxed{b_2}$ $\boxed{b_3}$ $\boxed{b_4}$ $\boxed{b_5}$ b_6 b_7 b_8 b_9 b_{10} b_{11} $\boxed{b_{12}}$ $\boxed{b_{13}}$ $\boxed{b_{14}}$ b_{15} b_{16} b_{17}

b_{18} b_{19} b_{20}

A nonbinary codeword sequence in {0, 1, 2, 3, 4, 5, 6, 7}

$\boxed{c_0}$ $\boxed{c_1}$ c_2 c_3 c_4 $\boxed{c_5}$ c_6

$\boxed{}$: where the channel error occurs

8 bit errors are treated as 3 symbol errors in a nonbinary image

§ 6.1 Finite Field Algebra

- Finite field (Galois field) $\mathbf{F}_q$: a set of q elements that perform " + " " **-** " " × " " / " without leaving the set.
- Let p denote a prime, e.g., 2, 3, 5, 7, 11, ⋯ , it is required $q = p$ *or* $q = p^\theta$ (θ is a positive integer greater than 1). If $q = p^\theta$, $\mathbf{F}_q$ is an extension field of $\mathbf{F}_p$.
- **Example 6.1:** " + " and " × " in $\mathbf{F}_q$.

$\mathbf{F}_2 = \{ 0, 1 \}$

+	0	1
0	0	1
1	1	0

×	0	1
0	0	0
1	0	1

all in modulo-2

$\mathbf{F}_5 = \{ 0, 1, 2, 3, 4 \}$

+	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

×	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

all in modulo-5

§ 6.1 Finite Field Algebra

– “ - ” and “ / ” can be performed as “ + ” and “ × ” with additive inverse and multiplicative inverse, respectively.

<u>Additive inverse of *a*</u> a': $a' + a = 0$ and $a' = -a$

<u>Multiplicative inverse of *a*</u> a': $a' \bullet a = 1$ and $a' = 1 / a$

– “ - ” operation:

Let $h, a \in \mathbf{F}_q$.

$h - a = h + (-a) = h + a'$.

E.g., in $\mathbf{F}_5$, $1 - 3 = 1 + (-3) = 1 + 2 = 3$;

– “ / ” operation:

Let $h, a \in \mathbf{F}_q$.

$h / a = h \times a'$.

E.g., in $\mathbf{F}_5$, $2 / 3 = 2 \times (1 / 3) = 2 \times 2 = 4$.

§ 6.1 Finite Field Algebra

– Nonzero elements of $\mathbf{F}_q$ can be represented using a primitive element σ such that
$\mathbf{F}_q = \{ 0, 1, \sigma, \sigma^2, \cdots , \sigma^{q-2} \}$.

– Primitive element σ of $\mathbf{F}_q$: $\sigma \in \mathbf{F}_q$ and unity can be produced by at least

$$\underbrace{\sigma \bullet \sigma \bullet \cdots \bullet \sigma}_{q-1} = 1, \text{ or } \sigma^{q-1} = 1.$$

all in modulo-q

E.g., in $\mathbf{F}_5$, $2^4 = 1$ and $3^4 = 1$. Here, 2 and 3 are the primitive elements of $\mathbf{F}_5$.

– **Example 6.2:** In $\mathbf{F}_5$,
If 2 is chosen as the primitive element, then

$$\mathbf{F}_5 = \{ 0, 1, 2, 3, 4 \} = \{0, 2^4, 2^1, 2^3, 2^2 \} = \{0, 1, 2^1, 2^3, 2^2 \}$$

If 3 is chosen as the primitive element, then

$$\mathbf{F}_5 = \{ 0, 1, 2, 3, 4 \} = \{0, 3^4, 3^3, 3^1, 3^2 \} = \{0, 1, 3^3, 3^1, 3^2 \}$$

§ 6.1 Finite Field Algebra

– If $\mathbf{F}_q$ is an extension field of $\mathbf{F}_p$ such as $q = p^\theta$, elements of $\mathbf{F}_q$ can also be represented by θ-dimensional vectors in $\mathbf{F}_p$.

– Primitive polynomial $p(x)$ of $\mathbf{F}_q$ ($q = p^\theta$): an irreducible polynomial of degree θ that divides $x^{p^\theta - 1} - 1$ but not other polynomials x^Φ - 1 with $\Phi < p^\theta$ - 1.
E.g., in $\mathbf{F}_8$, the primitive polynomial $p(x) = x^3 + x + 1$ divides x^7-1, but not x^6-1, x^5-1, x^4-1, x^3-1.

– If a primitive element σ is a root of $p(x)$ such that $p(\sigma) = 0$, elements of $\mathbf{F}_q$ can be represented in the form of

$$w_{\theta-1}\sigma^{\theta-1} + w_{\theta-2}\sigma^{\theta-2} + \dots + w_1\sigma^1 + w_0\sigma^0$$

where $w_0, w_1, \dots, w_{\theta-2}, w_{\theta-1} \in \mathbf{F}_p$, or alteratively in

$$(w_{\theta-1}, w_{\theta-2}, \cdots , w_1, w_0)$$

§ 6.1 Finite Field Algebra

– **Example 6.3:** If $p(x) = x^3 + x + 1$ is the primitive polynomial of $\mathbf{F}_8$, and its primitive element σ satisfies $\sigma^3 + \sigma + 1 = 0$, then

$\mathbf{F}_8$	$w_2\sigma^2 + w_1\sigma^1 + w_0\sigma^0$	w_2 w_1 w_0
0	0	0 0 0
1	1	0 0 1
σ	σ	0 1 0
σ^2	σ^2	1 0 0
σ^3	$\sigma + 1$	0 1 1
σ^4	$\sigma^2 + \sigma$	1 1 0
σ^5	$\sigma^2 + \sigma + 1$	1 1 1
σ^6	$\sigma^2 + 1$	1 0 1

§ 6.1 Finite Field Algebra

– Representing $\mathbf{F}_q = \{ 0, 1, \sigma, \cdots, \sigma^{q-2} \}$, " × " " / " " + " " - " operations become

" × ": $\sigma^i \times \sigma^j = \sigma^{(i+j) \% (q-1)}$

E.g., in $\mathbf{F}_8$, $\sigma^4 \times \sigma^5 = \sigma^{(4+5) \% 7} = \sigma^2$

" / ": $\sigma^i / \sigma^j = \sigma^{(i-j) \% (q-1)}$

E.g., in $\mathbf{F}_8$, $\sigma^4 / \sigma^5 = \sigma^{(4-5) \% 7} = \sigma^6$

" + ": if $\sigma^i = w_{\theta-1}\sigma^{\theta-1} + w_{\theta-2}\sigma^{\theta-2} + \cdots + w_0\sigma^0$

(&" - ") $\sigma^j = w'_{\theta-1}\sigma^{\theta-1} + w'_{\theta-2}\sigma^{\theta-2} + \cdots + w'_0\sigma^0$

$\sigma^i + \sigma^j = (w_{\theta-1} + w'_{\theta-1})\sigma^{\theta-1} + (w_{\theta-2} + w'_{\theta-2})\sigma^{\theta-2} + \cdots + (w_0 + w'_0)\sigma^0$

E.g., in $\mathbf{F}_8$, $\sigma^4 + \sigma^5 = \sigma^2 + \sigma + \sigma^2 + \sigma + 1 = 1$

§ 6.2 Reed-Solomon Codes

– An RS code[1] defined over $\mathbf{F}_q$ is characterized by its codeword length $n = q - 1$, dimension $k < n$ and the minimum Hamming distance d. It is often denoted as an (n, k) (or (n, k, d)) RS code.

– It is a maximum distance separable (MDS) code such that

$$d = n - k + 1$$

– It is a linear block code and also cyclic.

– The widely used RS codes include the (255, 239) and the (255, 223) codes both of which are defined in $\mathbf{F}_{256}$.

[1] I. Reed and G. Solomon, “Polynomial codes over certain finite fields,” *J. Soc. Indust. Appl. Math*, vol. 8, pp. 300-304, 1960.

§ 6.2 Reed-Solomon Codes

– Notations

$\mathbf{F}_q[x]$, a univariate polynomial ring over $\mathbf{F}_q$, i.e., $f(x) = \sum_{i \in \mathrm{N}} f_i x^i$ and $f_i \in \mathbf{F}_q$.

$\mathbf{F}_q[x, y]$, a bivariate polynomial ring over $\mathbf{F}_q$, i.e., $f(x, y) = \sum_{i, j \in \mathrm{N}} f_{ij} x^i y^j$ and $f_{ij} \in \mathbf{F}_q$.

$\mathbf{F}_q^{\bullet}$, $\bullet$ - dimensional vector over $\mathbf{F}_q$.

– Encoding of an (n, k) RS code.

Message vector $\bar{u} = (u_0, u_1, u_2, \cdots, u_{k-1}) \in \mathbf{F}_q^k$

Message polynomial

$$u(x) = u_0 + u_1 x + u_2 x^2 + \cdots + u_{k-1} x^{k-1} \in \mathbf{F}_q[x]$$

Codeword

$$\bar{c} = \left(u(\alpha_0), u(\alpha_1), u(\alpha_2), \cdots, u(\alpha_{n-1})\right) \in \mathbf{F}_q^n$$

$\alpha_0, \alpha_1, \alpha_2, \cdots, \alpha_{n-1} \in \mathbf{F}_q \backslash \{0\}$. They are often called code locators.

§ 6.2 Reed-Solomon Codes

– Encoding of an (n, k) RS code in a linear block code fashion

$$\bar{c} = \bar{u} \cdot \mathbf{G}$$

$$= (u_0, u_1, \cdots, u_{k-1}) \begin{bmatrix} (\alpha_0)^0 & (\alpha_1)^0 & \cdots & (\alpha_{n-1})^0 \\ (\alpha_0)^1 & (\alpha_1)^1 & \cdots & (\alpha_{n-1})^1 \\ \vdots & \vdots & \ddots & \vdots \\ (\alpha_0)^{k-1} & (\alpha_1)^{k-1} & \cdots & (\alpha_{n-1})^{k-1} \end{bmatrix}$$

– **Example 6.4:** For a (7, 3) RS code that is defined in $\mathbf{F}_8$, if the message is $\bar{u} = (u_0, u_1, u_2) = (\sigma^4, 1, \sigma^5)$, the message polynomial will be $u(x) = \sigma^4 + x + \sigma^5 x^2$. Let $\{\alpha_0, \alpha_1, \cdots, \alpha_6\} = \{1, \sigma, \cdots, \sigma^6\}$, the codeword can be generated by

- $\bar{c} = (u(1), u(\sigma), u(\sigma^2), u(\sigma^3), u(\sigma^4), u(\sigma^5), u(\sigma^6)) = (0, \sigma^6, \sigma^4, \sigma^3, \sigma^6, \sigma^3, 0)$
- $\bar{c} = \bar{u} \cdot \mathbf{G} = \left(\sigma^4, 1, \sigma^5\right) \cdot \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \sigma^1 & \sigma^2 & \sigma^3 & \sigma^4 & \sigma^5 & \sigma^6 \\ 1 & \sigma^2 & \sigma^4 & \sigma^6 & \sigma^1 & \sigma^3 & \sigma^5 \end{bmatrix} = (0, \sigma^6, \sigma^4, \sigma^3, \sigma^6, \sigma^3, 0)$

§ 6.2 Reed-Solomon Codes

- MDS property of RS codes $d = n - k + 1$
  - Singleton bound for an (n, k) linear block code, $d \le n - k + 1$
  - $u(x)$ has at most k - 1 roots. Hence, $\overline{c}$ has at most k - 1 zeros and

$$d_{\text{Ham}} = (\overline{c}, \overline{0}) \ge n - k + 1$$

- Parity-check matrix of an (n, k) RS code

$$\mathbf{H} = \begin{bmatrix} (\alpha_0)^1 & (\alpha_1)^1 & \cdots & (\alpha_{n-1})^1 \\ (\alpha_0)^2 & (\alpha_1)^2 & \cdots & (\alpha_{n-1})^2 \\ \vdots & \vdots & \ddots & \vdots \\ (\alpha_0)^{n-k} & (\alpha_1)^{n-k} & \cdots & (\alpha_{n-1})^{n-k} \end{bmatrix}$$

$\overline{c} \cdot \mathbf{H}^T = \overline{u} \cdot \mathbf{G} \cdot \mathbf{H}^T = \overline{0}$ ← an n - k all zero vector

§ 6.2 Reed-Solomon Codes

– Insight of $\mathbf{G} \cdot \mathbf{H}^T$

$$\begin{bmatrix} (\alpha_0)^0 & (\alpha_1)^0 & \cdots & (\alpha_{n-1})^0 \\ (\alpha_0)^1 & (\alpha_1)^1 & \cdots & (\alpha_{n-1})^1 \\ \vdots & \vdots & \ddots & \vdots \\ (\alpha_0)^{k-1} & (\alpha_1)^{k-1} & \cdots & (\alpha_{n-1})^{k-1} \end{bmatrix} \cdot \begin{bmatrix} (\alpha_0)^1 & (\alpha_0)^2 & \cdots & (\alpha_0)^{n-k} \\ (\alpha_1)^1 & (\alpha_1)^2 & \cdots & (\alpha_1)^{n-k} \\ \vdots & \vdots & \ddots & \vdots \\ (\alpha_{n-1})^1 & (\alpha_{n-1})^2 & \cdots & (\alpha_{n-1})^{n-k} \end{bmatrix}$$

– Let $i = 0, 1, \cdots, k-1$, $j = 0, 1, \cdots, n-1$, $v = 0, 1, \cdots, n-k-1$.

Entries of $\mathbf{G}$ can be denoted as $[\ \mathbf{G}\ ]_{i,j} = (\ \alpha_j\ )^i$

Entries of $\mathbf{H}^T$ can be denoted as $[\ \mathbf{H}^T\ ]_{j,v} = (\alpha_j)^{v+1}$

Entries of $\mathbf{G} \cdot \mathbf{H}^T$ is

$$[\mathbf{G} \cdot \mathbf{H}^T]_{i,v} = \sum_{j=0}^{n-1} (\alpha_j)^i \cdot (\alpha_j)^{v+1} = \sum_{j=0}^{n-1} (\alpha_j)^{i+v+1} = \left(\sum_{j=0}^{n-1} \alpha_j \right)^{i+v+1} = 0$$

§ 6.2 Reed-Solomon Codes

- Let $\{\alpha_0, \alpha_1, \cdots, \alpha_{n-1}\} = \{\sigma^0, \sigma^1, \cdots, \sigma^{n-1}\}$, $\mathbf{H}^T$ can be perceived as

$$\begin{bmatrix} (\sigma^1)^0 & (\sigma^2)^0 & \cdots & (\sigma^{n-k})^0 \\ (\sigma^1)^1 & (\sigma^2)^1 & \cdots & (\sigma^{n-k})^1 \\ \vdots & \vdots & \ddots & \vdots \\ (\sigma^1)^{n-1} & (\sigma^2)^{n-1} & \cdots & (\sigma^{n-k})^{n-1} \end{bmatrix}$$

- Perceiving codeword $\overline{c} = (c_0, c_1, \cdots, c_{n-1})$ as in

$$c(x) = c_0 + c_1 x + \cdots + c_{n-1} x^{n-1}$$

- $\overline{c} \cdot \mathbf{H}^T = \overline{0}$ implies

$$c(\sigma^1) = c(\sigma^2) = \cdots = c(\sigma^{n-k}) = 0$$

$\sigma^1, \sigma^2, \cdots, \sigma^{n-k}$ are roots of RS codeword polynomial $c(x)$.

§ 6.2 Reed-Solomon Codes

- An alternatively encoding
  - Message polynomial $u(x) = u_0 + u_1 x + \cdots + u_{k-1} x^{k-1}$
  - Codeword polynomial $c(x) = c_0 + c_1 x + \cdots + c_{n-1} x^{n-1}$
  - $c(x) = u(x) \cdot g(x)$ and deg(g(x)) = n - k
  - Since $\sigma^1, \sigma^2, \cdots, \sigma^{n-k}$ are roots of $c(x)$

    $$g(x) = (x - \sigma^1)(x - \sigma^2) \cdots (x - \sigma^{n-k})$$

    ↑ The generator polynomial of an (n, k) RS code
  - Systematic encoding

    $$c(x) = x^{n-k} u(x) + (x^{n-k} u(x)) \bmod g(x)$$

- **Example 6.5:** For a (7, 3) RS code, its generator polynomial is

  $$g(x) = (x - \sigma^1)(x - \sigma^2)(x - \sigma^3)(x - \sigma^4) = x^4 + \sigma^3 x^3 + x^2 + \sigma x + \sigma^3$$

  Given message vector $\bar{u} = (u_0, u_1, u_2) = (\sigma^4, 1, \sigma^5)$, the codeword can be generated by $c(x) = u(x) \cdot g(x) = 1 + \sigma^2 x + \sigma^4 x^2 + \sigma^6 x^3 + \sigma x^4 + \sigma^3 x^5 + \sigma^5 x^6$.

  For systematic encoding, $\left(x^{n-k} u(x)\right) \bmod g(x) = \left(x^4 \cdot u(x)\right) \bmod g(x) = x^3 + \sigma^4 x + \sigma^5$,

  and the codeword is $\bar{c} = (\sigma^5, \sigma^4, 0, 1, \sigma^4, 1, \sigma^5)$

§ 6.3 Syndrome Based Decoding

– The channel: $r(x) = c(x) + e(x)$

$c(x) = c_0 + c_1 x + \cdots + c_{n-1} x^{n-1}$ – codeword polynomial

$e(x) = e_0 + e_1 x + \cdots + e_{n-1} x^{n-1}$ – error polynomial

$r(x) = r_0 + r_1 x + \cdots + r_{n-1} x^{n-1}$ – received word polynomial

– Let $n - k = 2t$, $\sigma^1, \sigma^2, \cdots, \sigma^{2t}$ are roots of $c(x)$

– $2t$ syndromes can be determined as

$$S_1 = r(\sigma^1), S_2 = r(\sigma^2), \cdots, S_{2t} = r(\sigma^{2t})$$

If $S_1 = S_2 = \cdots = S_{2t} = 0$, $r(x)$ is a valid codeword. Otherwise, $e(x) \neq 0$, error-correction is needed.

§ 6.3 Syndrome Based Decoding

– If $e(x) \neq 0$, we assume there are ω errors with $e_{j_1} \neq 0, e_{j_2} \neq 0, \cdots, e_{j_\omega} \neq 0$. $e(x) = e_{j_1}x^{j_1} + e_{j_2}x^{j_2} + \cdots + e_{j_\omega}x^{j_\omega}$.

– Let $v = 1, 2, \cdots, 2t$

$$S_v = \sum_{j=0}^{n-1} c_j \sigma^{jv} + \sum_{j=0}^{n-1} e_j \sigma^{jv} = \sum_{j=0}^{n-1} e_j \sigma^{jv} = \sum_{\tau=1}^{\omega} e_{j_\tau} (\sigma^{j_\tau})^v$$

– For simplicity, let $X_\tau = \sigma^{j_\tau}$, we can list the $2t$ syndromes by

$$S_1 = e_{j_1} X_1^1 + e_{j_2} X_2^1 + \cdots + e_{j_\omega} X_\omega^1$$
$$S_2 = e_{j_1} X_1^2 + e_{j_2} X_2^2 + \cdots + e_{j_\omega} X_\omega^2$$
$$\vdots$$
$$S_{2t} = e_{j_1} X_1^{2t} + e_{j_2} X_2^{2t} + \cdots + e_{j_\omega} X_\omega^{2t}$$

– In the above non-linear equation group, $X_1, X_2, \cdots, X_\omega$ tell the error locations and $e_{j_1}, e_{j_2}, \cdots, e_{j_\omega}$ tell the error magnitudes. There are 2ω unknowns. It will be solvable if $2\omega \leq 2t$. The number of correctable errors is $\omega \leq \frac{n-k}{2}$.

– Since $X_{j_\tau}, e_{j_\tau} \in \mathbf{F}_q \setminus \{0\}$, an exhaustive search solution will have a complexity of $O(n^{2\omega})$.

§ 6.3 Syndrome Based Decoding

– In order to decode an RS code with a polynomial-time complexity, the decoding is decomposed into determining the **error locations** and **error magnitudes**, i.e., $X_1, X_2, \cdots, X_\omega$ and $e_{j_1}, e_{j_2}, \cdots, e_{j_\omega}$, respectively.

– Error locator polynomial

$$\Lambda(x) = \prod_{\tau=1}^{\omega} \left(1 - X_\tau x\right)$$

$$= \Lambda_\omega x^\omega + \Lambda_{\omega-1} x^{\omega-1} + \cdots + \Lambda_1 x + \Lambda_0 \quad (\Lambda_0 = 1)$$

$X_1^{-1} = \sigma^{-j_1}, X_2^{-1} = \sigma^{-j_2}, \cdots, X_\omega^{-1} = \sigma^{-j_\omega}$ are roots of the polynomial such that $\Lambda(X_1^{-1}) = \Lambda(X_2^{-1}) = \cdots = \Lambda(X_\omega^{-1}) = 0$.

– Determine $\Lambda(x)$ by finding out $\Lambda_\omega, \Lambda_{\omega-1}, \cdots,$ and Λ_1, and its roots tell the error locations.

§ 6.3 Syndrome Based Decoding

– How to determine $\Lambda_\omega, \Lambda_{\omega-1}, \cdots,$ and Λ_1?

Since $\Lambda(X_\tau^{-1}) = \Lambda_\omega X_\tau^{-\omega} + \Lambda_{\omega-1} X_\tau^{1-\omega} + \cdots + \Lambda_1 X_\tau^{-1} + \Lambda_0 = 0$

$$\sum_{\tau=1}^{\omega} e_{j_\tau} X_\tau^{v} \Lambda(X_\tau^{-1}) = 0, \text{ for } v = 1, 2, \cdots, 2t$$

$$\Downarrow$$

$$= e_{j_1}\Lambda_\omega X_1^{v-\omega} + e_{j_1}\Lambda_{\omega-1} X_1^{v-\omega+1} + \cdots + e_{j_1}\Lambda_1 X_1^{v-1} + e_{j_1}\Lambda_0 X_1^{v}$$

$$+ e_{j_2}\Lambda_\omega X_2^{v-\omega} + e_{j_2}\Lambda_{\omega-1} X_2^{v-\omega+1} + \cdots + e_{j_2}\Lambda_1 X_2^{v-1} + e_{j_2}\Lambda_0 X_2^{v}$$

$$\vdots$$

$$+ e_{j_\omega}\Lambda_\omega X_\omega^{v-\omega} + e_{j_\omega}\Lambda_{\omega-1} X_\omega^{v-\omega+1} + \cdots + e_{j_\omega}\Lambda_1 X_\omega^{v-1} + e_{j_\omega}\Lambda_0 X_\omega^{v}$$

$$= \Lambda_\omega S_{v-\omega} + \Lambda_{\omega-1} S_{v-\omega+1} + \cdots + \Lambda_1 S_{v-1} + \Lambda_0 S_v$$

$$\boxed{\Lambda_\omega S_{v-\omega} + \Lambda_{\omega-1} S_{v-\omega+1} + \cdots + \Lambda_1 S_{v-1} + \Lambda_0 S_v = 0}$$

– Error locator polynomial can be determined using the syndromes.

§ 6.3 Syndrome Based Decoding

– List all $\Lambda_\omega S_{v-\omega} + \Lambda_{\omega-1} S_{v-\omega+1} + \cdots + \Lambda_1 S_{v-1} + \Lambda_0 S_v = 0$

$v = 1$: $\Lambda_1 S_0 + \Lambda_0 S_1 = \cdot\cdot$

$v = 2$: $\Lambda_2 S_0 + \Lambda_1 S_1 + \Lambda_0 S_2 = \cdot\cdot$

$v = 3$: $\Lambda_3 S_0 + \Lambda_2 S_1 + \Lambda_1 S_2 + \Lambda_0 S_3 = \cdot\cdot$

$\vdots$

$v = \omega$: $\Lambda_\omega S_0 + \Lambda_{\omega-1} S_1 + \cdots + \Lambda_1 S_{\omega-1} + \Lambda_0 S_\omega = \cdot\cdot$

$v = \omega + 1$: $\Lambda_\omega S_1 + \Lambda_{\omega-1} S_2 + \cdots + \Lambda_1 S_\omega + \Lambda_0 S_{\omega+1} = 0$

$v = \omega + 2$: $\Lambda_\omega S_2 + \Lambda_{\omega-1} S_3 + \cdots + \Lambda_1 S_{\omega+1} + \Lambda_0 S_{\omega+2} = 0$

$\vdots$

$v = 2t$: $\Lambda_\omega S_{2t-\omega} + \Lambda_{\omega-1} S_{2t-\omega+1} + \cdots + \Lambda_1 S_{2t-1} + \Lambda_0 S_{2t} = 0$

$$S_v = -\sum_{\tau=1}^{\omega} \Lambda_\tau S_{v-\tau}$$

$$\begin{bmatrix} S_1 & S_2 & \cdots & S_\omega \\ S_2 & S_3 & \cdots & S_{\omega+1} \\ \vdots & \vdots & \ddots & \vdots \\ S_{2t-\omega} & S_{2t-\omega+1} & \cdots & S_{2t-1} \end{bmatrix} \cdot \begin{bmatrix} \Lambda_\omega \\ \Lambda_{\omega-1} \\ \vdots \\ \Lambda_1 \end{bmatrix} = -\begin{bmatrix} S_{\omega+1} \\ S_{\omega+2} \\ \vdots \\ S_{2t} \end{bmatrix}$$

Remark 2:
S_0 is not one of the $n - k$ syndromes.

§ 6.3 Syndrome Based Decoding

– Solving the linear system in finding $\Lambda(x)$ has a complexity of $O(\omega^3)$. It can be facilitated by the Berlekamp-Massey algorithm[2] whose complexity is $O(\omega^2)$.

– The Berlekamp-Massey algorithm can be implemented using the Linear Feedback Shift Register. Its pseudo program is the follows.

The Berlekamp-Massey Algorithm

Input: Syndromes $S_1, S_2, \ldots, S_{2t}$;
Output: $\Lambda(x)$;
Initialization: $r = 0, \ell = 0, z = -1, \Lambda(x) = 1, T(x) = x$;

1: Determine $\Delta = \sum_{i=0}^{\ell} \Lambda_i S_{r-i+1}$;
2: **If** $\Delta = 0$
3: $T(x) = xT(x)$;
4: $r = r + 1$;
5: **If** $r < 2t$
6: Go to **1**;
7: **Else**
8: Terminate the algorithm;
9: **Else**
10: Update $\Lambda^*(x) = \Lambda(x) - \Delta T(x)$;
11: **If** $\ell \geq r - z$
12: $\Lambda(x) = \Lambda^*(x)$;
13: **Else**
14: $\ell^* = r - z$; $z = r - \ell$; $T(x) = \Lambda(x) / \Delta$; $\ell = \ell^*$; $\Lambda(x) = \Lambda^*(x)$;
15: $T(x) = xT(x)$;
16: $r = r + 1$;
17: **If** $r < 2t$
18: Go to **1**;
19: **Else**
20: Terminate the algorithm;

[2] J. L. Massey, "Shift register synthesis and BCH decoding," *IEEE Trans. Inf. Theory*, vol. 15(1), pp. 122-127, 1969.

§ 6.3 Syndrome Based Decoding

– **Example 6.6:** Given the (7, 3) RS codeword generated in **Example 6.5**, after the channel, the received word is

$$\bar{r} = (\sigma^5, \sigma^4, \boxed{\sigma^3}, \sigma^0, \sigma^4, \boxed{\sigma^2}, \sigma^5).$$

With the received word, we can calculate syndromes as

$$S_1 = r(\sigma) = \sigma^0, S_2 = r(\sigma^2) = \sigma^6, S_3 = r(\sigma^3) = \sigma^6, S_4 = r(\sigma^4) = \sigma^0.$$

Running the above Berlekamp-Massey algorithm,we obtain

r	ℓ	z	$\Lambda(x)$	$T(x)$	Δ
0	0	–1	1	x	σ^0
1	1	0	$1-x$	x	σ^2
2	1	0	$1-\sigma^6 x$	x^2	σ
3	2	1	$1-\sigma^6 x-\sigma x^2$	$\sigma^6 x-\sigma^5 x^2$	σ^5
4			$1-\sigma^3 x-x^2$	$\sigma^6 x^2-\sigma^5 x^3$	

Therefore, the error locator polynomial is $\Lambda(x) = 1-\sigma^3 x - x^2$. In $\mathbf{F}_8$, $\sigma^5$and $\sigma^2$are its roots. Therefore, r_2 and r_5 are corrupted.

§ 6.3 Syndrome Based Decoding

– Determine the error magnitudes $e_{j_1}, e_{j_2}, \cdots, e_{j_\omega}$, so that the erroneous symbols can be corrected by

$$c_{j_1} = r_{j_1} - e_{j_1}, \; c_{j_2} = r_{j_2} - e_{j_2}, \; \cdots, \; c_{j_\omega} = r_{j_\omega} - e_{j_\omega}$$

– The syndromes $S_v = \sum_{\tau=1}^{\omega} e_{j_\tau} X_\tau^v$, $v = 1, 2, \cdots, 2t$. Knowing $X_1 = \sigma^{j_1}, X_2 = \sigma^{j_2}, \cdots, X_\omega = \sigma^{j_\omega}$ from the error location polynomial $\Lambda(x)$, the above syndrome definition implies

$$\begin{bmatrix} X_1^1 & X_2^1 & \cdots & X_\omega^1 \\ X_1^2 & X_2^2 & \cdots & X_\omega^2 \\ \vdots & \vdots & \ddots & \vdots \\ X_1^{2t} & X_2^{2t} & \cdots & X_\omega^{2t} \end{bmatrix} \begin{bmatrix} e_{j_1} \\ e_{j_2} \\ \vdots \\ e_{j_\omega} \end{bmatrix} = \begin{bmatrix} S_1 \\ S_2 \\ \vdots \\ S_{2t} \end{bmatrix}$$

– Error magnitudes can be determined from the above set of linear equations.

§ 6.3 Syndrome Based Decoding

– The linear equation set can be efficiently solved using Forney's algorithm.

– Syndrome polynomial

$$S(x) = S_1 + S_2 x + \cdots + S_{2t} x^{2t-1} = \sum_{v=1}^{2t} S_v x^{v-1}$$

– Error evaluation polynomial (The key equation)

$$\Omega(x) = S(x) \cdot \Lambda(x) \bmod x^{2t}$$

– Formal derivative of $\Lambda(x) = \Lambda_\omega x^\omega + \Lambda_{\omega-1} x^{\omega-1} + \cdots + \Lambda_1 x + \Lambda_0$

$$\Lambda'(x) = \underline{\omega \Lambda_\omega} x^{\omega-1} + \underline{(\omega-1)\Lambda_{\omega-1}} x^{\omega-2} + \cdots + \Lambda_1$$

$$\omega \Lambda_\omega \Rightarrow \underbrace{\Lambda_\omega + \Lambda_\omega + \cdots + \Lambda_\omega}_{\omega} \qquad (\omega-1)\Lambda_{\omega-1} \Rightarrow \underbrace{\Lambda_{\omega-1} + \Lambda_{\omega-1} + \cdots + \Lambda_{\omega-1}}_{\omega-1}$$

– Error magnitude e_{j_τ} can be determined by $e_{j_\tau} = -\dfrac{\Omega\left(X_\tau^{-1}\right)}{\Lambda'\left(X_\tau^{-1}\right)}$.

§ 6.3 Syndrome Based Decoding

– **Example 6.7:** Continue from **Example 6.6**,

The syndrome polynomial is $S(x) = S_1 + S_2 x + S_3 x^2 + S_4 x^3 = \sigma^0 + \sigma^6 x + \sigma^6 x^2 + \sigma^0 x^3$.

The error locator polynomial is $\Lambda(x) = 1 - \sigma^3 x - x^2$.

The error evaluation polynomial is $\Omega(x) = S(x) \cdot \Lambda(x) \bmod x^4 = \sigma^4 x + \sigma^0$.

Formal derivative of $\Lambda(x)$ is $\Lambda'(x) = \sigma^3$.

Error magnitudes are

$$e_2 = -\frac{\Omega\left(\sigma^{-2}\right)}{\Lambda'\left(\sigma^{-2}\right)} = \sigma^3 ,$$

$$e_5 = -\frac{\Omega\left(\sigma^{-5}\right)}{\Lambda'\left(\sigma^{-5}\right)} = \sigma^6 .$$

As a result, $c_2 = r_2 - e_2 = 0$, $c_5 = r_5 - e_5 = \sigma^0$.

§ 6.3 Syndrome Based Decoding

– BM decoding performances over AWGN channel with BPSK.