Chapter 5

BCH Codes

1. Introduction

- BCH (Bose – Chaudhuri - Hocquenghem) Codes form a large class of multiple random error-correcting codes.

  They were first discovered by A. Hocquenghem in 1959 and independently by R. C. Bose and D. K. Ray-Chaudhuri in 1960.

- BCH codes are cyclic codes. Only the codes, not the decoding algorithms, were discovered by these early writers.

- The original applications of BCH codes were restricted to binary codes of length $2^m - 1$ for some integer $m$. These were extended later by Gorenstein and Zieler (1961) to the nonbinary codes with symbols from Galois field $GF(q)$. 

The first decoding algorithm for binary BCH codes was devised by Peterson in 1960. Since then, Peterson’s algorithm has been refined by Berlekamp, Massey, Chien, Forney, and many others.

2. Primitive BCH Codes

For any integer \( m \geq 3 \) and \( t < 2^{m-1} \) there exists a primitive BCH code with the following parameters:

\[
\begin{align*}
  n &= 2^m - 1 \\
  n - k &\leq mt \\
  d_{\text{min}} &\geq 2t + 1
\end{align*}
\]  

This code can correct \( t \) or fewer random errors over a span of \( 2^m - 1 \) bit positions.

The code is a \( t \)-error-correcting BCH code.

For example, for \( m=6, t=3 \)

\[
\begin{align*}
  n &= 2^6 - 1 = 63 \\
  n - k &= 6 \times 3 = 18 \\
  d_{\text{min}} &= 2 \times 3 + 1 = 7
\end{align*}
\]

This is a triple-error-correcting \((63, 45)\) BCH code.
3. Generator Polynomial of Binary BCH Codes

- Let $\alpha$ be a primitive element in $\text{GF}(2^m)$.

  For $1 \leq i \leq t$, let $\phi_{2i-1}(x)$ be the minimum polynomial of the field element $\alpha^{2i-1}$.

  The degree of $\phi_{2i-1}(x)$ is $m$ or a factor of $m$.

- The generator polynomial $g(x)$ of a $t$-error-correcting primitive BCH codes of length $2^m - 1$ is given by

  $$g(x) = LCM\{\phi_1(x), \phi_3(x), \ldots, \phi_{2t-1}(x)\}$$  \hspace{1cm} (5-2)

- Note that the degree of $g(x)$ is $mt$ or less.

  Hence the number of parity-check bits; $n-k$, of the code is at most $mt$.

- Example 5.1, 5.2, 5.3, 5.4. (pp. 191-196)

  ($m = 4, m = 5$)
Note that the generator polynomial of the binary BCH code is originally found to be the least common multiple of the minimum polynomials $\phi_1, \phi_2, \cdots, \phi_{2t}$

i.e. $g(x) = LCM\{\phi_1(x), \phi_2(x), \phi_3(x), \cdots, \phi_{2t-1}(x), \phi_{2t}(x)\}$

However, generally, every even power of $\alpha$ in $GF(2^m)$ has the same minimal polynomial as some preceding odd power of $\alpha$ in $GF(2^{2m})$.

As a consequence, the generator polynomial of the $t$-error-correcting binary BCH code can be reduced to

$$g(x) = LCM\{\phi_1(x), \phi_3(x), \cdots, \phi_{2t-1}(x)\}.$$
Example: $m = 4, t = 3$

Let $\alpha$ be a primitive element in $GF(2^4)$ which is constructed based on the primitive polynomial $p(x) = 1 + x + x^4$

$\phi_1(x) = 1 + x + x^4$ corresponding to $\alpha$

$\phi_3(x) = 1 + x + x^2 + x^3 + x^4$ corresponding to $\alpha^3$

$\phi_5(x) = 1 + x + x^2$ corresponding to $\alpha^5$

$g(x) = \text{LCM}\{\phi_1(x), \phi_3(x), \phi_5(x)\}$

$= \phi_1(x)\phi_3(x)\phi_5(x)$

$= 1 + x + x^2 + x^4 + x^5 + x^8 + x^{10}$

The code is a $(15, 5)$ cyclic code.
4. Properties of Binary BCH Codes

- Consider a $t$-error-correcting BCH code of length $n = 2^m - 1$ with generator polynomial $g(x)$.

- $g(x)$ has as roots $\alpha, \alpha^2, \alpha^3, \ldots, \alpha^{2t}$, i.e.
  \[ g(\alpha^i) = 0 \quad \text{for} \quad 1 \leq i \leq 2t \quad (5-3) \]

- Since a code polynomial $c(x)$ is a multiple of $g(x)$, $c(x)$ also has $\alpha, \alpha^2, \ldots, \alpha^{2t}$ as roots, i.e. $c(\alpha^i) = 0$ for $1 \leq i \leq 2t$.

- A polynomial $c(x)$ of degree less than $2^m - 1$ is a code polynomial if and only if it has $\alpha, \alpha^2, \ldots, \alpha^{2t}$ as roots.
5. Decoding of BCH Codes

- Consider a BCH code with $n = 2^m - 1$ and generator polynomial $g(x)$.

- Suppose a code polynomial $c(x) = c_0 + c_1 x + \cdots + c_{n-1} x^{n-1}$ is transmitted.

  Let $r(x) = r_0 + r_1 x + \cdots + r_{n-1} x^{n-1}$ be the received polynomial.

- Then $r(x) = c(x) + e(x)$, where $e(x)$ is the error polynomial.

- To check whether $r(x)$ is a code polynomial or not, we simply test whether $r(\alpha) = r(\alpha^2) = \cdots = r(\alpha^{2t}) = 0$.

  If yes, then $r(x)$ is a code polynomial, otherwise $r(x)$ is not a code polynomial and the presence of errors is detected.

- Decoding procedure

  (1) syndrome computation.

  (2) determination of the error pattern.

  (3) error correction.
6. Syndrome computation

- The syndrome consists of 2t components in GF($2^m$)

$$\bar{S} = (S_1, S_2, \ldots, S_{2t})$$

(5-4)

and $S_i = r(\alpha^i)$ for $1 \leq i \leq 2t$.

- Computation:

  Let $\phi_i(x)$ be the minimum polynomial of $\alpha^i$.

  Dividing $r(x)$ by $\phi_i(x)$, we obtain

  $$r(x) = a(x)\phi_i(x) + b(x)$$

  Then $S_i = b(\alpha^i)$ (5-5)

- $S_i = b(\alpha^i)$ can be obtained by linear feedback shift-register with connection based on $\phi_i(x)$. 
7. Syndrome and Error Pattern

- Since \( r(x)=c(x)+e(x) \)
  
  \[ s_i = r(\alpha^i) = c(\alpha^i) + e(\alpha^i) = e(\alpha^i) \]  
  \[ (5-6) \]

  for \( 1 \leq i \leq 2t \).

  This gives a relationship between the syndrome and the error pattern.

- Suppose \( e(x) \) has \( \nu \) errors \( (\nu \leq t) \) at the locations specified by \( x^{j_1}, x^{j_2}, \ldots, x^{j_\nu} \).

  \[ e(x) = x^{j_1} + x^{j_2} + \cdots + x^{j_\nu} \]  
  \[ (5-7) \]

  where \( 0 \leq j_1 < j_2 < \cdots < j_\nu \leq n-1 \).

- From equations (5-6) & (5-7), we have the following relation between syndrome components and error location:

  \[ S_1 = e(\alpha) = \alpha^{j_1} + \alpha^{j_2} + \cdots + \alpha^{j_\nu} \]
  \[ S_2 = e(\alpha^2) = (\alpha^{j_1})^2 + (\alpha^{j_2})^2 + \cdots + (\alpha^{j_\nu})^2 \]
  \[ \vdots \]
  \[ S_{2t} = e(\alpha^{2t}) = (\alpha^{j_1})^{2t} + (\alpha^{j_2})^{2t} + \cdots + (\alpha^{j_\nu})^{2t} \]  
  \[ (5-8) \]
• It we can solve the 2t equations, we can determine 
  \( \alpha_1^h, \alpha_2^j, \ldots, \alpha_\nu^j \).

• The unknown parameter \( \alpha_j^u = Z_u \) for \( u = 1, 2, \ldots, \nu \) are called 
  the “error location number”.

  When \( \alpha_j^u, 1 \leq u < \nu \) are found, the powers \( j_u, u = 1, 2, \ldots, \nu \)
  give us the error locations in \( e(x) \). These 2t equation of eq. (5-8)
  are known as power-sum symmetric function.
8. Error-Location Polynomial

(Error-Locator Polynomial)

- suppose that \( \nu \leq t \) errors actually occur.

Define error-locator polynomial \( L(z) \) as

\[
L(z) = (1 + Z_1 z)(1 + Z_2 z) \cdots (1 + Z_\nu z)
= \prod_{i=1}^{\nu} (1 + Z_i z)
= \sigma_0 + \sigma_1 z + \sigma_2 z^2 + \cdots + \sigma_\nu z^\nu
\]

where \( \sigma_0 = 1 \).

- \( L(z) \) has \( Z_1^{-1}, Z_2^{-1}, \cdots, Z_\nu^{-1} \) as roots.

Note that \( Z_u = \alpha^{u^*} \).

- If we can determine \( L(z) \) from the syndrome \( \bar{S} = (S_1, S_2, \cdots, S_{2t}) \),

then the roots of \( L(z) \) give us the error-location numbers.
9. Relationship between $\bar{S}$ and $L(z)$

- From eq. (5-9), we find the following relationship between the coefficients of $L(z)$ and the error-locator numbers:

$$
\begin{align*}
\sigma_0 &= 1 \\
\sigma_1 &= Z_1 + Z_2 + \cdots + Z_v \\
\sigma_2 &= Z_1Z_2 + Z_2Z_3 + \cdots + Z_{v-1}Z_v \\
\vdots \\
\sigma_v &= Z_1Z_2\cdots Z_v
\end{align*}
\tag{5-10}
$$

eqeq. (5-10) is called “elementary symmetric functions”.

- From eq. (5-8) and eq. (5-10), we have the following relationship between the syndrome and the coefficients of $L(z)$:

$$
\begin{align*}
S_1 + \sigma_1 &= 0 \\
S_2 + \sigma_1S_1 + 2\sigma_2 &= 0 \\
S_3 + \sigma_1S_2 + \sigma_2S_1 + 3\sigma_3 &= 0 \\
\vdots \\
S_v + \sigma_1S_{v-1} + \sigma_2S_{v-2} + \cdots + \sigma_{v-1}S_1 + v\sigma_v &= 0
\end{align*}
\tag{5-11}
$$

Here for binary case $i\sigma_i = \sigma_i$ when $i$ is odd,

and $i\sigma_i = 0$ otherwise.

- The equations of (5-11) are called the Newton’s identities.
- If we can determine $\sigma_1, \sigma_2, \cdots, \sigma_\nu$ from the Newton's identities, then we can determine the error-location numbers $Z_1, Z_2, \cdots, Z_\nu$, by finding the roots of $L(z)$.

- Note that the Newton's identities in (5-11) can be expressed also in the following single-equation form:

$$S_i + \sigma_1 S_{i-1} + \sigma_2 S_{i-2} + \cdots + \sigma_{i-1} S_1 + i \sigma_i = 0 \quad \text{for} \quad i = 1, 2, \cdots, \nu$$

(5-12)

- Consider the case for \( i > \nu \)

First multiply \( L(z) \) in eq. (5-9) by \( z^{-i} \), we have

\[
z^{-i} L(z) = z^{-i} + \sigma_1 z^{1-i} + \sigma_2 z^{2-i} + \cdots + \sigma_{\nu-1} z^{\nu-1-i} + \sigma_{\nu} z^{\nu-i}
\] (5-13)

- Next substituting the roots of \( L(z) \) (i.e. \( Z_1^{-1}, Z_2^{-1}, \cdots, Z_{\nu}^{-1} \)) into eq. (5-13) produces the following set of equations:

\[
\begin{align*}
Z_1^i + \sigma_1 Z_1^{i-1} + \sigma_2 Z_1^{i-2} + \cdots + \sigma_{\nu-1} Z_1^{i-\nu+1} + \sigma_{\nu} Z_1^{i-\nu} &= 0 \\
Z_2^i + \sigma_1 Z_2^{i-1} + \sigma_2 Z_2^{i-2} + \cdots + \sigma_{\nu-1} Z_2^{i-\nu+1} + \sigma_{\nu} Z_2^{i-\nu} &= 0 \\
& \vdots \\
Z_{\nu}^i + \sigma_1 Z_{\nu}^{i-1} + \sigma_2 Z_{\nu}^{i-2} + \cdots + \sigma_{\nu-1} Z_{\nu}^{i-\nu+1} + \sigma_{\nu} Z_{\nu}^{i-\nu} &= 0
\end{align*}
\] (5-14)

Adding these \( \nu \) equation term by term yield

\[
(Z_1^i + Z_2^i + \cdots + Z_{\nu}^i) + \sigma_1 (Z_1^{i-1} + Z_2^{i-1} + \cdots + Z_{\nu}^{i-1}) + \cdots + \sigma_{\nu} (Z_1^{i-\nu} + Z_2^{i-\nu} + \cdots + Z_{\nu}^{i-\nu}) = 0
\] (5-15)

- Now express eq. (5-15) in terms of syndrome components, then

\[
S_i + \sigma_1 S_{i-1} + \cdots + \sigma_{\nu-1} S_{i-\nu+1} + \sigma_{\nu} S_{i-\nu} = 0 \quad \text{for} \quad i > \nu
\] (5-16)

In particular, for \( i = \nu + 1 \), we obtain

\[
S_{\nu+1} + \sigma_1 S_{\nu} + \cdots + \sigma_{\nu-1} S_2 + \sigma_{\nu} S_1 = 0
\] (5-17)

Thus, the Newton’s identities can be extended to the unknown syndrome \( S_i \) for \( i > \nu \).
From eq. (5-16) & eq. (5-17), we can see that the $\sigma_j$ for $0 \leq j \leq \nu$ are closely related to the syndrome components $S_i$, $1 \leq i \leq \nu+1$.

Thus, $\sigma_j$ can be determined by solving the set of syndrome equations eq. (5-16) & eq. (5-17). Then the error-locations for can be found by solving $\alpha^i_u = Z_u$ for $1 \leq u \leq \nu$ the root of $L(z)$. This $L(z)$ produces an error-pattern $e(x)$ with the minimum number of errors. Hence if $\nu \leq t$ errors occur, $L(z)$ will give the actual error pattern $e(x)$.

Finally, the error-correcting procedure for the binary BCH codes can be outlined as follows:

1. Compute the syndrome components $S_j$, $1 \leq j \leq 2t$, from the received polynomial $r(x)$. [eq. (5-5)]
2. Set each $\sigma_j = 0$ for $\nu + 1 \leq j \leq t$
   and solve the first $\nu$ equations of eq. (5-11) for the $\sigma_j$, $1 \leq j \leq \nu$ in terms of $S_j$.
3. Determine the error-locator polynomial $L(z)$ from these $\sigma_j$. 

in terms of syndrome component $S_j$ for $0 \leq j \leq 2t$.

(4) Find the error-location numbers $Z_1, Z_2, \ldots, Z_\nu$ by solving for the roots of $L(z)$. Use these roots to correct the errors in $r(x)$. 
11. Direct Solutions of Some Simple Cases.

(1) Single-error correction:

\[ \sigma_1 = S_1 \quad L(z) = 1 + S_1 z \quad S_1 \neq 0 \quad S_3 = S_1^3 \]

(2) Double-error correction:

\[ \sigma_1 = S_1 \quad S_1 \neq 0, \quad S_3 \neq S_1^3 \]

\[ \sigma_2 = S_1^{-1}(S_3 + S_1^3) \quad L(z) = 1 + S_1 z + \left[ \frac{S_1^3 + S_3}{S_1} \right] z^2 \]

(3) Triple-error correction:

\[ \sigma_1 = S_1 \quad S_1 \neq 0 \]

\[ \sigma_2 = \frac{S_1^2 S_3 + S_5}{S_1^3 + S_3} \quad S_3 \neq S_1^3 \]

\[ \sigma_3 = (S_1^3 + S_3) + S_1 \sigma_2 \]

Example: \( m = 4, t = 3 \) BCH code over GF(2^4).

The primitive polynomial for \( m = 4 \) is \( \phi(x) = 1 + x + x^4 \)

The minimum polynomials of \( \alpha, \alpha^3 \) and \( \alpha^5 \) are

\[ \phi_1(x) = 1 + x + x^4 \]

\[ \phi_3(x) = 1 + x + x^2 + x^3 + x^4 \]

\[ \phi_5(x) = 1 + x + x^2 \]

\[ n = 2^4 - 1 = 15 \]

\[ g(x) = LCM\{\phi_1(x), \phi_3(x), \phi_5(x)\} \]

\[ = \phi_1 \phi_3 \phi_5 \]

\[ = 1 + x + x^2 + x^4 + x^5 + x^8 + x^{10} \]
The code is a $(15, 5)$ cyclic code.

The generator polynomial $g(x)$ has $\alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^5, \alpha^6$ as roots. The roots $\alpha, \alpha^2$ and $\alpha^4$ have the same polynomial

$$\phi_1(x) = \phi_2(x) = \phi_4(x) = 1 + x + x^4$$

The root $\alpha^3$ and $\alpha^6$ have the same minimum polynomial

$$\phi_3(x) = \phi_6(x) = 1 + x + x^2 + x^3 + x^4$$

The minimum polynomial of $\alpha^5$ is

$$\phi_5(x) = 1 + x + x^2$$

Suppose all-zero code word $\overline{c} = (0 0 0 \cdots 0)$ is transmitted and $r(x) = x^2 + x^5 + x^{12}$ is received.

Dividing $r(x)$ by $\phi_1(x), \phi_3(x)$ and $\phi_5(x)$, respectively, we obtain the remainders:

$$b_1(x) = 1$$
$$b_3(x) = 1 + x + x^3$$
$$b_5(x) = x^2$$

The syndrome components are

$$s_1 = b_1(\alpha) = 1$$
$$s_2 = b_1(\alpha^2) = 1$$
$$s_4 = b_1(\alpha^4) = 1$$
$$s_3 = b_3(\alpha^3) = 1 + \alpha^6 + \alpha^9 = \alpha^{10}$$
$$s_6 = b_3(\alpha^6) = 1 + \alpha^{12} + \alpha^{18} = \alpha^5$$
$$s_5 = b_3(\alpha^5) = \alpha^{10}$$

Hence $\overline{S} = (1, 1, \alpha^{10}, 1, \alpha^{10}, \alpha^5)$. 
Example: (p. 209) consider $t = 2, m = 5$, $(32, 21)$ BCH code

The primitive polynomial for $m = 5$ is

$$\phi(x) = 1 + x^2 + x^5$$

$\alpha, \alpha^2, \alpha^4, \alpha^8$ and $\alpha^{16}$ have the same minimum polynomial

$$\alpha^3, \alpha^6, \alpha^{12}, \alpha^{24}$$ and $\alpha^{17}$ have the same minimum polynomial

$$\phi_3(x) = 1 + x^2 + x^3 + x^4 + x^5$$

$\alpha^5, \alpha^{10}, \alpha^{20}, \alpha^9$ and $\alpha^8$ have the same minimum polynomial

$$\phi_5(x) = 1 + x + x^2 + x^4 + x^5$$

$\alpha^7, \alpha^{14}, \alpha^{28}, \alpha^{25}$ and $\alpha^{19}$ have the same minimum polynomial

$$\phi_7(x) = 1 + x + x^2 + x^3 + x^5$$

$\alpha^{11}, \alpha^{22}, \alpha^{13}, \alpha^{26}$ and $\alpha^{21}$ have the same minimum polynomial

$$\phi_{11}(x) = 1 + x + x^3 + x^4 + x^5$$

The generator polynomial of $(32, 21)$ code is

$$g(x) = \phi_1(x)\phi_3(x) = 1 + x^3 + x^5 + x^6 + x^8 + x^9 + x^{10}$$

The roots of the generator polynomial include $\alpha, \alpha^2, \alpha^3$ and $\alpha^4$

Since $r(x) = x^2 + x^7 + x^8 + x^{11} + x^{12}$
We obtain
\[ S_1 = r(\alpha) = \alpha^7 \]
\[ S_2 = r(\alpha^2) = \alpha^{14} \]
\[ S_3 = r(\alpha^3) = \alpha^8 \]
\[ S_4 = r(\alpha^4) = \alpha^{28} \]

Note here that \( S_4 = (S_2)^2 = (S_1)^4 \)

Then the error-locator polynomial is obtained by the equation
\[ \sigma_1 = \alpha^7 \]
\[ \sigma_2 = \frac{\alpha^8 + (\alpha^7)^3}{\alpha^7} = \alpha^{15} \]
\[ \therefore L(z) = 1 + \alpha^7 z + \alpha^{15} z^2 = (1 + \alpha^5 z)(1 + \alpha^{10} z) \]
\[ \therefore Z_1 = \alpha^5 = \alpha^{31-26} \]
\[ Z_2 = \alpha^{10} = \alpha^{31-21} \]

Thereby indicating errors at 26th and 21st coordinates of \( \vec{r} \)

\[ \therefore c(x) = x^2 + x^5 + x^7 + x^8 + x^{10} + x^{11} + x^{12} \]

(Note \( Z_i = \alpha^r \))
12. Computation of Error-Location Numbers

– Chien Search. (R.T. Chien, 1964)

- A Chien-search circuit is shown in Fig. 5.2.

\[ L(z) = \sigma_0 + \sigma_1 z + \sigma_2 z^2 + \cdots + \sigma_v z^v = \prod_{i=1}^{v} (1 + Z_i z) \]

where \( \sigma_0 = 1 \)

The roots of \( L(z) \) in GF(\( 2^m \)) can be determined by substituting the elements of GF(\( 2^m \)) in \( L(z) \).

If \( L(\alpha^i) = 0 \), then \( \alpha^i \) is the root of \( L(z) \) and \( \alpha^{-i} = \alpha^{m-i} \) is an error-location number.

- To decode the first received digit \( r_{n-1} \), we check whether \( \alpha \) is a root of \( L(z) \). If \( L(\alpha) = 0 \), then \( r_{n-1} \) is erroneous and must be corrected.

  If \( L(\alpha) \neq 0 \), then \( r_{n-1} \) is error-free.

- To decode \( r_{n-i} \), we test whether \( L(\alpha^i) = 0 \). If \( L(\alpha^i) = 0 \), \( r_{n-i} \) is erroneous and must be corrected, otherwise \( r_{n-i} \) is error-free.
$\bar{r} = (r_0, r_1, \cdots, r_{n-1})$

$e_i = \begin{cases} 1, & \text{if } D_i = 1 \\ 0, & \text{otherwise} \end{cases}$

$D_i = \sum_{j=1}^{i} \sigma_j \alpha^j = L(\alpha^i) - 1$

Figure 5.2 Chien search and error-correction for binary code.
13. Peterson-Gorenstein-Zierler Decoding Algorithm

- From equation (5-16) [Chen / Reed eq. (5-38)]

\[ \sigma_v S_{j-v} + \sigma_{v-1} S_{j-v+1} + \cdots + \sigma_1 S_{j-1} = -S_j \quad \text{for} \quad j > v \]

Assuming that \( v = t \), then

\[ \sigma_t S_{j-t} + \sigma_{t-1} S_{j-t+1} + \cdots + \sigma_1 S_{j-1} = -S_j \quad \text{for} \quad j > t \quad (5-18) \]

e.g. \( j = t + 1 \), we have

\[ \sigma_t S_1 + \sigma_{t-1} S_2 + \cdots + \sigma_1 S_t = -S_{t+1} \]

The following matrix equation is obtained for the symmetric function \( \sigma_j \) as follows:

\[ S' \Lambda = \begin{bmatrix}
S_1 & S_2 & \cdots & S_{t-1} & S_t \\
S_2 & S_3 & \cdots & S_t & S_{t+1} \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
S_t & S_{t+1} & S_{2t-2} & S_{2t-1}
\end{bmatrix} \begin{bmatrix}
\sigma_t \\
\sigma_{t-1} \\
\vdots \\
\sigma_1
\end{bmatrix} = \begin{bmatrix}
-S_{t+1} \\
-S_{t+2} \\
\vdots \\
-S_{2t}
\end{bmatrix} \quad (5-19) \]

- It can be shown that \( S' \) is nonsingular if the received word contains exactly \( t \) errors. It also can be shown that \( S' \) is singular if fewer than \( t \) errors occur. If \( S' \) is singular, then the rightmost column and bottom rows can be removed and the determinant of the resulting matrix computed.
This process is repeated until one reaches a non-singular matrix.

The coefficients of the error-locator polynomial are then founded by the use of standard algebraic technique.

- Once the $\nu$ error locations are known, hen we can use the relation between syndrome components and error locations.

[eq. (5-8)]

The syndrome components can be computed by

$$S_j = e(\alpha^j) = \sum_{k=0}^{n-1} e_k \cdot (\alpha^j)^k = \sum_{l=1}^{\nu} e_{i_l} \cdot Z_l^j$$

where $e(x) = e_0 + e_1 x + e_2 x^2 + \cdots + e_{n-1} x^{n-1} = \sum_{k=0}^{n-1} e_k x^k$

Thus we have

$$S_1 = e_{i_1} Z_1 + e_{i_2} Z_2 + \cdots + e_{i_\nu} Z_\nu$$
$$S_2 = e_{i_1} Z_1^2 + e_{i_2} Z_2^2 + \cdots + e_{i_\nu} Z_\nu^2$$
$$\vdots$$
$$S_{2t} = e_{i_1} Z_1^{2t} + e_{i_2} Z_2^{2t} + \cdots + e_{i_\nu} Z_\nu^{2t}$$
The system equations can be reduced to the following matrix form:

\[
D \cdot \bar{e} = \begin{bmatrix}
Z_1 & Z_2 & \cdots & Z_{\nu} \\
Z_1^2 & Z_2^2 & \cdots & Z_{\nu}^2 \\
\vdots & \vdots & \ddots & \vdots \\
Z_1^{\nu} & Z_2^{\nu} & \cdots & Z_{\nu}^{\nu}
\end{bmatrix}
\begin{bmatrix}
e_{i_1} \\
e_{i_2} \\
\vdots \\
e_{i_{\nu}}
\end{bmatrix}
= \begin{bmatrix}
S_1 \\
S_2 \\
\vdots \\
S_{\nu}
\end{bmatrix}
\tag{5-20}
\]

Decoding is completed by solving for the \( \{e_{i_i}\} \)

This is a general case of nonbinary BCH codes.

Example (5.9)

Consider a nonbinary BCH code \((7, 3)\) of length 7 (symbols) this code is constructed over \(GF(8)\) with generator polynomial

\[
g(x) = (x - \alpha)(x - \alpha^2)(x - \alpha^3)(x - \alpha^4) = x^4 + \alpha^3 x^3 + x^2 + \alpha^3
\]

Let the received polynomial be

\[
r(x) = \alpha^2 x^6 + \alpha^2 x^4 + x^3 + \alpha^5 x^2
\]

Then the syndrome components are

\[
S_1 = \alpha^6, \ S_2 = \alpha^3, \ S_3 = \alpha^4, \ S_4 = \alpha^3
\]
Eq. (5-19) gives

\[ S' \Lambda = \begin{bmatrix} \alpha^6 & \alpha^3 \\ \alpha^3 & \alpha^4 \end{bmatrix} \begin{bmatrix} \sigma_2 \\ \sigma_1 \end{bmatrix} = \begin{bmatrix} \alpha^4 \\ \alpha^3 \end{bmatrix} \]

\[ \Rightarrow \sigma_1 = \alpha^2, \quad \sigma_2 = \alpha \]

Thus \( L(x) = \alpha x^2 + \alpha^2 x + 1 \)

The error locations are founded to be

\[ Z_1 = \alpha^3 = \alpha^{7-4}, \quad Z_2 = \alpha^5 = \alpha^{7-2} \]

\[ \Rightarrow e(x) = e_3 x^3 + e_5 x^5 \]

Eq. (5-20) gives

\[ D \cdot \bar{e} = \begin{bmatrix} \alpha^3 & \alpha^5 \\ \alpha^6 & \alpha^3 \end{bmatrix} \begin{bmatrix} e_3 \\ e_5 \end{bmatrix} = \begin{bmatrix} \alpha^6 \\ \alpha^3 \end{bmatrix} \]

The error magnitudes are found to be

\[ e_3 = \alpha, \quad e_5 = \alpha^5 \]

\[ \Rightarrow e(x) = \alpha x^3 + \alpha^5 x^5 \]

Finally

\[ c(x) = e(x) + r(x) = \alpha^2 x^6 + \alpha^5 x^5 + \alpha^2 x^4 + \alpha^3 x^3 + \alpha^5 x^2 \]

Note that \( m = 3 \), the primitive polynomial is

\[ p(x) = 1 + x + x^3 \]
14. BCH Codes as Industry Standards

(a) (511, 493) BCH code in ITU-T. Rec. H.261 “video codec for audiovisual service at kbit/s” a video coding a standard used for video conferencing and video phone.

\[ n = 511 \quad m = 9 \]
\[ k = 493 \quad n - k = 18 \]
\[ t = 2 \]

(b) (40, 32) BCH code in ATM (Asynchronous Transfer Mode) pp. 223-227.

This is shortened cyclic code that can correct 1-bit error and detect 2-bit errors.