The Mattson-Solomon Polynomials

- $a = (a_0, a_1, \ldots, a_{n-1}) \sim a(x) = a_0 + a_1 x + \cdots + a_{n-1} x^{n-1}$.
- MS poly. associated with $a \to A(z)$

$\cancel{a(x) \in GF(q^m)[x]/(x^n-1)}$

$a = (a_0, a_1, \ldots, a_{n-1})$, $a_i \in GF(q^m)$, $A(z) \in GF(q^m)[z]$,

$$A(z) = \sum_{j=1}^{n} A_j z^{n-j}$$

$\alpha \in GF(q^m)$ is a primitive n-th root of unity.

where $A_j = a(\alpha^j) = \sum_{i=0}^{n-1} a_i \alpha^{ij}$, $j = 0, \pm 1, \pm 2, \ldots$

($A(z)$ is not taken modulo $z^n - 1$)

- Alternative forms of $A(z)$ are:

$$A(z) = \sum_{j=0}^{n-1} A_{-j} z^j \qquad n - j = j'$$

$$= \sum_{i=0}^{n-1} a_i \sum_{j=0}^{n-1} (\alpha^{-i} z)^j$$

- $A_j = A_{j \bmod n}$
- $A(z)$ sometimes called discrete Fourier Transform of a.

__Remarks__ a) The co-efficients A_j are given by

$$\begin{bmatrix} A_0 \\ A_1 \\ \vdots \\ A_{n-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \alpha & \alpha^2 & \cdots & \alpha^{n-1} \\ 1 & \alpha^2 & \alpha^4 & \cdots & \alpha^{2(n-1)} \\ \cdots \\ 1 & \alpha^{n-1} & \alpha^{2(n-1)} & \cdots & \alpha^{(n-1)^2} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{bmatrix}$$

$\delta = 2t+1$ or $\delta = 2t$

b) if $a_i \in GF(q)$, then $(A_j)^q = A_{jq}$ (subscript mod n)

c) A __narrow-sense BCH code of designated distance__ $\underline{\delta}$ can now be defined as all vectors a for which

$A_1 = A_2 = \cdots = A_{\delta-1} = 0$ $\left(a(\alpha^1) = 0 = a(\alpha^2) = \cdots = a(\alpha^{\delta-1}) \right)$

$g(x) = \text{lcm.}\{M^{(1)}(x), M^{(2)}(x), \ldots, M^{(\delta-1)}(x)\}$

Theorem (Inversion formula) $A(z) = \sum_{i=0}^{n-1} a_i \sum_{j=0}^{n-1} (\alpha^{-i} z)^j$

The vector a is recovered from $A(z)$ by

- $\boxed{a_i = \frac{1}{n} A(\alpha^i) \quad , i = 0, 1, \ldots, n-1}$?
- $a = \frac{1}{n}\left(A(1), A(\alpha), \ldots A(\alpha^{n-1})\right)$
- $a(x) = \frac{1}{n} \sum_{i=0}^{n-1} A(\alpha^i) x^i$

if $c(x) = \sum_{i=0}^{n-1} c_i x^i$

then $c_i = \frac{1}{n} \sum_{j=0}^{n-1} c(\alpha^j) \alpha^{-ij}$

where α is a zero of $x^n - 1$

$A(z) = \sum_{j=0}^{n-1} A_{-j} z^j$

$A_{-j} = \frac{1}{n} \sum_{i=0}^{n-1} A(\alpha^i) \alpha^{-ji}$

$A_j = a(\alpha^j) = \sum_{i=0}^{n-1} a_i (\alpha^j)^i$

$\therefore a_i = \frac{1}{n} \sum$

X Notation

- $f(y)$ mod $y^n - 1$

$\rightarrow [f(y)]_n$ (remainder)

- $f(y) = \sum_{i=0}^{n-1} f_i y^i \quad , \quad g(y) = \sum_{i=0} g_i y^i$

- $f(y) * g(y) = \sum_{i=0}^{n-1} f_i g_i y^i$

(Componentwise product)

X Theorem (properties of MS polys). (omit proof)

i) if $c(x) = a(x) + b(x)$ then $C(z) = A(z) + B(z)$

$C(z) \rightarrow$ MS poly for $c(x)$

$A(z), B(z) \rightarrow$ MS poly for $a(x), b(x)$ resp'ly.

ii) $c(x) = [a(x) b(x)]_n$ iff $C(z) = A(z) * B(z)$

iii) $c(x) = a(x) * b(x)$ iff $C(z) = \frac{1}{n} [A(z) B(z)]_n$

iv) The MS poly of a cyclic shift of a is $A(\alpha z)$

v) The MS poly of $0 \rightarrow 0$, $1 \rightarrow 1$

vi) An overall parity check on a is given by $\sum_{i=0}^{n-1} a_i = A(0) = A_0$

- $A(z) \to$ MS poly $\quad a = \sum_{i=0}^{n-1} a_i x^i \sim (a_0, a_1, \dots, a_{n-1})$

$$A(z) = \sum_{i=1}^{n} A_i z^{n-i} \quad , \qquad A_i = a(\alpha^i)$$

$$= \sum_{i=0}^{n-1} A_{-i} z^i \qquad \text{subscript mod } n$$

- <u>inversion formula</u>

$$a_j = \frac{1}{n} A(\alpha^j) \ , \quad j = 0, 1, \dots, n-1 .$$

<u>proof</u>

$$A(\alpha^j) = \sum_{i=1}^{n} a(\alpha^i)\, \alpha^{(n-i)j}$$

$$= \sum_{i=1}^{n} \sum_{k=0}^{n-1} a_k \alpha^{ik} \cdot \alpha^{(n-i)j}$$

$$= \sum_{k=0}^{n-1} a_k \sum_{i=1}^{n} \alpha^{ik + (n-i)j}$$

$$k = j \to \quad a_j \sum \alpha$$

$\alpha \to$ n-th root of unity

$$\sum_{i=0}^{n-1} \alpha^i = \begin{cases} 0 & \text{if } \alpha \neq 1 \\ n & \text{if } \alpha = 1 \end{cases}$$

$$A(\alpha^j) = \sum_{i=0}^{n-1} A_{-i}\, \alpha^{ji}$$

$$= \sum_{i=0}^{n-1} a(\alpha^{-i})\, \alpha^{ji}$$

$$= \sum_{i=0}^{n-1} \sum_{k=0}^{n-1} a_k \alpha^{-ik} \cdot \alpha^{ji}$$

$$= \sum_{k=0}^{n-1} a_k \sum_{i=0}^{n-1} \alpha^{(j-k)i} = a_j \sum_{i=0}^{n-1} \alpha^0 + \sum_{\substack{k=0 \\ k \neq j}}^{n-1} a_k \left(\sum_{i=0}^{n-1} (\alpha^{j-k})^i \right)$$

$$= n a_j$$

$$\frac{1 - (\alpha^{j-k})^n}{1 - \alpha} = 0 \quad \text{as } \alpha^n = 1$$

proof of (ii) X

. if $c(x) = [a(x)\, b(x)]_n$ then $C_j = c(\alpha^j) = a(\alpha^j)\, b(\alpha^j)$
$= A_j B_j$

$\therefore$ $C(z) = A(z) * B(z)$
$= \sum_{j=0}^{n-1} A_j B_j z^j$

. Conversely, let $C_j = A_j B_j \ \forall j$ i.e. $C(z) = A(z) * B(z)$

Claim $\sum_{k=0}^{n-1} c_k x^k = \left(\sum_{i=0}^{n-1} a_i x^i\right)\left(\sum_{I=0}^{n-1} b_I x^I\right)$

$$\text{RHS} = \left(\sum_{i=0}^{n-1}\left\{\frac{1}{n}\sum_{j=0}^{n-1} a(\alpha^j)\,\alpha^{-ij}\right\} x^i\right) \times \left(\sum_{I=0}^{n-1}\left\{\sum_{J=0}^{n-1} b(\alpha^J)\,\alpha^{-IJ}\right\} x^I\right)$$

mod $x^n - 1$

$a(x) = \sum_{i=0}^{n-1} a_i x^i$
$\Rightarrow a_i = \frac{1}{n}\sum_{j=0}^{n-1} a(\alpha^j)\,\alpha^{-ij}$
when α is the n-th root of unity
i.e. $\alpha \in GF(q^m)$ is a zero of $x^n - 1$

Co-efficient of x^k in

$$\sum_{i=0}^{n-1}\left(\frac{1}{n}\sum_{j=0}^{n-1} a(\alpha^j)\,\alpha^{-ij}\right)\left(\sum_{J=0}^{n-1} b(\alpha^J)\,\alpha^{-(n+k-i)J}\right)$$

$\cancel{\sum d_i x^i \sum d_i x^i}$
$(d_0 + d_1 x + \cdots + d_{n-1}x^{n-1})$
$\times\, (e_0 + e_1 x + \cdots + e_{n-1}x^{n-1})$
x^k
$x^i \cdot x^{k-i}$
$\sum_{i=0}^{n-1} d_i e_{k-i}$
$d_0 e_0 \to 1$
$d_0 e_1 + d_1 e_0 \to x$
$d_0 e_2 + d_1 e_1 + d_2 e_0 \to x^2$
x^k
$d_0 e_k + d_1$

$$= \frac{1}{n^2}\sum_{j=0}^{n-1} a(\alpha^j)\sum_{J=0}^{n-1} b(\alpha^J)\,\alpha^{-(n+k)J} \times \underbrace{\sum_{i=0}^{n-1}\alpha^{i(J-j)}}_{0 \text{ unless } J=j}$$

$\sum_{i=0}^{n-1}\alpha^i = \begin{cases} 0 & \text{if } \alpha \neq 1 \\ n & \text{if } \alpha = 1 \end{cases}$

$$= \frac{1}{n}\sum_{j=0}^{n-1} a(\alpha^j)\, b(\alpha^j)\,\alpha^{-(n+k)j}$$

$$= \frac{1}{n}\sum_{j=0}^{n-1} a(\alpha^j)\, b(\alpha^j)\,\alpha^{-kj} = \frac{1}{n}\sum_{j=0}^{n-1} A_j B_j\,\alpha^{-kj}$$

$$= \frac{1}{n}\sum_{j=0}^{n-1} C_j\,\alpha^{-kj} = \frac{1}{n}\, C(\alpha^{-k}) = c_k$$

locator polynomial

- $a = (a_0, a_1, \dots, a_{n-1})$, $a_i \in F$ has non-zero components $= GF(q)$

  $a_{i_1}, a_{i_2}, \dots, a_{i_\omega}$ & no others.

  i.e. $wt(a) = \omega$

- We associate with a the following elements of $GF(q^m)$:

  $X_1 = \alpha^{i_1}, X_2 = \alpha^{i_2}, \dots, X_\omega = \alpha^{i_\omega}$,

  called the <u>locators of a</u> and following elements of $GF(q)$

  $Y_1 = a_{i_1}, Y_2 = a_{i_2}, \dots, Y_\omega = a_{i_\omega}$,

  giving <u>the values of the non-zero components</u>.

- Thus a is completely specified by

  $(X_1, Y_1), (X_2, Y_2), \dots, (X_\omega, Y_\omega)$.

- In case a is a binary vector, all Y_i's are 1.

- Note that $a(\alpha^j) = A_j = \sum_{i=1}^{\omega} Y_i X_i^j$

$$a_0 + a_1\alpha^j + a_2\alpha^{2j} \dots + a_{n-1}\alpha^{(n-1)j}$$

$$a_{i_1}(\alpha^j)^{i_1} + a_{i_2}(\alpha^j)^{i_2} + \dots + a_{i_\omega}(\alpha^j)^{i_\omega}$$

$$Y_1 X_1^j + Y_2 X_2^j + \dots + Y_\omega X_\omega^j$$

Defn The locator polynomial of the vector a is

$$\sigma(z) = \prod_{i=1}^{\omega}(1 - X_i z) = \sum_{i=0}^{\omega} \sigma_i z^i, \quad \sigma_0 = 1$$

- The roots of $\sigma(z)$ are the reciprocals of the locators
- The coefficients σ_i are the elementary symmetric functions of the X_i:

$$\sigma_1 = -(X_1 + X_2 + \cdots + X_\omega)$$

$$\sigma_2 = X_1X_2 + X_1X_3 + \cdots + X_{\omega-1}X_\omega$$

$$\cdots$$

$$\sigma_\omega = (-1)^\omega X_1 \cdots X_\omega .$$

Generalized Newton identities

The A_i's and the σ_i's are related by a set of simultaneous linear equations.

Theorem $\forall j$, the A_i's satisfy the linear recurrence

$$(1) \quad \boxed{A_{j+\omega} + \sigma_1 A_{j+\omega-1} + \cdots + \sigma_\omega A_j = 0.}$$

In particular, taking $j = 1, 2, \ldots, \omega$

$$\begin{pmatrix} A_\omega & A_{\omega-1} & \cdots & A_1 \\ A_{\omega+1} & A_\omega & \cdots & A_2 \\ \vdots & & & \\ A_{2\omega-1} & A_{2\omega-2} & \cdots & A_\omega \end{pmatrix} \begin{pmatrix} \sigma_1 \\ \sigma_2 \\ \vdots \\ \sigma_\omega \end{pmatrix} = - \begin{pmatrix} A_{\omega+1} \\ A_{\omega+2} \\ \vdots \\ A_{2\omega} \end{pmatrix}$$

proof of (i)

$$\sigma(z) = \prod_{i=1}^{\omega} (1 - X_i z) = \sigma_0 + \sigma_1 z + \sigma_2 z^2 + \cdots + \sigma_\omega z^\omega$$

put $z = \frac{1}{X_i}$

$\times Y_i$ $$0 = Y_i \sigma_0 + \frac{\sigma_1 Y_i}{X_i} + \frac{\sigma_2 Y_i}{X_i^2} + \cdots + \frac{\sigma_\omega Y_i}{X_i^\omega} \quad \times Y_i X_i^{j+\omega}$$ + sum $i=1$ to ω

$\times X_i^{j+\omega}$ sum $$0 = \sum_{i=1}^{\omega} Y_i X_i^{j+\omega} \sigma_0 + \sum_{i=1}^{\omega} Y_i X_i^{j+\omega-1} \sigma_1 + \sum_{i=1}^{\omega} Y_i X_i^{j+\omega-2} \sigma_2 + \cdots + \sum_{i=1}^{\omega} Y_i X_i^{j} \sigma_\omega$$

$$= \sigma_0 A_{j+\omega} + \sigma_1 A_{j+\omega-1} + \sigma_2 A_{j+\omega-2} + \cdots + \sigma_\omega A_j$$

$\sum_{i=1}^{\omega} Y_i X_i^j = A_j$

$\sigma_0 = 1.$

Exercise Let a be a vector of weight ω.

Show that

$$\begin{pmatrix} A_v & A_{v-1} & \cdots & A_1 \\ A_{v+1} & A_v & \cdots & A_2 \\ \cdots & & & \\ A_{2v-1} & A_{2v-2} & \cdots & A_v \end{pmatrix}_{v \times v}$$

is non singular if $v = \omega$, but singular if $v > \omega$.

$\omega = v$ $\qquad$ $\omega < v$

The usual form of Newton's identities

Let $X_1, X_2, \ldots, X_\omega$ be indeterminates, and

$$\sigma(z) = \prod_{i=1}^{\omega} (1 - X_i z) = \sum_{i=0}^{\omega} \sigma_i z^i$$

When σ_i's are elementary symm. fn's. of X_i's, $\sigma_0 = 1$ & $\sigma_i = 0$ for $i > \omega$

Define the power sum

$$P_i = \sum_{r=1}^{\omega} X_r^i \quad \forall i.$$

Thm (a) if $P(z) = \sum_{i=1}^{\infty} P_i z^i$, show that

$$\sigma(z) P(z) + z\sigma'(z) = 0.$$

(b) By equating coefficients, show that

$$P_1 + \sigma_1 = 0$$

$$P_2 + \sigma_1 P_1 + 2\sigma_2 = 0$$

$$\vdots$$

$$P_\omega + \sigma_1 P_{\omega-1} + \cdots + \sigma_{\omega-1} P_1 + \omega\sigma_\omega = 0.$$

and, for $i > \omega$

(2) agrees with (1) when all $Y_i = 1$ (i.e. binary case)

$$\boxed{P_i + \sigma_1 P_{i-1} + \cdots + \sigma_\omega P_{i-\omega} = 0} \quad \text{(2)}$$

Exercise

In the binary case, the above equations imply

$$\underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & \cdots & 0 \\ A_2 & A_1 & 1 & 0 & 0 & \cdots & 0 \\ A_4 & A_3 & A_2 & A_1 & 1 & \cdots & 0 \\ \cdots & & & & & & \\ A_{2\omega-2} & A_{2\omega-3} & \cdots & & & & A_{\omega-1} \end{pmatrix}}_{M_\omega} \begin{pmatrix} \sigma_1 \\ \sigma_2 \\ \vdots \\ \sigma_\omega \end{pmatrix} = \begin{pmatrix} A_1 \\ A_3 \\ A_5 \\ \vdots \\ A_{2\omega-1} \end{pmatrix} \quad (3).$$

Exercise: M_v nonsingular if $\cancel{v = \omega}$ $\omega = v$ or $v-1$

M_v singular if $\omega < v-1$

Applications to decoding BCH codes

- BCH code with designated distance δ.
- $a = (a_0, a_1, \ldots, a_{n-1}) \rightarrow$ error vector
- decoder can easily calculate MS $A_1, A_2, \ldots, A_{\delta-1}$ & finds $\sigma(z)$ using (hard) Newton identities.

$Y_1 = a_{i_1}$, $Y_2 = a_{i_2}$, $\ldots$, $Y_\omega = a_{i_\omega}$ $\rightarrow$ errors value

~~$X_1 = \alpha^{a_{i_1}}$, $X_2 = \alpha^{a_{i_2}}$, $\ldots$, $X_\omega = \alpha^{a_{i_\omega}}$~~

$X_1 = \alpha^{i_1}$, $X_2 = \alpha^{i_2}$, $\ldots$, $X_\omega = \alpha^{i_\omega}$ $\rightarrow$ error locators

$i_1, i_2, \ldots, i_\omega \rightarrow$ positions where errors occur.

Then $A_j = \sum_{i=1}^{\omega} X_i Y_i^j$, $\sigma(z) = \prod (1 - X_i z)$ $\rightarrow$ find zeros

error locator poly.

To determine the Y_i's, define the <u>evaluator polynomial</u>

$$\omega(z) = \sigma(z) + \sum_{k=1}^{\omega} z X_k Y_k \prod_{\substack{j=1\\ j\neq k}}^{\omega} (1 - X_j z)$$

$$\omega(X_i^{-1}) = \underset{0}{\sigma(X_i^{-1})} + X_i^{-1} X_i Y_i \prod_{\substack{j=1\\ j\neq i}}^{\omega} (1 - X_j X_i^{-1}) + \sum_{\substack{k=1\\ k\neq i}}^{\omega} X_i^{-1} X_k Y_k \prod_{\substack{j=1\\ j\neq k}}^{\omega} (\ )$$

$$1 - X_\ell X_i^{-1} = 0$$

Once $\omega(z)$ is known, Y_i is given by

$$\sigma(z) = \prod_{j=1}^{\omega}(1 - X_j z)$$

$$\sigma'(X_i^{-1}) = \prod_{\substack{j=1\\ j\neq i}}^{\omega}(1 - X_j X_i^{-1})(-X_i)$$

$$Y_i = \omega(X_i^{-1}) \Big/ \prod_{j\neq i}(1 - X_j X_i^{-1})$$

$$= -X_i\, \omega(X_i^{-1}) / \sigma'(X_i^{-1}).$$

<u>Theorem</u> $\omega(z) = (1 + S(z))\,\sigma(z)$ — (3)

where $S(z) = \sum_{i=1}^{\infty} A_i z^i$

Note that $\deg \omega(z) \le \deg \sigma(z) = \omega$, only $A_1, A_2, \ldots, A_\omega$ are needed to determine $\omega(z)$ from (3).

<u>Proof</u>

$$\frac{\omega(z)}{\sigma(z)} = 1 + \sum_{i=1}^{\omega} \frac{z X_i Y_i}{1 - X_i z} \qquad A_j = \sum_{r=1}^{\omega} X_r Y_r^{j}$$

$$= 1 + \sum_{i=1}^{\omega} Y_i \sum_{j=1}^{\infty} (z X_i)^j$$

$$= 1 + \sum_{j=1}^{\infty} A_j z^j = 1 + S(z).$$

Theorem The weight of a is $n-r$, where r is the no. of nth roots of unity which are zeros of the MS poly. $A(z)$.

proof $a = \frac{1}{n}\left(A(1), A(\alpha), \dots A(\alpha^{n-1})\right)$.

$wt(a) = n-r$ means $A(\alpha^i) = 0$
for $i = i_1, i_2, \dots, i_r$
$\in \{0, 1, 2, \dots, n-1\}$

$wt(a) = n-r \geq n - \deg A(z)$ i.e. $A(z)$ has r many n-th root of unity as its zero.
$\Rightarrow \deg A(z) \geq r$.
$-r \geq -\deg A(z)$

Corollary If a has MS poly $A(z)$, then

$$wt(a) \geq n - \deg A(z).$$

This gives another proof of the BCH bound.

Theorem (BCH bound)

Let $\mathcal{C}$ be a cyclic code with generator poly $g(x) = \prod_{i \in K} (x - \alpha^i)$, when K contains a string of $d-1$ ~~successive~~ consecutive integers $b, b+1, \dots, b+d-2$, for some b. Then minimum weight of any non-zero codeword $a \in \mathcal{C}$ is at least d.

proof. Since $a \in \mathcal{C}$, $g(x) \mid a(x)$

$\therefore\ a(\alpha^j) = 0$, for $b \le j \le b+d-2$

$$\therefore\ A(z) = \sum_{j=1}^{n} A_j z^{n-j} \qquad A_j = a(\alpha^j)$$

$$= a(\alpha) z^{n-1} + a(\alpha^2) z^{n-2} + \dots + a(\alpha^n)$$

$$= \underset{A_1}{\underset{\parallel}{a(\alpha) z^{n-1}}} + \dots + \underset{A_{b-1}}{\underset{\parallel}{a(\alpha^{b-1})}} z^{n-b+1} + \overset{A_{b+d-1}}{\overset{=}{a(\alpha^{b+d-1})}} z^{n-b-d+1} + \dots + \underset{A_n}{\underset{\parallel}{a(\alpha^n)}}$$

wt $\hat{A}(z) = z^{b-1} A(z) - \{a(\alpha) z^{b-2} + \dots + a(\alpha^{b-1})\}\{z^n - 1\}$

$$= \cancel{a(\alpha) z^{n+b-2}} + \cancel{a(\alpha^2) z^{n+b-1}} + \dots + \cancel{a(\alpha^{b-1}) z^n}$$

$$+ a(\alpha^{b+d-1}) z^{n-d} + \dots + a(\alpha^n) z^{b-1}$$

$$- \{\cancel{a(\alpha) z^{n+b-2}} + \cancel{\dots} + \cancel{a(\alpha^{b-1}) z^n}\}$$

$$+ a(\alpha) z^{b-2} + \dots + a(\alpha^{b-1})$$

$$= \underset{A_{b+d-1}}{\underset{\parallel}{a(\alpha^{b+d-1})}} z^{n-d} + \dots + \underset{A_n}{\underset{\parallel}{a(\alpha^n)}} z^{b-1} + \underset{A_1}{\underset{\parallel}{a(\alpha)}} z^{b-2} + \dots + \underset{A_{b-1}}{\underset{\parallel}{a(\alpha^{b-1})}}$$

of n-th roots of unity which are zeros of $A(z)$

= # of n-th roots of unity which are zeros of $\hat{A}(z)$

$\le$ deg $\hat{A}(z) = n-d$

$a \to$ MS poly $A(z)$, deg $A(z) \le n-d$

$\Rightarrow$ wt$(a) \ge d$.

deg $A(z) = r$

$\Rightarrow$ wt$(a) = n - r$

Decoding of BCH code (binary)

1. Calculate syndrome

$$S = Hy^T = \begin{pmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^{n-1} \\ 1 & \alpha^3 & \alpha^6 & \cdots & \alpha^{3(n-1)} \\ \cdots \\ 1 & \alpha^{\delta-2} & \alpha^{2(\delta-2)} & \cdots & \alpha^{(n-1)(\delta-2)} \end{pmatrix} \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{pmatrix}$$

Received vector

$y = (y_0, y_1, \cdots, y_{n-1})$

$y(x) = y_0 + y_1 x + \cdots + y_{n-1}x^{n-1}$

$$= \begin{pmatrix} y(\alpha) \\ y(\alpha^3) \\ \vdots \\ y(\alpha^{\delta-2}) \end{pmatrix} = \begin{pmatrix} A_1 \\ A_3 \\ \vdots \\ A_{\delta-2} \end{pmatrix}$$

where $A_l = y(\alpha^l)$

Note that $A_{2r} = y(\alpha^{2r}) = y(\alpha)^r = A_r^2$

Alternatively, calculate A_l's as follows:

$M^{(l)}(x) \to$ minimal poly. of α^l.

$y(x) = Q(x)M^{(l)}(x) + R(x)$, $\quad 0 \le \deg R(x) < \deg M^{(l)}(x)$

α^l

Then $A_l = y(\alpha^l) = R(\alpha^l)$

2. Find the error locator poly $\sigma(z)$

$e \to$ error vector, $wt(e) = \omega$

$e = 00\cdots 0 e_{i_1} 0 \cdots e_{i_2} \cdots e_{i_\omega} 0 \cdots 0$

let X_i: $X_1 = \alpha^{i_1}, \; X_2 = \alpha^{i_2}, \cdots X_\omega = \alpha^{i_\omega}$

- $\sigma(z) = \prod_{i=1}^{\omega} (1 - X_i z) = \sum_{i=0}^{\omega} \sigma_i z^i$ $\quad X_r = \alpha^{i_r}$

much harder part

$e(x) = e_{i_1} x^{i_1} + e_{i_2} x^{i_2} + \dots + e_{i_\omega} x^{i_\omega}$

$y(x) = y_0 + y_1 x + \dots + y_{n-1} x^{n-1}$

- σ_i's & A_i's are related by Newton's identities

$A_l = y(\alpha^l) = \cancel{y_0 + y_1}$
$= c(\alpha^l) + e(\alpha^l)$
$= e(\alpha^l)$
$= e_{i_1}(\alpha^l)^{i_1} + e_{i_2}(\alpha^l)^{i_2} + \dots + e_{i_\omega}(\alpha^l)^{i_\omega}$
$= X_{i_1}^l + X_{i_2}^l + \dots + X_{i_\omega}^l$
$= \sum_{r=1}^{\omega} X_r^l$

- $\sigma(z)$ is not uniquely determined.
- the decoder must find the vector e with least weight ω, or the lowest degree $\sigma(z)$ satisfying Newton's identities.
- This uncertainity in ω makes this stage so difficult.

$$
* \leftarrow \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & \cdots & 0 \\ A_2 & A_1 & 1 & 0 & 0 & \cdots & 0 \\ \cdots \\ A_{2\omega-4} & A_{2\omega-5} & & \cdots & & & A_{\omega-3} \\ A_{2\omega-2} & A_{2\omega-3} & & \cdots & & & A_{\omega-1} \end{pmatrix} \begin{pmatrix} \sigma_1 \\ \sigma_2 \\ \vdots \\ \sigma_{\omega-1} \\ \sigma_\omega \end{pmatrix} = \begin{pmatrix} A_1 \\ A_3 \\ A_5 \\ \vdots \\ A_{2\omega-3} \\ A_{2\omega-1} \end{pmatrix}
$$

Theorem let $A_l = \sum_{i=1}^{\omega} x_i^l$ | $A_l = \sum_{i=1}^{\omega} Y_i X_i^l$

binary case $Y_i = 1 \ \forall i$

The $v \times v$ matrix

$$M_v = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \cdots & 0 \\ A_2 & A_1 & 1 & 0 & 0 & \cdots & 0 \\ A_4 & A_3 & A_2 & A_1 & 1 & \cdots & 0 \\ \cdots & \cdots & \cdots & & & & \\ A_{2v-4} & A_{2v-5} & \cdots & & & & A_{v-3} \\ A_{2v-2} & A_{2v-3} & \cdots & & & & A_{v-1} \end{bmatrix}$$

$\bar{M}_{v-1}$

is non singular if $\omega = v$ or $\omega = v-1$, and singular if $\omega < v-1$.

$$\boxed{\det \underset{v \times v}{M_v} = \det \bar{M}_{v-1}}$$

~~proof~~ -

i) $\omega < v-1$

Then $M_v \begin{bmatrix} 0 \\ 1 \\ \sigma_1 \\ \vdots \\ \sigma_{v-2} \end{bmatrix} = 0$ using Newton's identity (binary case)

$\Rightarrow M_v$ is singular

ii) $\omega = v$

Then $\det M_v = \prod_{i<j} (X_i + X_j)$

? For if we put $X_i = X_j$, $\det M_v = 0$ by (i)

$\therefore \det M_v = \text{const.} \prod_{i<j} (X_i + X_j)$

& const. is easily found to be 1

~~$A_1 = X_1 + X_2 + \cdots + X_\omega$~~

$= \begin{cases} 0 & \text{if } \cancel{\omega \text{ even}} \\ X_1 & \cancel{\text{if } \omega \text{ odd}} \end{cases}$

iii) Finally, if $\omega = v-1$, M_v is nonsingular from (ii).

Using this theorem we have the following iteration algm. for finding $\sigma(z)$, for a BCH code of designated distance $\delta = 2t+1$, assuming ω errors occur where $\omega \le t$.

- Assume t errors occurred & try to solve eqns $(*)$ with ω replaced by t.
- By the above theorem, if t or $t-1$ errors have occurred, a soln. exists & we go to next stage.

(3) Finding the roots of $\sigma(z) = \prod_{i=1}^{\omega} (1 - X_i z)$

→ reciprocal of ~~$\sigma(z)$ are~~ zeros of $\sigma(z)$ are $X_1 = \alpha^{a_{i_1}}, X_2 = \alpha^{a_{i_2}}, \ldots, X_\omega = \alpha^{a_{i_\omega}}$ & errors occurred at co-ordinates $i_1, i_2, \ldots, i_\omega$.

→ ~~error values~~ if $\deg \sigma(z) = 1$ or 2 the zeros can be found directly.

→ in general, simplest technique is just to test each power of α in turn to see if it is a zero of $\sigma(z)$. (Chien search)

→ There is an error in co-ordinate i iff $\sigma(\alpha^{-i}) = 0$.

- But if fewer than $t-1$ errors occurred, the eqns. (*) have no soln.
- In this case assume $t-2$ errors occurred, & again try to solve (*) with ω now replaced by $t-2$.
- Repeat until a soln. found.

<u>Difficulty</u> in finding $\sigma(z)$

→ requires repeated evaluation of a large determinant over $GF(2^m)$.

→ if t large (> 3 or 4), this method is not preferred.

→ instead, generalized Newton's identities are used along with Berlekamp algm.

<u>Using the generalized Newton's identities – the Berlekamp algorithm:</u>

Assuming that ω errors occurred.

The σ_i's & A_i's are related by

$$A_{j+\omega} + \sigma_1 A_{j+\omega-1} + \cdots + \sigma_\omega A_j = 0.$$

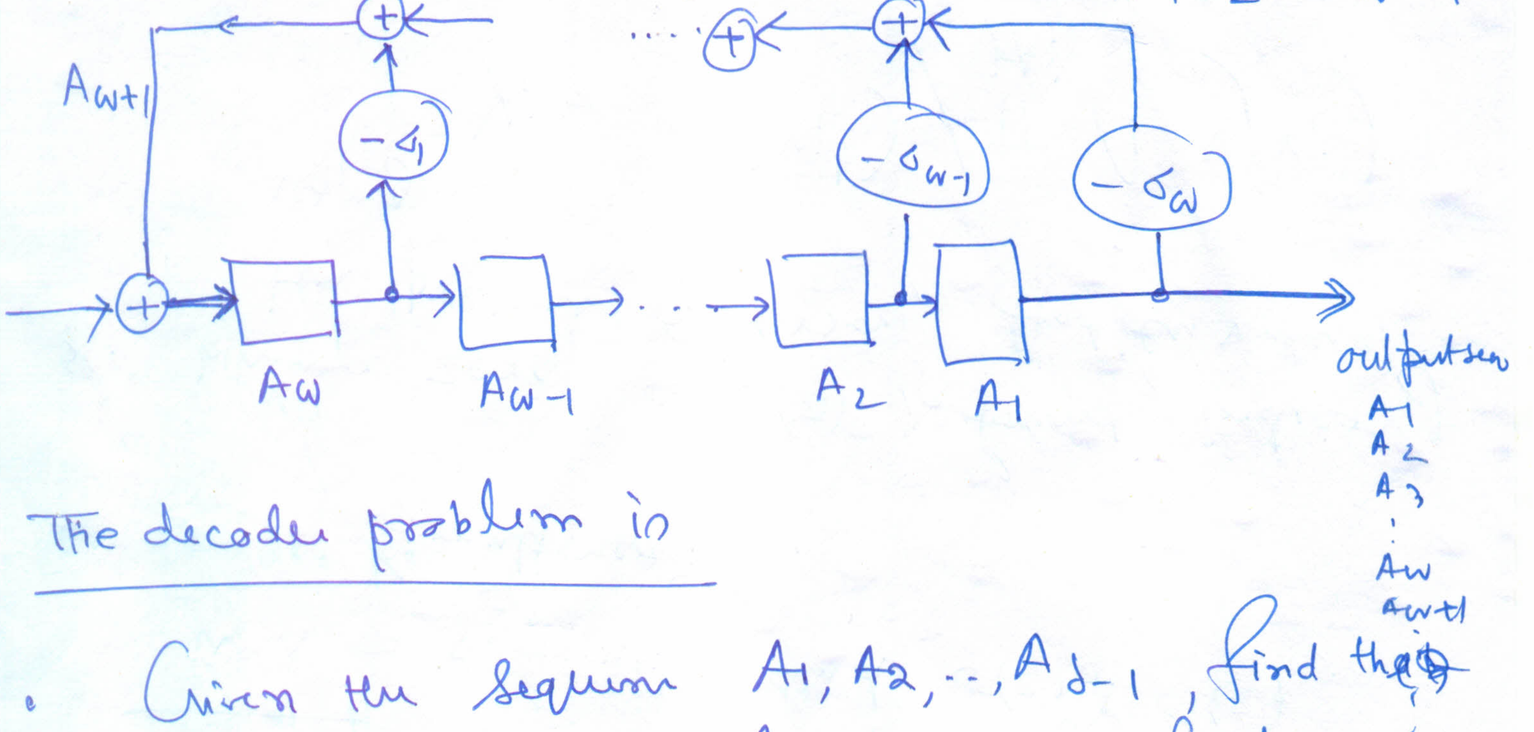
In particular, taking $j = 1, 2, \ldots, \omega$,

$$\begin{pmatrix} A_\omega & A_{\omega-1} & \cdots & A_1 \\ A_{\omega+1} & A_\omega & \cdots & A_2 \\ \cdots & & & \\ A_{2\omega-1} & A_{2\omega-2} & \cdots & A_\omega \end{pmatrix} \begin{pmatrix} \sigma_1 \\ \sigma_2 \\ \vdots \\ \sigma_\omega \end{pmatrix} = -\begin{pmatrix} A_{\omega+1} \\ A_{\omega+2} \\ \vdots \\ A_{2\omega} \end{pmatrix}$$

- this can be interpreted as

A_l's are output from a LFSR of ω stages, with initial contents $A_1, A_2, \ldots, A_\omega$.

$$A_{\omega+1} = -\sigma_1 A_\omega - \sigma_2 A_{\omega-1} - \cdots - \sigma_{\omega-1} A_2 - \sigma_\omega A_1$$



The decoder problem is

- Given the sequence $A_1, A_2, \ldots, A_{\delta-1}$, find the LFSR of shortest length ω which produces $A_1, \ldots, A_{\delta-1}$ as output when initially loaded with $A_1, A_2, \ldots, A_\omega$.

- There is an efficient algm. for finding such an LFSR (& hence error locator poly $\sigma(z)$) which is due to Berlekamp.

Decoding of non-binary BCH code

1. Decoder finds $A_1, \ldots, A_{\delta-1}$.
2. The generalized Newton's identities are used to find $\sigma(z)$ & the error evaluator poly

$$\omega(z) = (1 + S(z)) \sigma(z).$$

where $S(z) = \sum_{i=1}^{\infty} A_i z^i$

Alternatively, Berlekamp's algm. can be used to efficiently find $\sigma(z)$ & $\omega(z)$ simultaneously.

3. When a zero of $\sigma(z)$ is found, indicating the presence of error, find the value of error using

$$Y_i = \omega(X_i^{-1}) / \prod_{j \neq i} (1 - X_j X_i^{-1})$$

$$= -\frac{X_i \, \omega(X_i^{-1})}{\sigma'(X_i^{-1})}$$

Correcting more than t errors

The decoding algms. correct t or fewer errors in a BCH code of designated distance 2t+1.

Complete decoding algms. are known for all double & some triple-error correcting codes.

Unsolved problem

Find a complete decoding algm. for all BCH codes.