ELG 5372 Error Control Coding

Lecture 17: Berlekamp-Massey Algorithm for Binary BCH Codes

uOttawa

L'Université canadienne
Canada's university

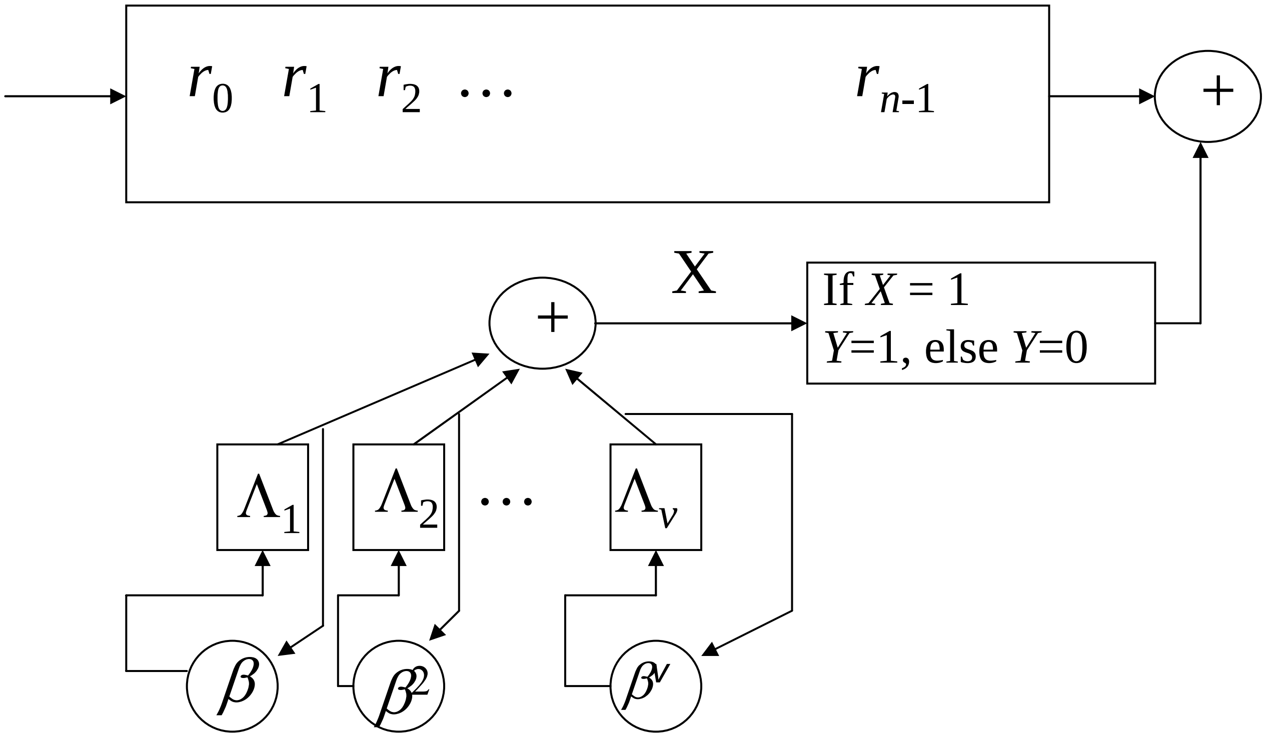
Université d'Ottawa | University of Ottawa

uOttawa.ca

Chien Search

- If $\Lambda(\beta^i) = 0$, then r_{n-i} is in error.
- This means that $\Lambda(\beta^i)+1 = 1$.
- $X(\beta^i) = \Lambda_1\beta^i + \Lambda_2\beta^{2i}+\ldots\Lambda_v\beta^{vi}$.
- If $X(\beta^i) = 1$, $c_{n-i} = r_{n-i}+1$, else $c_{n-i} = r_{n-i}$.
- If the Chien Search fails to find v roots of a error locator polynomial of degree v, then the error pattern is an uncorrectable error pattern.

Chien Search 2



Berlekamp-Massey Algorithm

- Peterson's method involves straightforward linear algebra, but it is computationally complex to implement.
- Should **A** be singular, the last two rows and columns are deleted and the determinant of the new **A** must be computed again.
- Thus, the Peterson method starts with a big problem and works it down to a small problem (thus if it is a small problem to begin with, the most computationally complex step is done for nothing).
- The Berlekamp-Massey algorithm starts with a small problem and works up to a large problem.
- Complexity of Peterson algorithm is proportional to v^3, while that of Berlekamp-Massey algorithm is proportional to v^2.

Berlekamp-Massey Algorithm 2

- It was observed from Newton's identities that

$$S_j = -\sum_{i=1}^{v} \Lambda_i S_{j-i}, \quad j = v+1, v+2, \ldots, 2t \qquad (*)$$

- (*) describes the output of a linear feedback shift register with coefficients Λ_1, Λ_2, …, Λ_v.
- Given a sequence S_1, S_2, …, S_{2t}, we can determine the LFSR coefficients.

Berlekamp-Massey Algorithm 3

- In the Berlekamp-Massey algorithm, we build the LFSR that produces the entire sequence by successively modifying an existing LFSR to produce increasingly longer sequences.
- We start with a LFSR that can produce S_1, then we check to see if that LFSR can produce $\{S_1, S_2\}$.
  - If so, no modification is necessary.
  - If not, then we need to modify the current LFSR to produce a new one that can produce the sequence.
- We repeat until we have a LFSR that produces the sequence $\{S_1, S_2, \ldots S_{2t}\}$.

Berlekamp-Massey Algorithm 4

- Let k be the iteration index of the algorithm and let L_k be the length of the LFSR on iteration k.
- Let $\Lambda^{(k)}(x)$ be the error locator polynomial at iteration k.

$$\Lambda^{(k)}(x) = 1 + \Lambda_1^{(k)} x + \Lambda_2^{(k)} x^2 + \ldots + \Lambda_{L_i}^{(k)} x^{L_i}$$

- At iteration k, we have a LFSR capable of producing sequence $\{S_1, S_2, \ldots, S_k\}$.

$$S_j = -\sum_{i=1}^{L_k} \Lambda_i^{(k)} S_{j-i}, \quad j = L_k + 1, \ldots, k$$

Berlekamp-Massey Algorithm 5

- Suppose after k-1 iterations, we have $\Lambda^{(k-1)}(x)$. On iteration k, we compute:

$$\hat{S}_k = -\sum_{i=1}^{L_{k-1}} \Lambda_i^{(k-1)} S_{k-i} \qquad (**)$$

- If this is equal to S_k, then the error locator polynomial is good to produce the sequence $\{S_1, S_2, \ldots, S_k\}$ and no changes are needed. Therefore $\Lambda^{(k)}(x) = \Lambda^{(k-1)}(x)$.
- If (**) is not equal to S_k, then the polynomial needs to be modified.
- This discrepancy is $d_k = S_k - \hat{S}_k = S_k + \sum_{i=1}^{L_{k-1}} \Lambda_i^{(k-1)} S_{k-i}$

Berlekamp-Massey Algorithm 6

$$d_k = S_k - \hat{S}_k = \sum_{i=0}^{L_{k-1}} \Lambda_i^{(k-1)} S_{k-i}$$

Let us produce a new polynomial $\Lambda^{(k)}(x) = \Lambda^{(k\text{-}1)}(x) + Ax^l \Lambda^{(m\text{-}1)}(x)$, where A is some element in the field, l is an integer and $\Lambda^{(m\text{-}1)}(x)$ is one of the prior error locator polynomials associated with an non-zero discrepancy d_m.

Let us compute the new discrepancy using this new polynomial.

$$d'_k = \sum_{i=0}^{L_{k-1}} \Lambda_i^{(k-1)} S_{k-i} + A \sum_{i=0}^{L_{m-1}} \Lambda_i^{(m-1)} S_{k-i-l} = d_k + Ad_m \text{ if we select } l = k - m$$

By choosing $A = -d_m^{-1} d_k$, $d'_k = 0$. Thus, new polynomial produces $\{S_1, S_2, \ldots, S_k\}$. Proof in text to show that this algorithm produces shortest LFSR.

Example

- Consider the two error correcting binary (15,7) BCH code. The generator polynomial has roots α, α^2, α^3 and α^4.
- Let $r(x) = x^2 + x^5$.
- $S_1 = \alpha$, $S_2 = \alpha^2$, $S_3 = \alpha^{13}$ and $S_4 = \alpha^4$.

Example Cont'd

k	S_k	d_k	$c(x)$	L	$p(x)$	d_m
0	0	1	1	0	1	1
1	α	α	$1+\alpha x$	0	1	1
2	α^2	0	$1+\alpha x$	1	1	α
3	α^{13}	$\alpha^{13}+\alpha^3 = \alpha^8$.	$1+\alpha x+x^{(3-1)}\alpha^8\alpha^{14} = 1+\alpha x+\alpha^7x^2$.	2	$1+\alpha x$	α
4	α^4	$\alpha^4+\alpha^{14}+\alpha^9 = 0$	$1+\alpha x+\alpha^7x^2$.	2	$1+\alpha x$	α^8

Simplification for binary codes

- Since S_{2k} is not independent of S_k, every even iteration of the Berlekamp-Massey algorithm will result in $d_k = 0$. Thus, we can skip every even iteration.