Introduction

- A Reed Solomon code is a special case of a BCH code in which the length of the code is one less than the size of the field over which the symbols are defined.

- It consists of sequences of length \( q - 1 \) whose roots include \( 2t \) consecutive powers of the primitive element of \( GF(q) \).

- The Fourier transform over \( GF(q) \) will contain \( 2t \) consecutive zeros.

- Note that because both the roots and the symbols are specified in \( GF(q) \), the generator polynomial will have only the specified roots: (NO Conjugates)
**Introduction**

- To construct the generator for a Reed Solomon code, and we decide that the roots will be from $\alpha^i$ to $\alpha^{i+2t-1}$, the generator polynomial will be
  
  \[ g(x) = (x + \alpha^i)(x + \alpha^{i+1}) \cdots (x + \alpha^{i+2t-2})(x + \alpha^{i+2t-1}) \]

- **Example:**
  Suppose we wish to construct a double-error correcting, length 7 RS code; we first construct $GF(8)$ using the primitive polynomial $X^3 + X + 1$ as shown. We decide to choose $i = 0$, placing the roots from $\alpha^0$ to $\alpha^3$. The generator polynomial is
  
  \[ g(x) = (x + \alpha^0)(x + \alpha^1)(x + \alpha^2)(x + \alpha^3) \]
  
  \[ g(x) = X^4 + \alpha^2X^3 + \alpha^5X^2 + \alpha^6X + \alpha^6 \]

---

**Time Domain Encoding**

- The encoding of a Reed Solomon code can be done by a long division method similar to that of Chapter 4

- **Example**
  - For the Reed Solomon code above, encode the data sequence 11100111.
  - The data maps to the symbols $\alpha^5 \alpha^0 \alpha^5$.
  - Four zeros are appended, corresponding to the four parity checks to be generated, and the divisor is the generator sequence.
The remainder is $\alpha^6 \alpha^5 \alpha^4 \alpha^1$ so that
the codeword is $\alpha^5 \alpha^0 \alpha^5 \alpha^6 \alpha^5 \alpha^0 \alpha^1$. Expressed as a binary sequence this is 111011110111001010.
Time Domain Encoding

For our example the encoder will be

\[ U(X) + \alpha^2 + \alpha^5 + \alpha^2 + \alpha^6 \]

Decoding Reed Solomon Code

- The frequency domain algebraic decoding method explained in Chapter 6.
- Error value calculation can be done using the Forney algorithm.
- Where we found that
  \[ A(z)E(z) = f(z)z^{2t} + \Omega(z) \quad 0 \leq k \leq 2t-1 \]
  - Where \( \Lambda(z) \) is the connection polynomial, \( E(z) \) is syndrome in frequency domain and \( \Omega(z) \) is \( t - 1 \) order evaluator polynomial.
Decoding Reed Solomon Code

- We calculate the error evaluator polynomial and also \( \Lambda'(z) \), the formal derivative of the connection polynomial. This is found to be

\[
\begin{align*}
\text{(even } t \text{)} & : \Lambda_{-2} z^{-2} + \Lambda_{-3} z^{-3} + \cdots + \Lambda_1 \\
\text{(odd } t \text{)} & : \Lambda_{-1} z^{-1} + \Lambda_{-3} z^{-3} + \cdots + \Lambda_1
\end{align*}
\]

- In other words, get rid of the zero coefficient of \( A \) and then set all the odd terms in the resulting series to zero.

- The error value in position \( m \) is now evaluated at \( z = \alpha^m \). The parameter \( i \) is the starting location of the roots of the generator polynomial.

Example: Consider the codeword \( \alpha^5 \alpha^0 \alpha^5 \alpha^0 \alpha^5 \alpha^0 \alpha^1 \) previously generated for the double-error correcting (7, 3) RS code. We create errors in positions 5 and 3, assuming that we receive \( \alpha^5 \alpha^4 \alpha^3 \alpha^5 \alpha^0 \alpha^1 \). The frequency domain syndrome of this sequence is

\[
\begin{align*}
S_0 &= \alpha^5 + \alpha^4 + \alpha^3 + \alpha^5 + \alpha^0 + \alpha^1 = \alpha^9 \\
S_1 &= \alpha^2 \alpha^6 + \alpha^4 \alpha^5 + \alpha^3 \alpha^3 + \alpha^5 \alpha^2 + \alpha^0 \alpha + \alpha^1 = \alpha^1 \\
S_2 &= \alpha^4 \alpha^5 + \alpha^3 \alpha^3 + \alpha^3 \alpha^1 + \alpha^3 \alpha^6 + \alpha^2 \alpha^4 + \alpha^0 \alpha^2 + \alpha^1 = \alpha^0 \\
S_3 &= \alpha^5 \alpha^4 + \alpha^2 \alpha^1 + \alpha^3 \alpha^2 + \alpha^5 \alpha^6 + \alpha^0 \alpha^3 + \alpha^1 = 0
\end{align*}
\]
Decoding Reed Solomon Code

- Now we form the key equation for which the solution is
  \[ \alpha^9\Lambda_2 + \alpha^4\Lambda_1 + \alpha^0 = 0 \]
  \[ \alpha^5\Lambda_2 + \alpha^6\Lambda_1 = 0 \]

- Which leads to
  \[ \Lambda_1 = \alpha^2 \quad \Lambda_2 = \alpha^4 \]

- And the connection polynomial is
  \[ \Lambda(z) = \alpha^4z^2 + \alpha^2z + 1 = 0 \]

- The roots for the connection polynomial is \( \alpha^4 \) and \( \alpha^2 \)

---

Decoding Reed Solomon Code

- Having found two roots for a connection polynomial of degree 2 indicates successful error correction with errors located at positions -4 and -2, i.e. positions 3 and 5. We now calculate the error evaluator polynomial, taking the powers from 0 to \( t - 1 \) of \( S(z)\Lambda(z) \).

  \[ \Omega(z) = (S_0\Lambda_1 + S_1\Lambda_0)z + S_0\Lambda_0 \]

  \[ \Omega(z) = \alpha^4z + \alpha^0 \]

- Therefore

  \[ e_m = \frac{\Omega(z)}{z\Lambda(z)} \bigg|_{z=\alpha^m} = \frac{\alpha^4z + \alpha^0}{\alpha^2z} \bigg|_{z=\alpha^m} \]

- Evaluating at \( m = 3 \) and \( 5 \)
Decoding Reed Solomon Code

- This leads to
  \[ e_5 = \frac{\alpha^4 \alpha^{-3} + \alpha^0}{\alpha^2 \alpha^{-3}} = \alpha^4 \]
  \[ e_3 = \frac{\alpha^4 \alpha^{-5} + \alpha^0}{\alpha^2 \alpha^{-3}} = \alpha^5 \]

- The received symbol \( \alpha^3 \) at position 3 is therefore corrected to \( \alpha^6 \). The received symbol \( \alpha^4 \) at position 5 is corrected to \( \alpha^5 \). This successfully completes the decoding.

Frequency domain encoding of RS Code

- As the Fourier transform of a Reed Solomon code word contains \( n - k \) consecutive zeros, it is possible to encode by considering the information to be a frequency domain vector, appending the appropriate zeros and inverse transforming.

- Example
  - We will choose again a \((7, 3)\) double-error correcting RS code over \( GF(2^3) \). Let the information be \( \alpha^2, \alpha^5, \alpha^9 \).
  - The frequency domain code word is \( \alpha^2 \alpha^5 \alpha^0 0000 \), and the inverse is \( \alpha^5 \alpha^0 \alpha^5 \alpha^6 \alpha^5 \alpha^0 \alpha^1 \).
  - We now create a two-symbol error, say \( \alpha^5 \) in position 5 and \( \alpha^4 \) in position 3, as in our previous example. The received sequence is \( \alpha^5 \alpha^4 \alpha^5 \alpha^3 \alpha^5 \alpha^0 \alpha^1 \).
Frequency domain encoding of RS Code

- The decoding proceeds with finding the Fourier transform of the sequence which gives $\alpha^5\alpha^2\alpha^3\alpha^0\alpha^1\alpha^0$ with syndrome being $0\alpha^5\alpha^1\alpha^0$.

We now form the key equation with $t = 2$ and $A_0 = 1$:

\[\alpha^5\Lambda_2 + \alpha^1\Lambda_1 + \alpha^0 = 0\]
\[\alpha^1\Lambda_2 + \alpha^0\Lambda_1 + 0 = 0\]

The solution is:

\[\Lambda_1 = \alpha^2\]
\[\Lambda_2 = \alpha^1\]

We use the shift register shown to generate the code sequence in the frequency domain.

In our example $\Lambda_0 = 1$, $\Lambda_1 = \alpha^2$, and $\Lambda_2 = \alpha$. $E_3 = 0$, $E_2 = \alpha$.

Cyclic the shift register generates $\alpha^1\alpha^3\alpha^3$, which are the frequency domain error at locations 4, 5, and 6.
Frequency domain encoding of RS Code

- The next two values generates $\alpha^0 \alpha^1$ which are the syndrome components $S_0$ and $S_1$, so the code will repeat

- The complete error sequence in the frequency domain is $\alpha^3 \alpha^3 \alpha^0 \alpha^0 \alpha^0$.

- Adding this to the Fourier transform of the received sequence which is $a^5 a^2 a^3 0 a^0 a^1 \alpha^0$.

- The result is $\alpha^2 a^5 \alpha^0 0000$, which is the original codeword without errors with four zeros syndrome.

---

Example 2

- For the previous example consider three errors have occurred, and the received sequence is $a^6 a^4 a^2 a^3 a^5 a^0 a^1$.

- The Fourier transform is
  $$R(z) = a^1 z^6 + a^1 z^3 + a^6 z^4 + a^3 z^3 + a^2 z^2 + a^3 z + a^3$$

- The key equations are found to be
  $$\alpha^3 \Lambda_2 + \alpha^1 \Lambda_1 + \alpha^3 = 0$$
  $$\alpha^3 \Lambda_1 + \alpha^2 \Lambda_1 + \alpha^3 = 0$$

- Or $\Lambda_0 = 1$, $\Lambda_1 = \alpha^0$, and $\Lambda_2 = \alpha^2$. 
Frequency domain encoding of RS Code

- The shift register become

After initializing the shift register with $\alpha^3 \alpha^2$. The output is $\alpha^2 \alpha^6 \alpha^5$ followed by $\alpha^5 \alpha$ which is not $S_0$, this means that there is an error which we cannot correct.

Example 3

- For the previous example consider single errors have occurred, and the received sequence is $\alpha^6 \alpha^4 \alpha^9 \alpha^6 \alpha^5 \alpha^0 \alpha^1$.
- The Fourier transform is

$$ R(z) = \alpha^6 z^6 + \alpha^2 z^3 + \alpha^5 z^4 + \alpha^6 z^3 + \alpha^1 z^2 + \alpha^1 z + \alpha^5 $$

- The key equations are found to be

$$ \alpha^3 \Lambda_2 + \alpha^5 \Lambda_1 + \alpha^1 = 0 $$
$$ \alpha^3 \Lambda_2 + \alpha^1 \Lambda_1 + \alpha^6 = 0 $$

- Or $\Lambda_0 = 1$, $\Lambda_1 = \alpha^5$, and $\Lambda_2 = 0$. 
Frequency domain encoding of RS Code

- The shift register become

\[ \alpha^2 \]

- After initializing the shift register with \( \alpha^6 \). The output is \( \alpha^4 \alpha^2 \alpha^0 \) followed by \( \alpha^5 \) which is the value of \( s_0 \).

- The complete error sequence is \( \alpha^4 \alpha^2 \alpha^0 \alpha^5 \alpha^0 \alpha^1 \).

- Adding this to \( \alpha^5 \alpha^4 \alpha^6 \alpha^0 \alpha^1 \). The result is \( \alpha^2 \alpha^5 \alpha^0 \alpha^0 \alpha^0 \).

Erasure Decoding

- Reed-Solomon has the ability to recover event known as erasure

- Erasure is a symbol which is more likely to be in error

Example

- Consider the same previous example but with one error a position 5 and two erasures in positions 6 and 1
- The received sequence will be taken as \( 0 \alpha^4 \alpha^5 \alpha^6 \alpha^0 \alpha^1 \)
- The Fourier transform is

\[ R(z) = \alpha^5 z^6 + \alpha^6 z^5 + \alpha^3 z^4 + \alpha^1 z^3 + \alpha^0 z^2 + \alpha^5 z + \alpha^0 \]
Erasure Decoding

- The lower terms of R forms the syndrome
- The erasure polynomial is
  \[ \Gamma(z) = (\alpha^5 z + 1)(\alpha^4 z + 1) = z^2 + \alpha^5 z + 1 \]
- The error locator polynomial is
  \[ \Lambda(z) = \Lambda_1 z + 1 \]
- and the product is
  \[ \Gamma(z)\Lambda(z) = \Lambda_1 z^3 + [\alpha^5 + \alpha^5 \Lambda_1] z^2 + [\alpha^4 + \Lambda_1] z + 1 \]
- By convolution with the syndrome polynomial
  \[ \alpha^5 \Lambda_1 + \alpha^5 + \alpha^4 \Lambda_1 + \alpha^4 + \alpha^3 \Lambda_1 + \alpha^1 = 0 \]
Erasure Decoding

- Loading with values $\alpha^6 \alpha^6 \alpha^1$ and shifting gives the sequence $\alpha^1 \alpha^1 \alpha^0$ and then regenerating the syndrome $\alpha^6 \alpha^5 \alpha^6$. The decoding is successful.

- The terms $\alpha^0 \alpha^1 \alpha^1$ are added to component $\alpha^6 \alpha^6 \alpha^1$ from the Fourier transform of the received sequence $y_0$ give the recovered information $\alpha^2 \alpha^5 \alpha^0$.

- If we choose Forney algorithm to correct errors in time domain

$$\Omega(z) = \left[S(z)\Gamma(z)\Lambda(z)\right] \mod z^4 = \alpha^5 z^2 + \alpha^2 z + \alpha^6$$

- We need also the formal derivative of $\Gamma(z)\Lambda(z)$ which is $\alpha^5 z^2$.

Erasure Decoding

- Therefore at position $i$ the error value is

$$e_i = \frac{z^i z^2 + \alpha^5 z + \alpha^0}{\alpha^2 z^3} \bigg|_{z = \alpha^i}$$

- Evaluating this at $\alpha^6$ and $\alpha^1$ which are the roots of $\Gamma(z)$ and $\alpha^5$ which is root of $\Lambda(z)$ and found by Chien search. The results are $e_i = \alpha^0, e_5 = \alpha^5, e_6 = \alpha^3$.

- The received sequence then is $\alpha^5 \alpha^0 \alpha^5 \alpha^6 \alpha^5 \alpha^0 \alpha^3$, corresponding to the original codeword.
Welch-Berlekamp Algorithm

- Consider the previous double error correction examples we transmit a codeword from a (7,3) RS code with roots $\alpha^3, \alpha^2, \alpha^1$, and $\alpha^0$.

- To start with, we need to calculate some values needed as input for the algorithm:

\[
\sum_{j=0}^{n-k-1} g_j x_j = \frac{g(X)}{X + \alpha^{n-k-1}} = (X + \alpha^0)(X + \alpha^1)(X + \alpha^2) = X^3 + \alpha^3 X^2 + \alpha^6 X + \alpha^3
\]

- Therefore $g_3 = \alpha^3$, $g_2 = \alpha^3$, $g_1 = \alpha^3$, and $g_0 = \alpha^3$

- For error value calculation, we compute

\[
C = \alpha^6 \alpha^1 \cdots \alpha^{n-k+1}(\alpha^0 + \alpha^1)(\alpha^0 + \alpha^2) \cdots (\alpha^6 + \alpha^{n-k+1})
\]

Welch-Berlekamp Algorithm

- For our example $C = \alpha^6$, we now need for each data location to find the value $h_i = C / g(\alpha^i)$. The values are found to be

\[
\begin{align*}
    h_6 &= \alpha^4 / \alpha^4 = \alpha^2 \\
    h_5 &= \alpha^5 / \alpha^3 = \alpha^5 \\
    h_6 &= \alpha^6 / \alpha^6 = \alpha^6
\end{align*}
\]

- Assume now we receive $\alpha^5 \alpha^4 \alpha^3 \alpha^2 \alpha^0$. The first step is to compute the syndrome by long division.

- This found to be $s_3 = \alpha^0$, $s_2 = \alpha^2$, $s_1 = \alpha^0$, and $s_0 = \alpha^6$

- The input to WB algorithm is the set of point $(S_j, \alpha^i)$ where

\[
S_j = s_j / g_j
\]
Welch-Berlekamp Algorithm

- The input points are therefore \((\alpha^0, \alpha^3), (\alpha^4, \alpha^2), (\alpha^1, \alpha^1), (\alpha^3, \alpha^0)\).
- We need to find two polynomials \(Q(X)\) and \(N(X)\) for which
  \[ Q(\alpha^j)S_j = N(\alpha^j) \quad \text{for} \quad 0 \leq j \leq n - k - 1 \]
- And the length \(L[Q(X), N(X)]\), defined as the maximum of \(\deg(Q(X))\) and \(\deg(N(X)) + 1\) has the minimum possible value.

The steps of the algorithms are now

1. Set \(Q^0(X) = 1, N^0(X) = 0, W^0(X) = X, V^0(X) = 1\) and \(d = 0\).
2. Evaluate \(D_1 = Q^d(\alpha^d)S_d + N^d(\alpha^d)\).
3. If \(D_1 = 0\), set \(W^{d+1} = W^d(X + \alpha^d), V^{d+1} = V^d(X + \alpha^d)\) and go to step 6. Otherwise, set \(D_2 = W^d(\alpha^d)S_d + V^d(\alpha^d)\).
4. Set \(Q^{d+1} = Q^d(X + \alpha^d), N^{d+1} = N^d(X + \alpha^d), W^{d+1} = W^d + Q^dD_2/D_1, V^{d+1} = V^d + N^dD_2/D_1\).
5. Check whether \(L[W^d, V^d]\) was less than or equal to \(L[Q^d, N^d]\); if it was then swap \(Q^{d+1}, N^{d+1}\) with \(W^{d+1}, V^{d+1}\). Increment \(d\).
6. If \(d < n - k\), return to step 1; otherwise, \(Q(X) = Q^d(X), N(X) = N^d(X)\).
**Welch-Berlekamp Algorithm**

For our example

<table>
<thead>
<tr>
<th>Table 7.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d$</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
</tbody>
</table>

The error locator polynomial is $X^2 + \alpha^2 X + \alpha^1$ which has roots $\alpha^3$ and $\alpha^5$. The error values in the data positions are

$$e_i = h \frac{N(\alpha^i)}{Q'(\alpha^i)}$$

**Welch-Berlekamp Algorithm**

Where $Q'(X)$ is the formal derivative of $Q(X)$. In this case $Q'(X) = \alpha^2$.

$$e_i = \alpha^5 (\alpha^5 + \alpha^1) / \alpha^2 = \alpha^5$$

The received information therefore is $\alpha^5 \alpha^0 \alpha^5$. 