Reed-Solomon Codes

Kwame Ackah Bohulu

May 29, 2018

1 Introduction

Reed-Solomon(R-S) codes are non-binary cyclic codes which were first introduced by Reed and Solomon in 1960.

The binary sequences are grouped into sequences of length equal to m . To design a t symbol-error correcting (n, k) R-S code the following conditions need to be met.

$$n = 2^m - 1, k = 2^m - 1 - 2t \tag{1}$$

where $n - k = 2t$ They are used in communication systems and especially in data storage systems and are very efffective in correcting random symbol errors and random burst errors.

2 Finite(Galois) Fields

for any prime number p there exists a finite field that contains p elements. This field is denoted by $GF(p)$. It is possible to extend $GF(p)$ to a field with p^m fields denoted by $GF(p^m)$, where m is a positive integer. The symbols used in the construction of R-S codes are taken from $GF(p^m)$ which is also called the extension field of $GF(p)$. In our study of R-S codes we focus on the extension field of $GF(2)$ denoted by $GF(2^m)$.

Besides 0 and 1, we can represent all other non-zero elements by powers of α, where α is a symbol introduced for convinience sake. Therefore to create a field F with infinite elements, all we need to do is to begin with the elements $\{0, 1, \alpha\}$ and continue to multiply the last element by α. However to create a truly finite field, we need to impose a condition on F so that it contains only 2^m elements and is closed under multiplication. This condition is given by the irreducible polynomial

$$\alpha^{(2^m-1)} = 1 = \alpha^0 \tag{2}$$

Using (2) , it is possible to reduce any field element element with power greater $2^m - 1$ to one with a power less than $2^m - 1$ as shown in (2)

$$\alpha^{(2^m+n)} = \alpha^{(2^m-1)}\alpha^{n+1} = \alpha^{n+1} \tag{3}$$

From the above discussion, we can conclude that elements of $GF(2^m)$ are given by

$$GF(2^m) = \{0, \alpha^0, \alpha^1, ..., \alpha^{2^m-2}\} \tag{4}$$

To define the finite fields of $GF(2^m)$ and also the R-S codes, primitive polynomials are used. For a polynomial $f(X)$ to be a primitive polynomial it must be irrreducible(cannot be factored to yield lower order polynomials) and the smallest positive integer n for which $f(X)$ divides X^{n+1} is $n = 2^m - 1$

For the case of $GF(2^3)$, the various representations of the elements of in that field as well as the addition and multiplication tables are shown below.

The primitive polynomial used is $f(X) = 1 + X + X^3$. For addition, we simply add the binary representation and write down its equivalent power notation. For multiplication, we just add the the powers modulo $2^m - 1$

Table 1: Power, Polynomial and Binary representation for elements in $GF(8)$

Power	Polynomial	Binary
$\alpha^0 = \alpha^7 = 1$	α^0	001
α^1	α^1	010
α^2	α^2	100
α^3	$\alpha^1 + 1$	011
α^4	$\alpha^2 + \alpha^1$	110
α^5	$\alpha^2 + \alpha^1 + 1$	111
α^6	$\alpha^2 + 1$	101

Table 2: Addition table for $GF(8)$ with $f(X) = 1 + X + X^3$

$\cdot$	α^0	α^1	α^2	α^3	α^4	α^5	α^6
α^0	0	α^3	α^6	α^1	α^5	α^4	α^2
α^1	α^3	0	α^4	α^0	α^2	α^6	α^5
α^2	α^6	α^4	0	α^5	α^1	α^3	α^0
α^3	α^1	α^0	α^5	0	α^6	α^2	α^4
α^4	α^5	α^2	α^1	α^6	0	α^0	α^3
α^5	α^4	α^6	α^3	α^2	α^0	0	α^1
α^6	α^2	α^5	α^0	α^4	α^3	α^1	0

Table 3: Multiplication table for $GF(8)$ with $f(X) = 1 + X + X^3$

$+$	α^0	α^1	α^2	α^3	α^4	α^5	α^6
α^0	α^0	α^1	α^2	α^3	α^4	α^5	α^6
α^1	α^1	α^2	α^3	α^4	α^5	α^6	α^0
α^2	α^2	α^3	α^4	α^5	α^6	α^0	α^1
α^3	α^3	α^4	α^5	α^6	α^0	α^1	α^2
α^4	α^4	α^5	α^6	α^0	α^1	α^2	α^3
α^5	α^5	α^6	α^0	α^1	α^2	α^3	α^4
α^6	α^6	α^0	α^1	α^2	α^3	α^4	α^5

3 Encoding

3.1 Generator Polynomial

To construct a R-S code we use a generator polynomial $g(X)$ of the form.

$$\begin{aligned}\mathbf{g}(X) &= (X-\alpha)(X-\alpha^2)\cdots(X-\alpha^2 t) \\ &= X^{2t} + g_{2t-1}X^{2t-1} + \cdots + g_2X^2 + g_1X + g_0\end{aligned} \tag{5}$$

The generator matrix is determined by selecting α from $GF(2^m)$ as a primitive element and find the minimal polynomials of α^i for $1 \leq i \leq 2t$ over $GF(2^m)$ and these polynomials are in the form $(X + \alpha^i)$

As an example, consider the $(7, 3)$ double error-correcting R-S code, ie $2t = 4$

$$\begin{aligned}\mathbf{g}(X) &= (X-\alpha)(X-\alpha^2)(X-\alpha^3)(X-\alpha^4) \\ &= (X^2-(\alpha+\alpha^2)X+\alpha^3)(X^2-(\alpha^3+\alpha^4)X+\alpha^7) \\ &= (X^2-\alpha^4X+\alpha^3)(X^2-\alpha^6X+\alpha^0) \\ &= X^4-(\alpha^4+\alpha^6)X^3+(\alpha^3+\alpha^{10}+\alpha^0)X^2-(\alpha^4+\alpha^9)X+\alpha^3 \\ &= X^4+(\alpha^3)X^3+(\alpha^0)X^2+(\alpha^1)X+\alpha^3\end{aligned} \tag{6}$$

3.2 Systematic Encoding

To encode a message polynomial $\mathbf{m}(X)$, we first multiply it by X^{n-k}. we then divide $X^{n-k}\mathbf{m}(X)$ by $\mathbf{g}(X)$ to get the parity polynomial $\mathbf{p}(X)$. The codeword $\mathbf{U}(X)$ is given by

$$\mathbf{U}(X) = \mathbf{p}(X) + X^{n-k}\mathbf{m}(X) \tag{7}$$

As an example, we encode the message 111110010 with the $(7, 3)$ R-S code. It can be seen that this input is a 3- symbol message with $111 = \alpha^5, 110 = \alpha^3$ and $010 = \alpha^1$. Using the convetion where the co-efficient of the highest power corresponds to the leftmost symbol, the message polynomial can be written as $\mathbf{m}(X) = \alpha^5X^2 + \alpha^3X + \alpha^1$

First we multiply $\mathbf{m}(X)$ by $= X^4$ to get $X^{n-k}\mathbf{m}(X) = \alpha^5X^6+\alpha^3X^5+\alpha^1X^4$. Next we divide $X^{n-k}\mathbf{m}(X)$ by $\mathbf{g}(X)$ to get $\mathbf{p}(X) = \alpha^6X^3 + \alpha^4X^2 + \alpha^2X + \alpha^0$. The addition and multiplication operations are done according to GF(8) arithmetic and are a bit tedious.

Finally

$$\begin{aligned}\mathbf{U}(X) &= \mathbf{p}(X) + X^{n-k}\mathbf{m}(X) \\ &= \alpha^5X^6 + \alpha^3X^5 + \alpha^1X^4 + \alpha^6X^3 + \alpha^4X^2 + \alpha^2X + \alpha^0\end{aligned} \tag{8}$$

3.3 Encoding Using and $(n - k)$ -Stage Shift Register

The diagram below is used to encode R-S codes. It should be noted that all operations are done in modulo $2^m - 1$. The encoding steps are as follows.

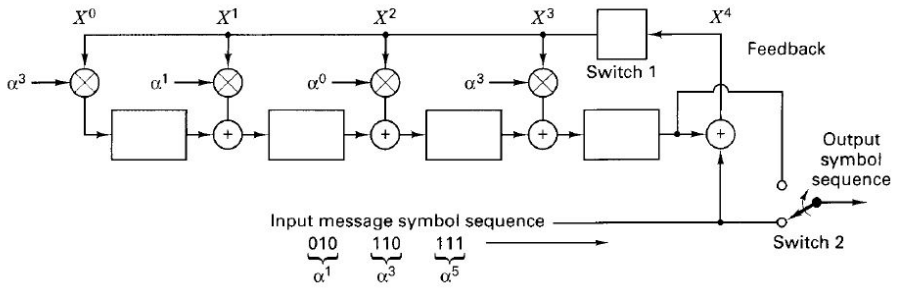

Figure 1: Encoder for (7,3) R-S code

- During the first k clock cycles, switch 1 is closed to allow shifting the message symbols into the $(n-k)$ stage shift register. It should be noted that switch 2 is also in the down position during that time to allow simultaneous transfer of the message symbols directly to the output.
- After the kth symbol has been transferred to the output register, switch one is opened and switch 2 is moved to the "up" position.
- Since the symbols that remain in the shift register correspond to the parity check bits, the $(n-k)$ clock cycles are used to move them to the output register.

Using the input used for the example in the previous sub-section, the table below shows the encoding process.

Shift Register Operations			
Input Queue	Clock Cycle	Register's Contents	Feedback
$\alpha^1 \quad \alpha^3 \quad \alpha^5$	0	$0 \quad 0 \quad 0 \quad 0$	α^5
$\alpha^1 \quad \alpha^3$	1	$\alpha^1 \quad \alpha^6 \quad \alpha^5 \quad \alpha^1$	α^0
α^1	2	$\alpha^3 \quad 0 \quad \alpha^2 \quad \alpha^2$	α^4
$-$	3	$\alpha^0 \quad \alpha^2 \quad \alpha^4 \quad \alpha^6$	$-$

It should be noted that the roots of $\mathbf{g}(X)$ should also be the roots of the codeword $\mathbf{U}(X)$ and a codeword is only valid if it has a value of zero when evaluated at the roots of the $\mathbf{g}(X)$

4 Decoding

Let us assume that after transmission through a channel we receive a corrupted codeword polynomial $\mathbf{r}(X)$. We can write $\mathbf{r}(X)$ as

$$\mathbf{r}(X) = \mathbf{U}(X) + \mathbf{e}(X) \tag{9}$$

Where $\mathbf{e}(X)$ is the error-pattern polynomial. In the decoding of R-S codes there is a need to determine the error locations as well as the error values. The

decoding process can be broken down into syndrome computation, error loaction computation, error value computation and codeword estimation. This process will be explained using an example. We assume the same $(7, 3)$ R-S code with generator matrix $\mathbf{g}(X) = X^4 + (\alpha^3)X^3 + (\alpha^0)X^2 + (\alpha^1)X + \alpha^3$. We also assume a double symbol error represented in pollynomial form as

$$\mathbf{e}(X) = 0X^6 + 0X^5 + \alpha^5 X^4 + \alpha^2 X^3 + 0X^2 + 0X + 0$$

which results in

$$\begin{aligned} \mathbf{r}(X) &= \mathbf{U}(X) + \mathbf{e}(X) \\ &= \alpha^5 X^6 + \alpha^3 X^5 + \alpha^6 X^4 + \alpha^0 X^3 + \alpha^4 X^2 + \alpha^2 X + \alpha^0 \end{aligned} \tag{10}$$

4.1 Syndrome Computation

The syndrome is used to determine wether a received signal is a valid codeword. A non-zero result indicates the prescence of an error in the received signal. The syndrome is made up of $(n-k)$ symbols and for our example there are 4 symbols in every syndrome vector $\mathbf{S}$. Using the idea that a valid codeword yields a value of zero when evaluated at the roots of the generator matrix, we go ahead and perform the syndrome calculations

$$\begin{aligned} S_1 = \mathbf{r}(X) &= \alpha^1 1 + \alpha^8 + \alpha^1 0 + \alpha^3 + \alpha^6 + \alpha^3 + \alpha^0 \\ &= \alpha^4 + \alpha^1 + \alpha^3 + \alpha^3 + \alpha^6 + \alpha^3 + \alpha^0 \\ &= \alpha^3 \end{aligned} \tag{11}$$

$$\begin{aligned} S_2 = \mathbf{r}(X) &= \alpha^1 7 + \alpha^1 3 + \alpha^1 4 + \alpha^6 + \alpha^8 + \alpha^4 + \alpha^0 \\ &= \alpha^3 + \alpha^6 + \alpha^0 + \alpha^6 + \alpha^1 + \alpha^4 + \alpha^0 \\ &= \alpha^5 \end{aligned} \tag{12}$$

$$\begin{aligned} S_3 = \mathbf{r}(X) &= \alpha^2 3 + \alpha^1 8 + \alpha^1 8 + \alpha^9 + \alpha^1 0 + \alpha^5 + \alpha^0 \\ &= \alpha^2 + \alpha^4 + \alpha^4 + \alpha^2 + \alpha^3 + \alpha^5 + \alpha^0 \\ &= \alpha^6 \end{aligned} \tag{13}$$

$$\begin{aligned} S_4 = \mathbf{r}(X) &= \alpha^2 9 + \alpha^2 3 + \alpha^2 2 + \alpha^1 2 + \alpha^1 2 + \alpha^6 + \alpha^0 \\ &= \alpha^1 + \alpha^2 + \alpha^1 + \alpha^5 + \alpha^5 + \alpha^6 + \alpha^0 \\ &= 0 \end{aligned} \tag{14}$$

Since $\mathbf{S} \neq 0$ there is an error. It is worth noting that the syndrome can also be determined by using symbol error polynomial.

$$\begin{aligned} S_i &= \mathbf{U}(\alpha^i) + \mathbf{e}(\alpha^i) \\ &= 0 + \mathbf{e}(\alpha^i) \end{aligned} \tag{15}$$

4.2 Error Location

As mentioned in the previous subsection, it is possible to compute the syndrome using the symbol error polynomial. In general, we assume that there are v errors in the codeword at $X^{j_1}, X^{j_2}, ..., X^{j_v}$, where $1, 2, ...v$ represents the 1st, 2nd,...,vth error and the index j represents the error locaton. It is possible to write the symbol error polynomial as

$$\mathbf{e}(X) = e_{j_1}X^{j_1} + e_{j_2}X^{j_2} + \cdots + e_{j_v}X^{j_v} \tag{16}$$

We need to determine each error value e_{j_l} and its location numberX^{j_l}, $l = 1, 2, ...v$ in order to correct the corrupted codeword. To do this , we first define an error locator number as $\beta_l = \alpha^{j_l}$ and then substitute α^i into the received polynomial for $i = 1, 2, ...2t$ to obtain the $2t$ syndrome symbols.

$$\begin{aligned} S_1 &= \mathbf{r}(\alpha) = e_{j_1}\beta_1 + e_{j_2}\beta_2 + \cdots + e_{j_v}\beta_v \\ S_2 &= \mathbf{r}(\alpha^2) = e_{j_1}\beta_1^2 + e_{j_2}\beta_2^2 + \cdots + e_{j_v}\beta_v^2 \\ &. \\ &. \\ &. \\ S_{2t} &= \mathbf{r}(\alpha) = e_{j_1}\beta_1^{2t} + e_{j_2}\beta_2^{2t} + \cdots + e_{j_v}\beta_v^{2t} \end{aligned} \tag{17}$$

Any equation that can be used to solve the above system of eequations is known as a R-S decoding algorithm. If after calculating the syndrome$\mathbf{S}$, it is a non-zero vector we know that an error has occured. We then have to determine the location of the errors. To do this we can define an error-locator polynomial as

$$\begin{aligned} \sigma(X) &= (1 + \beta_1 X)(1 + \beta_2 X) \cdots (1 + \beta_v X) \\ &= 1 + \sigma_1 X + \sigma_1 X^2 + \cdots + \sigma_v X^v \end{aligned} \tag{18}$$

$\sigma(\mathbf{X})$ has roots of the form $1/\beta_1, 1/\beta_2, ..., 1/\beta_v$ and the reciprocal of the roots give us the error location numbers $\mathbf{e}(X)$. We then use autoregressive modelling techniques to form a matrix from the syndromes where the first t syndromes are used to predict the next syndrome as shown below

$$\begin{bmatrix} S_1 & S_2 & S_3 & \dots & S_{t-1} & S_t \\ S_2 & S_3 & S_4 & \dots & S_t & S_{t+1} \\ & & & \vdots & & \\ S_{t-1} & S_t & S_{t+1} & \dots & S_{2t-3} & S_{2t-2} \\ S_t & S_{t+1} & S_{t+2} & \dots & S_{2t-2} & S_{2t-1} \end{bmatrix} \begin{bmatrix} \sigma_t \\ \sigma_{t-1} \\ \vdots \\ \sigma_2 \\ \sigma_1 \end{bmatrix} = \begin{bmatrix} -S_{t-1} \\ -S_{t+2} \\ \vdots \\ -S_{2t-1} \\ -S_{2t} \end{bmatrix} \tag{19}$$

We apply the autoregressive model of (21) to our example and this yields

$$\begin{aligned} \sigma(X) &= (1 + \beta_1 X)(1 + \beta_2 X) \cdots (1 + \beta_v X) \\ &= 1 + \sigma_1 X + \sigma_1 X^2 + \cdots + \sigma_v X^v \end{aligned} \tag{20}$$

$\sigma(\mathbf{X})$ has roots of the form $1/\beta_1, 1/\beta_2, ..., 1/\beta_v$ and the reciprocal of the roots give us the error location numbers $\mathbf{e}(X)$. We then use autoregressive modelling techniques to form a matrix from the syndromes where the first t syndromes are used to predict the next syndrome as shown below

$$\begin{bmatrix} S_1 & S_2 \\ S_2 & S_3 \end{bmatrix} \begin{bmatrix} \sigma_2 \\ \sigma_1 \end{bmatrix} = \begin{bmatrix} S_3 \\ S_4 \end{bmatrix}$$

$$\begin{bmatrix} \alpha^3 & \alpha^5 \\ \alpha^5 & \alpha^6 \end{bmatrix} \begin{bmatrix} \sigma_2 \\ \sigma_1 \end{bmatrix} = \begin{bmatrix} \alpha^6 \\ 0 \end{bmatrix} \tag{21}$$

To solve for the coefficients σ_1 and σ_2 we find the inverse of the know matrix on the left side (A) of (**??**) and multiply that by the known matrix on the right (B).

$$Inv[A] = \frac{cofactor[A]}{det[A]}$$

$$\begin{aligned} det \begin{bmatrix} \alpha^3 & \alpha^5 \\ \alpha^5 & \alpha^6 \end{bmatrix} &= \alpha^3\alpha^6 - \alpha^5\alpha^5 = \alpha^9 + \alpha^{10} \\ &= \alpha^2 + \alpha^3 = \alpha^5 \end{aligned} \tag{22}$$

$$cofactor \begin{bmatrix} \alpha^3 & \alpha^5 \\ \alpha^5 & \alpha^6 \end{bmatrix} = \begin{bmatrix} \alpha^6 & \alpha^5 \\ \alpha^5 & \alpha^3 \end{bmatrix} \tag{23}$$

$$Inv \begin{bmatrix} \alpha^3 & \alpha^5 \\ \alpha^5 & \alpha^6 \end{bmatrix} = \begin{bmatrix} \alpha^6 & \alpha^5 \\ \alpha^5 & \alpha^3 \end{bmatrix} \alpha^{-5} = \begin{bmatrix} \alpha^1 & \alpha^0 \\ \alpha^0 & \alpha^5 \end{bmatrix} \tag{24}$$

and

$$\begin{bmatrix} \sigma_1 \\ \sigma_2 \end{bmatrix} = \begin{bmatrix} \alpha^1 & \alpha^0 \\ \alpha^0 & \alpha^5 \end{bmatrix} \begin{bmatrix} \alpha^6 \\ 0 \end{bmatrix} = \begin{bmatrix} \alpha^0 \\ \alpha^6 \end{bmatrix} \tag{25}$$

Fixing the σ_1 and σ_2 into (20), we get the error-locator polynomial has the form

$$\sigma(X) = 1 + \alpha^6 X + \alpha^0 X^2 \tag{26}$$

Next , we proceed to find the roots of $\sigma(X)$. This can easily be done by inserting all the elements of $GF(8)$ into $\sigma(X)$ and the inputs that produce a zero output are the roots. For our example α^3 and α^4 are the roots of $\sigma(X)$.

$$\begin{aligned} \sigma(\alpha^3) &= \alpha^0 + \alpha^9 + \alpha^6 = 0 \\ \sigma(\alpha^4) &= \alpha^0 + \alpha^{10} + \alpha^8 = 0 \end{aligned} \tag{27}$$

finally, the error locators are the inverse of the roots and is given by $\beta_1 = \alpha^4$ and $\beta_2 = \alpha^3$ respectively.

4.3 Error Values

Now that we have determined the position in which the errors occur, we can proceed to determine the error values. We do this by inserting the values of β_1 and β_2 into (17). We need just $t = 2$ syndrome equations to calculate the error values and we can choose any of them. Fo this example, we proceed to use S_1 and S_2

$$\begin{aligned} S_1 &= \mathbf{r}(\alpha) = e_1\beta_1 + e_2\beta_2 \\ S_2 &= \mathbf{r}(\alpha^2) = e_1\beta_1^2 + e_2\beta_2^2 \end{aligned} \tag{28}$$

Where we have simplified the notation from e_{j_1} to e_1 and e_{j_2} to e_2. We proceed to write this equation in matrix form and use matrix algebra to find the values of e_1 and e_2 as shown below

$$\begin{bmatrix} \beta_1 & \beta_2 \\ \beta_1^2 & \beta_2^2 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} S_1 \\ S_2 \end{bmatrix}$$

$$\begin{bmatrix} \alpha^3 & \alpha^4 \\ \alpha^6 & \alpha^8 \end{bmatrix} \begin{bmatrix} \sigma_2 \\ \sigma_1 \end{bmatrix} = \begin{bmatrix} \alpha^3 \\ \alpha^5 \end{bmatrix} \tag{29}$$

also

$$Inv \begin{bmatrix} \alpha^3 & \alpha^4 \\ \alpha^6 & \alpha^1 \end{bmatrix} = \begin{bmatrix} \alpha^1 & \alpha^4 \\ \alpha^6 & \alpha^3 \end{bmatrix} \alpha^{-6} = \begin{bmatrix} \alpha^2 & \alpha^5 \\ \alpha^0 & \alpha^4 \end{bmatrix} \tag{30}$$

Finally, we solve (29).

$$\begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} \alpha^2 & \alpha^5 \\ \alpha^0 & \alpha^4 \end{bmatrix} \begin{bmatrix} \alpha^3 \\ \alpha^5 \end{bmatrix} = \begin{bmatrix} \alpha^2 \\ \alpha^5 \end{bmatrix} \tag{31}$$

4.4 Error Correction

Using the previous results, we write the estimated error polynomial as

$$\hat{\mathbf{e}}(X) = \alpha^2 X^3 + \alpha^5 X^5 \tag{32}$$

The estimated received polynomial $\hat{\mathbf{U}}(X)$ is calculated by adding the received polynomial $\mathbf{r}(X)$ to the estimated error polynomial $\hat{\mathbf{e}}(X)$

$$\hat{\mathbf{U}}(X) = \mathbf{r}(X) + \hat{\mathbf{e}}(X) \tag{33}$$

$$\begin{aligned} &\mathbf{r}(X) = (111)X^6 + (110)X^5 + (101)X^4 + (100)X^3 + (011)X^2 + (001)X + (100) \\ &\hat{\mathbf{e}}(X) = (000)X^6 + (000)X^5 + (111)X^4 + (001)X^3 + (000)X^2 + (000)X + (000) \\ &\hat{\mathbf{U}}(X) = (111)X^6 + (110)X^5 + (010)X^4 + (101)X^3 + (011)X^2 + (001)X + (100) \\ &= (\alpha^5)X^6 + (\alpha^3)X^5 + (\alpha^1)X^4 + (\alpha^6)X^3 + (\alpha^4)X^2 + (\alpha^2)X + (\alpha^0) \end{aligned} \tag{34}$$

The message symbol is given by the leftmost $k = 3$ symbols , ie $111 = \alpha^5, 110 = \alpha^3$ and $010 = \alpha^1$ which is the same as the message symbol used through out the example.

5 References