Quest Journals
Journal of Research in Applied Mathematics
Volume 11 ~ Issue 4 (April 2025) pp: 01-05
ISSN (Online): 2394-0743 ISSN (Print): 2394-0735
www.questjournals.org

Review Paper

An Application of Polynomial on RS Codes

Weam Mohammed Ahmed Elhadi Hamid
Maha Toufig
Department of Mathematics, Sudan University Science and Technology

Abstract
In this paper we discuss a class of codes called RS Codes are subset of BCH Code and linear block codes can be constructed to be multiple error-correcting. We use application of polynomial on RS Codes and apply them using Maple.

Keywords: *RS Code, polynomial, error-correcting codes.*

***R**eceived 05 Apr., 2025; **R**evised 13 Apr., 2025; **A**ccepted 15 Apr., 2025 © The author(s) 2025.*
P**ublished with open access at **www.questjournas.org

I. RS Codes

The RS Codes named after their inventors, who published them in 1960. The Reed-Solomon (RS) Codes are BCH Codes. Reed-Solomon Codes are based on a specialist area of mathematics known as Galois Field or Finite Field.

A Reed-Solomon code-word is generated using a special polynomial. All valid code-word are exactly divisible by the generator polynomial. The general form of the generator polynomial is

$$g(x) = (x - \alpha)(x - \alpha^2) \dots (x - \alpha^{2t})$$

II. Construction of RS

To construct a Reed-Solomon, by choosing a primitive polynomial $p(x)$ of degree n in $Z_2[x]$ and forming the field $F = Z_2[x]/(p(x))$ of order 2^n. We denote the element x in this field by α. Reed-Solomon code-words are then polynomials of degree less than $2^n - 1$. However, unlike BCH code-words which are elements in $Z_2[x]$, Reed-Solomon code-words are in the following generator polynomial $g(x) \in F[x]$.[1]

The code-words in C are then all multiples $b(x)g(x)$ of degree less than $2^n - 1$ with $b(x) \in F[x]$. The code-words in C have length $2^n - 1$ positions and from a vector space with dimension $2^n - 1 - 2t$. We will use the notation $RS(2^n, t)$ to represent a t-error correcting Reed-Solomon code with code-words of length $2^n - 1$ positions. [1]

Example (1):

Choose primitive polynomial $p(x) = x^4 + x + 1 \in Z_2[x]$. The nonzero elements in $F = Z_2/(p(x))$. Using this field F, we obtain the following generator polynomial $g(x)$ for an RS(15,2) code C.

$$g(x) = (x - \alpha)(x - \alpha^2)(x - \alpha^3)(x - \alpha^4)$$

$$= x^4 + \alpha^{13}x^3 + \alpha^6x^2 + \alpha^3x + \alpha^{10}$$

To construct one of the code-words in C, consider $b(x) = \alpha^{10}x^9 \in F[x]$. Then

$$b(x)g(x) = \alpha^{10}x^{13} + \alpha^8x^{12} + \alpha x^{11} + \alpha^{13}x^{10} + \alpha^5x^9$$

is one of the code-words in C. [1]

III. Error Correction in Reed-Solomon Codes

Let F be a field of order 2^n, and let C be an $RS(2^n-1,t)$ code in $F[x]$. Suppose $c(x) \in C$ is transmitted and we receive the polynomial $r(x) = c(x) + e(x)$ for some non-zero error polynomial $e(x)$ in $F[x]$ of degree less than $2^n - 1$. We can use the following steps to determine $e(x)$. [1]

1) We first compute the first $2t$ syndromes of $r(x)$, which we will denote by: $S_1 = r(\alpha)$, $S_2 = r(\alpha^2)$, $\cdots$, $S_{2t} = r(\alpha^{2t})$ and form the following syndrome polynomial $S(z)$.

$$S(z) = S_1 + S_2 z + S_3 z^2 + \cdots + S_{2t} z^{2t-1}.$$

2) Nest, we construct the Euclidean algorithm for the polynomials

$\alpha(z) = z^{2t}$ and $b(z) = S(z)$ in $F[z]$, stopping at the first row j for which $deg(r_j) < t$. Let $R(z) = r_j$ and $V(z) = v_j$.

3) We can find the error positions in $r(x)$ by finding the roots of $V(z)$. Specifically, if $\alpha^{i_1}, \alpha^{i_2}, \cdots, \alpha^{i_k}$ are the roots of $V(z)$, then $r(x)$ contains errors in positions $x^{-i_1}, x^{-i_2}, \cdots, x^{-i_k}$. Finally, we must find the coefficients of $e(x)$ at these error positions. Let e_{-i} be the coefficient of the x^{-i} term in $e(x)$.

Then

$$e_{-i} = \frac{R(\alpha^i)}{V'(\alpha^i)}$$

IV. Generator Matrices of RS Codes

Reed-Solomon Codes are cyclic codes, and a cyclic shift of a code-word is another code-word. The generator polynomial of a Reed-Solomon Code can be used to construct the generator matrix of the corresponding Reed-Solomon Code or as an alternative approach, we can find the generator matrix of Reed-Solomon Code using:

$$G = \begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ \vdots \\ g_k \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 & \dots & 1 \\ 1 & \alpha & \alpha^2 & \alpha^3 & \dots & \alpha^{n-1} \\ 1 & (\alpha)^2 & (\alpha^2)^2 & (\alpha^3)^2 & \dots & (\alpha^{n-1})^2 \\ & & & \vdots & & \\ 1 & (\alpha)^{k-1} & (\alpha^2)^{k-1} & (\alpha^3)^{k-1} & \dots & (\alpha^{n-1})^{k-1} \end{bmatrix} \quad \rightarrow (1)$$

where each α^i term can be represented using a column vector including m-bits, where m is the number appearing in $GF(2^3)$. [3]

4-1 Encoding of RS Codes

If data-word polynomial is expressed in vector form using the extended field elements as

$$d = [d_1\ d_2\ \cdots\ d_k]$$

then the encoding operation can be achieved using

$$c = dG$$

leading to

$$c = d_1 g_1 + d_2 g_2 + \cdots + d_k g_k$$

The encoding operation can also be achieved using

$$c(x) = d(x)g(x)$$

where $g(x)$ is the generator polynomial of the $RS(n,k)$ and $d(x)$ is the polynomial form of d. [3]

Example (2):

The generator polynomial of the double-error-correcting $RS(7,3)$ using the extended field element $GF(2^3)$ generated using the primitive polynomial $p(x) = x^3 + x + 1$ can be calculated as

$$g(x) = x^4 + \alpha^3 x^3 + x^2 + \alpha x + \alpha^3$$

We want to encode the data-word $d = [110111010]$ using $RS(7,3)$. The encoding of the data-word achieved using the following: [3]

First, let's give the field element of $GF(2^3)$ for the reminder

$$0 \rightarrow 0$$

$$1 \rightarrow 1$$

$$\alpha \rightarrow \alpha$$

$$\alpha^2 \rightarrow \alpha^2$$

$$\alpha^3 \rightarrow \alpha + 1$$

$$\alpha^4 \rightarrow \alpha^2 + \alpha$$

$$\alpha^5 \rightarrow \alpha^2 + \alpha + 1$$

$$\alpha^6 \rightarrow \alpha^2 + 1$$

The generator matrix of the code be formed using Eq.(1) as

$$G = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 \\ 1 & \alpha^2 & \alpha^4 & \alpha^6 & \alpha^8 & \alpha^{10} & \alpha^{12} \end{bmatrix}$$

which can be simplified by using $\alpha^7 = 1$ as

$$G = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 \\ 1 & \alpha^2 & \alpha^4 & \alpha^6 & \alpha & \alpha^3 & \alpha^5 \end{bmatrix}$$

The data-word given in the example can be expressed in polynomial form as

$$d = \left[\underbrace{110}_{\alpha^2+\alpha} \quad \underbrace{111}_{\alpha^2+\alpha+1} \quad \underbrace{010}_{\alpha}\right] \rightarrow d = [\alpha^4 \ \alpha^5 \ \alpha].$$

The encoding operation can be performed as

$$c = dG$$

$$\rightarrow c = \alpha^4[1\,1\,1\,1\,1\,1\,1] + \alpha^5[1\,\alpha\ \alpha^2\,\alpha^3\,\alpha^4\,\alpha^5\,\alpha^6] + \alpha[1\,\alpha^2\,\alpha^4\,\alpha^6\,\alpha\ \alpha^3\,\alpha^5]$$

$$\rightarrow c = [\alpha^4\ \alpha^4\ \alpha^4\ \alpha^4\ \alpha^4\ \alpha^4\ \alpha^4] + [\alpha^5\ \alpha^6\ \alpha^7\ \alpha^8\ \alpha^9\ \alpha^{10}\ \alpha^{11}] + [\alpha\ \alpha^3\ \alpha^5\ \alpha^7\ \alpha^2\ \alpha^4\ \alpha^6]$$

$$\rightarrow c = \begin{bmatrix} \alpha^4 + \alpha^5 + \alpha & \alpha^4 + \alpha^6 + \alpha^3 & \alpha^4 + \alpha^7 + \alpha^5 & \alpha^4 + \alpha^8 + \alpha^7 \\ \alpha^4 + \alpha^9 + \alpha^2 & \alpha^4 + \alpha^{10} + \alpha^4 & \alpha^4 + \alpha^{11} + \alpha^4 & \end{bmatrix}$$

$$\rightarrow c = [\alpha^3 \quad 0 \quad 0 \quad \alpha^6 \quad \alpha^4 \quad \alpha^3 \quad \alpha^6]$$

which can be expressed in binary form as

$$\rightarrow c = \left[\underbrace{\alpha^3}_{\alpha+1} \quad 0 \quad 0 \quad \underbrace{\alpha^6}_{\alpha^2+1} \quad \underbrace{\alpha^4}_{\alpha^2+\alpha} \quad \underbrace{\alpha^3}_{\alpha+1} \quad \underbrace{\alpha^6}_{\alpha^2+1}\right]$$

$$\rightarrow c = [011 \quad 000 \quad 000 \quad 101 \quad 110 \quad 011 \quad 101]$$

which can be expressed using concatenated bits as

$$\rightarrow c = [01100000101110011101].$$

V. Decoding BCH and RS Codes

There are many algorithms which have been developed for decoding RS codes.

The algebraic decoding BCH or RS codes has the following general steps:

1. Computation of the syndrome.
2. Determination of an error locator polynomial, whose roots provide an indication of where the errors are. There are many different ways of finding the locator polynomial. These methods include the Peterson–Gorenstein–Zierler algorithm for RS codes, the Berlekamp–Massey algorithm for RS codes.
3. Finding the roots of the error locator polynomial. This is usually done using the Chien search, which is an exhaustive search over all the elements in the field.
4. For RS codes non-binary, the error values must be determined. This is typically accomplished using Forney's algorithm. [4]

5-1 Computation of the Syndrome

Let $g(x)$ be the generator polynomial for a RS code, with roots $\beta^b, \beta^{b+1}, \cdots, \beta^{b+2t-1}$, where β is an element of order n. Since

$$g(\beta^b) = g(\beta^{b+1}) = \cdots = g(\beta^{b+2t-1})$$

it follows that a code-word $c = (c_0, \cdots, c_{n-1})$ with polynomial

$c(x) = c_0 + \cdots + c_{n-1}x^{n-1}$ has

$$c(\beta^b) = \cdots = c(\beta^{b+2t-1}) = 0$$

For a received polynomial $r(x) = c(x) + e(x)$, we compute

$$S_j = r(\beta^j) = c(\beta^j) + e(\beta^j) = e(\beta^j) \ , \qquad j = b, b+1, \cdots, b+2t-1$$

The values $S_b, S_{b+1} \cdots, S_{b+2t-1}$, are called the syndromes of the received data. Suppose that r has v errors in it which are at locations $i_1, i_2, \cdots, i_v$, with corresponding error values in these locations $e_{i_j} = 0$. The errors can be represented by the error polynomial

$$e(x) = \sum_{k=0}^{n-1} e_k x^k = \sum_{l=1}^{v} e_{i_l} x^{i_l}$$

Then

$$S_j = e(\beta^j) = \sum_{l=1}^{v} e_{i_l}(\beta^j)^{i_l} = \sum_{l=1}^{v} e_{i_l}(\beta^{i_l})^j \ , \qquad j = b, b+1, \cdots, b+2t-1$$

Let

$$X_l = \beta^{i_l}$$

Then we can write

$$S_j = \sum_{l=1}^{v} e_{i_j} X_l^j \quad , \qquad j = b, b+1, \cdots, b+2t-1$$

If we know the X_l, then we know the location of the errors. [4]

VI. Conclusion

Application of polynomial on RS Codes is an application of algebra that has increasingly important over the last several decades. And apply them using Maple.

References

[1]. Klima, Richard E. Application of Abstract Algebra with Maple. Richard E.Klima, Neil P.sigmon, Ernest stitzinger.
[2]. Lidl, Rudolf. Applied Abstract Algebra / Rudolf Lidl, Gunter pi\2-2end ed.
[3]. Orhan Gazi. Forword Error Correction via Channel Coding. Springer Nature Switzerland AG2020.
[4]. Todd K.Moon. Error Correction Coding "Mathematical Methods and Algorithms" Second edition 2021.