ANALYSIS OF BOUNDED DISTANCE DECODING FOR REED SOLOMON CODES

Oluwaseyi Paul BABALOLA

Masters Report

A report submitted in fulfillment of the requirements
for the degree of Master of Science (50/50)

in the

Centre for Telecommunication Access and Services (CeTAS)
School of Electrical and Information Engineering
Faculty of Engineering and the Built Environment

February 2017

Declaration

I, Babalola, Oluwaseyi Paul, declare that this Report titled, "Analysis of Bounded Distance Decoding of Reed Solomon Codes" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a Msc 50/50 degree at this University.
- Where any part of this report has previously been submitted for a degree or any other qualification at this University or any other institution, has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this is entirely my own work.
- I have acknowledged all main sources of help.

Signed: _____

Date: _____

Abstract

Bounded distance decoding of Reed Solomon (RS) codes involves finding a unique codeword if there is at least one codeword within the given distance. A corrupted message having errors that is less than or equal to half the minimum distance corresponds to a unique codeword, and therefore will decode errors correctly using the minimum distance decoder. However, increasing the decoding radius to be slightly higher than half of the minimum distance may result in multiple codewords within the Hamming sphere. The list decoding and syndrome extension methods provide a maximum error correcting capability whereby the radius of the Hamming ball can be extended for low rate RS codes. In this research, we study the probability of having unique codewords for $(7, k)$ RS codes when the decoding radius is increased from the error correcting capability t to $t + 1$. Simulation results show a significant effect of the code rates on the probability of having unique codewords. It also shows that the probability of having unique codeword for low rate codes is close to one.

I dedicate this to God the Father, the Son, and the Holy Spirit.

I also dedicate this to my Wife and Daughter

Mrs Toluwalope Mary and Oluwadarasimi Marie Babalola. They have always supported and trusted my ability to achieve the best in whatsoever I do.

Acknowledgements

I sincerely appreciate and thank my supervisor, Prof. D.J.J. Versfeld for his mentorship and support throughout the period of my research. He stood by me when the going was tough and ensured I successfully completed this work. I shall be forever grateful.

I also appreciate the financial and material supports of the Centre for Telecommunications Access and Services (CeTAS) group towards the completion of this research.

I would like to express my sincere gratitude to Onye and all my friends, especially members of Deeper Life Campus Fellowship (WITS branch) that have extended all kind of help and prayers for accomplishing this work.

I would like to say a very big thank you to my parents (Prof. and Mrs. J.B. Babalola) for the moral, spiritual, and financial supports they gave throughout the period of my studies. They taught me to know that the best investment in life is education. I say thank you.

Finally, I appreciate the moral support received from my siblings, you guys are the best.

Contents

List of Figures

List of Tables

Abbreviations

AWGN	**A**dditive **W**hite **G**aussian **N**oise
BCH	**B**ose **C**haudhuri **H**ocquenghem
BDD	**B**ounded **D**istance **D**ecoding
BER	**B**it **E**rror **R**ate
BMA	**B**erlekamp **M**assey **A**lgorithm
BPSK	**B**inary **P**hase - **S**hift **K**eying
CER	**C**odeword **E**rror **R**ate
EEA	**E**xtended **E**uclidean **A**lgorithm
FEC	**F**orward **E**rror **C**orrection
GF	**G**alois **F**ield
IRS	**I**nterleaved **R**eed **S**olomon
MDD	**M**inimum **D**istance **D**ecoding
MDS	**M**aximum **D**istance **S**eparable
PGZA	**P**eterson **G**orenstein **Z**ierler **A**lgorithm
QAM	**Q**uadrature **A**mplitude **M**odulation
RS	**R**eed **S**olomon
SER	**S**ymbol **E**rror **R**ate
SNR	**S**ignal-to-**N**oise **R**atio
SSB	**S**chmidt **S**idorenko **B**ossert

CHAPTER 1

Introduction

There are relationships between mathematical geometry and coding theory. For instance, the Hamming distance $\mathbf{d_H}$ is used geometrically to measure the distance between any two points. However, in this work, the minimum distance (d_{min}) of a code $\mathcal{C}$ is defined by the shortest Hamming distance that exist between any two codewords in the code [1]:

$$d_{min} = \min_{\mathbf{c}_i, \mathbf{c}_j \in \mathcal{C}, \mathbf{c}_i \neq \mathbf{c}_j} d_H(\mathbf{c}_i, \mathbf{c}_j).$$

Bounded distance decoding is based on sphere construction around each codeword [1]. The Hamming (decoding) sphere of a given radius t constructed about a codeword $\mathbf{c}$ has all vectors $\mathbf{r}$ at a Hamming distance less than or equal to the radius t from the codeword [1]. If the Hamming spheres about each codeword have the same radius, the distance between two nearest codewords (d_{min}) in the code determines the largest radius creating nonoverlapping spheres. Figure 1.1 illustrates decoding spheres of radius $t = \lfloor (d_{min} - 1)/2 \rfloor$. Whenever a vector $\mathbf{r}$ lies within a sphere around a codeword, the decoder assumes $\mathbf{r}$ is closer to the codeword in that particular sphere than to those in other spheres. The decoder therefore decodes $\mathbf{r}$ to the codeword inside the same sphere.

Irving Reed and Gustave Solomon discovered Reed Solomon (RS) codes in 1961 which is a type of forward error correction (FEC) code with several applications in storage systems, communications, spacecraft, etc. Most error correction applications of RS codes, q is fixed to have a characteristics of 2, and the code symbols are over

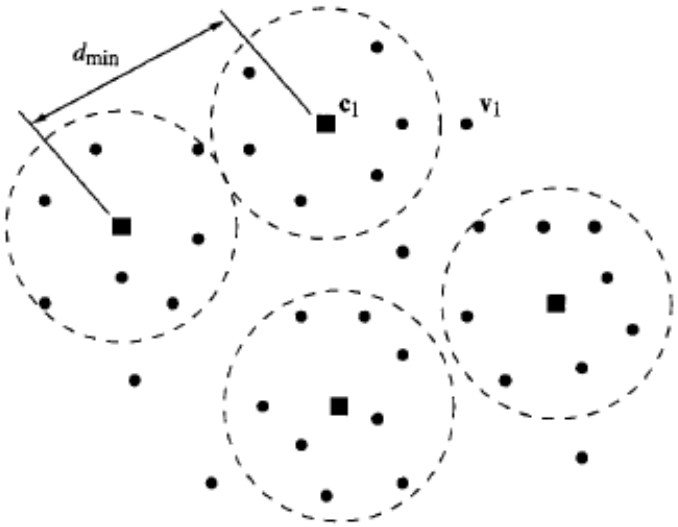

FIGURE 1.1: Decoding radius of distance d_{min} [1]

the Galois field $GF(2^m)$, where m is any positive integer. Constructing $GF(q^m)$ is based on a monic primitive polynomial $p(x)$ of degree m over $GF(q)$ [2] and will be discussed in chapter 2.

Reed Solomon codes have the following important properties [2]:

1. The code length n is less than the size of the code alphabet q by a factor of one, $n = q - 1$.

2. The minimum distance is greater than the number of parity check symbols by a factor of one, $d_{min} = n - k + 1$, where k is the dimension.

3. They are guaranteed to correct errors up to the decoder's error correcting capability t, where
$$t = \left\lfloor \frac{n-k}{2} \right\rfloor .$$

Gorenstein and Zierler [3] in 1961 presented the first practically realizable RS decoding algorithm, while Chien [4] and Forney [5] in 1964 and 1965 respectively improved on the algorithm. Berlekamp [6] in 1967 presented an efficient iterative decoding algorithm for RS codes that is able to decode multiple errors at a time. In 1975,

Sugiyama, Kashara, Hirasawa, and Namekawa [7] showed a conceptually simple Euclid's algorithm for finding the greatest common divisor (GCD) of two polynomials having coefficients from Galois fields which can be easily implemented to efficiently decode RS codes. These algorithms can successfully decode error patterns up to half the minimum distance d_{min} of the RS code.

An RS decoder may have some vectors, such as $\mathbf{v}_1$, in Figure 1.1, which lies outside Hamming spheres surrounding the codewords. If $\mathbf{v}_1$ happens to be the received vector, the decoding rule says $\mathbf{v}_1$ should be decoded to the nearest codeword $\mathbf{c}_1$. However, this cannot happen because a minimum distance decoder can only decode vectors within spheres of radius t, and any attempt to gradually increase the decoding radius will possibly give more than one codeword in Hamming sphere of a given radius [8], whereby the decoder fails.

List decoding and bounded distance decoding are approaches that can be used to efficiently identify all the codewords in any Hamming sphere of a given radius and decode beyond the minimum distance decoder's maximum error correcting capability t. Sudan [9] proposed an algorithm for (n, k, d) RS codes in the time domain. For a factor, $l > 0$, the algorithm can enumerate all the codewords for a radius up to

$$\tau = n - (m + 1) - l(k - 1), \tag{1.1}$$

where m is the smallest positive integer that satisfies

$$(m + 1)(l + 1) + (k - 1)\binom{l + 1}{2} \geq n + 1. \tag{1.2}$$

The factor l determines a code rate restriction for Sudan's algorithm. For instance, if $l = 2$, the code rate has a restriction, $R \leq 1/3$. Sudan's algorithm or list decoder produces a list of more than one codeword within the sphere radius of the received vector, based on polynomial interpolation and factorization.

Cheng and Wan [8] described the bounded distance decoding problem as finding a unique codeword if there is at least one codeword within a given distance, or produce

an empty set if there is none. The best known bounded distance algorithm is the one proposed by Schmidt, Sidorenko, and Bossert (SSB) [10]. They considered a syndrome based approach in the frequency domain, which is used to decode RS codes beyond half the minimum distance. The maximum decoding radius is:

$$t_{max} = \left\lfloor \frac{2ln - l(l+1)k + l(l-1)}{2(l+1)} \right\rfloor, \tag{1.3}$$

where l also dictates the rate restriction for increasing the decoding radius. SSB observed that for low rate codewords specified by (1.3) the probability of the bounded distance decoder's output containing only one codeword is very high, and therefore the BDD outputs the correctly decoded codeword. The decoder is allowed to fail with a small probability rather than having a list of solutions, while the technique was demonstrated to practically give the same decoding performance as the Sudan algorithm [9].

1.1 Research Question

The above observation by SSB then leads to investigating the relationship between increasing the decoding radius and the number of codewords contained in the decoding sphere for a given RS code rate. However, this turns out to be a very complicated problem to solve, and a subproblem to this is:

"*given a* $(7, k)$ *RS code, what is the probability* ρ *of obtaining a unique codeword when correcting up to* $t + 1$ *errors*?"

This research focuses on providing an answer to this question.

1.2 Outline

Chapter 2 reviews some algebraic properties such as rings and fields, and their relevance to error control in communication systems that includes codes over finite fields,

cyclic codes and Reed Solomon codes. Chapter 2 also covers relevant works on decoding RS codes, where minimum distance decoding and decoding beyond half the minimum distance are discussed in detail. Chapter 3 highlights the method used to analyze bounded distance decoding of RS codes. A framework is presented that shows the step taken to derive the probability of having unique codewords in the radius of a bounded distance decoder for $(7, k)$ RS codes. In Chapter 4, numerical analysis is done to examine $(7, k)$ RS codes of different rates. The performance of minimum distance decoding, and bounded distance decoding is compared. Finally, Chapter 5 summarizes the work done in this research and presents future works based on simulation results.

CHAPTER 2

Literature Review

In this chapter, definition of terms and algebraic basics are introduced as a background. Sections 2.1.1 and 2.1.2 deals with algebraic structures and manipulations, which emphasizes the importance of understanding fundamental algebraic structures that connects both fields and polynomials. Section 2.1.3 discusses some intrinsic algebraic properties for cyclic codes and RS codes, while section 2.1.4 shows how the generator and parity check matrices are constructed from literatures. Also, relevant work done in decoding RS codes using minimum distance decoder is reviewed in Section 2.2 and decoding beyond half the minimum distance in Section 2.3.

2.1 Definitions and Algebraic Basics

2.1.1 Commutative Rings and Fields

A commutative ring is obtained by a triple $(R, +, \cdot)$, where $R = \{0, 1\}$ and $+, \cdot$ are functions mapping $R \times R$ to R. For every $a, b, c \in R$, the commutative ring satisfies the following properties:

- **Associativity**: Both addition (+) and multiplication (.) are associative, that is, $a + (b + c) = (a + b) + c$ and $a \cdot (b \cdot c) = (a \cdot b) \cdot c$
- **Commutativity**: Both addition (+) and multiplication ($\cdot$) are commutative, that is, $a + b = b + a$ and $a \cdot b = b \cdot a$
- **Distributivity**: a, b, c distributes over addition (+), that is, $a \cdot (b + c) = a \cdot b + a \cdot c$

- **Identity**: $a + 0 = a$ and $a \cdot 1 = a$
- **Additive Inverses**: For every $a \in R$, there exists $-a \in R$ such that $a+(-a) = 0$

Moreso, if there exists a multiplicative inverse for every non-zero element, then R becomes a ***field****.*

In [1], a ring of polynomials over R, denoted as $R[x]$, is created from a given ring R, and an indeterminate symbol x. Given R and indeterminate x, the set $R[x]$ has $u = \langle u_0, \ldots, u_l \rangle$ finite sequences of R, which is referred to as the formal sum $\sum_{i=0}^{l} u_i x^i$. If $u = \langle u_0, \ldots, u_l \rangle$ and $v = \langle v_0, \ldots, v_k \rangle$ with $l \leq k$ then addition and multiplication over $R[x]$ are defined consecutively as $u + v = \langle u_0 + v_0, \ldots, u_l + v_l, v_{l+1}, \ldots, v_k \rangle$ and $u \cdot v = \langle w_0, \ldots, w_{l+k} \rangle$ where $w_i = \sum_{j=0}^{i} u_j v_{i-j}$. Also, the degree of a polynomial $u \in R[x]$, $\deg(u)$, given by $u = \sum_{i=0}^{\tau} u_i v^i$ is τ, where τ is any non-zero positive integer.

2.1.2 Codes over Finite Fields

Fields enables algebraic manipulations to be done efficiently. For instance, addition, multiplication, subtraction, and division can be defined in a field. Examples of fields include the field of real numbers $\mathbb{R}$, field of complex numbers $\mathbb{C}$, field of rational numbers $\mathbb{Q}$, etc. For the purpose of this research, the focus is on finite fields $\mathbb{F}$ or Galois field GF with finite number of elements. This is because finite field arithmetics are important algebraic tools used to construct and decode RS codes.

A field of order q is denoted by $\mathbb{F}_q$ or $GF(q)$ and if the order is a prime p, then the field is a prime field $GF(p)$. If $q = p^m$, where m is any positive integer, then $GF(q)$ can be extended, denoted by $GF(p^m)$ which contains a total of p^m elements. $GF(q)$ has at least one primitive element with order $(q-1)$ whose powers form the set of all nonzero elements of the field [2]. An example is to assume α to be primitive in $GF(q)$, then the $(q-1)$ distinct consecutive powers of α, $\{1, \alpha, \alpha^2, \ldots, \alpha^{q-2}\}$ are the $(q-1)$

nonzero elements of $GF(q)$. This exponential representation of a GF element is known as the power representation and used to describe the multiplication operation in [2]. An alternative to power representation is polynomial representation, which is used for elements of the extension field $GF(p^m)$.

The following property is useful for polynomials over $GF(2)$:

$$f^2(X) = f(X^2), \tag{2.1}$$

see derivation in [1], which is true for any $i \geq 0$, such that

$$f^{2^i}(X) = f(X^{2^i}). \tag{2.2}$$

Thus, it can be inferred from (2.2) that if β is a root of the polynomial over a $GF(p)$, then the conjugates of β, β^{2^i} must also be its root, that is, $f(\beta^{2^i}) = f^{2^i}(\beta) = 0$. Blahut showed in [11] that saying the roots of polynomials equals zero in the time domain is similar to saying the spectral components of a finite field transform is equal to zero.

Definition: A polynomial $\pi(x) \in \mathbb{F}[x]$ of degree m is said to be irreducible (or non-factorable) over $GF(p)$ if it is not divisible by any polynomial in $\mathbb{F}[x]$ of degree less than m.

An important property of irreducible polynomials is that $\pi(x)$ divides $x^{p^m-1}-1$ with no remainder [1]. Also, $\pi(x)$ is said to be primitive if the smallest positive integer n for which $\pi(x)$ divides $x^n - 1$ is $n = p^m - 1$. The roots are also primitive elements of some extension field $GF(p^m)$.

2.1.3 Cyclic Codes

Cyclic codes have lots of intrinsic algebraic properties which makes them efficient for error control in communication systems. Algebraic coding experts have constructed

several classes of cyclic codes such as BCH codes, Reed Solomon codes, etc, that are used for error correction.

An (n, k) code $\mathcal{C}$ over $GF(q)$ contains all vectors $\mathbf{c}$ of length n called codewords, where $\mathbf{c}$ forms a k-dimensional subspace in $GF(q)$. The code sends k information symbols using n symbols, and therefore have rate $R = k/n$. If the length n have a Fourier transform, then there exists a spectrum for each codeword in an extension field $GF(p^m)$ known as the frequency domain codeword [11].

Definition: An (n, k) linear code C is said to be cyclic over $GF(q)$, if every cyclic shift of a codeword $\mathbf{c} \in C$ is a codeword in C and the linear combination of any two codewords gives a codeword [1].

From [1], cyclic codes form ideals in a ring of polynomials having the following properties:

1. A cyclic code has a unique minimal monic polynomial (coefficients of leading term is equal to 1). This polynomial is known as the generator polynomial $g(x)$ of degree $n - k$ over $GF(q)$ that generates the code.

$$g(x) = g_0 + g_1 x + g_2 x^2 + \dots g_{n-k} x^{n-k}, \tag{2.3}$$

   where $n - k$ is the number of parity check symbols of the code.

2. Each codeword is represented by a code polynomial $c(x)$ of degree $n - 1$ which is a multiple of $g(x)$

$$c(x) = m(x)g(x), \tag{2.4}$$

   where $m(x)$ is an information polynomial of degree $k - 1$,

$$m(x) = m_0 + m_1 x + \cdots + m_{k-1} x^{k-1}$$

   and the coded symbols are given as

$$c(x) = (g_0 + g_1 x + g_2 x^2 + \dots g_{n-k} x^{n-k})(m_0 + m_1 x + \cdots + m_{k-1} x^{k-1}) \tag{2.5}$$

3. $g(x)$ is a factor of $x^n + 1$ in the ring $GF(q)[x]$

Given $g(x)$ of an (n, k) cyclic code, the coded symbols in (2.5) can be expressed systematically, such that the k digits on the right of each codeword are unchanged information symbols, and the $n - k$ digits on the left are parity-check symbols. This is explained in the following encoding steps [2]:

1. **Premultiplying** $m(x)$ by x^{n-k} gives a polynomial of degree $\leq n - 1$

$$m(x)x^{n-k} = m_0x^{n-k} + m_1x^{n-k+1} + \cdots + m_{k-1}x^{n-1}. \tag{2.6}$$

2. **Dividing** $m(x)x^{n-k}$ by $g(x)$ to obtain the quotient $a(x)$ and remainder $b(x)$ (the parity-check symbols)

$$m(x)x^{n-k} = a(x)g(x) + b(x), \tag{2.7}$$

degree of $b(x)$ is less than or equal to the degree of $g(x)$ that is, $b(x) = b_0 + b_1x + \cdots + b_{n-k-1}x^{n-k-1}$. Rearranging (2.7) yields

$$m(x)x^{n-k} + b(x) = a(x)g(x). \tag{2.8}$$

The polynomial in (2.8) is a code polynomial generated by $g(x)$, since it is a multiple of the generator polynomial.

3. **Expanding** the left hand side expression of (2.8) gives

$$b(x)+m(x)x^{n-k} = b_0+b_1x+\cdots+b_{n-k-1}x^{n-k-1}+m_0x^{n-k}+m_1x^{n-k+1}+\cdots+m_{k-1}x^{n-1}, \tag{2.9}$$

that equates to the codeword

$$(b_0, b_1, \ldots, b_{n-k-1}, m_0, m_1, \ldots, m_{k-1}). \tag{2.10}$$

2.1.4 Generator and Parity Check Matrices

Using coefficients of k code polynomial, the non-systematic encoding operation in (2.5) can be written as

$$c(x) = \left[m_0 + m_1 x + \cdots + m_{k-1}x^{k-1}\right] \begin{bmatrix} g(x) \\ xg(x) \\ x^2 g(x) \\ \vdots \\ x^{k-1}g(x) \end{bmatrix}, \tag{2.11}$$

and also as

$$c(x) = \left[m_0 + m_1 x + \cdots + m_{k-1}x^{k-1}\right] \underbrace{\begin{bmatrix} g_0 & g_1 & \cdots & g_{n-k} & 0 & 0 & \cdots & 0 \\ 0 & g_0 & g_1 & \cdots & g_{n-k} & 0 & \cdots & 0 \\ 0 & 0 & g_0 & g_1 & \cdots & g_{n-k} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & g_0 & g_1 & \cdots & \cdots & g_{n-k} \end{bmatrix}}_{G}, \tag{2.12}$$

where G is a non-systematic $(k \times n)$ generator matrix.

Also, since $g(x)$ divides $x^n + 1$, then there exists a polynomial $h(x)$ of degree k, such that

$$h(x) = (x^n + 1)/g(x), \tag{2.13}$$

known as the parity check polynomial of the cyclic code $\mathcal{C}$. A corresponding parity check matrix can be constructed from $h(x)$ as follows [1]: Suppose $c(x)$ is a code polynomial in $\mathcal{C}$, and for $m(x) = m_0 + m_1 x + \cdots + m_{k-1}x^{k-1}$, $c(x) = m(x)g(x)$.

Multiplying $c(x)$ by $h(x)$ gives

$$\begin{aligned} c(x)h(x) &= m(x)g(x)h(x) \\ &= m(x)(x^n + 1) \\ &= m(x) + m(x)x^n. \end{aligned}$$

Since degree of $m(x)$ is less than k, then powers of $x^k, x^{k+1}, \ldots, x^{n-1}$ are absent in $m(x) + m(x)x^n$. Hence, the coefficients of $x^k, x^{k+1}, \ldots, x^{n-1}$ in $c(x)h(x)$ must be equal to 0, and the following equalities are obtained:

$$\sum_{i=0}^{k} h_i c_{l-i} = 0 \,\forall \quad l = k, k+1, \ldots, n-1. \tag{2.14}$$

Taking the reciprocal of $h(x)$, the following polynomial is obtained

$$x^k h(x^{-1}) \equiv h_k + h_{k-1}x + h_{k-2}x^2 + \cdots + h_0 x^k, \tag{2.15}$$

which is also a factor of $x^n + 1$. Equation (2.14) can be written as follows:

$$\underbrace{\begin{bmatrix} h_k & h_{k-1} & h_{k-2} & \ldots & h_0 & 0 & 0 & \ldots & 0 \\ 0 & h_k & h_{k-1} & h_{k-2} & \ldots & h_0 & 0 & \ldots & 0 \\ 0 & 0 & h_k & h_{k-1} & h_{k-2} & \ldots & h_0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & h_k & h_{k-1} & \ldots & h_{k-2} & \ldots & h_0 \end{bmatrix}}_{H} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{n-1} \end{bmatrix} = \mathbf{0}, \tag{2.16}$$

where H is an $(n-k) \times n$ parity check matrix that generates an $(n, n-k)$ cyclic code.

A systematic expression for the generator matrix is also derived by dividing x^{n-k+i} by the generator polynomial $g(x)$, where $i = 0, 1, \ldots, k-1$ to obtain quotient q_i and remainder b_i

$$x^{n-k+i} = q_i(x)g(x) + b_i(x). \tag{2.17}$$

The remainder b_i is of the form:

$$b_i(x) = b_{i,0} + b_{i,1}x + \cdots + b_{i,n-k-1}x^{n-k-1},$$

and by (2.17),

$$x^{n-k+i} - b_i(x) = q_i(x)g(x). \tag{2.18}$$

The expression on the left side of (2.18) is a multiple of $g(x)$ and must therefore be code polynomials. Arranging the codewords for $i = 0, 1, \ldots, k-1$ as rows of the generator matrix yields

$$\mathbf{G} = \begin{bmatrix} b_{0,0} & b_{0,1} & b_{0,2} & \ldots & b_{0,n-k-1} & 1 & 0 & 0 & \ldots 0 \\ b_{1,0} & b_{1,1} & b_{1,2} & \ldots & b_{1,n-k-1} & 0 & 1 & 0 & \ldots 0 \\ b_{2,0} & b_{2,1} & b_{2,2} & \ldots & b_{2,n-k-1} & 0 & 0 & 1 & \ldots 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ b_{k-1,0} & b_{k-1,1} & b_{k-1,2} & \ldots & b_{k-1,n-k-1} & 0 & 0 & 0 & \ldots 1 \end{bmatrix}, \tag{2.19}$$

and the corresponding systematic parity check matrix for $\mathcal{C}$ is

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & \ldots & 0 & b_{0,0} & b_{1,0} & b_{2,0} & \ldots & b_{k-1,0} \\ 0 & 1 & 0 & \ldots & 0 & b_{0,1} & b_{1,1} & b_{2,1} & \ldots & b_{k-1,1} \\ 0 & 0 & 1 & \ldots & 0 & b_{0,2} & b_{1,2} & b_{2,2} & \ldots & b_{k-1,2} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 1 & b_{0,n-k-1} & b_{1,n-k-1} & b_{2,n-k-1} & \ldots & b_{k-1,n-k-1} \end{bmatrix}. \tag{2.20}$$

After studying the algebraic properties, and encoding of a cyclic code, the next section discusses Bose, Chaudhuri, and Hocquenghem (BCH) codes. The BCH code is categorized into binary and non-binary which forms a major subclass of cyclic codes.

2.2 BCH and Reed Solomon Codes

BCH codes can be described by a generator polynomial [1] because they are cyclic codes. The codes are categorized into binary and nonbinary codes. In 1961, Gorenstein and Zierler [3] generalized binary BCH codes to codes in p^m symbols, where p is prime. However, in this research, nonbinary BCH codes (n, k) with symbols from $GF(q)$ is discussed.

A subclass of nonbinary BCH codes is the Reed Solomon code having symbols in $GF(q)$ and are commonly used to correct multiple errors. It is usually with block-length $n = q - 1$, number of parity-check symbols $n - k$, dimension k, minimum distance $d_{min} = n - k + 1$, and t-error-correcting-capability, $t = (n - k)/2$ [2]. RS codes are important because of their effectiveness in correcting random errors and random burst errors. They consist of sequences of length n whose roots comprises of $2t$ successive powers of the primitive element of $GF(q)$ in the time domain, while the Fourier transform over $GF(q)$ will have $2t$ successive zeros.

Given α as a primitive element in $GF(q)$, the generator polynomial $g(x)$ of an RS code with t-error-correcting capability over $GF(q)$ has all its roots as $2t$ consecutive powers of the primitive element, $\alpha, \alpha^2, \ldots, \alpha^{2t}$. Since the primitive elements are roots of $x^n - 1$, then $g(x)$ divides $x^n - 1$. That is

$$\begin{aligned} g(x) &= \prod_{i=1}^{n-k} (x - \alpha^i) \\ &= g_0 + g_1 x + g_2 x^2 + g_{2t-1} x^{2t-1} + x^{2t}. \end{aligned} \tag{2.21}$$

2.3 Minimum Distance Decoding of RS Codes

Decoding RS codes is discussed and compared from the structural perspective in [12]. Whenever a codeword $c(x)$ of degree $n-1$ is transmitted over a channel, the decoding problem is described as finding the error polynomial $e(x)$ of degree $n - 1$ with the fewest possible number of non-zero coefficients that generate a received polynomial

$r(x)$ at the decoder's input. That is

$$c(x) = c_0 + c_1 x + \cdots + c_{n-1} x^{n-1},$$

and the channel adds error pattern

$$e(x) = e_0 + e_1 x + \cdots + e_{n-1} x^{n-1}.$$

Then, the received codeword at the decoder's input is

$$\begin{aligned} r(x) &= r_0 + r_1 x + \cdots + r_{n-1} x^{n-1} \\ &= c(x) + e(x). \end{aligned} \tag{2.22}$$

Suppose w-errors, $0 \leq w \leq t$ occur in locations $i_1, i_2, ..., i_w$ which are unknown. Then, we must know the error locations x^{i_j}'s and the error values e_{i_j}'s, where $j = 1, \ldots, w$ before we can determine the error polynomial, which is represented by

$$e(x) = e_{i_1} x^{i_1} + e_{i_2} x^{i_2} + \cdots + e_{i_w} x^{i_w}.$$

The received polynomial in (2.22) is evaluated at the roots, $\alpha, \alpha^2, ..., \alpha^{2t}$ of the generator polynomial $g(x)$ and used to compute a syndrome of length $2t$ over $GF(q^m)$. For $1 \leq i \leq 2t$,

$$S_i = r(\alpha^i), \tag{2.23}$$

and the syndrome polynomial

$$S(x) = \sum_{i=1}^{2t} S_i x^{i-1}.$$

From (2.22),

$$S_i = c(\alpha^i) + e(\alpha^i) = e(\alpha^i)$$

Hence, the error locations and error values are related to the syndrome by the following equation [2]:

$$\begin{aligned} S_1 &= e_{i_1}\alpha^{i_1} + e_{i_2}\alpha^{i_2} + \cdots + e_{i_w}\alpha^{i_w} \\ S_2 &= e_{i_1}\alpha^{2i_1} + e_{i_2}\alpha^{2i_2} + \cdots + e_{i_w}\alpha^{2i_w} \\ &\vdots \\ S_{2t} &= e_{i_1}\alpha^{2ti_1} + e_{i_2}\alpha^{2ti_2} + \cdots + e_{i_w}\alpha^{2ti_w}, \end{aligned} \tag{2.24}$$

where α^{i_j} for $j = 1, 2, \ldots w$ need to be determined from the syndrome components S_i's. Let the error location numbers $\alpha^{i_j} = \beta_j$, and $e_{i_j} = \delta_j$ then (2.24) can be written as:

$$\begin{aligned} S_1 &= \delta_1\beta_1 + \delta_2\beta_2 + \cdots + \delta_w\beta_w \\ S_2 &= \delta_1\beta_1^2 + \delta_2\beta_2^2 + \cdots + \delta_w\beta_w^2 \\ &\vdots \\ S_{2t} &= \delta_1\beta_1^{2t} + \delta_2\beta_2^{2t} + \cdots + \delta_w\beta_w^{2t}. \end{aligned} \tag{2.25}$$

The Error location polynomial $\sigma(x)$ is defined as [2]

$$\begin{aligned} \sigma(x) &= (x - \beta_1)(x - \beta_2)\ldots(x - \beta_w) \\ &= \sigma_0 + \sigma_1 x + \sigma_2 x^2 + \cdots + \sigma_w x^w. \end{aligned} \tag{2.26}$$

Example 1.

Consider a $(15, 9)$ 3-error correcting RS code. Let the transmitted codeword be

$$v = \{\underbrace{\alpha^4\ \alpha^8\ \alpha^{10}\ \alpha^{12}\ \alpha^1\ \alpha^{13}\ \alpha^9\ \alpha^5\ \alpha^4}_{message}\ \underbrace{\alpha^{13}\ \alpha^0\ \alpha^{11}\ \alpha^8\ \alpha^0\ \alpha^1}_{parity}\},$$

and

$$\boldsymbol{r} = \{\alpha^4\ \alpha^8\ \alpha^8\ \alpha^{12}\ \alpha^1\ \alpha^{13}\ \alpha^9\ \alpha^5\ \alpha^4\ \alpha^{13}\ \alpha^0\ \alpha^{11}\ \alpha^8\ \alpha^8\ \alpha^{12}\}$$

was received. Suppose the error locations x^{i_j} are known, where $i = 3, 14, 15$, and from (2.26), $\beta_1 = \alpha^8, \beta_2 = \alpha^8, \beta_3 = \alpha^{12}$. By expanding (2.26), we find that

$$\sigma(x) = \alpha^0 x^3 + \alpha^6 x^2 + 0x + \alpha^{13}$$

and the coefficients of $\sigma(x)$ are the error locators $\sigma = \{\alpha^0\ \alpha^6\ 0\ \alpha^{13}\}$.

For unknown error locations, the error locator can be found using the syndromes. A linear relationship between the syndromes S_i's and coefficients σ_i's of the error locator polynomial is described by the generalized Newton identities:

$$\begin{aligned} S_{w+1} + \sigma_1 S_w + \sigma_2 S_{w-1} + \cdots + \sigma_w S_1 &= 0 \\ S_{w+2} + \sigma_1 S_{w+1} + \sigma_2 S_w + \cdots + \sigma_w S_2 &= 0 \\ &\vdots \\ S_{2t} + \sigma_1 S_{2t-1} + \sigma_2 S_{2t-2} + \cdots + \sigma_w S_{2t-w} &= 0, \end{aligned} \tag{2.27}$$

which has been derived in [2], [1]. This relationship is similar to a linear feedback shift register, where

$$S_i = -\sum_{j=1}^{w} \sigma_i S_{i-j},\ i = w+1, w+2, \ldots, 2w \tag{2.28}$$

and can be expressed in a matrix form

$$\begin{pmatrix} S_1 & S_2 & \ldots & S_w \\ S_2 & S_3 & \ldots & S_{w+1} \\ \vdots & \vdots & \ddots & \vdots \\ S_w & S_{w+1} & \ldots & S_{2w-1} \end{pmatrix} \begin{pmatrix} \sigma_w \\ \sigma_{w-1} \\ \vdots \\ \sigma_1 \end{pmatrix} = \begin{pmatrix} S_{w+1} \\ S_{w+2} \\ \vdots \\ S_{2w} \end{pmatrix}.$$

Therefore, a key equation that relates coefficients of $\sigma(x)$ and syndrome components S_i can be formed as follows:

$$S(x)\sigma(x) = T(x) \mod x^{2t}. \tag{2.29}$$

Finding error location polynomial $\sigma(x)$ is the most difficult task and any algorithm that solves for $\sigma(x)$ is known as the decoding algorithm for RS codes. The error location polynomial can be obtained by using either the Peterson-Gorenstein-Zieler (PGZ) Algorithm [3], or the Berlekamp Massey (BM) Algorithm [2],[13], or the Extended Euclidean Algorithm (EEA) [2],[6].

2.3.1 Peterson-Gorenstein-Zieler Algorithm

The PGZ algorithm [14] is one of the algorithms that is capable of finding the error locator polynomial in (2.29). The first step is to determine the number of errors t by finding the rank of $\boldsymbol{S}$, where rank of $\boldsymbol{S} = t$. For an assumed number of errors $w \leq t$, it is possible to construct the square matrix $\boldsymbol{M}_w$ by picking the first w rows of $\boldsymbol{S}$ from (2.29), which results in a unique error locator polynomial. However, suppose too many errors are assumed where $w > t$, then $\boldsymbol{M}_w$ becomes singular. The next step is to row reduce w stepwise until the row where the first value of $w = t$ and $\boldsymbol{M}_w$ becomes nonsingular. Therefore an error locator polynomial $\sigma(x)$ is obtained that may not have deg $\sigma(x)$ distinct roots or the roots fall in the wrong field. For instance, the root may either be repeated or lie in the extension of the field where the operations are performed. If this was to happen during decoding, the decoder declares a decoding failure. Otherwise, solve for the coefficients $\sigma_1, \sigma_2, \ldots, \sigma_w$

$$\sigma = \boldsymbol{M}_t^{-1} \boldsymbol{T}_t, \tag{2.30}$$

where $\boldsymbol{T}_t$ has the first t entries of $\boldsymbol{T}$.

2.3.2 Berlekamp-Massey Algorithm

The BM algorithm is a more efficient approach for finding the error locator polynomial in (2.29) with computational complexity of $O(t^2)$ as compared to the PGZ algorithm with computational complexity of $O(t^3)$ [1].

Recall that determining the error location polynomial with coefficients $\sigma_1, \sigma_2, \ldots, \sigma_w$ in (2.29) is similar to a shift-register synthesis problem of finding the smallest w that generates the syndrome components S_i. Therefore, the BM algorithm constructs the linear feedback shift register (LFSR) that yields the complete syndrome sequence $\{S_1, S_2, \ldots, S_{2t}\}$.

The Berlekamp-Massey algorithm involves computing $\sigma(x)$ iteratively in $2t$ steps [2]. At the μth step, a polynomial

$$\sigma^{(\mu)}(x) = \sigma_0^{(\mu)} + \sigma_1^{(\mu)}x + \cdots + \sigma_{l_\mu}^{(\mu)}x^{l_\mu}$$

of minimum degree is determined. The coefficients of $\sigma^{(\mu)}(x)$ also satisfy the following $\mu - l_\mu$ identities [2]:

$$\begin{aligned} S_{l_\mu+1} + \sigma_1^{(\mu)}S_{l_\mu} + \cdots + \sigma_{l_\mu}^{(\mu)}S_1 &= 0 \\ S_{l_\mu+2} + \sigma_1^{(\mu)}S_{l_\mu+1} + \cdots + \sigma_{l_\mu}^{(\mu)}S_2 &= 0 \\ &\vdots \\ S_\mu + \sigma_1^{(\mu)}S_{\mu-1} + \cdots + \sigma_{l_\mu}^{(\mu)}S_{\mu-l_\mu} &= 0. \end{aligned} \tag{2.31}$$

If this is satisfied, proceed to the next step. For some $\mu + 1 < 2t$, find a new polynomial

$$\sigma^{\mu+1}(x) = \sigma_0^{(\mu+1)} + \sigma_1^{(\mu+1)}x + \ldots \sigma_{l_{\mu+1}}^{(\mu+1)}x^{l_{\mu+1}}$$

of minimum degree having coefficients that satisfies the $(\mu+1) - l_{\mu+1}$ identities. The iteration process continues until the $2t$ steps are completed. Then,

$$\sigma(x) = \sigma^{(2t)}(x).$$

If $\sigma(x)$ have a degree greater than t, it implies more than t errors have been received, and the minimum distance decoder will not be able to locate them.

Assume the solution $\sigma^{(\mu)}(x)$ is found after completing the μth step. To determine $\sigma^{(\mu+1)}$, check if the coefficients of $\sigma^{(\mu)}(x)$ satisfy the next generalized Newton's identity

$$S_{\mu+1} + \sigma_1^{(\mu)}S_\mu + \cdots + \sigma_{l_\mu}^{(\mu)}S_{\mu+1-l_\mu} = 0. \tag{2.32}$$

If (2.32) is equal to zero, then the minimum degree polynomial having its coefficients satisfy the generalized Newton's identities of (2.27) is $\sigma^{(\mu+1)}(x) = \sigma^{(\mu)}(x)$. However, if (2.32) is not equal to zero, a correction term is added to $\sigma^{(\mu)}(x)$ for the coefficients

to satisfy the set of identities in (2.27).

2.3.3 Extended Euclidean Algorithm

Apart from the Berlekamp-Massey (BM) algorithm, the Extended Euclidean algorithm (EEA) can be used to determine the error locator polynomial. This method finds the greatest common divisor (GCD) of two polynomials $a(x)$ and $b(x)$, over $GF(q)$ with deg $a(x) \geq$ deg $b(x)$, The algorithm yields both error locator polynomial and the error evaluator. The gcd$[a(x), b(x)]$ can be found by iteratively dividing $a(x)$ by $b(x)$ to obtain the quotient $q_i(x)$ and remainder $r_i(x)$ at the ith iteration as follows [2]:

$$\begin{aligned} a(x) &= q_1(x)b(x) + r_1(x) \\ b(x) &= q_2(x)r_1(x) + r_2(x) \\ &\vdots \\ r_{i-2}(x) &= q_i(x)r_{i-1}(x) + r_i(x) \\ &\vdots \\ r_{n-1}(x) &= q_{n+1}(x)r_n(x) + 0. \end{aligned} \tag{2.33}$$

The iteration stops when the remainder is equal to zero, therefore the gcd of $a(x)$ and $b(x)$ is the non-zero last remainder $r_n(x)$. An important property of the Extended Euclid algorithm is that gcd$[a(x), b(x)]$ can be expressed as multiples of two other polynomials $f(x)$ and $g(x)$ respectively over $GF(x)$. Hence, the remainder $r_n(x)$ can be expressed as [2]

$$\begin{aligned} r_1(x) &= f_1(x)a(x) + g_1(x)b(x) \\ r_2(x) &= f_2(x)a(x) + g_2(x)b(x) \\ &\vdots \\ r_i(x) &= f_i(x)a(x) + g_i(x)b(x) \\ &\vdots \\ r_n(x) &= f_n(x)a(x) + g_n(x)b(x), \end{aligned} \tag{2.34}$$

and for $1 \leq i \leq n$, the following recursive equations are obtained from (2.33) and (2.34) which solves for $r_i(x)$, $f_i(x)$, and $g_i(x)$:

$$\begin{aligned} r_i(x) &= r_{i-2}(x) - q_i(x) r_{i-1}(x) \\ f_i(x) &= f_{i-2}(x) - q_i(x) f_{i-1}(x) \\ g_i(x) &= g_{i-2}(x) - q_i(x) g_{i-1}(x) \end{aligned} \tag{2.35}$$

with initial conditions as

$$\begin{aligned} r_{-1}(x) &= a(x) \\ r_0(x) &= b(x) \\ f_{-1}(x) &= g_0(x) \quad = 1 \\ f_0(x) &= g_{-1}(x) \quad = 0. \end{aligned} \tag{2.36}$$

After finding the error location polynomial using any of the algorithm, the roots of the polynomial are then evaluated in $GF(q)$ to determine the error locations. Evaluating the roots can be done by substituting $\alpha^i, i = 0, 1, ..., n-1$ into the error polynomial and checking for a result equal to zero known as the Chien search method [4]. The error location numbers are then obtained by taking the reciprocal of the roots of the error location polynomial.

The final step of decoding using the minimum distance decoder is to determine the error values. These values are efficiently obtained using the Forney algorithm [5]. The Forney algorithm requires dividing the error value evaluator Z_0 by the formal derivative of the error locator polynomial over $GF(q)$. An expression for explicitly calculating Z_0 is given in [2] or it can be directly derived as one of the results from the Extended Euclidean Algorithm. Once the error values are evaluated, error correction is done by subtracting the error values from the received vector.

However, it is important to note that designers have a choice of decoding RS codes in the frequency domain. A frequency domain decoder can be implemented with any of the encoding techniques described in [13] and [11]. The choice depends on hardware and software implementation, available circuitry and the code parameters such as blocklength, and the minimum distance. However, the encoding method used

should be known before the information can be recovered. As illustrated in [11], if encoding is done in the frequency domain, then certain components of the spectrum will be specified by the information symbols whose inverse transform results in the time domain codeword. In this case, the corrected spectrum is the information.

To decode a received codeword $r(x) = r_0 + r_1x + \cdots + r_{n-1}x^{n-1}$ which is a noisy representation of an unknown codeword $c(x)$ from the (n, k) RS code in the frequency domain, the decoder first obtains a transform $R(X) = R_0 + R_1X + \cdots + R_{n-1}X^{n-1}$ of $r(x)$ and then computes the syndrome from the transform. $R(X)$ can be written as a vector $R_i = C_i + E_i$, for $0 \leq i < n$, where C_i is a Fourier transform of the codeword $c(x)$ and E_i is the error pattern. The codeword C_i is zero on a block of $2t$ components [11], which implies there are $2t$ known values of E_i referred to as the syndrome. The syndrome polynomial is therefore defined as $S(x) = S_0 + S_1x + \cdots + S_{2t}x^{2t}$, where $S_j = E_j = R_j = r(\alpha^j), j = 1, 2, \ldots, 2t$.

Thereafter, the decoder finds error-locator polynomial $\sigma(x)$ with the smallest possible degree from the syndrome $S(x)$. Assume there are $v \leq t$ errors that occurred, then we define error-locator polynomial over $GF(q^m)$ as $\sigma(x) = \sigma_0 + \sigma_1x + \cdots + \sigma_{v-1}x^{v-1} + \sigma_vx^v$. This yields $v + 1$ coefficients σ_i, $i = 0, 1, \ldots, v$ that must be determined from t. The steps of computing coefficients of error-locator polynomial $\sigma(x)$ can be done by a Linear System of Equations (LSE) [13], the Extended Euclidean Algorithm (EEA) [13], or the Berlekamp-Massey algorithm [6], [15]. In particular, if the EEA is implemented, the procedure returns both error locator polynomial $\sigma(x)$ and error evaluator polynomial $T(x)$.

The final step is to determine the error pattern E_i, $i = 0, \ldots, n-1$. Since the E_j spectral components of E_i is already known, we must therefore find the spectral components E_l, $l = 2t+1, 2t+2, \ldots, n$. This is derived by finding S_l using the initial conditions S_j and the recursion that is based on the coefficients of $\sigma(x)$:

$$S_{l \bmod n} = -\sum_{i=0}^{v} \sigma_i S_{l-i}. \tag{2.37}$$

Thus, computing the syndrome $\mathbf{S} = (S_0, S_1, \ldots, S_{n-1})$ is completed, and decoding is concluded in the frequency domain by taking the inverse Fourier transform of $\mathbf{S}$.

2.4 Decoding of Reed Solomon Codes Beyond half the Minimum Distance

As discussed in Chapter 1, the minimum distance decoder fails whenever the channel introduces more errors than the error correcting capability of the decoder. A list decoding algorithm proposed by Sudan [9] is a well known method that computes the list of all possible codewords and increases the decoding radius to a certain bound as a way of decoding beyond the error correcting capability t. The algorithm considers decoding low rate RS codes as a bivariate polynomial interpolation and factorization problem, which can be solved in polynomial time. For a given upper bound l on the decoding radius, the algorithm corrects any combination of errors up to τ errors

$$\tau = n - (m+1) - l(k-1), \tag{2.38}$$

where m is the smallest positive integer that satisfies

$$(m+1)(l+1) + (k-1)\binom{l+1}{2} \geq n+1. \tag{2.39}$$

For example, when $l = 1$, Sudan obtained the smallest possible m that satisfies (2.39) as $m = \left[\frac{n-k}{2}\right]$, and (2.38) becomes

$$\tau = n - k - m = \left[\frac{n-k}{2}\right],$$

which is the same as the error correcting capability for minimum distance decoding. Since the variable l decides a code rate restriction on the Sudan algorithm, then for $l = 2$, Sudan reduced the maximum rate of the RS codes by assuming $k \leq (n+1)/3$.

This gives $m = \left[\frac{n+1}{3} - k\right]$ and

$$\tau = n + 1 - 2k - m = \left[\frac{2(n+1)}{3}\right] - k.$$

The algorithm has a high quadratic computational complexity.

However, Guruswani and Sudan (GS) in [16] introduced an improved list decoding algorithm for decoding RS codes, which reduces the problem of list decoding for RS codes to a curve-fitting problem over a field. The algorithm improves over the initial Sudan algorithm in [9], for all rates and has been extended to weighted curve fitting, motivated by the soft-decision decoding problem. A new list decoding method for RS codes and BCH codes based on a rational curve fitting algorithm is also discussed in [17]. Wu presented a novel polynomial algorithm with rational interpolation and factorization. The algorithm has the same list error correction capability as the GS algorithm, but lower computational complexity due to the low multiplicity.

Another approach to correcting errors beyond half the minimum distance is one that depend on calculating an extended syndrome from the received word, and finding an error location polynomial. Schmidt et al. [10] implemented a bounded distance decoding for RS codes that is based on the syndrome extension technique in the frequency domain. This technique practically gives the same decoding performance as the Sudan algorithm in [9], but has a lower computational complexity than the Sudan algorithm. Syndrome extension decoding involves embedding an RS code into an Interleaved Reed Solomon (IRS) code where element-wise operations is performed on the symbols of a low rate Reed Solomon code.

For syndrome extension decoding, the depth of extension l cannot be freely selected as in the case of IRS codes, because of the limitation imposed on it by the rate of the underlying RS code. However, if there exists an integer l in such a way that the following inequality [10]:

$$l(k-1) + 1 \leq n \tag{2.40}$$

is satisfied, the l codewords can be created as follows:

$$\boldsymbol{c}^{\langle i \rangle} = (c_0^i, \ldots, c_{n-1}^i),\ i = 1, \ldots, l,$$

for l different RS codes of same length n but increasing rate. See proof in Lemma 1, [10].

In [10], [18], a codeword was virtually extended to a sequence of interleaved codewords which resulted in a multi-sequence linear shift register synthesis problem of varying lengths. They implemented a joint or collaborative decoding method that employs the generalized Berlekamp-Massey algorithm for enhancing the decoding capability of the RS code. As a result, an upper bound for l was given as:

$$l_{max} \leq \left\lfloor \frac{\sqrt{(Rn+3)^2 + 8(Rn-1)(n-1)} - (Rn+3)}{2(Rn-1)} \right\rfloor, \tag{2.41}$$

where $l \leq l_{max}$. It was assumed in [10], [18] that the error patterns generated by the channel was introduced at the same position in all RS codes. Hence, the error location polynomial remains the same for all l extended RS codes, resulting in the same roots and thus the same number of unknowns. The syndrome of each of the l extended RS codes produces additional independent equations which enables errors to be corrected beyond half the minimum distance.

The following steps show how syndrome extension is computed by SSB in [10]. Suppose $r(x) = c(x) + e(x)$ is received at the channel output, then the l polynomials $r^{(i)}(x) = r_0^i + r_1^i(x) + \cdots + r_{n-1}^i x^{n-1}$, $i = 1, \ldots, l$ of $r(x)$ over $GF(q)$ can be calculated from the embedded RS code and their coefficients is used to form an $l \times n$ matrix

$$\mathbf{r} = \begin{pmatrix} r^{(1)} \\ r^{(2)} \\ \vdots \\ r^{(l)} \end{pmatrix} = \begin{pmatrix} r_0 & r_1 & \ldots & r_{n-1} \\ r_0^2 & r_1^2 & \ldots & r_{n-1}^2 \\ \vdots & \vdots & \ddots & \vdots \\ r_0^l & r_1^l & \ldots & r_{n-1}^l \end{pmatrix}. \tag{2.42}$$

The following properties [7] exist for matrix $\mathbf{r}$:

1. if $e(x) \equiv 0$, then $\mathbf{r}$ is a codeword of a heterogeneous IRS code composed of the l RS codes.

2. if $e(x)$ has t non zero coefficients $e_{j_1}, e_{j_2}, \ldots, e_{j_t}$, then $\mathbf{r}$ is corrupted with exactly t erroneous columns at the positions $j_1, j_2, \ldots, j_t$.

According to properties (1) and (2), syndromes $S^{(1)}, \ldots, S^{(l)}$ are calculated in the frequency domain by finding the Fourier transform $R^{(i)}(x)$ of extended $r^{(i)}(x)$, $i = 1, \ldots, l$ for each row of $\mathbf{r}$ and selecting the last $n - i(k-1) - 1$ coefficients of each polynomial obtained from the transform, which is given by:

$$S^{(i)} = (S_0^{(i)}, \ldots, S_{n-i(k-1)-2}^{(i)}) = (R_{i(k-1)+1}^{(i)}, \ldots, R_{n-1}^{(i)}). \tag{2.43}$$

Since the codes $r^{(i)}$ have different dimensions $k^{(i)}$, then the calculated extended syndrome $S^{(i)}$ will be of decreasing length.

Extended syndromes $S^{(i)}$ yields a linear system of equations with unknown coefficients $\sigma_1, \ldots, \sigma_t$, represented by the following equation [10]:

$$\underbrace{\begin{pmatrix} \boldsymbol{S}^{(1)} \\ \boldsymbol{S}^{(2)} \\ \vdots \\ \boldsymbol{S}^{(l)} \end{pmatrix}}_{\boldsymbol{S}} \underbrace{\begin{pmatrix} \sigma_t \\ \sigma_{t-1} \\ \vdots \\ \sigma_1 \end{pmatrix}}_{\boldsymbol{\sigma}} = \underbrace{\begin{pmatrix} \boldsymbol{T}^{(1)} \\ \boldsymbol{T}^{(2)} \\ \vdots \\ \boldsymbol{T}^{(l)} \end{pmatrix}}_{\boldsymbol{T}}, \tag{2.44}$$

where

$$\boldsymbol{S}^{(i)} = \begin{pmatrix} S_0^{(i)} & S_1^{(i)} & \cdots & S_{t-1}^{(i)} \\ S_1^{(i)} & S_2^{(i)} & \cdots & S_t^{(i)} \\ \vdots & \vdots & \ddots & \vdots \\ S_{(n-i(k-1)-t-1)}^{(i)} & S_{(n-i(k-1)-t-1)}^{(i)} & \cdots & S_{(n-i(k-1)-3)}^{(i)} \end{pmatrix},$$

and

$$\boldsymbol{T}^{(i)} = \begin{pmatrix} -S_t^{(i)} \\ -S_{t+1}^{(i)} \\ \vdots \\ -S_{(n-i(k-1)-2)}^{(i)} \end{pmatrix}.$$

As long as the rank of $\boldsymbol{S}$ is equal to t, then a unique solution for (2.44) is guaranteed. Thus, for a given l the maximum error correcting radius is determined by [18]:

$$t_{max}^{(l)} = \left[\frac{2ln - l(l+1)k + l(l-1)}{2(l+1)} \right]. \tag{2.45}$$

The key equation in (2.44) is similar to (2.29). However, considering the Hankel structure of submatrices $\boldsymbol{S}^{(i)}$ in (2.44), the following linear system for $\tau \leq t_{max}^{(l)}$ assumed errors can be represented by

$$S_j^{(i)} + \sigma_1 S_{j-1}^{(i)} + \cdots + \sigma_\tau S_{j-v}^{(i)} = 0 \ \forall \ j \in \left[\tau, n - k^{(i)} - 1\right], \tag{2.46}$$

and the problem of finding the error locator polynomial $\sigma(x)$ can now be reformulated as finding the smallest integer τ and an ordered set $\sigma_1, \ldots, \sigma_t$ of τ weights for recursively generating the i different syndrome sequences $S^{(i)} = \{S_b^{(i)}\}_{b=0}^{n-k^{(i)}-1}$ over a field F_q, $i = 1 \ldots, l$, known as a multi-sequence shift-register synthesis problem.

Feng and Tzeng presented a generalized Euclidean algorithm based on generalized polynomial division algorithm [19] and also a generalized Berlekamp-Massey algorithm [20] to solve the multi-sequence problem. They gave conditions for (2.44) to have a unique solution and whenever the solution is not unique, a set of all possible solutions was derived. However, the algorithm does not always result in the shortest shift-register when all l sequences is of varying length. This shortfall has been demonstrated in [21], where a modification to the generalized Euclidean algorithm is presented so that it is guaranteed to correctly solve the problem of varying length. An algorithm was further presented in [22] that solves the problem of varying length and calculated $\sigma(x)$. This algorithm in [22] was applied in [10], [18] to correct errors beyond half the minimum distance.

The algorithm in [10] achieves the desired result whenever a unique solution is obtained within a certain error correction capability $t_{max}^{(l)}$ which is greater than the error correcting capability t for conventional minimum distance decoding. SSB calculated the probability of not having unique codewords for a bounded distance decoder with radius t_{max}. The probability of decoding failure is upperbounded by [10]:

$$P_f(t) \leq \bar{P}_f(t) = \underbrace{\left(\frac{q}{q-1} + \frac{1}{q}\right)^t}_{\gamma} \cdot \frac{q^{-3\cdot(t_{max}-t)}}{q-1}, \tag{2.47}$$

where it was found that γ approaches one, and the probability of failure for radius t_{max}, $P_f(t_{max})$ is in order of $1/(q-1)$, which exponentially decreases with decreasing t.

The next chapter introduces a method of analysis that is based on the result obtained from calculating the probability of failure for the bounded distance decoder of radius t_{max} by SSB in [10]. To obtain dependable results, RS codes over $GF(8)$ are examined for decoding errors up to $t+1$ in contrast to decoding t errors.

CHAPTER 3

Research Methodology

This chapter discusses a specific method used to obtain the probability of unique codewords in the radius of a Bounded Distance Decoder for $(7, k)$ RS codes. Figure 3.1 shows the conceptual framework of this research. The method includes creating lookup tables, examining all $(7, k)$ RS codes using the lookup table, and calculating probabilities of every k for $t + 1$ errors. To create lookup tables, all possible error vector combinations are generated with each row mapped to a syndrome vector. Also, to examine all $(7, k)$ RS codes, the frequency of syndrome occurrences is checked to find the number of unique occurrences of each syndrome vector in the table. Thereafter, the probability of having unique codewords in the decoding radius of $(7, k)$ RS codes is calculated as a ratio of the number of unique syndrome occurrences and syndrome possibilities. The following subsections describe each research step in details.

3.1 Lookup Tables for $(7, k)$ RS Codes

Two cases are analyzed, and lookup tables are computed for both cases. The first case to be examined is the minimum distance decoder, which corrects all error patterns of $w \leq t$ in distance t, where $t = \frac{n-k}{2}$. We consider t-error-correcting $(7, k)$ RS codes with symbols from $GF(2^3)$, generated by roots $(\alpha, \ldots, \alpha^{n-k})$ of a generator polynomial. The Lookup table for different code rates is created in the following steps:

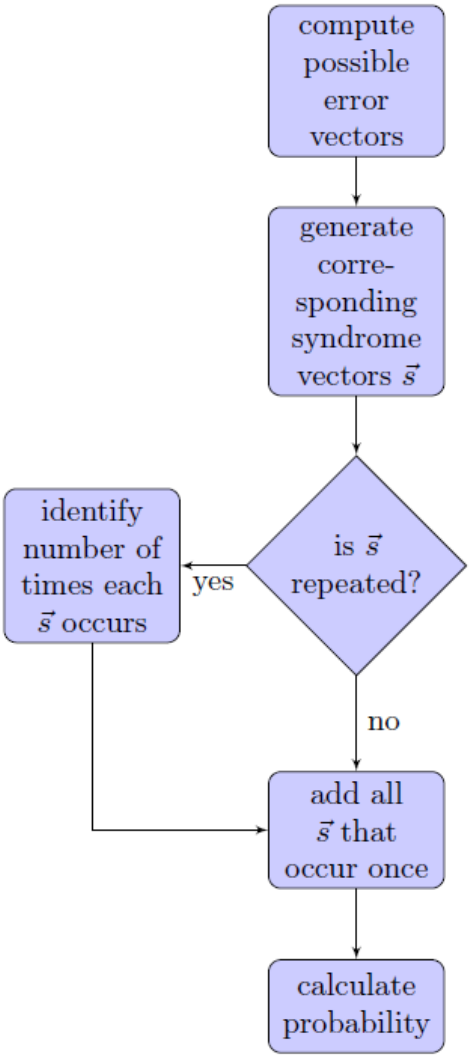

FIGURE 3.1: Conceptual framework for analyzing BDD of $(7, k)$ RS codes

- **Column of possible error vectors:** Recall from chapter 2 that $0 \leq w \leq t$ errors occur in unknown locations and the channel adds error pattern $e(x)$, which is a vector of length n and the elements are the coefficients of the polynomial $e(x)$. In this chapter, the number of possible error vectors from a field of q elements of length n and weight w, (w can be any field element apart from zero) is derived using the following combinatorial equation [23]:

$$\#\vec{\xi} = (q-1)^w \binom{n}{w}, \tag{3.1}$$

where $(q-1)^w$ is the error value, and $\binom{n}{w} = \frac{n!}{w!(n-w)!}$ is the number of unordered selections of w errors from a set of n vectors known as the error positions. The total number of possible error vectors $\vec{\varepsilon}$ is obtained as follows:

$$\vec{\varepsilon} = \sum_{w=1}^{t} (q-1)^w \binom{n}{w}. \tag{3.2}$$

- **Column of possible syndrome vectors:** The syndrome gives evidence of errors, which contains all the receiver's information about the errors [23]. From chapter 2, (2.24) relates the error locations and values to the syndrome of the received polynomial. This implies that corresponding syndrome vectors can be obtained from the possible error vectors. The column for possible syndrome vectors of length $n-k$, is derived by evaluating the value of each possible error vector at roots $(\alpha, \alpha^2, \ldots, \alpha^{n-k})$ of the generator polynomial $g(x)$.

  In order to decode errors correctly using the lookup table, all possible syndromes must be disjoint [23]. Since the syndromes are obtained from corresponding combination of error vectors, there may be one or more syndromes that are repeated, and must therefore be examined for uniqueness.

- **Column of unique syndrome occurrence:** Unique syndromes are obtained by searching through the column of possible syndrome vector and the frequency of syndrome occurrence (number of time each syndrome occurs) is observed. Syndromes with minimum occurrence of one are said to be unique, and placed in the column of unique syndrome occurrence in the lookup table.

Case 2 is to examine a bounded distance decoder, where it is assumed for $(7, k)$ RS code, the decoder may find a unique codeword if it exists within the decoder's radius $t+1$, and may correct error patterns of $w \leq t+1$ errors. We consider t-error-correcting $(7, k)$ RS codes with elements over $GF(2^3)$, and generated by $(n-k)$ consecutive roots, $(\alpha, \ldots, \alpha^{n-k})$ of $g(x)$. These codes containing M codewords with

q elements over the field must satisfy the following inequality [23]:

$$M\left(1+(q-1)\binom{n}{1}+\cdots+(q-1)^t\binom{n}{t}\right)\leq q^n. \tag{3.3}$$

The lookup table for different code rates is computed using the same steps for $w \leq t$.

3.2 Probability of Unique Codeword for $(7, k)$ RS Code

The probability of having unique codewords in decoding radius $[t, t+1]$ is obtained from the frequency of occurrence and possible syndrome vectors as given by the following equation:

$$\rho = \frac{\sum(Unique\ Syndrome\ Occurrence)}{\sum(Possible\ Syndrome\ Vectors)}. \tag{3.4}$$

The method is demonstrated in the next chapter, and the result shows the probabilities of having unique codewords when implementing $(7, k)$ RS codes for minimum distance decoding of radius t and bounded distance decoding of radius $t+1$.

3.3 Performance Analysis for Low Rate RS code

To complete the analysis, performance of the bounded distance decoder is examined for low rate RS code ($R \leq 1/3$) and the result is compared with the conventional minimum distance decoder. Figure 3.2 is a block diagram that shows the procedure for decoding low code rates using both the minimum distance decoder that corrects t errors and the bounded distance decoder that corrects $t+1$ errors.

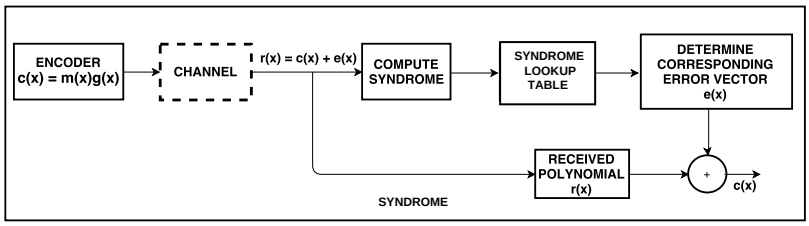

FIGURE 3.2: Experimental setup for decoding low rate RS codes

CHAPTER 4

Results and Analysis

This chapter provides numerical analysis to examine $(7, k)$ RS codes of different code rates (k/n), where $n = 7$ and $k = 2, 3, 4$, and 5. The research methodology discussed in Chapter 3 is implemented to create lookup tables, analyze the codes and calculate the probabilities of obtaining unique codewords of $\lfloor t, t+1 \rfloor$ radius in both minimum and bounded distance decoders. The following subsections provide numerical analysis of $(7, k)$ RS codes for t and $t + 1$ radius.

4.1 Minimum Distance Decoding for $w \leq t$

Recall from chapter 2 that given a received codeword $\mathbf{x} \in GF(q)$, minimum distance decoding chooses the codeword $\mathbf{y} \in \mathcal{C}$, which is as close as possible to $\mathbf{x}$. The following subsection provide numerical analysis of $(7, k)$ RS codes for decoding radius t.

4.1.1 $(7, 5)$ RS Code

Consider a one-error-correcting $(7, 5)$ RS code with symbols from $GF(2^3)$ generated by consecutive roots (α, α^2) of a generator polynomial. Assume the channel introduced one error $(w = 1)$ to the transmitted message, a lookup table that contains columns of possible combinations of error vectors and corresponding syndrome vectors $(\vec{\chi_1})$ is created. For a $(7, 5)$ RS code, the table has 49 possible error vectors obtained using (3.1), each error vector corresponds to a syndrome vector of length 2.

Also, the frequency of syndrome occurrences from the possible syndrome column ($\vec{\chi_1}$) indicates all 49 syndromes occurred only once, and implies they are unique syndrome vectors as shown in Table 4.1.

TABLE 4.1: Lookup table (LUT_1) for a $(7, 5)$ RS code of radius $t = 1$

Weight (w)	**Error vector**	**Syndrome** ($\vec{\chi_1}$)	**Syndrome**
$w \leq t$	$n^w \binom{n}{w}$	# possibilities	# occ.{**1**}
1	1 0 0 0 0 0 0	5 7	5 7
	2 0 0 0 0 0 0	1 5	1 5
	3 0 0 0 0 0 0	4 2	4 2
	4 0 0 0 0 0 0	2 1	2 1
	5 0 0 0 0 0 0	7 6	7 6
	6 0 0 0 0 0 0	3 4	3 4
	7 0 0 0 0 0 0	6 3	6 3
	0 1 0 0 0 0 0	7 3	7 3
	0 2 0 0 0 0 0	5 6	5 6
	0 3 0 0 0 0 0	2 5	2 5
	0 4 0 0 0 0 0	1 7	1 7
	0 5 0 0 0 0 0	6 4	6 4
	0 6 0 0 0 0 0	4 1	4 1
	0 7 0 0 0 0 0	3 2	3 2
	⋮	⋮	⋮
	0 0 0 0 0 1 0	2 4	2 4
	0 0 0 0 0 2 0	4 3	4 3
	0 0 0 0 0 3 0	6 7	6 7
	0 0 0 0 0 4 0	3 6	3 6
	0 0 0 0 0 5 0	1 2	1 2
	0 0 0 0 0 6 0	7 5	7 5
	0 0 0 0 0 7 0	5 1	5 1
	0 0 0 0 0 0 1	1 1	1 1
	0 0 0 0 0 0 2	2 2	2 2
	0 0 0 0 0 0 3	3 3	3 3
	0 0 0 0 0 0 4	4 4	4 4
	0 0 0 0 0 0 5	5 5	5 5
	0 0 0 0 0 0 6	6 6	6 6
	0 0 0 0 0 0 7	7 7	7 7
Total	**49**	**49**	**49**

The probability $'\rho'$ of having non-zero unique codewords in distance t of the decoder is a function of the number of possible syndrome vectors and unique syndrome vector occurrences in the lookup table:

$$\rho = \frac{\sum S(occ.)}{\sum S(poss.)} = \frac{49}{49} = 1.$$

4.1.2 $(7, 4)$ RS Code

The generator polynomial of a $(7, 4)$ RS code with symbols from $GF(2^3)$ has $(\alpha, \alpha^2, \alpha^3)$ as its consecutive roots. Assume the channel introduces one error $(w = 1)$ to the transmitted message. A lookup table LUT_1 is formed that contains columns of possible combinations of error vectors, possible syndromes $(\vec{\chi_1})$ that correspond to an error vector, and unique syndromes obtained from the frequency of syndrome occurrences in $(\vec{\chi_1})$. The search result shows there are 49 rows of possible error vectors derived from (3.1), which is mapped to 49 possible syndrome vectors of length 3. Also, the frequency of syndrome occurrences indicates all 49 syndromes in $(\vec{\chi_1})$ occurred once. Table 4.2 summarizes the RS $(7, 4)$ lookup table for radius $t = 1$ having 49 possible syndrome occurrences and 49 unique syndrome occurrences, which is similar to Table 4.1, and is used to compute the probability $'\rho'$ of obtaining unique codewords in the decoding radius t:

$$\rho = \frac{\sum S(occ.)}{\sum S(poss.)} = \frac{49}{49} = 1.$$

TABLE 4.2: Lookup table (LUT_1) for a $(7, 4)$ RS code of radius $t = 1$

Weight (w)	**Error vector**	**Syndrome**	**Syndrome**
$w \leq t$	$n^w \binom{n}{w}$	# possibilities	# occ.{**1**}
1	49	49	49
Total	**49**	**49**	**49**

4.1.3 $(7, 3)$ RS Code

Now a two-error-correcting $(7, 3)$ RS code generated by $(\alpha, \alpha^2, \alpha^3, \alpha^4)$ is considered. For a minimum distance decoder, the maximum error correcting capability $t = 2$, and all possible combinations of error vectors with their corresponding possible syndrome vectors $(\vec{\chi_2})$ are computed for $w \leq t$ errors. Suppose the channel adds one error to the transmitted message, there are 49 possible syndrome vectors in $\vec{\chi_1}$ as shown above for $(7, 4)$ and $(7, 5)$ RS codes. Assume further that the channel introduces two errors. From (3.1), there are an additional 1029 possible syndrome vectors $\vec{\chi_2}$, such that the lookup table now contains a total of 1078 possible syndromes as shown in Table 4.3. The frequency of syndrome occurrences is checked in the lookup table, which returns all 1078 possible syndromes occurring in one place; that is, all the syndromes are unique as shown in Table 4.3. The result of analysis is used to derive the probability of obtaining unique codewords in the given radius t as follows:

$$\rho = \frac{\sum S(occ.)}{\sum S(poss.)} = \frac{1078}{1078} = 1$$

TABLE 4.3: Lookup table (LUT_1) for a $(7, 3)$ RS code of radius $t = 2$

Weight (w)	**Error vector**	**Syndrome**	**Syndrome**
$w \leq t$	$n^w \binom{n}{w}$	# possibilities	# occ.{**1**}
1	49	49	49
2	1029	1029	1029
Total	**1078**	**1078**	**1078**

4.1.4 $(7, 2)$ RS Code

For the purpose of analysis, a two error-correcting $(7, 2)$ RS code that is formed by roots $\{\alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^5\}$ of a generator polynomial $g(x)$ is further considered. It is assumed that the channel introduces errors within the error correcting capability of the decoder, $w \leq t$. If $w = 1$ error is added, LUT_1 has 49 unique syndromes $\vec{\chi_1}$.

The unique syndromes are obtained by searching all possible syndrome vectors corresponding to rows of possible error vector combinations and selecting those syndromes that occurred in one place.

For $w = 2$ errors, the table has an additional 1029 possible error vectors with a one-to-one mapping of each error vector to possible syndrome vector rows $\vec{\chi_2}$. Since both $\vec{\chi_1}$ and $\vec{\chi_2}$ are derived from combination of errors, $\vec{\chi_1}$ may be repeated in $\vec{\chi_2}$ and some syndrome vectors in $\vec{\chi_2}$ may also occur more than once in the lookup table, therefore a check is done to confirm both possibilities. The search shows syndromes from $\vec{\chi_1}$ and $\vec{\chi_2}$ are completely independent of each other, that is, all 1078 syndromes occur once in the table. Table 4.4 shows the number of possible syndromes and unique syndrome occurrence which are used to compute the probability of decoding radius containing unique codewords. The probability is given as:

$$\rho = \frac{\sum S(occ.)}{\sum S(poss.)} = \frac{1078}{1078} = 1.$$

TABLE 4.4: Lookup table (LUT_1) for a $(7, 2)$ RS code of radius $t = 2$

Weight (w)	**Error vector**	**Syndrome**	**Syndrome**
$w \leq t$	$n^w \binom{n}{w}$	# possibilities	# occ.{**1**}
1	49	49	49
2	1029	1029	1029
Total	**1078**	**1078**	**1078**

4.1.5 Analysis

After examining the $(7, k)$ RS codes, where $k = 2, 3, \ldots, 5$ and the probability of having unique codewords is obtained, a relationship between the probabilities and message length k is shown by Figure 4.1. The probability of a Minimum Distance Decoder for designed distance t constantly gives one, which implies the number of syndrome occurrences is the same as the number of possible syndromes in the lookup

table and that the minimum distance decoder always have unique codewords in the decoder's sphere. Therefore, all error patterns $w \leq t$ are decoded correctly and the minimum distance decoding works for designed distance t.

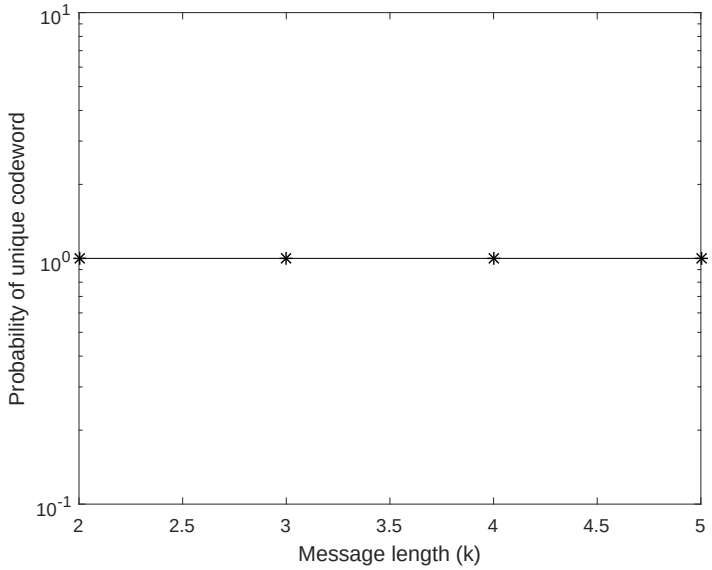

FIGURE 4.1: Probability of obtaining unique codewords for $(7, k)$ RS codes, $w \leq t$

The performance of minimum distance decoding for $(7, k)$ RS codes with different rates is analyzed and the result is shown in Figure 4.2. The result shows both $(7, 5)$ and $(7, 4)$ RS codes performing the same way, while $(7, 3)$ and $(7, 2)$ RS codes also exhibit similar performance. However, there is a 2dB code gain between $(7, 5)$, $(7, 3)$ and $(7, 4)$, $(7, 2)$ RS codes respectively. This implies low rate RS codes performs better than high rate codes with a trade off in the amount of energy used in transmitting the messages.

The next scenario examines $w \leq t + 1$ for RS$(7, k)$ codes where lookup table LUT_2 is computed and probabilities of having unique codewords in a radius $t + 1$ is derived.

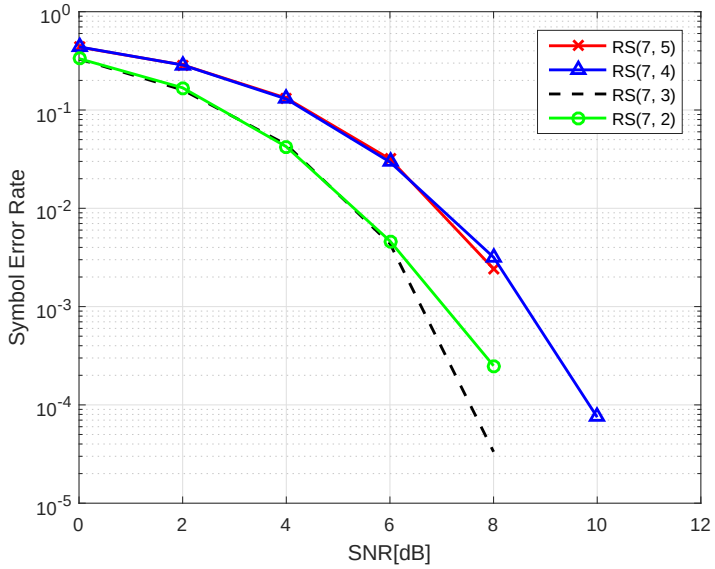

FIGURE 4.2: Performance of the MDD for $(7, k)$ RS codes

4.2 Bounded Distance Decoding for $w \leq t + 1$

4.2.1 $(7, 5)$ RS Code

Consider a $(7, 5)$ RS code with symbols from $GF(2^3)$ having (α, α^2) as roots of the generator polynomial. It is assumed that the bounded distance decoder can correct more errors within the decoding radius $t + 1$. An analysis is performed to determine a probability of having unique codewords within this radius. Suppose the channel introduces $w = 1$ error patterns, the lookup table LUT_2 contains 49 rows of possible error combinations with each row mapped to exactly 49 possible syndrome vector of length 2, in the column of possible syndrome vectors $(\vec{\chi_1})$. The frequency of syndrome occurrence is then checked to determine the number of syndrome occurrences in $(\vec{\chi_1})$, which yields all 49 syndromes in $(\vec{\chi_1})$ occurring once, and are therefore unique syndromes.

Furthermore, if the channel adds $w = 2$ error patterns, 1029 possible error combinations are obtained by (3.1), and their corresponding possible syndrome vectors $(\vec{\chi_2})$ are derived. Since the syndromes are possible combinations, there may be one or

more syndrome of $\vec{\chi_2}$ that are repeated, therefore, the frequency of syndrome occurrence is checked. Table 4.5 shows the result obtained from finding the frequency of syndrome occurrences of $\vec{\chi_2}$ where 14 syndromes are repeated 21 times. However, the

TABLE 4.5: Number of syndrome occurrences in $(\vec{\chi_2})$ for $w = 2$

Weight (w)	**Error vector**	**Syndrome**	**Syndrome**
$w \geq t$	$n^w \binom{n}{w}$	# possibilities	# occ.{**1** 15 21}
2	1029	1029	0 49 14
Total	**1029**	**1029**	**0**

result in Table 4.5 is not sufficient to conclude that syndromes in the lookup table are either unique or not. This is because LUT_2 now contains a total of 1078 possible syndromes from $\vec{\chi_1}$ and $\vec{\chi_2}$, and some or all 49 syndromes from $\vec{\chi_1}$ may be repeated in $\vec{\chi_2}$. Table 4.6 shows the frequency of syndrome occurrences and number of unique syndromes in the lookup table where all 49 syndromes from $\vec{\chi_1}$ are repeated 15 times in $\vec{\chi_2}$; that is, $\vec{\chi_1} \cap \vec{\chi_2} \neq \emptyset$, and thus, it can be concluded that syndromes in the lookup table are not unique.

TABLE 4.6: Lookup table (LUT_2) for a $(7, 5)$ RS code of radius $t + 1$

Weight (w)	**Error vector**	**Syndrome**	**Syndrome**
$w \geq t$	$n^w \binom{n}{w}$	# possibilities	# occ.{**1** 15 21}
1	49	49	0
2	1029	1029	0 49 14
Total	**1078**	**1078**	**0**

The probability of having unique codewords in the given radius $t+1$ is obtained from LUT_2 by finding the ratio of total syndrome occurrence to syndrome possibilities as follows:

$$\rho = \frac{\sum S(occ.)}{\sum S(poss.)} = \frac{0}{1078} = 0.$$

Next is to compute a lookup table LUT_2 for a $(7, 4)$ RS code whereby all syndrome vectors in the table is verified for uniqueness. The result determines a probability of having unique syndromes.

4.2.2 $(7, 4)$ RS Code

The generator polynomial $g(x)$ of a $(7, 4)$ RS code has $(\alpha, \alpha^2, \alpha^3)$ as all its roots. It is assumed $w \leq t + 1$ errors can be added to the transmitted messages and suppose $w = 1$ error patterns are introduced, the lookup table LUT_2 contains 49 rows of possible error combinations with each row corresponding to 49 possible syndrome vectors $\vec{\chi_1}$ of length 3. The syndromes in $\vec{\chi_1}$ are checked for uniqueness, which yields all 49 syndromes occurring once in LUT_2.

Similarly, based on the assumption that a bounded distance decoder may correct up to $t + 1$ errors, let $w = 2$ be the number of introduced error patterns. The lookup table, LUT_2 increases by adding 1029 possible error combinations with corresponding rows of syndrome vectors $\vec{\chi_2}$. Since the syndromes are possible combinations, some syndromes of $\vec{\chi_1}$ may be repeated in $\vec{\chi_2}$, while one or more syndromes of $\vec{\chi_2}$ may also occur more than once. Therefore, all 1029 syndromes of $\vec{\chi_2}$ are first checked for uniqueness, which gives 294 and 147 syndromes occurring twice and three times respectively, and no syndrome occurs once as shown in Table 4.7.

TABLE 4.7: Number of syndrome occurrences in $(\vec{\chi_2})$ for $w = 2$

Weight (w)	**Error vector**	**Syndrome**	**Syndrome**
$w \geq t$	$n^w \binom{n}{w}$	# possibilities	# occ.{1 2 3}
2	1029	1029	0 294 147
Total	**1029**	**1029**	**0**

The result from Tables 4.5 and 4.7 is important to compare the probability of decoding failure for $(7, 5)$ and $(7, 4)$ RS codes. Considering the frequency of syndrome occurrences in both tables, it can be inferred that the BDD has more probability of

decoding failure for a $(7, 5)$ RS code than a $(7, 4)$ RS code. This is because a $(7, 5)$ code has a minimum of 15 and maximum of 21 number of syndrome occurrences in contrast to a $(7, 4)$ code that contains a minimum of 2 and maximum of 3 number of syndrome occurrences.

A further check is done to identify the number of times syndromes of $\vec{\chi_1}$ would intersect with those of $\vec{\chi_2}$ in the lookup table. This search produces $\vec{\chi_1} \cap \vec{\chi_2} = \emptyset$; that is, all 49 syndromes of $\vec{\chi_1}$ occur once, and are therefore unique in the lookup table. The result is summarized in Table 4.8 which contains a total of 1078 possible syndrome vectors whereby 49 syndromes and the remaining 1029 syndromes are non-unique. The probability of having unique codewords for a $(7, 4)$ RS code is obtained by taking a fraction of the unique syndromes and possible syndromes in the table as follows:

$$\rho = \frac{\sum S(occ.)}{\sum S(poss.)} = \frac{49}{1078} \approx 0.05.$$

TABLE 4.8: Lookup table (LUT_2) for a $(7, 4)$ RS code of radius $t + 1$

Weight (w)	**Error vector**	**Syndrome**	**Syndrome**
$w \geq t$	$n^w \binom{n}{w}$	# possibilities	# occ.{**1** 2 3}
1	49	49	49
2	1029	1029	0 294 147
Total	**1078**	**1078**	**49**

4.2.3 $(7, 3)$ RS Code

Furthermore, consider a two-error-correcting $(7, 3)$ RS code over $GF(2^3)$ whose generator polynomial has consecutive roots $(\alpha, \alpha^2, \alpha^3, \alpha^4)$. Based on the assumption that the Bounded Distance Decoder can correct errors up to distance $t + 1$, the lookup table LUT_2 is computed and used to obtain the probability of having unique codewords within the decoder's radius. For error patterns of $w = 1$, the lookup table

LUT_2 contains 49 rows of possible error combinations and corresponding rows of possible syndrome vectors $\vec{\chi_1}$ of length 4. The number of syndrome occurrences is examined, which yields all 49 syndromes occurring once.

Let $w = 2$, then the numbers of possible error vectors and corresponding syndrome vectors in LUT_2 increases. From (3.1), there are 1029 possible combinations of error vectors mapped to corresponding syndrome vector rows $\vec{\chi_2}$. Hence, the total number of syndrome rows to be examined for uniqueness in LUT_2 increases to 1078. The frequency of syndrome occurrences shows all 1029 syndromes of $\vec{\chi_2}$ occurring without repetition, and are also independent of syndromes from $\vec{\chi_1}$ in the lookup table. Thus, all 1078 syndromes in LUT_2 are unique.

Now, suppose the channel introduces $w = 3$ errors, there are 12005 possible error combinations, which has a one-to-one mapping with 12005 syndrome vector rows $\vec{\chi_3}$. The lookup table further increases to contain 13083 possible syndromes that must be checked for uniqueness. Since the syndromes are obtained by combinations, it is possible that some or all of the syndromes of $\vec{\chi_1}$ and $\vec{\chi_2}$ are contained in $\vec{\chi_3}$, while rows of $\vec{\chi_3}$ may also be repeated.

Table 4.9 indicates the number of times each syndrome row in $\vec{\chi_3}$ are repeated. The frequency of syndrome occurrence shows 588, 833, 931, 1470, 196, and 14 syndromes occurs once, twice, three, four, five, and seven times respectively. From this result, any algorithm that decodes beyond half the minimum distance will perform better for $(7, 3)$ RS codes than $(7, 4)$ and $(7, 5)$ RS codes because RS$(7, 3)$ contains 588 syndromes that occur once, while both $(7, 4)$ and $(7, 5)$ RS codes do not. However,

TABLE 4.9: Number of Syndrome Occurrences ($\vec{\chi_3}$) for $w = 3$

w	**Error vector** $n^w \binom{n}{w}$	**Synd.** # possibilities	**Synd.** # occ. {**1 2 3 4 5 7**}
3	12005	12005	588 833 931 1470 196 14

the 588 syndromes of $\vec{\chi_3}$ that occurred once may not be unique in the lookup table.

This is because all the syndromes of $\vec{\chi_1}$, $\vec{\chi_2}$, and $\vec{\chi_3}$ must be disjoint of each other in LUT_2. Therefore, the frequency of syndrome occurrences is checked in the lookup table, which gives all 49 syndromes of $\vec{\chi_1}$ occurring once; that is, $\vec{\chi_1} \cap \vec{\chi_2} = \emptyset$ and $\vec{\chi_1} \cap \vec{\chi_3} = \emptyset$. It also shows that the 588 and 441 syndromes of $\vec{\chi_2}$ are repeated twice and three times as syndromes of $\vec{\chi_3}$ and are therefore non-unique syndromes.

Table 4.10 summarizes the result of finding the frequency of syndrome occurrences in disjoint vectors $\vec{\chi_1}$, $\vec{\chi_2}$ and $\vec{\chi_3}$. The result indicates there are 49 unique syndromes in LUT_2. The probability of having unique codewords is calculated from the result as follows:

$$\rho = \frac{49}{13083} \approx 0.004.$$

TABLE 4.10: Lookup table (LUT_2) for a $(7, 3)$ RS code of radius $t + 1$

w	**Synd.** # possibilities	**Synd.** # occ. {**1 2 3 4 5 6 7**}
1	49	49 0 0 0 0 0 0
2	1029	0 588 441 0 0 0 0
3	12005	0 392 931 1470 196 0 14
Total	**13083**	**49**

Next, the probability of having unique codewords is calculated for a $(7, 2)$ RS code within decoding radius $t + 1$.

4.2.4 $(7, 2)$ RS Code

Again, consider a two-error-correcting $(7, 2)$ RS code over $GF(2^3)$ generated by consecutive roots $(\alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^5)$ of a generator polynomial $g(x)$. The same assumption holds that the Bounded Distance Decoder corrects errors up to distance $t + 1$. We also compute a lookup table LUT_2 which is used to determine the probability of having unique codewords within the decoder radius. Suppose all $w = 1$ error patterns are introduced by the channel, there are 49 rows of possible error combinations, each

having corresponding rows of possible syndromes $\vec{\chi_1}$ in LUT_2. These syndromes are searched for uniqueness in the table. The search results in all 49 syndromes occurring once, which implies LUT_2 contains unique syndromes.

Assume all $w = 2$ error patterns are added, the lookup table now consists of an additional 1029 possible error vectors, each mapped to a row of 1029 syndrome vectors $\vec{\chi_2}$. LUT_2 now has a total of 1078 possible syndromes that must be checked for uniqueness. The result gives each syndrome in $\vec{\chi_2}$ occurring once, and none of the syndromes from $\vec{\chi_1}$ appear as a syndrome of $\vec{\chi_2}$; that is, $\vec{\chi_1} \cap \vec{\chi_2} = \emptyset$. Hence, all syndromes in LUT_2 are unique.

Let $w = 3$, the number of possible syndromes increases by combining more error patterns in LUT_2. From (3.1), there are 12005 possible combinations which is mapped to equivalent syndrome vectors $\vec{\chi_3}$ of length 5, which means LUT_2 contains a total of 13083 possible syndromes that must be examined for uniqueness. Since the syndromes are obtained from possible combinations, some or all syndromes of $\vec{\chi_1}$ and $\vec{\chi_2}$ may re-occur as syndromes of $\vec{\chi_3}$, while one or more syndromes of $\vec{\chi_3}$ may be repeated more than once. A search for the number of occurrences of each syndrome of $\vec{\chi_3}$ indicates 11025 and 980 syndromes occur once and twice respectively as shown in Table 4.11. The result in Tables 4.11 and 4.9 is important to analyze the probability

TABLE 4.11: Number of Syndrome Occurrences ($\vec{\chi_3}$) for $w = 3$

w	**Error vector** $n^w(n$ choose $w)$	**Synd.** # possibilities	**Synd.** # occ. {**1** **2**}
3	12005	12005	11025 980

of decoding failure for both $(7, 3)$ and $(7, 2)$ RS codes. A comparison between the number of syndrome occurrences for $(7, 3)$ and $(7, 2)$ RS codes obviously show more unique syndromes and less syndrome repetitions for a $(7, 2)$ than a $(7, 3)$ RS code. It can then be inferred that the BDD will perform better for a $(7, 2)$ RS code compared to a $(7, 3)$ RS code.

Recall, for the lookup table to contain unique syndromes, $\vec{\chi_1}, \ldots, \vec{\chi_3}$ must be disjoint of each other. Therefore, all the syndromes in LUT_2 must be verified against each other to avoid repetitions. A search for syndromes of $\vec{\chi_1}$ in $\vec{\chi_2}$ and $\vec{\chi_3}$ yield all 49 syndromes of $\vec{\chi_1}$ occur once; that is, $\vec{\chi_1} \cap \vec{\chi_2} = \emptyset$ and $\vec{\chi_1} \cap \vec{\chi_3} = \emptyset$. Also, verifying $\vec{\chi_2}$ against $\vec{\chi_3}$ returns all 1029 syndromes of $\vec{\chi_2}$ occur only one time; that is, $\vec{\chi_1} \cap \vec{\chi_2} = \emptyset$ and $\vec{\chi_2} \cap \vec{\chi_3} = \emptyset$. Table 4.12 summarizes the result, which shows 49, 1029 and 11025 syndromes of $\vec{\chi_1}$, $\vec{\chi_2}$ and $\vec{\chi_3}$ are unique in the lookup table. The probability of having unique codewords in decoding radius $t+1$ is obtained as

$$\rho = \frac{12103}{13083} \approx 0.93.$$

TABLE 4.12: Lookup table (LUT_2) for a $(7, 2)$ RS code of radius $t+1$

w	**Synd.** # possibilities	**Synd.** # occ. {**1** **2**}
1	49	49 0
2	1029	1029 0
3	12005	11025 980
Total	**13083**	**12103**

4.2.5 Analysis

A $(7, 5)$ RS code has a probability of zero, which means none of the codewords are unique within decoding radius $t+1$ as compared to a $(7, 4)$ RS code of the same number of possible codewords. Also the probability of having unique codewords for a $(7, 4)$ RS code is relatively higher compared to a $(7, 3)$ RS code. This is because a $(7, 3)$ RS code contains more possible codewords in decoding radius $t+1$ than a $(7, 4)$ RS code. However, both RS codes may have the same performance for any algorithm that decodes beyond half the minimum distance. For a $(7, 2)$ RS code, the probability is close to one (0.93), which indicates there are many more (12103) codewords that uniquely occurred in the decoding radius $t+1$ as against 49 unique codewords for a $(7, 3)$ RS code. Figure 4.3 shows a relationship between the probability of obtaining

unique codewords in radius $t+1$ of a bounded distance decoder and dimension k of $(7,k)$ RS codes, while Table 4.13 shows the maximum error correcting capability and the probability of having unique codewords for the MDD and BDD.

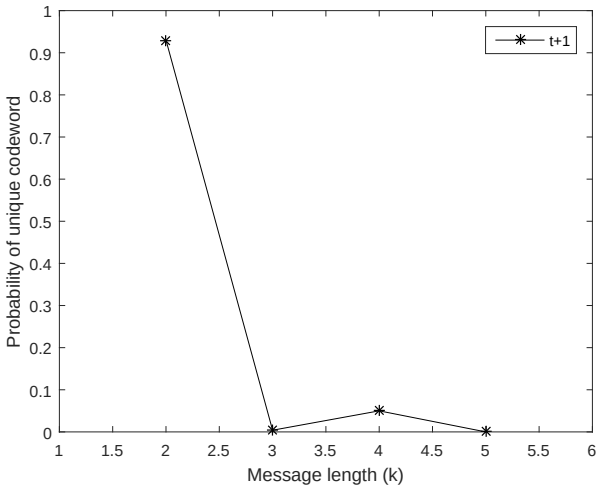

FIGURE 4.3: Probability of obtaining unique codewords for $(7,k)$ RS codes, $w \leq t+1$

TABLE 4.13: Comparison between MDD and BDD for low rate $(7,2)$ RS Codes

	MDD $(w \leq t)$	**BDD** $(w \leq t+1)$
w	2	3
ρ	1	0.93

This research has shown that for low code rates, the bounded distance decoder is capable of correcting $t+1$ error patterns with high probability. To verify how the bounded distance decoder performs for low code rates ($R \leq 1/3$), a $(7,2)$ RS code is tested using both the minimum distance decoder that corrects t errors and the bounded distance decoder that corrects $t+1$ errors. For the simulation, BPSK modulation is used in an AWGN channel. The result in Figure 4.4 indicates that however marginal, the bounded distance decoder does indeed perform better than the minimum distance decoder at low SNR (SNR $\leq$ 6dB), while at an increased SNR (SNR $>$ 6dB) it gives the same performance for both minimum distance and bounded distance decoders.

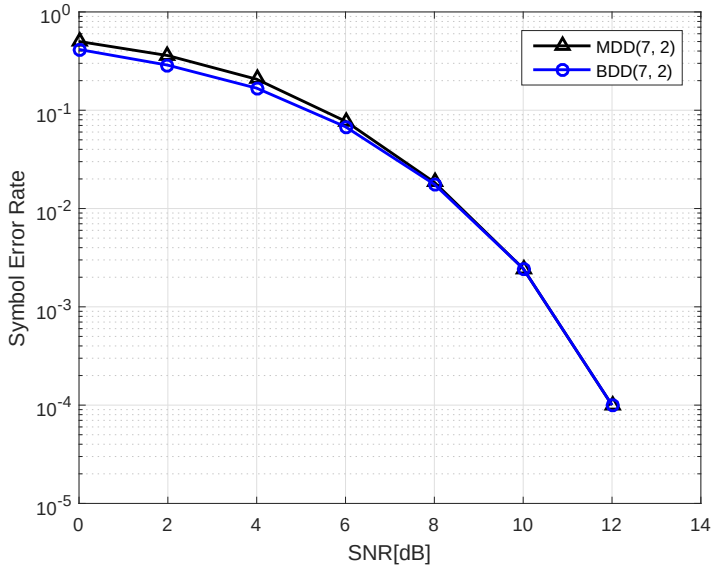

FIGURE 4.4: Performance of the MDD and BDD for (7,2) RS codes

CHAPTER 5

Conclusion and Future Work

5.1 Conclusion

The study computes lookup tables for both minimum distance decoding and bounded distance decoding with radius t and $t+1$ respectively. The table is computed by obtaining all possible syndrome vectors, and examining the possibilities of each syndrome to occur only one time. It also derived the probability of obtaining unique codewords by calculating ratios of unique syndromes occurrences to the total of possible syndromes in the tables. The case of $(7, k)$ RS codes with decoding radius t, where the error value (w) is less than or equal to t has been analyzed. All the syndromes occurred uniquely in the lookup table, and the probabilities obtained for all code rates equal one. This is the case of minimum distance decoding where Peterson-Gorenstein-Zieler, Berlekamp-Massey, or the Extended Euclidean Algorithm successfully decode all error patterns up to the designed radius t.

The analysis is also done for the case where the error value (w) is less than or equal to $t+1$; that is, a BDD of radius $t+1$, which is the aim of this research. The probability obtained for a $(7, 5)$ RS code indicates BDD of increased radius will always fail because it does not have any unique codewords. Similarly, a $(7, 4)$ RS code has a very low chance of correctly decoding error patterns containing values greater than t, due to the low probability of having unique codewords in the radius $t+1$. For a $(7, 3)$ RS code, the probability of obtaining unique codewords tends to zero, therefore the BDD fails, while the probability obtained for a $(7, 2)$ RS code closely approach one, which implies the decoder will correct error patterns of error

values greater than t, with very low probability of decoding failure. From the analysis in this work and simulation results, it can be concluded that the bounded distance decoder for low rate RS codes is capable of correcting error values greater than t, and specifically $t + 1$.

5.2 Future Work

Further work can be done on RS code construction for bounded distance decoding of radius $t + 1$, since $(7, 4)$ and $(7, 2)$ RS codes were shown to have better performance than $(7, 5)$ and $(7, 3)$ RS codes respectively. Also, more work can be done on improving the coding gain for bounded distance decoding. Combining soft-decision data with the list of all possible codewords might improve decoding performance.

References

[1] T. K. Moon. *Error Correction Coding - Mathematical Methods and Algorithms*. New York: John Wiley & Sons, 2005.

[2] S. Lin and D. J. Costello. *Error Control Coding: Fundamentals and Applications*. Saddle River, NJ 07458: Pearson Education Inc., Pearson Prentice Hall, 2004.

[3] D. Gorenstein and N. Zierler. "A Class of Error-Correcting Codes in p^m Symbols." In *Journal of the Society for Industrial and Applied Mathematics*, vol. 9, pp. 207–214. 1961.

[4] R. T. Chien. "Cyclic Decoding Procedures for Bose-Chaudhuri-Hocquenghem Codes." In *IEEE Transactions on Information Theory*, vol. IT-10, pp. 549–557. 1964.

[5] D. G. Forney. "On Decoding BCH Codes." In *IEEE Transaction on Information Theory*, vol. 2, pp. 549–557. 1965.

[6] E. R. Berlekamp. *Algebraic Coding Theory: Revised Edition*. Aegean Park Press, 1984.

[7] Y. Sugiyama, M. Kasahara, S. Hirasawa, and T. Namekawa. "A method for solving key equation for decoding Goppa codes." In *Information and Control*, vol. 27, pp. 87–99. 1975.

[8] Q. Cheng and D. Wan. "On the List and Bounded Distance Decodability of Reed Solomon Codes." *Society for Industrial and Applied Mathematics*, vol. 37, no. 1, pp. 195–209, 2007.

[9] M. Sudan. "Decoding of Reed Solomon Codes beyond the Error-Correction Bound." In *Journal of Complexity*, vol. 13, pp. 180–193. 1997.

[10] G. Schmidt, V. R. Sidorenko, and M. Bossert. "Syndrome Decoding of Reed-Solomon Codes Beyond half the Minimum Distance Based on Shift-Register Synthesis." In *IEEE Transaction on Information Theory*, vol. 56, pp. 5245–5252. 2010.

[11] R. E. Blahut. "Transform Techniques for Error Control Codes." In *IBM Journal of Research and Development*, vol. 23, pp. 299–315. 1979.

[12] H. M. Anwarul, V. K. Bhargava, and L.-N. Tho. *Algorithms and Architectures for the Design of a VLSI Reed-Solomon Codec Algorithms and Architectures for the Design of a VLSI Reed-Solomon Codec*, chap. 5, pp. 60–107. IEEE Press, 1994.

[13] M. Bossert. *Channel Coding for Telecommunications*. John Wiley and Sons, LTD, 1999.

[14] W. W. Peterson. "Encoding and Error-Correction Procedures for the Bose-Chaudhuri Codes." *IRE Transactions on Information Theory*, vol. 6, no. 4, pp. 459–470, September 1960.

[15] J. L. Massey. "Shift-Register Synthesis and BCH Decoding." In *IEEE Transactions on Information Theory*, vol. 15, pp. 122–127. 1969.

[16] V. Guruswami and M. Sudan. "Improved Decoding of Reed Solomon and Algebraic-Geometry Codes." *IEEE Transaction on Information Theory*, vol. 45, no. 6, pp. 1757–1767, September 1999.

[17] Y. Wu. "New List Decoding Algorithms for Reed Solomon and BCH Codes." In *IEEE Transactions on Information Theory*, vol. 54, pp. 3611–3630. 2008.

[18] G. Schmidt, V. R. Sidorenko, and M. Bossert. "Collaborative Decoding of Interleaved Reed Solomon Codes and Concatenated Code Designs." In *IEEE Transactions on Information Theory*, vol. 55, pp. 2991–3012. 2009.

[19] G.-L. Feng and K. K. Tzeng. "A generalized Euclidean algorithm for multisequence shift-register synthesis." In *IEEE Transaction Information Theory*, vol. 35, pp. 584–594. 1989.

[20] G.-L. Feng and K. K. Tzeng. "A Generalization of the Berlekamp-Massey Algorithm for Multisequence Shift-Register Synthesis with Applications to Decoding Cyclic Codes." *IEEE Transaction on Information Theory*, vol. 37, no. 5, pp. 1274–1287, September 1991.

[21] G. Schmidt and V. R. Sidorenko. "Multi-Sequence Linear Shift-Register Synthesis: The Varying Length Case." In *IEEE International Symposium Information Theory*, pp. 1738–1742. Seattle, USA, July 2006.

[22] G. Schmidt and V. R. Sidorenko. "Linear Shift-Register Synthesis for Multiple Sequences of Varying Length." *Preprint, available online at arXiv:cs/0605044 [cs.IT]*, 2006.

[23] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error-Correcting Codes*, vol. 16. North-Holland Publishing Company Amsterdam. New York. Oxford, 3rd ed., 1981.