Encoding of a DFT (Reed-Solomon like) code with consecutive time and frequency domain redundancy

Author: Ina Kodrasi
Supervisor: Prof. Dr. Werner Henkel
School of Engineering and Science
Jacobs University Bremen

Fall Semester 2008
Abstract

This report studies the encoding of Reed-Solomon codes with consecutive time and DFT domain redundancy. Many data transmission schemes rely on Reed-Solomon codes for error correction, therefore a thorough comprehension of their properties would be very constructive.

An arbitrary number of redundant zeros in both time and DFT domain codewords were considered for all the encoding methods. In order to enforce the respective number of zeros, the equations that the information and codeword symbols should satisfy were determined.

We concluded that the time domain encoding methods, are the straightforward methods for controlling the redundancy of the codeword in both domains. The time domain redundancy directly depends on the number of information symbols whereas the DFT domain redundancy depends on the degree of the generator polynomial.

However, controlling the codeword redundancy when using the DFT domain encoding method is not as simple. Enforcing an arbitrary number of zeros in the codeword in time domain requires a solution of a system of equations.

Since the error-correction capability of Reed-Solomon codes depends on the codeword redundancy, further standard investigations of the error correction patterns for time/DFT Reed-Solomon codes could be a next step in our analysis.
# Contents

1 Introduction 3  
2 Basic definitions 5  
3 Encoding in DFT domain 9  
4 Non-systematic encoding based on the generator polynomial 13  
5 Systematic encoding based on the generator polynomial 16  
6 Systematic encoding based on the parity-check polynomial 18  
7 Conclusion 20
Chapter 1

Introduction

Communication channels are very often corrupted by noise or various other distortions such as interference, fading or dispersion. In this context, it is crucial for a communication system to transmit data with very high reliability. Most modern communication systems aim at increasing performance through the usage of error control codes. An error control code is an algorithm for preprocessing the transmission data such that errors which are introduced during transmission can be detected and corrected. The most used error control codes are cyclic ones [1].

In this report, we will focus on a class of cyclic codes called Reed-Solomon (RS) codes. These codes were initially described by Irving Reed and Gustave Solomon in ”Polynomial Codes over Certain Finite Fields” [2]. In the period since their discovery, RS codes have been used in a wide variety of commercial applications such as for error correction in digital audio discs, in deep space telecommunication systems as in several of NASA and ESA’s planetary exploration missions, in systems with feedback which allow the transmission of information from the receiver back to the transmitter etc. They occupy a very extensive place in the theory and practice of error correction codes due to the existence of good coding and decoding techniques as well as their well-understood distance structure [1]. The thorough comprehension of the properties of Reed-Solomon codes with consecutive redundancy in time and DFT domain is crucial given their wide variety of applications.

The error-correction capability of RS codes is determined by the measure of redundancy in the block. If the locations of the errors are not known,
a Reed-Solomon code can correct up to half the number of the redundant symbols.

This report has been written in an attempt to capture the properties of Reed-Solomon codes with consecutive redundancy in time and DFT domain. The definition of RS codes together with an overview of their general structure will be presented in Chapter 2. Reed-Solomon codes are non-binary codes built using symbols from finite fields; an outline of the composition of a Galois field is also presented in Chapter 2. This chapter provides the reader with the necessary background for the redundancy analysis in chapters 3, 4, 5 and 6, where four different encoding methods are taken into account. These encoding methods are: encoding in DFT domain; non-systematic encoding based on the generator polynomial; systematic encoding based on the generator polynomial; systematic encoding based on the parity polynomial. A review of the Discrete Fourier Transform and inverse Discrete Fourier Transform will be provided for an accurate understanding of the DFT encoding method. Further, the analyses of the other encoding methods require the definition of the generator polynomial and the parity-check polynomial, which will be given in chapters 2 and 4, respectively. The encoding methods will be illustrated by examples. The results of the redundancy analyses are summarized in the final Chapter 8.
Chapter 2

Basic definitions

A communication system transmits data from the source to the sink through a channel. When using an error correction code, the information vector \( \mathbf{i} \) from the data source is preprocessed into a sequence of symbols \( \mathbf{c} = (c_0 c_1 \ldots c_{N-1}) \) called a codeword. This is done by the encoder as shown in the block diagram 2.1. Further, the block diagram shows that the codeword \( \mathbf{c} \) is transmitted through a channel which might be corrupted by noise. As a result, the received word \( \mathbf{r} \) may well not be the same as the transmitted one. The standard decoder then attempts to find the codeword \( \mathbf{c} \) that is closest to \( \mathbf{r} \). Formally “closeness” is defined through the Hamming distance concept.

**Definition 2.1**: The Hamming distance \( d_H(\mathbf{c}, \mathbf{r}) \) between two sequences \( \mathbf{c} \) and \( \mathbf{r} \) is defined as the number of places in which they differ [1].

The standard decoding algorithm finds the codeword or codewords \( \mathbf{c} \) such that \( d_H(\mathbf{c}, \mathbf{r}) \) is minimized. If there is one such codeword, it is decoded to get the estimated transmitted information \( \hat{i} \). Otherwise, an uncorrectable error is detected.

However, there are difficulties inherent in this communication scheme when considering arbitrarily large codes. The error correction capability of a code is limited by how close together any two error-free blocks are. Therefore, the quantity to examine is the minimum Hamming distance. Finding the minimum Hamming distance of an arbitrary code and also using the standard decoding procedure generally means comparing every possible pair of codewords, which becomes rather exhaustive. Further, encoding through arbitrary codes implies keeping a list of all possible message words and re-
To overcome all these difficulties, algebraic codes such as Reed-Solomon codes were defined in the 60s.

**Definition 2.2.1:** A Reed-Solomon code of length $N$ and minimum Hamming distance $d_{H_m}$ is the set of vectors, whose components are the values of a polynomial $C(x)$ of degree $\leq K - 1 = N - d_{H_m}$ at positions $z^j$, with $z$ being an element of order $N$ from an arbitrary number field. $N$ refers to the codeword length whereas $K$ refers to the number of information symbols [4].

$$c = (c_0 \ c_1 \ \cdots \ c_{N-1}) \quad c_j = C(x = z^j)$$

In the definition of RS codes, the minimum Hamming distance is given as $d_{H_m} = N - K + 1$. Therefore, using Reed-Solomon codes provides an advantage over using arbitrary codes because in the former the minimum Hamming distance is known. Further, Reed-Solomon codes are linear codes, which is a guarantee for the existence of efficient means of encoding. The problem of impractical decoding is also solved through RS codes since there exist fairly
simple algorithms for detecting and correcting errors [3]. Since RS codes are built using symbols from finite fields, it is important to endeavor into the area of Galois fields. For any prime number \( p \), there exists a finite Galois field denoted by GF(\( p \)) that contains exactly \( p \) elements. The integers 0, 1, \( \cdots \), \( p-1 \) form the field GF(\( p \)) under modulo-\( p \) addition and multiplication. Every finite field contains at least one primitive element \( z \), the powers of which generate all the elements of the Galois field. Example 2.2 illustrates the construction of RS codes with symbols from GF(5).

**Example 2.2:** Let us consider the Galois field GF(5) with primitive element \( z = 3 \). The elements of this field are generated by the powers of the primitive element: \( 3^0 \mod 5 = 1 \), \( 3^1 \mod 5 = 3 \), \( 3^2 \mod 5 = 4 \), \( 3^3 \mod 5 = 2 \), therefore GF(5) = \{0, 1, 2, 3, 4\}. Using these elements we specify an RS code of length \( N = 4 \). In the rest of this report the modulo operation will be omitted since we will be dealing only with modular arithmetic. Let the information vector be \( \mathbf{i} = (2 \ 3) \). Describing the information vector as a polynomial, the degree-limited information polynomial has only two components \( C_0 \) and \( C_1 \), and is given by \( C(x) = C_0 + C_1 x = 2 + 3x \). Following the definition of RS codes, the codeword for this information vector is found to be

\[
\begin{align*}
c_0 &= C(x = z^0 = 1) = 2 + 3 \cdot 1 = 0 \\
c_1 &= C(x = z^1 = 3) = 2 + 3 \cdot 3 = 1 \\
c_2 &= C(x = z^2 = 4) = 2 + 3 \cdot 4 = 4 \\
c_3 &= C(x = z^3 = 2) = 2 + 3 \cdot 2 = 3.
\end{align*}
\]

Example 2.2 illustrates the DFT domain encoding method for Reed-Solomon codes. As the name suggests, the DFT domain encoding method requires the definition of the Discrete Fourier Transform (DFT) and its inverse transformation (IDFT).

**Definition 2.2.2:** The sequence \( (c_0 \ c_1 \ \cdots \ c_{N-1}) \) is transformed into the sequence \( (C_0 \ C_1 \ \cdots \ C_{N-1}) \) by the DFT according to the formula

\[
C_k = \frac{1}{N} \sum_{j=0}^{N-1} c_j z^{-jk} \quad k = 0, \ 1, \ \cdots, \ N-1. \quad (2.1)
\]
Expressing this relation in vector notation, we have

\[
(C_0 \ C_1 \ \cdots \ C_{N-1}) = \frac{1}{N} (c_0 \ c_1 \ \cdots \ c_{N-1}) \left( \begin{array}{cccc}
1 & 1 & \cdots & 1 \\
1 & z^{-1} & \cdots & z^{-(N-1)} \\
1 & z^{-2} & \cdots & z^{-2(N-1)} \\
\vdots & \vdots & \ddots & \vdots \\
1 & z^{-(N-1)} & \cdots & z^{-(N-1)(N-1)} \\
\end{array} \right).
\]

Definition 2.2.3: The sequence \((C_0 \ C_1 \ \cdots \ C_{N-1})\) is transformed into the sequence \((c_0 \ c_1 \ \cdots \ c_{N-1})\) by the IDFT according to the formula

\[
c_j = \sum_{k=0}^{N-1} C_k z^{jk} \quad j = 0, 1, \cdots, N-1
\]

which in vector notation is given by

\[
(c_0 \ c_1 \ \cdots \ c_{N-1}) = (C_0 \ C_1 \ \cdots \ C_{N-1}) \left( \begin{array}{cccc}
1 & 1 & \cdots & 1 \\
1 & z & \cdots & z^{N-1} \\
1 & z^2 & \cdots & z^{2(N-1)} \\
\vdots & \vdots & \ddots & \vdots \\
1 & z^{N-1} & \cdots & z^{(N-1)(N-1)} \\
\end{array} \right),
\]

where the given matrix is the so called DFT Vandermonde matrix \(F_N\).

In the definitions above, \(z\) is a primitive \(N\)th root of unity such that \(z^N = 1\) and \(z^k \neq 1\) for \(k = 1, 2, \cdots, N-1\). In analogy to it, let us recall that the primitive element \(z\) of the Galois field satisfies \(z^N = 1\) and \(z^k \neq 1\) for \(k = 1, 2, \cdots, N-1\). Further, defining RS codes by means of a polynomial at position \(z^j\) with \(z\) being an element of order \(N\) resembles the IDFT transformation of a vector. Therefore the information vector can be put into a certain number of subsequent positions in DFT domain and the multiplication of the vector with the Vandermonde DFT matrix yields the corresponding time domain codeword.
Chapter 3

Encoding in DFT domain

When using the DFT domain encoding method, the uncertainty principle for the Fourier Transform of cyclic groups of prime order \( p \) gives some insight on the redundancy of the DFT and time domain vectors. For the time domain vector \( c \) and corresponding DFT domain vector \( C \) with symbols from \( \text{GF}(p) \), the uncertainty principle establishes

\[
\|c\|_0 + \|C\|_0 \geq p + 1,
\]

where \( \|x\|_0 \) refers to the support of \( x \) [5]. The uncertainty principle confirms that the sum of information symbols and the number of nonzero codeword symbols should be greater than or equal to \( p + 1 \).

For a generalization of the properties of RS codes with consecutive redundancy in time and DFT domain, we consider the Galois field \( \text{GF}(p) \) where \( p \) is a prime. We call the primitive element of this field \( z \). The codeword length is \( N = p - 1 \). The DFT representation of the information vector with added redundancy is denoted as \( C = (C_0 C_1 \cdots C_{K_F-1} 0 \cdots 0) \). Hence, the number of information symbols is \( K_F \) and the number of redundant zeros is \( N - K_F \). Suppose there are \( K_T \) nonzero codeword symbols and \( N - K_T \) zero codeword symbols in time domain, such that \( K_F + K_T \geq p + 1 \).

Using the DFT domain encoding method, we have

\[
\begin{pmatrix}
C_0 \\
C_1 \\
\vdots \\
C_{K_T-1}
\end{pmatrix}
= (z^{N-K_T})^T
\begin{pmatrix}
1 & z & z^2 & \cdots & z^{N-1} \\
1 & 1 & 1 & \cdots & 1 \\
1 & z & z^2 & \cdots & z^{N-1} \\
0 & z & z^2 & \cdots & z^{N-1} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & z^{N-1} & z^{2(N-1)} & \cdots & z^{(N-1)(N-1)}
\end{pmatrix}
\begin{pmatrix}
C_0 \\
C_1 \\
\vdots \\
C_{K_T-1}
\end{pmatrix}
\]

9
Considering the zero symbols in the DFT domain vector and codeword, the DFT matrix given in the equation above reduces to one of its principal minors through deleting the last $N - K_T$ rows and $N - K_T$ columns.

\[
(c_0 \ c_1 \ \cdots \ \ c_{K_T-1}|0 \ 0 \ \cdots \ 0) =
\]

\[
(C_0C_1 \cdots C_{K_F-1}|00 \ \cdots \ 0)
\]

We present below the final principal minor of the DFT matrix together with the remaining $N - K_T$ equations in vector form that need to be solved.

\[
(c_0c_1 \cdots c_{K_T-1}) = (C_0C_1 \cdots C_{K_F-1})
\]

and

\[
(00 \ \cdots \ 0) = (C_0C_1 \cdots C_{K_F-1})
\]

In equation 3.3 we presented the relations that $K_F$ information symbols need to satisfy in order to enforce $N - K_T$ redundant symbols in the codeword. The solution of a system of $N - K_T$ equations with $K_F$ unknowns where $K_F \geq N - K_T$ yields $K_T + K_F - N$ free variables while fixing the value of the remaining ones. The information can then be represented by these free
variables. Further, it is useful to investigate the properties of the codewords arising from adding $N - K_F$ redundant zeros to the information vector. For these analysis we recall the definition of the Discrete Fourier Transform. Using the definition of the multiplicative inverse in a Galois field, we have:

$$
\begin{pmatrix}
C_0 & C_1 & \cdots & C_{K_F-1} & 0 & \cdots & 0
\end{pmatrix} = \frac{1}{N} \begin{pmatrix}
c_0 & c_1 & \cdots & c_{K_T-1} & 0 & \cdots & 0
\end{pmatrix} \begin{pmatrix}
1 & 1 & \cdots & 1 & 1 & z^{N-1} & \cdots & z^1 \\
1 & z^{N-1} & \cdots & z^{N-(K_F-1)} & z^{N-K_F} & \cdots & z^1 \\
z^{2(N-1)} & \cdots & z^{2(N-(K_F-1))} & z^{2(N-K_F)} & \cdots & z^2 \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
z^{(K_T-1)(N-1)} & \cdots & z^{(K_T-1)(N-(K_F-1))} & z^{(K_T-1)(N-K_F)} & \cdots & z^{K_T} \\
z^{(N-1)(N-1)} & \cdots & z^{(N-1)(N-(K_F-1))} & z^{(N-1)(N-K_F)} & \cdots & z^{(N-1)}
\end{pmatrix}.
$$

Following the same reasoning as in the previous analysis, the matrix reduces to one of its principal minor by deleting the last $N - K_T$ rows and $N - K_F$ columns.

$$
(C_0 C_1 \cdots C_{K_F-1}|0 0 \cdots 0) = \frac{1}{N} (c_0 c_1 \cdots c_{K_T-1}|0 0 \cdots 0) \times
$$

$$
\begin{pmatrix}
1 & 1 & \cdots & 1 & 1 & z^{N-1} & \cdots & z^1 \\
z^{N-1} & \cdots & z^{N-(K_F-1)} & z^{N-K_F} & \cdots & z^1 \\
z^{2(N-1)} & \cdots & z^{2(N-(K_F-1))} & z^{2(N-K_F)} & \cdots & z^2 \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
z^{(K_T-1)(N-1)} & \cdots & z^{(K_T-1)(N-(K_F-1))} & z^{(K_T-1)(N-K_F)} & \cdots & z^{K_T} \\
z^{(N-1)(N-1)} & \cdots & z^{(N-1)(N-(K_F-1))} & z^{(N-1)(N-K_F)} & \cdots & z^{(N-1)}
\end{pmatrix},
$$

(3.4)

Below we present the final principal minor together with the remaining $N - K_F$ equations in vector notation.

$$
(C_0 C_1 \cdots C_{K_F-1}) = \frac{1}{N} (c_0 c_1 \cdots c_{K_T-1}) \begin{pmatrix}
1 & 1 & \cdots & 1 \\
z^{N-1} & \cdots & z^{N-(K_F-1)} & z^{N-K_F} \\
z^{2(N-1)} & \cdots & z^{2(N-(K_F-1))} & z^{2(N-K_F)} \\
\vdots & \vdots & \ddots & \vdots \\
z^{(K_T-1)(N-1)} & \cdots & z^{(K_T-1)(N-(K_F-1))} & z^{(K_T-1)(N-K_F)} \\
z^{(N-1)(N-1)} & \cdots & z^{(N-1)(N-(K_F-1))} & z^{(N-1)(N-K_F)}
\end{pmatrix},
$$

(3.4)
and

\[
(00 \cdots 0) = \frac{1}{N}(c_0 c_1 \cdots c_{K_T-1}) \begin{pmatrix}
1 & 1 & \cdots & 1 \\
1 & 1 & \cdots & 1 \\
\vdots & \vdots & \ddots & \vdots \\
1 & 1 & \cdots & 1 \\
\end{pmatrix}
\]

Equation 3.5 presents the conditions enforced on the code symbols in order to generate \(N - K_F\) redundant zeros in the DFT representation of the information vector.
Chapter 4

Non-systematic encoding based on the generator polynomial

As previously mentioned, another method of encoding Reed Solomon codes is the non-systematic encoding based on the generator polynomial. The information vector is \( i = (i_0 \ i_1 \ \cdots \ i_{K_T-1}) \) and is represented in time domain using symbols from GF\((p)\). Throughout this report, we will assume that the generator polynomial \( g(x) \) is of the form

\[
g(x) = (x-z)(x-z^2)\cdots(x-z^{N-K_T})
= g_0 + g_1 x + \cdots + g_{N-K_T-1} x^{N-K_T-1} + g_{N-K_T} x^{N-K_T}
\]

with \( g_{N-K_T} = 1 \).

In addition, for a given information vector \( i = (i_0 \ i_1 \ \cdots \ i_{K_T-1}) \) the information polynomial corresponding to this information vector is given by

\[
i(x) = \sum_{j=0}^{K_T-1} i_j x^j.
\]

The code polynomial can be found from the information and generator polynomials as

\[
c(x) = i(x)g(x).
\]

**Example 4.1:** To illustrate the concepts presented above, let us consider the information vector \( i = (2 \ 3) \) with symbols from GF\((5)\) and primitive element \( z = 3 \).

\[
i(x) = \sum_{j=0}^{K_T-1} i_j x^j = i_0 x^0 + i_1 x^1 = 2 + 3x
\]
\[ g(x) = (x - z)(x - z^2) = (x - 3)(x - 4) = 2 + 3x + x^2 \]
\[ c(x) = i(x)g(x) = (2 + 3x)(2 + 3x + x^2) = 4 + 2x + x^2 + 3x^3 \]
\[ c = (4213) \]

For a general information vector with \( K \) symbols from \( \text{GF}(p) \) of order \( N \) we have the following:

\[
\begin{align*}
    i(x) &= i_0 + i_1x + \cdots + i_{K_T-1}x^{K_T-1} \\
    g(x) &= g_0 + g_1x + \cdots + g_{N-K_T-1}x^{N-K_T-1} + x^{N-K_T} \\
    c(x) &= m(x)g(x) \\
    &= i_0g_0 + (i_0g_1 + i_1g_0)x + (i_0g_2 + i_1g_1 + i_2g_0)x^2 + \\
    &\quad + \cdots \cdots + \\
    &\quad + (i_0 + i_1g_{N-K_T-1} + i_2g_{N-K_T-2} + \cdots)x^{N-K_T} + \\
    &\quad + (i_1 + i_2g_{N-K_T-1} + i_3g_{N-K_T-2} + \cdots)x^{N-K_T+1} + \\
    &\quad + \cdots \cdots + \\
    &\quad + (i_{K_T-2} + i_{K_T-1}g_{N-K_T-1})x^{N-2} + \\
    &\quad + (i_{K_T-1})x^{N-1} \quad (4.3)
\end{align*}
\]

Let us consider the following standard form for the code vector:

\[
    \mathbf{c} = (c_0\ c_1\ \cdots\ c_{N-1}) \quad c(x) = c_0 + c_1x + \cdots + c_{N-1}x^{N-1} \quad (4.4)
\]

The above found expression for \( m(x)g(x) \) should equal the code polynomial \( c(x) \).

Summing up, we need the following relations to hold:

\[
\begin{align*}
    c_0 &= i_0g_0 \\
    c_1 &= i_0g_1 + i_1g_0 \\
    c_2 &= i_0g_2 + i_1g_1 + i_2g_0 \\
    \vdots & \vdots \vdots \\
    c_{N-2} &= i_{K_T-2} + i_{K_T-1}g_{N-K_T-1} \\
    c_{N-1} &= i_{K_T-1} \quad (4.5)
\end{align*}
\]

where, if the respective code symbol \( c_i \) is zero, the right hand side equation should be solved for 0 equality. We can clearly see that to enforce \( q \) zeros in
the time domain codeword, it suffices to limit the degree of the information polynomial to \( K_T - 1 - q \). Furthermore, the zeros in the DFT domain codeword are automatically generated by the generator polynomial.

\[
\begin{align*}
c(x) &= i(x)g(x) = i(x)(x - z)(x - z^2) \cdots (x - z^{N-K_T}) \\
C(k) &= \frac{1}{N}c(x = z^{-k})
\end{align*}
\]

Since the multiplicative inverse in Galois field is \( z^{-k} = z^{N-k} \), the coefficients of the DFT domain codeword can be found as shown below.

\[
\begin{align*}
C_0 &= \frac{1}{N}c(x = z^{0}) = i(1)(1 - z)(1 - z^2) \cdots (1 - z^{N-K_T}) \\
C_1 &= \frac{1}{N}c(x = z^{-1}) = i(z^{N-1})(z^{N-1} - z)(z^{N-1} - z^2) \cdots (z^{N-1} - z^{N-K_T}) \\
&\vdots \\
C_{K_T} &= \frac{1}{N}c(x = z^{-K_T}) = i(z^{N-K_T})(z^{N-K_T} - z)(z^{N-K_T} - z^2) \cdots (z^{N-K_T} - z^{N-K_T}) \\
&= 0 \\
&\vdots \\
C_{N-1} &= \frac{1}{N}c(x = z^{-(N-1)}) = i(z)(z - z)(z - z^2) \cdots (z - z^{N-K_T}) = 0
\end{align*}
\]

Summing up, we can say that when using the non-systematic encoding method based on the generator polynomial, the number of zeros in the time domain codeword is directly dependendent on the number of information symbols whereas the number of zeros in the DFT domain codeword is the same as the degree of the generator polynomial.
Chapter 5

Systematic encoding based on the generator polynomial

The codeword in systematic form is

\[ c = (c_0 \ c_1 \ \cdots \ c_{N-1}) = (p_0 \ p_1 \ p_{N-K_T-1} \ i_0 \ i_1 \ \cdots \ i_{K_T-1}), \]

(5.1)

where the first \( N - K_T \) symbols are parity-check symbols \( p_i \), whereas the last \( K_T \) symbols are the information symbols. Parity-check symbols are computed by taking a linear combination of a predetermined subset of information symbols. Referring to equation 5.1, the code polynomial is

\[ c(x) = p_0 + p_1 x + \cdots + p_{N-K_T-1} x^{N-K_T-1} + i_0 x^{N-K_T} + i_1 x^{N-K_T+1} + \cdots + i_{K_T-1} x^{N-1} \]

\[ = p_0 + p_1 x + \cdots + p_{N-K_T-1} x^{N-K_T-1} + x^{N-K_T} (i_0 + i_1 x + \cdots + i_{K_T-1} x^{K_T-1}) \]

\[ = p(x) + x^{N-K_T} i(x). \]

(5.2)

In Chapter 4 the code polynomial was described as a multiple of the generator polynomial.

\[ c(x) = a(x) g(x) \]

\[ = p(x) + x^{N-K_T} i(x) \]

\[ x^{N-K_T} i(x) = a(x) g(x) - p(x) \]

\[ -p(x) = \text{Remainder}(\frac{x^{N-K_T} i(x)}{g(x)}) \]

(5.3)

Let us illustrate the systematic encoding based on the generator polynomial method through the following example in GF(5) with \( z = 3 \).
Example 5.1: We reconsider the information vector $i = (2 \ 3)$ and $g(x) = (x - z)(x - z^2) = x^2 + 3x + 2$.

$$\frac{x^{N-K_T}i(x)}{g(x)} = \frac{x^2(2 + 3x)}{x^2 + 3x + 2} = \frac{3x^3 + 2x^2}{x^2 + 3x + 2} = 3x + 3 \text{ with Remainder } 4$$

(5.4)

Since $-p(x) = \text{Remainder}$, we determine $p(x) = 1$. Therefore the codeword $c$ is $c = (1 \ 0 \ 2 \ 3)$.

Clearly the number of zeros in the time domain codeword is directly controlled by the number of information symbols. Through decreasing the degree of the information vector by $q$, the time domain codeword will have $q$ redundant zeros as well.

On the other hand, the zeros in the DFT domain codeword are automatically generated by the generator polynomial since the code polynomial can always be expressed as a multiple of the generator polynomial. As for the non-systematic encoding method, the DFT domain codeword in the systematic encoding method always has $N - K_T$ redundant zeros.
Chapter 6

Systematic encoding based on the parity-check polynomial

The parity-check polynomial is defined as $h(x) = \frac{x^N - 1}{g(x)} = h_0 + h_1x + \cdots + h_{K_T}x^{K_T}$ where $h_{K_T} = 1$. As mentioned in the previous chapters, the code polynomial can be given as a multiple of the generator polynomial, $c(x) = a(x)g(x)$. When multiplying the code polynomial by the parity-check polynomial we obtain

$$c(x)h(x) = a(x)g(x)h(x)$$
$$= a(x)(x^N - 1)$$
$$= a(x)x^N - a(x). \quad (6.1)$$

Since the degree of the code polynomial is $N - 1$ and the degree of the generator polynomial is $N - K_T$, $a(x)$ should be of degree $K_T - 1$ or less.

$$a(x) = a_0 + a_1x + \cdots + a_{K_T-1}x^{K_T-1} \quad (6.2)$$

Furthermore, when performing the operation $a(x)x^N - a(x)$ we see that the coefficients in front of $x^{K_T}$, $x^{K_T+1}$, $\cdots$ $x^{N-1}$ should equal 0.

$$a(x)x^N - a(x) = (a_0 + a_1x + \cdots + a_{K_T-1}x^{K_T-1})x^N$$
$$- a_0 + a_1x + \cdots + a_{K_T-1}x^{K_T-1}$$
$$= -a_0 - a_1x - \cdots - a_{K_T-1}x^{K_T-1}$$
$$+ a_0x^N + a_1x^{N+1} + \cdots + a_{K_T-1}x^{N+K_T-1}$$

18
\begin{align*}
c(x)h(x) &= (c_0 + c_1 x + \cdots + c_{N-1} x^{N-1}) (h_0 + h_1 x + \cdots + h_{K_T} x^{K_T}) \\
&= c_0 h_0 + (c_0 h_1 + c_1 h_0) x + (c_0 h_2 + c_1 h_1 + c_2 h_0) x^2 + \\
&\quad + \cdots + \\
&\quad + (c_0 h_{K_T} + c_1 h_{K_T-1} + \cdots + c_{K_T} h_0) x^{K_T} + \\
&\quad + (c_1 h_{K_T} + c_2 h_{K_T-1} + \cdots + c_{K_T+1} h_0) x^{K_T+1} + \\
&\quad + \cdots + \\
&\quad + (c_{N-1} h_0 + c_{N-2} h_1 + \cdots + c_{N-K_T-1} h_{K_T}) x^{N-1} \\
&= c_0 h_0 + (c_0 h_1 + c_1 h_0) x + (c_0 h_2 + c_1 h_1 + c_2 h_0) x^2 + \\
&\quad + \cdots + \\
&\quad + (c_{N-1} h_0 + c_{N-2} h_1 + \cdots + c_{N-K_T-1} h_{K_T}) x^{N-1} \\
&= \sum_{i=0}^{N-1} c_i x^i \\
&\text{(6.3)}
\end{align*}

In this encoding method, the information symbols are put in the last \( K_T \) positions of the codeword whereas the first \( N - K_T \) symbols are found through solving the equations arising from equaling the coefficients in front of \( x^{K_T}, x^{K_T+1}, \cdots x^{N-1} \) to 0.

\begin{align*}
c_{N-K_T-1} &= -c_{N-K_T} h_{K_T-1} - \cdots - c_{N-1} h_0 \\
c_{N-K_T-2} &= -c_{N-K_T-1} h_{K_T-1} - \cdots - c_{N-2} h_0 \\
&\vdots \\
c_0 &= -c_1 h_{K_T-1} - \cdots - c_{K_T} h_0 \\
&\text{(6.4)}
\end{align*}

**Example 6.1**: To illustrate this encoding method we consider again GF(5) with \( z = 3 \) and information vector \( \mathbf{i} = (2 3) \).

\begin{align*}
c(x) &= (c_0 \ c_1 \ c_2 \ c_3) = (c_0 \ c_1 \ 2 \ 3) \\
g(x) &= 2 + 3x + x^2 \\
h(x) &= \frac{x^N - 1}{g(x)} = \frac{x^4 - 1}{2 + 3x + x^2} = x^2 + 2x + 2 \\
c_1 &= -c_2 h_1 - c_3 h_0 = -2 \times 2 - 3 \times 2 = 0 \\
c_0 &= -c_1 h_1 - c_2 h_0 = 0 - 2 \times 2 = 1 \\
c &= (1023)
\end{align*}

Clearly since this is a systematic encoding method where the information symbols appear at the end of the codeword, the number of redundant zeros is directly influenced by the degree of the information polynomial. In analogy to the rest of the time domain encoding methods, the DFT domain redundancy is again directly dependent on the degree of the generator polynomial.
Chapter 7

Conclusion

In this report we have shown that all the time domain encoding methods allow a straightforward inclusion of consecutive time and DFT domain redundancy of the codeword.

In Chapter 4 we have illustrated that decreasing the degree of the information vector by $q$, generates $q$ redundant zeros at the upper end of the time domain codeword for the non-systematic encoding method based on the generator polynomial. The code polynomial is found through multiplying the information polynomial with the generator polynomial. The latter consists of a product of linear factors that generate zeros in DFT domain. As a result, the code polynomial consists of the same product of linear factors which generate the zeros in the DFT domain. Therefore, the number of zeros in the DFT domain codeword is the same as the degree of the generator polynomial.

The same reasoning applies to the rest of the time domain encoding methods. As mentioned in chapters 5 and 6, the information symbols appear at the upper end of the time domain codeword for the systematic encoding methods. As a result, redundant zeros are automatically included when decreasing the number of information symbols. Further, since the code polynomial can be expressed as a multiple of the generator polynomial, the number of redundant zeros in the DFT domain codeword is the same as the degree of the generator polynomial.

In Chapter 3 of this report we analyzed the DFT domain encoding method. These analyses revealed the equations that $K_F$ information symbols need to satisfy in order to enforce $N - K_T$ redundant zeros in the time domain code-
word. We concluded that the actual information can be put into $K_T + K_F - N$ symbols while fixing the value of the remaining symbols so that the equations presented below are satisfied.

\[
(00 \cdots 0) = (C_0C_1 \cdots C_{K_F-1}) \begin{pmatrix}
1 & 1 & \cdots & 1 \\
z^{K_T} & z^{K_T+1} & \cdots & z^{N-1} \\
\vdots & \vdots & \ddots & \vdots \\
z^{(K_F-1)K_T} & z^{(K_F-1)(K_T+1)} & \cdots & z^{(K_F-1)(N-1)}
\end{pmatrix}
\]

(7.1)

In addition, we established the equations that $K_T$ codeword symbols should satisfy, in order to enforce $N-K_F$ redundant zeros in the DFT representation of the information vector.

\[
(00 \cdots 0) = \frac{1}{N}(c_0c_1 \cdots c_{K_T-1}) \begin{pmatrix}
1 & 1 & \cdots & 1 \\
z^{N-K_F} & z^{N-(K_F+1)} & \cdots & z^1 \\
z^{2(N-K_F)} & z^{2(N-(K_F+1))} & \cdots & z^2 \\
\vdots & \vdots & \ddots & \vdots \\
z^{(K_T-1)(N-K_F)} & z^{(K_T-1)(N-(K_F+1))} & \cdots & z^{K_T-1}
\end{pmatrix}
\]

(7.2)

As a next step in our investigations, we suggest the analysis of the error correction patterns of RS codes exploiting both the time and DFT domain redundancy.
Bibliography


